Full window stereo

R. Rodriguez,* G. Chinea,* N. Lopez,* and G. Vriend†

**Centre for Genetic Engineering and Biotechnology, Havana, Cuba*
†BIOcomputing, EMBL, Meyerhofstrasse 1, 69117 Heidelberg, Germany

Visualisation is the bioinformaticist's most important tool for the study of macromolecules, and being able to see molecules in stereo is a crucial aspect. Stereo vision is based on the principle that each eye is presented with the best possible image of what it would have seen if the object was really there in 3D. The simplest approach to stereo vision is to display the right eye picture on the right half of the screen and the left eye picture on the left half while using a mirror system to ensure that each eye sees what it is supposed to see. More expensive workstations use hardware to alternately display the left and right eye pictures while synchronously blocking the transparency in the right or left lens of the special glasses worn by the user. We present here some simple software that uses inexpensive hardware, originally designed for the computer game industry, to make full screen stereo available on Linux-based PCs. The quality of the stereo vision is similar to the top-of-the-line graphics workstations that are capable of quad-buffering. This stereo option has been incorporated in the X11 based version of WHAT IF (Vriend, G. J. Mol. Graphics 1990, **8,** *52–56), but the stereo source code is freely available and can easily be incorporated in other visualization packages. © 2000 by Elsevier Science Inc.*

INTRODUCTION

The human brain can study the invisible and untouchable by making mental, physical, or pictorial models. This is true for almost every field of science, and a series of Nobel prizes has been awarded for studies that converted a complicated, seemingly intangible object or idea into a visible model. Hundreds of software packages for the *in silico* study of macromolecules are available, and each of these packages has its own special features aimed at reducing nature's complexity by mapping it on one visible model or another. Some of these visualization modes seem to be well suited for the human brain and have become extremely successful. Examples are the chicken wire representation of electron density, the electrostatic surfaces in GRASP,[1] and the ribbon and spiral diagrams introduced by Richardson.[2]

The most successful visualization trick is undoubtedly stereo viewing. Not only does stereo vision give us an extra dimension so that more information can be represented (and understood!) in one picture, but a stereo model of a molecule is also one dimension closer to reality than a 2D picture.

The principle of stereo viewing is relatively simple (see Figure 1). The left and right eye each see an object from positions about 6 cm apart, and the brain combines these two pictures to create a stereo impression. *In silico* stereo vision is based on presenting each eye with a slightly different picture, and each of these two pictures should represent as precisely as possible what the eye would see if the 3D model were really there. Much has been written about how to project a 3D object as two mono pictures that can be viewed with special lenses or mirror boxes (see Figure 2) in order to create a stereo impression. At first one would think that two projections, one before and one after a slight rotation of the object around a vertical axis, is all that is needed, but it can easily be shown that this is not optimal.[3–5] Further discussion of this topic is, however, beyond the scope of this article.

STEREO DEVICES

Figure 2 shows a simple mirror box that can be used to fool the human visual system. Although it has been shown that this system needs some minor corrections,[6] it works well for most practical applications. Most molecular graphics software packages provide a display mode, called side-by-side stereo, in which the picture for the right eye is displayed on the right half of the screen and the picture for the left eye on the left half. Side-by-side stereo can be viewed either by using a device as shown in Figure 2, or without any special device at all. This is done by totally relaxing the eyes and letting each eye look only at "its own" picture. Alternatively, one can look at the screen cross-eyed, with each eye looking only at the picture projected in front of the other eye. In this latter case, the left and right picture need to be swapped over on the screen. Side-by-side stereo was until now called the poor man's solution to stereo viewing because a device as shown in Figure 2 costs less than $100.

The more expensive solution for stereo viewing is to use a system where the whole screen alternately shows the picture for the left and the right eye, while spectacles are worn that

Color Plate for this article is on page 333.

Corresponding author: G. Vriend.

E-mail address: vriend@embl-heidelberg.de (G. Vriend)

Journal of Molecular Graphics and Modelling 17, 310–314, 1999

655 Avenue of the Americas, New York, NY 10010
1093-3263/99/$–see front matter
PII S1093-3263(99)00038-8

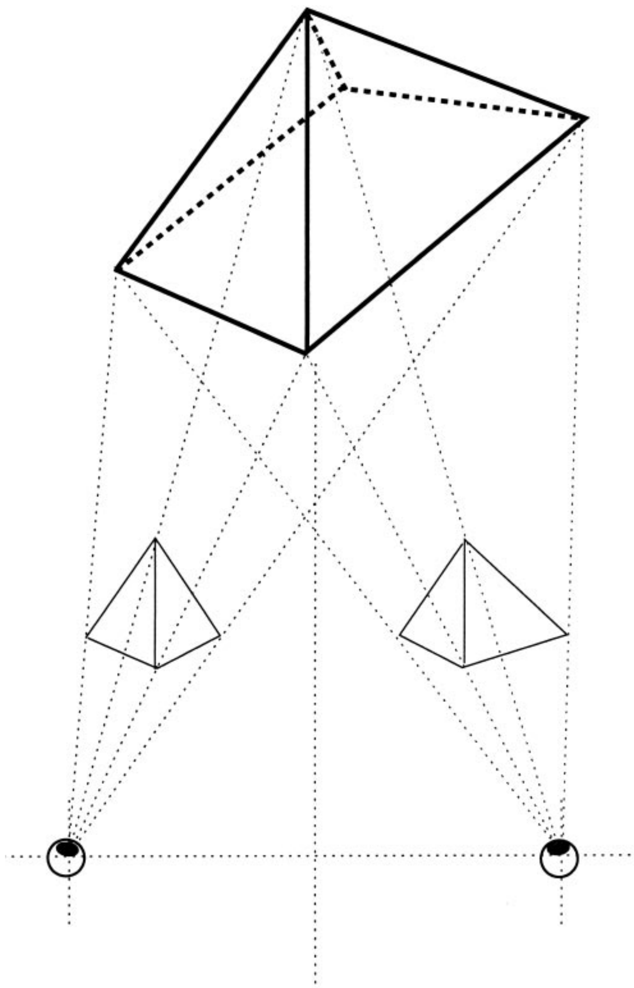


Figure 1. Principle of stereo vision. The two eyes, represented by circles, see the boldface pyramid from a different location. The two small pyramids represent what the two eyes see. For stereo vision, the graphics system should present each of the two small pyramids to the corresponding eye.

alternately make the right or left lens nontransparent. If the screen refresh rate and the transparent/nontransparent swapping of the lenses are properly synchronized, the user will see stereo.

The most common spectacles use liquid crystals (LC) that can switch from transparent to nontransparent at a high frequency. These glasses "shut off" visibility for the one eye while the screen is displaying the other eye's image and vice versa, hence the name LC shutter glasses.

Because the images for each of the two eyes must be separated in time, a perfect synchronization of the glasses to the graphics hardware is needed. This is handled in hardware by most modern graphic cards but the appropriate software drivers to use it, for example in Linux, are still missing.

X11 FULL WINDOW STEREO

To be able to use inexpensive stereo glasses in the X11 version of WHAT IF, it was necessary to write software that would display the picture for one of the eyes in the odd lines of the screen, but at the same time also display the picture for the other eye in the even lines. We tested three different hardware approaches for synchronizing the shutter glasses with the two pictures, and implemented all three methods in WHAT IF. This stereo source code is, however, totally universal and we make it publicly available.

The first and simplest of the methods is used in less expensive devices such as Tetratel's Eye-FX or VR Standard VR-Joy, and is done entirely by the device that scans the video signal, alternatingly switching off the odd and the even lines. This is called "line blanking" or "synthetic interlace" and its main advantage is that it works with every video card-monitor configuration without any need to modify the hardware setup.

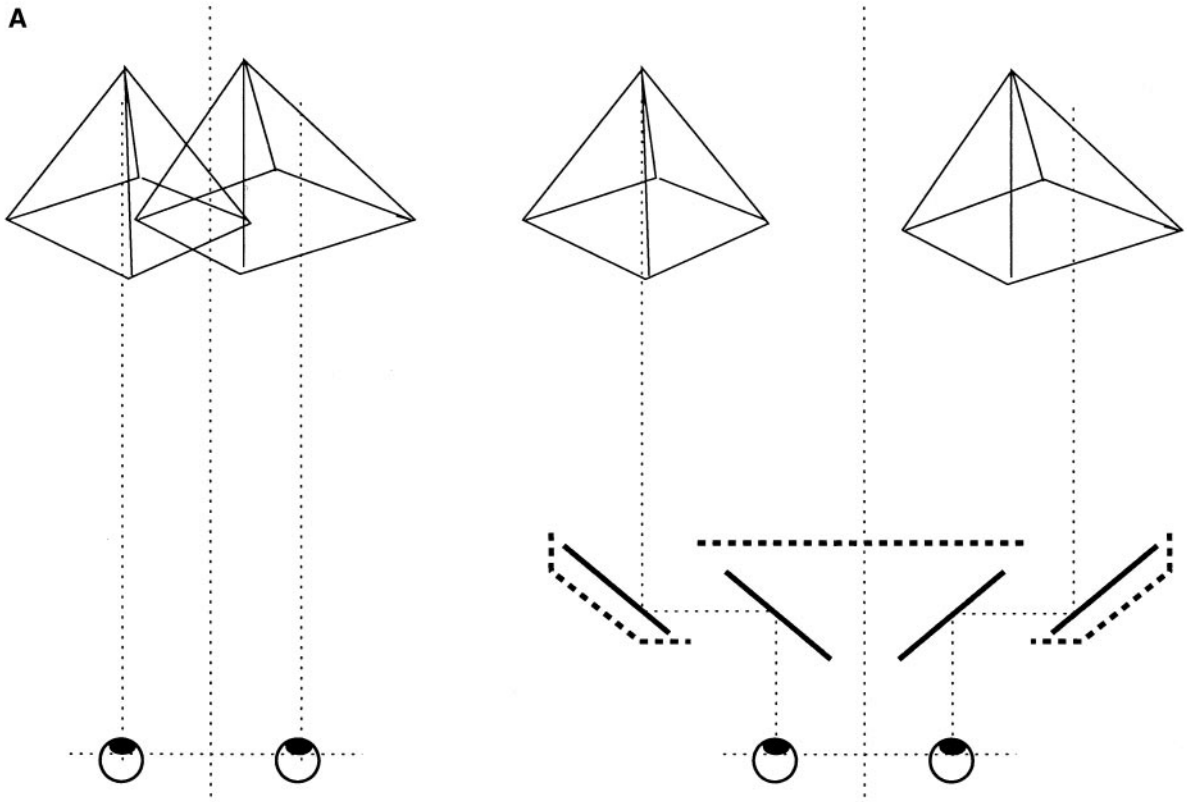
The other two methods require the use of interlace modes that can easily be programmed for most current operating systems by making simple changes to thesetup. In the case of XFree86, the X11 server most often used on Linux-based PCs, some changes should be made to the configuration file (XF86Config). In the interlace mode, the cathode ray tube (CRT) controller of the graphics card first draws the odd lines (i.e., 1, 3, 5 . . . 1023) and then returns to the top and draws the even lines (i.e., 2, 4, 6 . . . 1024). Each of these "draws" is called a field and the entire display is drawn in two separate fields (odd and even).

Using two different vertical sync signals the hardware can recognize which is the even and which the odd field. Systems such as the NuVision 60GX NSR, Tetratel EyeFX, or VR Joy can synchronize the shuttering of the lenses in the stereo glasses with these alternating fields.

StereoGraphics Corporation uses another trick to synchronize their CrystalEyes glasses. In the PC version (EPC-2), they use a procedure called White Line (WL) code. This involves using the last line of the display to trigger the transparency of the lenses. StereoGraphics Corporation uses the convention that a line composed of about four-fifths pure white and the rest of pure black pixels indicates the right eye picture, while a line composed of about one-fifth pure white and the rest black pixels indicates the left eye picture. To use the EPC-2, our software draws the white lines in the last two lines of the root window. The major drawback of this method is that the last two lines of the display have to stay clear of any overlap with other windows or desktop gadgets.

Silicon Graphics has a large line of graphics workstations (SGI stations). Their moderately expensive workstations combine special hardware with a variety of visualization techniques to create good stereo perception. By basically dividing the complete screen into top and bottom halves and doubling the monitor refresh rate, SGI was able to drive the LC glasses in such a way as to let the left eye see only the top half of the framebuffer, and the right only the bottom half. This same procedure can also be implemented on PCs by using an external device that doubles the refresh rate. The StereoGraphics and NuVision controllers provide such a tool and the technique is known as above/below stereo or sync-doubling. The major disadvantage of this technique is that the stereo pictures always cover the full screen. If one wants to see stereo in just one window while maintaining normal mono for the rest of the screen (stereo-in-a-window, which we call "full-window-stereo"), an SGI station that can do quad-buffering is needed. For the WHAT IF implementation of stereo vision on the PC, we initially started evaluating the old full-screen (above/below

A



B

Figure 2. Principle of stereo vision from 2D images. (a) If the two pyramids from Figure 1 are placed in front of the corresponding eyes, the pictures overlap, which makes stereo vision difficult and messy. (b) A stereo device as shown in the inset allows the two pyramids to be drawn separately. The mirror system ensures that each picture is cleanly presented to the corresponding eye.

or sync doubling) method. However, this method was discarded later in favor of the full-window interlaced method, because above/below showed no other advantage than simplicity of implementation.[7]

SOFTWARE IMPLEMENTATION

To generate the stereo pictures we have used the parallel camera model[4, 5, 8] rather than rotations around any axis. The stereo perception is further improved by Z-buffering and color depth cueing.

Figure 3 lists the most important parts of the X11 code used in WHAT IF to create full-window-stereo. The WHAT IF graphics code was modified to introduce the stereo multibuffering scheme, the parallel camera model for calculating the viewing matrices, and the code that checks for eye reversal (see below).

The multibuffering scheme consists of one buffer per eye: a combination buffer and a screen buffer. When a change is needed on the screen, the pictures for the two eyes are drawn in the two separate eye-buffers, then copied one after the other (the first in full, the second using a clipmask) to the combination buffer. The graphic context (GC) is updated using the XSetClipMask operation (Figure 3c).[9] The clipmask (Figure 3a) is a pixmap of depth one, with the same root as the GC. For our application, its geometry should be the same as that used in the drawable pixmap. The clipmask ensures that the X11CopyArea function (Figure 3c) will copy only the non-masked lines.

```
/* Create the two pixmaps that will be used for the GC clipmask  */
  {
    GC MaskGC;
    unsigned long MaskGC_valuemask = 0;
    XGCValues     MaskGC_values;
    int   Yi;

    OddMasked_pixmap=XCreatePixmap  (display, rootwin,
                                              DisplayWidth (display, screen),
                                              DisplayHeight(display, screen),1);
    EvenMasked_pixmap=XCreatePixmap (display, rootwin,
                                              DisplayWidth (display, screen),
                                              DisplayHeight(display, screen),1);
    /* I need a GC for 1 bit visual */
    MaskGC=XCreateGC(display, OddMasked_pixmap , MaskGC_valuemask, &MaskGC_values);

    /* Fill the pixmaps with lines */
    for (Yi=0;Yi<=winextent.y;Yi=Yi+2)
      {
        XSetForeground(display, MaskGC, 0);
        XDrawLine(display,EvenMasked_pixmap , MaskGC, 0, Yi,   winextent.x, Yi);
        XDrawLine(display,OddMasked_pixmap ,  MaskGC, 0, Yi+1, winextent.x, Yi+1);
        XSetForeground(display, MaskGC, 1);
        XDrawLine(display,EvenMasked_pixmap , MaskGC, 0, Yi+1, winextent.x, Yi+1);
        XDrawLine(display,OddMasked_pixmap ,  MaskGC, 0, Yi,   winextent.x, Yi);
      }
    /* We don't need this GC any longer */
    XFreeGC(display,MaskGC);
  }
```

a

```
/* Select now the active Stereo mask to be even or odd
   according to the new position of the window  */

    if ((event->xconfigure.y%2)==0)
        StereoMask_pixmap=EvenMasked_pixmap;
    else
        StereoMask_pixmap=OddMasked_pixmap;
```

b

```
/* Update the stereo screen buffer with the stereo mask
   after finishing the update in the right eye plane */

   if (stereodevice>2)
    if (device->matstack.eye==-1)
     {
       GC *cgc = &gc;

       XCopyArea (display, LeftEye_pixmap,  Stereo_pixmap, *cgc,
                           0, 0, winextent.x, winextent.y, 0, 0);
       XSetClipMask (display, gc, StereoMask_pixmap);
       XCopyArea (display, RightEye_pixmap, Stereo_pixmap, *cgc,
                           0, 0, winextent.x, winextent.y, 0, 0);
       XSetClipMask (display, gc,  None);
     }
```

c

```
/* Dump the stereo buffer to the real drawing window */

   if (stereodevice>2)
    if (stereomode==1)
     {
       GC *cgc = &gc;

       XSetClipMask (display, gc, None);
       XCopyArea (display, Stereo_pixmap, win,  *cgc,
                           0, 0, winextent.x, winextent.y, 0, 0);
       XSync(display,False);
     }
```

d

Figure 3. The key fragments of the source code used for stereo vision in a window.

Table 1. Stereo devices and their characteristics

Company/device name[a]	Price	Principle[b]	Quality	Remarks
Tetratel/EyeFX	NA	LB, INT	++	c
VRStandard/VR Joy	\$160.00	LB, INT	+	d
StereoGraphics/EPC-2 + CrystalEyes	\$1190.00	WLC	+++	e
NuVision 60GX NSR	\$399.00	INT	+++	f

[a]WWW addresses of vendors: Tetratel, http://www.tetratel.com/; VR Standard, http://www.vrstandard.com/; StereoGraphics, http://www.stereographics.com/; NuVision, http://www.nuvision3d.com/.
[b]LB, line blanking; INT, Interlace; WLC, White line code (this table shows only those modes used by WHAT IF).
[c]This system is discontinued, at present it is being replaced by the EyeFX Pro, an infrared wireless system.
[d]Wired system, two glasses per controller, small LCs.
[e]Wireless glasses, large LCs, WLC makes software control possible.
[f]Wireless glasses, large LCs, very good controller with eye reversal.

When an eye reversal situation occurs, for example by moving the full-window-stereo window by an odd number of pixels in a vertical direction, the clipmask for the right eye is simply inverted (Figure 3b) to avoid depth perception being flipped back to front.

After merging the two images the combination buffer is copied into the screen buffer a bit slower than the nonstereo or double-buffer modes, but given the speed of current CPUs, graphics hardware, and CRTs, this is certainly not a problem.

HARDWARE IMPLEMENTATION

There are no special hardware limitations, other than that this type of stereo vision will work only if you have a real CRT monitor and not an LCD display. It is advisable to use a monitor that supports a high refresh rate. In stereo the eye refresh rate will be half of the monitor refresh rate. If the eye refresh rate gets too low the picture will start to flicker. Although this is a matter of taste (and money), we suggest using CRTs with a monitor refresh rate of at least 120 Hz.

We looked at a series of stereo devices ranging in price from \$200 up to \$1200 (see Table 1). All gadgets are easy to install and require no major modifications to the computer setup. Table 1 lists these devices and some of their characteristics, as well as some of our comments on them.

DISCUSSION

We have created a procedure that uses the X11 Window system library to create stereo vision in a single window aided by LC shutter glasses. We have tried to keep this procedure simple and hardware independent.

Since they were first introduced to PCs by IBM, interlaced modes have acquired a bad reputation. We have observed a trend in the computer community to believe that one should avoid the use of interlace modes at all costs because they introduce flicker. Today's display technology, however, is much better than a decade ago, easily allowing for the flicker-free use of interlace modes.

The main advantage of the interlace stereo modes compared with the above/below method is that multiple stereo windows can be displayed at the same time. This is not very important for the computer game industry, but it is certainly crucial to science (see Figure 4). A detailed comparison of these two methods is beyond the scope of this article and is described elsewhere.[7]

The hardware discussed in this article represents the state of the art for stereo devices in summer 1999. By the time you read this article, some of the companies may perhaps no longer exist and others might have designed different or better devices. We tested only a series of radically different hardware stereo devices to ensure that our software can cope with all possible stereo vision techniques.

ACKNOWLEDGMENTS

We are grateful to Christoph Brungert for his WWW page,[10] which actually made us think about the possibility of doing this work. We thank Mike Ingram of Tetratel for his support and for sending one EyeFX unit for free. Our thanks also to Vince Power of NuVision for his technical support, and to the people of StereoGraphics for making extensive stereo literature freely available via their WWW pages.

REFERENCES

1 Nicholls, A., Sharp, K.A., Honig, B. Protein folding and association: Insight from the interfacial and thermodynamic properties of hydrocarbons. *Proteins* 1991, **11,** 281–296
2 Richardson, J.S. Schematics drawings of protein structures. *Methods Enzymol.* 1985, **115,** 359–380
3 Hodges, L.F., and McAllister, D.F. Rotation algorithm artefacts in stereoscopic images. *Optical Eng.* 1990, **29,** 973–976
4 CrystalEyes Software development kit. StereoGraphics Corp., 1998
5 Akka, R. Converting existing applications to support high quality stereoscopy. StereoGraphics Corp., 1998
6 Thomas, D.J. Towards more reliable printed stereo. *J. Mol. Graphics* 1993, **11,** 15–22
7 Rodriguez, R. and Vriend, G. http://swift.embl-heidelberg.de/whatif/chap59.html
8 Woo, M., Neider, J., and Davis, T. *OpenGL Programming Guide.* Addison Wesley, 1997
9 Scheifler, R., and Gettys, J. *X Window System: Core Library and Standards, X version 11, releases 6 and 6.1.* Digital Press, 1996
10 Brungert, C. http://www.stereo3d.com