

# Surgeon's Magic Wand: A Screen Pointing Interactive Method

**Naren Vira, Professor**

Department of Mechanical Engineering  
Howard University, Washington, D.C. 20059

**Shaleen Vira, Student**

College of Arts and Science  
New York University, New York, N.Y. 1000

**Abstract:** *A novel, non-touch, screen pointing "magic wand" interface is proposed for surgeon's use in an environment requiring simultaneous display of several patients' data over a continuous period of time. The magic wand or passive pointing device does not have any active energy source within it (as opposed to a laser pointer) and thus cannot easily be detected or identified. Thus, modeling and simulation task is carried out by generating high resolution color images of a pointer viewing via two digital cameras with a popular three-dimensional (3D) computer graphics and animation program, Studio 3D Max by Discreet is used for detection. These images are then retrieved for analysis into a Microsoft's Visual C++ program developed based on the theory of image triangulation. The program outputs the precise coordinates of the surgeon's wand in the 3D space along with its projection on a large view screen. The computational results of the pointer projection are compared to the known locations specified by the Studio 3D Max for different simulated configurations. High pointing accuracy is achieved: a pointer kept 30 feet away correctly hits the target location within a few inches. This preliminary work will lead to a complex development for interactive hand pointing gesture recognition system and its applicability to a large viewing display environment.*

**Keywords:** Surgeon's wand, screen pointing interface, passive pointer, image processing, hand pointing gesture

## 1.0 Introduction

Advances in medical research and technology drive the future of medicine, holding the promise of earlier and more accurate diagnosis of disease as well as safer and more effective treatments. Access to information remains crucial for delivering the most advanced care to patients. Despite a revolution in medical technology in the past few decades, access to mission-critical patient data continues to be inadequate in most settings for most physicians. As the health care information technology industry has evolved, information has been divided between numerous disparate systems with no common viewing portal. A physician confronted with the need to make fast critical patient care decisions often is forced to spend time "hunting and gathering" for clinically relevant data [1]. Data in the form of electronic medical records, digital images, video streams, radiology reports, patient records and pharmacy documents are available to the medical team via disparate systems. When the clinician must make patient care decisions outside the medical facility-from home, office or any other remote location, access to information is even more limited. But advances in computer and communication technologies make it possible to craft 21st century solutions to these problems, bridging the gap between the clinician and vital medical data. Thus, to address the issue of "putting together" necessary patient's information at one place, common-viewing large displays are now commercially available in the market place [1].

Rapid improvements in CPU performance, storage density, and network bandwidth have provided sufficient bandwidth and computational resources to support high-resolution displays and natural human-computer interactions. Nowadays, the main bandwidth bottleneck in an interactive computer system occurs in the link between computer and human, not between computer components within the system.



Figure 1: Large Display at Traffic Management Center

The large display devices, such as projectors and flat panels, are rapidly becoming commodity items. Figure 1 shows usage of a large display system at a typical traffic management center. Meanwhile, new display technologies, such as organic light-emitting diodes (OLED), will soon become available at inexpensive prices. They can be attached to almost any kind of surface, allowing unlimited freedom of design for the interiors and exteriors of rooms and buildings. We believe that new display technologies will revolutionize the way we use computers, making us rethink the relationship between information technology and our society. As an example, consider how wall-sized displays enable qualitatively different human-computer interactions than traditional desktop displays.

The research community and design industry have long been interested in interactions with large-format displays, with much of the early research focusing on single whiteboard-sized displays. More recently, the rapidly decreasing cost of projectors have spurred construction of wall-sized displays by tiling multiple projectors to form a single virtual image [2-8]. These multiple projector displays are particularly interesting from an interaction perspective in that the high resolution provided by the tiling of multiple projectors lets users view high-quality images even when they [the projectors] are up close to the display. The scenario of interaction with a large scale display system spanning an entire wall is completely different from single-user desktop applications. Specifically, one cannot rely on traditional control mechanisms using keyboard and mouse.

The laser pointers as pointing devices are used to indicate specific positions on viewing screens. Laser pointers utilize an active energy source, a concentrated photon/energy beam that streams from the device to the nearest physical object, hopefully the slide/screen. Occasionally, accidental pointing is hazardous. They are restricted in use as an interactive tool. Alternatively, the present research work demonstrates the use of a passive device, one that does not require any energy source. However, external detecting mechanisms to precisely identify where the pointer is pointing to are required. To achieve this requisite, two high resolution color cameras and image triangulation methodology for pointer detection analysis were employed. Figure 2 depicts conceptual utilization of surgeon's magic wand as an interactive pointing device.

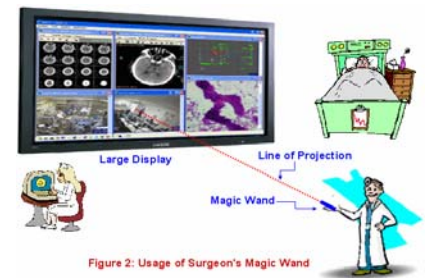


Figure 2: Usage of Surgeon's Magic Wand

In a standard graphical user interface, keyboard and mouse are typical input devices that are used to type commands, write text, and point and select graphical elements at specific locations of a graphic display. The graphical user interface can usually be categorized based on two-dimensional (icons are clickable to represent procedures or pieces of multimedia information) or three-dimensional (3D mouse or joystick) environments. The primary restriction on these interfaces is that the users have to sit in front of the computer monitor or view screen, thus limiting user mobility. Advanced user interfaces, such as those employed in many augmented and virtual reality applications, achieve higher user mobility [9]. The input devices for advanced interfaces can either be wearable devices (gloves, glasses or body marks) or non-wearable devices (microphones, laser pointers or vision cameras). Note that non-wearable devices are non-contact type. The audio processing and voice recognition software is needed in the case of microphone use whereas vision cameras require image processing software. These devices are non-intrusive, and can support natural interaction as the user can express commands and actions through voice and gestures in the same way as in everyday life.

An active research trend in computer vision is the development of robust, environment-independent tracking methodologies for the development of effective human-computer interfaces [10 & 11]. Several vision-based interaction approaches have been presented so far. The aim is to derive a semantic interpretation of human hand gestures and facial expressions [12]. The interaction by means of gesture languages to be recognized by computer vision technique provides several advantages: (1) there is no mechanical part in the interaction, avoiding the problems caused by degradation of those hardware parts; (2) recognition is remote: the user can be located far away from the perceptual and computational apparatus, no contact interaction is required; and (3) the definition of interaction language is very flexible: many variations could be defined for the same task in search of an optimal one.

Among most of the human-computer interfaces, the vision-based hand pointing systems appear to be particularly promising. Because hand pointing is an everyday life operation reflecting a specific interest in a specific portion of the visible space, it does not require any a priori skills or training, and is a perfect candidate for the design of a natural interaction device based on computer vision. Thus, to develop such an interface is our focus of the present research work. However, we first develop interface technology using "magic wand" as an interactive tool for large display environment. Such initial development can easily be adopted with minor modification for more complex interactive hand pointing interface and gesture recognition system.

Magic wands have a presence in the history and legends of human cultures from thousands of years ago all the way to the present day, and are surrounded by rich systems of belief. Often carried by wizards, these sticks, or in some cases large rods, focus magical strength. Some anthropologists believe that Stone Age cave paintings showing people with sticks are meant to portray leaders of clans holding wands to attest to their power. That is only a guess, but strong evidence goes back at least to the time of ancient Egypt, in

which hieroglyphs show priests holding small rods. In Greek mythology, the messenger of the Greek gods carries a special wand called a caduceus. This is a rod with wings, around which two serpents are twisted, meant to signify wisdom and healing powers. Physicians adopted it as their symbol hundreds of years ago and still use it today. This and more historical background on magic wands can be found in Ref [13].

Magic wands are simple objects that respond to human gesture, speech, emotion, and even thought, and thanks to modern-day books and movies, they are widely understood from an early age as symbols of great empowerment. Many who have experienced these stories may have gained somewhat of a mental model or an intuitive sense for how to use a magic wand, i.e. what kinds of gesture can be made with it or what words should be said to cast a spell ( “abracadabra!”, “hocus pocus!” ---). Considering these factors, the magic wand presents an interesting design opportunity as a form for a tangible computer interface.

This work is also a stepping stone for developing an intelligent non-touch computer screen interface. Let us visualize two web cameras mounted on top of a computer screen viewing the computer user. The camera can track a non-touch passive pointer or user’s finger as it approaches the screen. Once the camera and associated interface identify the pointing location on the screen, it can zoom in or out showing details as the finger, respectively, move towards or away from the screen. A simple example would be to view a geographical map with zooming in and out capability. The interface can also pop out or display additional details/ information, if needed, in another window of the pointing location. The example for this scenario would be its use in the Physician’s Dashboard where medical staff can instantly access a patient’s status and all relevant medical data by a single point clicking in the corresponding patient’s viewing window.

## 2.0 Description

The main system components of the human-computer interface for an intelligent interactive environment are shown in Figure 3. The computing system receives input from a pair of color cameras placed in a way that the user’s pointing action can be viewed for most of the work setting (to avoid occlusion). The user is holding a “passive pointer” or “magic wand” coded with two distinct colors as depicted in Figure 4. These colors are chosen for quick image processing and do not jeopardize underlying methodology. The pointer would be replaced by the user finger in the future work. The computing system performs image analysis using image triangulation technique and outputs interaction parameters reflecting user’s status. Based on computational analysis, the systems also send a specific color beam to a digital projector to identify the action of hand pointing (the point of hand projection). Different color beams are activated on the viewing screen when multiple users are employed (distinguishing who is pointing where).

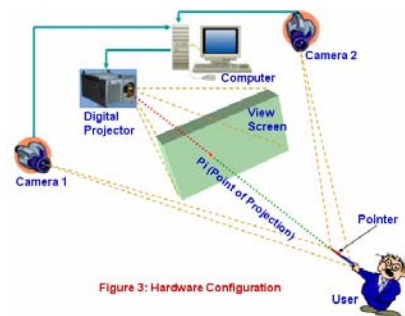


Figure 3: Hardware Configuration

## 3.0 Analysis

The basic geometry problem to be solved is how to compute the location of interest  $P_i$  in Figure 3 from image observation. The intensity of each image pixel in RGB color space for a pair of images can be represented in a vector form as

$$\mathbf{P}(I, J) = R(I, J) \mathbf{e}_1 + G(I, J) \mathbf{e}_2 + B(I, J) \mathbf{e}_3 \quad (1)$$

The symbol  $I$  and  $J$  stand for pixel coordinates, and  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  and  $\mathbf{e}_3$  are unit vectors along  $R$ ,  $G$ , and  $B$  color space, respectively. The terms  $R(I, J)$ ,  $G(I, J)$ , and  $B(I, J)$ , respectively, represent red, green and blue color intensities. As opposed to stereo matching algorithm (correspondence of every image pixel is found), here we are only interested in identifying those pixels that corresponds to the pointer in one image and respective matching pixels in another image viewed from a second camera. More precisely, if we marked the pointer’s ends with two distinct colors then only those pixels are required to be matched in both images. Without loss of generality, let us say that one end is marked with red color and other is with blue. Because we are only interested in matching the pointer’s red or blue color end pixels of each image, Equation (1) can be rewritten as  $\mathbf{P}(I, J) = R(I, J)$  for the red color end pixels and  $\mathbf{P}(I, J) = B(I, J)$  for the blue color end pixels. Alternatively, we scan the whole image to identify all pixel-coordinates  $I$  and  $J$  that represent either red or blue color end of the pointer. From this information, we compute the centroid of each color end. That is  $\mathbf{P}_1(I, J)$  and  $\mathbf{P}_2(I, J)$  centroids for the red color end as shown in Figure 5. The term centroid and mid point of the color end are interchangeable because of two dimensional coordinate system representation. We assume that the centroid points  $\mathbf{P}_1(I, J)$  and  $\mathbf{P}_2(I, J)$  represent the matching points. This assumption is valid because the pointer dimensions are relatively small with respect to the physical dimension of the room. Thus, the implication is that the process of disparity analysis needed for stereo matching is not required and the task of finding matching pixels is considerably simplified. The same analysis can be applied for finding the matching points

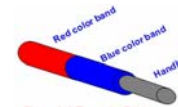


Figure 4: A Two-color Pointer

corresponding to the blue color end of the pointer. It should be emphasized that we deliberately chosen two distinct color-ends to simplify and speed up the process of image scanning. One can choose other pixel matching methods depending upon the application.

By knowing the x and y coordinates of each centroid point (after correlating image pixels with the space coordinates) of the pointer in a single image, we can mathematically pass a line through these two points to describe a pointer in a 2D space. Now the process of triangulation is required to compute the three-dimensional coordinates of the pointer from these two images (i.e., from four centroid points).

### 3.1 Three-dimensional Triangulation Technique

We apply ray casting analysis to triangulate three-dimensional coordinates of each image pixel point in a space as it is viewed by two cameras with respect to a chosen reference frame. Without loss of generality, the reference frame could be at one of the cameras' center. We have chosen the center location of camera 2 as the frame of reference. Each ray is cast from the viewpoint (in this case, center of the camera) through each pixel of the projection plane (in this case, image planes 1 and 2) into the volume dataset. The two rays wherever they intersect in a 3D space determine the coordinates of a point viewed by both cameras as shown in Figure 6. By connecting all intersecting points in the volume dataset, we can generate a 3D point cloud floating in a space. We utilize four points at a time (two in each image) to compute the three-dimensional coordinates of the pointer's end. Thus, the location of the pointer can be identified in a 3D space from the knowledge of its two ends.

The computation of a common point from two rays reduces to a problem of two-line intersection each radiating from the center of a camera. The ray line is generated by two points in each image as shown in Figure 7. One point on the line is defined by the camera center and the second point by the centroidal pixel of the pointer end in the image plane (i.e.,  $P_1$  or  $P_2$  in Figures 5, 6, and 7). Note that  $P_1$  and  $P_2$  are image matched points.

The point sets  $(C_1, P_1)$  and  $(C_2, P_2)$  are situated on the ray lines 1 and 2, respectively, in a frame of reference  $(x, y, z)$ . Since the points  $P_1(I, J)$  and  $P_2(I, J)$  are identified by the pixel coordinates, they need to be converted into the physical space by a transformation:

$$x \text{ distance per pixel} = \frac{f * \tan(\text{half view angle of camera})}{(\text{Image width in pixel}) / 2} \quad (2)$$

Similarly, y distance per pixel can be correlated. Note that  $f$  denotes camera focal length. Because we are interested in computing coordinates of the common point  $P$ , let us define each point on the line in  $x, y,$  and  $z$  reference frames as  $\mathbf{P} = x \mathbf{i} + y \mathbf{j} + z \mathbf{k}$ . That is

$$\mathbf{P}_1 = P_{x1} \mathbf{i} + P_{y1} \mathbf{j} + P_{z1} \mathbf{k}, \mathbf{P}_2 = P_{x2} \mathbf{i} + P_{y2} \mathbf{j} + P_{z2} \mathbf{k}, \mathbf{C}_1 = C_{x1} \mathbf{i} + C_{y1} \mathbf{j} + C_{z1} \mathbf{k}, \mathbf{C}_2 = C_{x2} \mathbf{i} + C_{y2} \mathbf{j} + C_{z2} \mathbf{k} \quad (3)$$

Where  $\mathbf{i}, \mathbf{j},$  and  $\mathbf{k}$  are unit vectors along  $x, y$  and  $z$  axes, respectively. With the condition that the four points must be coplanar (when the lines are not skewed), we can write

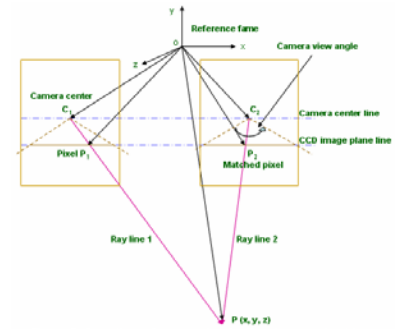
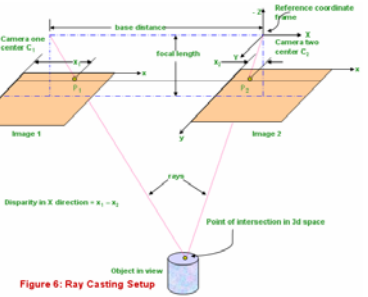
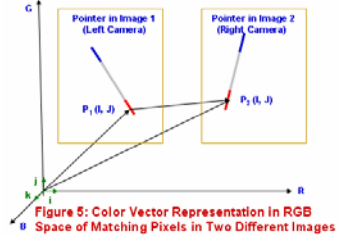
$$(\mathbf{C}_2 - \mathbf{C}_1) \cdot [(\mathbf{P}_1 - \mathbf{C}_1) \times (\mathbf{P}_2 - \mathbf{C}_2)] = 0 \quad (4)$$

where the symbols “ $\cdot$ ” and “ $\times$ ” represent vector dot and cross product respectively. If  $s$  and  $t$  are scalar quantities then the common point  $\mathbf{P}$  can be expressed parametrically as

$$\mathbf{P} = \mathbf{C}_1 + s(\mathbf{P}_1 - \mathbf{C}_1) = \mathbf{C}_1 + s \mathbf{A} = \mathbf{C}_2 + t(\mathbf{P}_2 - \mathbf{C}_2) = \mathbf{C}_2 + t \mathbf{B} \quad (5)$$

Simultaneous solution of equations (5) yields the value of  $s$  as

$$s = \frac{[(\mathbf{C}_2 - \mathbf{C}_1) \times \mathbf{B}] \cdot (\mathbf{A} \times \mathbf{B})}{|\mathbf{A} \times \mathbf{B}|^2} \quad (5b)$$



### 3.2 Accounting for Camera's Rotations

Six degrees-of-freedom are required to uniquely describe a point in three-dimensional space. One can choose three linear and three rotational coordinate axes. Determination of the pointer's position defined by three linear coordinates (x, y, z) is presented above, whereas orientation of the pointer specified by three rotations ( $\theta$ ,  $\phi$ ,  $\psi$ ) is given in this section. Thus the rotational motion of the camera is accounted for by the pointer's position and orientation analysis. Define camera's each axis of rotation as pitch, yaw and roll along x, y and z axes, respectively, as depicted in Figure 8. Hence, each axis transformation is given by

$$R(x, \text{pitch}) = R(x, \phi) = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & C\phi & -S\phi \\ 0 & S\phi & C\phi \end{Bmatrix} \quad (6)$$

$$R(y, \text{yaw}) = R(y, \psi) = \begin{Bmatrix} C\psi & 0 & S\psi \\ 0 & 1 & 0 \\ -S\psi & 0 & C\psi \end{Bmatrix} \quad (7)$$

$$R(z, \text{roll}) = R(z, \theta) = \begin{Bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{Bmatrix} \quad (8)$$

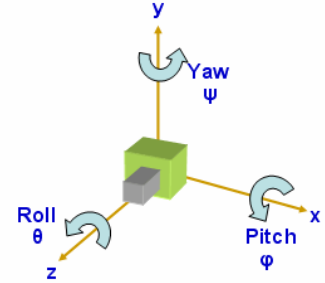


Figure 8: Camera's Pitch Yaw and Roll Axes

Where, the notations  $S(\text{angle}) = \sin(\text{angle})$  and  $C(\text{angle}) = \cos(\text{angle})$  are used. The combined transformation pitch-yaw-roll can be written as PYR

$$\begin{aligned} \text{PYR} &= R(x, \text{pitch}) R(y, \text{yaw}) R(z, \text{roll}) \\ &= R(x, \phi) R(y, \psi) R(z, \theta) \\ &= \begin{Bmatrix} C\theta C\psi & -S\theta C\psi & S\psi \\ C\theta S\phi S\psi + C\phi S\theta & C\theta C\phi - S\theta S\phi S\psi & -C\psi S\phi \\ S\theta S\phi - C\theta C\phi S\psi & S\theta C\phi S\psi + C\theta S\phi & C\phi C\psi \end{Bmatrix} \quad (9) \end{aligned}$$

The world coordinates (x, y, z) are, thus, related to camera's view coordinates ( $x'$ ,  $y'$ ,  $z'$ ) as

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} \text{PYR} \end{Bmatrix} \begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} \quad \begin{Bmatrix} x' \\ y' \\ z' \end{Bmatrix} = \begin{Bmatrix} -1 \\ \text{PYR} \end{Bmatrix} \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} \quad (10)$$

Note that inverse transformation is used to account for camera rotations.

### 3.3 Point of Projection on a View Screen

Knowing the three-dimensional coordinates of a common point corresponding to each end of the pointer (after triangulation of red and blue centroids), we can represent the pointer in a 3D space by a line passing through these two points. Figure 9 depicts the pointer connecting red and blue centroidal points  $P_r$  and  $P_b$  respectively. The projection of this line on a plane described by the view screen is of our interest. Thus, the problem is now simplified to finding coordinates of intersecting point between line and a plane as shown by point  $P_i$  in Figure 9.

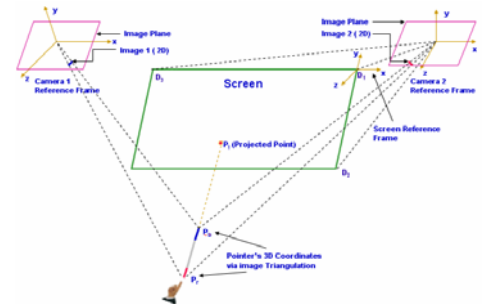


Figure 9: Three-dimensional Pointer Projection on the View Screen

### 3.4 Equation of a Plane Describing the Screen

The standard equation of a plane in a 3D space is:

$$Ax + By + Cz + D = 0 \quad (11)$$

Where, the normal to the plane is the vector (A,B,C). Let us define the plane representing view screen by three points  $D_1(x_1, y_1, z_1)$ ,  $D_2(x_2, y_2, z_2)$ , and  $D_3(x_3, y_3, z_3)$  in the camera 2 coordinate system (consistence with earlier calculations). The coefficients in Equation (11) are thus given by the following determinants.

$$A = \begin{vmatrix} 1 & y_1 & z_1 \\ 1 & y_2 & z_2 \\ 1 & y_3 & z_3 \end{vmatrix} \quad B = \begin{vmatrix} x_1 & 1 & z_1 \\ x_2 & 1 & z_2 \\ x_3 & 1 & z_3 \end{vmatrix} \quad C = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \quad D = - \begin{vmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{vmatrix} \quad (12)$$

Further simplification to the above equation is

$$A=y_1(z_2-z_3)+y_2(z_3-z_1)+y_3(z_1-z_2), B=z_1(x_2-x_3)+z_2(x_3-x_1)+z_3(x_1-x_2), C=x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2) \\ D = - [x_1 (y_2 z_3 - y_3 z_2) + x_2 (y_3 z_1 - y_1 z_3) + x_3 (y_1 z_2 - y_2 z_1)] \quad (13)$$

Note that if the points are colinear then the normal (A,B,C) will be (0,0,0). The sign of s (which equals Ax + By + Cz + D) determines which side the point (x,y,z) lies with respect to the plane: if s > 0 then the point lies on the same side as the normal (A,B,C); if s < 0 then it lies on the opposite side; if s = 0 then the point (x,y,z) lies on the plane.

### 3.5 Intersection of a Line and a Plane

The parametric representation of the equation of the line passing through points **Pr** (rx, ry, rz) and **Pb** (bx, by, bz) of the pointer is made as  $\mathbf{P} = \mathbf{Pr} + u (\mathbf{Pb} - \mathbf{Pr})$  (14)

The point of intersection of the line and plane can be found by solving the system of equations represented by Eqs. (11) and (14).

$$A [rx + u (bx - rx)] + B [ry + u (by - ry)] + C [rz + u (bz - rz)] + D = 0 \quad (15)$$

Hence the value of u is given by

$$u = \frac{A * rx + B * ry + C * rz + D}{A (rx - bx) + B(ry - by) + C(rz - bz)} \quad (16)$$

Substituting u into the equation of a line given by Eq (14) results in the point of intersection between line and plane as **P<sub>i</sub>** shown in Figure 9. This projected point is where the pointer is pointing towards the view screen. Remember, the denominator of u in Eq. (14) is 0; the normal to the plane is perpendicular to the line. Thus the line is either parallel to the plane and there are no solutions or the line is on the plane in which case there are infinite solutions.

## 4.0 Simulation

A virtual pointer of size 2.7 inches was modeled with red and blue color ends in a large presentation room setting using the popular three-dimensional computer graphics and animation program called Studio 3D Max by Discreet, a subsidiary of Autodesk Inc. The pointer was situated at a distance of around 30 feet away from the view/presentation screen of size 12 feet x 3.5 feet. While the pointer was pointing towards view screen, two snapshots with high resolution (1920 x 1200 pixels) were taken from two cameras located near upper corner of the screen. Figure 10 depicts left- and right- camera static images clipped to reduce image size for presentation purpose. Several configurations were simulated with different locations of the pointer in the room as well as various sizes of the pointer (small pointer: 2.7 in., medium pointer: 7 in. and large pointer: 21 in. in length).

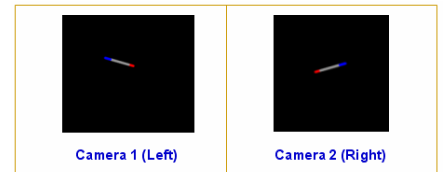


Figure 10: High Resolution Clipped Images (1920 x 1200 Pixels)

Based on the methodology presented above, a Visual C++ program was written to test the proposed analysis. The program takes images of two cameras as an input and computes the three-dimensional coordinates of the pointer (both position and orientation). In addition, the pointer's pointing projection on the view/presentation screen is outputted. These computed results were compared with the actual projections retrieved from Studio 3D Max.

## 5.0 Results

The output results of the C++ program executions are grouped into three categories: one, the pointer's pointing accuracy on the view screen without rotating any cameras; two, when camera rotations are included in the analysis; and three, when variation in the

pointer's sizes is included in the analysis. Table 1 presents five different test cases for the group one. The highlighted pink area describes changes in the configuration with respect to the case # 1. The output of the algorithm (the pointer's projection on the view screen) using triangulation method is compared to the corresponding retrieved values from the 3D Studio Max animation program. The worse case scenario is if it is off by 0.041 feet in y coordinate. The absolute average position accuracy for all five cases is 0.006 and 0.034 feet in the x and y coordinates, respectively. Note that the pointer is in neighborhood of 30 feet away from the screen. When the maximum absolute average accuracy (0.034 feet) is compared with 30 feet distance away from the screen, it is off by 0.11 % which is very small. Alternatively, compared to the size of the view screen (12 feet), it is off by around 0.28%. However, it should be emphasized that the pointer's projection accuracy on the view screen does not depend upon what size of the screen is chosen in the analysis. Rather, it merely gives relative judgment on the pointing direction.

| Test Case                                                                                   | Camera 1 Position |     | Camera 2 Position |       | Screen Position |      |          |     |          |     | Actual Screen Projection - Studio 3DMax<br>x y | Computed Screen Projection<br>x y | Difference in Position |        |        |       |        |
|---------------------------------------------------------------------------------------------|-------------------|-----|-------------------|-------|-----------------|------|----------|-----|----------|-----|------------------------------------------------|-----------------------------------|------------------------|--------|--------|-------|--------|
|                                                                                             | x y z             |     | x y z             |       | Point D1        |      | Point D2 |     | Point D3 |     |                                                |                                   | x                      | y      | x      | y     |        |
|                                                                                             | pitch             | yaw | roll              | pitch | yaw             | roll | x        | y   | z        | x   |                                                |                                   |                        |        |        |       | y      |
| 1                                                                                           | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | 1.7                    | -5.999 | 1.663  | 0.001 | 0.001  |
| 2                                                                                           | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | 1.7                    | -5.993 | 1.730  | 0.017 | -0.030 |
| 3                                                                                           | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | 1.7                    | -6.169 | 0.000  | 0.041 | 0.000  |
| 4                                                                                           | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | -1.7                   | -5.999 | -1.663 | 0.001 | -0.037 |
| 5                                                                                           | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | -3.0                   | -5.012 | -2.904 | 0.012 | -0.016 |
| All dimensions are in feet. Highlighted area describes the changes with respect to case # 1 |                   |     |                   |       |                 |      |          |     |          |     |                                                | Absolute Average                  |                        | 0.006  | 0.034  |       |        |

Table 1: Positioning Accuracy Comparison

Table 2 presents the results when camera rotations are included. Note that case # 9 accounts for all three-axis camera rotations. These specific angles are considered in order to maximize view coverage of the presentation room. The accuracy in this category is relatively poor due to errors in rotational calibration of the camera. This can be considerably improved upon choosing appropriate calibration techniques.

| Test Case                                                                                                                       | Camera 1 Position |     | Camera 2 Position |       | Screen Position |      |          |     |          |     | Actual Screen Projection - Studio 3DMax<br>x y | Computed Screen Projection<br>x y | Difference in Position |        |        |       |       |
|---------------------------------------------------------------------------------------------------------------------------------|-------------------|-----|-------------------|-------|-----------------|------|----------|-----|----------|-----|------------------------------------------------|-----------------------------------|------------------------|--------|--------|-------|-------|
|                                                                                                                                 | x y z             |     | x y z             |       | Point D1        |      | Point D2 |     | Point D3 |     |                                                |                                   | x                      | y      | x      | y     |       |
|                                                                                                                                 | pitch             | yaw | roll              | pitch | yaw             | roll | x        | y   | z        | x   |                                                |                                   |                        |        |        |       | y     |
| 6                                                                                                                               | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | 1.7                    | -6.677 | 1.484  | 0.677 | 0.216 |
| 7                                                                                                                               | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | 1.7                    | -5.777 | 1.484  | 0.223 | 0.216 |
| 8                                                                                                                               | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | 1.7                    | -6.002 | 1.756  | 0.002 | 0.069 |
| 9                                                                                                                               | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | -1.7                   | -4.747 | -1.718 | 0.747 | 0.018 |
| All Dimensions are in feet. Pitch, Yaw and Roll are in degrees. Highlighted area describes the changes with respect to case # 1 |                   |     |                   |       |                 |      |          |     |          |     |                                                | Absolute Average                  |                        | 0.412  | 0.126  |       |       |

Table 2: Rotational Accuracy Comparison

Table 3 considers the variation in the pointer's length. It is very encouraging to see that the smallest pointer of size 2.7 inches was detected with a high accuracy even with both cameras rotated. Also, the size of the pointer does not have much effect in the analysis.

| Test Case                                                                                                                                                                                                                   | Camera 1 Position |     | Camera 2 Position |       | Screen Position |      |          |     |          |     | Actual Screen Projection - Studio 3DMax<br>x y | Computed Screen Projection<br>x y | Difference in Position |        |        |       |       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|-----|-------------------|-------|-----------------|------|----------|-----|----------|-----|------------------------------------------------|-----------------------------------|------------------------|--------|--------|-------|-------|
|                                                                                                                                                                                                                             | x y z             |     | x y z             |       | Point D1        |      | Point D2 |     | Point D3 |     |                                                |                                   | x                      | y      | x      | y     |       |
|                                                                                                                                                                                                                             | pitch             | yaw | roll              | pitch | yaw             | roll | x        | y   | z        | x   |                                                |                                   |                        |        |        |       | y     |
| 10                                                                                                                                                                                                                          | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | -1.7                   | -6.001 | -1.8   | 0.001 | 0.100 |
| 11                                                                                                                                                                                                                          | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | -1.7                   | -6.024 | -1.642 | 0.024 | 0.058 |
| 12                                                                                                                                                                                                                          | -120              | 0.0 | 0.0               | 0.0   | 0.0             | 0.0  | 0.0      | 0.0 | -35      | 0.0 | 0.0                                            | -6.0                              | -1.7                   | -5.727 | -1.779 | 0.273 | 0.079 |
| All Dimensions are in feet. Pitch, Yaw and Roll are in degrees. Highlighted area describes the changes with respect to case # 1. Small pointer size = 2.7 in., Medium pointer size = 7 in., and Large pointer size = 21 in. |                   |     |                   |       |                 |      |          |     |          |     |                                                | Absolute Average                  |                        | 0.099  | 0.047  |       |       |

Table 3: Length Accuracy Comparison

## 6.0 Conclusion

The screen pointing interactive method using a magic wand for the application in a typical medical setting is developed. The computational analysis reveals that the image triangulation method works reasonably well for locating the pointer in a relatively large three-dimensional work space (a room). Furthermore, the pointer's projections on the view screens are accurate well within many applications. The computational errors are considered to be small when one views the screen from the audience located in the neighborhood of 30 feet away where precise visualization of pointer's direction is not that clear. Future investigation includes choosing actual hardware in the loop for hand pointing gesture recognition system.

## 7.0 References

- [1] <http://www.gcqhealth.com/index.htm>
- [2] R. Clodfelter and Y. Nir, "Multi-channel Display Systems for Data Interpretation and Command and Control," [http://barco.com/projection\\_systems/downloads/Multi\\_Channel\\_Display\\_Systems.pdf](http://barco.com/projection_systems/downloads/Multi_Channel_Display_Systems.pdf)
- [3] Chen, Sukthankar, Wallace and Li, "Scalable Alignment of Large-Format Multi-projector Displays using Camera Homography Trees," IEEE Visualization Conference, Oct 27-Nov 1. 2002. 0-7803-7498.
- [4] A. Bezerianos and R. Balakrishnan, "View and Space Management on Large Displays," IEEE Computer Graphics and Applications, July/August 2005, pp. 34-43.
- [5] Robertson, & et al., "The Large-Display User Experience," IEEE Computer Graphics and Applications, July 2005, pp. 44-51.
- [6] Jedrysik, Moore, Stedman & Sweed, "Interactive Displays for Command and Control," Proc of IEEE 2000 Aero Conf, p. 341-351.
- [7] <http://www.afrlhorizons.com/Briefs/Sept01/IF0012.html>, US Air Force Research Laboratory, Rome, New York.
- [8] G. Firzmaurice & et al., "Cinematic Meeting Facilities Using Large Displays," IEEE Comp. Graphics and Appl., 2005, pp. 17-21.
- [9] R.J.K. Jacpn, "Human-computer interaction: Input devices," ACM Computer Survey, Vol. 28, No.1, March 1996, pp. 177-179.
- [10] A. Blake and M. Isard, "Active Contours: The Application of techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion," Springer-Verlag, 1998.
- [11] B. Carlson, "The consumer desktop looks at vision: R&D directions at Microsoft and MIT," Advanced Imaging, 1998, pp. 26-28.
- [12] A.P. Pentland, "Smart rooms," Scientific America, Vol. 274, No. 4, 1996, pp.68-76.
- [13] D. Calbert, "The Magic Worlds of Harry Potter: A Treasury of Myths, Legends, and Fascinating Facts," Lumina Press, 2001.