Subject: Re: VTK + IDL
Posted by Michael Wallace on Wed, 25 Aug 2004 03:24:21 GMT
View Forum Message <> Reply to Message

It sounds from your description that you should just be able to use IDL and not have to worry about OpenGL at all. When I said "IDL doesn't have all of the graphics constructs" what I meant to say was doesn't allow you direct access into the OpenGL API. However, IDL does have a lot of graphics for analysis work. I don't know what VTK is, but medical images were one of the main things that IDL was developed for. If all you're doing is graphics display of medical data sets, IDL should more than cover you.

The way I think of IDL vs. OpenGL is like this: IDL = analysis/visualization; OpenGL = video games/very complex visualization.

-Mike

Hee Chun wrote:
> Hello,
> I learned from the previous post(from Mike) that:
> 'OpenGL doesn't have the scientific analysis capabilities of IDL and
> IDL
> doesn't have the graphics constructs of OpenGL'.
> But I need both analysis capabilities and graphics display for the
> medical image data sets.
> Is there any way to interface between VTK and IDL using DLM? If there
> is, Is it very hard to implement it? What kinds of problems do I need
> to consider? Where is the good starting point at least to open the 3d
> object rendered by VTK on the draw window of IDL?
>
> Any tips/suggestions/advices will be greatly appreciated.
>
> Thanks.
>
> HC

Subject: Re: VTK + IDL
Posted by David Fanning on Wed, 25 Aug 2004 03:28:45 GMT
View Forum Message <> Reply to Message

Hee Chun writes:

> But I need both analysis capabilities and graphics display for the
> medical image data sets.

I think you might find IDL has plenty enough graphics
display capability for most medical image data sets. :-)

Cheers,

David

P.S. At least I could point you to an awful lot of
customers who think so.

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: VTK + IDL
Posted by Karl Schultz on Wed, 25 Aug 2004 14:33:38 GMT
View Forum Message <> Reply to Message

"Hee Chun" <chun.42@osu.edu> wrote in message
news:da6f35c7.0408241737.7a23a499@posting.google.com...
> Hello,
> I learned from the previous post(from Mike) that:
> 'OpenGL doesn't have the scientific analysis capabilities of IDL and
> IDL
> doesn't have the graphics constructs of OpenGL'.
> But I need both analysis capabilities and graphics display for the
> medical image data sets.

I'll echo the other replies and say that IDL should have most of what you
would need. Particularly with IDL 6.1, *most* of the OpenGL features are
now exposed via Object Graphics.

> Is there any way to interface between VTK and IDL using DLM? If there
> is, Is it very hard to implement it?

I've never done it or have heard of anyone who has. I think that it should
be possible because VTK exposes an API that can be used from a C/C++
program. So, you should be able to code a DLM that implements some set of
IDL function/procedures that turn around and call VTK functions.

I have not done an IDL-VTK graphics comparison lately, but I can't imagine
that VTK graphics capabilites are that different from IDL's in terms of pure
graphics features. There may be some differences in higher-level
visualization techniques and features. If I found myself wanting to call
VTK from IDL, I would think that it might be for using some non-graphical
analysis function in VTK that does not exist in IDL.

> What kinds of problems do I need
> to consider?

If I remember right, VTK exposes a C++ API. So, your DLM would have to be coded in C++. IDL calls DLM's using a C interface. No problem here, just something to keep in mind.

> Where is the good starting point at least to open the 3d
> object rendered by VTK on the draw window of IDL?

You would need two things for this -

1) A way to get an OS-level window ID for the window that IDL created in a WIDGET_DRAW widget.
2) A way to pass this window ID to VTK and tell it to use that instead of creating its own.

I am pretty sure that (1) does not exist, and the VTK docs would indicate if (2) is possible.

I once accomplished something like this, but in a slightly different manner:

If you can get an external library like VTK to make OpenGL calls without making its own GL context the current context, then you can call the library during the execution of the Window's Draw method. You would do this by implementing a subclass of one of the object graphics objects, say IDLgrPolygon. Override that object's Draw method with code that calls your DLM that in turn calls VTK to draw the VTK object, say an isosurface or somesuch.

This approach is really tricky and NOT supported at all by RSI. You can see an example of it in the RSI user contrib library on the RSI website. I put an example there that calls a Volume graphics library (VGL).

But I think that this will be hard to make work with VTK because I doubt that VTK exposes an entry point that will just make blind OpenGL calls without making its own GL context current or otherwise somehow relying on its ownership of the window. I could be wrong about that - you might find a low-level "draw myself" method on some VTK primitive you are interested in that would normally be called by a VTK scene graph traverser.

Karl

Subject: Re: VTK + IDL
Posted by David Fanning on Wed, 25 Aug 2004 14:52:06 GMT

Karl Schultz writes:

> But I think that this will be hard to make work with VTK because I doubt
> that VTK exposes an entry point that will just make blind OpenGL calls
> without making its own GL context current or otherwise somehow relying on
> its ownership of the window. I could be wrong about that - you might find a
> low-level "draw myself" method on some VTK primitive you are interested in
> that would normally be called by a VTK scene graph traverser.

Yeah, good luck! :-(

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: VTK + IDL
Posted by Chris Lee on Wed, 25 Aug 2004 14:59:56 GMT

In article <da6f35c7.0408241737.7a23a499@posting.google.com>, "Hee Chun"
<chun.42@osu.edu> wrote:

> Hello,
> I learned from the previous post(from Mike) that: 'OpenGL doesn't have
> the scientific analysis capabilities of IDL and IDL
> doesn't have the graphics constructs of OpenGL'. But I need both
> analysis capabilities and graphics display for the medical image data
> sets.
> Is there any way to interface between VTK and IDL using DLM? If there
> is, Is it very hard to implement it? What kinds of problems do I need to
> consider? Where is the good starting point at least to open the 3d
> object rendered by VTK on the draw window of IDL? Any
> tips/suggestions/advices will be greatly appreciated. Thanks.
> HC

IDL can do a lot of what VTK appears to do. I.e generate 2d and 3d
images/plots. If you wanted to draw a surface for example, you would you
SHADE_SURF

;make some data

```
x=(findgen(100)-50.)#replicate(1,100)
y=transpose(x)
r=sqrt(x^2+y^2)
r=r*10.*!pi/180.
d=(cos(r)*exp(-r^2/50.))

;draw the data
shade_surf, d, ax=50.
```

IDL works a bit differently to VTK, instead of making analytic functions
and sampling them (this is what MATLAB does, right?), you make the
data first (or read the data) and make the function out of the data.

IDL is VTK with added data analysis. PV-WAVE, the poor ,
disowned sibling of IDL, uses VTK as the graphics backend (hidden beneath
layers of API). If you want to draw some 3d objects in IDL you can,
you'll probably need to learn the object graphics part of IDL, which
probably isn't any harder than learning the VTK API.

Have a look at the idldemo if you're still unsure. There are a few 3d
isosurface and medical mri images there to play with. Have a look at
David's website, http://dfanning.com/ , the IDL programs section has some
imaging examples.

Chris.

Subject: Re: VTK + IDL
Posted by chun.42 on Wed, 25 Aug 2004 15:33:21 GMT
View Forum Message <> Reply to Message

Michael Wallace <mwallace.no.spam@no.spam.swri.edu.invalid> wrote in message
news:<10io1j5s450cf21@corp.supernews.com>...
> It sounds from your description that you should just be able to use IDL
> and not have to worry about OpenGL at all. When I said "IDL doesn't
> have all of the graphics constructs" what I meant to say was doesn't
> allow you direct access into the OpenGL API.

Thanks Mike for your reply. I am using IDL and I know that IDL DOES
have some graphics functionalities for medical image data sets. I just
borrowed from your comment to explain why I need IDL and OpenGL API.
If I misused it, sorry for that.

> However, IDL does have a
> lot of graphics for analysis work. I don't know what VTK is, but
> medical images were one of the main things that IDL was developed for.
> If all you're doing is graphics display of medical data sets, IDL should
> more than cover you.

VTK is the Visualization Tool Kit from Http://www.kitware.com
Maybe I am a beginner for IDL's object graphics. I tried the following
code but the image is not good enough for my application.

```
file = Filepath('head.dat',Subdirectory=['examples',$
  'data'])
 volumeData = READ_BINARY(file, DATA_DIMS = [80, 100, 57])
oVol =  Obj_new('IDLgrVolume',volumeData,/Zbuff,Hints=2,/No_copy,/Ze ro_opacity_skip)
-- normalization --
oModel = Obj_new('IDLgrModel')
 oView = obj_new('IDLgrView', viewplane_rect=[-1.,-1.,2.5,2.5], $
   color=[128,128,128],ZClip=[2.0,-2.0], Eye=60.0,Projection=2  )

 oModel ->Add, oVol;
 oView -> Add, oModel;
 oDrawWin -> Draw,oView;
```

So, I come up with the idea calling VTK from IDL. What do I need to do
more to display the image better? If you know who is using IDL's
object a lot, please let me know. Thanks.

HC

>
> Hee Chun wrote:
>> Hello,
>> I learned from the previous post(from Mike) that:
>> 'OpenGL doesn't have the scientific analysis capabilities of IDL and
>> IDL
>> doesn't have the graphics constructs of OpenGL'.
>> But I need both analysis capabilities and graphics display for the
>> medical image data sets.
>> Is there any way to interface between VTK and IDL using DLM? If there
>> is, Is it very hard to implement it? What kinds of problems do I need
>> to consider? Where is the good starting point at least to open the 3d
>> object rendered by VTK on the draw window of IDL?
>>
>> Any tips/suggestions/advices will be greatly appreciated.
>>
>> Thanks.
>>
>> HC

Subject: Re: VTK + IDL
Posted by Rick Towler on Wed, 25 Aug 2004 16:48:55 GMT

Karl Schultz wrote:

> You would need two things for this -
>
> 1) A way to get an OS-level window ID for the window that IDL created in a
> WIDGET_DRAW widget.
> 2) A way to pass this window ID to VTK and tell it to use that instead of
> creating its own.
>
> I am pretty sure that (1) does not exist, and the VTK docs would indicate if
> (2) is possible.

I'll join the chorus in saying that IDL can probably do all you need it to do and that I would think you would be crazy to write an IDL->VTK .dlm. But then again, I know I have made my share of questionable project design decisions because I couldn't resist a challenge :)

To Karl's point, (1) exists in the win32 api. The GetActiveWindow() function will return (you guessed it) a handle to the active window. If you write your IDL code such that you call this function in your .dlm right after the window is realized then you're pretty safe. I use this in my directInput .dlm.

Good luck. If you get something working you'll surely post the code, right? ;)

-Rick

Subject: Re: VTK + IDL
Posted by David Fanning on Wed, 25 Aug 2004 16:57:36 GMT

Hee Chun writes:

> VTK is the Visualization Tool Kit from Http://www.kitware.com
> Maybe I am a beginner for IDL's object graphics. I tried the following
> code but the image is not good enough for my application.
>
> file = Filepath('head.dat',Subdirectory=['examples',$
> 'data'])
> volumeData = READ_BINARY(file, DATA_DIMS = [80, 100, 57])
> oVol = Obj_new('IDLgrVolume',volumeData,/Zbuff,Hints=2,/No_copy,/Ze ro_opacity_skip)
> -- normalization --

> oModel = Obj_new('IDLgrModel')
> oView = obj_new('IDLgrView', viewplane_rect=[-1.,-1.,2.5,2.5], $
> color=[128,128,128],ZClip=[2.0,-2.0], Eye=60.0,Projection=2)
>
> oModel ->Add, oVol;
> oView -> Add, oModel;
> oDrawWin -> Draw,oView;
>
> So, I come up with the idea calling VTK from IDL. What do I need to do
> more to display the image better?

If looks are what you are after, I would try setting
the INTERPOLATE keyword on your volume object.

> If you know who is using IDL's object a lot, please let me know.

Why, *everyone* is using objects a lot. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: VTK + IDL
Posted by Karl Schultz on Wed, 25 Aug 2004 18:17:55 GMT
View Forum Message <> Reply to Message

"Rick Towler" <rtowler@xxx.u.washington.edu> wrote in message
news:cgifuc$6pt$1@gnus01.u.washington.edu...
> Karl Schultz wrote:
>
>
>> You would need two things for this -
>>
>> 1) A way to get an OS-level window ID for the window that IDL created in
a
>> WIDGET_DRAW widget.
>> 2) A way to pass this window ID to VTK and tell it to use that instead
of
>> creating its own.
>>
>> I am pretty sure that (1) does not exist, and the VTK docs would
indicate if

>> (2) is possible.
>
>
> I'll join the chorus in saying that IDL can probably do all you need it
> to do and that I would think you would be crazy to write an IDL->VTK
> .dlm. But then again, I know I have made my share of questionable
> project design decisions because I couldn't resist a challenge :)
>
> To Karl's point, (1) exists in the win32 api. The GetActiveWindow()
> function will return (you guessed it) a handle to the active window. If
> you write your IDL code such that you call this function in your .dlm
> right after the window is realized then you're pretty safe. I use this
> in my directInput .dlm.

Neat. I was actually going to mumble-post something about using X11 calls to root around in the X window tree, looking for the drawing window, but then couldn't think of the equivalent sort of thing for Windows. Do you call the dlm from an event handler for something like an expose event? It seems like you would stand a better chance of getting the right window if so.

Karl

Subject: Re: VTK + IDL
Posted by Karl Schultz on Wed, 25 Aug 2004 18:21:00 GMT
View Forum Message <> Reply to Message

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.1b9655853c8250989854@news.frii.com...
> Karl Schultz writes:
>
>> But I think that this will be hard to make work with VTK because I doubt
>> that VTK exposes an entry point that will just make blind OpenGL calls
>> without making its own GL context current or otherwise somehow relying
on
>> its ownership of the window. I could be wrong about that - you might
find a
>> low-level "draw myself" method on some VTK primitive you are interested
in
>> that would normally be called by a VTK scene graph traverser.
>
> Yeah, good luck! :-(

Well, yeah. I probably shouldn't have mentioned it. Just because it worked for VGL, doesn't mean it would for VTK. It does, however, point out that you can call a DLM that makes OpenGL calls, using the IDL context.

I have a feeling that the OP really wants to call VTK to have it do more
than just draw stuff.

Karl

Subject: Re: VTK + IDL
Posted by David Fanning on Wed, 25 Aug 2004 18:26:08 GMT
View Forum Message <> Reply to Message

Karl Schultz writes:

> I have a feeling that the OP really wants to call VTK to have it do more
> than just draw stuff.

I have a feeling he doesn't *really* want to learn IDL. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: VTK + IDL
Posted by Karl Schultz on Wed, 25 Aug 2004 18:30:48 GMT
View Forum Message <> Reply to Message

"Hee Chun" <chun.42@osu.edu> wrote in message
news:da6f35c7.0408250733.1528c0d2@posting.google.com...
> Michael Wallace <mwallace.no.spam@no.spam.swri.edu.invalid> wrote in
message news:<10io1j5s450cf21@corp.supernews.com>...
>> It sounds from your description that you should just be able to use IDL
>> and not have to worry about OpenGL at all. When I said "IDL doesn't
>> have all of the graphics constructs" what I meant to say was doesn't
>> allow you direct access into the OpenGL API.
>
> Thanks Mike for your reply. I am using IDL and I know that IDL DOES
> have some graphics functionalities for medical image data sets. I just
> borrowed from your comment to explain why I need IDL and OpenGL API.
> If I misused it, sorry for that.
>
>> However, IDL does have a
>> lot of graphics for analysis work. I don't know what VTK is, but

>> medical images were one of the main things that IDL was developed for.
>> If all you're doing is graphics display of medical data sets, IDL should
>> more than cover you.
>
> VTK is the Visualization Tool Kit from Http://www.kitware.com
> Maybe I am a beginner for IDL's object graphics. I tried the following
> code but the image is not good enough for my application.
>
> file = Filepath('head.dat',Subdirectory=['examples',$
> 'data'])
> volumeData = READ_BINARY(file, DATA_DIMS = [80, 100, 57])
> oVol =
Obj_new('IDLgrVolume',volumeData,/Zbuff,Hints=2,/No_copy,/Ze ro_opacity_skip)
> -- normalization --
> oModel = Obj_new('IDLgrModel')
> oView = obj_new('IDLgrView', viewplane_rect=[-1.,-1.,2.5,2.5], $
> color=[128,128,128],ZClip=[2.0,-2.0], Eye=60.0,Projection=2)
>
> oModel ->Add, oVol;
> oView -> Add, oModel;
> oDrawWin -> Draw,oView;
>
> So, I come up with the idea calling VTK from IDL. What do I need to do
> more to display the image better? If you know who is using IDL's
> object a lot, please let me know. Thanks.
>
> HC

OK, now we're talking about volume rendering, which has little to do with
OpenGL.

IDL uses an internal ray-casting algorithm to produce a 2D image (along with
Z-buffer data) of the volume rendering and then draws the image. It doesn't
use OpenGL for volume rendering, except to draw this 2D image to the screen.
And OpenGL itself does not support volume rendering, except if you consider
3D texture mapping as such.

VTK has a lot of volume rendering capabillity and in fact, volume vis is a
huge focus of VTK.

If you tell us more about *why* the IDL volume rendering is not good enough
for your application, maybe we can be more helpful.

Karl

Subject: Re: VTK + IDL

Posted by Rick Towler on Wed, 25 Aug 2004 20:31:51 GMT
View Forum Message <> Reply to Message

Karl Schultz wrote:

> "Rick Towler" wrote in message
>
>> Karl Schultz wrote:
>>
>>> You would need two things for this -
>>>
>>> 1) A way to get an OS-level window ID for the window that IDL created
>>> 2) A way to pass this window ID to VTK and tell it to use it
>>>
>>> I am pretty sure that (1) does not exist

>> To Karl's point, (1) exists in the win32 api. The GetActiveWindow()
>> function will return (you guessed it) a handle to the active window. If
>> you write your IDL code such that you call this function in your .dlm
>> right after the window is realized then you're pretty safe. I use this
>> in my directInput .dlm.
>
>
> Neat. I was actually going to mumble-post something about using X11 calls
> to root around in the X window tree, looking for the drawing window, but
> then couldn't think of the equivalent sort of thing for Windows. Do you
> call the dlm from an event handler for something like an expose event? It
> seems like you would stand a better chance of getting the right window if
> so.

Since the handle won't change over the lifetime of the window I find it
easier to grab it right after the base is realized. Then there aren't
any wasted calls when multiple expose events are triggered.

I suppose I could automate things by writing it as a stub widget and
grabbing the handle during a "realize" callback but aside from the fun
of doing it, it isn't worth it.

-Rick

Subject: Re: VTK + IDL
Posted by chun.42 on Thu, 26 Aug 2004 02:29:13 GMT
View Forum Message <> Reply to Message

Rick Towler <rtowler@xxx.u.washington.edu> wrote in message >

> To Karl's point, (1) exists in the win32 api. The GetActiveWindow()

> function will return (you guessed it) a handle to the active window. If
> you write your IDL code such that you call this function in your .dlm
> right after the window is realized then you're pretty safe. I use this
> in my directInput .dlm.

Thanks for pointing it out.

> Good luck. If you get something working you'll surely post the code,
> right? ;)
>
> -Rick

I am still wondering that I can do it in time, and I am waiting for
the response from the idl users.

HC
