Subject: VTK + IDL
Posted by chun.42 on Wed, 25 Aug 2004 01:37:29 GMT
View Forum Message <> Reply to Message

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 chun.42 on Thu, 26 Aug 2004 01:47:05 GMT
View Forum Message <> Reply to Message

"Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message > >
> OK, now we're talking about volume rendering, which has little to do with
> OpenGL.

I used VTK a little bit and I know that VTK uses the OpenGL's library.
But it turns out I am in the wrong track.
If I misled you/IDL users, sorry about that again.

> 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

I want to have the decent-layered image
object(ex,tissue,bone,organ)from the volumetric data sets with
acceptable rendering speed. So I can distinguish/differentiate the
lesion from the normal tissue with some colors. There is an example
from the website: http://www.kitware.com
It is the transparent half-head image on the right bottom. That is not
the exact image I want, but I need that kind of image.
Also, it will be fantastic to have a data(volume image intensity)
picking capability to the layered object with trackball movement. So,

I can use it for modeling of drug delivery to the lesion in the quantitive way by using IDL. I know IDL has the data picking capability. How can I do that? Please, let me know that.

Thanks.

HC

Subject: Re: VTK + IDL
Posted by andrew.cool on Sun, 29 Aug 2004 23:07:00 GMT
View Forum Message <> Reply to Message

chun.42@osu.edu (Hee Chun) wrote in message news:<da6f35c7.0408251829.46882913@posting.google.com>...
> 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

Hi,

Perhaps a picture is still worth a dozen posts? Check out www.paulsorenson.com for some IDL medical visualisations.

The animation is via Java, but the visualisation belonga IDL.

Andrew
DSTO, Adelaide, OZ

Subject: Re: VTK + IDL

Posted by Karl Schultz on Mon, 30 Aug 2004 16:07:49 GMT
View Forum Message <> Reply to Message

"Hee Chun" <chun.42@osu.edu> wrote in message
news:da6f35c7.0408251747.6fa067d1@posting.google.com...
> I want to have the decent-layered image
> object(ex,tissue,bone,organ)from the volumetric data sets with
> acceptable rendering speed. So I can distinguish/differentiate the
> lesion from the normal tissue with some colors. There is an example
> from the website: http://www.kitware.com
> It is the transparent half-head image on the right bottom. That is not
> the exact image I want, but I need that kind of image.
> Also, it will be fantastic to have a data(volume image intensity)
> picking capability to the layered object with trackball movement. So,
> I can use it for modeling of drug delivery to the lesion in the
> quantitive way by using IDL. I know IDL has the data picking
> capability. How can I do that? Please, let me know that.

Data picking with volumes works by using the PickVoxel method in the
IDLgrVolume object. It will basically tell you the non-transparent voxel
that is closest to the viewer.

You can also mix in isosurfaces with the volume rendering to make the
layered objects. See the ZBUFFER property on IDLgrVolume.

So, yes, you should be able to do something like the kitware picture.

Karl

Subject: Re: VTK + IDL
Posted by chun.42 on Wed, 08 Sep 2004 02:17:46 GMT
View Forum Message <> Reply to Message

"Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message >
> You can also mix in isosurfaces with the volume rendering to make the
> layered objects. See the ZBUFFER property on IDLgrVolume.
>
> So, yes, you should be able to do something like the kitware picture.
>

Thanks Karl for the tips. But I still need to ask about the layered
objects.
I has been trying to generate isosurfaces of skin and bone. It looks
fine as I draw them separately, but the semi-transparent skin doesn't
cover the bone completely as I put them togeter. I know I didn't catch
the proper isovalue(0~232) for bone/skin with 'head' data set. But I
tried the following code with the different data sets where I know the

exact isovalue for skin and bone. The result is the same. Some portion of a bone-face is covered by the skin, and the others(ex: the bone around eyes) are not. Do I miss something? Is there any alternative way?

```
image = [[[red]],[[green]],[[blue]],[[alpha]]]
oImage = OBJ_NEW('IDLgrImage', image,
INTERLEAVE=2,Blend_function=[3,4])

isovalue = 120
Isosurface, volumeData, isovalue, vertex, connect
oPolyBone = Obj_new('IDLgrPolygon',Data=vertex,Polygons = connect,$
 Shading=1,Style=2, Color=[230,230,230],/Reject,/Zero_opacity_skip)

isovalue1 = 40;
Isosurface, volumeData, isovalue1, vertex1, connect1

oPolySkin = Obj_new('IDLgrPolygon',Data=vertex1,Polygons = connect1,$
  Color=[255,255,255],Shading=1,Style=2,texture_map = oImage)
;
;
oModel ->Add, oPolyBone
oModel ->Add, oPolySkin
```

Thanks

HC

Subject: Re: VTK + IDL
Posted by Karl Schultz on Wed, 08 Sep 2004 21:16:27 GMT
View Forum Message <> Reply to Message

"Hee Chun" <chun.42@osu.edu> wrote in message
news:da6f35c7.0409071817.4ffbaa88@posting.google.com...
> "Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message >
>> You can also mix in isosurfaces with the volume rendering to make the
>> layered objects. See the ZBUFFER property on IDLgrVolume.
>>
>> So, yes, you should be able to do something like the kitware picture.
>>
>
> Thanks Karl for the tips. But I still need to ask about the layered
> objects.
> I has been trying to generate isosurfaces of skin and bone. It looks
> fine as I draw them separately, but the semi-transparent skin doesn't

```
>  cover the bone completely as I put them togeter. I know I didn't catch
>  the proper isovalue(0~232) for bone/skin with 'head' data set. But I
>  tried the following code with the different data sets where I know the
>  exact isovalue for skin and bone.  The result is the same. Some
>  portion of a bone-face is covered by the skin, and the others(ex: the
>  bone around eyes) are not.  Do I miss something?  Is there any
>  alternative way?
>
>
>  image = [[[red]],[[green]],[[blue]],[[alpha]]]
>  oImage = OBJ_NEW('IDLgrImage', image,
>  INTERLEAVE=2,Blend_function=[3,4])
>
>  isovalue = 120
>  Isosurface, volumeData, isovalue, vertex, connect
>  oPolyBone = Obj_new('IDLgrPolygon',Data=vertex,Polygons = connect,$
>  Shading=1,Style=2, Color=[230,230,230],/Reject,/Zero_opacity_skip)
>
>  isovalue1 = 40;
>  Isosurface, volumeData, isovalue1, vertex1, connect1
>
>  oPolySkin = Obj_new('IDLgrPolygon',Data=vertex1,Polygons = connect1,$
>  Color=[255,255,255],Shading=1,Style=2,texture_map = oImage)
>  ;
>  ;
>  oModel ->Add, oPolyBone
>  oModel ->Add, oPolySkin
>
```

I can't tell too much just from your description. But I would guess that your skin isosurface is intersecting with your bone isosurface near the eyes and is just disappearing into the bone. Perhaps the data is not good enough to resolve the two surfaces, or perhaps the isovalues are not quite right. If I (almost) poke myself in the eye, I notice that there is not a lot of distance between skin and bone.

As an aside, the ZERO_OPACITY_SKIP property only has effect when there is a texture map, so you don't need it on the bone polygon.

Other than that, you seem on the right track.

Karl