

---

Subject: Rendering method software vs hardware

Posted by [Helder Marchetto](#) on Tue, 02 Dec 2014 13:58:36 GMT

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

---

Hi,

I have a software that shows images in one windows and plots in other window (3 at the moment). I have a list widget to change images and when I change images, I recalculate something and update the plots.

I've done this using FG. So far so good...

The problems arise when I use hardware rendering. If I do so, on my laptop, I get once in a while pretty awful crashes (pc crash, win 7 pro, IDL 8.4) and the pc reboots.

Ok, I understand that this is caused by some graphics driver problems. Then I switched to software rendering because I thought that this would help the stability. And it did! But it did so at the cost of performance...

However, according to [http://www.exelisvis.de/docs/Performance\\_Tuning\\_Objec.html](http://www.exelisvis.de/docs/Performance_Tuning_Objec.html) I would expect that hardware rendering is better for 3d stuff. But I just draw images and plot lines!

I used the profiler to have a look at what's happening.

With hardware rendering (crashes permitting) I spend ~ 1.3 seconds in `idlgrmodel::draw` for switching between 4 images. In the same conditions, with software rendering i need for the same method 10.7 seconds.

The weird thing is that the images and plots update in software rendering pretty fast and then the widget is not responding for a second or so.

Does anybody have a clue on what's going or how I can improve this? I really need to switch between images faster and I'm not sure relying on Hardware rendering is a good idea if I want to distribute this...

I'm really starting to like FG, but this is quite a setback I must say.

Cheers,  
Helder

---

---

Subject: Rendering method software vs hardware

Posted by [Helder Marchetto](#) on Tue, 02 Dec 2014 14:35:44 GMT

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

---

Ok, so I spent some time looking for the time draining procedure and found that I set a lot of graphic properties after changing image. This results in a huge waste of time. I'll try to make some sample code to show what I'm having problems with and post tomorrow some code+results.

At the moment, it takes .5 sec to get images and plots shown and 2.5 seconds to set a bunch of properties (hide, color, gridstyle,...)

Cheers, Helder

---

---

Subject: Re: Rendering method software vs hardware

Posted by [natha](#) on Tue, 02 Dec 2014 16:52:28 GMT

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

---

Sometimes it can be very useful to work with the IDLgrBuffer before plotting stuff. The idea is to create all your plots in memory (off-screen) and then convert them to an IDLgrImage. This can help you save some rendering time...

I've got very good results using this method.

Here is the routine I use:

```
function view2image, view

  view->getproperty, viewplane_rect=viewplane_rect, dimensions=dimensions

  aux=get_screen_size(resolution=resolution)

  obuffer=obj_new('idlgrbuffer', dimensions=[viewplane_rect[2],viewplane_rect[3]],
resolution=resolution)
  obuffer->erase, color=[255,255,255]

  obuffer->draw, view
  oimage=obuffer->read()

  obj_destroy, obuffer

  return, oimage
end

nata
```

---

Subject: Re: Rendering method software vs hardware

Posted by [chris\\_torrence@NOSPAM](#) on Tue, 02 Dec 2014 18:48:07 GMT

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

---

On Tuesday, December 2, 2014 7:35:46 AM UTC-7, Helder wrote:

> Ok, so I spent some time looking for the time draining procedure and found that I set a lot of graphic properties after changing image. This results in a huge waste of time. I'll try to make some sample code to show what I'm having problems with and post tomorrow some code+results.

> At the moment, it takes .5 sec to get images and plots shown and 2.5 seconds to set a bunch of properties (hide, color, gridstyle,...)

>

>

> Cheers, Helder

Hi Helder,

Another option is to disable redraws until all of your properties are set. For example:

```
p = PLOT(/TEST)
p.Refresh, /DISABLE
p.COLOR = 'red'
p.LINESTYLE = 'dashed'
...
p.Refresh
```

Hope this helps!

Cheers,  
Chris  
Exelis

---

---

Subject: Re: Rendering method software vs hardware  
Posted by [Helder Marchetto](#) on Wed, 03 Dec 2014 10:02:56 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Tuesday, December 2, 2014 7:48:09 PM UTC+1, Chris Torrence wrote:

> On Tuesday, December 2, 2014 7:35:46 AM UTC-7, Helder wrote:

>> Ok, so I spent some time looking for the time draining procedure and found that I set a lot of graphic properties after changing image. This results in a huge waste of time. I'll try to make some sample code to show what I'm having problems with and post tomorrow some code+results.

>> At the moment, it takes .5 sec to get images and plots shown and 2.5 seconds to set a bunch of properties (hide, color, gridstyle,...)

>>

>>

>> Cheers, Helder

>

> Hi Helder,

>

> Another option is to disable redraws until all of your properties are set. For example:

>

> p = PLOT(/TEST)

> p.Refresh, /DISABLE

> p.COLOR = 'red'

> p.LINESTYLE = 'dashed'

> ...

> p.Refresh

>

> Hope this helps!

>

> Cheers,

> Chris

> Exelis

Hi Chris,  
that worked brilliantly. Thanks.

Question:

when I set:

p.color = 'red' or whatever else...

shouldn't the plot procedure only replot if p.color ne 'red'? In other words, only update a property if this property is different from the previously set property.

This would save quite some time and there would be no need for the refresh, /disable option.

Or is this not that easy?

Cheers,  
Helder

---

Subject: Re: Rendering method software vs hardware

Posted by [chris\\_torrence@NOSPAM](#) on Wed, 03 Dec 2014 16:14:59 GMT

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

---

On Wednesday, December 3, 2014 3:02:57 AM UTC-7, Helder wrote:

> On Tuesday, December 2, 2014 7:48:09 PM UTC+1, Chris Torrence wrote:

>> On Tuesday, December 2, 2014 7:35:46 AM UTC-7, Helder wrote:

>>> Ok, so I spent some time looking for the time draining procedure and found that I set a lot of graphic properties after changing image. This results in a huge waste of time. I'll try to make some sample code to show what I'm having problems with and post tomorrow some code+results.

>>> At the moment, it takes .5 sec to get images and plots shown and 2.5 seconds to set a bunch of properties (hide, color, gridstyle,...)

>>>

>>>

>>> Cheers, Helder

>>

>> Hi Helder,

>>

>> Another option is to disable redraws until all of your properties are set. For example:

>>

>> p = PLOT(/TEST)

>> p.Refresh, /DISABLE

>> p.COLOR = 'red'

>> p.LINESTYLE = 'dashed'

>> ...

>> p.Refresh

>>

>> Hope this helps!

>>

>> Cheers,

>> Chris

>> Exelis

>

> Hi Chris,  
> that worked brilliantly. Thanks.  
> Question:  
> when I set:  
> p.color = 'red' or whatever else...  
> shouldn't the plot procedure only replot if p.color ne 'red'? In other words, only update a  
property if this property is different from the previously set property.  
> This would save quite some time and there would be no need for the refresh, /disable option.  
>  
> Or is this not that easy?  
>  
> Cheers,  
> Helder

Hi Helder,

Yes, in general, we do try to only set the property if it is different. However, for some things like color, testing whether the color is the same can sometimes take just as long as setting it. Again, it depends upon your rendering speed. So we usually just set "quick" properties like color, linestyle, etc., without checking the old value. For more complicated properties (like maybe the map projection), we will check to make sure it needs to be set.

Cheers,  
Chris  
Exelis

---

Subject: Re: Rendering method software vs hardware  
Posted by [mick.mitanirc3](#) on Fri, 12 Dec 2014 21:11:42 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I can tell you that with very few expensive exceptions the GPUs on laptops are weaker than their desktop versions; with too little memory, too little bandwidth and they are prone to overheating when hardware acceleration is specified. It is not just drivers but a lack of internal cooling that causes random freezes and reboots. The reason can be from poor design such as undersized heat sinks, tiny fans, cramped cases and not enough dedicated Graphics RAM or from environmental issues such as blocked vents, dust buildup, and dried out heat transfer compounds.

---