31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

Stereo 3D Mouse (S3D-Mouse): Measuring Ground Truth for Medical Data in a Virtual 3D Space

Hossein Azari, *Student Member*, Irene Cheng, *Member*, and Anup Basu, *Senior Member, IEEE*

Abstract— **We introduce a novel approach of applying stereoscopy principles to simulate a virtual 3D pointing device based on two or more views of an ordinary mouse cursor. The system assists a user point to a desired location inside the virtual 3D space projected by a stereoscopic display. The technique is designed for easy perception, and thus interpretation, of depth information in a 3D context. A target application is the measuring of ground truth for 3D medical data. To illustrate how our virtual stereoscopic cursor works, we have implemented a multi-view stereo visualization software and a simple 3D editing toolset for manipulating 3D objects. Experimental results suggest the effectiveness of using the device in term of detection accuracy and user satisfaction compared to using an ordinary mouse on a conventional 2D screen.**

I. Introduction

USING Computer Aided Detection (CAD) techniques for processing 3D medical data is a part of many state-of-the-art medical diagnostic procedures. Automatic detection and segmentation of anatomical structures are examples of CAD algorithms that are useful for eliminating a large number of false positives and help the clinicians to prioritize their limited resources. However, before applying CAD algorithms for clinical use, it is necessary to assess how precise the computer generated measurements align with the features manually detected by the clinicians. This is an important step in making sure that doctors can rely on CAD algorithms and make decisions based on accurate information. The question becomes how to verify these feature extraction algorithms and ensure that they are consistent with the ground truth defined by the clinicians. Manual or semi-automatic extraction of ground-truth data from 2D-images is straightforward considering the sophisticated user tools available on 2D interfaces. In [5] we have introduced a methodology for dealing with such 2D data. However, measuring ground-truth on 3D data can be quite complex and erroneous because of the weak representation of the third dimension (depth information) on a conventional 2D display.

A common approach currently used by researchers is to divide the 3D object into a sequence of 2D slices and ask the experts to draw a contour on each slice. A 3D surface is then constructed using the set of 2D contours, which can then serve as the ground truth for volumetric measurement. This process is very labor intensive and is not feasible given the clinical resource constraints. Despite the excellent research conducted in the area of 3D visualization, there is no simple and direct means to allow clinicians to define the ground truth in 3D space efficiently.

In this paper, we take advantage of the advances in 3D display technologies, such as multiview autostereoscopic displays [1] and ColorCode3D glasses [6], which enable the viewers to experience much more realistic representation of 3D content. We introduce a 3D virtual pointing device or 3D mouse cursor, which we call Stereo 3D Mouse (S3D-Mouse), employing stereoscopic principles to form a 3D cursor in the virtual space projected by the stereoscopic display device. Differing from other 3D mouse devices, which in essence have no difference from the functionalities of a conventional computer mouse, S3D-Mouse allows the user to place the cursor at a 3D location instead of a 2D point. On the other hand, its use is simple enough so that the user working in a 3D space feels like using an ordinary mouse in 2D space. Although the target application of this novel method is to exploit stereoscopy for manipulating 3D medical imagery, S3D-Mouse can be used as a general interaction tool in other virtual 3D applications.

The rest of this paper is organized as follow. In Section 2, we discuss related techniques. Section 3 explains the mathematical model behind the S3D-Mouse. Section 4 is devoted to implementation issues and accuracy analysis. In Section 5, we discuss the application of using S3D-Mouse to extract ground-truth data for evaluating the accuracy of boundary detection and reconstruction algorithms. Finally, Section 6 gives the conclusion and future work.

II. Background of 3D mouse

Several attempts have been made in recent years to simplify interaction with 3D application environments. These include invention of different types of 3D mouse devices such as 3D space navigator, OptiBurst^{TM}, and 3D air mouse [8] which employ some sort of mechanical, ultrasonic, or IR tracking techniques to introduce more natural ways of working with 3D applications. However, these 3D devices just simplify the functions essentially executable by using a conventional mouse, but the visible mouse cursor, if any, are still only capable of representing a 2D location of the display screen. The S3D-Mouse is a virtual 3D pointing device composed of two or more views of the same mouse cursor presented on the (auto)stereoscopic display at a specific

Manuscript received April 7, 2009. This work was supported in part by ASRA, iCORE from Alberta and NSERC, CANADA. Hossein Azari, Irene Cheng are Anup Basu are with the department of Computing Science, University of Alberta, emails: {hazari,lin,anup}@cs.ualberta.ca.

978-1-4244-3296-7/09/$25.00 ©2009 IEEE

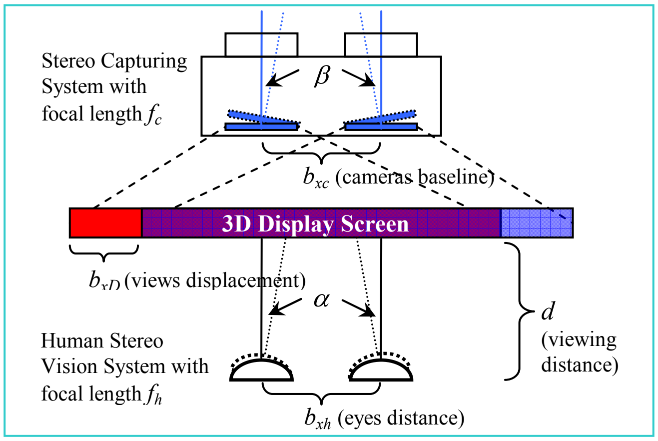

Fig. 1: Process of capturing, display, and watching stereo images.

disparity. When the user watches the stereoscopic display, the 3D cursor is created along with the formation of the virtual 3D space. The technique adds another degree of freedom to the mouse cursor itself so that it enables the user to point to a 3D location in the virtual space. It does not require any assumption about the pointing device hardware and is fully compatible with the current functionality of commercial 2D and even 3D mice available in the market.

III. S3D-Mouse Mathematical Model

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. Either of these two systems can have its own camera configuration, *i.e.* parallel or with vergence, independent of the other system. In the simplest case we may assume parallel geometry for both capturing and viewing sides. In this case, the projection of a 3D-point (X_c,Y_c,Z_c) of the camera space on the left and right camera image planes are given by:

$$x_{rc} = \frac{f_c X_c}{Z_c}, x_{lc} = \frac{f_c(X_c - b_{xc})}{Z_c}, y_c = y_{lc} = y_{rc} = \frac{f_c Y_c}{Z_c} \quad (1)$$

Assuming the images are scaled by a factor S when presented on the display, the corresponding 2D point coordinates on the display screen can be computed as:

$$x_{rD} = Sx_{rc}, \quad x_{lD} = Sx_{lc}, \quad y_D = Sy_c \quad (2)$$

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

$$x_{rh} = \frac{f_h x_{rD}}{(f_h + d)}, x_{lh} = \frac{f_h(x_{lD} + b_{xD} - b_{xh})}{(f_h + d)}, y_h = \frac{f_h y_D}{(f_h + d)} \quad (3)$$

From Equation (3), the 3D point reconstructed by the human eyes is theoretically given by:

$$Z_h = \frac{f_h b_{xh}}{x_{rh} - x_{lh}} = \frac{(f_h + d) b_{xh}}{(x_{rD} - x_{lD} - b_{xD} + b_{xh})},$$
$$X_h = Z_h \frac{x_{rh}}{f_h} = Z_h \frac{x_{rD}}{(f_h + d)}, Y_h = Z_h \frac{y_h}{f_h} = Z_h \frac{y_D}{(f_h + d)} \quad (4)$$

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. In [4] we have shown that Equation (4) can also be applied for estimating the 3D point reconstructed by human eyes in the vergence case. In other words, if the stereo images are captured under parallel configuration, the 3D-scene reconstructed by human eyes does not depend on the amount of vergence between the eyes.

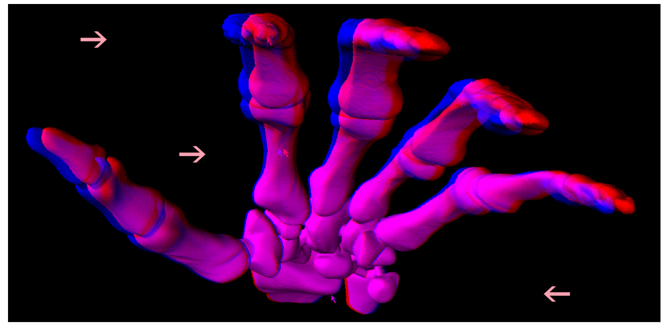

Fig. 2: A snapshot of Hand skeleton red-blue visualization with three samples of stereo cursors. Follow the direction of arrows to find them.

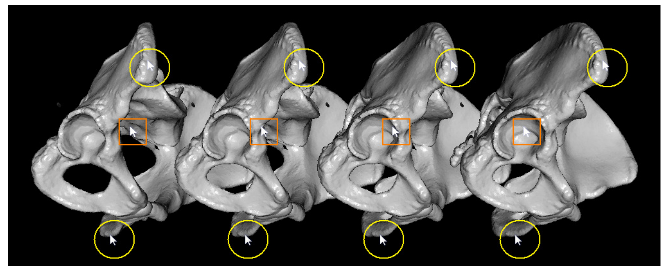

Fig. 3: Donna four-view stereo with three samples of stereo cursors on occluded (squares) and non-occluded (circles) areas.

When the stereo images are captured under vergence, the corresponding projections on the display screen do not locate on the same raster line. However, as we discussed in [4], we may assume that eyes compensate for vertical differences of the corresponding points so that the 3D point location estimation by human eyes mainly depends on the horizontal disparity of the corresponding projections. Based on this assumption, Equations (4) can be used as a good approximate model for 3D point estimation by human eyes via stereo images presented on a stereoscopic device.

Applying this theory to the mouse cursor and managing the amount of disparity appropriately provides an efficient virtual 3D-pointing device useful for interaction with many 3D application environments including 3D medical image visualization and manipulation.

IV. S3D-Mouse Implementation

In order to illustrate the functionalities of the S3D-Mouse, we implemented an underlying (multiview) stereo rendering application. For this purpose, we have extended the QSplat, a real-time point-based rendering program [3], to support rendering multiple views of a 3D object. QSplat uses a progressive hierarchical method especially useful to render large scale geometric models. Our extended version, 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 between the views on the display screen (b_{xD}). The current version assumes parallel configuration and the same baseline for all cameras. This simple implementation facilitates camera rotation and

Table 1: Values used for calculating max error on estimation of 3D point.					
f_h	17 mm	x_r	22 mm	d	500 mm
b_{xh}	65 mm	y	0 mm	$disp$	22 mm
R	17.76 pixel/mm^2				

translation, as well as the cursors' disparities calculations. The system also supports a special red/blue rendering mode which gives the flexibility of using the application on all conventional displays.

A. *Binocular S3D-Mouse Implementation*

The stereo mouse cursor is implemented on top of QSplatMV. Two different disparity adjustment modes are available for the cursors: manual and automatic. The *automatic* mode assumes that the depth of each pixel is known or 3D model is available. In this mode, one view (say the left one) is considered as the reference view. When the left mouse cursor points to a pixel in the left view, the corresponding pixel on the right view (or the position of the right cursor) can be determined using the basic stereo imaging computation explained below.

Considering the OpenGL default perspective projection, which implies a normalization as well [7], the relationship between the point depth in the virtual camera coordinate Z_c and the depth maintained in the depth buffer Z_w is given by:

$$Z_c = \frac{f_c d_{far}}{Z_w(d_{far} - f_c) - f_c} \tag{5}$$

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 Equation (1) Z_c can be determined as:

$$Z_c = \frac{f_c b_{xc}}{x_{rc} - x_{lc}} = \frac{f_c b_{xc}}{disp} \tag{6}$$

Finally, from (5) and (6) the amount of the disparity is given by:

$$disp = \frac{(Z_w(d_{far} - f_c) - d_{far})b_{xc}}{d_{far}} \tag{7}$$

In practice, when the user slides the cursor over the display screen the automatic disparity adjustment based on the values calculated from formula (7) generates the illusion of touching the surface of a real 3D object so that the user perceives the 3D cursor sinking into the concavities or climbing on top of the convex hulls. Figure 2 shows a snapshot of hand skeleton visualized in red-blue with three samples of stereo cursors with different disparities. Viewing this figure in color using anaglyph glasses will show how a 3D cursor is formed at different distances in front of the user.

Implementing the automatic disparity adjustment over 2D image stereo pairs, where 3D model is unavailable, implies that an efficient stereo matching algorithm be incorporated into the system in order to match the corresponding projections in the left and right views. In contrast to classical stereo matching, here the correspondence needs to be established only for the pixel located under the current position of the left (or right) mouse cursor.

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. In this way, the user is able to have better depth perception or manipulate inside of the object.

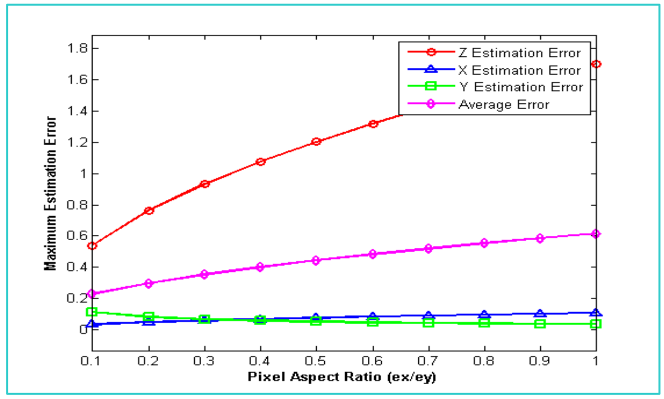

Fig. 4: Comparison of maximum error on estimation of 3D point coordinate components at different pixel aspect ratios.

B. *Multiview Implementation*

The multiview implementation is particularly useful for autostereoscopic multiview displays. The implementation is essentially similar to the two-view case. Assuming all cameras are parallel and located on the same baseline, again one of the views can be considered as reference for disparity calculation and the other projections can be determined with respect to the reference view. Although, this implementation is simple, it may cause some problems in occluded areas. As depicted in Figure 3, the implementation works fine as long as the corresponding projections are visible in all views (cursors inside the circles). The implementation becomes problematic when corresponding projections fall into occluded areas in two or more consecutive views (observe cursors inside the squares). When the viewer moves his or her head to watch the second and third 3D views, the reconstructed 3D cursor is incorrectly estimated. A more accurate algorithm should detect the occluded areas and hide the cursor in corresponding views.

C. *S3D-Mouse Accuracy Analysis*

Assuming f_h, d, and b_{xh} are constant values, the accuracy of the reconstructed stereoscopic 3D points is dictated by the size of the pixels or the resolution (width and height) of the display. In fact, as understood from Equations 4, the width (e_x) and the height (e_y) 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 error on each coordinate component with respect to different pixel aspect ratios (e_x/e_y) for a single 3D point using the typical values mentioned in Table 1. R is the fixed total resolution. The errors are calculated as the difference of the values obtained from Equations 4 and the corresponding maximum deviation values obtained from the following equations:

$$\hat{Z}_h^{\max} = \frac{(f_h + d)b_{xh}}{(x_{rD} - x_{lD} - e_x)}$$
$$\hat{X}_h^{\max} = \frac{x_{rD} + (e_x/2)}{(x_{rD} - x_{lD} - e_x)}, \quad \hat{Y}_h^{\max} = \frac{y_D + (e_y/2)}{(x_{rD} - x_{lD} - e_x)} \tag{8}$$

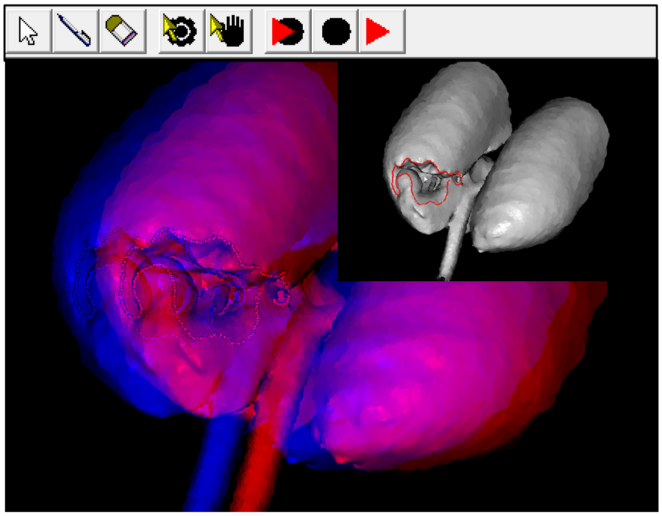

Fig. 5: Marking the TB cavity boundaries on a 3D lung model using S3D toolset.

Figure 4 shows that although the estimation error of *Y* is slightly increased for smaller aspect ratios, the average estimation error and particularly the error in the estimation of *Z* is considerably decreased, which means that a finer horizontal discretization on display screen yields a finer stereoscopic resolution. Applying this to a viewing volume, we have shown previously that in general, given a total resolution *R*, a finer horizontal (e_x) versus vertical (e_y) discretization produces smaller 3D point estimation error, or equivalently a better 3D visualization (see [4] for details). Applying the same theory to the stereo mouse cursor, we can say that a finer horizontal resolution yields a more accurate stereo-based 3D pointing device as well.

Another issue related to the stereo 3D mouse accuracy is its non-uniform behavior across the depth dimension. This is the result of the intrinsic behavior of the perspective projection. The farther the viewer is from the stereo camera, the less is the depth resolution. As a result, the stereo mouse is not accurate enough when the objects are far away from the viewer. This drawback can be compensated to a certain extent by zooming into 3D objects.

V. Using S3D-Mouse in Measuring Ground-Truth

To use the S3D-Mouse for ground-truth data measurement, we have developed a simple 3D object manipulation toolset (Figure 5-top). The toolset contains a marking (a pen) and demarking (a rubber) tool with a few auxiliary state control tools which collectively allow the user to mark or demark a desired 3D point in the virtual 3D space.

Figure 5 show sample results of applying the S3D editing toolset (automatic mode) 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 bigger lung image shows the red-blue representation and the smaller image in the upper-right corner 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. In fact the 3D-visualization allows the user to better distinguish the convex and concave surfaces and the border areas. Moreover, our user surveys suggest that the depth movements of the S3D-Mouse cursor in virtual 3D space is really helpful in efficiently specifying the region of interest compared to using a normal mouse on a 2D display.

A new tool can be included in the toolset for highlighting the 3D-surface surrounding a specific volume, similar to the magic wand used in region growing based on a given stopping condition. However, the current toolset is sufficient for our evaluation purpose. For volume comparison, a 3D curve matching technique such as least squares [2] can be adopted to find the corresponding points between the ground-truth and the surface contour surrounding the segmented volume. The Euclidean distance between the corresponding contour points can then be used as a metric to compute the deviation between the ground truth and the algorithm generated measurement.

VI. Conclusion and Future Work

In this paper we introduced our stereo-based 3D cursor, S3D-Mouse, as an appropriate tool for extracting 3D ground-truth data, which is essential in order to verify the accuracy of algorithm-generated results. Our user surveys support the advantages of using our technique compared to an ordinary mouse cursor on a 2D screen. However, further developments are necessary to realize the full functionalities. These include improvement of the accuracy in farther depths, more efficiently handling the occluded areas, and developing more advanced stereoscopic visualization techniques.

Acknowledgment

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

References

[1] N.A. Dodgson, “Autostereoscopic 3D Displays”, Computer, vol.38, no.8, pp. 31-36, Aug. 2005.

[2] A. Gruen, and D. Akca, “Least squares 3D surface and curve matching”, ISPRS Journal of Photogrammetry and Remote Sensing, 2005, vol. 59, pp. 151-174.

[3] S. Rusinkiewicz, and M. Levoy, “QSplat: A Multiresolution Point Rendering System for Large Meshes”, SIGGRAPH, 343-352, 2000.

[4] H. Azari, I. Cheng, and A. Basu, “Optimal Pixel Aspect Ratio for Stereoscopic 3D Displays under Practical Viewing Conditions”, 3DTV-CONFERENCE, May 2009, Potsdam, Germany.

[5] I. Cheng, C. Flores-Mir, P. Major and, A. Basu, "Measuring and Evaluating Ground Truth for Boundary Detection in Medical Images," IEEE Engineering in Medicine and Biology Conference, 2008.

[6] S.E.B. Sorensen et al., “Method for recording and viewing stereoscopic images in color using multichrome filters”, US Patent 6687003, http://www.patentstorm.us/patents/6687003.html.

[7] K. E. Hoff, “Deriving the OpenGL Perspective Depth Transformation”, Technical Report, Spring 2000, Available at: http://www.cs.unc.edu/~geom/HOFF/techrep/perspective.doc.

[8] Mouse (Computing), Wikipedia Article, Available at: http://en.wikipedia.org/wiki/Mouse_(computing).