Subject: Re: IDL virtual reality (was 3D Object IDL)
Posted by Dick Jackson on Wed, 08 Aug 2001 21:56:34 GMT
View Forum Message <> Reply to Message

Hi George,

I'm quite sure it can be done in IDL using two different views on the same set of objects, as follows:

- put all objects you want to view in one model, I'll call it the object model
- create a leftView and a rightView, using Perspective projection (Projection=2)
- add a model to each view (each view requires one)
- add the object model to one of these models as usual
  (leftModel -> Add, objModel), and add it to the other as an alias
  (rightModel -> Add, objModel, /Alias), so the same objects are shown in both
  without duplication

I see two ways to get the different views that you need for left and right eye, maybe someone else can make a case for one or the other:

1. Translate leftModel and rightModel horizontally (to right and left respectively, I think)

or

2. Shift the ViewPlane_Rect of leftView to the left and rightView to the right, as if you are defining the shape and position of rectangular lenses on a pair of eyeglasses in front of each eye (with the origin being the point between the centres of the two lenses)

Any zoom/pan/rotate controls should operate on the object model *only*, but they will need to tell both views to redraw when any change happens. If you do all of this, I think you should be in business.

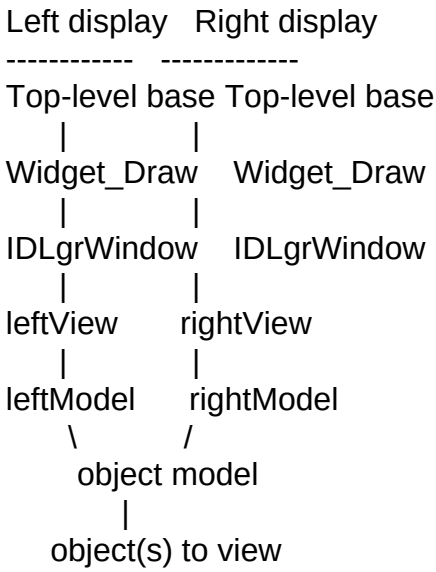
Now to the practical matter of getting the images to the user. If we go with your idea of two polarized projectors, I assume they behave like two monitors (except you can overlap their images!), so you would need two cards, or one card that can support two monitors (the Matrox Millennium G400 DualHead Max is one example).

(Question: can you really get good pixel-for-pixel alignment across the whole display with two projectors? I've never seen it done, but of course this will be important for good results.)

So if this is going to two displays, I think you'll want a full-screen

top-level base on each display, with each base having a Widget_Draw with one of these two views inside it.

To sum it up:



If you haven't built your own 3D viewing application, it may be possible to build this on top of the example code in xobjview.pro, idlexobjviewwid__define.pro and idlexobjview__define.pro, either by modifying the code or by making subclasses of these to add this functionality.

I hope this is of some help, it's an interesting topic!

--
-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

"george Millward" <george@apg.ph.ucl.ac.uk> wrote in message news:d90c0773.0108080052.6baccbe5@posting.google.com...
> My question really
> is: What is the most efficient way to write 2 separate scenes (L and
> R) at the same time. I.E., are there any techniques in IDL that make
> this a "no brainer" - has anyone done this before ? I am really
> looking for technical IDL input. Do you write the data to the
> Z-buffer and alternate between the 2 scenes in real time (as fast as
> the render engine will do it) etc. etc. ?
>
>> "George Millward" <ghm@appleonline.net> wrote in message
>> news:B795A847.14F%ghm@appleonline.net...

>>> Hi there,
>>>
>>> I am wanting to generate full 3D output from IDL object scenes. I can
do
>>> this offline (i.e., create two different views in which the "eye" is
>> offset
>>> for left and right view). But I want to be able to do all this fully
>>> interactively.
>>> So my setup would be:
>>> Computer running IDL - output to (maybe) 2 graphics cards - > 2
projectors
>>> (with polarised filters) and then viewed using 3D (polarised glasses).
>>>
>>> In practice therefore I need a system for outputting 2 slightly
different
>>> images (left and right) of the same scene.
>>> Does anyone know how to do this ? Are 2 graphics cards required or can
it
>> be
>>> done with one ?
>>>
>>> Thanks in advance for any help.
>>>
>>> George Millward.

Subject: Re: IDL virtual reality (was 3D Object IDL)
Posted by Rick Towler on Wed, 08 Aug 2001 23:40:00 GMT
View Forum Message <> Reply to Message

(now that I think I understand what you are doing I posted this back to the group)

I played around with my camera object and *I think* I have it doing what you need to do. I wrote a program that will "flicker" the display where the frames alternate between the left and right "eye". It is based on one of my camera demos and should be fairly bug free. :)

There are plenty of details to attend to. I just made up a distance between the eyes. Also, right now if you project a line from each eye into space the lines are parallel. If you need them to cross at some focal point (which I assume real eyes do) you will need to set the LOOKAT value to some point on the horizon (for example lookat=[0,0,-20]) and also set the TRACK keyword so the camera "stares" at that point. I have made some notes at the appropriate place in the demo on this.

The demo program is here:

http://www.acoustics.washington.edu/rht/programs/camdemo_fli ck.pro

Note that the program displays the frames rendered per second but the upper bound is set by IDL's timer event which seems to be able to issue maximum 100 events per second (at least on my machine). The best visual quality can be had if you enable Vsync (which will limit your FPS to your monitors vert. refresh rate).

You will need my camera object and my quaternion_object both are available here:
http://www.acoustics.washington.edu/rht/3d_animation.html

As for projecting this from two projectors, could you write the program so it can run in master or slave mode and use the SOCKET procedure to communicate between to machines (a left machine and a right machine). The master would dictate when the slave renders a new scene. Otherwise you'll have to do something at the hardware level (some sort of switch) and I think synchronizing the rendering with your switch would be difficult.

Enjoy!

-Rick Towler

----- Original Message -----
From: "George Millward" <george@apg.ph.ucl.ac.uk>
To: "Rick Towler" <rtowler@u.washington.edu>
Sent: Wednesday, August 08, 2001 2:33 PM
Subject: Re: IDL virtual reality (was 3D Object IDL)

>
> Hi Rick,
>
> That is exactly what i am doing. If you have a 3D scene created then the
> right and left views are just a matter of displacing the "eye" by a
> suitable amount along the X-axis (or moving the outermost containing
> object in the opposite direction).
> At present I do this in a really clunky, unelegant manner. I am writing
> left and right views to different draw windows and I can tell when the
> offsets are about right by forcing the two images to align and refocussing
> (i.e., a practiced "bozeyed" technique - though a bit painful, this
> technique is really useful in 3D work because you can tell when something
> is working without doing anything other than producing he 2 views on the
> screen).

> If you have a more elegant object for doing the transformation then this
> would be great. The next step is to understand how to integrate this
> with the technology requirements of the 3D system (i.e., getting the 2
> views to integrate into OpenGL or something - then output to the "flicker
> glasses" system). What I want to do is get this to drive 2 independent
> projectors (with polarizing filters etc.) - so we can build a cheap
"virtual
> reality suite for looking at atmospheric data. So in this case we would
> want the two rendered scenes to be "writing" independently to the 2
outputs
> (either by having 2 graphics cards, or by having some sort of interlacing
> system, essentially the same as the "flicker glasses" system).
> So the first step is to get IDL to output 2 stereo scenes as efficiently
> as possible. the next is then to get this routed correctly with the
> hardware..
> Anything you have would be really appreciated.
> Cheers for now,
> George.
>
> On Wed, 8 Aug 2001, Rick Towler wrote:
>
>> I think I might have just the solution but I am not confident that I
know
>> how you render "one for left, one for right". Do you render a scene for
the
>> left eye, then translate the objects in your view so they are shifted
and
>> render the right eye?
>>
>> I have written a camera object that gives intuitive control over object
>> graphics view composition. If rendering your R and L views are a matter
of
>> simply moving the camera an inch or so back and forth along the "eye
space"
>> x axis then it would work for you.
>>
>> -Rick

"george Millward" <george@apg.ph.ucl.ac.uk> wrote in message
news:d90c0773.0108080052.6baccbe5@posting.google.com...
> Hi There,
> Thanks for your input. Actually I am well aquainted with the "flicker
> glasses" technique. I have done a fair amount of 3d animation and
> used both the "flicker glasses", and twin polarised projectors, as the
> virtual reality technique. However, up till now my method has been to
> render 2 separate animations in IDL (one for left, one for right) and

> then recombine these in special 3D video software to create the final
> virtual reality scene. This works fine - but is completely "offline".
> So, my question here is how to do this in fully interactive mode. I
> am not thinking here about the actual 3D technique ("flicker glasses
> etc.") as much as how to achieve this within IDL. My question really
> is: What is the most efficient way to write 2 separate scenes (L and
> R) at the same time. I.E., are there any techniques in IDL that make
> this a "no brainer" - has anyone done this before ? I am really
> looking for technical IDL input. Do you write the data to the
> Z-buffer and alternate between the 2 scenes in real time (as fast as
> the render engine will do it) etc. etc. ?
>
> Any help much appreciated.
> Cheers ,
>
> George Millward
>
> "Rick Towler" <rtowler@u.washington.edu> wrote in message
news:<9kp8m8$16n2$1@nntp6.u.washington.edu>...
>> You might want to look into 3d shutter glasses. They work by
alternately
>> rendering a left and right eye view to the screen while simultaneously
>> covering the opposite eye by darkening the lens. All of this is done at
the
>> driver level so you don't have to render your two views by hand.
>>
>> I would start with a consumer level product. Both Elsa (glasses are
called
>> Revelator) and Asus (glasses are called VR Spectacle) make glasses that
work
>> with some of their graphics cards for the PC. The glasses tend to be a
>> little flimsy and aren't the most comfortable but they are cheap and a
good
>> place to start with this sort of thing. They do work quite well but I
know
>> some people that get headaches or sick to their stomachs using them.
>>
>> If you decide that shutter glasses are the way to go, you can start
looking
>> for a more professional product. I know they exist but I just haven't
>> gotten that far.
>>
>> good luck!
>>
>> -Rick Towler
>>
>>
>> "George Millward" <ghm@appleonline.net> wrote in message

>> news:B795A847.14F%ghm@appleonline.net...
>>> Hi there,
>>>
>>> I am wanting to generate full 3D output from IDL object scenes. I can
do
>>> this offline (i.e., create two different views in which the "eye" is
>> offset
>>> for left and right view). But I want to be able to do all this fully
>>> interactively.
>>> So my setup would be:
>>> Computer running IDL - output to (maybe) 2 graphics cards - > 2
projectors
>>> (with polarised filters) and then viewed using 3D (polarised glasses).
>>>
>>> In practice therefore I need a system for outputting 2 slightly
different
>>> images (left and right) of the same scene.
>>> Does anyone know how to do this ? Are 2 graphics cards required or can
it
>> be
>>> done with one ?
>>>
>>> Thanks in advance for any help.
>>>
>>> George Millward.
>>>

Subject: Re: IDL virtual reality (was 3D Object IDL)
Posted by Pavel A. Romashkin on Wed, 08 Aug 2001 23:47:39 GMT
View Forum Message <> Reply to Message

Rick Towler wrote:
>
> If you need them to cross at some focal point
> (which I assume real eyes do) you will need to set the LOOKAT value to some
> point on the horizon (for example lookat=[0,0,-20])

So, not just my eyes after a day at the computer are crossed behind the
back of my head - [0,0,*-20*]?
This makes me feel better :-)
Cheers,
Pavel

Subject: Re: IDL virtual reality (was 3D Object IDL)
Posted by Rick Towler on Thu, 09 Aug 2001 01:04:27 GMT

"Pavel A. Romashkin" <pavel.romashkin@noaa.gov> wrote in message news:3B71CF9C.BE64EAFD@noaa.gov...
> Rick Towler wrote:
>>
>> If you need them to cross at some focal point
>> (which I assume real eyes do) you will need to set the LOOKAT value to some
>> point on the horizon (for example lookat=[0,0,-20])
>
> So, not just my eyes after a day at the computer are crossed behind the
> back of my head - [0,0,*-20*]?
> This makes me feel better :-)

Ha hahaha. That's why my head hurts at the end of the day |-)

IDL OG uses a right handed coordinate system where +Z is coming out of the screen. By default a IDLgrView places the viewer at some +Z value looking back torwards the axis on a vector [0,0,-1]. My camera does the same. So the camera would be at some point say [0,0,10] looking along a vector [0,0,-1] past the orgin to [0,0,-20] off in the distance. You probably know this.

So you must be giving me a hard time about how I was unconventional with "eye space" coordinates and "world space" coordinates. Typically viewing systems like my camera use LHCS's (or so the literature says) but I have never used any other "camera" viewing systems and the way my camera developed just seemed natural for IDL users.

-Rick

> Cheers,
> Pavel