

SPATIAL VIEW OF 3D OBJECTS USING STEREOGRAMS

V. PREJMEREAN, S. MOTOGNA, AND V. CIOBAN

ABSTRACT. We present a method to view in space a three-dimensional solid body using polygonal representation. This method has been chosen because it allows a visualization of a body without other instruments. We prefer the screen visualization instead of paper visualization, because we can also produce an animation, for example by rotating the body (around the axis, using the keyboard), increasing the image reality.

1. INTRODUCTION

There are several methods to visualize a body in space, in the way we see it in reality. Stereograms allow such a view without other special instruments, just a certain skill to view these objects.

A similar method, that can be easily applied, needs for visualization a pair of glasses with different colored lenses. The spatial view is based on the fact that each eye sees the same body, but from a different angle, rotated. It's obvious that we have to draw two images simultaneously (one for each eye) and then to distinguish them (each eye should see its corresponding image). This technique can be found in some spatial geometry books, that also contain the glasses, and the effect is quite spectacular. The stereograms have a totally different functioning principle (we shall see this difference in their construction), an object in space, as in reality, having the third dimension (the depth). Unfortunately that bodies are not moving! Working on the screen, this thing will become possible, and even more, the user can obtain the desired move.

Either on paper or on screen the three-dimensional objects are drawn after they have been transformed through a projection in a real (bidimensional) plane, then a new transformation is applied through which the plane image is framed in a window of the paper or of the screen. The screen image is no longer seen in space (it is a plane image), we just imagine how the initial three-dimensional object looks [5]. Oftenly the same plane image can lead to false interpretations (optical

Received by the editors: October 22, 2003.

2000 *Mathematics Subject Classification*. 68U05, 65D20.

1998 *CR Categories and Descriptors*. I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling – *Curve, surface, solid, and object representations*; I.3.7. [**Computer Graphics**]: Three-Dimensional Graphics and Realism – *Animation*.

illusions) since the reverse transformation is not unique. In the following we will refer to the screen representations, because these objects can be rotated around the coordinate axis using the keyboard, increasing the reality of the image. We will argue why we have chosen the perspective projection instead of the parallel one, besides the fact that is the real one. If we do not consider the animation, the images (stereograms) can be printed and viewed in space, but statically.

2. THE PRINCIPLE OF STEREOGRAMS

Visualizing a stereogram doesn't need any extra instrument, only a certain skill. Practically, you need to look behind the screen (at a certain distance), each eye seeing two pyramids, as in figure 1. When two of the four pyramids will overlap (only three will be seen), the one from the middle will be seen in space.

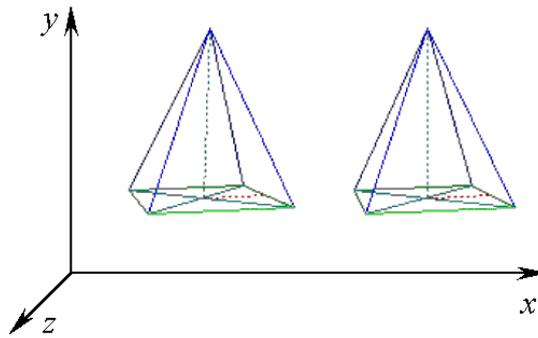


FIGURE 1. An example of a stereogram

The two drawn objects will not be identical! If this would be the case, then the middle object will be better seen, but not in space. This image will be plane too, as a picture, in which we just imagine its depth, but we don't see it in space, as in reality. In order to have a spatial view, the two images must be different, similar to reality when each eye sees the same image but from a different angle, allowing to sense the third dimension, the depth. If the second object (the right-hand side pyramid) would have been obtained rotating the first object around the Oy axis with some angle (assuming that the observation point is on Oz coordinate, that expresses the depth), then the resulting object is not a stereogram. If we would be able to see each object with the corresponding eye, then we will see the spatial representation of the object. An easy method to achieve this purpose uses glasses with different colour lenses, allowing to filter the image. The two objects will have different colours corresponding to the lenses' colours. This method has at least two disadvantages: we need the special glasses, and we have only some types of

viewable images (depending on colours). Even for this method, a certain skill in visualizing the object is needed, different from person to person.

Our method overcomes the two disadvantages, being able to use all combinations of colours in the image, just having some training to visualize the stereograms.

The principle of stereogram visualization is: the shorter the distance p between two pixels is, the closer the point seen in space (in depth) is to the screen (distance d), as shown in Figure 2. If we want this point to be farther to the observer (and implicitly to the screen), then we must increase the distance between the two points [1,2,3]. From Figure 2, we can easily deduce that:

$$d(p) = e \times p / (g - p)$$

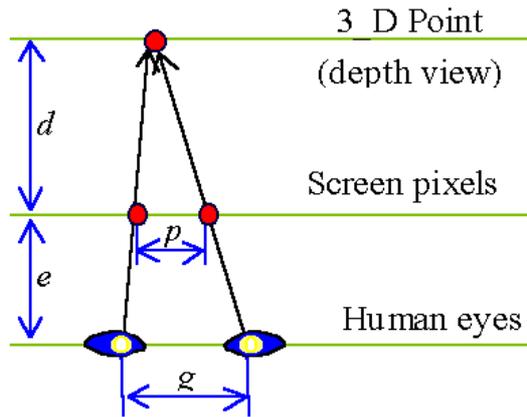


FIGURE 2. Visualizing a stereogram

From a practical point of view in constructing a stereogram, we are more interested in the inverse function to determine where to put a pair of pixels on the screen: which distance p do we need to obtain a desired depth d . The inverse equation is:

$$p(d) = g \times d / (d + e)$$

If we consider the identical pair figure, at a distance p_0 , then the composed figure will be in a plane situated at the distance $d_0 = d(p_0) = e \times p_0 / (g - p_0)$ from the screen. But what we want is to have some points nearer, and some farther. If we want to have a point in a farther plane, at the distance $d = d_0 + \Delta d$, we must know how much we should increase the distance p_0 . So, we have to compute Δp

$$\begin{aligned}\Delta p &= p(d) - p(d_0) = p(d_0 + \Delta d) - p_0 = g \times (d_0 + \Delta d)/(d_0 + \Delta d + e) - p_0 = \\ &= g \times (e \times p_0/(g - p_0) + \Delta d)/(e \times p_0/(g - p_0) + \Delta d + e) - p_0\end{aligned}$$

and obtain it as an expression of pixels:

$$\begin{aligned}\Delta p_p &= [\Delta p \times Nph/Dsh], \text{ where} \\ Nph &= \text{the number of horizontal pixels} \\ Dsh &= \text{horizontal size of the screen}\end{aligned}$$

The above equations also depend on the distances e and g :

- the distance e is in the range 50 to 70 cm ($\approx 20''$ - $25''$);
- the distance g representing the stereographic ability of a pupil gauge, with values around 6.5cm ($2.56''$).

It is necessary to note that the distance between two pixels from a pair is strictly positive, and obviously could not exceed the pupil gauge: ($0 < p = p_0 + \Delta p < g$). For example, if we want a distance of approximately 5cm ($\approx 2''$), considering the distance between two pixels of 0.042cm ($0.0166''$), then the initial value of p_0 should be 120 pixels. If the stereographic ability of the user is good, then this distance can be increased, without exceeding g , and if this stereographic quality is low, then the decrease of the initial distance p_0 is recommended, but not too much in order to avoid the overlapping of the two images (in which case the stereogram visualization is difficult).

3. THE CONSTRUCTION OF A STEREOGRAM

Firstly, we will choose the screen windows according to the conditions mentioned in the previous section. In the first window we will represent the first object in a perspective projection, and in the second window the same object will be projected modifying the x coordinates corresponding to the depth view, as described above.

In figure 3, we considered an example in which we have increased the distance AA' and decreased the distance BB' , yielding to a move of the point A in a farther plane and of point B in a nearer plane ($AA' > VV' > BB'$). The resulting effect is a triangle visible in space and no longer parallel with the screen.

One may notice that by decreasing the base AB we obtain a rotation of the triangle in one direction, and by increasing the base we obtain the rotation in the opposite direction, as shown in Figure 4.

We also mention that as much as we further the extremity of a segment from the screen, the longer the segment becomes (in projection), as in the case of the segment VB from the left image in Figure 4. This seems like an anomaly, because in reality the farther the object is, the smaller it looks. In order to overcome this effect we recommend the perspective projection by which the farther sides will be

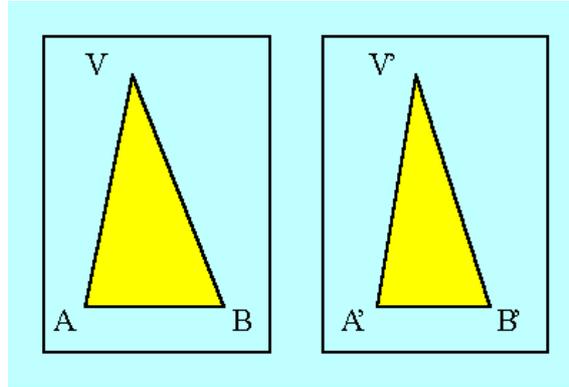


FIGURE 3. Construction of a surface

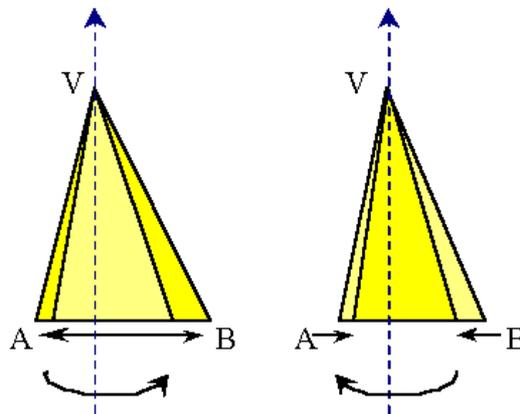


FIGURE 4. Rotation of a surface

drawn smaller, and when increasing their lengths they do not exceed the lengths of the nearer segments. Therefore, it is obvious why the perspective projection is preferred for spatial view.

Practically, for the first figure from the stereogram we follow the standard steps of representing a polyhedron:

- (1) establish the screen window $V_1(u_1, v_1, u_2, v_2)$; for example $Viewport(200, 100, 300, 200)$;
- (2) determine the real window (of the projected figure), namely the minimal rectangle containing all the projections of the points;

- (3) represent each segment (side) with its correponding characteristics: line type, colour,...

The second figure is drawn in a similar way, with the following differences:

- (1) establish the screen window $V_2(u_1 + p_0, v_1, u_2 + p_0, v_2)$, such that:
 - $u_2 < u_1 + p_0$ to avoid windows intersection (and consequently figure overlaping);
 - $p_0 < g$ such that the distances between two pair-pixels is less than the pupil gauge;
for example: $ViewPort(340, 100, 440, 200)$, so $p_0 = 140$;
- (2) the real window computed before is preserved, since the body and the projections characteristics remain unchanged;
- (3) represent each side of the body, performing the necessary corrections for the x coordinate of each pixel based on the z coordinate value of the corresponding point (assuming that the observer is on the positive semi-axis Oz). These corections can be made either increasing the horizontal distances between the pixels representing the vertexes without exceeding the distance g , either decreasing these distances, taking care not to overlap the left figure, that will affect the "readability" of the stereogram.

Let $P(x, y, z) \in \mathfrak{R}^3$ be the point we want to represent, and its projection $P'(x', y') \in \mathfrak{R}^2$ will be determined in the following way:

$$\begin{aligned} x' &= x \times (\delta - q) / (\delta - z) \\ y' &= y \times (\delta - q) / (\delta - z) \end{aligned}$$

where δ represents the distance of the observer from the origin $Obs(0, 0, \delta)$, on the positive semi-axis Oz , and q represents the distance of the projection plane from the origin, that is parallel with the plane xOy . These conditions skip any other geometrical ransformations [4,5], and allow future rotations of the body, such that it can be put into a certain desired position. We have also taken into consideration that the observer should be placed at a considerable distance from the figure (greater than the biggest side of the figure) such that the body will not be represented too irregular, but also not too big, when the projection will seem like a parallel one.

The pixel $P''(u, v) \in V_2$ from the declared screen window obtained through window transformation is given by:

$$\begin{aligned} u &= (x' - x_1) / (x_2 - x_1) \times (u_2 - u_1) + u_1 + p_0 \\ v &= (y' - y_1) / (y_2 - y_1) \times (v_2 - v_1) + v_1 \end{aligned}$$

for a screen window $V_2(u_1 + p_0, v_1, u_2 + p_0, v_2)$ and a real window $W(x_1, y_1, x_2, y_2)$, already determined for the first figure.

In order to have a spatial view we must apply a depth corection, yielding to the point $P_S(u - \Delta p_P, v)$.

Although the computing equations have been presented in section 2 (including for Δp_P) and can be used, we consider that is much simpler to make a direct corection, based on the coordinate z of the point P using the following formula:

$$\Delta p_P = (z - \min\{z_i\})/\Delta z \times k$$

where:

- Δz is the height:
 $(\Delta z = \max\{z_i | P_i(x_i, y_i, z_i) \in V\} - \min\{z_i | P_i(x_i, y_i, z_i) \in V\})^1$
- k is the depth amplification constant, for example 1.5. This constant can be determined either by objective factors (the ratio $\Delta x/\Delta z$), either subjective factors: the distances g and e from Figure 2.

This correction is applied only in one direction because $\Delta p_P \in [0, k]$, so it allows just to see some points nearer to the screen. In order to view them farther, we will modify the equation in the following way, taking care not to exceed the distance e :

$$\Delta p_P = ((z - \min\{z_i\})/\Delta z \times 2 - 1) \times k, \text{ so } \Delta p_P \in [-k, k]$$

If the observation will be performed in front of the screen (as we will explain later), then the windows can be choosen as the two halves of screen (left and right), and k can be greater, for example 50. If this constant is too big, for example 100, then certain points are formed to closer to the observer and the figure is more difficult to be seen. This kind of visualization also require changing the sign of the constant, and therefor of the correction.

It is possible to apply a corection such that the distance between the objects could be modified, in order to give a kind of freedom to the application regarding the observation ability of users. The final horizontal coordinate of a pixel will be:

$$u' = u - \Delta p_P - Np_0$$

where Np_0 represents the closing distance, expressed in pixels.

The depth correction can be applied even before the window transformations, in the projection plane, in the following way:

$$x' = x - \Delta p_R, \text{ with } \Delta p_R = u^{-1}(((z - \min\{z_i\})/\Delta z \times 2 - 1) \times k)$$

where u^{-1} is the inverse transformation, from screen window $V_2(u_1 + p_0, v_1, u_2 + p_0, v_2)$ into the real window $W(x_1, y_1, x_2, y_2)$:

$$x' = (u - u_1 - p_0)/(u_2 - u_1) \times (x_2 - x_1) + u_1.$$

¹ V represents the set of body vertexes

Of course, in this situation, the depth correction will not be applied in the screen window, so:

$$u' = u - Npo.$$

The effect is more signifying if we draw three pyramids: the initial one, one on the left and one on the right at different distances. Seen with one eye, there will be six images: two of them will overlap, and will be seen in space (in the middle), and two will be simple, in plane, visible on sides. As a consequence, there will be four pyramids, the middle ones in space and not at the same distance, the left one will rotate in the reverse direction and its projection is incorrect drawn (the farther side is seen longer, therefor incorrect). In order to have a valid representation we must modify the equation for the left pyramid with $+$ yielding the point $(P_S(u + \Delta p_P, v))$ obtaining the same direction (see Figure 4) and a correct projection.

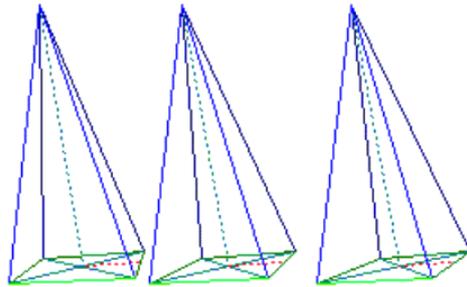


FIGURE 5. Bodies at different distances

An interesting remark is that if we apply different values of Npo , then the two pyramids (which now will be rotated simultaneously, and in the same direction, and will be correctly visualized through a perspective projection) will no longer be in the same plane (one will be closer to the observer than the other one), and even more will have different dimensions (surprisingly, the farther pyramid will look bigger, contrary to what be would probably expect), as shown in Figure 5.

An even more interesting possibility is to visualize a stereogram in front of the screen (not in the back, as in our previous examples), as shown in figure 6. This case offers the great advantage of visualizing a stereogram represented on a big surface (for example, projected on a wall), where it is obvious that the distance p is greater than the pupil gauge g .

If we want to visualize the figure in front of the screen, then the projection corrections will be applied inversely to the back screen representation, namely positively for the right figure: $P_S(u + \Delta p_P, v)$ and negatively for the left figure:

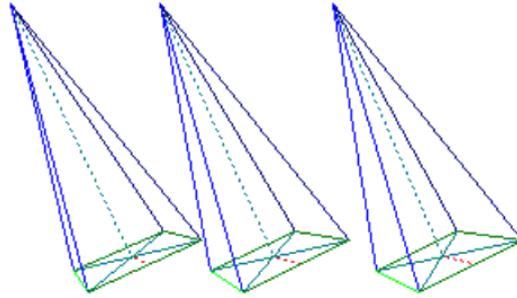


FIGURE 6. Pyramides in front of the screen

$P_S(u - \Delta p_P, v)$). The effect is somehow stronger, since the objects are clearer and nearer.

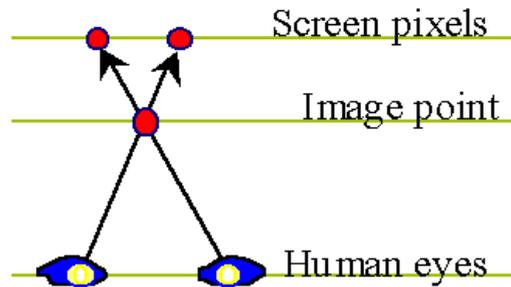


FIGURE 7. Visualization in front of the screen

4. PERFORMING ANIMATION

It is well known that animation helps understanding a spatial image, since the dynamic projection is an important factor in increasing its reality. The fact that the image is rotating, for example the pyramid is rotated around its height, helps us imagining this pyramid in space, even without representing it as a 3-dimensional object, since a nearer point is rotated in one direction and a farther one is rotated in the opposite direction. This effect is increased even more in case of a stereogram, yielding to a better visualization of the image.

In certain cases it is more convenient to allow the user the possibility to manipulate the represented object through rotations, using the keyboard:

- the keys **O** and **P** to rotate the object around the Oy axis with some angle α (not too big), in one direction, respectively in the other direction;
- the keys **Q** and **A** to rotate the object around the Ox axis with some angle β in one direction, respectively in the other direction;
- the keys **S** and **D** to rotate the object around the Oz axis with some angle γ in one direction, respectively in the other direction;

In order to simulate the increase of the rotation speed, that can be performed around the weight centers, the angles α, β and γ can be automatically increased after each rotation. In this way, holding one key, after we visualized the body in space, this will be rotated correspondingly to the pressed keys faster and faster. The coordinates of the point G , around which we execute the rotation, can be determined as the arithmetic mean of the body vertexes. The rotation will be executed around some lines parallel with the axis, that cross through the point G . This rotation can also help us to visualize the stereogram.

In order to have a continuous move of the body we use two active and two visual pages for alternative draw and visualization of the images.

As a final remark, we note that in observing a stereogram at the beginning everything seems “foggy” and then it becomes more clearer, as some real objects we could touch. This observation is easier for a trained viewer, so “Good luck!”.

5. REMARKS

The article contains figures in black and white, losing some image quality, and of course its animation. That’s why we have put the examples from this article on the web page of the publishing journal: <http://www.cs.ubbcluj.ro/~studia-i>.

REFERENCES

- [1] A.A. Kisman, *Random Dot Stereograms*, Kinsman Physics P.O. Box 22682, Rochester, N.Y., 1992.
- [2] V. Prejmerean, V. Cioban, *3D Images Simulation through Stereograms*, Research Seminars, Seminar on Computer Science, Preprint 2/1996, pp. 75-78.
- [3] V. Prejmerean, *Modelling Stereograms by Matrix Grammars*, Research Seminars, Seminar on Computer Science, Preprint 2/1997, pp. 173-178.
- [4] V. Prejmerean, *Computer Graphics and Image Processing*(in, Universitatea de Nord Baia-Mare, Facultatea de Științe, 2000.
- [5] A. Watt, *3D Computer Graphics*, University of Sheffield, Addison Wesley, England, 1993.

BABEȘ-BOLYAI UNIVERSITY, DEPARTMENT OF COMPUTER SCIENCE, RO-400084 CLUJ-NAPOCA, ROMANIA

E-mail address: `per@cs.ubbcluj.ro`

E-mail address: `motogna@cs.ubbcluj.ro`