

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.

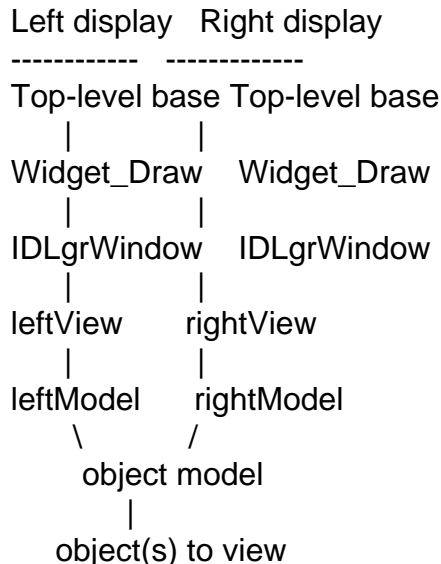
Now to the practical matter of getting the images to the user. If we go with your idea of two polarized projectors, I assume they behave like two monitors (except you can overlap their images!), so you would need two cards, or one card that can support two monitors (the Matrox Millennium G400 DualHead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