Int J Comput Vis
DOI 10.1007/s11263-012-0533-8

Virtual Volumetric Graphics on Commodity Displays Using 3D Viewer Tracking

Charles Malleson · John Collomosse

Received: 5 December 2011 / Accepted: 27 April 2012

Abstract Three dimensional (3D) displays typically rely on stereo disparity, requiring specialized hardware to be worn or embedded in the display. We present a novel 3D graphics display system for volumetric scene visualization using only standard 2D display hardware and a pair of calibrated web cameras. Our computer vision-based system requires no worn or other special hardware. Rather than producing the depth illusion through disparity, we deliver a full volumetric 3D visualization—enabling users to interactively explore 3D scenes by varying their viewing position and angle according to the tracked 3D position of their face and eyes. We incorporate a novel wand-based calibration that allows the cameras to be placed at arbitrary positions and orientations relative to the display. The resulting system operates at real-time speeds (∼25 fps) with low latency (120–225 ms) delivering a compelling natural user interface and immersive experience for 3D viewing. In addition to objective evaluation of display stability and responsiveness, we report on user trials comparing users' timings on a spatial orientation task.

Keywords Volumetric display · 3D graphics · Tracking · Kalman filter · Camera Calibration

Electronic supplementary material The online version of this article (doi:10.1007/s11263-012-0533-8) contains supplementary material, which is available to authorized users.

C. Malleson · J. Collomosse (✉)
Centre for Vision Speech and Signal Processing,
University of Surrey, Guildford, Surrey, UK
e-mail: J.Collomosse@surrey.ac.uk

C. Malleson
e-mail: C.Malleson@surrey.ac.uk

1 Introduction

Three dimensional (3D) displays have emerged as the next generation of viewing technology, exploiting visual disparity cues to create the illusion of depth. Current displays facilitate this illusion through a mechanism to independently control the left and right eye viewpoints. The mechanism is commonly embedded in glasses (e.g. colored anaglyph filters, Sorensen et al. 2004; polarized lenses or shutter glasses, Woods 2009) or through lenticular or parallax barriers built into the screen itself (e.g. auto-stereoscopic displays, Woodgate et al. 2000). Such mechanisms increase the cost of 3D through specialist glasses or display hardware.

In this paper we describe how commodity hardware (a standard flat-screen monitor, with a pair of web-cams mounted on top) can be used to create the illusion of 3D without the expense of specialist hardware. In addition, we move beyond simple depth perception and disparity effects to create a *volumetric* or free-viewpoint display; enabling the user to interactively vary their point of view relative to the scene. This interaction model enables users to "look around corners", to reveal aspects of the scene previously hidden, considering issues such as occlusion and apparent object size (Fig. 1). None of these volumetric attributes are considered in conventional 3D displays, and to the best of our knowledge none have been synthesized on standard 2D hardware in a non-invasive (glasses-free) format.

Our non-invasive volumetric display runs robustly at real-time speeds (25 fps) on an Intel Core i7 1.6 GHz laptop with 18.4 inch flat-panel 2D display using two Microsoft standard definition web-cams.[1] The viewer's 3D position is triangulated and tracked using a Kalman filter supplied with

[1] A video demo of the system is available in the Supplementary Material.

Published online: 17 May 2012

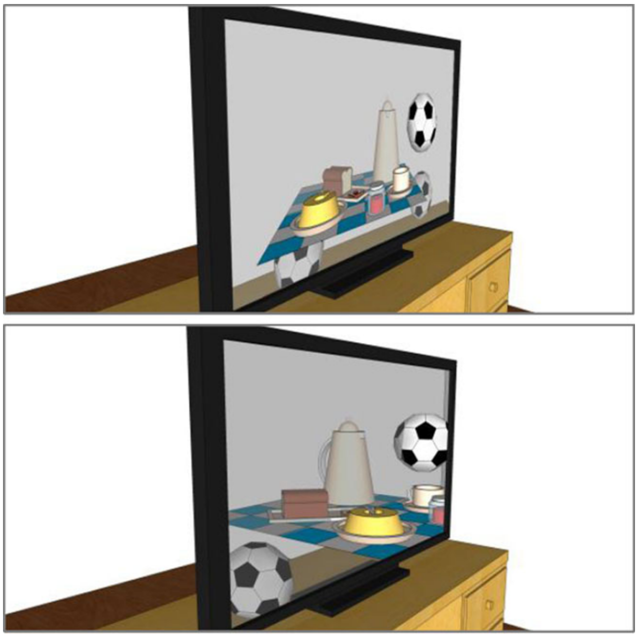


Fig. 1 *Above*: Standard display; distortion from oblique viewing and no viewpoint adaption. *Below*: The proposed display system; no distortion and passive viewpoint selection through user gaze tracking conveys the illusion of 3D content

face and eye position data from video frames captured on the stereo camera pair. In addition to standard methods for calibrating the stereo pair, we introduce a novel wand-based calibration process to learn the position of the display face relative to the cameras, to enable accurate positioning of the simulated views. Accurate (sub-centimeter) determination of this spatial relationship is critical to the synthesis of a believable volumetric illusion.

Our display has potential applications in virtual reality (VR) including the visualization and exploration of scanned 3D models and volumetric 3D viewing of free-viewpoint 3D video. Because the system provides an accurately projected view of a 3D scene for any viewer position, it effectively converts a standard planar display into a virtual volumetric display. This could be used as a form of low-cost virtual-reality (VR) if a larger LCD panel or a projector screen were used as the display. Standard augmented-reality (AR) systems overlay graphics onto a video feed of a real-world scene and display the composite scene to a screen. Our adaptive viewer-tracking display makes an alternative configuration possible: real-world objects could be placed between the screen and the viewer. In this configuration, virtual graphics could be approximately spatially-registered with the real-word objects placed in front of the screen. We provide such an example in Fig. 10.

We describe our display system in Sect. 3 and report experimental data in Sect. 4 focusing on quantitative evaluation of display stability, accuracy and timing. We also report a set of user trials to objectively measure the usability of the display in a user task requiring exploration of 3D spatial volume, comparing this against a mouse-based interface.

1.1 Related Work

There has been considerable investment into specialized hardware for stereoscopic and autosteroscopic displays, following the trend toward 3D content generation. Urey and Erden (2011) categorize these approaches into head-mounted displays, and 'direct-view' display panels with and without the requirement for eye-wear. The latter category of direct-view display is of relevance to us, in particular reactive displays using computer vision to track viewer location.

The most common form of eyewear-free 3D display incorporates a parallax barrier, typically of LCD construction, either in front of (Sandin et al. 2001; Perlin et al. 2001) or embedded within the display surface (Ezra et al. 1995; Tsai et al. 2009). The barrier selectively occludes pixels enabling certain pixels to be viewed selectively from certain angles, so projecting two disparate (half intensity) images in slightly different directions from a single pixel raster. A variety of barrier patterns have been experimented with, from slanted barriers (Chen et al. 2009) to aperture grills (Yamamoto et al. 2002; Nishimura et al. 2007) which can enhance intensity and spatial resolution. If the viewer is positioned so that the two projections are directed independently to each eye—the "sweet spot" of the display—a convincing depth perception illusion is created. In the first 3D displays, manufactured by Sharp R&D, the user-self calibrated their position to acquire the sweet spot via a 'viewer position indicator' (VPI) pattern (Woodgate et al. 2000) at the base of the display. More recently, viewer position is actively tracked (typically via the head) to direct the sweet spot toward the viewer by reconfiguring the LCD barrier in real-time (Woodgate et al. 1997).

Early head-tracking 3D displays include the electro-mechanical systems of Schwartz (1985), where head position was tracked using projected infra-red (IR) light. Differences in reflected IR light were computed between imaging cycles yielding a rudimentary form of optical flow estimate in the horizontal axis. A similar electro-mechanical system is described by Tetsutani et al. (1989). Within the mid-to-late nineties experiments in autostereoscopic displays typically featured head-tracking through IR retroreflective markers or ultra-sonic positioning devices (Sandin et al. 2001; Tetsutani et al. 1994). However the desire to avoid worn hardware has motivated software solutions to viewer tracking, primarily through face detection (Surman et al. 2008b). A number of tracking displays such as those of the recent EU MUTED project (Brar et al. 2010), draw upon the Viola and Jones face detection algorithm (Viola and Jones 2001). The face is detected via a decision cascade of Haar wavelet basis functions that can rapidly reject poor candidate regions, and so is suitable for real-time tracking on resource constrained embedded hardware. The cascade is trained using a boosting approach on a large and diverse face detection set (Freund and Schapire 1999). Several other autostereoscopic systems incorporate this tracker either for single (Free2C 2010; Erden et al. 2009) or multiple participants (Takaki 2006; Surman et al. 2008a, 2008b); the latter requiring laser-based hardware to accurately localize projection of the imagery. Our system also adopts (Viola and Jones 2001), as a precursor to more accurate localization of the eye regions.

All the systems reviewed so far require specialized display hardware to create the 3D illusion. The novelty of our system is in the combination of head and eye tracking with a regular 2D flat-screen display. We first described this system in Malleson and Collomosse (2011) and here describe a modified tracking process in greater detail, also introducing a comparison with the mouse through a user evaluation based on 3D spatial exploration. Prior to Malleson and Collomosse (2011), the most closely related to our work is the IR tracking system of Lee, in which a commodity games controller (Wiimote) is used to track the position of the viewer (Lee 2008). As with our work, a form of skewed frustum is used to synthesis camera viewpoint from the determined 3D world-coordinates of the viewer. In Lee (2008) viewer position is determined using two IR point sources mounted on the user's head. Our system differs in that we eschew wearable hardware for passive camera-based observation; necessitating robust tracking and accurate camera calibration using computer vision techniques.

2 System Overview

The system comprises two Microsoft Live HD-6000 USB web-cams running at 25 fps with a resolution of 640 × 360 i.e. in wide-screen mode, mounted on a baseline of approximately 6.5 centimeters separation, and fixed to a 18.4 inch wide-screen laptop display as illustrated in Fig. 2(b). We adopt the OpenGL API to create the 3D scene, with viewpoint selected and specified by a viewing frustum. The system runs in a closed loop (Fig. 2(a)), at each frame estimating the focal point and geometry of the viewing frustum using the triangulated midpoint of the viewer's eyes (Sect. 3.1). Prior to computing the frustum, the 3D viewer position is passed through a Kalman filter to reduce viewpoint jitter. The integration of OpenGL coordinates and viewer (world) coordinates is facilitated by a calibration preprocess that both estimates both the inter-camera relative positions, and the position of the cameras relative to the display surface.

3 Volumetric Display Through Eye Position Tracking

The web cameras are calibrated in a manual pre-processing step, yielding: intrinsic estimates for their focal length $\{f_x, f_y\}$, optical center $(\mathbf{c}_x, \mathbf{c}_y)$ and radial distortion k; and extrinsic estimates for the second camera relative to the first in the form of baseline offset $\mathbf{T} = [\mathbf{t}_x, \mathbf{t}_y, \mathbf{t}_z]^{\mathrm{T}}$ and orientation (3 × 3 matrix $\mathbf{R}$).

3.1 3D Viewer Position Estimation

As with prior head-tracking 3D systems, we initially employ a cascaded detector to determine location of candidate facial regions (Viola and Jones 2001). Due to rendering overhead and the resolution of the dual video feeds, a full pass of the detector was impractical even using an optimized implementation (OpenCV 2011). The detector was therefore modified to return the first face found (which is the largest face). Further speed optimization was achieved by scaling the input image according to the size of the last detected face (making detection time independent of viewer distance), and by only searching a region around the last detected face (see Fig. 3). Unfortunately the centroid of this region can exhibit up to 5–10 pixels (∼1–2 % variation generating unacceptable scintillation in the z-depth during later triangulation. We therefore use the initial face detection estimate as a local

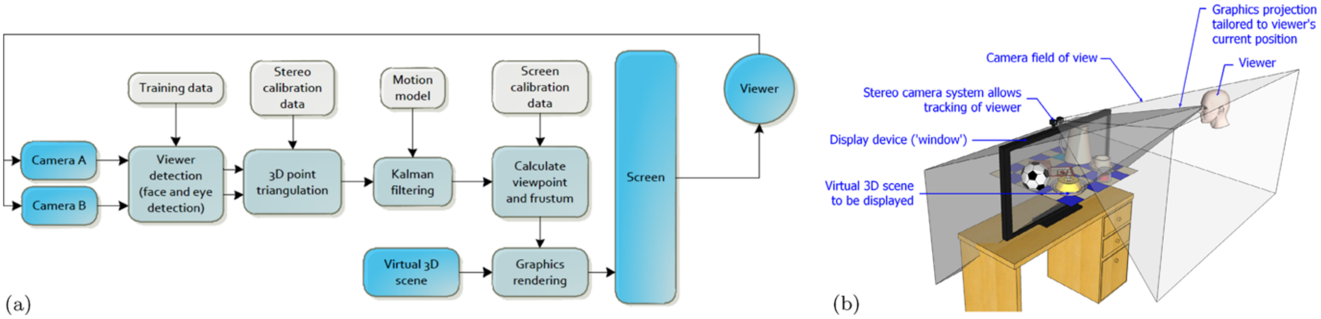

Training data
Stereo calibration data
Motion model
Screen calibration data
Viewer
Camera A
Camera B
Viewer detection (face and eye detection)
3D point triangulation
Kalman filtering
Calculate viewpoint and frustum
Screen
Virtual 3D scene
Graphics rendering
(a)
Graphics projection tailored to viewer's current position
Camera field of view
Viewer
Stereo camera system allows tracking of viewer
Display device ('window')
Virtual 3D scene to be displayed
(b)

Fig. 2 Overview of the display system. (**a**) The control flow of for rendering a single frame of 3D content. (**b**) The geometry of our display setup. The viewing frustum (*grey*) is derived from the tracked eye position and the corners of the display surface

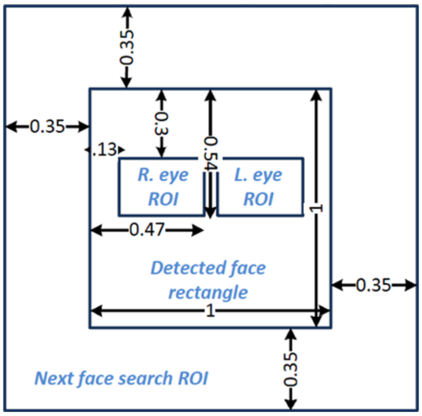


Fig. 3 Eye search ROI positions and sizes relative to the current detected face size and position. By specifying normalized coordinates within the ROI the system is independent to scale (face distance from camera) and video resolution

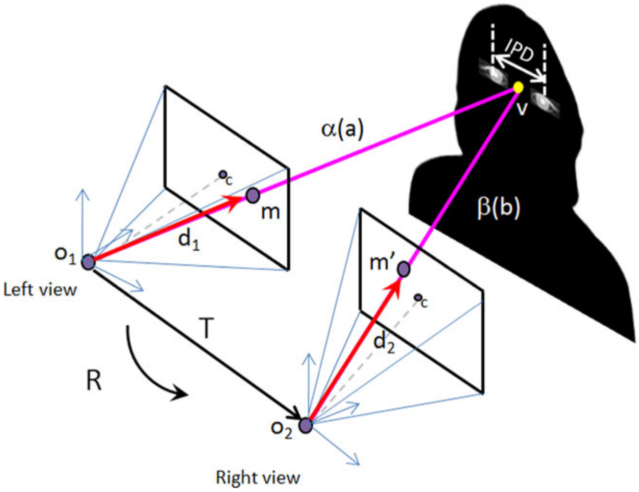


Fig. 4 Triangulation of the eye position **V** using the calibrated camera setup

window within which to search for the viewers' eyes. As an additional performance improvement, we search only within a coarse region of interest local to the previously detected location of the face (if available).

Using a template image $T(u,v)$ for a single eye cut from an initial training image, we search each of the full resolution images $I(x,y)$ within the bounds identified by the face detector to position $\mathbf{p}=(x,y)$ minimizing score $\gamma(\mathbf{p})$:

$$\gamma(x,y) = \frac{\sum_{u,v}[I(x,y)-\hat{I}][T(x-u,y-v)-\hat{T}]}{(\sum_{u,v}[I(u,v)-\hat{I}]^2\sum_{u,v}[T(x-u,y-v)-\hat{T}]^2)^{\frac{1}{2}}} \tag{1}$$

where $\hat{T}$ and $\hat{I}$ are average intensities of the template and image respectively. $T(x,y)$ is scaled in proportion to the area of the detected face region prior to matching. These templates are matched over the region of interest (ROI) defined by a subregion of the last detected face. These ROIs are a fixed sub-region of the detected face rectangle (see Fig. 3). The size and position of these rectangles (relative to the detected face rectangle) were experimentally chosen so that they are large enough for the eyes to fall consistently within the regions yet are as small as possible for greatest efficiency and also to make false locks (e.g. onto eye-brows) as low as possible. The process is repeated for each eye; minimizing $\gamma(.)$ to yield 2D coordinates for the first and the second eye ($\mathbf{e}_1$ and $\mathbf{e}_2$ respectively).

The image area occupied by the eyes varies with viewer distance. Therefore, the template images are scaled prior to matching, in proportion to the area of the detected face rectangle. This results in a more stable localization than the initial face detection. For improved precision, the template matching operates on full-resolution camera frames, rather than the down-sampled video used to perform the initial face detection.

Our system triangulates the viewer's position using the 2D midpoint of the two eyes $\mathbf{m}=\mathbf{e}_1+\frac{d_{ipd}}{2}$, where $d_{ipd}=\mathbf{e}_2-\mathbf{e}_1$ is the *inter-pupillary distance* (IPD); see Sect. 3.2.4. We denote the 2D position of the midpoint as $\mathbf{m}$ and $\mathbf{m}'$ within the two camera views respectively.

For each view $i=\{1,2\}$ we extrude a parametric ray $\mathbf{r}_i(s)$ from the center of projection $\mathbf{o}_i$ through the eye centroids on the respective camera image planes. Within our calibrated coordinate system o_1 is at the origin and $o_2=-\mathbf{R}^{-1}\mathbf{T}$. We denote the pair of rays extracted for $\mathbf{m}$ and $\mathbf{m}'$ as $\alpha(a)=\mathbf{o}_1+a\mathbf{d}_1$ and $\beta(b)=\mathbf{o}_2+b\mathbf{R}^{-1}\mathbf{d}_2$ respectively, where ray direction $\mathbf{d}_i$ is defined via the camera intrinsics and coordinates of the respective midpoint image (m_x,m_y) as follows:

$$\mathbf{d}_i = \begin{bmatrix} s(m_x-c_x)/f_x \\ s(m_y-c_y)/f_y \\ 1 \end{bmatrix} \tag{2}$$

$$s = \begin{cases} (\sqrt{1+4ek}-1)/ek & \text{if } e>1; \\ 1 & \text{else} \end{cases} \tag{3}$$

$$e = \sqrt{(m_x-c_x)^2+(m_y-c_y)^2} \tag{4}$$

The distance between the two rays $d_{(\alpha,\beta)}=|\alpha(a)-\beta(b)|$ is computed by solving for a and b:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1\cdot\mathbf{d}_1 & -\mathbf{d}_1\cdot\mathbf{d}_2 \\ \mathbf{d}_1\cdot\mathbf{d}_2 & -\mathbf{d}_2\cdot\mathbf{d}_2 \end{bmatrix}^{-1} \begin{bmatrix} s\mathbf{R}_1^{-1}\mathbf{d}_1\cdot(\mathbf{o}_2-\mathbf{o}_1) \\ s\mathbf{R}_1^{-1}\mathbf{d}_2\cdot(\mathbf{o}_2-\mathbf{o}_2) \end{bmatrix} \tag{5}$$

The resulting distance is used to obtain the 3D position of the viewer (Fig. 4) $\mathbf{V}=\alpha(a)+\frac{\beta(b)-\alpha(a)}{2}$.

3.1.1 Trajectory Smoothing

Successive estimates for 3D viewer position $\mathbf{V} = [x\ y\ z]^{\mathrm{T}}$ are combined via Kalman filter under a second-order model where, at time t, the instantaneous state is represented as $\mathbf{x}_t = [\mathbf{V}\ \dot{\mathbf{V}}\ \ddot{\mathbf{V}}]^{\mathrm{T}}$ of which only the first vector is observable. We model noise in this observation as additive Gaussian distribution:

$$\mathbf{V}_t = \mathbf{H}\mathbf{x}_t + \mathcal{N}(0, \mathbf{R}) \tag{6}$$

where constant 3×9 matrix $\mathbf{H}$ provides a measurement from $\mathbf{x}_t$:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & \dots \\ 0 & 1 & 0 & \dots & 0 & \dots \\ 0 & 0 & 1 & \dots & 0 & \dots \end{bmatrix}, \tag{7}$$

and the measurement noise (covariance) for each new observation is a diagonal matrix comprising estimates for variance in x, y, and z:

$$\mathbf{R} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}. \tag{8}$$

The co-variances are updated at each frame to reflect confidences in our measurement of $\mathbf{V}_t$ using a product of three heuristic measures encoding: (i) the stereo disparity between observed positions of the viewer; (ii) a function of the inter-pupillary distance (IPD) d_{ipd} indicating the deviation of the measured IPD from the true IPD measured *a priori*; and (iii) from the confidence score $\gamma(.)$ used to determine the viewer position in (1). We outline these measures in Sect. 3.2. Finally, the state transition matrix $\mathbf{F}$ of the Kalman filter is:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.99 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.99 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.99 \end{bmatrix} \tag{9}$$

where Δt is the time elapsed between measurements. Because of the nature of the processes involved in one cycle of operation, the measurement sample-time Δt (i.e. the inter-frame period) is not constant. Therefore at every cycle $\mathbf{F}$ is updated with a newly calculated Δt (measured via CPU clock ticks). The factors of 0.99 were introduced to stall unstable 'run-away' acceleration which can otherwise occur during operation.

3.2 Measurement Confidences

There are several indicators gathered during the 3D triangulation of the eye position that could usefully contribute to a measure of confidence in eye position i.e. to drive $\mathbf{R}$ in the Kalman Filter. These include: eye template matching scores (one from each camera), stereo-disparity, and the estimated distance of the user from the camera. Although σ_x^2 and σ_y^2 measurements may be expected to vary similarly, the nature of triangulation is such that the σ_z^2 should be inherently higher. When considering both eyes, bounds on the *inter-pupillary distance* (IPD) can also be used to measure of confidence in the correct localization of the eyes.

3.2.1 Template-Match Confidence

The normalized correlation coefficient method (1) used to localize the eye produces scores in the range $[-1, 1]$ with 1 being an idealized match. One may use the score of this match as an indicator of the quality of the match.

Let γ_{ec} be the template match score for eye e on camera c and let p^t_{ec} represent its heuristic likelihood ($p^t_{ec} \in [0, 1]$). The score t_{ec} needs to map to p^t_{ec} such that $-1 \to 0$ and $1 \to 1$. Between these extremes a soft threshold is required. This is achieved using a hyperbolic-tangent (sigmoid) function the parameters of which were selected empirically:

$$p^t_{ec} = \frac{1 + \tanh[7(t_{ec} - 0.45)]}{2}. \tag{10}$$

This mapping is plotted in Fig. 5(a). This is calculated for both of the eyes (l and r) on each of the cameras (a and b) producing four likelihoods p^t_{la}, p^t_{lb}, p^t_{ra} and p^t_{rb}.

3.2.2 Stereo Disparity

Before the point pairs from each camera are triangulated they are first undistorted and row-aligned in a standard stereo rectification process. Ideally, a given point in 3D space should project to the same y-coordinate from both cameras (i.e. with no vertical disparity). In practice, the noise in the detection leads to some y-disparity. The larger this disparity is, the less confident one can be in the measurement.

The (absolute) y-disparity lies in the range $[0, \infty)$ with 0 being a perfect match. Let d_e be the y-disparity for eye e (in pixels) and let p^d_e represent its heuristic likelihood. For this mapping: $0 \to 1$ and $\infty \to 0$. Again, a hyperbolic-tangent function is used to get a soft threshold between these extremes:

$$p^d_e = \frac{1 + \tanh[-0.4(|(d_e| - 7)]}{2}. \tag{11}$$

This mapping is plotted in Fig. 5(b). This is calculated for each eye yielding another two likelihoods p^d_l and p^d_r.

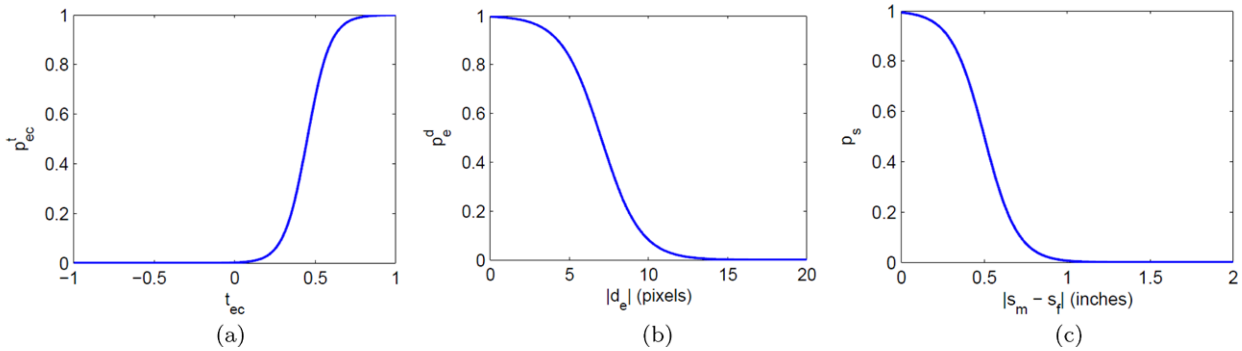


Fig. 5 Mappings from raw values to heuristic likelihoods used to get viewer position measurement variances: (**a**) template match score (**b**) y-disparity and (**c**) inter-pupillary distance error

3.2.3 Viewer Distance

As viewing distance (depth) increases, the size of the eye regions decreases and the search region and scaled template occupy fewer pixels. The template matching is only precise up to the nearest pixel. To illustrate the dependence of accuracy on viewer depth consider a template match that is off by one pixel in the x-direction. This would introduce an error in reconstructed horizontal position proportional to the viewer depth. The standard deviation of the measurement in the x and y directions is therefore proportional to the viewer depth.

However care must be taken using the viewer depth in the confidence calculation. One cannot use an unfiltered measurement of viewer depth since in outlier cases where the distance is significantly under-estimated (e.g. close to zero), the confidence in the measurement would be over-estimated. Using the filtered viewer distance directly is also problematic; were the filtered distance to start growing, the confidence in new measurements would continually decrease, allowing the growth in filtered depth to continue, as new measurements (deemed to be less and less reliable) would increasingly be ignored, leading to instability. To robustly include the filtered depth in the confidence calculation a range check is done on the filtered depth z_f before using it. A face cannot be detected when it is too close to the cameras to fit in the frame (less than about 6 inches). It is also assumed that—in normal operation—the viewer will lie within about 50 inches of the cameras. Therefore a face position outside this range is regarded as invalid for the purposes of the confidence calculation and a nominal value of 20 inches is used in the depth confidence factor instead.

3.2.4 Inter-pupillary Distance

The Kalman filter is applied only to the triangulated midpoint of the viewer's eyes. Despite this, there is still further information to be gleaned from the individual measured eye positions. The viewer's true IPD does not change over time. This fact can be used as yet another heuristic. The measured IPD is simply the Euclidean distance between the 3D measured eye positions.

According to Dodgson (2004), the IPD of adults varies between 45 and 80 mm. This is far too wide a range for a single value to be assumed for the true IPD for all users of the system. Instead, the system should learn the IPD of the current user. As an estimate of the true IPD of the current user, the system uses a heavily low-pass filtered version of the measured IPD. Let s_f denote the low-pass filtered IPD, s_m the measured IPD (i.e. $|\alpha(a) - \beta(b)|$, of (5)) and p_s its likelihood for which a sigmoid mapping is again used (Fig. 5(c)):

$$p_s = \frac{1 + \tanh[-5(|s_m - s_f| - 0.5)]}{2}. \tag{12}$$

3.3 Combining the Heuristics

All of these confidence measures may be combined to yield an estimate of the measurement covariances σ_x^2, σ_y^2 and σ_z^2. Variances σ_x^2 and σ_y^2 are determined as the z value divided by the product of the three heuristic likelihoods outlined in (10)–(12). The z variance (σ_z) is set a constant factor larger than σ_x^2 and σ_y^2; in our experiments we used a factor of 40. In addition, we perform a check on the sign of the detected viewer position; if negative (i.e. behind the cameras) this is deemed to be a gross triangulation error and the variances are set to infinity, so recording no observation.

These choices reflect a rise in uncertainty as disparity increases and observation confidence of the viewer decreases. In addition the penalty on large inter-pupillary distance prevents sporadic mis-identifications of one or both eyes from drawing the Kalman filter away from the true location of the midpoint. The use of the Kalman filter is critical to the stability of viewpoint, which can be adversely affected by poor template matches in (1) and sporadic failures in face detection.

We note the following concerning the robustness of the proposed viewer tracking approach. The speed optimizations of the viewer detection routine have useful side-effects in increasing robustness against possible failure modes. Consider a third-party 'spectator' comes into view of the cameras during operation. Using the face-size-normalized ROI (Fig. 3) around the previous face limits the face detector to search for the (largest) face in the area where the viewer is likely to have moved within the last frame period (this ROI was empirically determined by having a viewer make sudden movements while viewing a scene). This means that the face detector cannot lose lock on the viewer and lock onto any spectators who may be visible in the background unless they fall within the ROI. However the compactness of the ROI means that it is unlikely for a spectator's face to be completely visible at a scale comparable to that of the viewer. Therefore in our experiments additional faces provided no distraction, provided the tracker was initially locked onto the desired viewer.

There are situations in which the Viola Jones face detector can fail. This occurs when the camera's view of the face differs too much from head on (e.g. rotated more than about 10°), when too much of the face is occluded, when there are strong shadows (e.g. from sunlight), or when fast head motion causes strong motion blur. At tracker initialization (startup or loss of lock), the face search ROI defaults to the full frame and the scale to full resolution . If a face detection on a frame fails the system does not immediately revert to full frame and resolution, as this has a large performance penalty. Rather if a face is not found for more than 5 consecutive frames, it is assumed that lock has been lost and the tracker reinitializes. This prevents undesirable reinitializations when for example a hand briefly occludes the face.

The eye detectors are also effected by poor lighting, particularly if the templates were captured under conditions different from the operating conditions. This did not pose a problem in the indoor settings where we performed the experiments (and where anticipated use would take place) but degrades performance in settings where strong and varying shadows are present. The worst case error of the eye detectors is limited by the template matching ROIs, thus erroneous matches that can occur when, for instance the viewer blinks have limited effect. Poor tracking performance results can occur with viewers who wear glasses if the reflection in the lenses predominates the image.

3.4 Dynamic Adjustment of Viewpoint

The scene viewpoint is rendered through specification of a virtual camera (viewing frustum) such that the viewing is looking in the direction of the negative z-axis). This can be achieved by setting the OpenGL projection matrix with an asymmetrical frustum (Fig. 6), specified using the z-coordinates of the near clipping plane—n and the far clipping plane—f and the four edges of the rectangle defining the front of the frustum (at the near clipping plane) i.e. the x-coordinates of the left (l) and right (r) and y-coordinates of the top (t) and bottom (b). The near clipping plane can be set arbitrarily close to the viewer (for this application it was chosen to be 15 cm) and the far clipping plane is scene specific, set in our experiments to 5 m.

The other four parameters l, r, t and b can then be calculated in terms of n; x_c, y_c and z_c (the positions of the viewer with respect to the center of the display in each direction); w and h (the width and height of the display respectively). Values x_c, y_c and z_c are derived from $\mathbf{V}$ by performing a rotation and translation corresponding to the offset of the display center from the center of the camera baseline, as described in Sect. 3.5.

Figure 6 shows the geometry of the frustum viewed from the top (in the $y = 0$ plane) from which the parameters l and r can be derived in terms of n and the viewer position. Note that the coordinate system has been shifted such that the viewer is now at the origin as is required. By similar triangles $\triangle ADE \sim \triangle ABC$ and $\triangle AFE \sim \triangle AGC$ it follows that

$$l = -n(x_c + w/2)/z_c \tag{13}$$

$$r = -n(x_c - w/2)/z_c \tag{14}$$

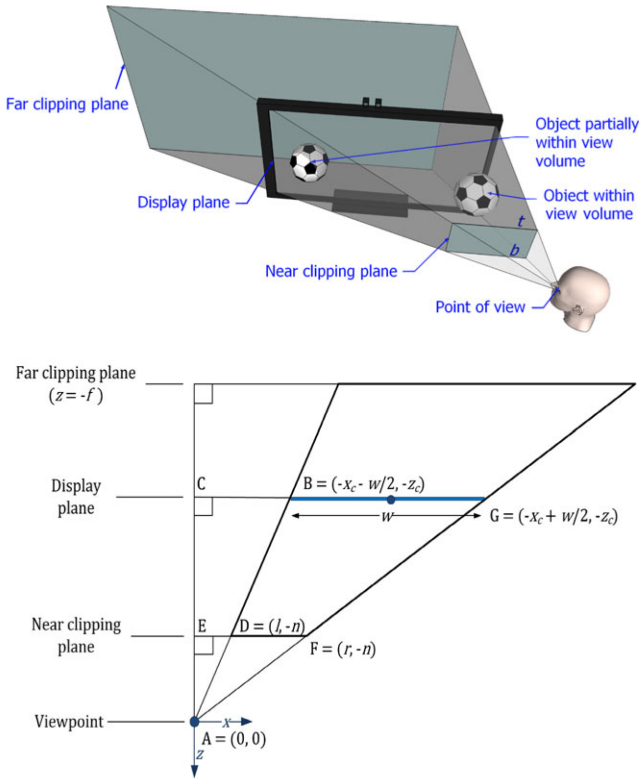


Fig. 6 The asymmetric viewing frustum and its derivation

By analogy with the above, a diagram and geometric arguments can be produced for the vertical geometry:

$$b = -n(y_c + h/2)/z_c \qquad (15)$$

$$t = -n(y_c - h/2)/z_c \qquad (16)$$

3.5 Display Calibration

In addition to the stereo camera calibration pre-process, a subsequent calibration step must be performed to determine the world coordinates of the display corners. This enables viewer position estimate **V** to be transformed to coordinates (x_c, y_c, z_c) when setting up the frustum (Sect. 3.4).

The display calibration is performed with assistance of a colored wand prop introduced into the scene. The wand comprises a hollow tube containing a laser pointer, capped at each end with a distinctively colored spherical marker (Fig. 7). The laser in the wand is shone at the each corner of the screen from several different positions (Fig 8(a)); and an image is captured from both web-cam's for each wand position.

Wand markers are identified via their distinctive color and shape. An Eigenmodel is trained from RGB color pixel samples, collected *a priori* from images of the markers. The learned color model is used to extract a binary mask of the marker region by thresholding the Mahalanobis distance of pixels from this model. A Canny edge detector is run over the hue and saturation channels of the image to create two edge masks, which are combined via binary OR, and intersected with the binary mask obtained from the color thresholding step. The resulting mask is passed through a circular Hough Transform (HT). The centroid of the highest scoring HT candidate is used as the 2D marker position (Figs. 7(b)–(e)).

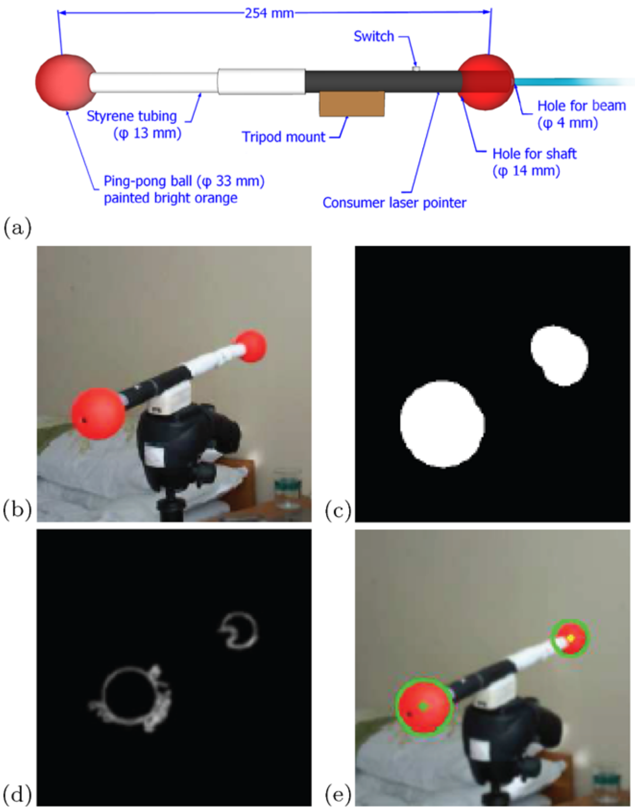


Fig. 7 Construction of the laser-pointer based calibration wand (**a**). The localization of each wand end (**e**) from the video feed (**b**) using color thresholding (**c**) and shape detection (**d**)

After the markers are localized in 2D, the 3D positions of the spherical markers are recovered via the triangulation approach of Sect. 3.1 yielding the equation for a ray passing between the two marker positions and the corner of the screen. All combinations of rays for a given corner are exhaustively intersected via a further triangulation, yielding putative 3D positions for the corner of the screen (see the blue dots in Fig 8(b)). The process is repeated for every screen corner, with six wand poses per corner (i.e. fifteen intersection points) representing an acceptable trade-off between accuracy and required user effort. Having gathered point distributions for each corner, the final location estimate for each of the display corners is inferred by the following process.

First, the median of each corner's point distribution is selected as an initial approximation to the four screen corners; points in the distribution beyond a threshold distance from the median are discarded leaving only inlier points (indicated in green, Fig. 9). A best-fit rectangle is then fitted by taking an eigendecomposition $\mathbf{C} = \mathbf{U}\Lambda\mathbf{U}^{\mathrm{T}}$ of the 3×3 scatter matrix **C** produced by these four data points. The eigenvector columns within **U** corresponding to the first and second largest eigenvalues are deemed to describe the horizontal and vertical sides of the display in world coordinates. The definition of the display plane location is completed by computing the mean location μ of the initial screen corner estimates, which is deemed to lie upon the plane. The average distances between the pairs of initial corner points along

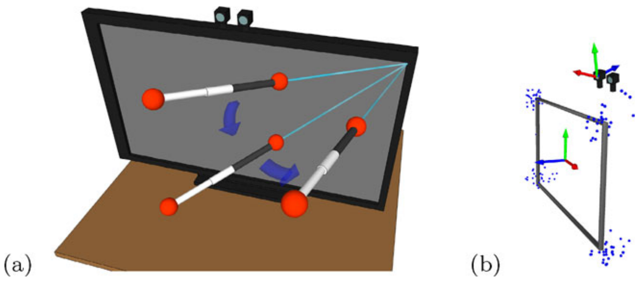


Fig. 8 Display calibration. (**a**) Wand prop containing a laser pointer and two colored markers. Several rays passing through the display corner are obtained by recovering the 3D position of the markers. (**b**) The rays for a given corner are intersected to yield a distribution of possible 3D positions for the display corner

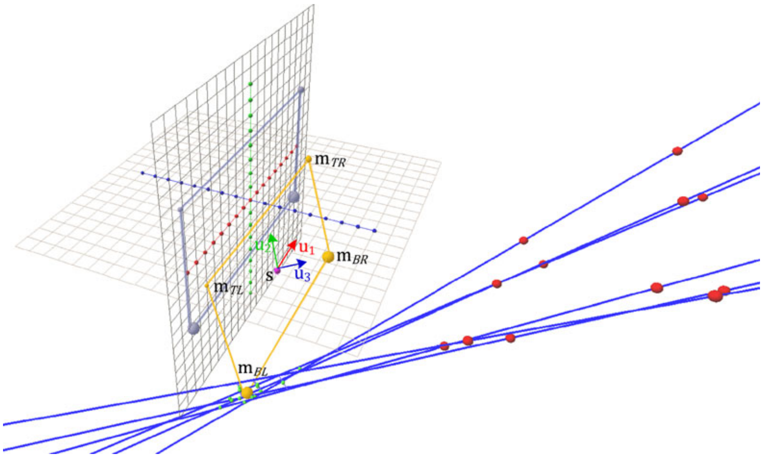


Fig. 9 Deducing the geometry of the display and its relative position to the world reference frame. Detected and rectified positions of the display in *yellow* and *white*. *Green spheres* indicate putative corners (inliers), and *red spheres* correspond to the colored markers on the calibration wand (Color figure online)

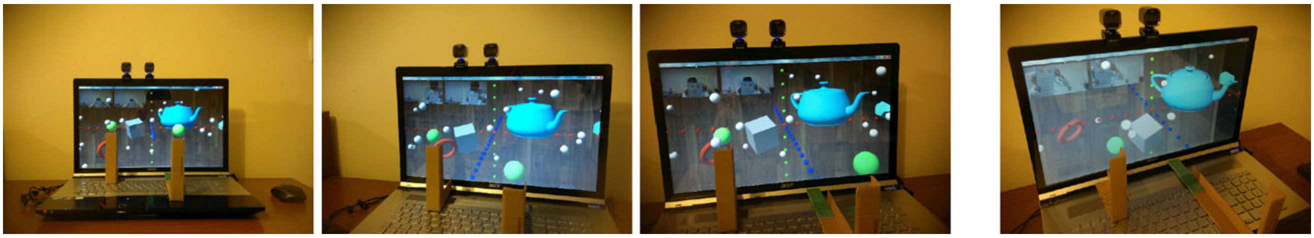


Fig. 10 *Left 3 images*: Representative results from the display, filmed through a small aperture in a cardboard cut-out of a face. As the camera changes position, the viewing frustum updates to create the illusion of viewpoint change. The relative scales of objects are correctly preserved; cf. the size of the green sphere relative to the visual reference provided by the cardboard pillars in front of the display. *Right image*: Image of the display captured by a camera immediately in-front of the display, whilst tracking a face to the top-left. Image distortion is apparent from this viewpoint, but is perceived as correct perspective when viewed from the top-left (Color figure online)

the horizontal and vertical axes of the display plane are used to derive the width (w) and height (h) of the display.

Figure 9 illustrates the automatic detection of the display plane. The yellow rectangle represents the deduced position of the display in world coordinates. The gray rectangle represents the window rectified to pass through the origin of the global coordinate system and oriented to its x–y plane, as assumed by the frustum calculation of Sect. 3.4. Thus the transformation used to align the triangulated viewer position **V** (Sect. 3.1) with the global reference frame used to position the frustum (and the virtual scene) is:

$$\mathbf{V}' = \mathbf{U}^{\mathrm{T}}\mathbf{V} - \boldsymbol{\mu} \tag{17}$$

where $\mathbf{V}' = [x_c + w/2, y_c + h/2, z_c]^{\mathrm{T}}$ defines the global display-centered reference frame required in Sect. 3.4.

4 Evaluation and Discussion

We objectively measured the performance of our display to determine its stability and responsiveness (latency). We also quantified the accuracy of the wand calibration process. Further, we conducted user evaluations to measure the efficacy of the display in promoting spatial awareness in a simple object counting task. This was measured objectively via time taken to perform the task. Finally, user feedback was collected through a structured interview debrief of the participants.

We now describe the experimental setup, and discuss the results, for each of the evaluations in turn.

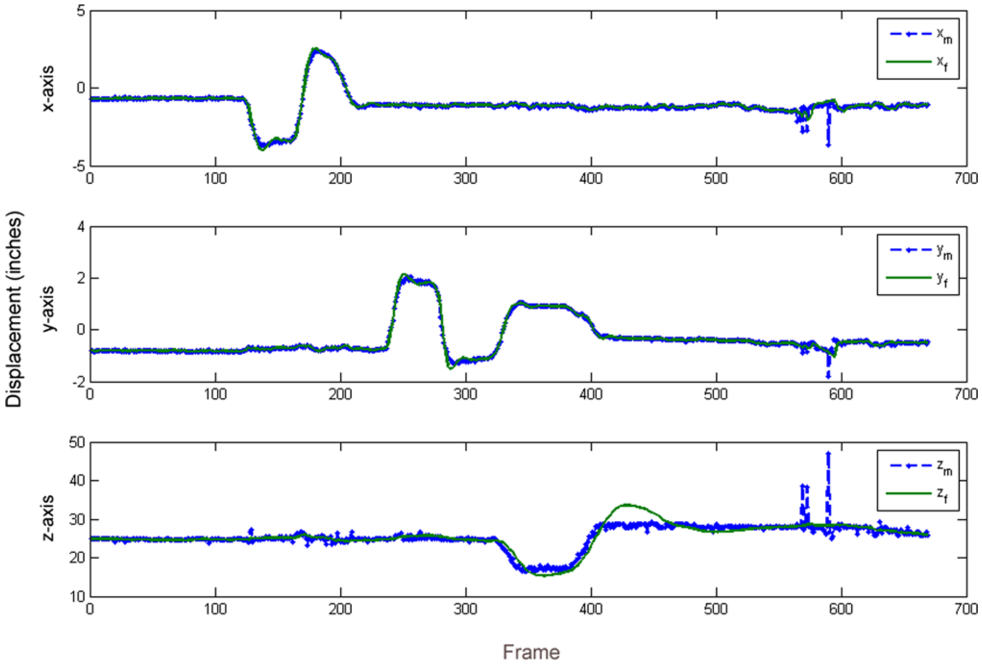
4.1 Viewpoint Adaptation Evaluation

The display was evaluated using a synthetic scene of around 50 objects at varying depths, creating the illusion of presence behind and in front of the display. Figure 10 illustrates the test scene rendered from a first-person perspective. Two cardboard pillars have been inserted into the scene. The display correctly scales and positions the virtual spheres to give the impression of each resting upon its corresponding cardboard pillar.

4.1.1 Tracker Stability

We first evaluate the stability of the viewer tracking, with and without the Kalman filtering. Figure 11 illustrates the influence on each of the 3D tracked coordinates of the viewer.

Fig. 11 Measuring stability and responsiveness of the eye tracker. Triangulated position in *blue*, filtered position in *green* (Color figure online)



Agile motion of the viewer (e.g. between frames 100–300) in which the eye is tracked successfully cause the filter to update quickly to the new viewer position; noisy erroneous tracks such as those around frame 600 are smoothed out. Note that at greater scene depths, the z estimate lags by ~400 ms serving to smooth out discontinuities due to noise in the 2D eye localization that, when triangulated, adversely affects the depth estimation.

In a further stability experiment, a stationary viewer was positioned at a central location in front of the display. The standard deviation in triangulated viewer position was measured over several hundred frames. Sub-millimeter scintillations were observed on the x- and y-axes and around 4 mm in the z-axis (Fig. 12(a)). As can be observed in the accompanying video, this results in virtually no perceptible position jitter. These accuracies compare favorably with commercially available Infra-Red (IR) based wearable tracking solutions such as the UM16 and RUCAP U-15 (both claiming 1 mm accuracy) though the frame rate of these solutions is much higher at up to 160 fps using their bespoke wearable hardware rather than commodity web cameras as here. Although the Kinect camera was not released at the time of this work, recent performance characterizations (Alnowami et al. 2011) place its depth (z) accuracy at ±1 mm at 1 meter though decreasing rapidly at both nearer or greater ranges, unlike our approach, and with a comparable frame rate of 30 fps.

4.1.2 Frame-Rate and Latency

Figure 12(b) illustrates the frame-rate of the system, which runs at an average of 24.872 fps (standard deviation ±1.001) regardless of the distance of the viewer from the display. A small performance drop is observed when the viewer is very close to the display, as an artifact of the cascade based detector (Viola and Jones 2001) and resulting larger face area to scan for the eye template. The breakdown of CPU time per frame, over 700 frames, is as follows:

- Frame acquisition 2.309 ms (±0.326).
- Viewer detection 22.158 ms (±1.276).
- Projection and rendering 15.283 ms (±0.802).

To measure the latency of the display an external high-definition camera running at 50 frames per second was used. Initially, the latency of the web camera hardware itself was measured using a light source introduced into the visual field of all cameras. The demonstration scene includes video feeds from the cameras showing the face tracking in operation. To obtain a value for the raw latency of the web cameras and display, the delay between the light being switched on, and its image appearing within the video feed was measured. A typical lag of 6 fields at 50 fps, equivalent to 120 ms (±10 ms), was observed. This delay is caused by the combined effect of the camera and the display lag. A similar test using a mouse-click as a trigger revealed that 80 ms (±10 ms) of this is due to display lag (perhaps due to hardware buffering or the display itself) and therefore 40 ms to the camera image acquisition. Both these are limitations of the commodity hardware platforms used, and the former would not be alleviated were one to replace our tracker with some hypothetical zero-latency tracking system.

Because there is negligible lag in the filter response for the x and y-axes, the total lag in response to viewer motion in these directions is about 120 ms. This delay was

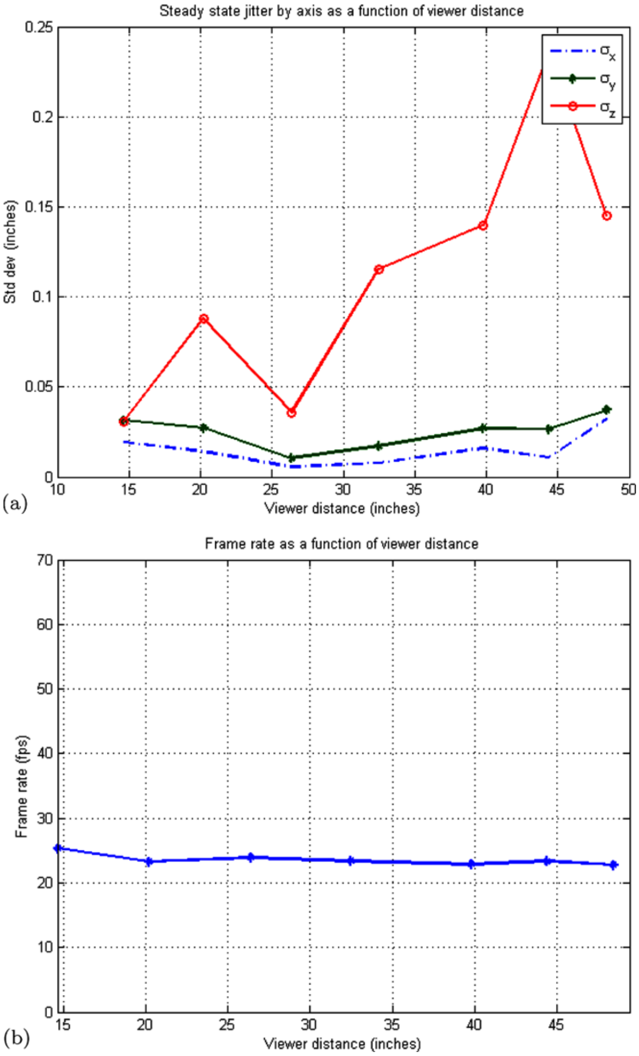


Fig. 12 Analyzing system performance versus viewer distance, using (**a**) scintillation (jitter) in position when stationary, and (**b**) overall frame rate (fps)

confirmed visually by measuring the number of frames lag between motion in the external video feed, and viewpoint change occurring on the display. The delay for significant motion in the z-axis is somewhat greater because of the filter-induced lag of approximately 4 cycles which brings the total lag for z-axis motion up to 225 ms. This leads to an overall system latency of ∼120–220 ms. While this latency is perceptible, it is well within the range required for interactivity (Ellis et al. 2002) (outside of which the user would have to adopt a ‘move and wait’ approach).

4.2 Wand Calibration Evaluation

We measured the accuracy of the wand-based display calibration process by comparing the estimated width and height of the screen with the physically measured values of 16 and 9 inches respectively. The average screen width was 16.9 inches (±0.31), and height was 8.6 inches (±0.13) over five repetitions of the calibration process. An example from one such run is given in Fig. 9.

Table 1 For each screen corner, the mean and standard deviation of the estimated position across five repetitions of the calibration procedure (in inches). The *last row* is the mean of the standard deviations over the corners

Corner	x-position	y-position	z-position
TL	-7.38 ± 0.48	-10.11 ± 0.34	2.51 ± 0.90
TR	-7.10 ± 0.23	-1.75 ± 0.35	-0.22 ± 0.71
BL	9.42 ± 0.49	-1.99 ± 0.38	0.82 ± 0.98
BR	9.15 ± 0.59	-10.34 ± 0.47	3.55 ± 0.83
Mean Std.	0.445	0.383	0.853

Estimation error in the calibrated corner positions over these five runs are given in Table 1. The final row describes the average standard deviation in displacement over the four corners. As could be expected, the z-axis measurements exhibit more variation across calibration sessions. To promote robust measurements of wand pose some automatic checks are done when measuring the wand position. As the angle between the wand and the any previous wand capture for a particular corner is too small (less than 5°) the associated intersection is deemed unreliable, the measurement discarded and the user asked to reposition the wand. The system also checks that the detected length of the wand is close to its *a priori* known true length and that only two circles were detected in each frame. The main source of error in the screen calibration is imprecise circle localization leading to imprecise detected wand poses. Further work will investigate use of alternative sphere detection methods (for instance taking into account projection-induced elongation near frame boundaries).

The supplementary video accompanying this paper contains footage of the 3D display in operation, with the illusion filmed both from the perspective of the viewer and from a fixed on-looking position.

4.3 Usability Evaluation

To determine the efficacy of the display as a tool for interactively exploring 3D space, a user study was conducted. Users were set the task of counting a set of labeled objects within the 3D scene. The scene contained a large hollow Bucky-ball-like object, in addition to visual cues such as axes, grids and a background texture. Several small spheres which textured with the labels A, B and C are distributed throughout this complex structure (Fig. 13). At the start of each test session, a random number (between 1 and 5 inclusive) of these spheres are placed in the scene at randomly generated positions mostly within the ball (75 % chance) but

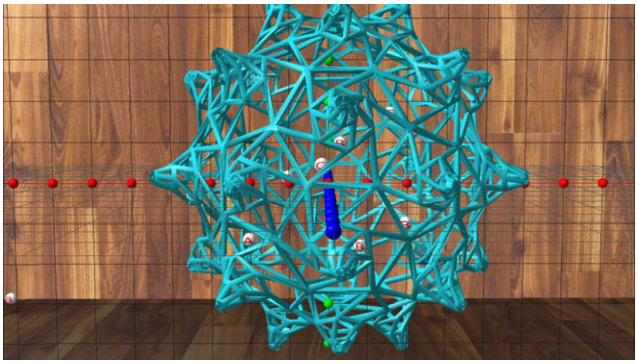


Fig. 13 Example of the user test scene, which is generated with a random number of spheres A, B and C located at random positions, mostly within the mesh ball (thus requiring the viewer to move around see all of them)

also in peripheral areas in front of the screen (10 % chance) and behind it (15 % chance).

4.3.1 Experimental Setup

The evaluation task involves the user mentally counting the number of each sphere type (A, B, or C) within the scene as quickly and accurately as possible, and recording the result on a physical questionnaire. Users repeat the test three times and for each run, we log the true number of each type of sphere as well as the user's time taken to complete the task.

Eleven volunteers were recruited, each with everyday working knowledge of PCs and mouse pointing devices. The volunteer group comprised nine male and two female participants, of which three wore glasses. All of the participants were in their mid-to-late twenties.

We wished to determine whether our 3D display was comparable to a mouse-based interface, in terms of time taken to complete this task. Users were therefore also asked to repeat the task using a mouse to navigate the scene. In the mouse-control configuration, left-right and up-down motion rotates the scene about the y and x-axes, respectively; the scroll wheel moves the viewpoint forward or back so affecting a zoom.

Each test subject was given a brief verbal overview of the task to be performed. Each subject was given a test run in each mode before his/her timed tasks began. On each test subject, three repetitions of the task were made in each mode (alternating between modes to avoid a bias). The program was then executed, the task timer beginning when a freshly generated scene is first displayed and stopping when the space-bar is pressed (at which point the scene is hidden from view). Because of the deliberate occlusion of some of the spheres by the Bucky-ball object and the positioning of some of them off to the side or relatively far out in front of the screen, the test subject is required to move about the scene to count all the spheres.

Two evaluation criteria were used for the counting task. The first is the total time taken to count all the spheres divided by the (true) number of spheres: the counting rate in seconds per sphere. The second criterion is the counting error-rate: the sum of counting errors (absolute difference between true count and reported count) across all sphere types divided by the number of spheres (a ratio or percentage).

4.3.2 Usability Results

Our experimental setup is to determine whether the usability of our 3D display was comparable to a mouse-based interface. The null hypothesis is therefore that a significant difference in timing should be observable between the proposed viewer-tracking and mouse configurations.

The mean time to completion (TTC) of task (averaged across all test subjects and all repetitions) was 2.76 s/sphere for viewer-tracking mode and 2.58 s/sphere for mouse-control mode. To see whether or not there is a significant difference in TTC between the modes of operation over the test sample, a paired t-test was performed. The mean time difference (tracking-mode time minus mouse-mode time) is 0.092 s/sphere with standard deviation 0.797 s/sphere. This leads to a t value of 0.381 and a p value of 0.359. The 95 % confidence interval for the time difference is −0.44 to 0.63 seconds per sphere. We therefore conclude that neither interface mode is significantly quicker than the other.

The mean counting error-rate (averaged across all test subjects and all repetitions) was 9.40 % for viewer-tracking mode and 10.21 % for mouse-control mode. Again, a paired t-test was done to check the significance. The error-rate difference (tracking-mode time minus mouse-mode time) is −0.81 % with standard deviation 7.95 %. This leads to a t value of −0.336 and a p value of 0.365. The 95 % confidence interval for the error-rate difference is −6.15 % to 4.54 %. Because this lies on either side of zero, we conclude that neither mode leads to a significantly higher error-rate than the other.

We conclude that on average the proposed display offers no significant disadvantage in 3D spatial exploration tasks that a mouse, though one mode or the other may be better for a particular user. To test the latter hypothesis, a t-test was performed for each participant individually. This is possible as user has three repetitions of the task for each mode. The Welch's t-test (Welch 1947) can be used to test the hypothesis when the variance of the two populations is not necessarily the same. The Welch–Satterthwaite equation was used to get the degrees of freedom for a Student's-t distribution from which the 95 % confidence bounds can be determined. No significant difference in speed or accuracy performance was found for any of the 11 test subjects.

5 Conclusion

We have demonstrated a novel display system, providing a robust natural user interface for volumetric (free-viewpoint) 3D visualization. The system requires no glasses or other specialist hardware, beyond a pair of fixed VGA web-cam's and a standard 2D display. The display is capable of running at sustained real-time (25 fps) rates on a commodity laptop PC, and exhibits very little perceptible jitter (only a few millimeters) due to depth-adaptive Kalman filtering of the tracked viewer position.

Furthermore we have shown that a 3D spatial exploration task performed using a mouse can be performed in statistically similar time using our proposed display. This, despite users having everyday prior experience of the former input device. This argues in favor of the display's usability, and suitability for 3D manipulation tasks when a mouse or similar pointing device is undesirable (e.g. tablets or wall-mounted flat-screen displays).

In addition to our tracking system we developed, implemented and tested a novel wand-based screen calibration system. The wand is simple to construct and the proposed calibration routine uses it to measure the size, position and orientation of the screen with respect to the cameras. Thus, the cameras do not need to directly capture an image of the screen in order to determine its geometry. This calibration process allows for easy system set up with any display device from a small laptop LCD to a larger projector screen with the camera rig placed in any convenient position and orientation. The calibration is performed once during setup and is not required in subsequent interaction with the display. Future improvements might harness solutions for the online tracking of calibration parameters (Dang et al. 2009; Thacker 1992) to enable movement of the screen relative to the stereo camera pair. The fixed hardware configuration of our project did not raise this requirement.

The display device used in these experiments is a single 18.4 inch LCD monitor built into a laptop PC. It would be informative to test the performance of the adaptive viewer-tracking system on a variety of display devices (including a stereoscopic display). Among these should be included a larger LCD display, a projector screen and a tablet PC. The user test task of examining a 3D object/scene may be easier with a tablet PC than with a fixed screen since the user can hold the tablet and easily rotate it. The system could also be extended to work with multiple LCD monitors arranged in an arc giving a more immersive viewing experience, whilst remaining relatively inexpensive.

Although our system exhibits real-time frame rates, it currently exhibits a lag of ∼120 ms for x- and y-axes up to ∼225 ms for the z-axis. As discussed in Sect. 4 much of this is not due to our tracking, but to the display hardware. Emerging consumer depth cameras such as the Microsoft Kinect could also potentially be substituted for the gaze triangulation step of Sect. 3.1, and were unavailable at the outset of this project. We contrast our tracking accuracy and frame rate with Kinect in Sect. 4.1. One of the main concerns of our test subjects was the breakdown in the 3D illusion which occurs whenever the viewers exceed the field of view of the cameras or tilt their heads too much. A further enhancement might be to consider more than two cameras (enabling wider viewing angles to be covered by the system) and to introduce robustness against large head rotations. However we do not believe such improvements are necessary to demonstrate the robustness and efficacy of our novel display system. Rather, ongoing work explores graphics applications of the system including pre-visualization of captured 3D assets and animations.

References

Alnowami, M., Alnwaimi, B., Copland, M., & Wells, K. (2011). A quantitative assessment of using the Kinect for Xbox 360 for respiratory surface motion tracking. In *Proc. SPIE medical imaging*.

Brar, L., Sexton, I., Surman, P., Bates, R., Lee, W., Hopf, K., Neumann, F., Day, S., & Williman, E. (2010). Laser-based head-tracked 3D display research. *Journal of Display Technology*, *6*(10), 531–543.

Chen, C., Huang, Y., Chuang, S., Wu, C., Shieh, H., Mphepo, W., Hsieh, C., & Hsu, S. (2009). Liquid crystal panel for high efficiency barrier type autostereoscopic 3D displays. *Applied Optics*, *48*(18), 3446–3454.

Dang, T., Hoffmann, C., & Stiller, C. (2009). Continuous stereo self-calibration by camera parameter tracking. *IEEE Transactions on Image Processing*, *18*(7), 1536–1549.

Dodgson, N. (2004). Variation and extrema of human interpupillary distance. *Proceedings of SPIE*, *5291*, 36–46.

Ellis, S. R., Wolfram, A., & Adelstein, B. D. (2002). Three dimensional tracking in augmented environments: user performance trade-offs between system latency and update rate. *Proceedings of the Human Factors and Ergonomics Society annual meeting*, *46*(26), 2149–2153.

Erden, E., Kishore, V., Urey, H., Baghsiahi, H., Willman, E., Day, S., Selviah, D., Fernandez, F., & Surman, P. (2009). Laser scanning based autostereoscopic 3D display with pupil tracking. In *Proc. IEEE photonics* (pp. 10–11).

Ezra, D., Woodgate, G., Omar, B., Holliman, N., Harrold, J., & Shapiro, L. (1995). New autostereoscopic display system. In *Proc. SPIE Intl. Society of Optical Engineering* (pp. 31–40).

Free2C (2010). *The free2c desktop display* (Technical report). Heinrich Hertz Institute.

Freund, Y., & Schapire, R. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, *14*(7), 771–780.

Lee, J. (2008). *Head tracking for desktop VR displays using the WiiRemote* (Technical report). Carnegie Mellon University.

Malleson, C., & Collomosse, J. (2011). Volumetric 3D graphics on commodity displays using active gaze tracking. In *Proc. ICCV workshop on human computer interaction*.

Nishimura, H., Abe, T., Yamamoto, H., Hayasaki, Y., Nagai, Y., Shimizu, Y., & Nishida, N. (2007). Development of a 140-inch autostereoscopic display by use of full-color LED panel. *Proceedings of SPIE, the International Society for Optical Engineering*, *6486*, 64861B.

OpenCV. Open source computer vision library. Accessed July 2011.

Perlin, K., Poultney, C., Kollin, J., Kristjansson, D., & Paxia, S. (2001). Recent advances in the NYU autostereoscopic display. *Proceedings of SPIE, the International Society for Optical Engineering*, *4297*, 196–203.

Sandin, D., Margolis, T., Dawe, G., Leigh, J., & DeFanti, T. (2001). Varrier autostereographic display. *Proceedings of SPIE, the International Society for Optical Engineering*, *4297*, 204–211.

Schwartz, A. (1985). Head tracking stereoscopic display. In *Proc. IEEE intl. conf. on display research* (pp. 141–144).

Sorensen, S., Hansen, P., & Sorensen, N. (2004). Method for recording and viewing stereoscopic images in color using monochrome filters. U.S. Patent 6687003.

Surman, P., Sexton, I., Hopf, K., Lee, W., Buckley, E., Jones, G., & Bates, R. (2008a). European research into head tracked autostereoscopic displays. In *Proc. conf on 3DTV* (pp. 161–164).

Surman, P., Sexton, I., Hopf, K., Lee, W., Neumann, F., Buckley, E., Jones, G., Corbett, A., Bates, R., & Talukdar, S. (2008b). European research into head tracked autostereoscopic displays. *Journal of the Society for Information Display*, *16*, 743–753.

Takaki, Y. (2006). High-density directional display for generating natural 3D images. *Proceedings of the IEEE*, *94*(3), 654–663.

Tetsutani, N., Ichinose, S., & Ishibashi, M. (1989). 3D-TV projection display system with head-tracking. In *Japan Display* (pp. 56–59).

Tetsutani, N., Omura, K., & Kishino, F. (1994). Study on a stereoscopic display system employing eye-position tracking for multiviewers. *Proceedings of SPIE, the International Society for Optical Engineering*, *2177*, 135.

Thacker, N. A. (1992). Online calibration of a 4 DOF stereo head. In *Proc. British machine vision conference (BMVC)* (pp. 528–537).

Tsai, R., Tsai, C., Lee, K., Wu, C., Lin, L., Huang, K., Hsu, W., Wu, C., Lu, C., Yang, J., & Chen, Y. (2009). Challenge of 3D LCD displays. *Proceedings of SPIE, the International Society for Optical Engineering*, *7329*, 732903.

Urey, H., & Erden, E. (2011). State of the art in stereoscopic and autostereoscopic displays. *Proceedings of the IEEE*, *99*(4), 544–555.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proc. computer vision and pattern recognition*.

Welch, B. L. (1947). The generalization of student's problem when several different population variances are involved. *Biometrika*, *34*(1–2), 28–35. doi:10.1093/biomet/34.1-2.28.

Woodgate, G., Ezra, D., Harrold, J., Holliman, N., Jones, G., & Moseley, R. (1997). Observer-tracking autostereoscopic 3D display systems. *Proceedings of SPIE, the International Society for Optical Engineering*, *3012*, 187–198.

Woodgate, G., Harrold, J., Jacobs, A., Mosely, R., & Ezra, D. (2000). Flat-panel autostereoscopic displays: characterization and enhancement. *Proceedings of SPIE, the International Society for Optical Engineering*, *3957*, 153–164.

Woods, A. (2009). 3D displays in the home. *Information Display*, *7*, 8–12.

Yamamoto, H., Kouno, M., Muguruma, S., Hayasaki, Y., Nagai, Y., Shimizu, Y., & Nishida, N. (2002). Enlargement of viewing area of stereoscopic full-color LED display using parallax barrier combined with polarizer. *Applied Optics*, *41*(32), 6907–6919.