
Subject: NVidia Quadro4 980 XGL card + IDL
Posted by [raval.chintan](#) on Thu, 02 Feb 2006 08:01:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear All,

I am using NVidia Quadro 4 980 XGL. One can visualize stereo images using Quad Buffer , available on this card. Proper interface (functions) for rendering images using this buffer are given in OpenGL.. Quad Buffer includes Left and Right Buffer ,which will be rendered one by one.For Visulization of images one needs, shutter glasses with infrared emitter to be connected with graphics card.

Will it be possible to render an image with use of this buffer on IDL Draw widget? or in other way ,Will it be possible call this interface(functions) of OpenGL through IDL?

In IDL help i found that IDL uses two type of rendering 1) Hardware (OpenGL) 2) Software.
I also found the difference between hardware and software rendering in IDL.

It would be fine if any one helps me in this regard.

Awaiting for Reply...

Regards,
Chintan

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [Rick Towler](#) on Thu, 02 Feb 2006 19:16:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

raval.chintan wrote:

> Dear All,
>
> I am using NVidia Quadro 4 980 XGL. One can visualize stereo images
> using Quad Buffer , available on this card. Proper interface
> (functions) for rendering images using this buffer are given in
> OpenGL.. Quad Buffer includes Left and Right Buffer ,which will be
> rendered one by one.For Visulization of images one needs, shutter
> glasses with infrared emitter to be connected with graphics card.
>
> Will it be possible to render an image with use of this buffer on IDL
> Draw widget? or in other way ,Will it be possible call this
> interface(functions) of OpenGL through IDL?

No, and Yes. IDL's OpenGL implementation is fairly generic and doesn't support many (any?) vendor specific extensions like the quadro's quad-buffer stereo API. So you can't do this in IDL. But, there are ways to do this from within a DLM. I just don't know if I would want to do it :)

You would be better off learning how to control the shutter glasses from within IDL (hack them to run off of the serial port?) and then modify IDL's IDLgrView/IDLgrModel parameters to render the left and right eye views as you run your glasses. My camera object makes rendering the left and right eye views easy. It also has the added bonus of allowing you to easily manipulate your model as you view it.

<http://www.acoustics.washington.edu/~towler/RHTgrCamera.html>

I've played around with this using multiple monitors, Saran wrap (<http://www.sas.org/E-Bulletin/2003-06-20/labNotesAS/body.htm> I) and polarized glasses. I have also used multiple projectors with polarizing lenses. It can be tricky setting up the views/monitors/viewer so you get a "natural" 3d effect but it can be done.

Dick Jackson has worked with this as well and has been quite successful. He could probably give you more tips on setting up views if you decided to go this route.

- > In IDL help i found that IDL uses two type of rendering 1) Hardware
- > (OpenGL) 2) Software.
- > I also found the difference between hardware and software rendering in
- > IDL.

The difference? There is only one? ;)

-Rick

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [Dick Jackson](#) on Thu, 02 Feb 2006 21:40:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Rick Towler" <rick.towler@nomail.noaa.gov> wrote in message
news:<[drntng\\$135\\$1@news.nems.noaa.gov](mailto:drntng$135$1@news.nems.noaa.gov)>...

- >
- > You would be better off learning how to control the shutter glasses from
- > within IDL (hack them to run off of the serial port?) and then modify
- > IDL's IDLgrView/IDLgrModel parameters to render the left and right eye

- > views as you run your glasses. My camera object makes rendering the left
- > and right eye views easy. It also has the added bonus of allowing you to
- > easily manipulate your model as you view it.
- >
- > <http://www.acoustics.washington.edu/~towler/RHTgrCamera.html>
- >
- > I've played around with this using multiple monitors, Saran wrap
- > (<http://www.sas.org/E-Bulletin/2003-06-20/labNotesAS/body.htm>) and
- > polarized glasses. I have also used multiple projectors with polarizing
- > lenses. It can be tricky setting up the views/monitors/viewer so you get
- > a "natural" 3d effect but it can be done.
- >
- > Dick Jackson has worked with this as well and has been quite successful.
- > He could probably give you more tips on setting up views if you decided to
- > go this route.

This OpenGL site helped me to get the exact setup for the two views.

On Stereo Viewing:

<http://www.opengl.org/resources/tutorials/sig99/advanced99/notes/node24.html>

Section on setting up transforms:

<http://www.opengl.org/resources/tutorials/sig99/advanced99/notes/node26.html>

I think Rick's Camera object is great and can easily give you two views from different eye positions looking at the same point in space. (if you use this to generate the stereo views, that last link refers to this as the "alternative, but less correct, method")

Be aware that for perfect stereo rendering, the views should be rendered from different positions (eye positions) looking in the *same* direction (parallel) and then offset on the viewing device so that each eye's "straight ahead" point is directly in front of that eye. Note that this is dependent on display resolution and distance from the user to the display. Tricky business, but the results can be tremendous!

Cheers,

--

-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / <http://www.d-jackson.com>
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [Karl Schultz](#) on Fri, 03 Feb 2006 17:06:46 GMT

On Thu, 02 Feb 2006 22:22:05 -0800, raval.chintan wrote:

> Karl Schultz wrote:

>> On Thu, 02 Feb 2006 00:01:57 -0800, raval.chintan wrote:

>>

>>> Dear All,

>>>

>>> I am using NVidia Quadro 4 980 XGL. One can visualize stereo images

>>> using Quad Buffer , available on this card. Proper interface

>>> (functions) for rendering images using this buffer are given in

>>> OpenGL.. Quad Buffer includes Left and Right Buffer ,which will be

>>> rendered one by one.For Visulization of images one needs, shutter

>>> glasses with infrared emitter to be connected with graphics card.

>>>

>>>

>>> Will it be possible to render an image with use of this buffer on IDL

>>> Draw widget? or in other way ,Will it be possible call this

>>> interface(functions) of OpenGL through IDL?

>>

>> This is not supported in IDL 6.2, nor will it be in 6.3. We're getting

>> enough requests for it that it may be in a release soon.

>>

>> Although it is not documented or supported, you can make OpenGL calls from

>> C-language DLM's if you do things carefully. To do what you want, you'll

>> probably have to subclass IDLgrView with IDL code where you would override

>> the Draw method. You would also write a C DLM that calls glDrawBuffer

>> with GL_BACK_RIGHT.

>>

>> By the time IDLgrView::Draw is called, the correct GL context is in

>> place and IDL has already called glDrawBuffer with GL_BACK (implies

>> GL_LEFT as well). Your new Draw method would call the C DLM code to call

>> the superclass Draw method, call the C DLM code to set GL_BACK_RIGHT, and

>> then call the superclass Draw method again.

>

>

> If i am correct with my limited knowledge of IDL, IDLgrView class does

> not have the draw

> method. IDLgrWindow , IDLgrModel,IDLgrBuffer have draw method. Now the

> question comes that which class's draw method i should override.

> IDLgrWindow , IDLgrModel or IDLgrBuffer? Where will C dlm render the

> Left and right image? Will it be on IDL's window or on C's window?

I did say you need to do it for IDLgrView. Actually, IDLgrScene would probably be more correct, but I tend to not use Scenes or Viewgroups in anything but the more complex programs.

You can't do this by overriding a destination (Window or Buffer) object's

Draw method because you need to sneak in the glDrawBuffer call *after* IDL issued it for the destination object and before it actually starts drawing objects.

You could do it at the grModel level, but the glDrawBuffer call would not take effect until IDL reached that Model in the scene graph, so anything before that would end up in the wrong buffer.

So Scene or View is the Right Place.

The C DLM does nothing but slip in a call to glDrawBuffer at the right place. IDL continues to render as it did before, but just to the GL buffer specified in the glDrawBuffer call.

glDrawBuffer just sets GL state (the current draw buffer). It does not actually perform a Draw on a Buffer.

>
> It would be helpful if you can provide sample code or detailed
> explanation of your suggested technique.

okay....

I don't have any stereo hardware at the moment. So, I'm going to make a grView subclass that renders the scene graph twice into the same destination object, but with a slight offset the second time so that you can see the second rendering.

Notice that only because I'm doing this to the same GL buffer, I need to keep the view from erasing on the second draw, so I set the TRANSPARENT flag. You would not do this in the real stereo case because you would want to go ahead and paint the view with the view background color in the "other" (right) stereo buffer.

```
pro IDLstereoView::Draw, dest
```

```
self->IDLgrView::Draw, dest
self->GetProperty, VIEWPLANE_RECT=vpr, TRANSPARENT=trans
vprsave = vpr
vpr[0] += 0.1
self->SetProperty, VIEWPLANE_RECT=vpr, TRANSPARENT=1
;; call DLM that calls glDrawBuffer(GL_BACK_RIGHT) here!
self->IDLgrView::Draw, dest
self->SetProperty, VIEWPLANE_RECT=vprsave, TRANSPARENT=trans
end
```

```
pro IDLstereoView__define
  struct = { IDLstereoView, $
    INHERITS IDLgrView}
end
```

test program:

```
pro test_stereo_view
```

```
oWin = OBJ_NEW('IDLgrWindow')
oView = OBJ_NEW('IDLstereoView')
oModel = OBJ_NEW('IDLgrModel')
oPolyline = OBJ_NEW('IDLgrPolyline', [[-0.9,-0.9],[0.9,0.9]])
```

```
oView->Add, oModel
oModel->Add, oPolyline
```

```
oWin->Draw, oView
```

```
end
```

All you'd have to do to try the stereo stuff is to code a DLM C procedure that does absolutely nothing else except:

```
glDrawBuffer(GL_BACK_RIGHT);
```

and call it from where I indicated above. You don't need to "put things back" by calling `glDrawBuffer(GL_BACK_LEFT)` after the second draw because IDL will do that on the next draw. But it may be good form to do so.

Oh, and you'd have to figure out the eye transform stuff (see other posts in this thread) and how to activate your hardware to go into stereo mode.

Please refer to the External Design Guide and the accompanying samples for more information on the DLM.

Karl

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [Rick Towler](#) on Fri, 03 Feb 2006 19:00:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

raval.chintan@gmail.com wrote:
> Karl Schultz wrote:

```

>> On Thu, 02 Feb 2006 00:01:57 -0800, raval.chintan wrote:
>>
>>> Dear All,
>>>
>>> I am using NVidia Quadro 4 980 XGL. One can visualize stereo images
>>> using Quad Buffer , available on this card. Proper interface
>>> (functions) for rendering images using this buffer are given in
>>> OpenGL.. Quad Buffer includes Left and Right Buffer ,which will be
>>> rendered one by one.For Visulization of images one needs, shutter
>>> glasses with infrared emitter to be connected with graphics card.
>>>
>>>
>>> Will it be possible to render an image with use of this buffer on IDL
>>> Draw widget? or in other way ,Will it be possible call this
>>> interface(functions) of OpenGL through IDL?
>> This is not supported in IDL 6.2, nor will it be in 6.3. We're getting
>> enough requests for it that it may be in a release soon.
>>
>> Although it is not documented or supported, you can make OpenGL calls from
>> C-language DLM's if you do things carefully. To do what you want, you'll
>> probably have to subclass IDLgrView with IDL code where you would override
>> the Draw method. You would also write a C DLM that calls glDrawBuffer
>> with GL_BACK_RIGHT.
>>
>> By the time IDLgrView::Draw is called, the correct GL context is in
>> place and IDL has already called glDrawBuffer with GL_BACK (implies
>> GL_LEFT as well). Your new Draw method would call the C DLM code to call
>> the superclass Draw method, call the C DLM code to set GL_BACK_RIGHT, and
>> then call the superclass Draw method again.
>
>
> If i am correct with my limited knowledge of IDL, IDLgrView class does
> not have the draw
> method. IDLgrWindow , IDLgrModel,IDLgrBuffer have draw method. Now the
> question comes that which class's draw method i should override.
> IDLgrWindow , IDLgrModel or IDLgrBuffer? Where will C dlm render the
> Left and right image? Will it be on IDL's window or on C's window?

```

All IDLgr* atoms have a draw method. When you call IDLgrWindow::Draw, it in turn calls IDLgrView::Draw which in turn calls IDLgrModel::Draw which in turn calls IDLgrPolygon::Draw and so on.

See for yourself. Create a subclass of IDLgrView and override the Draw method like so:

```

pro MyView::Draw, destObject

    print, 'Well I know my name is Simon, ' $

```

```
'and I like to do drawings...'  
self -> IDLgrView::Draw, destObject
```

end

Now every time you draw this view, it will print out some text.

> It would be helpful if you can provide sample code or detailed
> explanation of your suggested technique.
>

I *strongly* suggest that you pursue the method I suggested. The method Karl outlined isn't for the faint of heart. Especially when there is an alternative method that will be easier to implement. There *may* be some performance benefits to using the Stereo API but the work involved doesn't justify pursuit of that method without at least trying my method. IMO, of course.

But if you really want to do this the hard way, Karl posted code that demonstrates his technique of making OpenGL calls from a DLM on the user contrib site. It is in the advanced visualization section, the file is IDL_VGL.zip.

-Rick

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [raval.chintan](#) on Thu, 09 Feb 2006 06:26:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Karl,

Thanks for your tip. I was able to create the dlm successfully and the calls to `glDrawBuffer(GL_BACK_RIGHT)` were also happening but could not see the stereo rendering then I found that the stereo mode was not getting switched on and a call to

```
sprintf(szMessage,  
        "Current OpenGL Driver:\n%s\n%s\n%s\n ",  
glGetString(GL_VENDOR),  
glGetString(GL_RENDERER),  
glGetString(GL_VERSION),  
)  
glGetBooleanv(GL_STEREO, &bStereo)
```

returned null values for vendor, renderer and version and returned

GL_FALSE for GL_STEREO flag.

Now my question is (sorry for asking too many questions but I hear the folks here are tolerant and are kind) how do I turn the stereo mode programmatically ?

if we use PixelFormatDescriptor than we need an handle to the window or the context how can I get handle to IDL's window in dlm or is it that I am completely going on wrong track here???

Desperatly *seeking* answers
Chintan

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [Karl Schultz](#) on Thu, 09 Feb 2006 16:05:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 08 Feb 2006 22:26:27 -0800, raval.chintan wrote:

> Dear Karl,
>
> Thanks for your tip. I was able to create the dlm successfully and the
> calls to glDrawBuffer(GL_BACK_RIGHT) were also happening but could not
> see the stereo rendering then I found that the stereo mode was not
> getting switched on and a call to
>
> sprintf(szMessage,
> "Current OpenGL Driver:\n%s\n%s\n%s\n",
> glGetString(GL_VENDOR),
> glGetString(GL_RENDERER),
> glGetString(GL_VERSION),
>)
> glGetBooleanv(GL_STEREO, &bStereo)
>
> returned null values for vendor, renderer and version and returned
> GL_FALSE for GL_STEREO flag.

Getting null values here is bad. Are you making the glGetString calls while an IDLgrWindow Draw is in progress? In other words, are you calling this from a Draw method that you have overridden in a subclass?

Null values indicate that a GL context is not current and you are making these calls "out of band". You can't just make these calls from any point in your IDL program. They must be made while a draw in is progress.

>
> Now my question is (sorry for asking too many questions but I hear the

> folks here are tolerant and are kind) how do I turn the stereo mode
> programmatically ?

This I do not know at the moment. How do you do it in a stand-alone GL application (outside of IDL)?? I'd have to research this, but maybe I'll leave it to you. :-) Of course, I'll figure it out when/if we add this to IDL. :-)

In any case, you need to address the above context problem first.

>
> if we use Pixelformatdescriptor than we need an handle to the window or
> the context how can I get handle to IDL's window in dlm or is it that I
> am completely going on wrong track here???

Well, if this ends up needing to use a different PFD, then we are sunk. IDL does that deep inside and you cannot influence PFD selection. Further, you cannot change the PFD on a window once it is made current. I notice that there is a PFD_STEREO flag, but the Microsoft docs say this isn't supported. It may be that some ICD's support or pay attention to it. I know IDL does NOT set this flag when selecting a PFD, so if a PFD with PFD_STEREO set is required to activate stereo for your card, it isn't going to happen until RSI formally supports stereo.

>
> Desperatly *seeking* answers

Well, I never said that this would work out, and we gave it a try. IDL 6.2 and 6.3 simply does not support stereo and we're just trying to hack around it, and it doesn't look good now. But maybe you might still get someplace. If stereo support is really important to you, let RSI know about it.

Karl

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [Rick Towler](#) on Thu, 09 Feb 2006 18:52:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chintan,

While I very much respect doing it the hard way (the journey is almost always more enjoyable than the destination), I'll second my offer, and sweeten the deal.

I have modified my camera object so it now offers stereo support. At this time it has only been tested with shutter glasses but would work

equally well with dual-monitor or dual-projector setups.

Some caveats:

As I suggested in a previous email, the shutter glasses are controlled via the serial port, **not** the glasses port on your video card. This is good, in that it allows anyone with access to the glasses to use them, but bad in that you have to build a little circuit to interface the glasses to the serial port. It is **very** simple, and costs only a few dollars (a couple of LM301 or similar op-amps, a cap, and a couple of resistors). This approach also allows multiple glasses to be controlled by the pc.

The dlm I have written for controlling the glasses is for windows. If someone wants to do this on another platform they need to modify the dlm. It is very simple, and should be trivial.

If you are using shutter glasses, you need a good CRT monitor with a high refresh rate. I forgot this when testing the glasses and couldn't trace down the sync problem I was having... LCD monitors are too slow.

I have only done limited testing. Further testing needs to be done to confirm that the asymmetric frustum projection is correct. In IDL we don't have access to the perspective projection matrix, so I apply a "correction" to the modelview matrix. I think this works... If it doesn't there are other ways to fake this, just not as clean.

The code needs to be cleaned up and documented so I'm not posting it right now but you are welcomed to a copy, along with the circuit diagram.

-Rick

Karl Schultz wrote:

```
> On Wed, 08 Feb 2006 22:26:27 -0800, raval.chintan wrote:
>
>> Dear Karl,
>>
>> Thanks for your tip. I was able to create the dlm successfully and the
>> calls to glDrawBuffer(GL_BACK_RIGHT) were also happening but could not
>> see the stereo rendering then I found that the stereo mode was not
>> getting switched on and a call to
>>
>> sprintf(szMessage,
>>         "Current OpenGL Driver:\n%s\n%s\n%s\n ",
>>         glGetString(GL_VENDOR),
>>         glGetString(GL_RENDERER),
>>         glGetString(GL_VERSION),
```

```
>> )
>> glGetBooleanv(GL_STEREO, &bStereo)
>>
>> returned null values for vendor, renderer and version and returned
>> GL_FALSE for GL_STEREO flag.
>
> Getting null values here is bad. Are you making the glGetString calls
> while an IDLgrWindow Draw is in progress? In other words, are you calling
> this from a Draw method that you have overridden in a subclass?
>
> Null values indicate that a GL context is not current and you are making
> these calls "out of band". You can't just make these calls from any point
> in your IDL program. They must be made while a draw in is progress.
>
>> Now my question is (sorry for asking too many questions but I hear the
>> folks here are tolerant and are kind ) how do I turn the stereo mode
>> programmatically ?
>
> This I do not know at the moment. How do you do it in a stand-alone GL
> application (outside of IDL)?? I'd have to research this, but maybe I'll
> leave it to you. :-) Of course, I'll figure it out when/if we add this to
> IDL. :-)
>
> In any case, you need to address the above context problem first.
>
>> if we use Pixelformatdescriptor than we need an handle to the window or
>> the context how can I get handle to IDL's window in dlm or is it that I
>> am completely going on wrong track here???
```

>

```
> Well, if this ends up needing to use a different PFD, then we are sunk.
> IDL does that deep inside and you cannot influence PFD selection.
> Further, you cannot change the PFD on a window once it is made current.
> I notice that there is a PFD_STEREO flag, but the Microsoft docs say this
> isn't supported. It may be that some ICD's support or pay attention to
> it. I know IDL does NOT set this flag when selecting a PFD, so if a PFD
> with PFD_STEREO set is required to activate stereo for your card, it isn't
> going to happen until RSI formally supports stereo.
>
>> Desperatly *seeking* answers
>
> Well, I never said that this would work out, and we gave it a try. IDL
> 6.2 and 6.3 simply does not support stereo and we're just trying to hack
> around it, and it doesn't look good now. But maybe you might still get
> someplace. If stereo support is really important to you, let RSI know
> about it.
>
> Karl
>
```

Subject: Re: NVidia Quadro4 980 XGL card + IDL
Posted by [raval.chintan](#) on Tue, 14 Feb 2006 11:50:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rick,Karl, Dick

Thank you very much for your *valuable* support.

I have find out a temporary solution. In this solution i am using JAVA, JAI and JAVA3D and *of course* IDL.

I have used IDL java bridge for this application. In my application i am rendering stereo on Java's window. For rendering I am passing my data from IDL.

And of course this a temporary solution. It will be very nice if RSI can provide this facility in IDL's future version.

As Rick has said that Journey is more *enjoyable* rather than destination which is actually true.

Rick - If you can provide me a code and circuit diagram , then it would be very helpful. Awaiting for your reply.

-Chintan
