# We are IntechOpen, the world's leading publisher of Open Access books
# Built by scientists, for scientists

**6,900**
Open access books available

**186,000**
International authors and editors

**200M**
Downloads

Our authors are among the

**154**
Countries delivered to

**TOP 1%**
most cited scientists

**12.2%**
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

## Interested in publishing with us?
## Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Chapter 3

# High Fidelity Immersive Virtual Reality

Stuart Gilson and Andrew Glennerster

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/50655

## 1. Introduction

Virtual reality is now used as an aid to teaching and practice in a wide range of fields. In education, surgery, architecture and many other disciplines, virtual reality provides a tool that allows users to experience situations that could not be readily reproduced in the real world. As virtual reality technology becomes more pervasive, there will be a rising demand for improvements in the quality of VR displays, both in the accuracy with which they portray the location, shape and surface detail of objects and also the speed and fluidity with which the images on the display change in response to the observer's head movements, becoming ever closer to the visual feedback that observers experience in the real world.

The ultimate goal of virtual reality is to present to the user computer-generated scenes in such a manner that the user should not be aware that they are in a virtual reality system at all – a sort of Turing test [29] for VR. To achieve this, the images should be of such high resolution that the individual pixels cannot be distinguished, the display should change with the observer's head movements without perceptible lag and the rendered scenes should be exquisitely detailed with realistically rendered three-dimensional models.

Current capability in virtual reality falls some way short of this ideal. Nevertheless, recent advances in head tracking technology and pixel-drawing speed make this a useful point to pause and consider the state of the art in presenting high-fidelity immersive virtual reality. Issues that we will address in this chapter include approaches to minimise tracking latency; detection of and compensation for spatial distortions in head mounted displays and methods to check that a rendered scene corresponds well with the real-world rays of light that are being simulated (or 'virtualized'). These questions are, to a large extent, independent of the technology used.

To clarify terminology and lay a framework for subsequent discussion, we consider a virtual reality system to be comprised of three subsystems: tracking, rendering and display. Furthermore, we consider a 'data pipeline' to consist, broadly, of a real world event being sampled by the tracker, the generation of an appropriate scene on the VR graphics computer, and the final rendering of pixels in the display device. In this chapter we will examine the

INTECH
open science | open minds

©2012 Gilson and Glennerster, licensee InTech. This is an open access chapter distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

sources of error in each of these three subsystems, as data travels along the pipeline. First, though, we briefly consider the differences between 'virtual' and 'augmented' reality in the context of the VR Turing test.

### 1.1. Virtual and augmented reality

Given the speed at which computer graphics is developing and the increasing realism of films based on computer-generated scenes, there is a strong argument that augmented reality will be the first virtual technology to pass the VR Turing test and fool observers completely as to whether they are viewing a real or a virtual scene. Augmented reality – superimposing computer drawn graphics on the real world – is at present best achieved with a video see-through HMD, which uses small cameras to capture real-world images and displays them in the HMD. The digital images can have computer graphics overlaid, augmenting the real world scene. In practice, most systems suffer from an incorrect placement of the optic centres of the images – the optic centres of the two cameras do not coincide with the observer's eyes' optic centres, and thus lead to a mis-match between proprioception ('muscle sense') and vision. In practice, appropriately placed mirrors can minimize this mis-match [12] although most video-see-through augmented reality (AR) systems do not yet go to this trouble. There are distinct advantages to video-see-through AR. Because only a small portion of the entire scene is being computer-generated a great deal of computing power can be devoted to rendering the virtual objects accurately and in detail. Also, the problem of spatially aligning the virtual object so that it sits stably on a real surface is much easier to handle when both the real and virtual scenes are available in electronic form. Algorithms for ensuring that this alignment is correct can be run on each frame before display to the user, an option that is not possible in see-through augmented reality where the user sees the real world directly and virtual objects are superimposed, for example using a half-silvered mirror. Results in this area even for cameras with very rapid, jerky movements, are impressive [19].

On the other hand, a technology that is unlikely to fool observers completely is a CAVE. In a CAVE, the rendered VR scene is projected on up to 6 surfaces enclosing the observer. Their key advantage is that the observer need only wear a light-weight pair of shutter glasses, which provides the stereo signal to the observer's eyes. However, CAVEs suffer from various physical limitations, including the amount of space they require, and the fact that they can only really accommodate one observer at a time. But most significantly, the effective resolution of the VR scene is dependent on the distance of the projection surface to the observer and, for a freely-moving observer, this effective resolution will change. For our goal of virtualizing the rays an observer would see in the real world, a CAVE-based VR system will never suffice.

### 1.2. Head mounted displays

The majority of head-mounted displays – as the name suggests – involve miniature displays mounted to a helmet-like structure which rests on the observer's head, such that the displays lay in front of the eyes. Helmet designs range from fully-enclosed helmets, typically found in military applications, to bare-minimum ski-goggle like systems.

At present, all systems have non-negligible weight, which can be a problem for prolonged use. Even in the shorter term, having such a weight mounted on the head will restrict freedom of

movement, and introduce rendering errors during high acceleration head movements – even when tightly strapped to the head, the weight of the headset under acceleration will 'twist' the scalp and the resulting non-rigid motion will appear as lag between the observer's movement and the viewed VR scene.

Furthermore, all present HMDs are tethered to some form of control box. This control box typically converts the video signal generated by the VR graphics computer, sent via VGA or DVI cable, into the proprietary-format signal required by the custom display in the HMD. To reduce damage from the continual usage, these tethering cables are generally sturdy, robust and comparatively inflexible. VR users will be aware of the cable, and may modify their behaviour to reduce any 'tugging' that the cable might cause on the HMD. This makes our ideal "freely moving" observer less free to move. Some HMD systems offer wireless HMDs, with the control box transmitting the video signal which is in turn detected by a battery powered receiver connected to the HMD. While this can alleviate many of the problems associated with thick tethering cables, the encode-transmit-receive-decode cycle of the video frames will introduce extra latency into the system (e.g. one contemporary wireless system adds "2 video frames latency"), which can be problematic in itself.

## 2. High fidelity virtual reality

Consider a VR application that attempts to virtualize a natural scene – consisting of a wide spectrum of colours, and numerous objects (e.g. trees, rocks) spanning a considerable distance in depth. While observing this scene, the user is free to move around the scene, turn their head, and to shift their gaze. At any moment in time, there will be an array of light rays hitting each retina, and it is these rays – their direction, intensity and wavelength – that we wish to duplicate in our VR system. If this is achievable for an object that is viewed from a wide range of viewpoints (in theory, all viewpoints), then it is justifiable to claim that the real object has been re-created in virtual reality, or 'virtualized'.

With this goal in mind, what are the obstacles preventing us from creating such an application with present-day hardware? Ultimately, these obstacles are issues of accuracy which we discuss here in terms of spatial, temporal and spectral components.

### 2.1. Spatial accuracy

Spatial errors are due to a misalignment of the real and virtual rays, and can arise from an incorrect estimation of the observer's head pose or location, or due to mis-calibration of the head mounted display. The first is a technology limitation, but the latter is a largely-soluble calibration issue.

#### *2.1.1. Tracker accuracy*

All tracking systems monitor movement by sampling changes of some continuous variable which is sensitive to motion. Magnetic systems sample the tiny current induced in sensors as they move through a (static) magnetic field generated by an emitter. Inertial systems use extremely sensitive accelerometers to sample changes in acceleration in any of the three planes of movement. Mechanical systems use articulated arms to modify potentiometers at the joints

of the arms to produce a single 3D location. Finally, optical systems use three or more cameras to reconstruct movement in 3D from multiple 2D images.

All these systems have numerous advantages and disadvantages, and the choice for any given VR system will depend on many factors, including cost, the size of volume to be tracked, and the types of actions/movements that will be expected in the application (e.g. walking versus object manipulation by hand). Consideration should also be given to whether a given system needs to combine the signals from several sensors, which may lead to discontinuities as a tracked object moves between sensors [3].

For high-fidelity VR, we have found that optical systems offer the best attributes. Multiple cameras can be mounted to the perimeter of the tracked volume, and will track the position of passive markers on the observer (notably, on the HMD) and on tracked objects. Modern optical systems are real-time, using high-resolution, high-frame-rate cameras to sample the tracked volume to a high spatial and temporal accuracy (typically less than 1mm and 10ms respectively).

### *2.1.2. Spatial calibration of head mounted display*

We consider the correct spatial calibration of the HMD to be one of the most critical components of a VR system. Without calibration, users can misinterpret the virtual world (for example, they often underestimate distances to objects which can be a symptom of an incorrect calibration [11]). Users can also experience premature fatigue and, with severely mis-configured HMDs, even nausea. [17, 18, 22].

Each of the HMD's displays are characterized by a frustum, which completely determines how 3D vertices in the VR world are transformed to pixel locations in the display (see figure 1). The frustum is determined from the HMD's resolution in pixels ($w$ and $h$), and the horizontal and vertical field-of-view (FOV) in degrees ($x$ and $y$). From these we can compute the centre pixel ($c_x = \frac{w}{2}$ and $c_y = \frac{h}{2}$) and the focal length ($f_x = \frac{c_x}{\tan(x)}$, $f_y = \frac{c_y}{\tan(y)}$), and then compute the intrinsic matrix:

$$\mathbf{P} = \begin{bmatrix} \frac{2\times\text{ncp}}{\text{right}-\text{left}} & 0 & \frac{\text{right}+\text{left}}{\text{right}-\text{left}} & 0 \\ 0 & \frac{2\times\text{ncp}}{\text{top}-\text{bottom}} & \frac{\text{top}+\text{bottom}}{\text{top}-\text{bottom}} & 0 \\ 0 & 0 & -\frac{\text{fcp}+\text{ncp}}{\text{fcp}-\text{ncp}} & -\frac{2\times\text{fcp}\times\text{ncp}}{\text{fcp}-\text{ncp}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

where:

$$\text{left} = -\text{ncp} \times \frac{c_x}{f_x}, \qquad \text{right} = \text{ncp} \times \frac{w - c_x}{f_x},$$

$$\text{bottom} = -\text{ncp} \times \frac{c_y}{f_y}, \qquad \text{top} = \text{ncp} \times \frac{h - c_y}{f_y}$$

The near- and far-clipping planes (ncp and fcp) are application dependent. The intrinsic matrix, $\mathbf{P}$, describes how view-centric 3D coordinates are transformed to 2D pixel locations. The view-centric coordinates are obtained via the extrinsic matrix, $\mathbf{S}$, which transforms the 3D world coordinates relative to the position and orientation of the display:

$$\mathbf{S} = \begin{bmatrix} \mathbf{R} & \mathbf{T}^{\mathrm{T}} \\ 0 & 1 \end{bmatrix}$$

where $\mathbf{R}$ is a $3 \times 3$ rotation matrix and $\mathbf{T}$ is a $1 \times 3$ translation matrix.

Using the manufacturer's specification, it is possible to create a pair of intrinsic matrices for a stereo HMD. Conversely, the contents of the extrinsic matrix are entirely determined by the position of the 'tracked centre' – the point that is reported by the tracking system to the VR graphics computer – which depends on how the tracked markers are attached to the HMD. Unfortunately, manufacturer specifications are rarely of sufficient accuracy to qualify for high-fidelity VR display, and it can be very difficult to obtain accurate physical measurements from within the confines of a small HMD for the extrinsic matrix.

Instead, we need to calibrate the display. Yet, a thorough calibration is often neglected, perhaps understandably given the difficulties in obtaining suitable calibration values. We approach HMD calibration using a technique from camera calibration [27], called photogrammetry. If we treat the HMD displays as virtual cameras, we can use a modified version of this method to calibrate our HMD.

In Tsai's method [27], the objective is to take multiple pictures of a calibration object of a known geometry, and to pair the object's corners with their corresponding pixel locations in each of the images. Given sufficient images, these point correspondences can be used to constrain a minimization function, whose parameters are the intrinsic (five parameters: width, height, focal length and location of the centre pixel) and extrinsic properties (six parameters: 3D position and pose) of the camera.

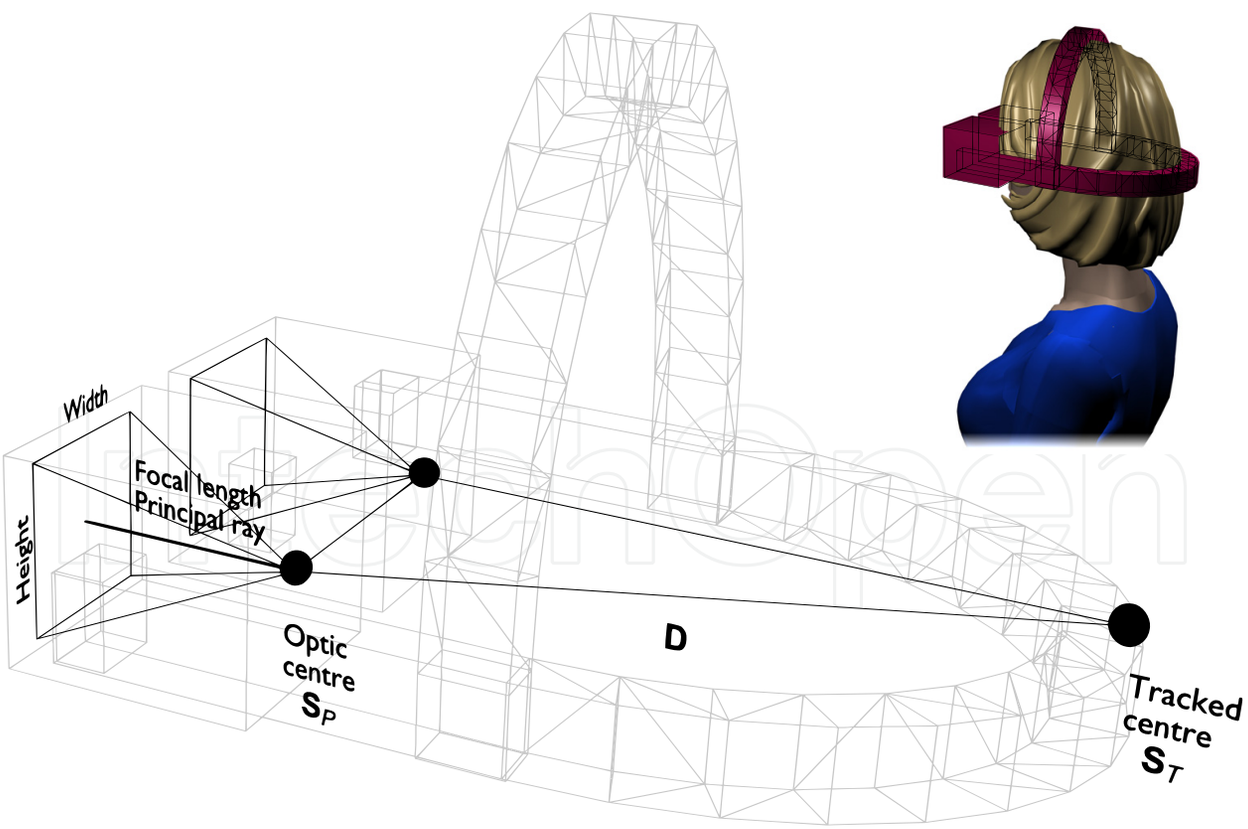


**Figure 1.** Intrinsic (width, height, focal length) and extrinsic (location of optic centre, and direction of principal ray, $\mathbf{S}_P$) shown relative to the tracked centre of the HMD ($\mathbf{S}_T$). $\mathbf{D}$ is the transformation between $\mathbf{S}_T$ and $\mathbf{S}_P$, i.e. $\mathbf{D} = \mathbf{S}_P\mathbf{S}_T^{-1}$.

Our method is as follows. A camera is mounted on a table in such a way that the HMD can be placed atop the camera and removed without moving or otherwise disturbing the camera. The HMD is positioned in such a way that the camera can 'see' as much of one of the displays as possible. A simple chequerboard grid is shown in this HMD display, and an image of this grid is captured and saved by the camera. Each of the vertices in this image (automatically extracted with image processing) can be paired with the known coordinates from the generated grid in the HMD display and, hence, we have a means of converting any camera image location to the corresponding HMD image location (see [4] for more details). We subsequently express all captured camera image coordinates in HMD coordinates and, crucially, we derive estimates of the HMD display frustum instead of the camera.

Ultimately, we need to know the display's extrinsic parameters relative to the tracked centre of the HMD. Yet, photogrammetry will give us the extrinsic parameters relative to the tracker's origin. If we record the HMD position and orientation at this time, we will be able to compute this relative transform (the component **D** in figure 1).

The next step is to record the image locations of a tracked calibration object. Somewhat counter-intuitively, this step must be carried out without the HMD in place. It is essential that the camera is not moved at this time, since this preserves the relationship between the recorded HMD chequerboard image and the corresponding 3D marker projections. Thus, we are able to relate any camera image features to the corresponding HMD display location, as if the HMD were in place and transparent.

Now we capture multiple images of a moving tracked marker. For each captured image, $i$, we will obtain 5 numbers – the marker's 3D location (in tracker coordinates, $\mathbf{X}_i$), and the 2D pixel coordinate of the marker projection in the camera. At this point (and henceforth in this section) the camera image coordinate are transformed into the corresponding HMD image location ($\mathbf{x}_i$).

The set of 3D coordinates $\mathbf{X}$ and 2D projections $\mathbf{x}$ are linked by a unique relationship – the 11 parameters which describe the frustum at the time the coordinates were recorded. The aim of photogrammetry is to find the 11 parameters that define this relationship. A proper treatment of this approach is beyond the scope of this chapter (see instead [5, 7, 27] for more information). Briefly, for any hypothesized frustum, we can compute the projection of $\mathbf{X}$ in the frustum image plane, to give $\mathbf{y}$. The difference between $\mathbf{x}$ (the original 2D projection) and $\mathbf{y}$ (the computed projection from $\mathbf{X}$) can be used as an error measure, and minimized to provide an estimate of the frustum's 11 parameters (see figure 2).

We thus obtain estimated values for the HMD display's intrinsic parameters, and its position and pose (denoted $\mathbf{S}_P$ in figure 1) relative to the tracker's origin. As we discussed earlier, since we also know the HMD tracked centre in the same coordinate frame ($\mathbf{S}_T$), we are now able to compute the transformation between the two ($\mathbf{D}$):

$$\mathbf{D} = \mathbf{S}_P \mathbf{S}_T^{-1}$$

Consequently, for any HMD position ($\mathbf{S}'_T$), we can now compute the appropriate optic centre and principal ray for the HMD display.

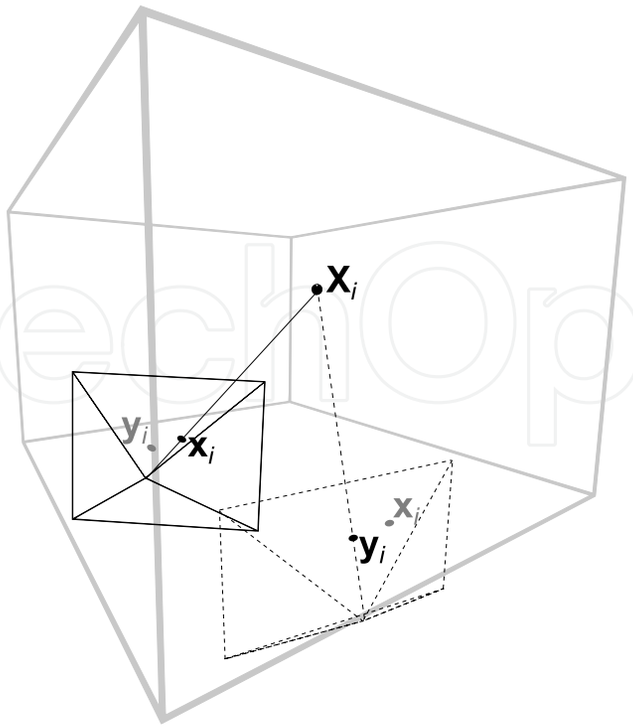


**Figure 2.** The $i^{th}$ 3D point ($\mathbf{X}_i$) projects to the camera image plane (solid lines) as pixel $\mathbf{x}_i$ (solid line). We estimate an HMD frustum (dashed lines) based on the projections for many $i$, and from this frustum are able to compute where $\mathbf{X}_i$ will project to in the new, hypothesised frustum ($\mathbf{y}_i$). Due to measurement errors, this estimate will be wrong, with an error of $|\mathbf{y}_i - \mathbf{x}_i|$ for all $i$. The difference in position and shape of the camera and frustum has been exaggerated here for illustrative purposes – after calibration, there is typically less than one pixel error per point, and the camera and frustum will almost be coincident.

For virtual reality applications, we can easily use the estimated frustum in common 3D graphics languages like OpenGL:

```
// Switch to intrinsic (projection) matrix mode.
glMatrixMode(GL_PROJECTION);

// Load intrinsic matrix, P.
glLoadMatrix(P);

// Switch to extrinsic (modelling) matrix.
glMatrixMode(GL_MODELVIEW);

// Load HMD_to_optic_centre transform.
glLoadMatrix(D);

// Incorporate the latest tracker transform.
glMultMatrix(S_prime_T);

// Draw the left-eye's scene.
drawScene();

// Repeat for the right-eye scene...
```

Values must be resolved for 11 parameters for each display. Inherent noise in the measurement systems (camera and tracker) means that many samples must be captured to allow robust estimation of these parameters. The method can be generalized to that of a single tracked point being dynamically moved within the field of view of the frustum. The camera, in

'movie mode', records the 2D locations of $\mathbf{x}_i$ in real time, permitting a high number of point correspondences to be gathered in a short period (see figure 3). We have found that 100 such correspondences are necessary for an accurate calibration, with 1000 or more being desirable [5].

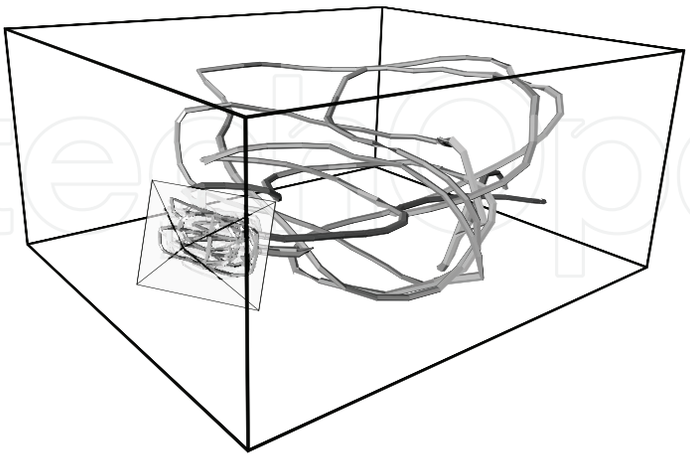

**Figure 3.** Graphical reproduction of one of the calibration trajectories and its projection into the frustum image plane (not all ≈3000 points are shown, for clarity).

With the modified photogrammetry method described above, we have obtained HMD display calibrations with a mean error less than one pixel. For reference, deriving intrinsic parameters from the HMD manufacturer's specifications, and extrinsic parameters by manual measurement yielded a mean error in excess of 100 pixels. This error would be very noticeable to the user, and is largely due to mis-measurement of the display's principal ray when measuring the extrinsic parameters. With trial and error this value could be reduced, but this is precisely the kind of error that proper calibration avoids. For example, keeping our calibrated extrinsic parameters but retaining the manufacturer's specifications for the intrinsic parameters gave an error of 13 pixels. In other words, an out-of-the-box HMD calibration will be at least this bad.

We believe this is a marked improvement over existing HMD calibration methods [15, 20, 28], although these papers did not provide pixel errors and so could not be compared directly with our method. These methods also require many judgments from a skilled human observer which, as we outlined above, would take a prohibitive amount of time to capture enough samples to robustly estimate the display parameters.

Note that the optic centre and principal ray obtained using this method are actually those of the camera, and not of the display. Our early constraint that the camera should be positioned in order to capture as much of the HMD display as possible will generally ensure that the difference between these two will be small. We return to this issue at the end of the chapter.

## 2.2. Temporal accuracy

To recreate natural viewing conditions, the latency between a head movement and the visual consequences of the movement should be minimized. Delays at any point of the VR pipeline will result in the perception of virtual objects lagging behind a hand or head movement, as if attached by elastic. At moderate levels of lag (≈300ms) users begin to disassociate their own

movements from the VR movement [8, 13]. In the worst cases, latency can cause usability issues, including nausea [1, 10]. We consider now a procedure for measuring total system latency and discuss individual elements that contribute to this. Delays will primarily arise from the rendering system, and the tracker.

*2.2.1. Rendering latency*

Rendering typically involves reading coordinates from the tracker, positioning the virtual frustum accordingly, rendering the objects in the scene, and then swapping the freshly-rendered 'back' buffer to the 'front' buffer, which is controlled in hardware to coincide with the sending of pixel data to the display device. The rendering step must be repeated for the second display in a stereo HMD, and the whole cycle usually repeats at 60 Hz or more.

Thus, rendering time can be split into the following components: coordinate capture from the tracker; graphics pre-processing (e.g. determining what to draw based on the current coordinate); graphics rendering; swapping the 'back' buffer for the 'front' buffer (which will introduce a delay while the graphics hardware awaits the 'vertical sync' of the display device, in order to provide tear-free rendering). For a typical VR system, the HMD displays will be updated 60 times per second, which means the total rendering time must be less than 16.7ms if smooth animation is to be achieved (see figure 4, top).

Ultimately, the speed at which frames are rendered (and, hence, how smooth the VR scene moves) is determined by the swapping of the display buffers, which is the last event in the rendering cycle. This means that the tracker coordinate from which the current frame was rendered could be nearly 16.7ms old. For simple VR scenes, rendering may only take a few milliseconds (<4ms), and coordinate retrieval and processing could be less than a millisecond (measured using internal timestamps in the VR application). This would mean that there could be as much as 10ms delay introduced just waiting for the display device's vertical sync.

Dynamically monitoring the delays at each stage of the rendering cycle would allow the application software to alter an initial delay to accommodate a dynamically changing VR scene, where some frames will require more rendering time than others. To the VR user, this delayed rendering is not evident, except as reduced lag.

Specifically, the VR application software should inject time stamps into the rendering cycle, monitor how long coordinate retrieval, rendering, etc, take and introduce a small delay to the start of rendering each frame, before coordinates are retrieved from the tracker. In the above example, introducing a 10ms delay at the start of the rendering cycle means that the resulting graphics would have been rendered using a tracker coordinate that was only 6ms old, which is a significant reduction in latency (see figure 4, bottom).

For complex VR scenes, graphics rendering will consume more time, reducing the advantages obtained by this delayed rendering method. Even so, the speed of rendering of complex scenes can be improved using techniques like frustum culling, multiple-level-of-detail models, and GPGPU programming, a proper treatment of which is beyond the scope of this chapter. The reader is guided toward the multitude of real-time graphics books available.

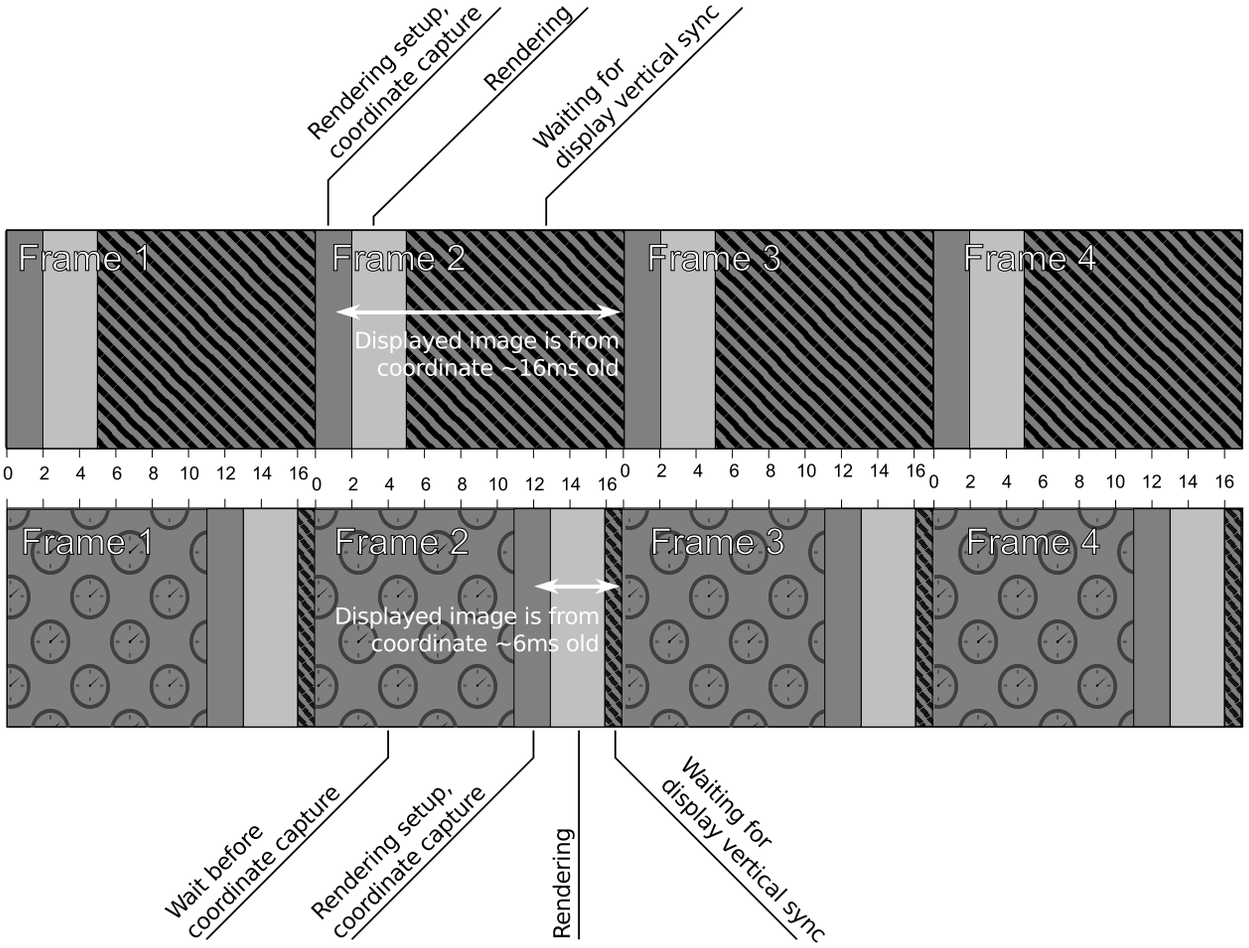


**Figure 4.** Minimizing tracker latency in a frame-rate limited VR application. Most of the computer's time is spent waiting for the display's vertical sync, which can mean that any given VR scene is from a tracked coordinate as much as 16.7ms old (top). By inserting a deliberate pause before capturing coordinates, the computer spends less time waiting for the display's Vsync, and the resulting VR scene is much younger (bottom).

### *2.2.2. Tracker latency*

The tracking system will inevitably introduce a delay between a real world event (such as a head movement) and the corresponding movement of the VR scene in the HMD display. The real world event must be sampled using, for example, video cameras, which will have their own inherent frame rate and transmission time of the captured event to the motion processor. The motion processor must interpret the incoming signal, estimate the 3D structure of the tracked objects (e.g. from multiple camera images), apply filtering to reduce noise, and package the tracking data into a suitable data protocol before sending the data down the output medium (e.g. serial or Ethernet cable) to the graphics computer. Even on the very fastest hardware, this whole procedure takes a finite amount of time, and the latency – or lag – introduced will be ultimately manifest in the HMD display as a disassociation between movement and visual scene, especially for movements with high acceleration. Indeed, tracker latency has been described as "the largest single error source" in VR systems [9, page 134].

In the following section we describe a method for measuring the overall latency of the system. Computer graphics processing is now so fast that it contributes little to the overall latency, leaving the tracker as the largest source of delays.

*2.2.3. Measuring overall latency*

It is important to be able to measure the overall latency from head movement to image change for a VR system. It is not simple to deduce this value from manufacturers' specifications as there are several steps in series with potentially important delays between each.

An easy way to measure latency is to use a standard video camera to record a movie of a tracked object being moved in a roughly sinusoidal motion, fronto-parallel to the frustum. Also within the camera's field of view, is a computer monitor showing a VR scene of the tracked object. Using image processing, it is possible to extract the 2D coordinates (in pixels) of the tracked object, and its corresponding rendering on the computer monitor. This results in two streams of sinusoidal coordinates to which a root-mean-squared error (RMS) can be assigned to indicate the difference between the two streams. Minimizing the RMS error (e.g. with a gradient descent algorithm) yields the latency between the real-world event, and the commensurate changes in the HMD display (see figure 5).

Unfortunately, we are not just measuring the latency of the tracker, but also of the communication system between tracker and graphics PC, the decoding of tracked coordinates, the rendering of the 3D model, the delay before the rendered pixels appear on the monitor display, and the delay before those pixels are captured by the camera. However, we know the refresh rate of the monitor (60Hz, in this case, so a new image every 16.7ms) and of the camera (100Hz), and the amount of time the graphics PC spends on decoding coordinates and rendering (less than 1ms). We can thus assume the mean latency added by our measurement procedure to be $16.7/2 + 10/2 + 1 \approx 12$ms. We can subtract this number from the total measured latency to obtain the delays introduced by the tracker and communication protocol.

Using this method, we have measured latencies in three different tracking systems. An acoustic/inertial system added between 15 and 45ms depending on how many tracked devices were attached. A magnetic tracker showed latencies varying between 5 and 12ms for a single tracked object. A real-time optical tracker, with nine cameras tracking a single object with 4 markers, had a mean tracker latency less than 10ms.

Note that this method is independent of any possible latency introduced by the video camera. Since all coordinates are extracted from the frames of the recorded movie, latency in the camera's own image capture, compression, and storage do not impact the subsequent analysis.

Also note that the method does not include any additional delays that may be introduced by the controller for the HMD displays. Some HMDs may render the incoming video stream directly to the HMD displays resulting in no additional delay. Other systems, particularly those using digital wireless transmission, will need to decode the incoming signal to internal video store before forwarding to the HMD displays, adding at least one frame of latency. Such delays can only be measured by repeating the above procedure with the camera mounted inside the HMD, yet still able capture images of the real world. The compact nature of most HMD displays makes this a technical challenge.

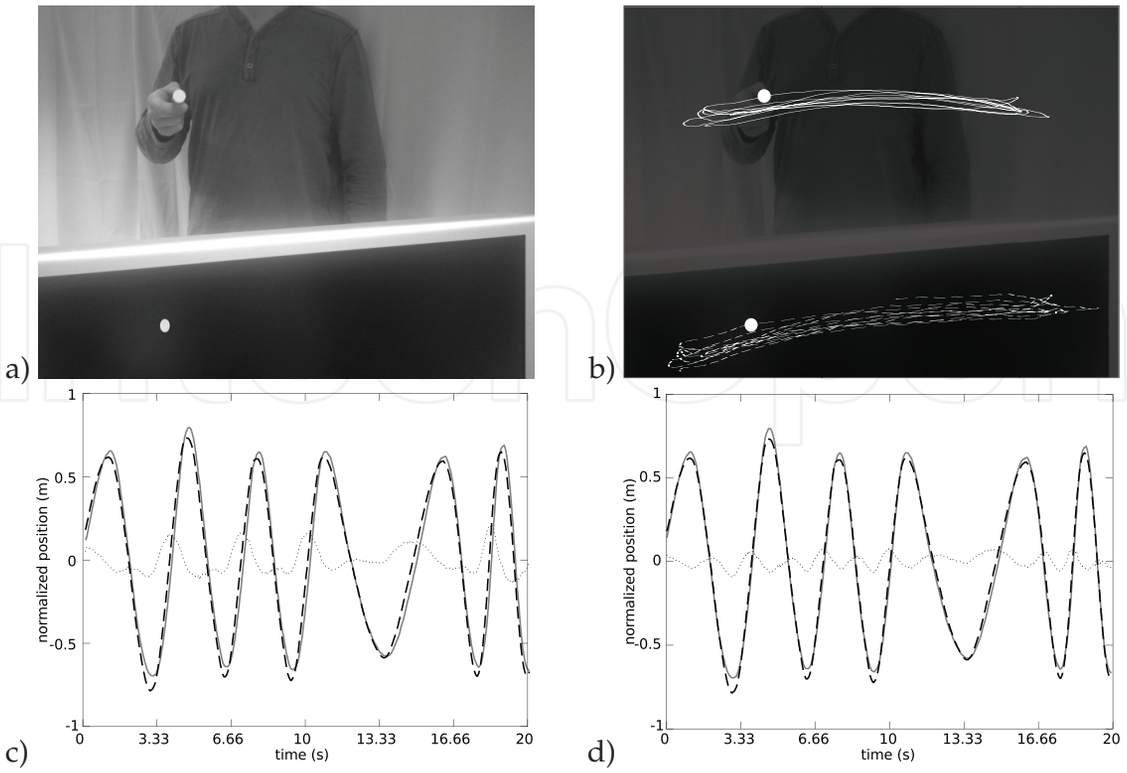


**Figure 5.** Measuring latency using a video camera to capture images of a hand-held tracked object and its VR representation on a computer monitor (a). Darkening and thresholding the image makes extraction of the centroids of the tracked object (solid line, overlaid onto camera image) and virtual object (dashed) easier (b). Plotting the lateral component of the two trajectories, aligned by camera frame, shows a small horizontal offset between the peaks (c). Finding the horizontal displacement of the tracked object which minimizes the RMS error (dotted line) between the two curves gives a measure of the latency in the system, about 60ms in this example (d).

## 2.3. Spectral

In our quest for true virtualization of light rays, we must consider the intensity and spectral composition (wavelength) of the rays, and how well modern HMD displays are able to produce such rays. Here, though, are numerous problems:

- display gamut – the displayable range of colours (gamut) of modern display technologies is significantly smaller than the gamut of the human eye.
- contrast – most modern HMD displays use some variation of liquid crystal display (LCD) technology. While these displays have many attributes making them suitable for HMDs, they do rely on back-lighting (illumination behind the crystals) to make the image visible. Unfortunately, the crystals can not be made completely opaque and will still allow some of the back-lighting to permeate through, so it is impossible to create a true black pixel. This can lead to contrast ratios (luminance of the brightest possible pixel to the darkest) as little as 100:1. At the time of writing, organic light emitting diode (OLED) displays are maturing rapidly and, as their name suggests, they emit light where needed, without the need for uniform back-lighting, and can achieve contrast ratios of 1000000:1.

- colour generation – some display technologies employ some 'tricks' to generate or improve colour appearance, but which can introduce artifacts. The authors have used a cathode-ray-tube (CRT) HMD, where the CRT itself was a greyscale unit, but a clever set of colour filters permitted colours to be displayed. While the gamut produced was on-par with other true-colour CRT systems, slight mis-alignment of the colour filters did introduce a distinct purple tinge toward the bottom of the display, which was not user-correctable. In another HMD based on liquid crystal on silicon (LCoS) technology, the display was only capable of displaying colours to 6 bits resolution per colour channel. It used a Frame Rate Control algorithm to create the appearance of more colours. While the end result was generally agreeable, such a display would be unsuitable for high-fidelity colour work.
- miniaturization – the desire to make the displays small and portable may lead to compromises in their construction, which may manifest as compromised colour fidelity.
- customer needs – For most VR users, colour fidelity is less of an issue than spatial and temporal accuracy. Manufacturers will therefore place less emphasis on colour fidelity when choosing displays for their HMDs.

Assessing the spectral properties of a conventional computer monitor would require pointing a spectrometer at a region of the display and recording the CIE 1931 $xyY$ values for a set of test colours, and computing the gamut. Performing the same procedure for a given HMD display is made somewhat harder by the limited access to the HMD displays – quality spectrometers are large and will not fit within most helmet-based HMD designs. Furthermore, modern LCD-based HMD's are likely to be more difficult to calibrate satisfactorily [23]. Even so, it maybe possible to calibrate an HMD to some extent using a visual comparison technique [2].

## 3. Applications

One of the applications of virtual reality is in vision research. Immersive virtual reality allows experimental control of the visual scene in ways that would not be possible in the real world. This ability is critical for exploring the mechanisms of human vision in freely moving observers. Free movement is, of course, the normal, general case but vision is rarely studied in this state by comparison with the intensive investigation of static-observer vision. It is not just a practical issue. Many experiments are carried out in real-world scenes with parameters such as the direction of a pointer or the distance of a comparison object being changed physically and, of course, virtual reality circumvents these laborious aspects allowing data to be gathered in much larger quantities. In addition, though, virtual reality permits stimuli that would be, to all intents and purposes, impossible to create in the real world and yet which are critical in distinguishing between hypotheses about the workings of the visual system. We shall illustrate this with examples.

Before the results of experiments that use such 'impossible' stimuli can be considered, it is vital to establish that immersive virtual reality can recreate the conditions of the real world at least to the extent that performance on the task under investigation is similar to that in the real world when the virtual scene is designed to mimic a real scene. Hence, the emphasis on precise calibration that we have discussed here and on tests of performance in a variety of tasks under conditions that simulate, in relevant respects, the real world. For example,

we have shown in our experiments that walking blind to previously-displayed objects has a similar accuracy to that in a real environment [26]; we have shown that biases in the judgment of distance [25] and object size [6, 21] are small, as is the case under real-world conditions.

Having established that participants behave as expected under 'normal' conditions, the implications of performance in manipulated environments should be taken seriously. The expanding room demonstration from our laboratory is a good example. As participants walk from one side of a virtual room to the other, the room expands by as much as fourfold but participants fail to notice that anything odd has happened (see figure 6). They believe they are in the same, static room and are amazed when told about the change in size. We have demonstrated the room to well over 100 participants and only one or two experienced stereo observers, who knew in advance about the phenomenon, reported being able to detect any change in room size. How can this be? The trick, which is only possible in virtual reality, is that the room expands about a point that is half way between the eyes (the 'cyclopean point' after the Cyclops, from Greek mythology, who had only one eye). What this means is that as each object in the room gets further away it also gets larger in such a way that the image (as seen by the cyclopean eye) remains the same. Given only the images that the cyclopean eye receives, the expanding room is indistinguishable from a normal stable room. But two cues that are usually thought of as providing reliable information about the 3D structure of the world, binocular disparity and motion parallax, are still present and should be just as effective at signalling the true size of the room as in a static room. We have investigated this phenomenon in a number of experiments [6, 21, 24, 25]. The important point here is that the isolation and manipulation of binocular disparity and motion parallax cues in this paradigm, independent of other depth cues and while still allowing observers to explore an environment freely, could not be achieved without virtual reality.

Another example from our laboratory shows how virtual reality can be used to test hypotheses about the signals people use when they are interacting with moving objects. Footballers running to head a ball could, in theory, compute the three-dimensional coordinate of football relative to their head and calculate how to move in order to intercept it. This is not considered a serious possibility in the neuroscience community; instead the argument in the literature is about what *type* of simple visual parameters might be used as control variables in determining how a player moves to intercept a ball. Many studies looking at players catching balls have analysed the curved trajectories they make as they run and fitted these using different models. However, a key test is to manipulate the variables in question during the flight of a ball and measure the effect that this has on the player's movement. We used virtual reality to do this using experienced football players as participants. Although they did not report noticing any difference between trials on which the ball travelled in a normal or manipulated trajectory, there were clear differences between the movements of players in these two conditions which could help discriminate between opposing hypotheses [16].

One heroic study by Wallach (1974) did *not* use virtual reality and serves as an illustration of what can be done by building elaborate physical apparatus instead. Wallach was interested in the mechanisms underlying visual stability. He arranged for a patterned ball to rotate so that on some trials it faced the participant as they walked while on other trials it would counter-rotate. The participant wore a headband that was attached via a series of pullies

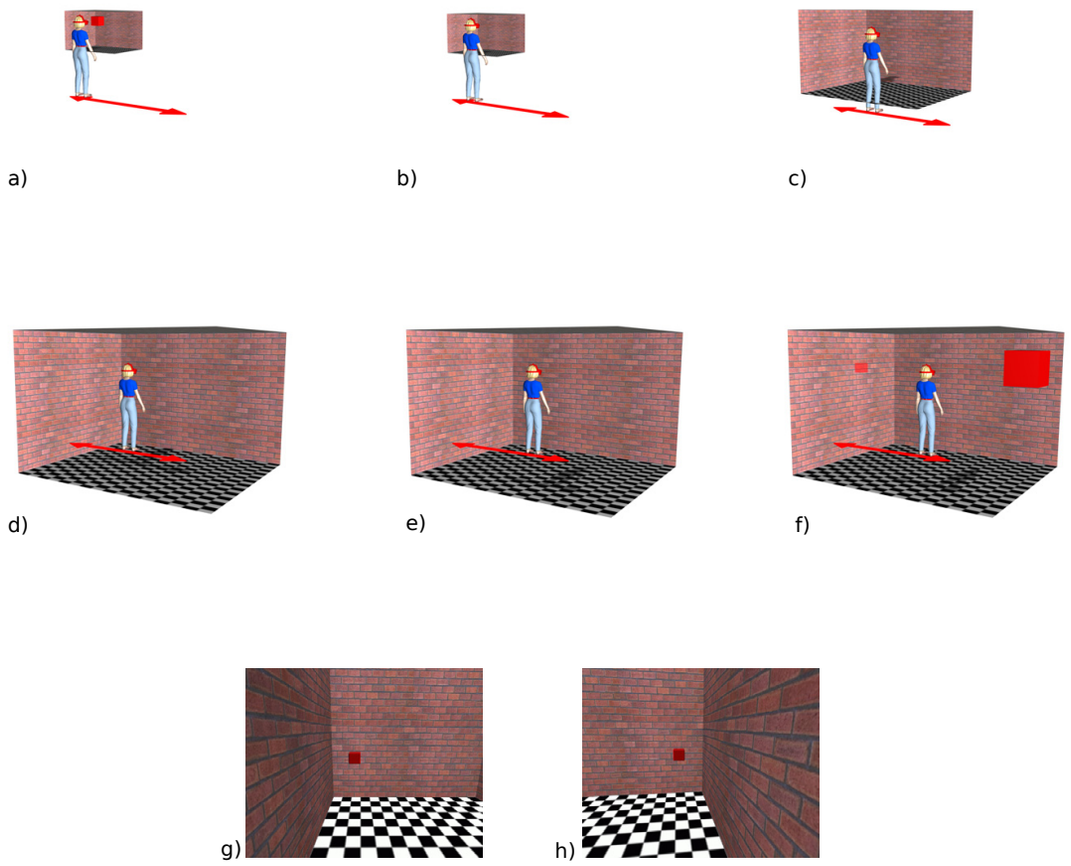


**Figure 6.** Schematic diagram of the expanding room experiment. In a), the observer views a reference cube in a 'small' room and makes some judgment of its size. b)–e) the observer walks to the right, and the room smoothly expands around their cyclopean point until it is four times bigger than the initial room. f) the observer sees another cube (possibly of a different physical size) and determines if it is bigger or smaller than the reference cube (shown semi-transparent for reader's benefit). In this example, the second cube is physically four times bigger than the reference yet, incredibly, most observers would call them the same size. g) and h) actual scenes from this trial.

to the ball, with a variable gearing affecting the ball's rotation. Generating a very similar stimulus in virtual reality, we were able to collect a significantly larger set of data and show that a static ball is reliably perceived as rotating away from them as they move (at least under these conditions, where the participant knows that the ball's rotation may be yolked to their own movement) [26]. Instead of building more elaborate mechanical devices, virtual reality allows the rotating-ball experiment to be extended to many different cases in which the visual change of a given object caused by an observer's movement (rotation, expansion, change in slant) can be manipulated independently from other features of the image (such as those that define the location of the object with respect to others in the scene). Such a systematic and comprehensive study of the parameters of visual stability has yet to be carried out, but it seems unlikely that it could be done in freely moving observers without using immersive virtual reality.

## 4. Future challenges

### 4.1. See-through augmented reality and eye tracking

We briefly touched upon augmented reality (AR) at the start of the chapter, and return to it here due to the inevitable fusion of AR and VR that is likely to take place as display technologies advance. With see-though OLED panels in early stages of development, it is likely that a "one-size-fits-all" HMD design will emerge which would be suitable for both AR (in see-though mode) and VR (in some form of black-out mode). Yet, an enduring problem with see-through HMD's is their inability to obscure rays from the real world. Creating truly opaque virtual objects is beyond current display technology.

However, such an HMD will present a new challenge which has not been met with existing video-based AR headsets, or conventional (non-see-through) VR HMDs. The challenge is that when the observer changes gaze direction, the eye does not rotate around its optic centre, but around some other point which will shift the optic centre. Thus, any static calibration (such as that obtained from our method described above) will fail to align virtual objects with their real-world counterpart. The misalignment will be worse for more extreme gaze angles. A possible solution would be to use a miniature, passive eye tracking system mounted within the HMD to track the gaze direction, report this to the graphics generating computer, which will modify the graphics frustums accordingly. Whether this can be done rapidly and unobtrusively remains to be seen.

### 4.2. High-resolution foveal rendering

While we wait for technology to catch up with our idealized VR system, and deliver the rich, high resolution scenes necessary to truly virtualize the real-world, there are ways in which the VR experience could be improved, for example using selective rendering. The human eye's fovea has a much higher spatial sensitivity than peripheral vision and, hence, rendering a high-detailed scene in regions of the HMD display that project to the eye's periphery is a waste of computational resources. Instead, eye tracking technology like that described above could be used to render a small, highly detailed sub region of the scene where the observer is fixating. The remainder of the display would comprise a low resolution rendering of the scene, which to peripheral vision is indistinguishable from highly detailed rendering. This type of technique has been used in reading research for many years [14].

## 5. Conclusions

Virtual reality is likely to have an increasingly pervasive influence on our lives in the future, driven in part by progressive advances in the realism of computer graphics and the ability, through augmented reality technology, to mix real and virtual objects seamlessly in the visual scene. Virtual reality technology still falls short of a full and accurate re-creation of the spatial, temporal and spectral properties of the rays arriving at the eye as an observer explores a real-world scene but, as we have discussed, great progress has been made towards this goal. Finally, we have emphasised the unique possibilities opened up by virtual reality for investigating the mechanisms of human vision and how, in this case, a rapid and faithful reproduction of the rays arriving at each eye is particularly important.

## Author details

Stuart Gilson
*Department of Optometry and Visual Science, Buskerud University College, Kongsberg, Norway.*

Andrew Glennerster
*School of Psychology and Clinical Language Sciences, University of Reading, Earley Gate, Reading, United Kingdom.*

## 6. References

[1] Cobb, S., Nichols, S., and Wilson, J. (1995). Health and safety implications of virtual reality: in search of an experimental methodology. In *Proc. FIVE'95 (Framework for Immersive Virtual Environments)*, University of London, London, UK.

[2] Colombo, E. and Derrington, A. (2001). Visual calibration of crt monitors. *Displays*, 22(3):87 – 95.

[3] Gilson, S. J., Fitzgibbon, A. W., and Glennerster, A. (2006). Quantitative analysis of accuracy of an inertia/acoustic 6dof tracking system in motion. *J. of Neurosci. Meth.*, 154(1-2):175–182.

[4] Gilson, S. J., Fitzgibbon, A. W., and Glennerster, A. (2008). Spatial calibration of an optical see-through head-mounted display. *J. of Neurosci. Meth.*, 173(1):140 – 146.

[5] Gilson, S. J., Fitzgibbon, A. W., and Glennerster, A. (2011). An automated calibration method for non-see-through head mounted displays. *J. of Neurosci. Meth.*, 199(2):328 – 335.

[6] Glennerster, A., Tcheang, L., Gilson, S. J., Fitzgibbon, A. W., and Parker, A. J. (2006). Humans ignore motion and stereo cues in favour of a fictional stable world. *Current Biology*, 16:428–43.

[7] Hartley, R. and Zisserman, A. (2001). *Multiple view geometry in computer vision*. Cambridge University Press, UK.

[8] Held, R. and Durlach, N. (1991). Telepresence, time delay and adaptation. In Ellis, S. R., editor, *Pictorial Communication in Virtual and Real Environments*. Taylor and Francis.

[9] Holloway, R. L. (1995). *Registration Errors in Augmented Reality Systems*. PhD thesis, University of North Carolina at Chapel Hill.

[10] Kennedy, R., Lanham, D., Drexler, J., Massey, C., and Lilienthal, M. (1995). Cybersickness in several flight simulators and VR devices: a comparison of incidences, symptom profiles, measurement techniques and suggestions for research. In *Proc. FIVE'95 (Framework for Immersive Virtual Environments)*, University of London, London, UK.

[11] Kuhl, S. A., Thompson, W. B., and Creem-Regehr, S. H. (2009). HMD calibration and its effects on distance judgments. *ACM Trans. Appl. Percept.*, 6:19:1–19:20.

[12] Land, M. F., Mennie, N., and Rusted, J. (1999). The roles of vision and eye movements in the control of activities of daily living. *Perception*, 28:1311–1328.

[13] Liu, A., Tharp, G., French, L., Lai, S., and Stark, L. (1993). Some of what one needs to know about using head-mounted displays to improve teleoperator performance. *IEEE Transactions on Robotics and Automation*, 9(5):638–648.

[14] McConkie, G. W. and Rayner, K. (1975). The span of the effective stimulus during a fixation in reading. *Perception & Psychophysics*, 17:578–586.

[15] McGarrity, E. and Tuceryan, M. (1999). A method for calibrating see-through head-mounted displays for AR. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 75–84.

[16] McLeod, P., Reed, N., Gilson, S., and Glennerster, A. (2008). How soccer players head the ball: A test of optic acceleration cancellation theory with virtual reality. *Vision Research*, 48(13):1479 – 1487.

[17] Mon-Williams, M., Plooy, A., Burgess-Limerick, R., and Wann, J. (1998). Gaze angle: a possible mechanism of visual stress in virtual reality headsets. *Ergonomics*, 41:280–285.

[18] Mon-Williams, M., Wann, J. P., and Rushton, S. (1993). Binocular vision in a virtual world: visual deficits following the wearing of a head-mounted display. *Ophthalmic & Physiological Optics*, 13:387–391.

[19] Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). Dtam: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)*.

[20] Owen, C. B., Zhou, J., Tang, A., and Xiao, F. (2004). Display-relative calibration for optical see-through head-mounted displays. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*.

[21] Rauschecker, A. M., Solomon, S. G., and Glennerster, A. (2006). Stereo and motion parallax cues in human 3d vision: Can they vanish without trace? *Journal of Vision*, 6:1471–1485.

[22] Regan, C. (1995). An investigation into nausea and other side-effects of head-coupled immersive virtual reality. *Virtual Reality*, 1:17–31. 10.1007/BF02009710.

[23] Sharma, G. (2002). LCDs versus CRTs – colour-calibration and gamut considerations. In *Proceedings of the IEEE*, volume 90.

[24] Svarverud, E., Gilson, S. J., and Glennerster, A. (2010). Cue combination for 3D location judgements. *Journal of Vision*, 10(1).

[25] Svarverud, E., Gilson, S. J., and Glennerster, A. (2012). A demonstration of 'broken' visual space (in press). *PLoS ONE*.

[26] Tcheang, L., Gilson, S. J., and Glennerster, A. (2005). Systematic distortions of perceptual stability investigated using immersive virtual reality. *Vision Res.*, 45:2177–2189.

[27] Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3-D machine vision. In *IEEE Computer Vision and Pattern Recognition or CVPR*, pages 364–374.

[28] Tuceryan, M., Genc, Y., and Navab, N. (2002). Single point active alignment method (SPAAM) for optical see-through HMD calibration for augmented reality. *Presence-Teleop. Virt.*, 11:259–276.

[29] Turing, A. M. (1950). I. – Computing Machinery and Intelligence. *Mind*, LIX(236): 433–460.

[30] Wallach, H., Stanton, L., and Becker, D. (1974). The compensation lot movement-produced changes of object orientation. *Perception & Psychophysics*, 15:339–343.