

Research Article

Stereo 3D Mouse Cursor: A Method for Interaction with 3D Objects in a Stereoscopic Virtual 3D Space

Hossein Azari, Irene Cheng, and Anup Basu

Department of Computing Science, 2-21 Athabasca Hall, University of Alberta, Edmonton, Alberta, Canada T6G 2E8

Correspondence should be addressed to Anup Basu, anup@cs.ualberta.ca

Received 30 April 2009; Accepted 11 September 2009

Academic Editor: Pietro Zanuttigh

Copyright © 2010 Hossein Azari et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We introduce a different approach of applying stereoscopy principles to implement a virtual 3D pointing technique called stereo 3D mouse cursor (S3D-Cursor) based on two or more views of an ordinary mouse cursor. The basics of such an idea have already been applied as a by-product of some stereo-based visualization applications with usually less attention to its strengths or weaknesses as a generic alternative of its 2D counterpart in stereoscopic 3D space. Here, we examine if such an idea satisfies all or the main expected requirements of an abstract 3D cursor. Moreover, we analyze its accuracy and evaluate the applicability of this approach in terms of different efficiency factors. For this purpose, we have adapted a real-time point-based rendering software called QSplat to a multiview rendering version named QSplatMV. We have implemented the S3D-Cursor on top of this new application and developed a simple editing toolset for manipulating the virtual 3D objects. Our user evaluation results suggest the effectiveness of the approach in terms of detection accuracy and user satisfaction compared to using an ordinary mouse cursor on a conventional 2D screen.

1. Introduction

The stereopsis capability of the human vision is one of the primary means used by our vision system to give us a 3D perception of the world [1]. This ability allows us to have a sense of the 3rd dimension from the two slightly different images of the world projected onto our eyes retina. Benefiting from this intrinsic ability, a wide variety of different stereoscopic devices and techniques are employed to create the illusion of depth from the visual stereo contents. Recent advances in these technologies enable the users to watch contents in 3D without using any filtering glasses (multiview autostereoscopic 3D displays [2]) or watch 3D contents in full-colour on conventional displays (colour code glasses [3]). Along with these advances in stereoscopic display technologies it is also necessary to evolve the current 2D/3D interaction techniques and devices for interplay with these new attractive virtual 3D environments. Among these, pointing to the targets (objects or other GUI components) using some devices such as mouse is probably the most appealing interaction method especially useful to work within the graphics environments [4]. For this reason, providing the possibility of pointing to any arbitrary voxel

of the 3D space is one of the first expectations of any user who wants to migrate from 2D to 3D space applications. In this regard, several 3D pointing techniques (or in broader category 3D selection techniques) have been introduced during the years [4]. On the other hand, several commercial or experimental 3D input devices such as 3D space navigator [5], OptiBurst [6], and 3D air mouse [7] and many others are introduced to facilitate working with 3D objects in 2D/3D space by simplifying the fundamental tasks such as zooming, rotating, panning, and so forth.

Stereo-based 3D cursor is among the 3D pointing techniques, which usually has been implemented as a by-product of the stereoscopic 3D visualization (and/or 3D object manipulation) systems or applications, while less attention has been paid to its capabilities as a generic replacement of 2D mouse cursor in stereoscopic 3D space. The stereo 3D cursor (hereafter S3D-Cursor) can be achieved through providing two or more different views of the 2D mouse cursor at a specific disparity. The cursor depths can be controlled by adjusting the amount of the disparity. This enables the user to point to any arbitrary 3D location inside the virtual 3D space projected by a stereoscopic display.

In this paper, we study different aspects of this technique and show how this method satisfies the main functionality requirements of an abstract 3D cursor. Furthermore, in addition to discussing some issues such as accuracy and occlusion handling, we will show how the scene stereo content may be used to virtually enhance the performance of the technique according to 3D Fitts' law [4]. To realize the applicability of these ideas we have improved the QSplat [8], a single-view point-rendering software, to QSplatMV which enables the fast multi-view rendering of a 3D object described by a collection of 3D points or a triangular 3D mesh. We have implemented the S3D-Cursor on the top of this new application and developed a simple 3D editing toolset to manipulate the virtual 3D objects. The application is used to evaluate the capability of the 3D versus 2D cursor in manipulating 3D visualized data. Our user evaluation results suggest the effectuality of using the technique in term of several factors including detection accuracy, simplicity of usage, and overall user satisfaction compared to using an ordinary mouse on a conventional 2D screen. Regarding these advantages, the idea potentially can be used as a generic method in many applications including 3D games, 3D medical data manipulation, and 3D GUIs.

The remaining sections of this paper are organized as follow. In Section 2 we briefly discuss the relevant devices and techniques. In Section 3 we explain the fundamentals of stereo imaging and describe the mathematical model behind the S3D-Cursor. Section 4 is dedicated to the design and implementation aspects of S3D-Cursor. In Section 5 we focus on S3D-Cursor as a generic pointing technique in stereoscopic 3D space. In Section 6 we discuss the issues related to the accuracy of the stereo mouse and possible methods of improving its accuracy. Section 7 discusses the application aspects of our method and presents some of our application toolset outputs and user evaluation results. Finally, Section 8 is devoted to concluding remarks and possible future extensions.

2. Background and Related Works

Several efforts have been done during the years to simplify interaction with 3D application environments. On the one hand, benefiting from different mechanical, electromagnetic, optical, acoustic, and inertial sensors and technologies [9], several kind of 3D input devices are introduced to facilitate working with 3D models and applications. Among the recent commercial ones we may refer to some 3D mice such as space navigator and its counterparts which benefit from a pressure sensing technology on a controller cap to provide simultaneous panning, zooming, and rotation of the 3D object. In fact, the movements such as push, pull, twist, and tilt applied to the cap are translated to the appropriate movement of the 3D object [10]. The 3D air-mouse is another type of 3D computer mouse which uses the ultrasonic technology. A small ultrasonic transmitter is part of the mouse equipments which is worn on the index finger like a ring, and an array of receivers is used to track the movements of the transmitter (finger). These

movements are translated to appropriate actions on the 3D object. For example, zooming is achieved by moving the hand closer to or farther from the screen [7]. IR tracking is also used in some products like OptiBurst to map the natural hand movement composed of translations and rotations (6DOF) to the appropriate actions inside the 3D application [6]. Some other ideas such as GlobeFish, GlobePointer, SquireBone and others are introduced in [11] which use a combination of sensor technologies to provide 6 or more DOF interaction devices.

On the other hand, several task-specific or general-purpose interaction techniques are developed [4] which in combination with the 3D input devices and technologies provide an easier, more natural way of interfacing with the 3D environments. These interactions may be classified into three different task domains: object selection and manipulation, viewpoint manipulation (navigation and travel), and application (system) control [11, 12]. Pointing to the targets (objects or other application components) may be considered as one of the most widely used interaction techniques in GUIs which is pre-requirement of performing many tasks in each of above-mentioned task domains. The pointing is usually achieved by manipulation of a 2D/3D cursor position and/or orientation using an appropriate input device. A *chronological* survey of several 3D pointing techniques and 3D cursor concepts are presented in [4]. This includes *Skitters and Jacks* (1987), *Ray casting* (1994), *Spotlight* (1994), *Virtual hand* (1995) and many other older or recent techniques. According to the review, all these methods are mainly based on virtual hand, ray, or spotlight pointing techniques. Later in the paper, a more formal definition for a 3D cursor is presented. The definition assumes 6 or more (at least 3 translational and 3 rotational) DOF for the pointing device and assumes that the selectable target has to be visible to the user and implies that a typical 3D cursor has to satisfy all the following requirements and constraints.

- (a) Visual presentation—the 3D cursor has a graphical presentation which makes its position and orientation observable to the user.
- (b) Behaviour—the 3D cursor movements are controllable by the user using an appropriate input device.
- (c) Constraints—the 3D cursor reaches all the positions on the (3D) graphical display and is able to touch only one target at a moment.

Then, according to this definition, two main types of 3D cursors are proposed for 3D UIs: the 3D point cursor and the 3D line cursor. The author describes how these two main types satisfy all above-mentioned requirements and how all aforementioned 3D pointing techniques can be derived from these two main types. In fact, the other 3D pointing techniques may be considered as the result of exploiting some possible virtual enhancements to improve the performance of the 3D pointing (or the 3D target acquisition time). Here, a virtual enhancement means changing the values of one or more parameters effective in the 3D target acquisition time according to the 3D Fitts' law. The 3D Fitts' law states that the

target acquisition time or the average *movement time* (MT) to select a target depends on the *distance to move* (A), the *target size* (W : width, H : height, and D : depth of the target), and also the *viewing angle* (θ) at which the target is seen by the user through the following equation:

$$MT \approx 56$$

$$+ 508 \log_2 \left(\sqrt{f_W(\theta) \left(\frac{A}{W} \right)^2 + \frac{1}{92} \left(\frac{A}{H} \right)^2 + f_D(\theta) \left(\frac{A}{D} \right)^2} + 1 \right), \quad (1)$$

where $f_W(0^\circ) = 0.211$, $f_W(90^\circ) = 0.717$, $f_W(45^\circ) = 0.242$, $f_D(0^\circ) = 0.194$, $f_D(90^\circ) = 0.312$, and $f_D(45^\circ) = 0.147$ [4]. In this regard, reducing the movement distance A , increasing size of the target (or equivalently size of the cursor), or a combination of these changes are examples of such virtual enhancements.

Even though, the review in [4] does not refer to the stereo 3D cursor as a 3D pointing method, the technique is known for years and already applied in some stereoscopic 3D visualization and manipulation applications and GUIs. OrthoEngine 3D Stereo offers a 3D stereo cursor among its advanced tools for 3D viewing and manipulation of aerial pictures or satellite imagery data [13]. BioMedCache as an application for molecular design also offers stereoscopic 3D visualization and 3D stereo cursor [14]. In [15], authors present their success in modifying the functionality of X Window System with the purpose of constructing generic tools for displaying 3D-stereoscopic content. In this context, they refer to the implementation of the 3D pointer via creating a shadow pointer which follows the motion of the real pointer in both fields of the stereo window. In their implementation the depth of the 3D cursor is controlled by automatic adjustment of the disparity of the shadow pointer with respect to the real one. Moreover, recently the prototype of a simultaneous 2D/3D GUI for (auto)stereoscopic displays is introduced that refers to the implementation of a 3D stereo cursor. The stereo cursor is enabled upon entering of the mouse pointer into the 3D GUI area [16]. Here again the disparity is automatically adjusted to keep the virtual 3D cursor in touch with the surface of the 3D objects and 3D GUI components.

In spite of all these efforts, there are less focus on the studying the capabilities of the stereo cursor itself as a generic extension of 2D cursor to the stereoscopic 3D space. While such a study is necessary considering increasing popularity of some emerging technologies such as autostereoscopic displays and other stereo-based display techniques. In this paper, we will study different theoretical and implementation aspects of stereo cursor. Our implementation, which simply named as S3D-Cursor, composed of two or more views of the same 2D mouse cursor presented on the (auto)stereoscopic display screen at a specific disparity which essentially follows the same principles applied by others to form the 3D cursor. However, our implementation supports two disparity adjustment modes (manual and automatic) which in fact enables us to show that such a simple technique not only satisfies all the above-mentioned requirements of an

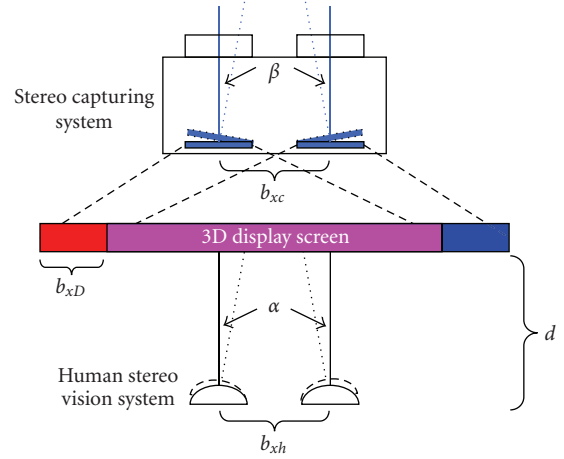


FIGURE 1: Process of capturing, display, and watching stereo images.

abstract 3D cursor but also it is possible to improve the performance, flexibility, and simplicity of object selection and pointing tasks by applying several virtual enhancements. Furthermore, we discuss some practical issues such as S3D-Cursor accuracy and ambiguities caused by occlusions on autostereoscopic displays. Our implementation does not imply any assumption about the input pointing device and fully complies with the current functionality of the 2D mice. However, special input devices such as available 3D mice may facilitate performing 3D tasks. Moreover, the disparity calculations in most cases happen concurrent to the 3D model rendering process, therefore, it does not impose extensive additional computational load to the application. This issue totally will be resolved by providing a system level support for stereo cursor.

3. S3D-Cursor Mathematical Model

Before going into the details of the stereo 3D mouse design and implementation we briefly describe the stereoscopy process through establishing a supporting mathematical model. This mathematical model equally can be applied to the stereo 3D cursor formation process as well. We will use the following notation and definitions in this and the subsequent sections (see Figure 1):

(X, Y, Z) : a 3D point, and $(\hat{X}, \hat{Y}, \hat{Z})$: the corresponding estimated 3D point

$(x_{l/r}, y_{l/r})$: projection of a 3D point on the left/right image plane

$(\hat{x}_{l/r}, \hat{y}_{l/r})$: estimation of the projection of a 3D point on the left/right image plane, considering the nearest pixel

$e_x(e_y)$: distance between two neighboring pixels in the x (y) direction

b_x : baseline of the stereo setup or horizontal displacement between the left and right images on a display screen

$\alpha(\beta)$: eyes (cameras) vergence angle

d : viewing distance

f : focal length

R : total resolution, that is, the total number of pixels over the unit square.

The subscripts c , D , and h are also used to refer to a stereo imaging system (cameras), 3D display, and human eyes features, respectively. For example f_c stands for the focal length of cameras while f_h means the eyes' focal length.

Figure 1 illustrates the process of capturing, displaying and watching stereo images. Two different stereo systems are involved in this process: stereo capturing and human stereo vision. As showed in this figure, either of these two systems can have its own configuration independent of the other one. Restricting ourselves to the two basic stereo camera configurations, that is, parallel and with vergence ones, four different scenarios may happen in this process. In the simplest scenario we may assume parallel geometry for both capturing and viewing sides. Then, assuming a pinhole camera model [17], the projection of a 3D point (X, Y, Z) on left and right camera image planes are given by:

$$\begin{aligned} x_{rc} &= \frac{f_c X}{Z}, & x_{lc} &= \frac{f_c (X - b_{xc})}{Z}, \\ y_c &= y_{lc} = y_{rc} = \frac{f_c Y}{Z}. \end{aligned} \quad (2)$$

The images captured by the stereo cameras are scaled by a factor S and presented on the display. Thus, the corresponding 2D point coordinates on the display screen can be computed as:

$$x_{rD} = Sx_{rc}, \quad x_{lD} = Sx_{lc}, \quad y_D = Sy_c \quad (3)$$

Finally, the 3D point projections on the eyes through a display medium placed at distance d are obtained as

$$\begin{aligned} x_{rh} &= \frac{f_h x_{rD}}{(f_h + d)}, \\ x_{lh} &= \frac{f_h (x_{lD} + b_{xD} - b_{xh})}{(f_h + d)}, \\ y_h &= \frac{f_h y_D}{(f_h + d)} \end{aligned} \quad (4)$$

From the formulae (4) the 3D point reconstructed by human eyes is theoretically given by

$$\begin{aligned} Z_h &= \frac{f_h b_{xh}}{x_{rh} - x_{lh}} = \frac{(f_h + d) b_{xh}}{(x_{rD} - x_{lD} - b_{xD} + b_{xh})}, \\ X_h &= Z_h \frac{x_{rh}}{f_h} = Z_h \frac{x_{rD}}{(f_h + d)}, \\ Y_h &= Z_h \frac{y_h}{f_h} = Z_h \frac{y_D}{(f_h + d)}. \end{aligned} \quad (5)$$

In a more realistic scenario we may assume that there is a small vergence angle α acting on the human eyes when

they are watching a stereo pair through a stereoscopic device. From the formulations established in [18] for the vergence-stereo configuration, the 3D point reconstructed by the eyes in this case is given by

$$Z_h = \frac{b_{xh} AB}{CD + EF}, \quad X_h = Z_h \frac{D}{B}, \quad Y_h = Z_h \frac{y_{rh}}{F}, \quad (6)$$

where

$$\begin{aligned} A &= (f_h \cos \alpha + x_{lh} \sin \alpha), & B &= (f_h \cos \alpha - x_{rh} \sin \alpha) \\ C &= (f_h \cos \alpha + x_{lh} \sin \alpha), & D &= (f_h \sin \alpha + x_{rh} \cos \alpha) \\ E &= (f_h \sin \alpha - x_{lh} \cos \alpha), & F &= (f_h \cos \alpha - x_{rh} \sin \alpha) \end{aligned} \quad (7)$$

and x_{rh} , x_{lh} , and y_{rh} are given by:

$$\begin{aligned} x_{rh} &= \frac{f_h (x_{rD} \cos \alpha - (f_h + d) \sin \alpha)}{(f_h + d) \cos \alpha + x_{rD} \sin \alpha} \\ x_{lh} &= \frac{f_h ((x_{lD} + b_{xD} - b_{xh}) \cos \alpha + (f_h + d) \sin \alpha)}{(f_h + d) \cos \alpha - (x_{lD} + b_{xD} - b_{xh}) \sin \alpha} \\ y_{rh} &= \frac{f_h y_D}{x_{rD} \sin \alpha + (f_h + d) \cos \alpha}. \end{aligned} \quad (8)$$

If we apply the formulae in (8) into (6) to obtain the 3D point estimation in terms of display coordinates, after some simplification, the formulae in (5) are obtained again (details are skipped here). This means that if the stereo images presented on the display screen are captured under parallel configuration, then the 3D-scene reconstructed by human eyes, theoretically does not depend on the amount of vergence of the eyes.

For the other two possible scenarios the stereo images are captured under vergence. Therefore, assuming a vergence angle β and again using formulations given in [18] the 3D point projections on the camera image planes are obtained as

$$\begin{aligned} x_{rc} &= f_c \frac{X \cos \beta - Z \sin \beta}{Z \cos \beta + X \sin \beta} \\ y_{rc} &= f_c \frac{Y}{X \sin \beta + Z \cos \beta} \\ x_{lc} &= f_c \frac{(X - b_{xc}) \cos \beta + Z \sin \beta}{Z \cos \beta - (X - b_{xc}) \sin \beta} \\ y_{lc} &= \frac{f_c Y}{Z \cos \alpha - (X - b_{xc}) \sin \beta} \end{aligned} \quad (9)$$

Thus, the coordinates of these projections after representing on the display by a scale factor S are

$$\begin{aligned} x_{rD} &= Sx_{rc}, & y_{rD} &= Sy_{rc} \\ x_{lD} &= Sx_{lc}, & y_{lD} &= Sy_{lc} \end{aligned} \quad (10)$$

Contrary to the first and the second scenarios, here the corresponding projections on the display screen do not locate

on the same raster line. However, as we discussed in [19] assuming the eyes are able to establish conformance between the corresponding projections in the left and right views, then Z_h and X_h can be calculated using either Equations (5) (assuming parallel geometry for human eyes) or Equations (6) (assuming vergence geometry for human eyes). If we assume that eyes compensate for vertical differences of the corresponding points (see our justifying experiments in [19]) then we can say that the 3D point location estimation by human eyes mainly depends on the horizontal disparity of the corresponding projections in stereo pairs. Regarding this simplification assumption, we can apply Equations (5) as a good (approximate) model for 3D point estimation (3D cursor position estimation) by human eyes via stereo images (2D cursor images) presented by a stereoscopic device in all above-mentioned possible scenarios.

4. S3D-Cursor Implementation

For the best realization of the S3D-Cursor, it was necessary to implement an underlying (multiview) stereo rendering application. For this purpose, we have extended the QSplat to support rendering multiple views of a 3D object. QSplat is a real-time point-based rendering program which uses a progressive hierarchical method especially useful to render large geometric models composed of huge number of 3D point descriptions [8]. The program uses OpenGL as its graphic library to implement different types of rendered-point primitives called splot. Our new extension which is called QSplatMV enables the user to decide on the number of cameras (views), the distance between the cameras (b_{xc}), and the amount of horizontal displacement of the views on the display screen (b_{xD}). The current version assumes parallel configuration and the same baseline for all cameras. This allows a simplified implementation of the cameras' rotation and translation. These movements are applied to a central virtual camera and then the position and direction of all cameras are set with respect to this virtual camera. The cursors' disparities calculations are also simplified under parallel geometry. The system also supports a special red/blue rendering mode which gives the flexibility of using the application on all conventional displays just by wearing simple anaglyph glasses.

4.1. Binocular Implementation. The stereo mouse cursor is implemented on the top of the QSplatMV. Two different disparity adjustment modes are considered for the cursors: manual and automatic. The automatic mode is more useful in targeting the existing objects in the applications such as 3D games and 3D object manipulation tools where the scene 3D information is already available. In the automatic mode one view, say the left one, is considered as the reference view. When the left mouse cursor points at a pixel in the left view, having the depth information of the pixel (usually available in the depth buffer) and the camera parameters, the corresponding pixel on the right view (or the position of the right cursor) can be determined using the basic stereo imaging formulations as follow.

Considering the OpenGL default perspective projection, which implies a normalization transformation as well [20], the relationship between the point depth in the virtual camera coordinate Z_c and the depth maintained in the depth buffer Z_w can be stated as

$$Z_c = \frac{f_c d_{\text{far}}}{Z_w (d_{\text{far}} - f_c) - f_c}, \quad (11)$$

where d_{far} is the far clipping plane distance and f_c is the near clipping distance or camera focal length. On the other hand, from (2) Z_c can be determined as

$$Z_c = \frac{f_c b_{xc}}{x_{rc} - x_{lc}} = \frac{f_c b_{xc}}{\text{disp}} \quad (12)$$

From (11) and (12) the amount of the disparity is obtained as

$$\text{disp} = \frac{(Z_w (d_{\text{far}} - f_c) - d_{\text{far}}) b_{xc}}{d_{\text{far}}}. \quad (13)$$

The disparity calculated in (13) should be properly scaled and adjusted considering the viewport transformation settings and the amount of the views displacement b_{xD} .

As already mentioned, the automatic disparity adjustments causes the illusion of touching the surface of the real 3D object so that when the user slides the mouse pointer over the display screen the virtual 3D cursor follows the holes or other irregularities of the 3D object surface. Figure 2 shows a red-blue stereo pair of Lucy model with three samples of stereo mouse cursor at different disparities which actually ensemble three cursors at three different distances from viewer (watch this figure using red-blue anaglyph glasses to see the formation of these 3D cursors at different distances).

Implementing the automatic disparity adjustment over the image/video stereo pairs implies that an efficient stereo matching algorithm be incorporated into the system to find the corresponding projections in the left and right views. Contrary to the classic stereo matching algorithms, which involves establishing correspondence for all pixels, here the correspondence need to be find only for the pixel located

under the current position of the left (or alternatively right) mouse cursor. This assumption may lead to more efficient algorithms for real-time applications.

In the manual mode the user is able to change the depth of the 3D cursor by manually adjusting the disparity between the left and right cursor. The manual mode is useful when the user wants to point somewhere other than the visible surface of the 3D object or to adjust the disparity estimated by a stereo matching algorithm. The latter case especially can be used as a cost-effective, accurate method for extracting ground-truth data from stereo image pairs.

4.2. Multiview Implementation. The multiview implementation is particularly useful for autostereoscopic multiview displays. The implementation is essentially similar to two-view case. Assuming all cameras are parallel and located on the same baseline at equal distances then all corresponding projections of a 3D point on the display screen will be located at the same raster line with the same disparity. As result, again one of the views can be considered as reference for disparity calculations and the other corresponding projection can be determined with respect to the reference view.

Although, this implantation is pretty simple, it may cause some problems in occluded areas. This is because that in the autostereoscopic 3D displays the viewer is only able to see two consecutive views of the scene at a time. As result, as depicted in Figure 3, the implementation works fine as while as the corresponding projections, similar to the instances inside the circles, are visible in all views. The implementation becomes problematic when corresponding projections fall into occluded areas in two or more consecutive views. For example observe the third and the forth view of the cursors denoted by squares. When the viewer moves his/her head to watch the third and the forth views, the 3D position of the 3D cursor reconstructed from these two cursor views, is incorrectly estimated. The problem could be fixed if each view serves as the reference for the next adjacent view. However, this may lead to the ambiguity in converting the cursor positions to a unique 3D position. A more advanced algorithm may detect the occluded areas and hide or highlight the cursor in corresponding views. Upon receiving such a hint, the user may change the viewpoint or the cursor position to access a specific point from any desired view.

5. S3D-Cursor as a Generic 3D-Pointing Technique

In general, S3D-Cursor can be classified as a 3D point cursor which satisfies all assumptions and requirements of an abstract 3D cursor as follows.

- (i) It has obviously three translational DOF, and three rotational DOF can be achieved by viewpoint manipulation.
- (ii) Although the occlusion problem on autostereoscopic displays may necessitate a further move to grab the target, all visible parts of the scene (all visible targets) are selectable by the user.
- (iii) The cursor has a visual presentation which in addition to its position may also make its orientation observable to the user.
- (iv) The S3D-Cursor movements are simply controllable by the user using a conventional mouse; however, more appropriate input devices such as 3D mice may be adapted for efficiency purposes.
- (v) The S3D-Cursor is able to reach to all positions in the comfortable viewing range of the stereoscopic graphical display by appropriate (manual) adjustment of the disparity. This is especially important when the user aims to point to an empty space for example for the purpose of creating a new object.
- (vi) Finally, since in a stereoscopic 3D space the user only sees the surface of the opaque objects, so he/she will be able to touch only one target at a moment in automatic mode (a priority mechanism may be applied for transparent or translucent objects). In manual mode appropriate visual hints may be implemented to aware the user of moving the cursor to the physically invisible areas.

Regarding these properties, the S3D-Cursor can be applied as a general alternative of 2D cursor in stereo space. Some virtual enhancements may be implemented to improve the basic functionality of S3D-Cursor. In fact, auto disparity adjustments may already be considered as such an enhancement which virtually reduces the distance to the target (reduces A in (1)) by “*removing the empty space between the cursor and the targets*”—the enhancement that according to [4] not already tried in other 3D pointing techniques. “*Increasing the target size*”—that is, increasing W , H , or D in (1)—is another enhancement which especially useful in handling the accuracy deficiencies of the stereo cursor (see Section 5). If the application control components are also implemented in 3D, then several enhancements may be applied on the GUI components especially on menus and the application window itself. These include appearance of the pop-up menus on the same depth of the 3D cursor, and dynamically managing the position and size of the windows depending on the scene composition and user actions. These types of enhancements also may be considered as virtual reduction of distance to the target.

6. S3D-Cursor Accuracy Analysis

Considering the Equations (5) in Section 3, the whole stereoscopy system depicted in Figure 1 can be considered as a (parallel) stereo camera system whose focal length is equal to the human focal length plus the viewing distance ($f_h + d$), the distance between its cameras or its baseline length is equal to the distance of the human eyes (b_{xh}), and the display screen displaced to the left/right has the role of its left/right image plane. Assuming f_h , d , and b_{xh} are constant values, the accuracy of the reconstructed 3D points or the stereoscopic resolution mainly depends on the size (width and height) of the pixels of the display. In fact, as understood from Equations (5), the width and the height of the pixel have different contribution on the accuracy of the 3D point estimation from stereo. Figure 4 shows the comparison between the maximum possible estimation errors on each coordinate component with respect to the different pixel aspect ratios for a single pixel using the typical values mentioned in Table 1. The errors are calculated as the difference of the values obtained for (X_h, Y_h, Z_h) using Equations (5) and corresponding maximum deviated values obtained from following equations (b_{xD} is assumed to be equal to b_{xD}):

$$\hat{Z}_h^{\max} = \frac{(f_h + d)b_{xh}}{(x_{rD} - x_{lD} - e_x)}$$

$$\hat{X}_h^{\max} = \frac{x_{rD} + (e_x/2)}{(x_{rD} - x_{lD} - e_x)}, \quad \hat{Y}_h^{\max} = \frac{y_D + (e_y/2)}{(x_{rD} - x_{lD} - e_x)}. \quad (14)$$

Figure 4 shows that although the estimation error for Y component is a little increased with smaller aspect ratios, the average estimation error is decreased. Particularly, the error in estimation of Z is considerably decreased which means that a finer horizontal discretization on display screen yields in a finer depth resolution.

We have generalized this concept in our previous research [21] and through theoretical analysis shown that in general for a typical stereo setup and a given total resolution R , a more finer horizontal discretization (e_x) versus vertical discretization (e_y) yields in less 3D point estimation error. This has been achieved by putting an upper bound on the depth estimation error and then minimizing the error with respect to horizontal to vertical discretization ratio. For a typical parallel stereo setup the procedure can be briefly described as follow (see [21] for details):

$$\hat{Z} = \frac{fb_x}{\hat{x}_r - \hat{x}_l} = \frac{fb_x}{(x_r - x_l) \pm e_x} = Z \left(1 \pm e_x \frac{Z}{fb_x} \right)^{-1}. \quad (15)$$

Assuming higher order terms in Taylor expansion of (15) are negligible:

$$\hat{Z} \cong Z \left(1 \pm \frac{e_x Z}{fb_x} \right), \quad (16)$$

and then

$$\hat{Y} = Z \frac{\hat{y}}{f} \cong \frac{Z}{f} \left(1 \pm \frac{e_x Z}{fb_x} \right) \left(y \pm \frac{e_y}{2} \right) \quad (17)$$

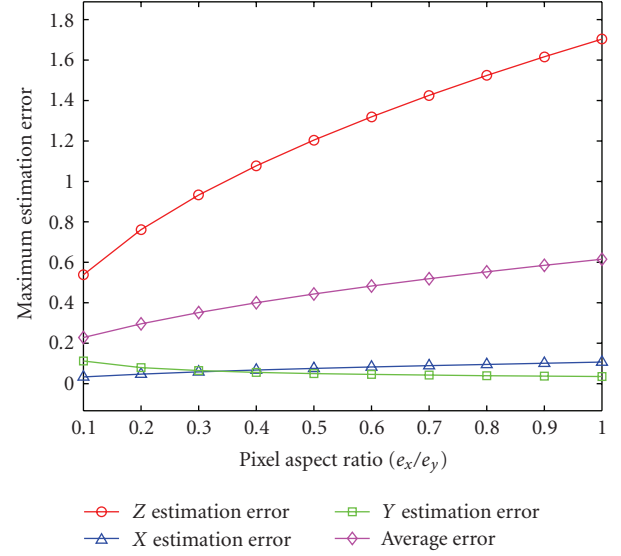


FIGURE 4: Comparison of maximum error on estimation of 3D point coordinate components at different pixel aspect ratios.

TABLE 1: Values used for calculating maximum error on estimation of 3D point coordinate components.

Parameter	Value	Parameter	Value
f_h	17 mm	R	17.76 pixel/mm ²
b_{xh}	65 mm	d	500 mm
x_r	22 mm	disparity	22 mm
y	0 mm		

or

$$\left| \frac{\hat{Y} - Y}{Y} \right| \leq f_p(e_x) = \left\{ \frac{e_x Z}{fb_x} + \frac{e_y}{2|y|} + \frac{e_x e_y Z}{2f|y|b_x} \right\}. \quad (18)$$

Considering a unit viewing or image capture area:

$$\left(\frac{1}{e_x} \right) \left(\frac{1}{e_y} \right) = R \quad \text{or} \quad e_y = \frac{1}{e_x R}. \quad (19)$$

Thus, from (18) and (19) we can restate the obtained upper bound on relative estimation error on Y component as:

$$f_p(e_x) = \left\{ \left(\frac{Z}{fb_x} \right) e_x + \left(\frac{1}{2R|y|} \right) \frac{1}{e_x} + \frac{Z}{2f|y|Rb_x} \right\}. \quad (20)$$

Minimizing equation (20) with respect to e_x gives the optimal pixel width (and then optimal pixel aspect ratio) for a single 3D point. In practice the estimation error need to be minimized over a reasonable viewing volume and some other parameters such as vergence may also be included in the optimization process (see [22, 23]).

In [19] we have applied this theory to the above-mentioned stereo setup (Equations (5)) and shown for a typical desktop computer or laptop the 2 : 3 pixel aspect ratio (i.e., three horizontal versus two vertical pixels) gives a better 3D visual experience than the uniform (square)

TABLE 2: S3D-Cursor user evaluation criteria.

Experiment	Criterion
Experiment 1	C1: Stereo mode gives better 3D visual experience C2: S3D-Cursor helps in better 3D experience
Experiment 2	C3: Object (boundaries) are better distinguishable in stereo mode C4: S3D-Cursor improves distinguishability
Overall	C5: Simplicity of using S3D-Cursor C6: Satisfaction and usefulness of S3D-Cursor

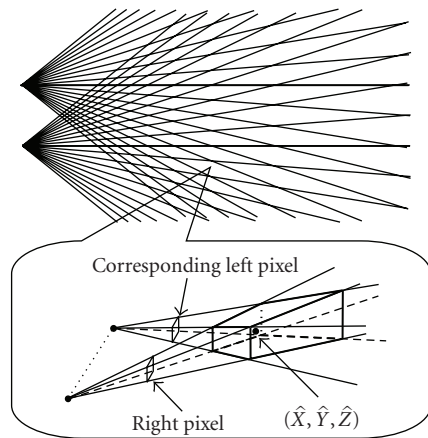


FIGURE 5: Top: Resolution pattern for parallel-stereo geometry. Bottom: 3D view of a typical voxel formed by two corresponding pixels.

pixel distribution. Applying the same theory to the stereo mouse cursor, we can say that a finer horizontal resolution not only yields in a better 3D visualization but also more accurate stereo-based 3D pointing especially across the depth dimension. However, this is a hardware solution which requires establishing new standards for stereo capturing and displaying devices.

Another issue related to the stereo 3D mouse accuracy is its heterogeneous behaviour mainly across the depth dimension. This fundamentally results from the intrinsic behaviour of the perspective projection plus the discretized nature of digital images. As illustrated in Figure 5, all points located inside the 3D diamonds (voxels) formed by the corresponding pixels are estimated to the same 3D point. These voxels are non-uniformly distributed so that the resolution (especially the depth resolution) decreases with the distance to the stereo cameras (viewer) [24]. As result, the stereo mouse cursor is not accurate enough when the objects are not close enough to the viewer. Several virtual enhancements may be applied to compensate for this drawback up to some acceptable extent. Zooming across the third dimension (*bringing the target to the cursor*), enlarging the objects across the depth (*changing the target size across the depth, i.e., changing the D parameter*), or manipulating the target under a fish-eye implementation (which again may be interpreted as some sort of *bringing target to the cursor*) are among possible enhancements. Another possibility is using the 3D object information when

such information is available. This is different from simply calculating the underneath disparity and can be quite helpful for disambiguation of the targeted object where the objects are not close enough to be distinguished by the amount of the disparity. In this situation, the closest 3D object (or 3D object element) to the estimated 3D cursor position may be prioritized as the targeted object.

7. Practical Application of S3D-Cursor

Stereo 3D mouse cursor can be used in a wide range of applications including 3D computer games and 3D objects manipulation. To realize the applicability of the S3D-Cursor on manipulating 3D objects we have developed a simple 3D object manipulation toolset which contains a marking (a pen) and demarking (an eraser) tool with a few auxiliary displaying-state control tools which all together allow the user to mark/demark a desired 3D point in the virtual stereoscopic 3D space.

Figure 6 shows a sample result of applying the 3D editing toolset (automatic mode) for ground-truth data measurement. Here the S3D-Cursor is used to specify a 3D-contour surrounding a TB cavity on a 3D lung model. The red-blue mode of QSplatMV is used in order to provide stereo effect of the model in a virtual 3D environment. The left lung image shows the red-blue representation and the right image shows the corresponding single-view version of the same model and the same contour defined in the red-blue visualization mode. Although these images are degraded due to down-scaling, the reader should still be able to watch the stereo effect in 3D using simple anaglyph glasses and compare it with the corresponding 2D image.

7.1. User Evaluation. We used the QSplatMV and its editing toolset to evaluate stereoscopic visualization and manipulation of 3D objects versus the similar tasks in corresponding ordinary perspective representation. Again the red-blue mode of QSplatMV is used to provide the stereo effect. We conducted two sets of experiments. Table 2 shows a brief description of the criteria used in these experiments. None of the people who participated in these experiments had former experience of working with stereo cursor, but many of them had one time or more experience of watching stereo contents. Moreover, some of them were already familiar with some 3D input devices such as space navigator [10] and a few with some line cursors used to point to a 3D location within the CAVE VR systems [25].

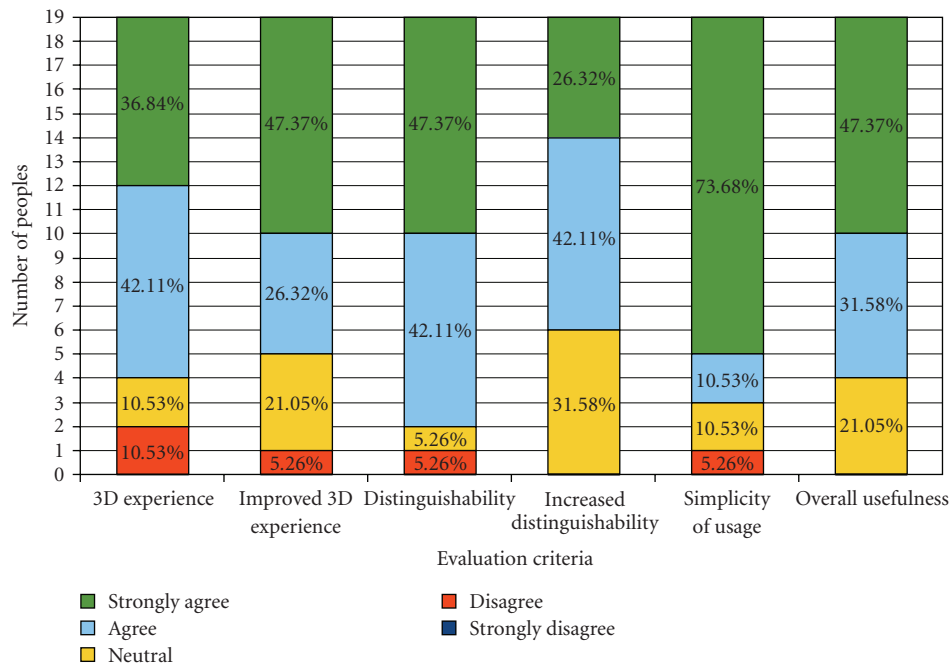


FIGURE 7: S3D-Cursor user evaluation results.

In the first experiment the peoples are asked to compare their visual experience with 3D stereo visualization versus 2D visualization. The purpose of this experiment was to show that if the people have better understanding of the 3D shape of the objects when they are able to touch the surface of the objects using S3D-Cursor. Three different models, that is, Lucy, Dragon, and Donna randomly chose and used for performing this experiment. As the first criterion in this experiment we wanted to make sure that the participants are able to observe the 3D effects of the stereo using anaglyph glasses and also to separate the 3D feeling caused by stereo and those possible 3D experience that may be caused by 3D cursor movements itself. Even though the red-blue representation may cause some ghosting effects for some peoples, the majority of peoples (strongly) agree that they

have better 3D experience with stereo mode without the intervention of the 3D cursor (see Figure 7, column 1). The second criterion asks the peoples if the S3D-Cursor helps in a better 3D visual experience when they touch different parts of the object. Again majority of users believe that when they slide the cursor over the 3D object, the depth movements of the 3D cursor gives them a better sense of its 3D shape (Figure 7, column 2). According to these results, contrary to the 2D cursor which may cause some disturbance in the formation of the 3D environment, the S3D-Cursor helps in better visual experience via giving the users the interesting feeling of touching the surface of the virtual 3D object.

In the second experiment we asked the user to perform the same manipulation task over the 3D Lung model in both single and stereo view modes. In fact, we asked them

to mark the boundaries of a TB cavity in both 2D and 3D representations. Again our user surveys suggest that the S3D-Cursor technique provides better depth information (Figure 7 column 3) and the users can specify the region of interest more accurately compared to using a 2D mouse cursor on a 2D display (Figure 7 column 4). In fact, 3D visualization allows the user to better distinguish the convex and concave surfaces and the border areas.

Finally, we asked the users to express their overall opinion in terms of the simplicity of usage and the usability of the S3D-Cursor through their short work experience with S3D-Cursor. The majority of people strongly agreed that it can be used as simple as an ordinary mouse and many of them have found it a useful device for manipulating 3D objects.

8. Conclusion and Future Work

In this paper we discussed on different theoretical aspects of a stereo-based 3D pointing technique which enables the users to point to an arbitrary location inside the 3D space composed by a stereoscopic 3D display and described our approach toward the implementation of such a 3D cursor. We also discussed how such a technique satisfies all the requirements of an abstract 3D cursor, so that it potentially can be considered as a simple extension of 2D cursor to the 3D stereoscopic space. The user experience with the S3D-Cursor has shown us that it has great potential for interaction with this virtual space. Here we showed its usefulness and efficiency in working with 3D objects and presented some promising results. However, further improvement is necessary to overcome some minor drawbacks like low accuracy in the farther depths. This can be achieved through the implementation of suggested virtual enhancements. Also we need to improve our UI tools for further smart manipulation of 3D contours, surfaces and volumes based on the more advanced new colorful stereoscopic visualization techniques.

Acknowledgments

The authors wish to thank iCORE and NSERC for financial support, the Stanford Computer Graphics Laboratory and the Stereolithography Archive at Clemson University, VCG-ISTI (the AIM@SHAPE Shape Repository) for providing 3D models, and Alexey Abadlov for extracting the 3D lung model.

References

- [1] N. Holliman, "3D display systems," Department of Computer Science, University of Durham, August 2009, <http://www.dur.ac.uk/n.s.holliman/Presentations/3dv3-0.pdf>.
- [2] N. A. Dodgson, "Autostereoscopic 3D displays," *Computer*, vol. 38, no. 8, pp. 31–36, 2005.
- [3] S. E. B. Sorensen, P. S. Hansen, N. L. Sorensen, et al., "Method for recording and viewing stereoscopic images in color using multichrome filters," US patent no. 6687003, August 2009, <http://www.patentstorm.us/patents/6687003.html>.
- [4] N.-T. Dang, "A survey and classification of 3D pointing techniques," in *Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future (RIVF '07)*, pp. 71–80, Hanoi, Vietnam, March 2007.
- [5] P. Best, "Ultimate 3D design navigation tool," *Gizmag Emerging Technology Magazine*, August 2009.
- [6] "Mouse (Computing)," August 2009, [http://en.wikipedia.org/wiki/Mouse_\(computing\)](http://en.wikipedia.org/wiki/Mouse_(computing)).
- [7] L. Blain, "The 3D air-mouse you wear as a ring," *Gizmag Emerging Technology Magazine*, August 2009.
- [8] S. Rusinkiewicz and M. Levoy, "QSplat: a multiresolution point rendering system for large meshes," in *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics (SIGGRAPH '00)*, pp. 343–352, New Orleans, La, USA, July 2000.
- [9] C. Hand, "A survey of 3-D input devices," Tech. Rep. TR94/2, Department of Computer Science, De Montfort University, Leicester, UK, 1994.
- [10] "What is a 3D mouse?" August 2009, http://www.3dconnexion.com/3dmouse/what_is_3dmouse.php.
- [11] B. Frohlich, J. Hochstrate, A. Kulik, and A. Huckauf, "On 3D input devices," *IEEE Computer Graphics and Applications*, vol. 26, no. 2, pp. 15–19, 2006.
- [12] C. Hand, "A survey of 3D interaction techniques," *Computer Graphics Forum*, vol. 16, no. 5, pp. 269–281, 1997.
- [13] "Orthoengine 3D stereo," Technical Specification, August 2009, http://www.pcigeomatics.com/pdfs/3D_Stereo.pdf.
- [14] "Biomedcache user guide," Software User Guide, August 2009, <http://www.ch.ic.ac.uk/local/organic/medchem/UserGuide.pdf>.
- [15] S. A. Safier and M. W. Siegel, "3D-stereoscopic X windows," in *The International Society for Optical Engineering*, vol. 2409 of *Proceedings of SPIE*, pp. 160–167, San Jose, Calif, USA, 1995.
- [16] F. Steinicke, G. Bruder, K. Hinrichs, and T. Ropinski, "Simultane 2D/3D user interface konzepte für autostereoskopische desktop-VR systeme," in *Proceedings of the 4. GI-Workshop AR/VR*, pp. 125–132, Shaker, Organization: GI-Fachgruppe VR/AR, 2007.
- [17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, London, UK, 2003.
- [18] H. Sahabi and A. Basu, "Analysis of error in depth perception with vergence and spatially varying sensing," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 447–461, 1996.
- [19] H. Azari, I. Cheng, and A. Basu, "Optimal pixel aspect ratio for stereoscopic 3D displays under practical viewing conditions," in *Proceedings of the 3DTV Conference: The True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON '09)*, pp. 1–4, Potsdam, Germany, May 2009.
- [20] K. E. Hoff, "Deriving the OpenGL perspective depth transformation," Tech. Rep., August 2009, <http://www.cs.unc.edu/~geom/HOFF/techrep/perspective.doc>.
- [21] A. Basu, "Optimal discretization for stereo reconstruction," *Pattern Recognition Letters*, vol. 13, no. 11, pp. 813–820, 1992.
- [22] I. Cheng, K. Daniilidis, and A. Basu, "Optimal aspect ratio under vergence for 3D TV," in *Proceedings of the 3DTV-Conference: The True Vision—Capture, Transmission and Display of 3D Video (3DTV-CON '08)*, pp. 209–212, Istanbul, Turkey, May 2008.
- [23] I. Cheng and A. Basu, "Optimal aspect ratio for 3D TV," in *Proceedings of the 1st International Conference on 3DTV (3DTV-CON '07)*, pp. 1–4, Kos, Greece, May 2007.

- [24] L. Matthies and S. A. Shafer, "Error modeling in stereo navigation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp. 239–248, 1987.
- [25] S. Zhang, Ç. Demiralp, D. F. Keefe, et al., "An immersive virtual environment for DT-MRI volume visualization applications: a case study," in *Proceedings of the IEEE Visualization Conference*, pp. 437–440, 2001.

