Towards Achieving Convincing Live Interaction in a Mixed Reality Environment for Television Studios

- Gregory Hough

DMTLab

Faculty of Computing, Engineering and the Built Environment

Birmingham City University

Submitted for the Degree of Doctor of Philosophy

October 2015

ABSTRACT

The virtual studio is a form of Mixed Reality environment for creating television programmes, where the (real) actor appears to exist within an entirely virtual set. The work presented in this thesis evaluates the routes required towards developing a virtual studio that extends from current architectures in allowing realistic interactions between the actor and the virtual set in real-time. The methodologies and framework presented in this thesis is intended to support future work in this domain.

Heuristic investigation is offered as a framework to analyse and provide the requirements for developing interaction within a virtual studio. In this framework a group of experts participate in case study scenarios to generate a list of requirements that guide future development of the technology. It is also concluded that this method could be used in a cyclical manner to further refine systems post-development.

This leads to the development of three key areas. Firstly a feedback system is presented, which tracks actor head motion within the studio and provides dynamic visual feedback relative to their current gaze location. Secondly a real-time actor/virtual set occlusion system that uses skeletal tracking data and depth information to change the relative location of virtual set elements dynamically is developed. Finally an interaction system is presented that facilitates real-time interaction between an actor and the virtual set objects, providing both single handed and bimanual interactions.

Evaluation of this system highlights some common errors in mixed reality interaction, notably those arising from inaccurate hand placement when actors perform bimanual interactions. A novel two stage framework is presented that measures the magnitude of the errors in actor hand placement, and also, the perceived fidelity of the interaction from a third person viewer.

The first stage of this framework quantifies the actor motion errors while completing a series of interaction tasks under varying controls. The second stage uses examples of these errors to measure the perceptual tolerance of a third person when viewing interaction errors in the end broadcast.

The results from this two stage evaluation lead to the development of three methods for mitigating the actor errors, with each evaluated against its ability to aid in the visual fidelity of the interaction. It was discovered that the adapting the size of the virtual object was effective in improving the quality of the interaction, whereas adapting the colour of any exposed background did not have any apparent effects. Finally a set of guidelines based on these findings is provided to recommend appropriate solutions that can be applied for allowing interaction within live virtual studio environments that can easily be adapted for other mixed reality systems.

TABLE OF CONTENTS

TABLE OF ACRONYMS

Acronym	Term
CPU	Central Processing Unit
DSCQE	Double Stimulus Continuous Quality Evaluation
DSIS	Double Stimulus Impairment Scale
GPU	Graphical Processing Unit
ITU	International Telecommunication Union
IVS	Interactive Virtual Studio
MDOS	Mean Distance to Object Surface
MOS	Mean Opinion Score
NPR	Non-Photorealistic Rendering
POV	Point of View
QoE	Quality of Experience
RGB	Red, Green, Blue
ScaMP	Scanning Mirror Projector
SS	Single Stimulus
SSCQE	Single Stimulus Continuous Quality Evaluation
SSMR	Single Stimulus with Multiple Repetitions
VDBH	Variability in the Distance Between Hands

TABLE OF FIGURES

INDEX OF TABLES

Acknowledgments

The completion of my dissertation and subsequent Ph.D. has been a long journey. I would like to thank those who have contributed along the way.

First of all, I want to say a massive thank you to my PhD supervisor Professor Cham Athwal and my Director of Studies Dr Ian Williams. Both have helped me immeasurably throughout with their tireless encouragement and excellent advice, from editing academic papers in intense situations to helping me out in my personal life.

Furthermore, thank you to all my other colleagues in DMTLab who have all contributed in some way over the years, especially Alan Dolhasz, Muadh Al Kalbani, Sean Enderby who have particularly been a source of great ideas.

I would like to thank my family, in particular my parents Stephen Hough and Pauline Hough for the amount of patience and support they've given me, without whom I would not have been able to complete this thesis.

And finally thank you to my friends, particularly Mark Williams, Paul Butler and Natalie Shoham who all offered valuable support, comforting words and timely distractions throughout my studies.

Chapter 1 : INTRODUCTION

1.1. Motivation

Suppose you want to present a live television show from space or from a seabed - the logistical difficulties of making such a television show are enormous, practically making it impossible. Now imagine creating a method that would allow you to present television shows from these unimaginable and exotic locations from the comfort of a television studio. That method is known as the virtual studio (Shimode, et al., 1989) (Blonde, et al., 1996), which was developed in the late 1980's and widely adopted in the early 1990's. In this studio a real actor is able to appear inside an entirely virtual environment, which is broadcast live. Although the virtual studio offers many advantages over regular television studios, it is often hindered by the lack of interaction between the actor and the virtual set. Interaction is the manipulation of a virtual object by an actor, which can range from abstract manipulation using a remote console (low level interaction) to direct manipulation similar to how one would manipulate a real object (high level interaction).

In the virtual studio an actor stands in front of a blue or green screen, which is captured by the studio camera. Using a process known as chromakey the blue or green is removed from the camera's image and replaced by a fully 3D virtual environment. This is done to achieve continuous live broadcasting from a single studio where entire sets can be changed at the click of a button during an advert break, or to produce large or unreal environments that would not be possible in regular television studios.

The earliest methods of achieving interaction in the virtual studio used simple buttons that would be pressed to trigger some event, such as an animation. These devices are typically seen in weather broadcasts, where the presenter presses a button and the weather map progresses a few hours. Gradually this approach has evolved into more complex systems, where the actor has a tablet PC or touchscreen that presents multiple options to control individual elements of the scene. Whilst these approaches allow some control over the virtual objects, they still appear unrealistic because the actor is not touching and moving the object as they would in real life, creating a disconnected feeling.

Virtual studios are still typically limited to these low-level interaction techniques. The successful enactment of high-level and plausible interactions, akin to an interaction one would observe between a person and an object in real life, has so far been elusive.

From a review of the literature it is found that the three main limitations for achieving realistic interaction in the virtual studio are:

- The lack of sufficient feedback to the actor.
- The lack of a reliable approach towards achieving visual occlusion between actors and virtual objects.
- The lack of a sufficiently realistic method for interaction between the actor and the virtual set.

Overcoming these interaction limitations of virtual studio technology would allow the range of applications that it is capable of to be radically extended. While attempts have been made to construct virtual studio systems that offer high-level interaction between actor and virtual object, they have been met with limited success and acceptance in both commercial and academic domains. We believe the reason is that these systems have been developed without addressing the issues facing them at a fundamental level, where the requirements of the technology and the human physiological factors have not yet been fully explored. These three limitations are further bound by the need to use imperceptible actor motion capture. This is required to conceal the workings of the system from the viewer and maintain their 'suspense of disbelief' (Zerroug, et al., 2009).

This serves as justification for this work to address the issues of interaction in the virtual studio at a more fundamental level, with the goal of achieving realistic interaction with virtual objects in real time.

This provides the research aim of this thesis, which is *to develop and test a framework for analysing the requirements towards developing realistic appearing interactions the virtual studio*.

Two approaches are taken towards achieving this aim:

The first is a heuristic evaluation surrounding hardware and software design of the virtual studio, with the evaluation conducted for three key elements: Visual Feedback, Real-Time Occlusion and Interaction. In addition, the motion capture system is also selected using heuristic methods (discussed in Appendix #A), although this is not discussed here as it is not a novel development in itself. From this evaluation the key requirements are described and systems compatible with these are developed. To demonstrate their transferability to current virtual studio designs these systems are applied to a current standard virtual studio, transforming it into an interactive virtual studio.

The second is development of a novel framework that evaluates the visual impact of motion errors created by actors in the virtual studio in two stages to improve the *apparent* actor performance. In the first stage the motion of a group of actors as they complete a series of bimanual interaction tasks under varying conditions is quantified. In the second stage the errors made by the actors are replicated in a series of videos, which are presented to a group of observers who rate how visually credible the interaction was, allowing measurement of how adept the average viewer is at spotting particular errors.

This method also allowed scene manipulations that can serve to diminish the perceived magnitude of the errors to be assessed. The results from the first and second stage are then compared to inform future design decisions that can be used to improve the quality of interaction in the virtual studio.

1.2. Outline of Approach

Figure 1-1 depicts the theoretical framework of the work presented in this thesis, which shows each area of research and the relationship between them. The actor represents the point of the heuristic evaluation, the findings of which are used to suggest requirements for the development of the hardware and software design stages. The framework to identify perceptible errors is then used to assess the level of interaction realism that can be achieved and suggest solutions that improve the apparent interaction quality, which feed back into the development stage. These stages are discussed in detail throughout sections 1.2.1 and 1.2.2.

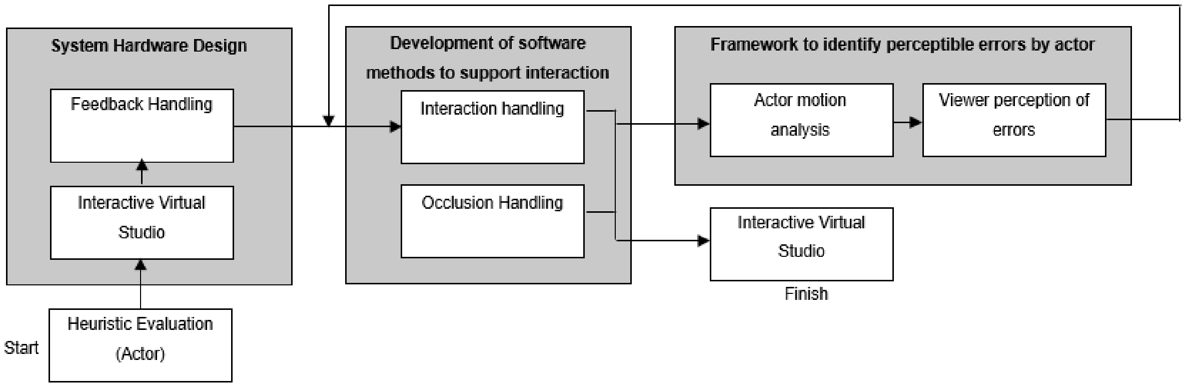

System Hardware Design
Feedback Handling
Interactive Virtual Studio
Heuristic Evaluation (Actor)
Start
Development of software methods to support interaction
Interaction handling
Occlusion Handling
Framework to identify perceptible errors by actor
Actor motion analysis
Viewer perception of errors
Interactive Virtual Studio
Finish

Figure 1-1. Theoretical framework

1.2.1. Heuristic Evaluation and the Development of Methods to Support Interaction

To define the current limitations of actor interaction in the virtual studio a heuristic investigation (Nielsen & Molich, 1990) is conducted, which is an informal method of identifying common issues with user interfaces and allowing the focussed design of appropriate solutions. In this method experts participate in a series of tasks they want their system to perform and describe the problems that are encountered using a set of heuristics, which can then be used as a base for making design decisions. In our study the experts participated in the scenario of a dental training tele-lecture, where they are required to instruct remote screen viewing students on the anatomy of teeth.

The heuristic analysis led to design decisions in two key areas, hardware design and software design. Hardware design encompasses the architecture of the virtual studio and the development/selection of motion capture and feedback methods. The software design encompasses the developments made for the occlusion and interaction systems.

Hardware Design

Interactive Virtual Studio. The virtual studio architecture is described - covering where the interactive functions (occlusion and interaction) are positioned within the system, the justification for the actor motion capture system selection and the manner in which the locations of all elements in the studio are described. The feedback system developed for this work is designed to be independent of the interactive virtual studio and so is not formally included in this architecture.

Feedback. In the context of this thesis feedback is defined as any method that provides the actor with information on the virtual objects around them in a manner appropriate for interaction (e.g. informs them on the location of nearby object surfaces). Providing feedback to the actor would be used to enhance their ability to identify the location or surfaces of virtual objects, allowing them to place their hands close for interaction. In this work a device is developed that provides ubiquitous visual feedback to the actor by guiding a projection to the gaze location of the actor. This device is formed of a projector with a servo guided mirror mounted in front of the lens alongside a camera-based head tracking system. In addition the projected image is corrected for any warping that may occur in real time and three modalities of feedback are provided to the actor.

Software Design

Occlusion. In the context of this thesis occlusion is defined as the visual blocking of one object by another. In a virtual studio domain, this specifically describes the blocking between real and virtual objects. Providing a real time occlusion method for the virtual studio will enhance the realism of the scene, particularly when the actor is required to interact with the object.

In this work a system is implemented to achieve a range of occlusions in a standard virtual studio environment. Three forms of occlusion that could occur are defined and methods for implementing each of them into a layer-based virtual studio are provided.

Interaction. In the context of this thesis interaction is defined as the ability of the actor to directly manipulate a virtual object, ideally where the actor places their hands on the surface of the virtual object to move it. This ideal interaction is comparable to how one would appear to move a real object. This style of direct manipulation of a virtual object would appear more plausible to an audience than the less direct methods that already exist. In this system the imperceptible motion capture system is used to create methods of producing triggered, single-handed and bimanual interaction techniques.

1.2.2. Novel Framework to Identify Perceptible Errors by Actor

One of the key issues with interaction in the virtual studio is that a bimanual interaction that would be simple with a real object becomes a complex task for an actor to complete with a virtual object. Estimating the surfaces of the virtual object itself poses a difficult challenge when no solid surface exists. The issues caused by the lack of sufficient surface feedback not only affected the performance of the actor, but also the plausibility of the interaction from the perspective of the viewer, who would negatively perceive the misestimation. A novel two-stage framework is designed that allowed the measurement of errors made by the actor and an assessment of how these would be perceived by the viewer.

Stage 1: Actor Motion Analysis. In the first stage the types of estimation errors that the actor is likely to make during a bimanual interaction with a virtual object are defined and then quantified. This involved the measurement of the performance of 16 actors who completed a series of 168 bimanual interaction tasks. Each interaction task contains a permutation of the following conditions: Size of the Virtual Object, Speed of the Virtual Object, Axis of Object Motion, Axis of Hand Placement and Direction of Interaction. Two performance metrics are presented that allow the misestimation of the object size and the amount of variability between the actor's hands to be measured. From the analysis of the results conclusions are drawn on how each of these conditions affects the motion of the actor. The results indicate that the size of the virtual object and the placement of the actor's hands with regards to the axis of motion both have a significant impact on the performance of the actor.

Alongside bimanual interaction the actor is also presented with two alternative interaction modalities. The first is an 'animated' modality, where the actor follows the path of an animated virtual object. This is included to compare interaction with a current standard technology (Gibbs & Baudisch, 1996), where the actor follows an animated virtual object. The second is a 'no-object' modality, where the actor mimes an interaction with a virtual object, typical in post-production (where the graphics are added to the video at a later stage. These are included to determine whether an interactive virtual studio could also be useful as a tool for aiding the constraint of actor motion in a blue screen studio, where virtual objects will be added in post-production (e.g. for film production). The results demonstrated that interactive modality yields a superior level of actor performance to the no-object modality and a similar level of performance to the animated modality, except without the lag/lead error (where the actor fails to correctly estimate and maintain the same velocity as the virtual object).

Stage 2: Viewer Perception of Errors. The second stage is an analysis of viewer's ability to perceive the estimation errors made by the actor. In this study a series of videos that replicate the errors created by the actor from the Stage 1 interactive tasks are presented to a group of observers for rating, allowing a profile of perceptible errors to be constructed.

As well as containing the replicated motion errors, some videos included in the presentation contained manipulations to the scene to determine whether they could mitigate the perception of estimation errors. These manipulations included changing the size of the virtual object to match the distance between the actor’s hands and changing the colour of the gap that appears between the actor’s hand and the surface of the virtual object in the case of an overestimation. In addition, this method is also used to test the effect that occlusions inconsistent with real life have on scene plausibility.

1.3. Contributions

From the study described the following contributions are made:

1. Heuristic evaluation (chapter 3) of issues affecting feedback (chapter 4.3), occlusion (chapter 5) and interaction (chapter 6), leading to the development of novel solutions for each of these areas that are compatible with our findings.

   These developments fit within an existing virtual studio architecture typical of one used in the industry, allowing the implementation of interaction between the real actor and virtual object in a manner that can be widely applied to comparable systems. They allowed the actor to interact with the virtual elements of the scene in a simple manner.

2. Novel framework for analysing the impact of common errors associated between real and virtual elements in a mixed reality environment; in this case applied to interaction the virtual studio.
   a. Methods for classifying and quantifying actor motion errors in bimanual interactions with virtual objects in the virtual studio (chapter 7).
   b. Method for analysing the viewer perception of errors specified in the motion capture study (chapter 8). It is also demonstrated that this technique can be used for identifying effective methods for mitigating the effects of errors in bimanual interactions.

   Together the results from this framework are effective in informing on the impact that actor motion errors have on the plausibility of the scene from the perspective of a TV viewer. It also allowed solutions based on manipulating properties of the real and virtual elements of the scene to be tested, allowing the perception of the actor motion errors to be mitigated.

Publications

Hough, G., Williams, I. & Athwal, C., 2014. Measurements of Live Actor Motion in Mixed Reality Interaction. IEEE International Symposium on Mixed and Augmented Reality, pp. 99-104.[1]

Hough, G., Williams, I. & Athwal, C., 2014. Measurement of Perceptual Tolerance for Inconsistencies within Mixed Reality Scenes. IEEE International Symposium on Mixed and Augmented Reality, pp. 343-344.

[1] This paper was nominated for the best short paper award at ISMAR 2014 and subsequently an extended version has been requested for the IEEE Transactions on Visualization and Computer Graphics (TVCG).

Hough, G., Athwal, C. & Williams, I., 2012. Advanced occlusion handling for virtual studios. Convergence and Hybrid Information Technology, Lecture Notes in Computer Science, Volume 7425, pp. 287-294.

Hough, G., Athwal, C. & Williams, I., 2012. ScaMP: A Head Guided Projection System. ACM Designing Interactive Systems 2012.

Workshops

The framework to identify perceptible errors presented in this thesis has been included as part of a workshop for the upcoming IEEE ISMAR 2015 on analysing the Quality of Experience and plausibility of Mixed Reality scenes, titled “Measuring Perception and Realism in Mixed and Augmented Reality”.

Chapter 2 : LITERATURE REVIEW

2.1. From the Movies to the Virtual Studio

In 1878 photographer Eadweard Muybridge was commissioned by Leland Stanford, an industrialist and fanatic of horses, to analyse the gait of a galloping horse (The Museum of the City of San Francisco, 2013). Stanford wanted to know whether a horse lifts all four of its feet completely off the ground at any one time during a gallop cycle. To answer this question, Muybridge set up a line of 24 cameras that each took a single photograph as a horse galloped past them, hoping that one of the cameras would capture the horse with all four feet off the ground. Muybridge was successful in proving that a galloping horse lifted all four feet off the ground. However, the interesting result of this experiment was that when the 24 photos were shown in quick succession over a 3 second period they could not be distinguished from one another, instead appearing like a single photograph where the horse appeared to move. Muybridge had inadvertently invented the moving photographed image, otherwise known as a "video image" or "movie" (a contraction of 'Moving Image').

Muybridge later met with Thomas Edison, who was inspired to build the world's first device that captured images in quick succession to create these movies, later known as the 'movie camera' (henceforth camera). The images captured from this camera could be projected sequentially onto a projection screen at the same rate, allowing an audience to see what they would believe is a moving photograph (Edison, 1891).

These innovations and the initial curiosity of seeing a moving image meant that movies quickly gained popularity as a form of entertainment. Initially they were shown as short features less than a minute long as a novelty at fairgrounds, but by the end of the century they had evolved into longer features that were shown in theatres and ballrooms. In the early 20th century movies had become a form of mass entertainment and as demand grew specially built movie theatres began to appear and longer, more complex movies were made.

With demand for movies growing, alternative distribution avenues were sought. One avenue looked at developing methods of transmitting them straight into the homes of the consumer. In 1926 John Logie Baird produced the first live transmission of a video image via radio signals (Baird, 1929). The device used to display the images was formed of a rotating disc with a spiral of lenses through which light could be projected onto a photo-sensitive Selenium screen, creating an image. Over the following years televisions grew more sophisticated and practical, eventually being replaced by the cathode ray tube televisions (Farnsworth, 1927) that became a fixture for the remainder of the 20th century. By the 1950s televisions had found their way into the homes of millions of people worldwide, allowing them to watch movies from the comfort of their living room.

One of the key advantages that television had over the movie theatres was that it did not require a "print" of the film to be made, as video images could be captured by a camera and transmitted to televisions live. This allowed events such as horseracing and football matches to be broadcast directly into the viewer's home. One application that was quickly found for live TV was transmitting television shows such as news and weather broadcasts straight from a television studio, allowing events to be reported on in real time - something that was not possible in the movie theatres. This was a big advantage for the television, as it became the first method of visually reporting on events as they were happening.

Meanwhile, the movie industry was also moving fast, in both size and technological achievements. Films were becoming larger in scope and the size of the productions was increasing to capture larger audiences. Soon a market developed where people went to see films purely for the "spectacle"; where incredible stunts, exciting action and amazing special effects entertained them.

One of the developments that became a staple in producing special effects for films was chromakey, a method of removing an arbitrary colour, known as the 'key colour', from the background of a video and replacing it with another image. Typically the key colour is either blue or green as these are the colours that exist furthest away from red in the colour spectrum, the primary constituent colour of skin tones. With chromakey an actor will stand in front of a green screen and be captured by the camera, then the green is removed from the camera image and replaced with an image of some other location. This has the effect of allowing actors to appear as though they are present in some exotic location, without having to leave the film studio. A basic example of this process is shown in Figure 2-1 where an image of an actor in front of a removed blue screen is overlaid onto the image of a beach.

The technology was first demonstrated in the film "The thief of Baghdad" from 1940 (The Thief of Baghdad, 1940). In this film a flying carpet effect was achieved by filming a person standing on a carpet against a blue background. The blue background was removed and replaced by footage of the sky, creating the effect of the carpet flying in the sky. Figure 2-2 shows an image of the flying carpet effect from this film. This innovation earned the Thief of Baghdad the Academy Award for special effects in 1941.

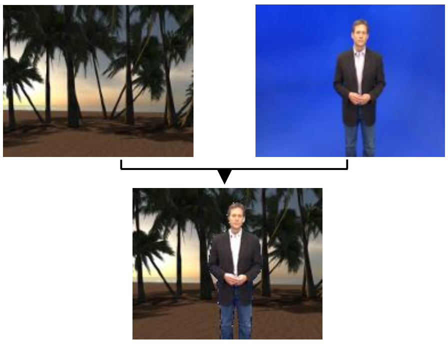


Figure 2-1. Compositing two images together using Chromakey

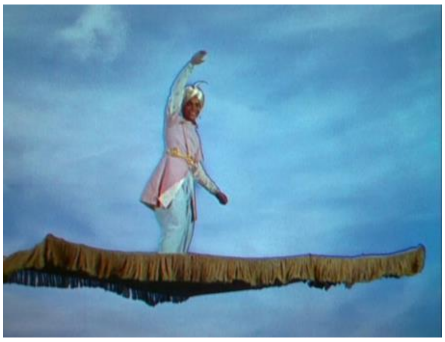


Figure 2-2. An image from the flying carpet scene in "The thief of Baghdad", 1940, the first use of chromakey (The Thief of Baghdad, 1940)

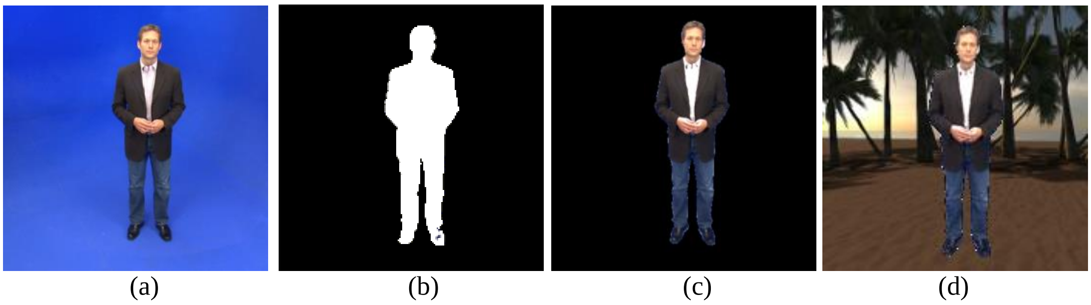


Figure 2-3. Digital chromakey method. (a) Original digital image. (b) Alpha matte. (c) Final matte. (d) Final composite with actor keyed in over a background

During this time film stock was primarily used for recording movies, where chromakey involved a process of re-photographing film stock multiple times. Since then film making has moved into the digital age and the process of chromakey has had to evolve to meet modern technological standards. In the digital realm, chromakey is based purely on the hue of the image, with the luminosity and saturation of the image disregarded. Here, chromakey is accomplished by producing an ‘alpha matte’, which is a black and white image where the key colour is indicated by black pixels and the actor is indicated by white pixels. An example of an alpha matte is shown in Figure 2-3b, where it is used to describe which pixels should be visible in the final matte (Figure 2-3c).

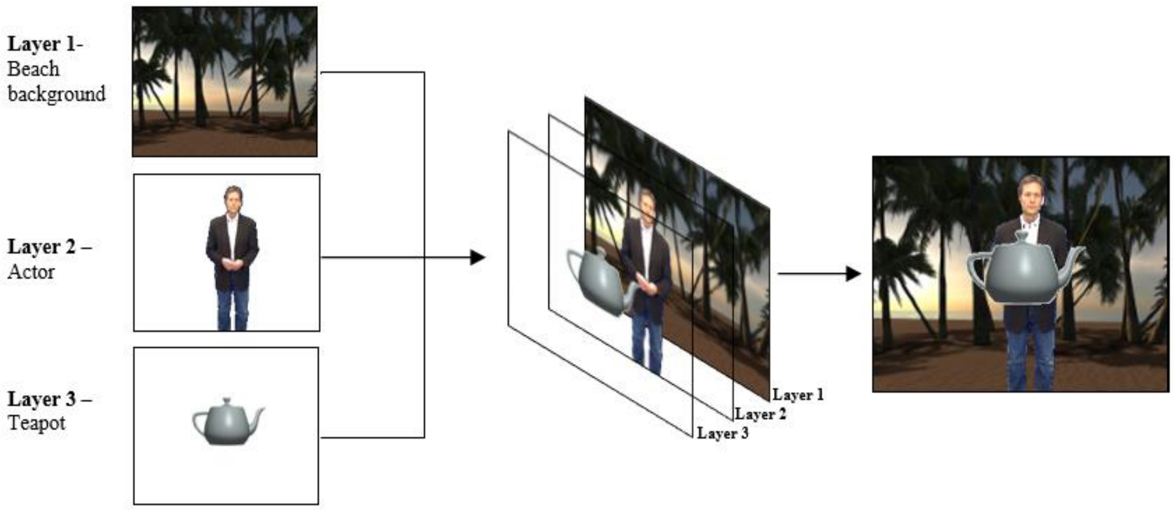
The alpha matte is constructed using Equation 2-1, which is conducted for each pixel in the image. Here M_{pixel} describes a single pixel in the alpha matte, H_{key} is the desired value of the colour to be keyed out (described using Red, Green and Blue colour channels), H_{pixel} is the value of the pixel in the original image and T is a tolerance value. If H_{pixel} is within the tolerance range, then it is set to 1, describing the pixel as part of the blue background; if not, then the pixel is set to 0.

$$M_{Pixel} = \{ \begin{matrix} 1\ if\ (H_{Key} - T\ ) < \ H_{Pixel} < \ (H_{Key} +\ T) \\ else\ 0 \end{matrix}$$

Equation 2-1. Digital chromakey algorithm for Alpha Matte

One of the key advantages of digital chromakey is that the final matte can be applied to footage that has been uploaded onto a computer to be used in “Digital compositing”. Digital compositing is a method of creating scenes from many different elements for post-production. Here different images, real or virtual, are assembled together using a computer to produce a single image. For example, in Figure 2-3d, the key image of the actor from Figure 2-3c has been overlaid on an image of a virtual beach, causing the actor to appear as though they are in that location.

For digital compositing in movies a system of layers is typically used to produce this effect, where each layer containing certain elements of the scene are rendered on top of the previous layer until they produce the final image. This process is shown in Figure 2-4 (page 11), where Figure 2-4a shows the individual scene elements arranged into layers (which in this case are a beach scene, an actor (final matte) and a teapot (transparent background)); Figure 2-4b shows the 3 scene elements layered on top of each other (as seen from an arbitrary angle), with layer 1 (beach) appearing in the background and layer 3 (teapot) appearing in the foreground; Figure 2-4c shows the final composited image.



(a) Scene Elements (b) Organisation of layers (c) Final Output

Figure 2-4. Example of Digital Compositing, where the scene elements (a) are presented as a sequence of layers (b) rendered on top of each other to produce the final output of a man standing on a beach with a teapot (c).

Digital compositing has become ubiquitous in modern film-making, as the streamlined process allows for faster and more versatile composition of video images than was previously possible. One particular area that has benefitted greatly from digital compositing is the inclusion of Computer Generated Images (CGI) in scenes.

The significance of using chromakey in digital compositing began to expand when computers could be used to create CGI for films. With CGI, fully virtual objects are created and placed into a scene alongside real actors. Although a limited amount of CGI was used in films as early as 1973 (Westworld, 1973), it came to prominence in the early 90s with the films Terminator 2: Judgement Day (Terminator 2: Judgement Day, 1991) and Jurassic Park (Jurassic Park, 1993), which had virtual characters (designed and animated entirely using CGI) that appeared alongside real actors throughout the film.

Compositing these effects together is done in an offline process known as “post production”. When producing scenes that include CGI the visual effects artist will have time to design and animate the virtual objects in a manner that is optimised to the real scene elements and the actor. If the scene calls for the actor to move a virtual object, the actor mimes the interaction in the studio and the visual effects artist will be able to map the virtual object to the movements of their hands in the digital compositing stage. The actor does not require their movement to be entirely accurate, as the size and shape of the virtual object can be designed and adjusted to fit their estimations. If accuracy is required from the actor, then their motion can be constrained and multiple takes can be recorded with only the best one being used.

The film Iron Man 2 (Iron Man 2, 2010) has many examples of real actors interacting with virtual objects. In one particular scene Robert Downey jr interacts with a holographic 3D User Interface, where he appears to manipulate the various holographic elements with his bare hands.

Figure 2-5 presents a section of this scene as an example, where Robert Downey jr picks up a virtual globe and pulls it towards him while expanding its size. To produce this effect Robert Downey jr mimed the scripted interaction with the virtual object, placing his hands where he believed the sides of the globe would be, moved his hands towards him and increased the distance between them. The visual effects artist would then analyse the raw footage of this mimed interaction and track the motion of a single point on each of Robert Downey jr's hands, a task which is supported by motion tracking tools available in most compositing software packages that support CGI. The globe is then locked to the midpoint of these tracked points and moves with them synchronously, with the size of the object scaling accordingly to match the change in distance between them.

Although tools exist for automating the various stages of this process, producing this interaction is still a time consuming task that can only be done after the scene has been recorded and requires the visual effects artist to ensure that the motion tracking is accurate and that the virtual object is moving in a plausible manner. Despite these time constraints, the spectacle that can be created using these visual effects methods can provide a more immersive experience for the audience.

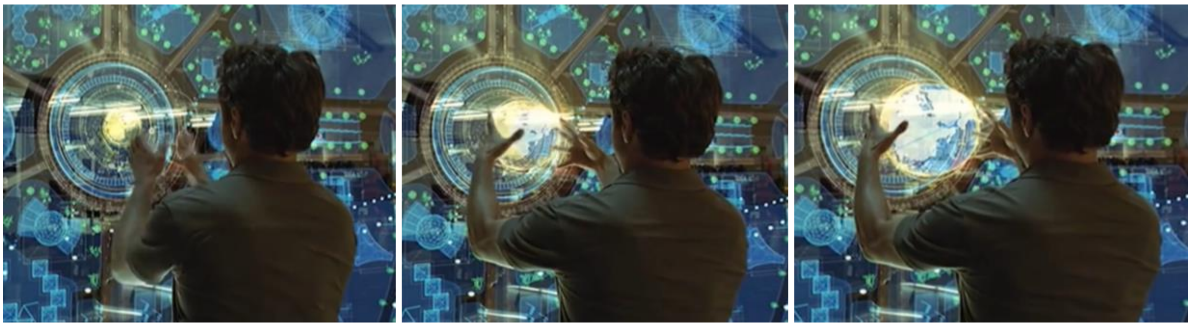


Figure 2-5. Scene from Iron Man 2, where Robert Downey jr interacts with a holographic globe using effects achieved by digital compositing and CGI **(Iron Man 2, 2010)**.

The combination of chromakey, CGI and digital compositing significantly benefitted the production of movies by allowing the range of possible effects to be extended drastically, adding a significant advantage over live television. Shortly after its proven success in the film industry, similar efforts were made to introduce these technologies into live television (e.g. (Blonde, et al., 1996)). The result is the virtual studio, a type of television studio that allows productions where the actor appears to exist in a television set formed entirely of virtual objects in real time. Figure 2-8 shows this process in the virtual studio, where the actor stands in front of a blue screen and is keyed into a virtual room. Figure 2-7 shows a photo of the virtual studio used in this body of work. The virtual studio has three key features that differ from standard television studios and film production sets:

1. Live digital chromakey capabilities.
2. A dedicated Graphics Processor Unit (GPU) for handling real-time rendering of the virtual set and compositing of real and virtual scene elements.
3. A tracked studio camera whose position and orientation in 6 Degrees of freedom (DOF - see Figure 2-6) is matched 1:1 with a virtual camera inside the virtual set.

Figure 2-9 (page 14) presents a diagram that illustrates the functionality of our virtual studio. There are three key elements in the studio, the actor, the studio camera and the blue screen. The camera captures the video image of the Actor standing in front of the blue screen, which is used for two purposes: using the captured image for chromakey and analysing the unique pattern of the blue screen to calculate the location of the camera. The latter is used for by the GPU to render the appropriate perspective of the virtual set.

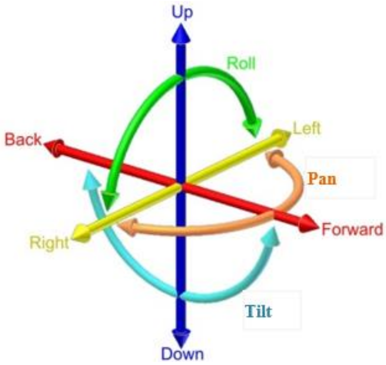


Figure 2-6. The 6 Degrees of Freedom (6DOF) for axes of motion and orientation. For describing movement, the 'Z' axis refers to backwards/forwards movement, the 'Y' axis refers to up/down movement and the 'X' axis refers to left/right movement. Roll, Pan and Tilt refer to axes of orientation.

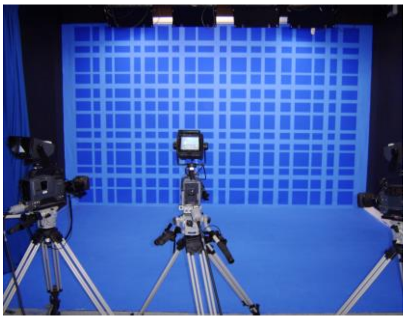


Figure 2-7. Example of a virtual studio

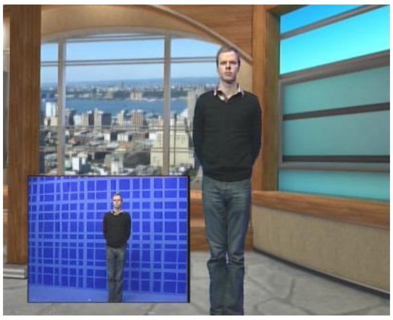


Figure 2-8. An example of a virtual studio production. By using digital compositing and CGI the actor, who is really standing against a blue background, appears to be in a room**.**

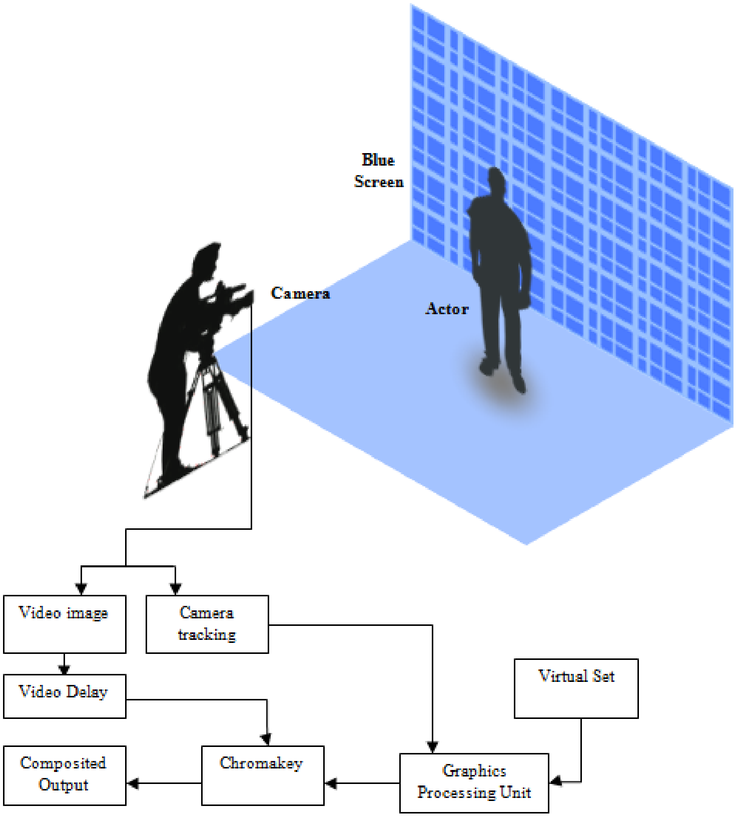


Figure 2-9. Diagram of the virtual studio architecture

Tracking the motion of the studio camera is essential, because when computer generated images are inserted into a scene the correct aspect must be maintained between the real and the virtual sets. If the studio camera moves then the perspective of the virtual set should update accordingly, else the set will remain static as the actor will appear to rotate, which would appear implausible to viewers.

The background of Figure 2-7 (page 13) contains an example of a 'tracking grid', a unique pattern formed of two shades of blue that can be removed during the digital chromakey process when a sufficient tolerance is defined. The portion of the tracking grid captured by the studio is compared to a reference image using pattern recognition, which is then used to calculate the distance and orientation of the camera and the zoom of the camera lens. Other common camera tracking techniques for the virtual studio are typically sensor based, or optical based. Approaches include mechanical sensor tracking and camera mounted infrared markers (both overviewed in (Orad, 2010)), ceiling based marker systems (Thomas, et al., 1997), using natural features in the scene (such as the BBC's MATRIS (Chandaria, et al., 2007)), and SLAM based systems (Yang, et al., 2008).

A dedicated GPU is used to render the graphics. Rendering high quality graphics is a resource intensive challenge, especially for the live conditions of the virtual studio where the render must occur within 3 to 4 frames. The GPU is specifically designed for the quick render of graphics and compositing; for example the Orad HDVG used in this work (Orad, 2012b) has 16GB RAM and uses an optimised Linux Operating System to achieve the real-time render of graphics.

Once the virtual set is rendered the next stage is to apply chromakey to the video image and composite it together with the virtual set. A delay is added to the video image before this stage to account for the latency of the GPU (in our studio, this was measured to be 120ms).

Similar to the digital compositing methods presented in Figure 2-4 (page 11), the various elements of a virtual studio production are ordered using layers. Here layer 1 would contain the virtual objects that form the majority of the virtual set, which would appear behind the actor, layer 2 would contain the final matte of the actor and layer 3 onwards would contain objects that are required to appear in front of the actor. Multiple layers for virtual objects can be used, but typically only one layer can be used for the actor. Once the final scene has been composited, the broadcast output is produced.

This composited output can be considered a form of 'mixed reality'. A mixed reality environment is any scene that consists of the combination of virtual and real elements, usually in real time. These environments are described by the Reality-Virtuality Continuum (Milgram, et al., 1995), a continuous scale that describes the proportional blend of virtual and real elements, as shown in Figure 2-10.

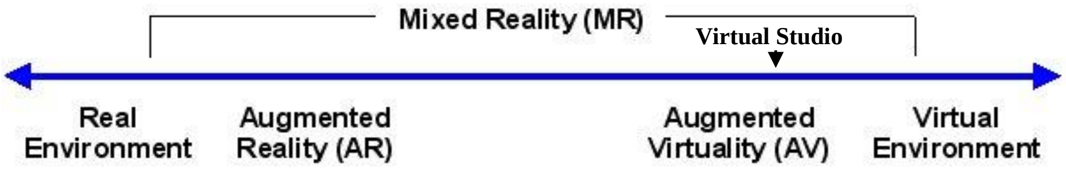


Figure 2-10. Reality-Virtuality Continuum and the respective placement of the virtual studio

The extreme ends of this scale describe pure environments; these are 'Real Environment' which is the physical environment a person would normally see, and 'Virtual Environment' which describes an environment entirely created using computer graphics. Everything in-between these two extremes is considered a form of Mixed Reality, as they are formed of elements from both.

Towards the Real end of the spectrum is 'Augmented Reality', where an image of a real environment is augmented by including virtual elements. Here an image captured by the camera is taken and a virtual object is rendered somewhere into the scene and displayed on a screen.

Towards the Virtual end of the spectrum is 'Augmented Virtuality', which describes a virtual scene augmented with real elements. In the case of the virtual studio the entire set is virtual and is augmented by the real actor, which conforms to the definition of Augmented Virtuality. In this thesis the actor is the only real element that augments the virtual scene. As such, many of the findings from research into the virtual studio can be applied to mixed reality environments.

The first virtual studio systems used virtual sets that were pre-rendered; this is to say that although the chromakey was processed in real time the virtual set had been rendered by a computer beforehand. This was necessary as GPUs were not sufficiently advanced to achieve the real time render of a virtual set. The first commercial use of a virtual studio was for the Seoul Olympics in 1988, where NHK (Japan Broadcasting Corporation) developed a system (called SynthVision) that keyed in a pre-rendered 2D background behind a presenter called (Shimode, et al., 1989); by 1992 NHK’s system was capable of using pre-rendered 3D backgrounds, and this was followed by similar systems developed by Ultimatte and British Broadcasting Corporation (BBC) (Gibbs, et al., 1996). Eventually the 3D graphics of virtual sets moved into the real time format that we are familiar with today where the graphics are rendered live. By the mid-90s several commercial, real-time 3D systems became available, produced by companies such as Accom and Orad (Gibbs, et al., 1996).

One of the first prominent uses of a virtual studio with a full 3D set rendered in real-time was for the 1996 Eurovision song contest, which was viewed by 300 million people worldwide. The opportunity was taken to use this show as a case study for the feasibility of producing television programmes using a virtual studio; here the virtual studio was used for an hour long section where a presenter announced how each nation scored the contestants. In this case study Hughes (Hughes, 1996) discussed the considerations that had to be made to produce an hour long high quality virtual studio production for live TV. This included suggestions made on lighting, camera frame rate and corresponding time to render the 3D virtual set, animations, the chromakey process and acting within the virtual studio. Many of these issues could be resolved - except for acting, an issue that proved difficult to overcome.

2.2. Interaction in the Virtual Studio

Hughes (Hughes, 1996) noted that acting in the virtual studio was hindered by serious shortcomings of the technology, where even simple tasks one would normally do with ease in the real world become difficult or even impossible. The ability of the actors to orientate themselves with regards to the location of props in the virtual set was a particular concern. In the blue/green screen environment the actor cannot see the objects of the virtual set; with no assistance they can only see the blue/green walls of the studio and perhaps some real props. This leaves the actor disoriented in the virtual studio, not knowing where to go and how to position themselves according to the virtual set. Hughes suggested the following measures to assist them:

- Placing physical (real) props and blue/green markers on the floor of the studio to provide location cues to the actor, telling them where objects presented in the virtual set are relative to these markers.
- Placing static monitors to provide the actor with the final composite, as used for broadcast.

We call the process of the actors being given information about their virtual surroundings (as described by Hughes) 'feedback'

Hughes' case study was followed by another study conducted by Gibbs and Baudisch, who focused on acting techniques in the virtual studio (Gibbs & Baudisch, 1996).

Gibbs and Baudisch (Gibbs & Baudisch, 1996) state that with acting in the virtual studio the experience of the audience is the primary concern, where the main goal is to make the illusion of the actor and a virtual object existing in the same space appear as plausible as possible. They state that providing feedback for the actor to co-ordinate themselves is an important consideration that needs to be made for achieving this. An example method they discuss is by providing the actor with markers or objects that are in the key colour, an example being a real box that has been painted blue placed in the same location that a virtual table will be rendered in, which the actor could use to navigate around the table. However some objects, particularly interactive ones, may change form or location which means that these static objects are not entirely reliable.

Gibbs and Baudisch (Gibbs & Baudisch, 1996) also categorised as two further types of issues that need to be solved for acting:

1. Modification of the behaviour of virtual objects so they appear to respond to real objects, and
2. Modification of the compositing process so that the real and virtual objects appear better integrated.

In the context of the research presented in this thesis, point 1 would be considered 'interaction' between the actor and virtual set and point 2 would be considered 'occlusion' between the actor and virtual set.

Interaction. In current systems if an actor touches a virtual object it does not move, when it would in a real setting. The ability to move an object is called interaction. This separation becomes even more apparent if the actor wishes to interact with the virtual set as they would a real one, where they may need to pick up an object to show to the audience but will not be able to. Gibbs and Baudisch (Gibbs & Baudisch, 1996) describe the state of the art interaction methods as being limited to the single-handed triggering of animations, where a human operator placed off set would trigger a virtual object to move along a path that was defined before the production and the actor would react accordingly. For example, if an actor goes to push a door, an animation where the door opens can be triggered by a person off set when the actor appears to touch it. While this is not true interaction, this does enable to the actor to *appear* as though they are interacting with the virtual object. However, this form of interaction requires the production to be tightly scripted and follow a linear sequence of events, with the actor well trained in what events are going to happen and when. This method would not be suitable for any extemporised interactions, where the actor may need to make an unplanned interaction with a virtual object, or where the object needs to follow unscripted actor movements. This method is also high risk and there is no chance for multiple takes, so if the actor creates a mistake they would have no second chance to correct it.

With CGI a visual effects artist has time to animate the virtual scene to correspond with the movements of the actor in post-production. But in the virtual studio where a production is live there is no time luxury to add this kind of effect - all interactions must occur with a high success rate in real time.

Occlusion. Occlusion is the visual blocking of one object by another, for example, the medium you are reading this thesis on is blocking something that is positioned behind it. In a purely real environment occlusion occurs naturally and in a purely virtual environment realistic occlusion can be achieved to near realistic levels using methods such as ray-tracing (Appel, 1968) (Whitted, 1980) or Z-buffering (Greene, et al., 1993). However, in a mixed reality environment, like the virtual studio, there is no universal method of creating this occlusion. A layering method similar to Figure 2-4 (page 11) is the pervasive in virtual studio systems. Here virtual objects can either be placed in front of or behind the actor and rarely switched during a broadcast. If an actor happens to walk from behind a virtual object to in front of it in real space, the virtual object will still appear in the foreground in front of them. This appears unrealistic to the audience and limits the ability of the actor to walk around the set.

The occlusion methods described in Gibbs and Baudisch's paper (Gibbs & Baudisch, 1996) allowed a virtual object to be set to appear in front of or behind the actor before a production. Gibbs and Baudisch state that changing how the objects appear relative to the actor during the production is possible using a method called Z-mixing, where the position of the virtual objects in the foreground or background can be changed in real time based on the location of the actor. This method will be discussed further in the review of literature for occlusion in the virtual studio (chapter 5.2.). The authors noted that this method can lead to errors, where the actor can appear to walk through a virtual object. If the actor standing behind a virtual object walks forward, they can suddenly appear in front of the virtual object or appear to walk through the virtual object.

Ultimately the popularity of the virtual studio as a general tool for producing entire television programmes was short lived due to the limitations associated with feedback, occlusion and interaction. These limitations reduced the range of programmes that the virtual studio could be practically used for to news and weather broadcasts, where the actor remains limited to a small, safe area and has no need to interact with or even acknowledge the virtual set. The following sections discuss the literature for each of these areas.

2.3. Feedback

The virtual objects that are used to construct the virtual set are invisible to the actor, who are only able to perceive the blue or green studio space (Hughes, 1996). Two broad categories of feedback have been previously investigated for their use in the virtual studio – Visual and Haptic.

2.3.1. Visual Feedback

'Visual' refers to the sense of sight and visual feedback is any method that provides information about the virtual set to the actor via their sight. Many visual feedback devices typically used for providing feedback in virtual environments, such as Head Mounted Displays (HMDs). Significant progress has been made in the field of HMD technology since the advent of technological convergence brought on by the dominance of smart phones and tablet PCs. These devices contain most components relevant to

HMDs, such as high resolution flat screen displays and low latency Inertial Measurement Units, and as such can be adapted easily into high quality HMDs at a low price point. Examples of this new breed of HMDs can be seen in Oculus Rift (Oculus Rift, 2013) and Google Cardboard (Google, 2014). The technology has also been ported to AR HMDs, such as Google Glass (Google, 2013) and Microsoft Hololens (Microsoft, 2015). However, HMDs would be visible to the audience and thus not feasible for use in the virtual studio. Hence, any feedback techniques need to be obscured from the view of the camera and subsequently the audience.

As discussed, from the Eurovision 1996 case study Hughes (Hughes, 1996) suggested that markers placed on the floor, hidden video monitors and real props can all be used to help the actor identify the locations of virtual objects. These methods have frequently been used in virtual studio productions since, but are limited in the level of feedback they can provide to the actor, particularly in complex productions such as those that involve interaction.

This is further supported by Daemen (Daemen, et al., 2013), who placed invisible markers around the virtual studio to help the actors navigate and locate the virtual objects. They noted that this helped the actors to locate the virtual objects, but was not useful in relaying any changes to the virtual object's state as the markers were placed in a static location and did not change. It was suggested that a haptic or auditory approach towards feedback may benefit actors.

One alternative to static markers is to move an object using a system of ropes and winches. SpiderFeedback (Simsch & Herder, 2014) is a physical system developed at the University of Dusseldorf in which servos guide a suspended real object in the studio to replicate the location of a particular virtual object. The real object was removed during the chromakey process and the ropes were imperceptible to the camera. This provided the actor with an accurate estimation of the object location, specifically to allow them to orientate themselves and their location better.

To counter this lack of awareness, out-of-shot static video monitors are usually placed around the studio that shows the mixed output that the end home viewer would see. However, the static nature of the monitors can cause the actor difficulty when attempting to locate or interact with virtual objects, particularly if there is no monitor directly in their line of sight or they are moving around the set.

This means the actor will need to look at the static monitor while interacting with a virtual object, which results in difficulty during orientation tasks and does not allow them to appear as though they are looking directly at the virtual object. The actor benefits from having an ever-present form of display available to them, allowing them to locate the virtual objects and produce a correct gaze (Thomas & Grau, 2002).

Sophisticated approaches towards concealing the presence of static monitors have been developed that allow a more versatile placement. A popular technique is to place a flat panel display in the scene itself behind the presenter, where the colour of the bezel and the content shown on the monitor matches the key colour and can be removed in the chromakey process. This process is shown in Figure 2-11 where the presenter is able to accurately point to data rendered in the virtual set by referring to its location on the monitor (vizrt, 2013).

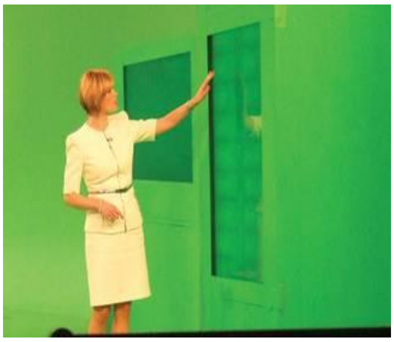


Figure 2-11. Example of a presenter receiving feedback from a monitor that displays information in the key colour (image courtesy of VizRT (vizrt, 2013)).

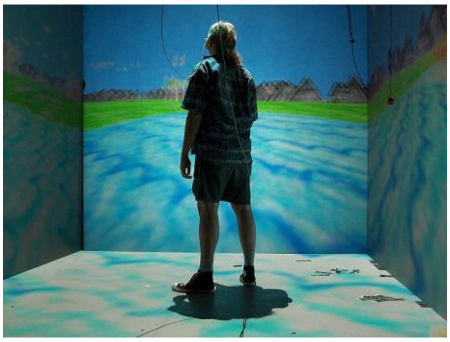


Figure 2-12. An example of a CAVE (Image Courtesy of Dave Pape, University of Buffalo) (CC BY 2.0)

Another approach, as used by Kim (Kim, et al., 2006), is to place the feedback monitor in the virtual studio and conceal it by placing a virtual object in the same location to mask it from the viewer. Despite solutions existing that allow static monitors to be placed in more locations, the static nature of them still means that actors will occasionally find themselves without adequate feedback if no monitor is in their line of sight.

Projection technology has been a considerable area of focus for providing advanced visual feedback in the virtual studio, as it can overcome the static nature of the monitors and can provide ever-present feedback to the actor. Two CAVE (Cave Automatic Virtual Environment) (Cruz-Neira, et al., 1992) style systems have been developed for this domain. CAVE is an alternative approach towards HMDs used in virtual reality, where a virtual environment is presented to the user by projecting it on up to six surfaces around them, as shown in Figure 2-12 (page 20). In the virtual studio these methods provide the actor with a POV perspective of the virtual set on every surface around them. POV feedback means the virtual set would be presented as seen from the eyes of the actor.

Origami (Grau, et al., 2005) was a European council funded project to produce advanced tools for 3D film and television productions where real and virtual elements are merged together. The environment they produced was similar to a virtual studio, but built with the purpose of being a tool for film production too.

One of the key aims of Origami was to provide the actor with enhanced visual feedback. The authors noted that when acting in a blue space, even simple acts of interaction like maintaining eye contact with a virtual character was difficult. Although displaying feedback onto a monitor placed off set could help, there were many cases where the actor was required to look at a virtual object in a location where no feedback monitor was in their line of sight, concluding that the visual feedback needed to be available to the actor wherever they looked. The solution they implemented was to project an image of the virtual set from the POV perspective onto every surface in the studio area. However, projecting onto the walls of the virtual studio would interfere with the keying process and would produce visible artefacts in the broadcast output.

The solution to this issue was to not paint the walls of the studio in the key colour as is normally done, but to use a retro-reflective cloth instead. Unlike the diffuse surface of the paint where light would be scattered in all directions, the retro-reflective cloth was formed of many tiny glass beads that only reflected light back in the direction that it came from. From the perspective of the actor the projected image could be seen clearly, although bright projector was required. The studio camera had a ring of bright LEDs (Light Emitting Diodes) of the desired key colour placed around its lens, which would be reflected back and made the retro-reflective cloth appear in that colour to the camera.

A similar approach was discussed by Mitsumine *et al* (Mitsumine, et al., 2005), which used diffuse projection surfaces for every wall of the virtual studio. In this project an array of projectors were placed behind the projection screens that formed the acting space, projecting inwards towards the projection surfaces. From the perspective of the actor the matted surfaces would show the virtual set.

Feedback for Origami proved to be successful as it allowed the actor to maintain eye contact with a moving virtual character, but had large resource and space requirements. To achieve the spread required for the projectors, they had to be placed far away from the desired surfaces and in some cases outside of the acting area itself. This reduces the practicality of this method for use in many virtual studios, where one of the benefits is that they are a compact environment.

A single projector, The Invisible Light Projection system (Fukaya, et al., 2003) has also been developed. This system was formed of a static projector providing feedback onto a portion of one surface in the studio. In this system the visual feedback was projected in phase with the camera shutter so that it only projected when the studio camera's shutter was closed, ensuring that the viewers would not be able to see the feedback but the actor would. Whilst resource and space requirements are lower, it did not provide the same level of ubiquitous feedback as the CAVE style systems as only a small portion of one surface could be covered.

In both cases projection technology have been demonstrated as a useful tool, but to improve the suitability of projection technology in the virtual studio a projection device that has both the low space requirements of the Invisible Light Projection systems and the ever-present feedback provided by the CAVE style systems would be required.

Recently CAVE has been used to train ballet dancers in an immersive virtual environment (Sun, et al., 2014), enabling feedback on the pose of their bodies to be provided. The pose of the ballet dancer was captured using a Microsoft Kinect and the features of postures were extracted from the skeleton joints. The system used a library of pre-defined postures, which it matched to the same postures the dancer was attempting to complete, allowing a quantitative assessment of individual movements. The system was very effective in this recognition, implying that a similar surround projection based system with motion capture may be able to aid the motion of the actor for interacting with virtual objects.

A variation on CAVE is the system developed by Kuchera-Morin *et al* (Kuchera-Morin, et al., 2014) which is a full surround projection environment in a spherical form. That system utilised 4 large high lumen projectors and 22 small footprint projectors to create a seamless display of an environment that could be used by up to 3 users to view complex data. The stereoscopic effect was achieved using polarised glasses. Via a pair of pinch gloves, the users could interact with the 3D environment that being displayed to them.

2.3.2. Haptic feedback

Haptic refers to the sense of touch. When a person feels resistance from a rigid object surface or feels a vibration, these can be described as haptic sensations. Devices have been created that can replicate these haptic sensations for providing information to a person, a popular example being the vibrate function on a mobile phone that alerts the owner of an incoming call. The use of similar haptic devices has been applied for feedback in the virtual studio to help aid the actor in identifying the locations of virtual objects.

The earliest use of haptic devices in the virtual studio was by Kim *et al* (Kim, et al., 2006), who explored their use in the wider context of creating a fully interactive virtual studio. This project used vibrotactile haptic devices placed in the palm of the actor that vibrated when a collision with a virtual object was detected. It was reported that while these appeared to improve the performance of the actor by allowing them to appear more confident when reaching towards a virtual object, they were still liable to incorrectly estimate the surface of the virtual object. The size of the haptic devices and the method of fixing them to the actor's hands would have also rendered these devices visible to the home viewer.

The department of Media at the University of Applied Sciences in Düsseldorf is active in developing methods that improve human performance in virtual studio systems. One of the focal points of their research is enhancing the ability of the actor to identify the location of objects for navigation and interactive tasks using haptic methods.

Initially this group’s work considered the use of a haptic belt that provided vibrotactile feedback to aid the navigation performance of the actors (Woldecke, et al., 2009), which the authors defined as the simplest form of interaction (or identification of object boundary locations). The belt consisted of a series of wireless tactors (a type of vibrotactile device), which were placed around the torso of the actor. Feedback was provided in one of two modes. The first mode vibrated the tactors nearest to a virtual object surface to relay its location to the actor, allowing the actor to walk an arbitrary path through the set. The second mode informed the actor of a predefined motion path by vibrating the tactors in the direction he should be moving in.

From this research the group also applied these techniques to guide the actor's arm towards the nearest virtual object (Woldecke, et al., 2010). Woldecke compared this method to feedback using a monitor and found that a visual approach allowed a faster movement of the actor's arm towards the virtual object and each method allowed an (approximately) equally accurate arm placement. The research of this group is discussed in greater detail in chapter 7.2.3 (page 90), as their measurements in human motion with regards to identifying object locations in the virtual studio are more appropriate there.

Rekimoto (Rekimoto, 2014) proposed an alternative to vibrotactile devices for providing haptic feedback in virtual environments called the Traxion, which, without any mechanical links to the ground, uses human illusory sensations to create the perception of a force. This is achieved by using an actuator to move a weight in one direction in an asymmetric manner, which the human user perceives to be a force in a particular direction.

2.3.3. Summary of Feedback

Woldecke *et al* demonstrated that haptic feedback techniques were successful in allowing actors to identify the locations of virtual objects and guiding them through pre-set paths, although they also demonstrated haptic methods were no more effective than screen based visual feedback techniques. Woldecke *et al* also identified that their haptic belt system could be improved by including a visual feedback system. Current commonplace visual feedback systems such as static monitors and markers are limited in the locations that can be provided to the actor. Visual feedback systems have been developed that allow a far greater range (such as projection systems Origami (Grau, et al., 2005) and that by Mitsumine *et al* (Mitsumine, et al., 2005), although they are associated with various footprint and resource limitations.

2.4. Occlusion

2.4.1. Occlusion in Virtual Environments

Creating authentic appearing occlusions akin to those present in a real environment formed a significant part of early research into virtual environments and computer graphics. One of the earliest approaches towards occlusion in virtual environments was the Painter's algorithm (or the Priority algorithm). This is a simple method of rendering objects in a virtual environment, where the objects in the scene are rendered in a hierarchical order from those furthest away to those nearest the virtual camera (Jacobs, 2004). Each 3D object is presented as it would appear projected in a 2D image, with each layered on top of the previous one (as presented in Figure 2-13). This method of creating occlusion is computationally inefficient as objects that are later occluded need to be rendered fully first. The algorithm also cannot handle objects that overlap in a cyclical manner or intersect each other, which is an issue known as the Painter's problem. This issue will be discussed here due the nature of a similar problem that presents itself in chapter 5.

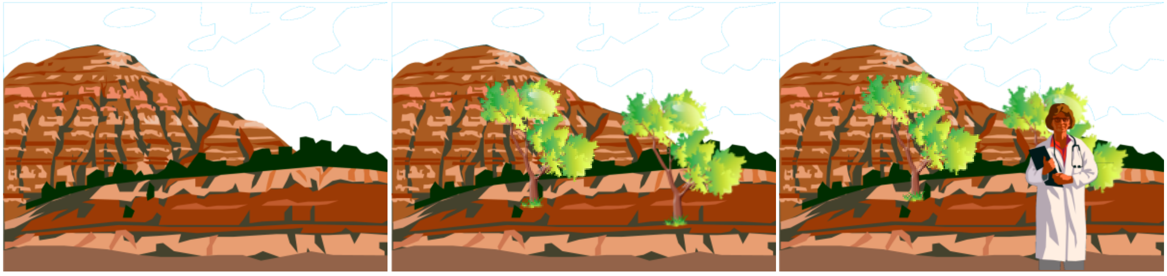


Figure 2-13. Demonstration of Painter's algorithm, where (from left to right) the rearmost object is rendered first (the mountain), followed by the trees and then the foremost object in the scene (person).

An example of the Painter's problem is presented in Figure 2-14. Here Figure 2-14a shows an example of the intended occlusion, where the rectangles occlude each other in a cyclical manner. In this image each rectangle occludes and is occluded by the other two rectangles. Figure 2-14b shows the same image created using the Painter's algorithm, where each rectangle could only be inserted one after the other (in the order of Blue, Green, Red). As a result the image is rendered incorrectly, where the blue rectangle is occluded at both of the other rectangles as it was rendered first and the red rectangle occludes the other two as it was rendered last.

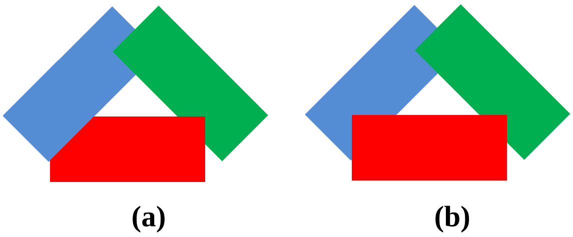


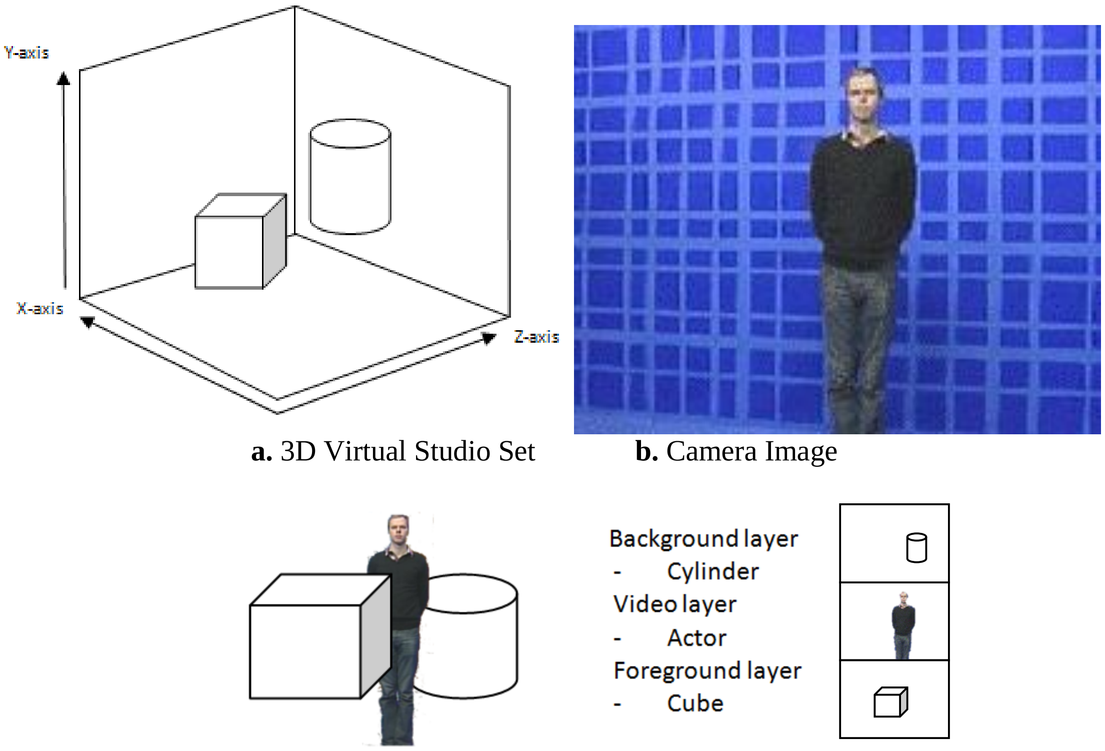
Figure 2-14. Example of the Painter's problem. (a) Shows the intended 'cyclical' occlusion. (b) Shows the incorrect occlusion caused by the Painter's problem.

The Painter’s problem has been solved in purely virtual environments using several different techniques, notably with Z-buffering (Greene, et al., 1993) and Ray Tracing (Appel, 1968) (Whitted, 1980) (the latter more commonly used for lighting, but also applicable to hidden surface determination); however, the problem still persists in the comparable technique of layer-based compositing, where elements of a scene are layered on top of each other and do not allow the type of cyclical occlusion seen in Figure 2-14a. While this issue can be solved manually in fields like post-production where the scene can be manually segmented and composited together, it is difficult to solve in a real time layer-based environment such as the virtual studio as the process would need to be automated.

2.4.2. Occlusion in the Standard Virtual Studio

Current commercial virtual studios handle occlusions by using a system of layers analogous to the Painter’s algorithm and digital compositing, where virtual objects in the set can be rendered on either a background layer or a foreground layer (with respect to the video layer containing the actor). Objects are set to a layer prior to broadcast and typically remain on the same layer during a live broadcast.

The virtual set is designed entirely as a 3D virtual environment and each object is rendered on its appropriate layer. An illustration of this format is shown in Figure 2-15 (page 26), with the studio camera assumed to be pointing along the Z axis. Here Figure 2-15a shows the overall layout of a simple 3D virtual set, constructed using two virtual objects (a cube and a cylinder). Figure 2-15b shows the real camera’s viewpoint with the actor captured as a 2D video layer with no definite Z location. Figure 2-15c shows objects in the 3D virtual set rendered onto two 2D layers with respect to the 2D video layer - with the cylinder on the background layer and the cube on the foreground layer. Objects set to the background layer (in this example the cylinder) always appear behind the actor in the video layer, whilst objects set to the foreground layer (in this example the cube) always appear in front of the actor. This method requires any changes in occlusion to be scripted and rehearsed, as well as being subject to errors caused by incorrect timing.



a. 3D Virtual Studio Set **b.** Camera Image

c. Example of Simple Layering system (Composite of Figure 2-15a and Figure 2-15b)
Figure 2-15. Illustration of layering system

Several systems for the virtual studio have been developed that address the change of occlusion properties according to the movement of the actor with respect to the virtual objects in an automated manner. Prometheus (Price & Thomas, 2000) was a UK LINK funded project that aimed to analyse the changing state of virtual studio technology, which was spurred by advances in real-time 3D technology at the turn of the 21st century. One aspect of this project was occlusion, where two modes of actor tracking for occlusion were discussed.

The first method was to create a realistic 3D model of the actor before the production using AvatarMe, developed by Hilton *et al* (Hilton, et al., 1999). This captures the actor using cameras from several different viewpoints and reconstructs the actor in an offline process, creating an ‘Avatar’ (a 3D model of the actor). The avatar is then animated live by analysing the pose of the actor from multiple camera views during the production and matching their movements. The audience would not see the real actor in the virtual studio, but would instead see the avatar inside a full virtual environment (as defined by the Reality-Virtuality continuum (Milgram, et al., 1995)). Occlusion would be produced using standard virtual environment occlusion techniques.

Global adoption of this method would require considerable changes to be made to the existing structure of virtual studio systems, as it would remove the layer-based mixed reality nature of the current systems and make the production entirely virtual. In addition, reconstructing the actor as a virtual object resulted in a poor visual fidelity, causing them to appear inauthentic.

The second method saw the actor tracked as a single point using an auxiliary camera placed above and behind them, as shown in Figure 2-16. From the captured image the actor is separated from the studio floor (which is in the key colour), their location in the camera image can then be found and referenced to a real space location on the studio floor.

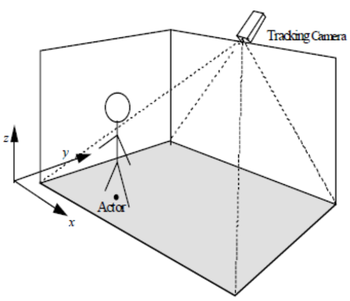


Figure 2-16. Auxiliary camera used to track actor's feet (image courtesy of Price & Thomas, 2000)

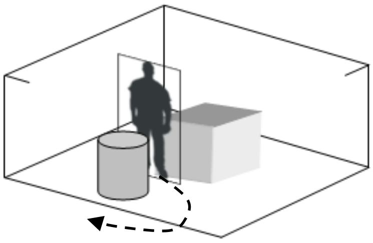


Figure 2-17. Demonstration of 'Billboard' method, where the actor is represented on a 2D plane within a full virtual environment (the border of the plane is in black for demonstration purposes). The dashed black line shows a potential route the actor could take to walk in front of the cylinder.

This single point was used to achieve a simple occlusion effect called the ‘billboard’ technique (Figure 2-17), where a textured plane containing the live image of the actor is presented in the virtual set as though it were a virtual object in itself. The plane is then matched to the location of the actor’s feet and moves around the virtual set as they walk. It is important to note that occlusion does not occur in layers as it would in a standard virtual studio, as again occlusion would be produced using a full virtual environment method. Similar tracking systems, such as the X-Ploro developed by Xync/Orad (Digital Broadcasting, 2001), could also be used in a similar manner to achieve this method of occlusion.

The authors (Price & Thomas, 2000) note that one major limitation is that when using this technique the whole of the actor would have to be captured by the studio camera as their entire body would need to appear on the plane. This consideration limits the range of camera movements in the studio. If the actor’s feet were not captured in the camera image, yet the audience could see the entire plane, it would appear as though the actor had no feet.

A similar approach was used with radarTOUCH (Marinos, et al., 2012) a device that identifies where a rotating laser is interrupted, which allows the location of the interruption to be measured on a 2D plane, providing the location of the actor’s feet. The laser uses 905nm wavelength, which is not visible to a studio camera, meaning that is can be used for imperceptible capture of the actor’s location in the virtual studio. It was implemented into the virtual studio by Marinos *et al* who used the device placed horizontally to track the location of an actor’s feet inside the virtual studio space.

However, the key limitation for occlusion with this method is that using only a single point for actor tracking is insufficient, as many interactions and occlusion events occur at the extremities of the actor's body (such as the hands and on occasion with the legs and head). More advanced motion capture techniques have been developed that track the extremities of the actor's body, such as the Microsoft Kinect/OpenNI approach used in this work (Microsoft, 2011) (Primesense, 2011).

Z-keying (or Z-mixing) (Kanade, et al., 1995) is another method that has been implemented into virtual studio systems to achieve occlusion. Z-keying is an attempt to apply the Z-buffering (Greene, et al., 1993) approach to occlusion in the virtual studio by combining multiple depth maps. A depth map is an image that contains information on the distance of each pixel from the camera. Visually the distance is represented as an 8-bit greyscale, but can be translated to centimetres.

Figure 2-18 shows an example of the Z-keying process. Here the three leftmost images (a-c) show a real environment (a), a segmented actor (b) and a group of virtual objects (c). The images to the right of each of these (d-f) show the corresponding depth maps, where lighter objects represent those close to the camera and darker objects represent those further away. The depth maps are combined to form a single depth map where the highest Z value for each pixel (the lighter pixels) occludes the darker pixels, which is used to describe which pixel in each of the colour images should appear in the composited image. The two images on the right of the figure show the combined depth map (h) and how the final composited image appears (g).

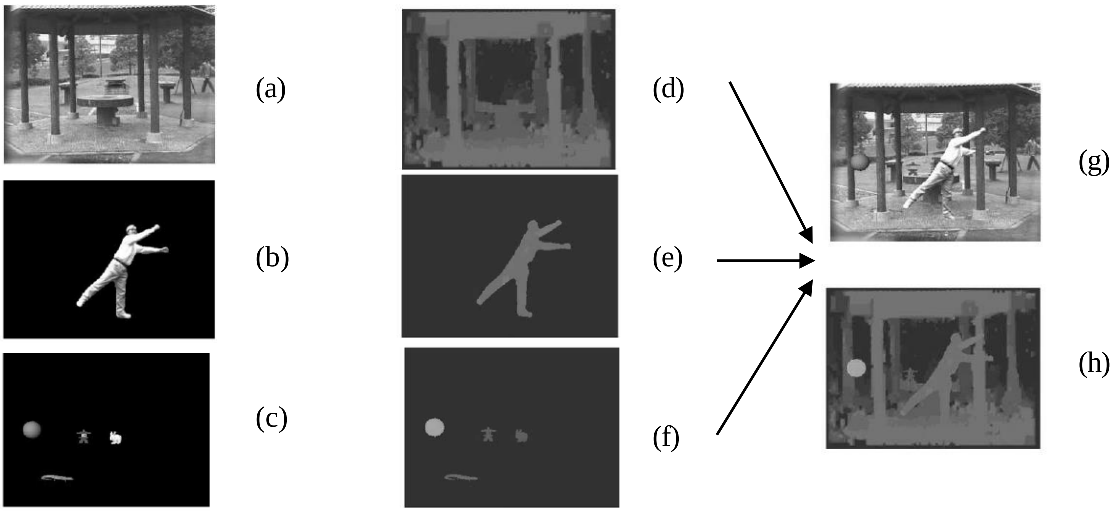


Figure 2-18. Example of Z-keying (figure courtesy of **(Kim, et al., 2006)**)

These Z-keying systems, the stereo camera system (Kim, et al., 2006) and Time-Of-Flight (TOF) system (Koch, et al., 2009), have been used to obtain a depth map image of the studio to assess occlusions compared to the virtual set. Unlike the developments presented in this section, these Z-keying occlusion systems are not layer-based and will suffer from compatibility issues with the layer-based system of the virtual studio. These systems may also yield unwarranted occlusions if pixel-by-pixel comparison is used, such as actors appearing to intersect what should be a solid object (an image illustrating this situation is presented in

Figure 2-19 where the actors are intersecting the virtual globe). The work presented in this thesis proposes an evolutionary approach to the standard layer-based system instead, offering greater compatibility with existing virtual studios.

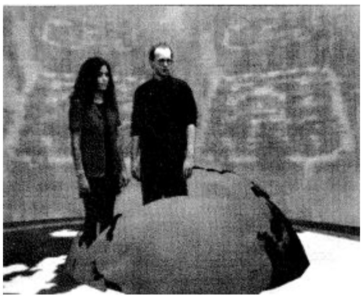


Figure 2-19. Example of actors appearing to walk through a solid globe, an incorrect occlusion image courtesy of (Gibbs & Baudisch, 1996).

The occlusion method used by Daemen (Daemen, et al., 2013) references the occlusion method presented in this chapter, where it is extended for use with up to three actors. Little information is provided about the technicalities of their occlusion system, although it is stated that it is based on the skeletal tracking data of the actor.

The state of the art for Z-mixing is Khattak *et al* (Khattak, et al., 2014) who developed a z-mixing occlusion system for a head mounted AR display that correctly merged virtual objects and the user's hands in real time. The system used a Creative RGB-D camera mounted on the front of an Oculus Rift (Oculus Rift, 2013). The RGB-D camera's location was calculated from an AR marker that was positioned on a flat surface, from which the AR object was also anchored to. With this mutual reference point, a depth map of the virtual environment drawn from the perspective of the RGB-D camera and the depth map of the real environment from the RGB-D camera could be merged.

Katahira & Soga (Katahira & Soga, 2015) developed a z-mixing occlusion method for AR that allows a realistic form of grasping. The system they developed focused on the occlusion of a human hand, utilising a depth camera (a Leap Motion Controller) in a different manner to traditional z-mixing approaches. The depth camera was used to acquire the depth data of the hand and fingers, which was then applied to a virtual model of a hand. This information was used to create transparencies in the virtual object in a manner that creates the correct occlusion in a grasping scenario. A similar solution to occlusion is proposed in this thesis in section 5.3.2.3.

2.4.3. Summary of occlusion

Current tracker based occlusion systems do not allow occlusions to be changed at the extremities of the actor's body, which means that this approach is inadequate when considering interactive applications taking place at hand locations. Other depth based applications using Z-mixing or reconstructing the actor as a virtual model account for occlusions that occur at these extremities, but have issues associated with them (e.g. drifting through the virtual object, removal of the layer-based standard and significant changes to existing virtual studio infrastructures).

2.5. Interaction

Moving an object is an activity that we perform frequently throughout our daily lives and is often done with little cognitive effort. If we want to pick up a phone, we simply place our hand around the handle and pick it up. If we want to pick up a box we would place our hands on either side and lift it. Guiard's seminal paper on interaction (Guiard, 1987) defines three key forms of interaction between humans and objects:

- Unimanual (or manual), where an interaction is completed using one hand only (e.g. brushing your teeth)
- Bimanual-asymmetric, where one hand performs coarse movements to guide a hand performing fine movements to achieve an interaction (e.g. interacting with a touchscreen tablet)
- Bimanual- symmetric, where both hands move in a similar manner (e.g. picking up a box)

The research presented throughout this thesis explores creating interactions in a unimanual and bimanual-symmetric domain in chapter 6, as defined by Guiard. Bimanual-symmetric interactions are explored further in chapters 7 & 8 when looking at the plausibility of interaction. This is because hand placement errors were found in the analysis of the bimanual interaction system discussed in chapter 6, deeming further investigation on exploring plausibility necessary. This issue did not affect manual interaction as the object surface is matched to the location of the actor's hands.

As discussed in section 2.2 in the study by Gibbs & Baudisch (1996), interaction between the actor and the virtual set has historically been limited to predefined events that can be triggered by the actor using a basic control input or off set by an operator. This approach is commonly used for weather broadcasts, where the presenter will press a button to relay a command that will change the weather map. Actors would find it difficult to use this method of interaction should they need to react to the virtual object (i.e. mime interaction to an animated virtual object), where hand placement accuracy and creating a plausible interaction requires intricate timing. As any complex form of reaction to the animated virtual object is a difficult task and is prone to errors this style of interaction is kept to an absolute minimum. Several interactive systems have been developed to improve on this limitation by allowing the virtual object to react to the movement of the actor. In this review these techniques are broadly categorised as tracked devices, 2D interaction and 3D interaction.

2.5.1. Tracked devices

Tracked devices have been used in several ways to create direct interaction between the actor and the virtual object. The tracked device would allow its orientation and location to be measured and a virtual object can be rendered in the corresponding place.

Interaction with virtual objects can be achieved by using tracked Augmented Reality markers (Kato & Billinghurst, 1999) (Kato, et al., 2000). This is a form of technology that utilises flat textured markers, which are captured in a camera image that is processed using a standard PC to identify its location and orientation in the image, then a virtual object is rendered to that position in the scene. A studio based system based on these markers was developed with MixTV (Lalioti & Woolard, 2003), which used Augmented Reality markers on which virtual objects were rendered; this allowed the actor to manipulate virtual objects by moving the marker. This approach has not become widely used due to several innate problems, such as the potential occlusion of markers from the camera and problems caused by the physical nature of the markers (e.g. cannot be easily disposed of).

Augmented reality technology has developed further in recent years, yet the developments have not been used to enhance interaction in the virtual studio. The advantages that recent developments in augmented reality bring would allow for some of the aforementioned limitations to be overcome. An example of this being HandyAR (Lee & Höllerer, 2008.), which is a markerless take on marker-based augmented reality system by rendering the object to the hand of the user instead of the marker.

The tracked wand is an interaction method that has been introduced to the commercial virtual studio domain, notably by VizRT (VizRT, 2014). The wand is typically a cylindrical device with a number of reflective markers located on protruding antennae. The markers reflect Infrared (IR) light and are detected by an array of Infrared cameras each with an Infrared light source positioned around the lens. The images from the tracking cameras are combined to calculate the location and orientation of the wand in 6DOF. Figure 2-20 (page 32) presents an example of this system in use, where the left image shows an actor holding the retro-reflective wand tracked by an array of auxiliary cameras and the right image showing a virtual shield mapped to the location and orientation of the wand. As the actor moves the wand the shield moves.

The tracked wand method provides a robust approach towards interaction, as the actor has a tactile device that allows them to control the virtual object with ease and the nature of the tracking system is highly accurate. However, this method suffers in terms of versatility as the actor is required to use a physical device manually and their interactions are limited to one virtual object that is assigned to that device. The virtual object that is represented by the tracked wand would also have to be large enough to mask it from the audience, ruling out interactions with small objects.

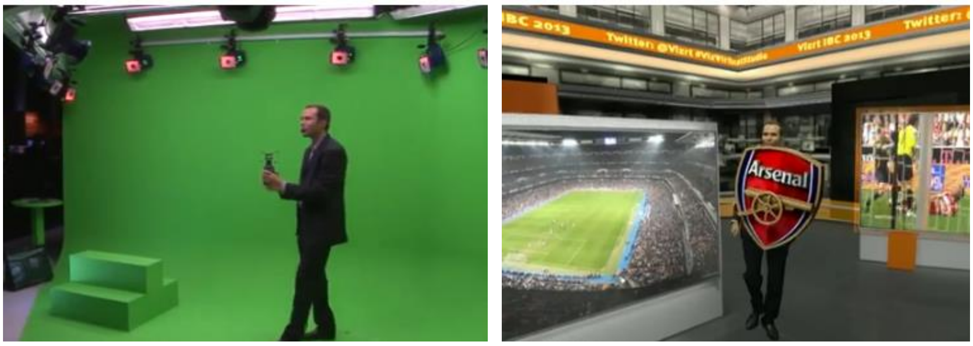


Figure 2-20. Example of the tracked wand system and its use by an actor and the final composite courtesy of (VizRT, 2014).

A wired glove has been used for interaction with a virtual set where the location and posture of the actor’s hand was tracked (Minoh, et al., 2007), which enabled the actor to interact directly with a virtual object using one hand. Like many hardware based forms of motion capture the wired glove is visually invasive, which makes it impractical for broadcast use in its raw state as the audience would be able to see it. The solution that Minoh proposed was to digitally replace the wired glove with a virtual model of a hand. The results still appeared somewhat unrealistic (Figure 2-21 shows an example of this method in use), demonstrating that the visual invasiveness associated with hardware based motion capture is difficult to overcome. A feasibility study was conducted that explored the development and use of an imperceptible wired glove (discussed in Appendix #A), but this was not pursued any further.

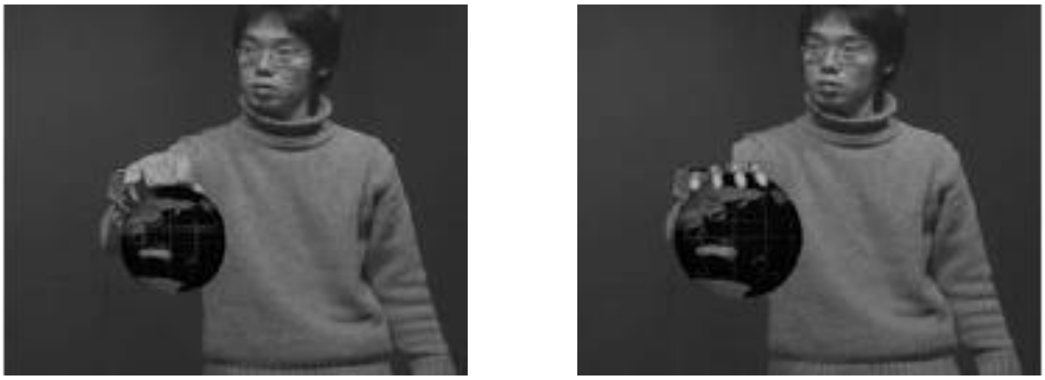


Figure 2-21a. Interaction using wired glove **Figure 2-21b.** Virtual hand rendered over glove

Figure 2-21. Wired glove interaction in the virtual studio with virtual hand rendered over the actor's hand (from Minoh (Minoh, et al., 2007))

2.5.2. 2D Interaction

2D Interaction occurs in a planar space. Using this type of interaction allows the actor to select options and create simple 2D interactions. As an extension of his previous work in occlusion (discussed in chapter 5.2), Marinos (Marinos, et al., 2012) also demonstrated the use of radarTOUCH as a tool for interaction. Here the virtual objects in the studio were set to react to the actor as they approached them, with the example provided being one of a virtual door opening automatically as the actor walked towards it.

Marinos extended this approach towards the 'Zoomable User Interface' (Marinos, et al., 2010). This technique used a radarTOUCH positioned in a vertical setting, where it could detect where the user's hands were when they broke the laser, providing a multi-touch interface. Users interacted with a menu projected onto a surface in front of them, which they navigated through using a two hand command to zoom and a single hand to select items and pan through options. It was found that this method was more intuitive for users than a standard mouse-based interface, where the buttons are used to select items and the scroll wheel is used to zoom. Although a successful interface was built in a vertical setting, the method was not used in the context of the virtual studio.

A recent development for interaction in the virtual studio has been telestration, which is a method that uses a touchscreen or touch-sensitive surface placed in the studio space that the actors can use to control the behaviour of the virtual objects. The touchscreen is typically concealed from the viewer, where the bezel and on screen graphics are presented in the key colour, allowing the monitor to be overlaid by a background object in the virtual set. An example of this technology is discussed in a case study by Ian White (White, 2010), where a telestration screen was successfully used for ITN's coverage of the 2010 UK election, allowing the presenter to move through and select data to present different scenarios that could arise from the election. An example of the touchscreen being used in this case is presented in Figure 2-22.

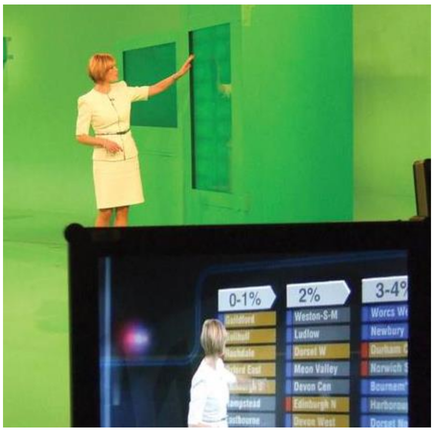


Figure 2-22. Example of an actor interacting with the virtual set using a touchscreen linked with a VizRT virtual studio. Image courtesy of VizRT (image courtesy of VizRT (vizrt, 2013)).

In Figure 2-22 (page 33) the top part of the image shows the presenter, who is able to see virtual graphics presented in the key colour on the screen so that they can be keyed out, touching the screen to decide which elements to interact with. The presented can swipe the screen to move to the next section of information or touch one piece of information to enhance it for the viewer.

The popularity of this approach is due to the reliable interaction that can be offered. The touchscreen provides an ergonomic foundation for interaction, a rigid surface with a familiar 2D interface similar to a touchscreen tablet computer or smartphone. The actor can trigger interactions, draw on-screen graphics (e.g. for sports broadcasts) and navigate through menus and data. However, the limitation of the telestration system is in its constrained range of interactions. The actor is only able to interact with virtual objects in a disconnected manner within a small 2D space, so this method cannot produce a realistic looking interaction.

2.5.3. 3D Interaction

Two forms of achieving 3D interaction currently exist, those using depth-sensitive cameras and those using multiple 2D cameras.

2.5.3.1. Depth Cameras

A natural virtual studio for “inter”–acting (Kim, et al., 2006) used a stereo camera setup (two cameras placed side by side) to track the motion of the actor. Here the disparity between the corresponding pixels in the two captured images was used to create a depth image of the scene.

Kim *et al* (Kim, et al., 2006) used two methods to detect possible interaction events between the actor (captured using the depth camera) and the virtual set. The first was to detect whether the points in the depth image that represented the actor intersected the points that represented the virtual object. The second was to use collision detection between the hull of the actor and the hull of the virtual object. Collision detection is a common computational procedure for identifying when and where two virtual objects intersect (i.e. their surfaces come into contact). Because the actor was represented as a 3D point cloud in the depth image, the points had to be connected together to produce a 3D hull. This proved to be computationally inefficient when done in real time, so only the points that were detected within a specified distance of the virtual object were converted.

The level of interaction that was achievable from this method was limited to simple applications and was unable to replicate complex single-handed and bimanual manipulations of virtual objects. To demonstrate the level of interactivity, Kim provided two examples of interaction possible with this virtual studio. In the first example the actor was able to change the path of a virtual object orbiting around them by placing their hand in the object's motion path. When collision was detected the path of the object was reversed, in this case changing from clockwise to anti-clockwise and vice-a-versa. In the second example information that was overlaid on a 3D weather map was changed based on the location of the actor. As the actor walked from one part of the map to the other, the information for the weather in that part of the country was displayed.

While the level of interaction made possible by Kim's virtual studio was relatively basic, it did demonstrate that creating interaction using a depth-sensitive camera is a potentially useful method that could be extended.

It is also important to note that Kim also defined three classes of virtual object, based on their interactive characteristics:

1) Objects that do not interact with each other or the actor,
2) Objects that interact with each other, but not the actor, and
3) Objects that interact with the actor.

Our work will use virtual sets created with objects that fit these definitions.

Flasko et al (Flasko, et al., 2012) developed a similar system to Kim *et al,* but improved the method for tracking the head and hands of an actor in the virtual studio for interaction. Here the actor was tracked using a stereo camera, obtaining the depth image using a similar method to Kim *et al* (Kim, et al., 2006). The colour images from the cameras were used to identify the location of the hands and head using an algorithm to segment skin colours (van den Bergh & Lalioti, 1999), which were mapped to the corresponding pixels in the depth image, creating Regions of Interest (RoI) which provided the 3D hand locations. The largest RoI is assumed to be the head of the actor and the remaining two RoIs are assumed to be the actor’s hands. The interaction demonstrated in this paper was a virtual box placed in the location of each of the actor’s hands that followed these location as they moved, demonstrating a relatively basic single-handed interaction.

Stereo cameras were the standard method of inferring depth in a captured image for many years, but recent advances in technology spurred by new commercial applications have produced new types of depth sensing cameras that are capable of achieving much higher resolutions and accuracies (such as Time Of Flight cameras). These can be used to capture the markerless motion of the actor in the virtual studio and have the potential to support a higher level of interaction with the virtual set than would be possible with stereo cameras as they do not suffer from the correspondence problem.

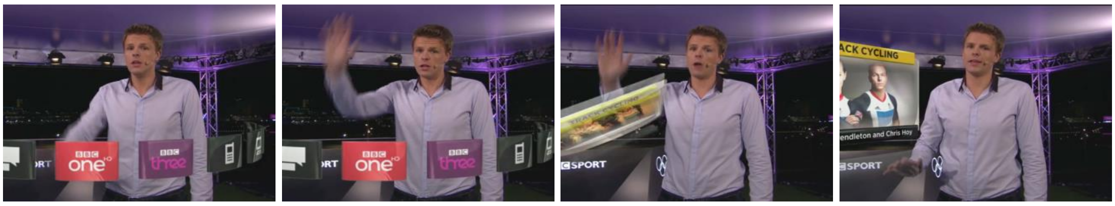


Figure 2-23. Use of Kinetrak during the London 2012 Olympics, where the actor’s gesture selects an option. (Image courtesy of (Mammoth Graphics and Kenziko Ltd, 2014)).

The current extent of using advanced depth-sensitive cameras in the virtual studio has been to create basic gesture-driven interactions between the presenter and a virtual menu that appears in front of them. This interactive system was developed by Mammoth Graphics and Kenziko called Kinetrak (Mammoth Graphics and Kenziko Ltd, 2014), which used a depth-sensitive camera. This was first used for the London Olympics, where the presenter used the system to bring up a menu by making a 'thrust' gesture with his hand and selected the option he wanted to present from the menu by placing his hand behind it and making an 'up' gesture, as shown in Figure 2-23. zLense (zLense, 2014) also offers a depth-sensitive camera-based approach for interaction with a virtual set, which allows the actors to take part in basic interactions where they are able to knock over virtual objects with their hands or feet. The technology appears to be based on a physics engine, allowing virtual objects to react when the actor's point cloud collides with them.

The gesture or location based interactions reflect only a small section of what the depth-sensitive cameras are capable of, as the information from the skeletal tracking of the actor that can be achieved using these methods could be exploited for more detailed interaction.

Lee *et al* (Lee, et al., 2015) produced interaction via a depth sensitive camera (Leap Motion (Leap, 2014)) attached to the front of an Oculus Rift for interaction in a VR system. The Leap Motion could detect the pose of the user's hands as they were held out in front of them. This was used to control video games that were played from the first person perspective, where users could pick up objects in their environment and interact with them in a realistic manner (e.g. picking up a cup or a knife and completing challenges). The system was accurate in the range allowed by the leap motion (i.e. up to 1 meter).

Mendes *et al* (Mendes, et al., 2014)created a motion capture based interface using a depth sensitive camera to track the head and hands for a stereoscopic tabletop display that presented virtual objects from a user's perspective. The system used two Kinects both looking down on the table, one positioned above the centre and one positioned along the long edge. The authors implemented five different interaction techniques - of which four techniques were mid-air interaction, while the other was a baseline multi-touch technique that used the table top display's touchscreen - and performed a user evaluation. The results suggested that direct mid-air 6DOF manipulation was the best form interaction that was implemented.

2.5.3.2. Multiple Cameras

Another method for capturing the actor's motion for interaction with a virtual set is to create a 3D virtual model of the actor and place them directly into a full virtual environment. In this method multiple cameras, pointed towards a particular volume of space, are used to reconstruct the visual hull of the actor as a virtual object.

A commercial motion capture system based on this method named OpenStage has been developed by OrganicMotion. The specifications of this system are provided in (OrganicMotion, 2012). This system captured the volume of an actor, typically for animation, and used this model to fit a humanoid skeleton into this volume, providing markerless skeletal position and orientation tracking of 21 locations on an actor's body (Brooks & Czarowicz, 2012).

OpenStage was used for markerless tracking of the actor in the virtual studio by Daemen *et al* (Daemen, et al., 2013), which was integrated with a VizRT virtual studio to create a system that allowed interaction and occlusion between the actor and virtual objects.

The interaction example provided in this paper was a sphere that changed colour when the actor was sensed as intersecting its bounding volume. The authors state that this is only a basic interaction and the data provided by OpenStage could be extendable to other types of interaction. Although no publication has yet been made, they have since introduced an interaction where the actor is able to throw a Frisbee, which is matched to the location of one of the hands and as the actor makes a throwing gesture it is released and continues along the trajectory of the gesture (Daemen, et al., 2013).

The authors also demonstrated that OpenStage could be used to map the motions of the actor to a virtual avatar (a virtual model that has the form of a human), who would mimic their movement live. Again this appears to have been extended on further by the authors who have created a "cyborg", where the actor is partially covered with rigid virtual clothes that follow their motion (Daemen, et al., 2013).

The system was assessed by 11 experts who provided their opinions on the functionality of the system via a questionnaire. The experts generally agreed that integrating an OpenStage based interactive system into an existing virtual studio would interfere with its operation (e.g. via changes to lighting set-up to accommodate both television and OpenStage appropriate lighting and the extensive camera arrangement). The experts were also critical about the ability of the actors to navigate the virtual set and orientate themselves, as adequate feedback was not provided. However, the level of interaction that this virtual studio provides was responded to positively by experts, who also suggested that the motion capture could even be applied to more subtle interaction effects (e.g. creating virtual footprints).

2.5.4. Summary of Interaction

Attempts at creating a more 'direct' interaction have been investigated, but have various issues associated with them. Physical devices such as wired gloves will be perceptible to the viewer and are difficult to conceal (Minoh, et al., 2007); Tracked devices allow single-handed interactions, but the interactive virtual objects are tethered to a physical device that would be difficult to dispose of, place down in a scene or interact with any other object (VizRT, 2014). Depth-sensitive cameras have been used to allow interactions, but only to the extent of adjusting an object's motion path based on collision detection (Kim, et al., 2006). Since that study the field of motion capture using depth-sensitive cameras has rapidly advanced as commercial applications (such as videos games) have become established, but again these are limited to interactions based on collision detection (zLense, 2014) or selecting options (Mammoth Graphics and Kenziko Ltd, 2014).

Chapter 3 : CASE STUDY AND ANALYSIS

This chapter presents a discussion on the current limitations of virtual studio technology in terms of feedback, occlusion and interaction using heuristic evaluation techniques. Heuristic evaluation (Nielsen & Molich, 1990) is an informal method of identifying common issues with computer user interfaces, where experts participate in a series of tasks and describe the problems that were encountered using a set of heuristics. A heuristic is a consideration that should be made when designing a system, where problems are identified in relation to that consideration's description; for example:

"***Visibility of system status****: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time*" (taken from Nielsen (Nielsen, 1994))

This heuristic describes to an assessor the level of information that the user is given during any given task. Given this description the experts will have to identify via experience with the system any problems that hinder the successful realisation of this feature and design an appropriate solution.

Typically heuristic evaluation is used because it is faster than standard user measurement and evaluation techniques, yet is able to produce many of the same results (although it is usually not as comprehensive). This usually requires several trained expert evaluators (3 to 5), but can be conducted with one - albeit at the expense of more detailed findings (Nielsen, 1994).

Originally ten heuristics were designed by Nielsen (Nielsen, 1994) to aid user interface design (1. Visibility of system status; 2. Match between system and the real world; 3. The system should speak the user's language; 4. Consistency and standards; 5. Error prevention; 6. Recognition rather than recall; 7. Flexibility and efficiency of use; 8. Aesthetic and minimalist design; 9. Help users recognize, diagnose, and recover from errors; 10. Help and documentation).

Nielsen's ten heuristics are still in use today for assessing user interfaces, although many researchers have developed their own task-specific set of heuristics for analysing specialised user interfaces. A set of specialised heuristics can be seen in the work of Sutcliffe and Gault (Sutcliffe & Gault, 2004), who used them to classify the key shortcomings of interaction in virtual environments. For example, the authors used the heuristic *Navigation and orientation support*, which was described as "The users should always be able to find where they are in the VE and return to known, pre-set positions. Unnatural actions such as fly-through surfaces may help but these have to be judged in a trade-off with naturalness". Upon investigation the authors concluded that the "Ability to walk through walls caused disorientation", which they suggested could be solved by adding movement constraints to the user.

From the twelve heuristics they designed, Sutcliffe and Gault were able to produce an analysis of the problems facing current interactive virtual reality techniques and were able to describe the requirements that need to be considered in future research efforts.

The success of Sutcliffe and Gault's use of heuristic analysis prompted the adoption of a similar approach in our work. We created a set of heuristics, identified the associated problems via the enactment of common scenarios and described the requirements for any further developments; from this the appropriate solutions were developed. The heuristic evaluation was conducted with 3 experts using the sample application of a virtual studio based teleseminar[2] on dental anatomy, as a teaching scenario similar to this is considered a probable use of the interactive virtual studio.

These scenarios came from work by Professor Cham Athwal and Dr Bruce Elson, who were working with Kings College dental hospital for the development of haptic based techniques for dentistry students (Elson, et al., 2009). As part of this project they were interested in using the virtual studio to investigate its potential to teach dentistry principles by having their teacher interact with 3D dental models. The author of this thesis supported this study as the technical operator of the virtual studio.

The issues were analysed by enacting these teaching scenarios in the virtual studio using the current level of technology, while trying to consider the needs for developing more advanced systems. The case study used was a dental anatomy seminar, which was an application well suited for heuristic analysis as it required multiple interaction events to be performed (such as selection and manipulation of a virtual object), covered a wide range of necessary occlusions and also required the actors to position themselves based on visual feedback. Examples of these scenarios and some findings (discussed formally in sections 3.1 to 3.3) are provided.

Figure 3-1 (page 41) shows an image from a heuristic analysis scenario that aimed to analyse the requirements for visual feedback (the feedback medium that was selected for this work). Here an actor attempted to place their hand on specific locations on the virtual object (the mandible) using a standard monitor-based visual feedback method. The image presented to the actor was flipped along the horizontal axis. This and similar scenarios informed the feedback requirements for correct actor

[2] Teletraining (a.k.a. Teleseminar or Teleconference) is a remote audio-visual method of teaching, where the students in remote locations learn by watching content distributed to televisions (e.g. Open University) or more recently computers (e.g. Khan Academy). The advantage of teletraining is that students do not need to assemble in one location for a lecture, and are free to pursue their education from anywhere in the world. One of the key benefits of teletraining since the introduction of the internet is the ability for the student to interact with the teacher in real time. Over the internet the student may ask the teacher to provide more information on a particular feature or artefact. Using the interactive virtual studio the teacher would have access to a large library of virtual models that they can bring into the scene, manipulate and dismantle to answer any queries.

orientation (e.g. identifying the surface location of the virtual object) based on the current level of visual feedback.

It was found that when using a static monitor for interaction the actors would frequently turn their heads and briefly lose their visual guidance, meaning that they were unable to orientate themselves correctly for completing tasks such as accurately pointing at the virtual objects. It was also found that when the actor looked at the monitor and not the virtual object it appeared false from the perspective of the viewer, as typically people look at the objects they are pointing towards or interacting with.

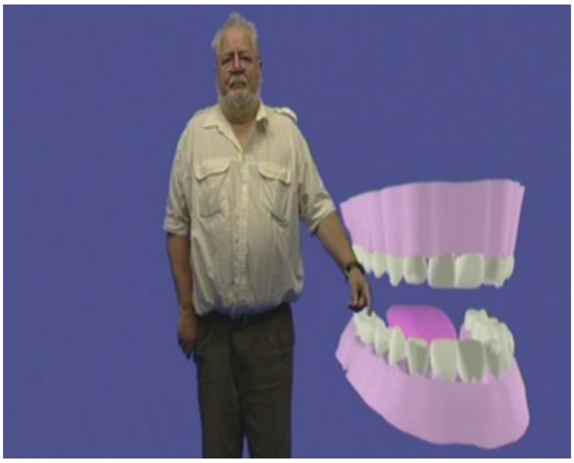


Figure 3-1. Screenshot of scenario to examine the requirements for feedback

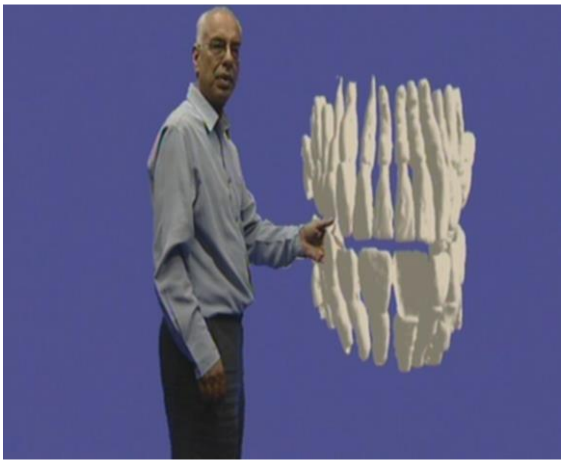


Figure 3-2. Screenshot of scenario to examine the requirements for occlusion

Figure 3-2 shows an image from a scenario that aimed to analyse the requirements for occlusion. In this case the front teeth and back teeth were separated into two object groups, with the actors placing their hands near to the surface of certain teeth as they explain what they are. The hands could be placed in front of all the teeth, behind the teeth or in between the two groups. The manner in which the teeth appeared (relative to the actor's hand) was manually adjusted by an off set operator in real time. This and similar scenarios informed the requirements necessary for the construction of an occlusion system.

It was discovered that the actors would often place their hands in front of or behind the virtual object with a high frequency, requiring any occlusions to ideally be defined in real time (or shortly before they happen). It was also found that in some cases they would place their hand in gaps that were present in an object (such as in the gap between teeth in a model of the mandible), meaning that the same virtual object would have to both occlude and be occluded by the actor. In addition, it was found that actors frequently attempted to place one hand behind the virtual object and one hand in front, meaning the object should appear between the two hands.

Figure 3-3 shows two images from a scenario where the actor was attempting to 'mime' an interaction with a pre-animated virtual object. In this case the actor was asked to follow the object with both hands as it moved up and down along a vertical path, attempting to appear as though they are holding and moving it. This and similar scenarios informed the requirements for interaction (as well as visual feedback in the context of interaction).

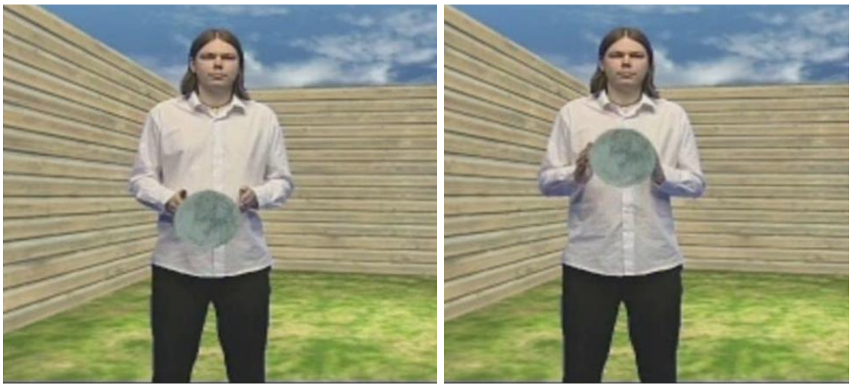


Figure 3-3. Screenshot of scenario to examine the requirements for bimanual interaction

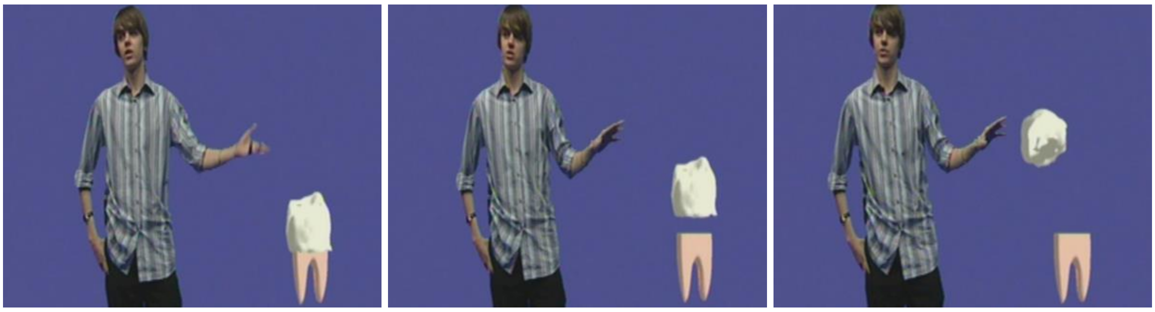


Figure 3-4. Screenshot of scenario to examine the requirements for interaction and feedback with a triggered interaction.

Figure 3-4 shows an image from a scenario where an actor made a gesture to move the virtual object. In the case presented the actor 'triggered' the interaction by making an upwards motion with one hand, at which point an operator off set initiated a predefined animation. When the tooth was floating, the actor then attempted to place one hand near the surface of the object. This and similar scenarios informed on requirements for more abstracted forms of interaction and actor orientation when performing with an object that is in motion.

From these interaction scenarios it was found that the actor struggled to place both hands consistently in line with the surfaces of the virtual object and that spaces between them appeared to be false (this effect is quantified in section 8.5.1.). The hand-surface accuracy was particularly noticeable with two handed interactions where the variance between the hands was also a factor. It was also found that pre-animated interactions were limited in terms of the possible positions of objects, highlighting the need for a versatile interaction system that allows the virtual objects to be moved freely.

The following sections provide a summary of the findings from the heuristic investigation and the requirements of the system that have been derived.

3.1. Visual Feedback Findings

#1: Visual feedback should be ubiquitous to the actor

Heuristic: The actor should have visual feedback available to them at all times during the production.

Problem: The static nature of the off-screen monitor means that a trade-off exists between correct orientation with respect to virtual objects (e.g. placing hands near the surfaces for interaction) and maintaining a correct gaze with a virtual object in cases where the actor is facing away from the feedback monitor.

For example, in cases when the actors want to appear to maintain gaze with a specific virtual object, they will find it difficult to orientate themselves correctly if they do not have a feedback monitor in their field of view. On the other hand, if the actors wish to orientate themselves with regards to the virtual object they would need to look at the nearest feedback monitor, which will cause them to appear as though they are looking away from the virtual object.

Requirement: This trade-off can be reduced by implementing a form of visual feedback that is in the actors' field of view regardless of where they are looking.

#2: Visual feedback should be able to support different tasks

Heuristic: The visual feedback should support the range of tasks the actor would face in the interactive virtual studio; including orientation, gaze correction and interaction.

Problem: Currently, the main form of visual information presented to the actor in the virtual studio is the standard broadcast output, which is then mirrored on the horizontal axis. The broadcast output would help the actors orientate themselves in a manner appropriate for the home viewer. However, this does not appear to be beneficial for some tasks, such as maintaining a correct gaze with a virtual object, where an alternative form of visual feedback presented from the viewpoint of the actor may be more suitable.

Requirement: A range of feedback options should be provided to the actors, allowing them to select a visual feedback modality they feel would be suitable for the task they are trying to achieve.

3.2. Occlusion Findings

#1: Should be compatible with existing layer-based systems

Heuristic: The occlusion system should support the common layer-based method of the virtual studio (as described in Section 2.4, page 26).

Problem: The current standard method of creating occlusion in the virtual studio is to use a series of virtual layers (containing the virtual set) that sandwich a video layer (containing the actor), and any approaches to occlusion would need to be compatible with this structure.

Requirement: Any virtual studio occlusion system developed should be based on the layer format.

#2: Occlusion should be locked while the actor is directly behind or in front of a non-interactive object

Heuristic: While the actor is positioned either directly in front of or behind a non-interactive virtual object, the relative layer positions between the actor and that object should remain locked until they are clear of each other.

Problem: It would appear unrealistic if the actor was positioned behind a virtual object, moved forward and then suddenly appeared in front of it. This would give the appearance of the actor passing through the object as though they were a ghost; this effect would appear unrealistic and therefore unsuitable. Further evidence for the importance of this requirement is provided in Chapter 8 (page 168).

Requirement: While the actor is positioned behind or in front of the virtual object the relative occlusion properties between the two must remain fixed.

#3: Occlusion should occur in real time and reliably

Heuristic: The occlusion should be defined correctly in real time.

Problem: It was found that occlusion events can occur with high frequency and complexity, making any manual intervention unreliable.

Requirement: The occlusion system should be fully automated to support the high frequency of the occlusion events and have low latency.

#4: Occlusion system should support different occlusion types

Heuristic: It was found that in a layer-based system there would not be a universal occlusion method suitable for all object types.

Problem: Different objects would hold different occlusion requirements. For example, in the case of a solid object, such as a wooden block, it should be rendered entirely on either the foreground or the background layer, but in the case of a liquid object, such as a pool of water, the water behind the actor should be rendered in the background layer and the water in front should be rendered in the foreground layer. Three key types of occlusion were identified: absolute occlusion, actor intersection occlusion and object intersection occlusion – these are described in chapter 5.

Requirement: The different forms of occlusion that exist in the virtual studio should be identified and a layer-based processing technique designed for each.

#5: The occlusion must factor in the extremities of the actor's body

Heuristic: The occlusion system must be based accurately on the part of the actor that is most likely to occlude it.

Problem: Defining occlusion based on some point near to the centre of the actor, such as the shoulder, may result in an incorrect occlusion if another point, such as the hand, is the closest point to the virtual object

Requirement: The closest tracked point from the skeletal motion capture should be used as the basis of occlusion.

3.3. Interaction Findings

#1: The interaction should be suitably constrained

Heuristic: The interaction should be as simple and intuitive as possible for the actor to make.

Problem: The virtual object is different from a real object as it has no physical boundaries, which makes successful interaction a difficult task. As a result the actor may make movements that result in an unintended object motion. For example, an object that should have a fixed range of motion, such as a globe that should only rotate along one axis (example presented in Figure 6-10, page 81), could accidentally be moved or rotated along an unrealistic if the actor is given total control over the range of motion.

Requirement: In order to reduce the number of errors made on the part of the actor, the interaction should be simplified where possible.

#2: The interaction should appear realistic

Heuristic: The interaction should appear as realistic to the viewer as possible.

Problem #1: Because the virtual object has no physical surfaces, actors are likely to misestimate the location of the object's surface, resulting in them either underestimating or overestimating the size of the virtual object. This issue is discussed further in Chapter 7.

Problem #2: The tracking system and the graphical render of the virtual set both add latency to the system, which could result in the virtual object lagging behind the movements of the actor.

Requirement #1: The actor should also be able to estimate the surfaces of the virtual object accurately.

Requirement #2: The interaction should appear to react appropriately to the movements of the actor with imperceptible latency. A delay should be added to the video feed from the studio camera to synchronise it with the graphics.

#3: The actors should be able to position the object as they wish

Heuristic: The actors should have adequate control over the motion of the virtual object, allowing them to position it however they desire (assuming no conflict is present with Heuristic #1).

Problem: The actors need to manipulate the virtual object and position it how they wish it to be seen by the audience. Pre-animated virtual objects only allow a very limited range of positioning. Interaction systems based on physics engines would add an element of unpredictability that would not be well suited to the live environment of the virtual studio.

Requirement: The actors must be able to manipulate the virtual object to any position they desire with a high degree of confidence (assuming no conflict with heuristic #1).

Chapters 4, 5 and 6 will discuss the developments that were made based on the findings of the heuristic study.

Chapter 4 : SYSTEM HARDWARE DESIGN

This chapter disucsses the hardware design of the virtual studio system implemented in this work. Section 4.1 describes the archetecture of theinteractive virtual studio used in this work. Section 4.2. discusses the motion capture system selected for this system and the reasons for its selection. Section 4.3. describes the feedback system that was developed to provide the actors with real time visual feedback and aid them with interaction tasks.

4.1. Interactive Virtual Studio Archetecture

To ensure a high degree of transportability into other virtual studio systems, the interactive virtual studio archetecture used in this work is based on that of a standard system (see Figure 2-9, page 14 for a description of a standard archtecture). In this case an Orad virtual studio is used as a representitive system, utilising an Orad HDVG (Orad, 2012b) for rendering and compositing the graphics in real time and Orad 3Designer 3.6 (Orad, 2012a) for the virtual set design and real time control of the virtual objects. The floorspace of the virtual studio is 18m^2 and uses a 3 studio camera set-up (although only one is used at any time). The orientation and location of the 3 cameras is found using a 4x3m blue tracking grid situated along the back wall (seen in Figure 4-1, page 47). The motion data and the interaction with the virtual objects was computed using a 3.6GHz Pentium 4 PC with 2GB of RAM and a Windows XP 32-bit operating system. One key limitation of the virtual studio system used in this work (and of some virtual studio systems in general) was that information on the hull of the virtual objects was not accessible during a production. Only the 6DOF location and orientation of the virtual object and the bounding box, on which all interactions were based, were available (the bounding box describes the maximum length, width and height of the virtual object) – on which all interactions were based. Despite this limitation, the methods described in this work could be adapted and transported to systems where detailed information on the objects’ hull is available.

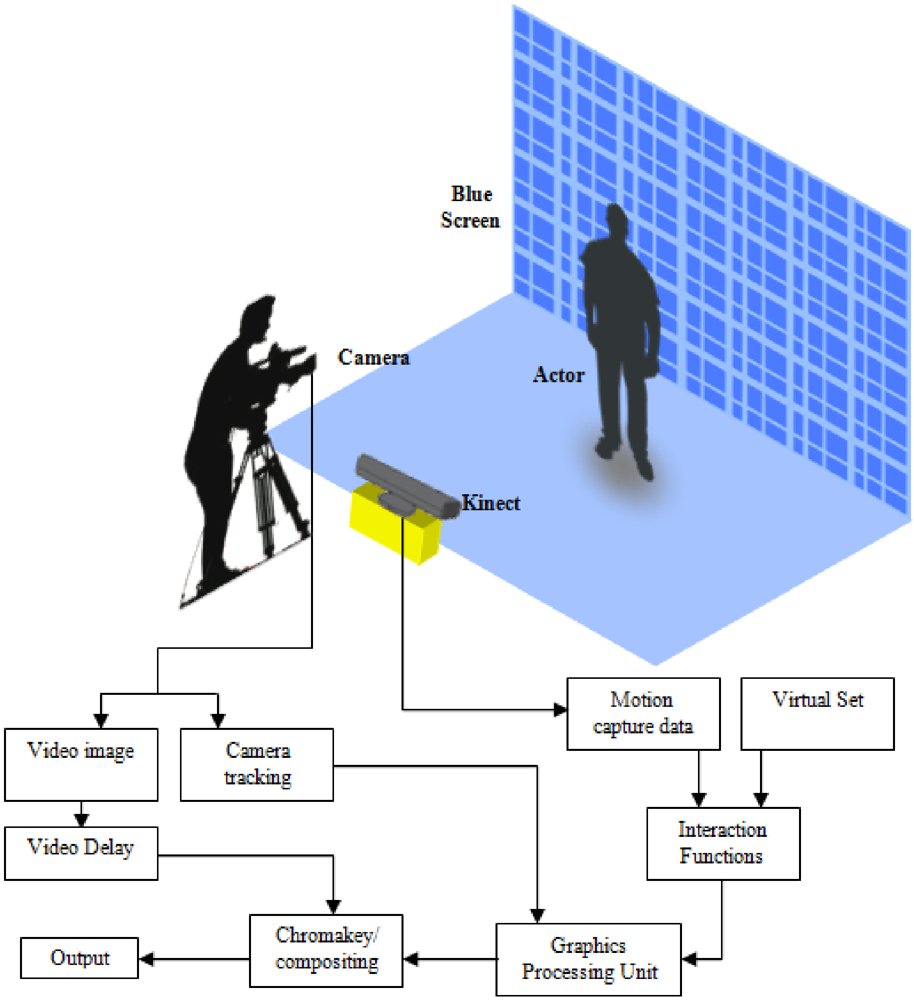


Figure 4-1. Architecture of the Interactive Virtual Studio

Figure 4-1 presents a diagram of the interactive virtual studio system. There are four key elements in the studio, the actor, the tracking device (here a Microsoft kinect), studio camera and a blue screen. The camera captures the video image of the actor standing in front of the blue screen, which is used for two purposes: using the captured image for chromakey and analysing the pattern of the blue screen to calculate the location of the camera. The latter is used by the Graphics Processing Unit (GPU) to render the appropriate perspective of the virtual set. The tracking device (a depth-sensitive camera, here a Kinect) provides the skeletal motion capture data of 21 joints on the actor, which is used in conjunction with data from the virtual set to calculate the various interaction functions (occlusion and interaction). The mode of motion capture data is described in section 4.2. After the calculation of the interaction functions the virtual set is updated accordingly and rendered using the GPU.

The next stage is to apply chromakey and composite the studio camera image in with the virtual set. However, before this stage an artificial delay is added to the video image from the studio camera to compensate for latency produced by the Kinect and the GPU. A 40ms latency for the Kinect and a 120ms latency for the GPU to render the virtual set were measured. A 120ms artificial latency was added to account for the render only, producing a 1 frame error. The latency added by the calculation of the interaction functions was negligible. Adding this artificial delay ensures that the movements between the actor and the virtual objects are synchronised. Once the final scene has been composited the broadcast output is produced.

4.2. Motion Capture

4.2.1. Selected Motion Capture

A crucial consideration when dealing with visual effects technology for entertainment is the concept of “suspension of disbelief” (Martin, 2012): the viewers are willing to make concessions on what they will believe is happening to enhance their entertainment, but this will break down if they see how it is achieved. One of the key issues is that to achieve feedback, occlusion and interaction, the motion of the actor must be 'captured'. Motion capture is a well-established field that uses many techniques to measure the overall position of a person's body. However these are typically used for applications such as virtual reality and medicine, where the use of visible hardware is permissible. A visible motion capture system, for example employing mechanical devices (e.g. (Young, et al., 2010)) or marker based optical methods (e.g. (Weber, 2008)), will almost certainly break the audience's suspension of disbelief in a virtual studio because the viewer will see how the actor is being tracked, so finding an imperceptible tracking system is mandatory. Therefore methods of capturing the motion of the actor must be investigated for their suitability for virtual studio productions and their effect on suspension of disbelief (i.e. Imperceptibility), an opinion that is shared by Zerroug *et al* (Zerroug, et al., 2009).

Appendix #A discusses an investigation into finding a suitable method of motion capture for use in the virtual studio using heuristic analysis. The results of the analysis are similar to the recommendations by Zerroug (Zerroug, et al., 2009) but were discovered independently, thereby validating these findings. The investigation yielded the following 5 (abridged) requirements of a motion capture system:

#1: Must be imperceptible to the viewer - The motion capture system must be imperceptible to the audience, else it could compromise their suspension of disbelief.

#2: Must have low impedance on actor motion - The method of motion capture must not impede the motion of the actor, nor the natural motion of the actor interfere with the quality of the tracking.

#3: Must provide sufficiently detailed motion capture data - The motion capture system must be able to provide 3DOF location data for all joints that are relevant for interaction; primarily the hands.

#4: Must be low latency - For the benefit of the actor the end-to-end latency of the system should be as low as possible to provide quick visual feedback. As the motion capture is a major factor in this, it should be low latency.

#5: Tracking system must accommodate acting space - The motion capture system must accurately support the range of the acting space the actor requires.

From the results of the investigation into suitable motion capture techniques, a depth-sensitive camera based approach was selected. An example of skeletal motion tracking using this method with annotated joints is presented in Figure 4-2. The device used in this study is the Microsoft Kinect (Microsoft, 2011), which offers sufficient and cost effective performance. The findings would be portable to higher performance depth-sensitive cameras.

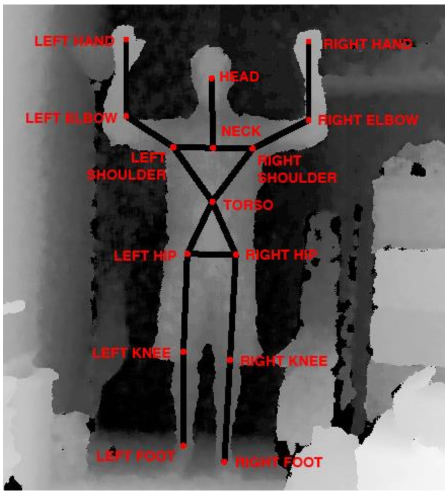


Figure 4-2. Annotated Kinect Skeletal tracking (image courtesy of Greg Borenstein)

The tracking of the skeletal joints was achieved using OpenNI (version 1.4) (Primesense, 2011), which allows the 3DOF pose of the actor's joints to be located relative to the Kinect unit. Typically this method is used for applications where the actor is standing in an upright position approximately 1.2 to 3.5 metres away from the device and unoccluded (Pece, et al., 2011), which are conditions congruent to acting in the virtual studio. The Kinect unit is placed in a discrete location 1-3m away from the main acting space and outside of the camera's view.

The data captured using this arrangement can be used for creating simple single-handed and bimanual interactions or occlusions with a virtual object by using the palm locations of the actor, or other joints if necessary.

4.2.2. Implementation into Final Studio Plan

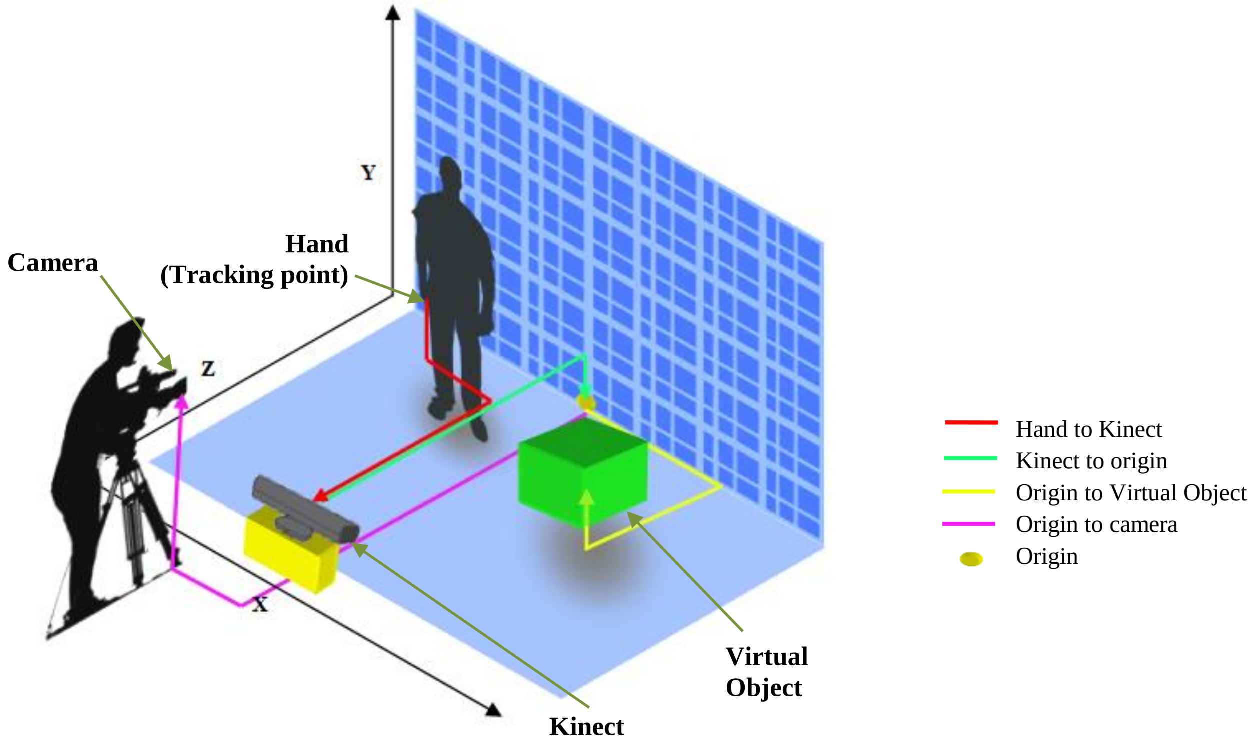


Figure 4-3. Location of elements in the virtual studio relative to an origin.

In the system presented in this work, the location of each element in the virtual studio is defined by the distance along the X, Y and Z axes from a common origin point. Figure 4-3 shows how the various elements of the virtual studio (Tracking Point, Virtual Object, Studio Camera, Tracking Camera/Kinect) are measured in relation to this common origin, which is represented by the gold oval. This origin is positioned at the bottom of the tracking grid, at the midpoint along the X axis.

The yellow line in Figure 4-3 shows the location of the centroid of any virtual object measured relative to the origin, while the features of the virtual object such as the surface can also be measured relative to this point by adding additional dimensions. The pink line illustrates the location of the tracked camera lens from the origin.

The joints of the skeletal tracking cannot be measured directly from the origin, as they can only be measured relative to the location of the Kinect. To calculate the location of the skeletal tracking joints from the origin, the location of the Kinect relative to the origin must be taken into account by using Equation 4-1.

Here K_{xyz} refers to the location of the Kinect from the origin as indicated by the green line in Figure 4-3. T_{xyz} refers to the location of the tracked point measured from the Kinect as indicated by the red line (in this case the tracked point is the hand of the actor). The result, T'_{xyz}, refers to the location of the tracked point from the origin.

$$T'_{xyz} = T_{xyz} + K_{xyz}$$

Equation 4-1. Calculating the location of a tracking point from the origin

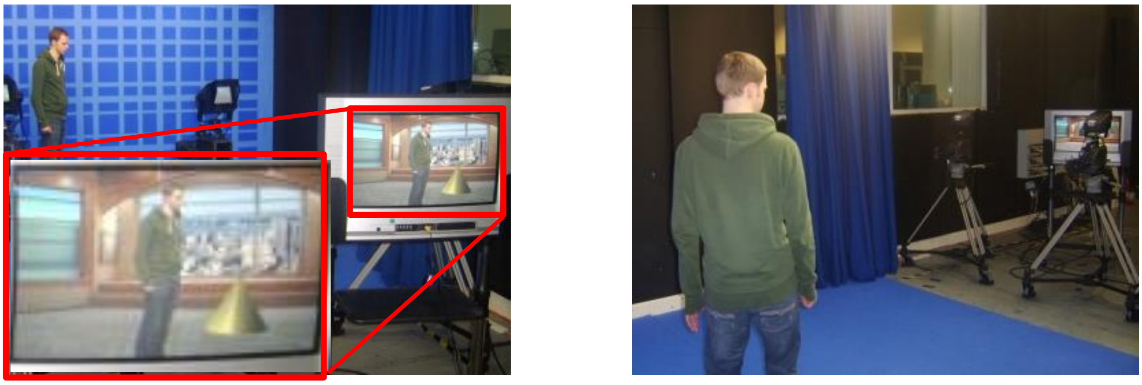
Calibration was achieved using the standard calibration processes of the Orad virtual studio and the Kinect, combined with pre-measured geometric knowledge of the system layout (i.e. the kinect's location relative to the origin). Using these measurements, each object and the tracking point of the actor in the virtual studio can be found as a relative location from the common origin.

4.3. Feedback System

The work in this section was presented at ACM Designing Interactive Systems 2012 as "ScaMP: A Head Guided Projection System" (Hough, et al., 2012a).

As part of the system design a ubiquitous feedback system was developed to help aid the actor in the virtual studio. To the actors the virtual objects that are used to construct the virtual set are invisible, as they can only see the blue or green studio space. To counter this, out-of-shot static video monitors are typically placed around the studio that show the actor the mixed output the viewer would see (albeit mirrored, which is more intuitive for the actor to base their movements on). However, as previously stated the static nature of the monitors can cause difficulties when attempting to locate or interact with virtual objects, particularly if there is no monitor directly in the actor's line of sight.

To illustrate this point Figure 4-4b shows the actor looking at a monitor placed outside the TV camera's view for feedback while trying to appear to stand near the yellow cone ready for interaction; as a result the actor's gaze and coordination appears incorrect, looking off set as opposed to looking at the yellow cone (as depicted in the red square in Figure 4-4a).



a: Off set monitor (turned to camera to illsutrate feedback)

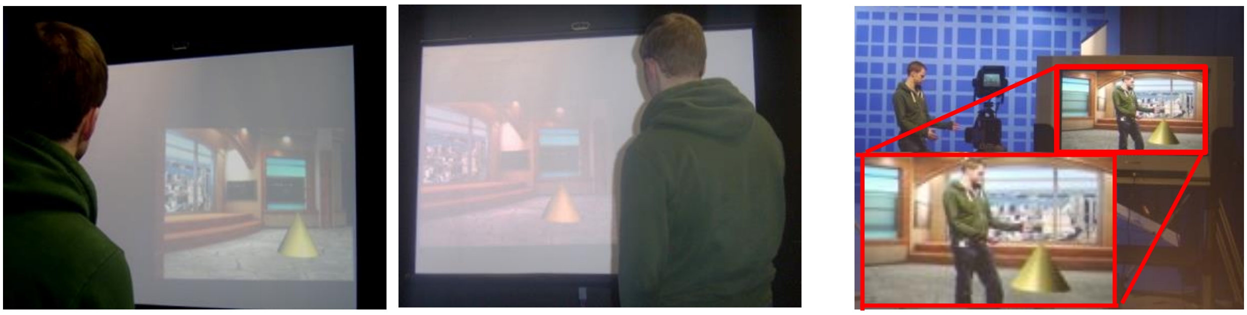
b: Actor in studio

Figure 4-4. Monitor technique

CAVE style ubiquitous projection was reported to have a positive effect on the actor's ability to recognise the location of moving virtual objects over static monitors, allowing them to maintain a correct gaze with a virtual object and solving the problem (Thomas & Grau, 2002). However, these systems come with large resource and space requirements, which compromises the compact nature of virtual studios.

This led to the creation of ScaMP (Scanning Mirror Projector), a device designed to deliver visual feedback to the gaze location of an actor, providing ever-present visual information in a compact form. ScaMP is formed of a single "steerable" projector fixed to the lighting gantry of the studio and a camera for tracking the orientation of the actor's head. The image projected by ScaMP is corrected in real time to account for any warping that may occur when it is projected at arbitrary angles. Three distinct feedback modalities are also provided to aid the actor with a range of tasks (Standard Broadcast, Point of View render and Teleprompter). It was developed for a single planar surface - a wall - with the possibility of being extended to multiple planar surfaces.

Figure 4-5 shows the same scene as Figure 4-4 with the actor using ScaMP instead. Figure 4-5a and Figure 4-5b show ScaMP projecting a Point of View (POV) image (described in 4.3.2.3. Feedback Modalities, page 58) of the virtual set to different areas of the planar projection surface (placed out of shot) depending on the gaze point of the actor. Figure 4-5c shows how the actor is able to correct their gaze and orientate themselves. To ensure portability for non-interactive applications, ScaMP is designed to work independently of the interaction system described in this thesis.



a and **b.** ScaMP Feedback at different gaze locations

c. Actor in studio

Figure 4-5. ScaMP technique

4.3.1. Background of Steerable Projection

The visual feedback device developed for this study was based on an existing technology known as 'Steerable Projection', which is where the path of a projection is diverted from its original course to a new one. Steerable projection systems have previously been used for a variety of purposes such as real-world Augmented Reality (Ehnes & Hirose, 2006), virtual character projection (Ehnes, 2010) and as a form of computer display that can be projected to several pre-defined locations (Pinhanez, 2001). The advantage of using a steerable projector in each of these applications is the ability to provide information to a user in multiple real space locations without the need for user based hardware (e.g. Head Mounted Projectors). In the context of the work presented in this thesis the lack of user based hardware is advantageous as it is imperceptible to the viewer while remaining ever-present to the actor.

Two main forms of steerable projection technology currently exist: servo-guided projectors and mirror-guided projectors.

Servo-guided projection is based on moving the whole projector using two servo motors to adjust its pan (horizontal movement) and tilt (vertical movement), as used in the work of Ehnes *et al* (Ehnes & Hirose, 2006). Typically these systems have a wide range of motion (up to 360° pan), but are required to move the heavy projector and as such are limited in their agility of movement. These limitations in movement agility would become more severe as larger and more powerful projectors are required, which will be the case with ScaMP as the projection needs to be powerful. With head movements a high speed of rotation can occur and so agility is a necessity.

Mirror-guided projectors are based on moving a servo controlled mirror fixed in front of the projector lens, as used in (Pinhanez, 2001). Whilst the movement of the mirror is limited to the range of the servo due to the constraints of the frame it is mounted on, it is a lighter mass to move and would consequently provide better agility for a steerable projection system based on head movement.

ScaMP utilises the mirror-guided projection technique to project visual information onto the planar surfaces of the walls in the virtual studio as it is a more appropriate method for the requirements of this work.

Steerable projection has been employed for similar purposes to ScaMP in previous work, each providing visual information to a user based on their location as discussed here.

The Everywhere Displays Projector (Pinhanez, 2001) projected visual information onto one of multiple pre-defined planar locations. The Everywhere Displays Projector determined which surface to display information on based on the location of the user, relying on a 3D geometric model of the environment (including specified surfaces practical for projection) and the tracked location of the user for the correction of the image. The Everywhere Displays Projector was designed to be used with a Personal computer, with the display projected to one of multiple pre-set locations instead. ScaMP uses a similar concept of user guided display, but is designed to be able to project an image at any arbitrary location instead of pre-defined ones.

Lee et al (Lee, et al., 2009) developed a display for ‘Intelligent Space’ which has the ability to provide visual information using a roaming robot with a servo guided projector fixed to it. The ‘Intelligent Space’ is an environment that contains an array of cameras which are used to construct a model of the environment, including finding which way a person is facing and any suitable projection surfaces. The robot then moves to a suitable location near the person and projects onto a suitable surface in front of them. The purpose of this display is to provide instructions for people as they move around the environment (e.g. Aiding users by projecting directions to their intended destination on the floor in front of them).

‘Projected/Augmented Reality’ (Ehnes & Hirose, 2006) (Ehnes, et al., 2004) is a form of user guided projection system based on the location of an AR marker. The system was a steerable projector/camera combination that augments real world objects with projected graphical information. An AR marker, detected using the camera, is placed in the real environment which then denotes where graphical information is to be projected. Whilst not directed by the movements or location of the user, it allows the user to manipulate the location of the projection in real time by moving the marker, which is similar to ScaMP in that visual information can be projected to any arbitrary location as defined by the user.

A similar form of steerable projection to ScaMP guided by the gaze direction of a user was developed at the University of Texas at Austin as part of their ECE Senior Design Contest (ECE, University of Texas at Austin, 2011), although no formal publication has been made. The system appears to require the user to sit in a single static position and it does not appear to make any image corrections.

Although several similar systems do exist, they do not meet the specific requirements of feedback in the virtual studio as well as ScaMP aims to, with gaze directed and imperceptible feedback.

4.3.2. Development

The setup of ScaMP is typically a planar surface with a camera affixed to it for head tracking, with the ScaMP projection unit placed in the centre of the performance area (ideally fixed to the lighting gantry in the ceiling to minimise projection occlusion from the environment or actor). The ScaMP projection unit projects onto the planar surface the tracking camera is fixed to.

Figure 4-6 shows the ScaMP projection unit, which is composed of a standard video projector, a frame to support a pan/tilt servo arrangement and a mirror fixed to the lower servo that reflects the projected image. The camera portion of ScaMP was a standard webcam mounted to the wall intended for projection. The current version of ScaMP is intended to be used over a performance area of around 6 square meters, limited by practical aspects of both camera-based head tracking and projection intensity.

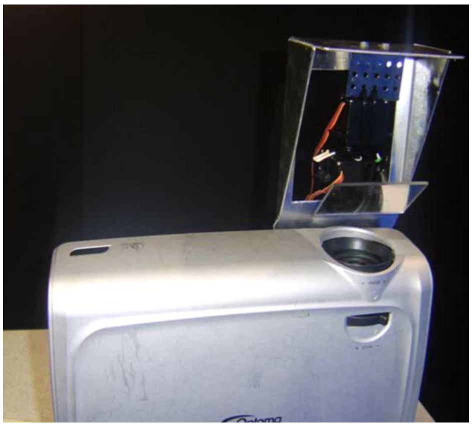


Figure 4-6. The ScaMP projection unit

Figure 4-7 illustrates an actor using ScaMP in a virtual studio environment (compared to a static monitor). While travelling from location A to B the actor's view of the virtual object will change respectively. Consequently the gaze direction moves from point 1 to point 2 while looking at the virtual object. The tracking camera will register this motion and the servo guided mirror fixed in front of ScaMP's lens will steer the projection from point 1 to point 2 in correspondence. With a conventional system, the monitor is only visible when the actor is at location A, whereas the ScaMP system ensures the feedback is visible at A and B, as well as locations in between.

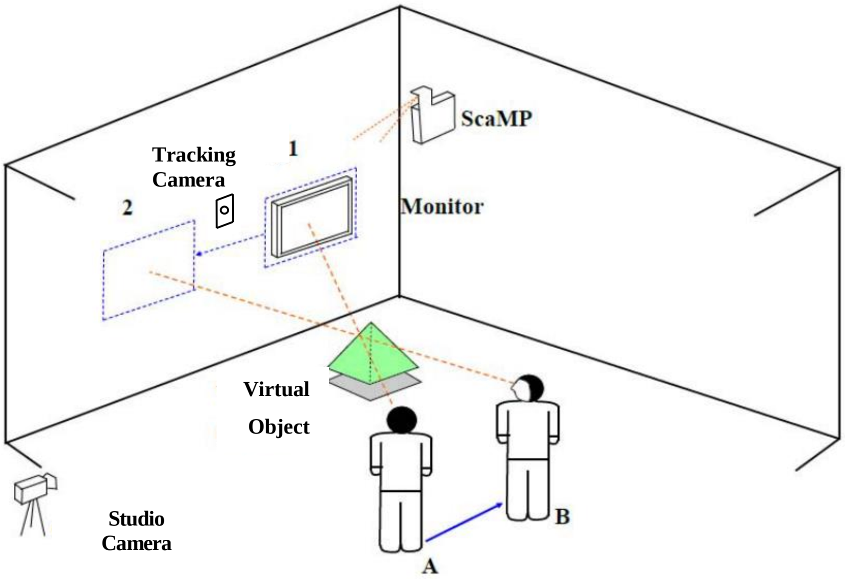


Figure 4-7. Illustration of an actor looking at a virtual object whilst travelling from location A to B, with their gaze moving from point 1 to point 2 and ScaMP projecting accordingly (represented by the dashed blue outlines). Include a comparison of monitor and ScaMP feedback systems (not to scale).

4.3.2.1. Head Tracking and Calculation of Gaze Point

To calculate the gaze location of the actor on the planar surface either the orientation of the actor's head or their gaze direction needs to be known, so that the angle of projection can be adjusted to project onto the location they are facing. There are two options to achieve this - Gaze tracking (Morimoto & Mimica, 2005), where the orientation of the actor's eyes are tracked to find the direction that they are looking; or head tracking (Murphy-Chutorian & Trivedi, 2009), which is used to find the direction that the head is facing. This particular version of ScaMP used head tracking as this method works over a larger area than gaze tracking.

The motion capture method discussed in Chapter 3.3 cannot be used to find the orientation of the actor's head, as no rotational information is provided about the head. Consequently an alternative system must be used, taking the requirements for a tracking system into account as discussed in Appendix #A. A suitable alternative markerless system was identified, FaceAPI (Seeing Machines, 2013), which provides 6DOF head tracking data relative to the location of a tracking camera. While it provides robust head tracking for close range applications, it does not track effectively when the actor is occluded, is further than 2 metres from the camera, or is outside the camera's field of vision.

Alternative tracking systems were explored, but they either violated the requirements for tracking systems, as outlined in chapter 3, or did not offer a superior tracking range to FaceAPI. Despite these limitations of FaceAPI, it did allow the validation of a workable system over a small area.

The head tracking data from FaceAPI was used alongside ScaMP's known position to perform two geometric calculations (Equation 4-2 and Equation 4-3) that gave the correct pan and tilt angles for the servo guided mirror to reflect towards the gaze point. In this geometry the displacement of the ScaMP unit relative to the tracking camera must be known and it is assumed the tracking camera is fixed flat against the planar surface and also represents the centre of the planar surface being projected onto.

The notation system used for these equations uses a descriptor and a suffix to denote the axis, where P represents projector position (ScaMP) relative to tracking camera, H represents user's head location relative to the tracking camera (with the suffixes x,y and z representing the distance along that axis) and α represents the pan angle the actor's head and β representing the angle of tilt. The geometric functions can be modified by defining additional camera parameters, such as rotation and location of the camera from the planar surface, allowing the tracking camera to be placed elsewhere if desired.

Mirror Pan = $\tan^{-1}(\frac{Pz}{(Hz\tan\propto)+Hx+Px})$

Equation 4-2. Angle of Mirror Pan

Mirror Tilt = $\tan^{-1}(\frac{Pz}{(Hz\tan\beta)+Hy+Py}) + 45°$

Equation 4-3. Angle of Mirror Tilt (A 45° correction is made to the tilt servo to reflect the projection forward, assuming the lens of ScaMP is pointing towards the ceiling)

4.3.2.2. Image Correction

As ScaMP diverts the projection by arbitrary angles to the gaze point of the actor it results in an image distortion known colloquially as the 'Keystone effect', which is caused by the uneven distances that the light from the projector travels at different points in the image. As a result of this effect the projection area becomes warped, leading to the image being projected in the shape of a trapezoid.

When ScaMP is projecting at arbitrary angles the keystone effect is present on both the Pan and Tilt axes. This leads to a warped projection area equal to the sum of two trapezoids, an example of which is shown in Figure 4-8b (page 57). This type of image distortion would render any projected information difficult to interpret.

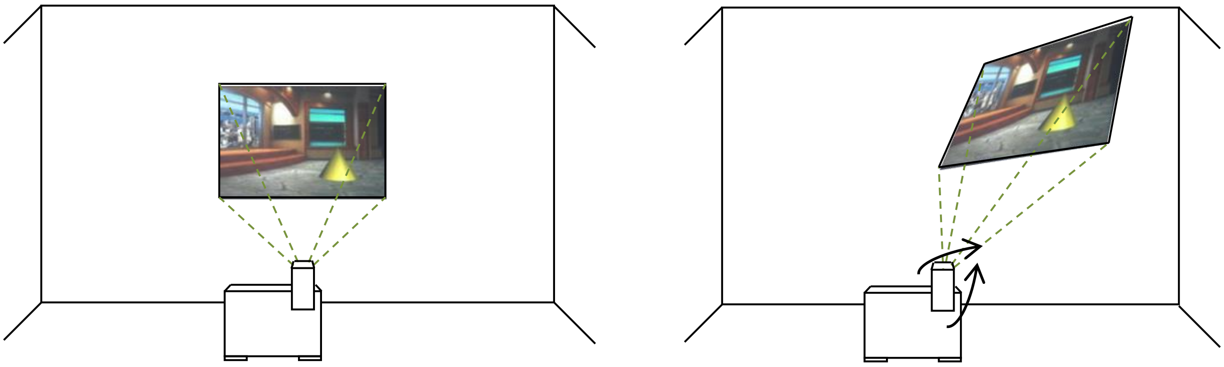


Figure 4-8a. ScaMP projecting at a 0° tilt/pan **Figure 4-8b.** ScaMP projecting at arbitrary angles

Figure 4-8. The Keystone Effect when using ScaMP.

For most projectors Keystone correction is typically handled manually beforehand, as real-time automatic correction is an uncommon task. Real time systems have already been designed to automatically counter the image distortion for projectors positioned at arbitrary angles to the desired projection surface. For example, Raskar and Van Baar (Raskar & van Baar, 2003) presented a hand held projector that contained a tilt sensor that provided the orientation state of the projector relative to the projection surface and a camera that could be used to correct the projection on non-planar surfaces.

As such, the process of correcting a projected image itself is not novel to this work; the following description of image correction is included only to demonstrate how this issue was solved in this particular context. The correction method developed for ScaMP is comparatively simple to other methods as known parameters of the environment are utilised, providing the following assumptions can be made:

1. **ScaMP will be projecting onto a non-complex, planar surface.** Flat walls with no objects placed against them (such as the walls of a virtual studio) will allow the warp of the projection to be predicted.
2. **The location of ScaMP relative to the desired projection surface is static.** The static location of ScaMP relative to the projection surface is used as a base for any calculations.
3. **The mirror's tilt and pan of ScaMP are known at any point.** This information is used for calculating the direction of projection and the warp of the image.

With these assumptions the warping issue is solved by identifying the location of each corner from the projection centre (the gaze point) and subsequently identifying the outline of the projection. From this an appropriate counter transform is applied using the affine transformation function in OpenCV (OpenCV 2.3.1., 2011), which allows the image to be corrected based on a target image state that can fit within the warped projection. The three steps required to complete the correction are:

Step 1: Define a constant target image size and shape.

Step 2: Calculate the area of the warped projection.

Step 3: Calculate how to warp the projected image to fit the target size defined in step 1.

These three steps are discussed in detail in Details of Image Transformation Technique. An illustration of the image being corrected is presented in Figure 4-9, which illustrates the image being projected at arbitrary angles (with the warping represented by the black lines) and the correction applied to it to create a true image. The results of this can be seen in Figure 4-5 (page 52).

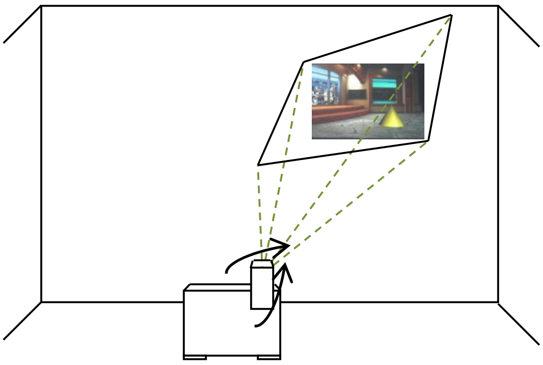


Figure 4-9. Illustration of the image being corrected when projected at arbitrary angles, where the black outline represents the projection area, with the corrected image inside it.

4.3.2.3. Feedback Modalities

Three forms of feedback were developed for ScaMP. These are:

Standard broadcast render – This modality presents the mixed render of the real and virtual sets of the standard broadcast output. This can be used for tasks that require the actor to travel between locations in the virtual set or to assess how an interaction will appear from the viewer's perspective, useful for orientation based tasks such as placing a hand near to the surface of the virtual object.

Point Of View (POV) render – An image of the virtual set dynamically rendered from the actor's perspective which is shown being used in Figure 4-5 (page 52). This modality can be used to aid the actors in assessing the objects directly in front of them for tasks such as correcting gaze. This was achieved by matching the location and orientation of the actor's head to a virtual camera inside a 3D environment identical to the virtual set. In the system presented the game engine of the open source 3D graphics engine 'Blender' was used, which contained a simplified version of the set (Blender Foundation, 2011). The real-time manipulation of the 3D camera inside the environment was made possible by using a Python script that interfaced with the tracking data of FaceAPI.

Teleprompter – This modality provides text to the actor, which can be written live. It can be used to deliver scripts as a normal teleprompter would, or to provide instructions that could be used for interaction or travelling tasks. This was achieved by showing the output of a standard PC running teleprompter software via ScaMP.

4.3.2.4. Extending ScaMP to Multiple Planar Surfaces

Currently ScaMP only supports feedback for a single planar surface. Future work will focus on extending the method to multiple planar surfaces, which could be achieved by switching between the planar surface geometries depending on which way the actor is facing. In this system, instead of defining the geometry based on the location of the tracking camera (i.e. Equation 4-2 and Equation 4-3, page 56), a similar geometry based on a central location on each of the planar surfaces will be used. ScaMP will switch between the multiple surfaces by determining which one the actor is looking towards (thereby basing the various geometric equations on the corresponding central location of the relevant planar surface). This would require a method that only uses one camera that can detect the orientation of the actor's head, even when their face is not directly visible.

Figure 4-10 presents an illustration of this proposed method where a single tracking camera placed off set that can detect the angle of the actor's head as they look in arbitrary locations (illustrated by the dashed red line), with the black dashed lines showing the angle between the actor's head and each corner between the planar surfaces. The surface that the actor is facing is calculated by finding which two corners the actor's gaze is between. In the case of Figure 4-10 the actor's gaze is between corner 2 and 3, indicating they are looking at surface 2.

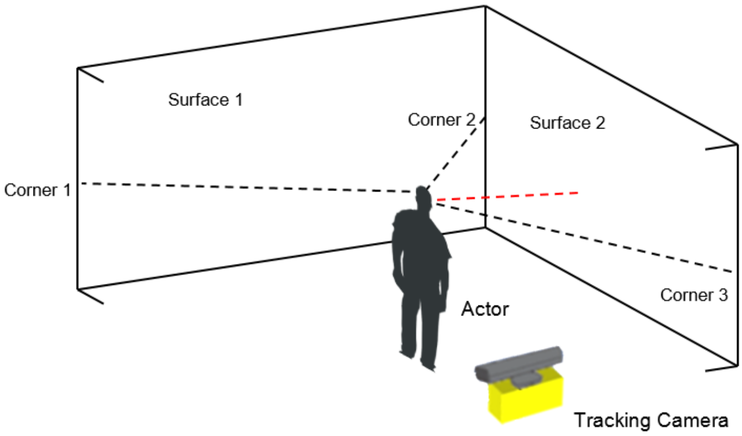


Figure 4-10. Example of detecting the surface the actor is facing with a single camera system, where the black dotted lines represents the angle between the actor and the corners of the planar surfaces and the red dotted represents the direction the actor is facing – in this case Surface 2, which can be calculated as being any angle between corner 2 and 3.

However, accurate head tracking typically requires the presence of salient features such as the eyes and nose to accurately calculate the correct angle of gaze, but as the back of the head lacks any of these features accurate tracking becomes a very difficult challenge. However, due to the current limitation in head tracking technology using imperceptible methods, it was not possible to implement this feature. A proposed process using a hypothetical tracking method that can achieve this level of head tracking is discussed in detail in Extending ScaMP to Multiple Planar surfaces.

Alternative methods could include using multiple tracking cameras or a ScaMP system for each surface, or by calculating the angle the actor is facing by calculating the angle at a tangent to their two shoulders. A workable system was demonstrated using one planar surface and expansion of this system to multiple planar surfaces was outside of the context of this work as developing this feature would have been time consuming and added little to the objective of the goals of this study.

4.3.3. Discussion

ScaMP removed the need for multiple projectors required for the CAVE style visual feedback systems by using a single projector with a servo guided mirror fixed in front of the lens. This system projected feedback in the direction the actor was facing, creating ever-present visual feedback as required by the heuristic investigation. ScaMP functioned as anticipated and worked in execution; although the stability of the projection was an issue, where small and frequent head movements were found to lead to a 'jittery' movement.

To support the findings of the heuristic analysis, three feedback modalities were provided that supported a range of common tasks. These included feedback from a POV perspective common to the CAVE style systems, the standard broadcast output and a teleprompter output. It is noted that when using the POV render ScaMP will only provide the actor with a view of the virtual environment directly in front of them. This means that any objects that would normally appear in the actor's peripheral vision using a CAVE style system would not appear when using ScaMP.

It was found that the 3000 lumen[3] output of the projector used for ScaMP was insufficient for projecting onto some surfaces that do not reflect light well, such as a black or blue wall, particularly in the bright studio environment. To remedy this brighter projectors and more appropriate projection surfaces should be used where possible (e.g. the retro-reflective cloth used in (Thomas & Grau, 2002)). In this work placing a white projection screen over the black surfaces of the studio's side walls was favoured.

The light projected from ScaMP was found to cause two issues. The first was that the light was visible to the studio camera. The second was that when projecting onto the tracking camera the light would interfere with the quality of head tracking, as it overpowers the appearance of the actor. These issues could both be solved by rendering the light invisible to the studio camera and tracking camera by implementing a synchronised shutter system comparable to that used in the Invisible Light Projection system (Fukaya, et al., 2003).

It was found that if the tracking camera needed to be placed on the blue wall of the virtual studio it would potentially be visible to the studio camera. No method has been developed to entirely conceal it from the viewing audience, although using key colour for the body of the camera would reduce its overall visual impact significantly.

[3] a measurement of light intensity

ScaMP is currently limited to the walls of the virtual studio, but would not be able to project off set towards the studio cameras as no wall or projection surface can be present (as it would obscure the view of the studio cameras). Fortunately, many monitors can be placed off set as they would not be visible to the studio cameras. If the actor was required to look in this direction the availability of feedback would still be better than if they were facing in any other direction and only had standard forms of feedback available to them.

The feedback of ScaMP met the requirement of ever-present visual feedback for a single planar surface. The ubiquity of ScaMP could be improved by extending the range of the system to cover multiple planar surfaces, providing the actor with feedback that surrounds them. This would require a frame that would support 360° pan servo movement and a head tracking method that could detect the angle of the head even when the actor is not looking towards it. A proposal for this extension to ScaMP was discussed in section 4.3.3 and is described further in Appendix #C.

The contributions to knowledge from this work are the resulting developments as guided by the heuristic evaluation, demonstrating the impact that the heuristic evaluation framework had.

First, it was identified that visual feedback should be ubiquitous to the actor. It was identified that the static nature of the off-screen monitor means that a trade-off exists that could impact actor performance with respects to their ability to maintain a convincing gaze while correctly placing their hands near an object. The ability of ScaMP to scan an environment with a projection in the gaze location of the actor, meeting the requirement to one surface. Appendix #C describes the expansion of this system to multiple surfaces.

Second it was identified that visual feedback should be able to support the actor with different tasks they would face in the interactive virtual studio; including actor orientation, gaze correction and interaction. To resolve this, a range of feedback options were provided to the actors to support them with gaze based tasks (POV feedback mode), accurate interaction tasks and actor orientation tasks (third person standard broadcast render mode), and written instruction tasks (teleprompter mode).

For the analysis of human motion during interaction in chapter 7 ScaMP was not used as the feedback device as it was deemed to be too unstable for use in the experiment. Although the concept of ever-present visual feedback was preserved in this experiment as the monitor used was within the actor's field of view at all times, replicating the functionality of an ideal ScaMP system.

Chapter 5 : OCCLUSION

This work was published in Lecture Notes for Computer Science volume 7425 as "Advanced Occlusion Handling for Virtual Studios" (Hough, et al., 2012b)

5.1. Introduction

This chapter explores the development of an automatic occlusion system that is compatible with standard virtual studio systems (i.e. one with a layer-based format), conforming to the heuristic #1. To summarise the findings of the literature (section 2.4), unlike occlusion systems in fully virtual environments the way in which the objects and actor occlude each other in the virtual studio must currently be defined manually by a virtual studio operator. Usually this is done beforehand and typically the occlusions remain static throughout the production.

Several methods have been developed to allow occlusion in the virtual studio to be changed in real time, but these are either incompatible with the layer-based systems used in standard virtual studios (as shown in Figure 2-15, page 26) or only use a single point on the actor's body to define occlusion events, which does not account for the extremities of the actor's body.

Based on the findings of the heuristic evaluation (section 3.2. Occlusion Findings), the automated method described here is a real time approach that compares the relative distances between the joints of the skeletal motion capture data of the actor and the location of the virtual objects to create realistic occlusion. This approach is compatible with the standard system and accounts for the multiple extremities of the actor's body. This work also presents a taxonomy of three modes of occlusion that could occur in the virtual studio and describes a separate layer-based method for each.

5.2. Occlusion

Occlusion (occasionally known as occultation or interposition), is a visual phenomenon where opaque objects block other objects that are behind them, from the perspective of a single viewer. Occlusion acts as a vital component in the monocular perception of depth, as it is one of the most significant cues that humans use to judge the distance of objects from them (Cutting, 1997). If an object blocked by another, it is an automatic response for a person to conclude that object is further away. In essence, it allows a person to rank the relative nearness of objects in their field of view.

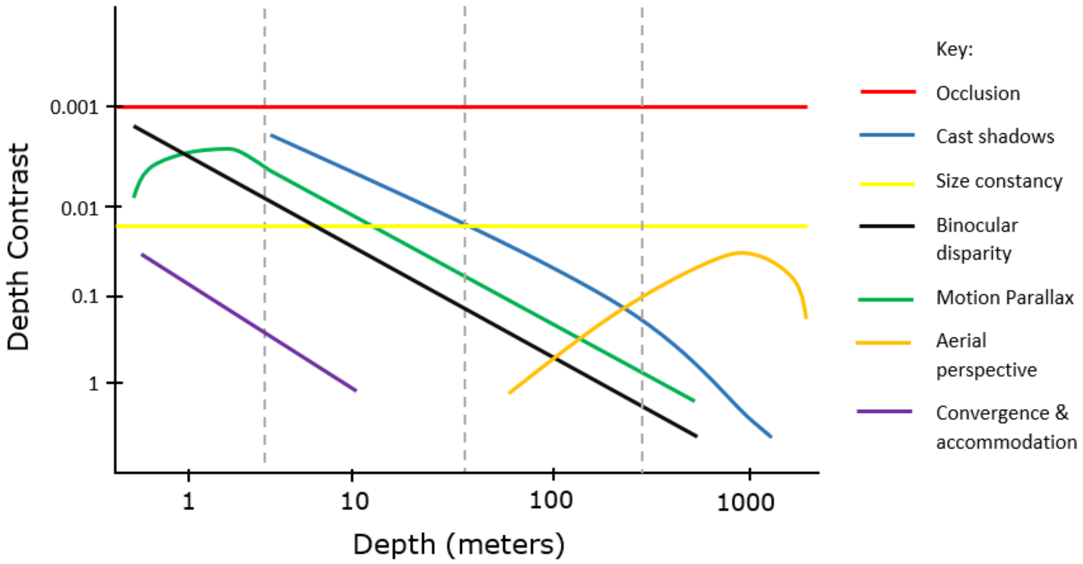


Figure 5-1. Graph illustrating the importance of depth cues at various distances between objects **(Cutting, 1997)**. Here Depth Contrast is calculated using $\frac{2(D_1-D_2)}{D_1+D_2}$ and Depth is calculated using $\frac{D_{1}+D_{2}}{2}$, where D_1 and D_2 represent the distance of two objects.

In real world environments occlusion occurs naturally and this is the basis for what we deem an authentic occlusion when aiming for realism. In virtual and mixed reality environments occlusion between objects should be replicated in a manner that reflects the occlusions we see naturally. Any violation of naturalistic occlusions will lead to incorrect apparent relationships between objects, causing them to appear unrealistic to a viewer.

Figure 5-1 shows a graph illustrating the importance of depth cues at various distance cues, which operate as a vital component of scene realism (Cutting, 1997). It shows that occlusion (the red line) is one of the consistent depth cues that serves to inform the viewer of the relative nearness of objects in a scene (alongside size constancy), demonstrating its importance. Inconsistent occlusion at any distance will negatively affect the plausibility of the scene. Consequently, providing authentic occlusion handling in the virtual studio is important. The benefits of providing occlusion in the virtual studio are two-fold, as it creates a more authentic scene for the viewer (as evidenced in chapter 8 of this thesis) and provides relative depth cues to the actor. However, no prevailing automated occlusion method for the traditional virtual studio has yet been developed.

5.3. Development

This section will describe the development of the occlusion system, first describing the framework, followed by the development of each occlusion mode. This system will be designed to be consistent with the findings of the heuristic study, where the occlusion is compatible with a standard virtual studio format in a real-time and realistic manner (heuristic #3) and must support different modes of occlusion at the extremities of the actor's body (Heuristic #5).

5.3.1. System Framework

A framework that allows occlusion to be automated in real time without live manual intervention and is compatible with a standard system is presented in a flow diagram illustrated in Figure 5-2 (page 64). The first step of the process is to capture the skeletal tracking data of the actor and information on the properties of the virtual object (including location, orientation, bounding box data and mode of occlusion). This step is followed by identifying the primary tracking point, the location of the actor most likely to create an occlusion. The system then identifies whether an occlusion event should be taking place (which is true when a portion of the actor appears to be either behind or in front of a virtual object from the perspective of the studio camera) and then renders the virtual objects on the appropriate layers. For a non-interactive objects (as defined by Kim *et al* (Kim, et al., 2006), if an occlusion event was found to take place then the system will call the updated motion capture data and object properties and will analyse whether the occlusion is still true, taking into account the primary tracking point as well as all other tracked skeletal points (and the space in-between them). If the occlusion event is detected as still taking place the objects involved are "locked" to their layers; if not, then the process restarts.

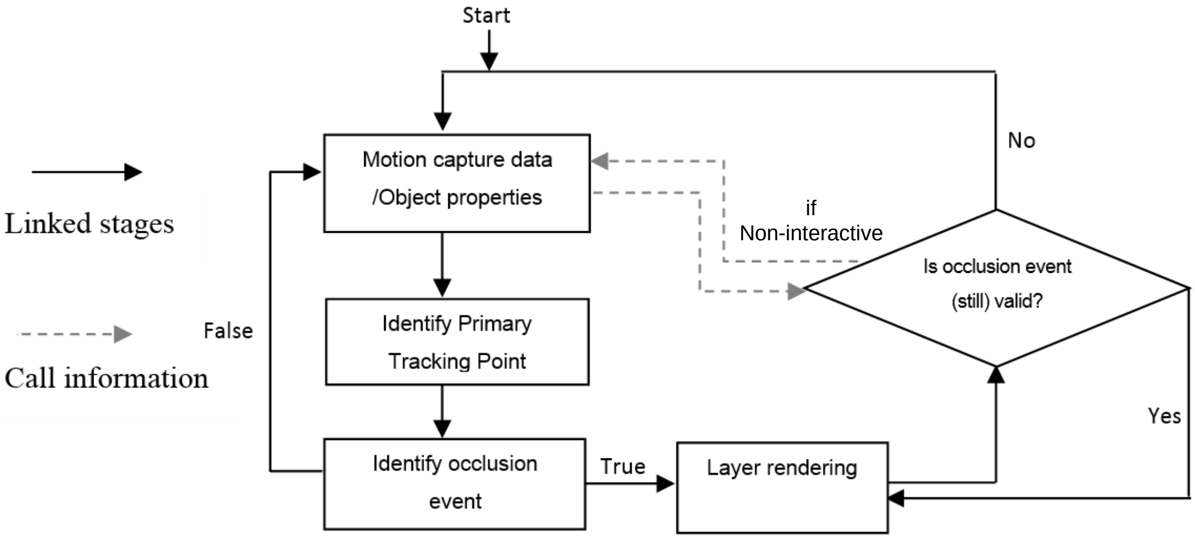


Figure 5-2. Occlusion System Framework

Identify Primary Tracking Point - From the multiple skeletal tracking points a single "primary tracked point" (the tracked point closest to the virtual object) is selected as the basis for the occlusion. This allows occlusion events to be based on the extremities of the actor's body. In Figure 5-3 the primary tracked point is represented by the red circle over the actor's hand, which is the closest tracked point to the virtual cube. The primary tracked point is found using the pseudocode presented in Figure 5-4.

In the heuristic study it was found that most occlusions that occur during an interaction are found to depend on the positions of the actor's hands. The tracked point on the hand in OpenNI is based in the middle of the palm, which in the context of the method presented in this chapter was found to be sufficient. The latency of the Microsoft Kinect system was also low enough for the system to work in real time.

Identify Occlusion event - The occlusion of the virtual object was assessed using a bounding box. This was found to be sufficient for assessing occlusion, but may fail with objects that have particularly significant protruding features where assessing occlusion by the hull of the object may be more appropriate. Figure 5-6 (Positioned on page 64 as it is more relevant there) shows an example of Utah teapot represented by a single bounding box (green lines), consistent with the method used for this occlusion system.

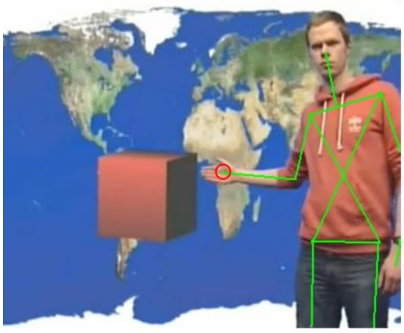


Figure 5-3. Points of actor's body tracked by the Kinect device. The red circle is the primary tracked point in this case.

```
acquire data (tracking points)
for each object do
   acquire data (object location)
   for each tracking point do
      T = distance from tracking point to object location
   end
   Primary tracking point = lowest value (T)
end
```

Figure 5-4. Pseudocode to find primary tracked point

Layer Rendering/Is occlusion event still valid? - If any part of the actor appears to be directly in front of or behind a non-interactive object's bounding box and moves along the Z-axis, the occlusion could suddenly change and they would appear to pass through the object. This effect would not appear plausible to the viewer, as later evidenced in section 8.5.5.2.. The solution incorporated into the developed occlusion system was to not allow the object to change layer whilst any other tracked point on the actor (or any point in between them) appears in front of or behind the object, which conforms to the requirement of heuristic #2. Whilst this carries a risk of incorrect scaling between the actor and object, it is perhaps preferable to allowing the actor to appear as though they are passing through the object. We assume that an interactive virtual object would move accordingly and there would be no need to lock the occlusion, in which case the occlusion is assumed to be invalid and thus recalculated each time to ensure the occlusion is constantly updated.

5.3.2. Occlusion Modes

Different modalities of occlusion are needed to achieve a range of effects, as no universal approach would work in a standard system. As part of this study and the heuristic investigation three types of occlusion were identified (heuristic #4), which are replicated here. Presented in Figure 5-5 is a taxonomy of these occlusions. This section aims to describe methods of achieving each of these occlusions in a layer-based format.

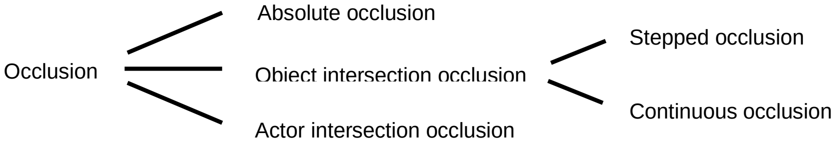


Figure 5-5. Taxonomy of occlusions in a standard (layer-based) virtual studio

5.3.2.1. Absolute Occlusion

Absolute occlusion occurs when the entire object appears either in front of or behind the actor, being positioned on either the foreground or background layer. If the primary tracked point falls behind the object (bounding box) then the entire object can be rendered on the foreground layer in front of them, if it is in front of the object then the object can be rendered in the background layer.A successful demonstration of this type of occlusion is found in Figure 5-17 (page 73), where the actor moves their hand from behind the object to in front of it. The pseudocode for processing this occlusion modality is the same that is presented in Figure 5-8. An example of the bounding box used for this occlusion is the same as that presented in Figure 5-6.

Determining occlusion by the bounding box requires it to be calculated relative to the Z axis of the studio camera and taking into account the rotation of the virtual object itself, so that occlusion appears correct to the audience. The need for this is illustrated in Figure 5-7, which shows a demonstration of the effect that the apparent object and camera angles can have on occlusion. Figure 5-7a shows an object that is occluded by the actor's hand. Figure 5-7b shows the object and the actor's hand in the same location as before, but with the object rotated slightly; the result being that the actor's hand is now occluded by the object.

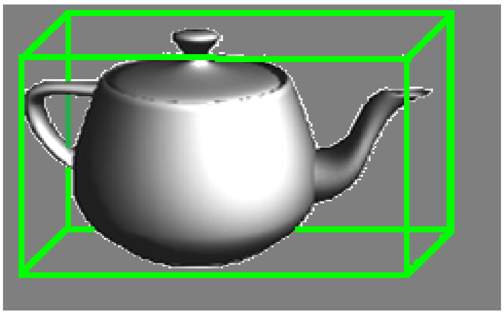
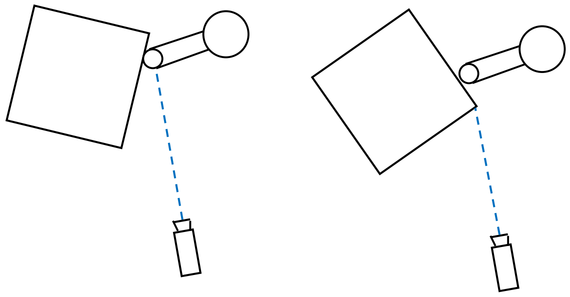


Figure 5-6. Bounding box for a Utah teapot



a. Actor occluding box (15° angle)

b: Box (55° angle) occluding actor

Figure 5-7. The apparent object rotation affecting the overall occlusion of the object.

```
A1 = acquire data (Primary track point)
A2 = acquire data (Bounding box surface)
t = threshold
if A1 < (A2 + t):
    if $V^{\circ\prime}/(V^\circ + \tan^{-1}\frac{(BBz/2)}{(BBx/2)})$ => 1:
       Virtual object in foreground
    if $V^{\circ\prime}/(V^\circ + \tan^{-1}\frac{(BBz/2)}{(BBx/2)})$ < 1:
       Virtual object in background
    end
end
```

Figure 5-8. Pseudocode for determining whether the virtual object should appear on a background or foreground layer.

$$V^{\circ\prime} = \tan^{-1}\frac{Cz - Vz}{Cx - Vx} + V^\circ + \tan^{-1}\frac{BBz/2}{BBx/2}$$

Equation 5-1. Determination of apparent object angle from studio camera

To determine whether the bounding box (and the object) should appear in front of or behind the actor from the perspective of the studio camera, the following process is used. First the apparent rotation of the virtual object as it appears from the active studio camera is calculated using Equation 5-1 where C denotes the location of the studio camera, V the location (or angle in Figure 5-8) of the virtual object and BB the dimensions of the object's bounding box. Then the pseudocode in Figure 5-8 is used to determine the occlusion, using the same notations.

5.3.2.2. Object Intersection Occlusion

Objects with spaces in the hull or a liquid form factor should allow the actor to intersect them; the resulting occlusion is called 'Object Intersection Occlusion'. Two distinct forms of intersection have been identified, with each appropriate for different types of virtual object; these intersections are named 'Stepped Intersection' and 'Continuous Intersection'. Examples of these occlusions (and results achieved by this system) are presented in Figure 5-17 (page 73), where the actor has placed one hand into the virtual object, causing part of the object to appear behind the hand and part of it in front.

Objects that can be intersected and the positions of the allowed intersections need to be predefined within the system. Occlusion is determined by identifying the Z-location of the primary tracked point and automatically splitting the object at that location (or the closest allowed position) into two separate objects (Bounding Boxes). The object portion furthest from the studio camera must appear behind the actor, and the object portion closest to the camera in front.

Stepped Intersection. This allows the actor to intersect the object at defined locations only: for example an actor reaching into a mandible with one tooth missing would only be able to intersect at the gap between the teeth, but not intersect where no gap exists. An example of the virtual object 'split' into two constituent sections is presented in Figure 5-9, where the object is split at the point of an intersectable gap (the missing tooth). Multiple split locations can be used if required for an object that contains multiple intersectable points. The pseudocode for calculating the split is presented in Figure 5-10. Here the object can be split into multiple sections at the pre-defined split points, with the object portions rendered on the either the foreground or background layer depending on their position relative to the location of the primary tracked point.

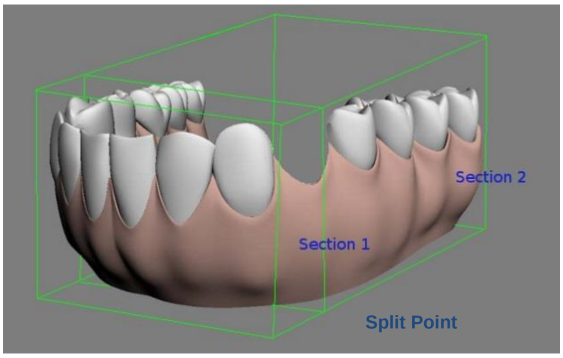


Figure 5-9. Stepped intersection bounding box example

```
A1 = acquire data (Primary track point)
A2 = acquire data (Object Bounding Box)
Step = acquire data (pre-defined split(s))
valid = evaluate intersection is True
if valid = True:
    split object at Step
    object sections < A1 = Foreground
    object sections > A1 = Background
end
```

Figure 5-10. Pseudocode for Stepped Occlusion

Continuous Intersection. This occlusion allows the actor to intersect a virtual object at any location: for example the intersection of a liquid object that the actor's hand can enter at any point. In the continuous intersection system the two sections the object is split into are continuously updated according to the Z location of the primary tracked point An example of the virtual object split into its two constituent sections is presented in presented in Figure 5-11. Pseudocode for achieving this occlusion is presented in Figure 5-12.

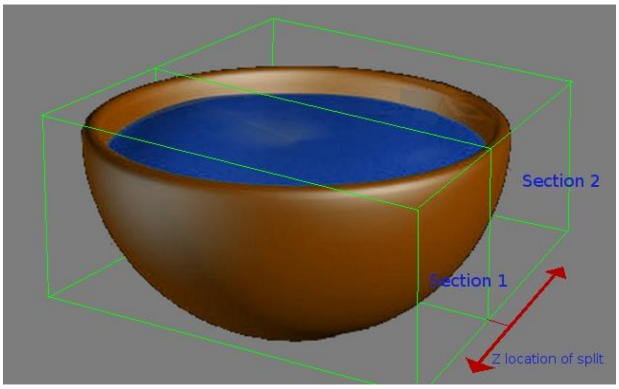


Figure 5-11. Continuous intersection bounding box example

```
A1 = acquire data (Primary track point)
A2 = acquire data (Object Bounding Box)
valid = evaluate intersection
if valid:
     split object at A1(Z')
     object portion < A1 = Foreground
     object portion > A1 = Background
end
```

Figure 5-12. Pseudocode for Continuous Occlusion

5.3.2.3. Actor Intersection Occlusion

Actor Intersection occlusion occurs when part of the actor's body is required to appear in front of the virtual object and the other part appear behind. For example, when holding an object between two hands when standing perpendicular to the camera, as shown in Figure 5-17 (page 73 – Row: Actor Intersection Occlusion).

The developed system achieves this by splitting the object so as to render one portion on the foreground layer and the other portion on the background layer, based on the location of two tracked points on the appropriate parts of the actor's body. These two tracked points are the primary tracked point and a secondary tracked point that is the second closest. The object is split perpendicular at the midpoint between these two tracked points. The z location of the tracked points from studio camera are then analysed, with the object portion corresponding to the tracked point nearest the camera being rendered in the background layer; with the object portion corresponding to the furthest tracked point being rendered in the foreground. A demonstration of this occlusion using the current method of this system is presented in Figure 5-17 (page 73). The separation of the object into two bounding boxes illustrated in Figure 5-13 (page 69), with pseudocode for this method is present is presented in Figure 5-14 (page 69).

In the current state of development the system is limited to cases where the two parts of the body do not self-occlude. If the actor were to stand behind the virtual object and place one arm around and in front of it, they would remain entirely behind the virtual object, as the video layer containing the actor cannot appear both partially in front of and behind the virtual object situated on the foreground layer simultaneously. This is analogous to the Painter’s problem, where occlusion is achieved by layering objects on top of each other, not allowing for cyclical occlusions (i.e. the actor cannot both occlude and be occluded by the same virtual object). A solution to this issue is proposed in Appendix #D, which presents a prototype system that cannot currently be fully realised due to technological limitations.

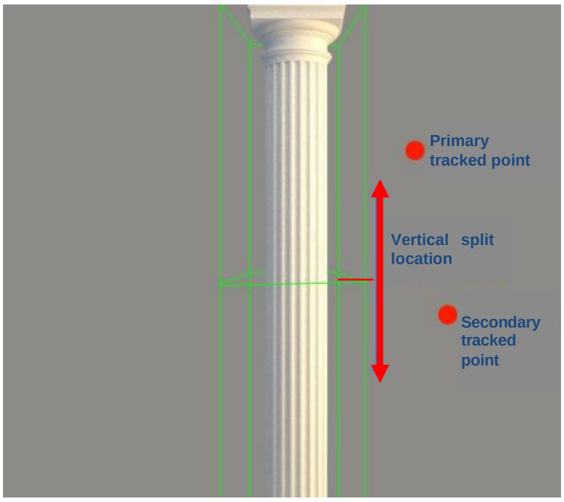


Figure 5-13. Example illustrations of the portioned object Bounding Boxes for the Actor Intersection Occlusion

```
A1 = acquire data (Primary track point)
A2 = acquire data (Secondary track point)
A3 = acquire data (Object Bounding Box)
valid = evaluate intersection
if valid:
    split object at (A1+A2)/2
    if A1(Z) < A2(Z):
       object portion A1 = Background
       object portion A2 = Foreground
    if A2(Z) < A1(Z):
       object portion A1 = Foreground
       object portion A2 = Background
end
```

Figure 5-14. Pseudocode for Actor Intersection occlusion

Future directions. A method based on ray casting (Roth, 1982) is proposed as a future direction for this study where the object is only partially rendered, exposing part of the actor on the video layer underneath. As demonstrated in a feasibility study, this method will allow the actors to appear as though they are being intersected by an object that is actually set entirely on the foreground layer.

Future work in this area could utilise a 3D ‘avatar’ to enhance occlusions in the layer-based system for systems that require more complex occlusions, such as that posed by grasping, where one portion of the hand should appear in front of the object and one portion is occluded behind the object. As a layer-based system only presents the actor on a single layer an exception needs to be made for these cases, with a system that allows a single layer to appear as such.

A proof of concept system was developed using a subtractive ray casting technique that has been implemented into the layer-based system. A volumetric model (avatar) of the actor will be matched to their motion capture data to calculate where the occluded portions of a virtual object are. Only the potions of the virtual object visible from the perspective of the studio camera (the origin of the cast ray) will be rendered, effectively creating a “hole” in the virtual object the same shape (as seen by the camera) as the portion of the actor occluding it. The actor, placed behind the foreground layer that the object is on, would appear through this gap achieving an effect that makes that portion of the body appear as though it is in front of the object.

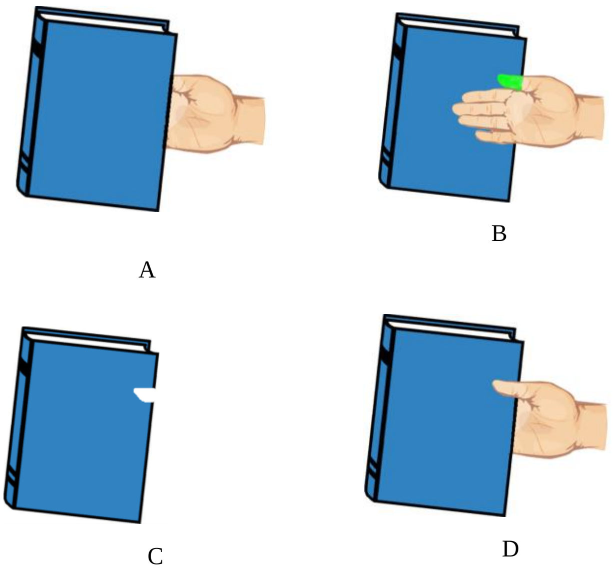


Figure 5-15. Example of partial rendering system

This would allow the actor to appear more visually consistent within the virtual environment, allowing an actor on a single layer to be presented as if they appear across two layers. Figure 5-15 shows an example of the technique. Figure 5-15A shows the hand existing on a layer behind the virtual object (the book) it is grasping. Figure 5-15B shows the identified area in green that should be in front of the virtual object to make the hand appear as though it is grasping the book. Figure 5-15C shows the partially rendered virtual object with a section missing where the identified area was. Figure 5-15D shows the result of the partial render, with the thumb appearing to be over the virtual object, even though the virtual object exists on a foreground layer.

A feasibility study of this approach was conducted by building a proof of concept system, where the virtual hand had no articulation. This system used an "avatar" model of an actor's forearm and matched it to the movements of the actual arm, with no other articulation provided. The result, presented in Figure 5-16, demonstrated that this method is capable of achieving the Actor Intersection Occlusion in cases where self-occlusion is present, thus overcoming the limitations present in the system described.

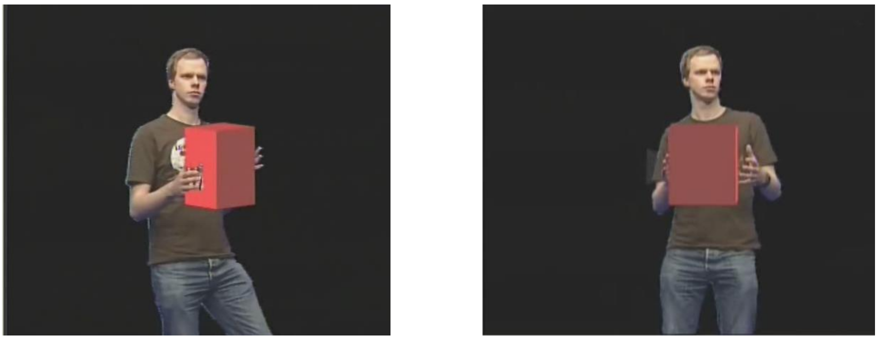


Figure 5-16. Images showing the prototype model based ray casting system from the feasibility study.

However, limitations in the current level of technology mean that an accurate model based ray-casting approach to Actor Intersection Occlusion is not yet feasible. This is due to the level of articulation that needs to be made to the avatar for a plausible outcome (i.e. closely tracked finger movement, accurate modelling of clothes), which cannot currently be achieved by imperceptible motion capture methods to a satisfactory level. Consequently, this approach was not investigated further.

5.4. Discussion

The skeletal tracking approach presented in this chapter has been developed to enhance the level of occlusion handling in virtual studio systems by automating the process for a range of occlusions in a real-time setting. The system was able to replicate many of the occlusions that would be typical in real television studios, extending the functionality over that of current virtual studios.

The occlusion system was built to the requirements of the heuristic evaluation. It worked in real-time, was compatible with standard virtual studio designs, accounted for the extremities of the actor's body, provided several appropriate occlusion modalities and locked the occlusion where necessary. The occlusion programme ran in real-time on a PC comprised of a 3.6GHz Pentium 4 processor, 2GB of RAM and a Windows XP operating system. Individual layer-based methods were provided for absolute, object intersection and actor intersection occlusions, the results of which are demonstrated in Figure 5-17 (page 73). In this table a reference description of what occlusion an observer would expect to see in a real television studio is given based on the analysis of real world equivalents for each occlusion. It then shows images of occlusions achieved using a current standard virtual studio and then finally with the addition of the automated occlusion-handling functionality described in this chapter. The occlusion-handling virtual studio is shown to allow a more accurate representation of absolute occlusions and effective object intersection.

Currently actor intersection occlusion has only achieved to a limited level, based on the splitting of a virtual object into two parts (horizontally in the case given in Figure 5-17 (page 73)). However, if different parts of the actor's body were to appear in front of and behind the object simultaneously the occlusion will become unsustainable (as described in 5.3.2.3), creating an issue analogous to that of the Painter's problem. Nevertheless the current method was still useful because the actor can be cognisant of the limitations and attempt to position their arms and other parts of their body appropriately.

The contributions to knowledge from this chapter are the resulting developments as guided by the heuristic evaluation.

First, it was identified that occlusion should be compatible with existing layer-based systems. The design of the occlusion system was embedded within an existing virtual studio that used the common layer based system, satisfying this heuristic.

Secondly, it was identified that occlusion should be locked while the actor is directly behind or in front of a non-interactive object, as changes would cause the virtual objects to appear as ghosts when the actor passes through them. The layer locking method that was developed stopped the actor form passing through the virtual objects as they passed them, eliminating the ghost effect and meeting the requirement of this heuristic.

Thirdly, it was identified that occlusion should occur in real time and reliably. This is demonstrated in figure 5-17 (page 70), which illustrates the fully automated occlusions that could be achieved in the real time system using data obtained from the skeletal motion capture system, thus meeting the requirement of this heuristic.

Fourthly, it was identified that the occlusion system should support different occlusion types, as it was found that in a layer-based system there would not be a single method of occlusion that would be suitable for all object types. From user studies three different occlusion modes that were identified to meet this heuristic, Absolute, Object interaction and Actor interaction (page 62). This is demonstrated in figure 5-17 (page 70), thus meeting the requirements of this heuristic.

And finally, the occlusion must factor in the extremities of the actor's body. The extremities are required because if the actor's torso is used for defining occlusion, the occlusion properties of the scene will appear incorrect if the actor's hand is the correct point of occlusion. To meet this requirement, the actor tracking system used multiple points of the actor's body to base occlusion on, using the closest tracked on the actor's body to the virtual object.

Current systems can be reviewed by further heuristic analysis using the developments that have been made in this research to further define the quality of occlusion in these systems. For example, further heuristic analysis would help with exploring an occlusion system for actor intersection that's more consistent with real life, such as the prototype described in section 5.3.2.3.

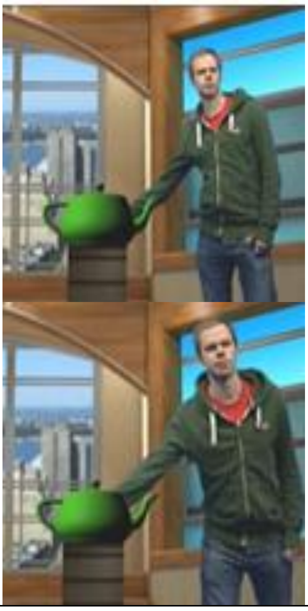
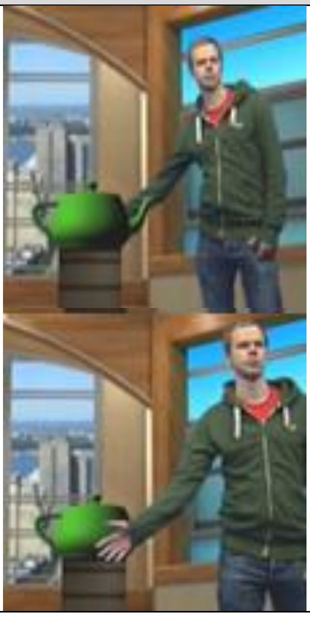
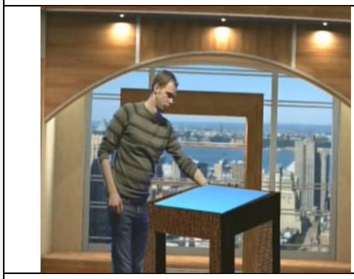
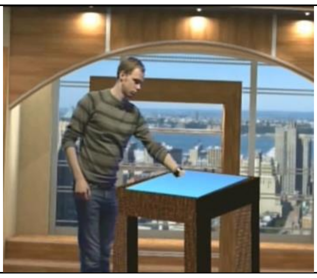
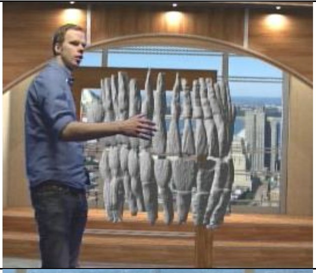
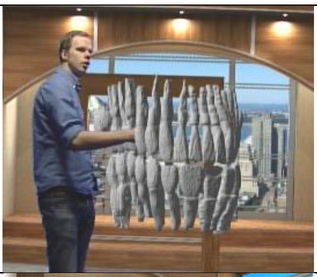
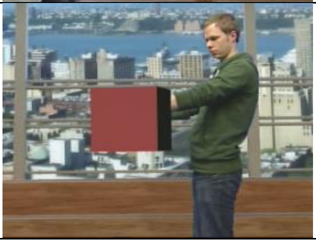
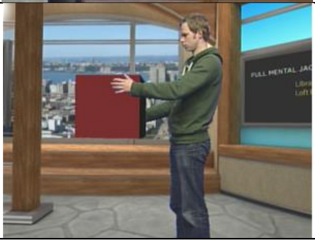
Occlusion type	Real television studio	Standard virtual studio	Occlusion-handling virtual studio
Absolute occlusion	An observer would see that the actor would be occluded when they stand behind a real object. Conversely when the actor appears in front of the object it should be occluded.		
Object intersection occlusion	An observer would see an actor both occluding and being occluded by the real object. The portion of the object in front of the actor occludes him, and the portion behind the actor is occluded.	Continuous occlusion 	
		Stepped occlusion 	
Actor intersection occlusion **-** **Object split**	An observer would see the actor touch two points on the same object. This would result in him being occluded by and occluding the object simultaneously.		

Figure 5-17. Sample set of occlusions from a standard virtual studio and the occlusion handling virtual studio. The ‘real television studio’ column contains a description of a real occlusion’, the ‘standard virtual studio’ column contains a descritpion of what can currently be achieved and the column ‘occlusion handling studio’ shows what can be achieved using the methods presented in this chapter.

Chapter 6 : INTERACTION

This work was a selected project for Interactivos 2014 as 'A mixed reality Environment for Interactive Presentation and Performance' (Interactivos, 2014), an event where novel systems are demonstrated to peers for criticism and further collaborative development. The work presented in this chapter describes the development of the interaction system up until this event.

6.1. Introduction

This chapter explores the enabling of interaction in the virtual studio. To summarise of the findings of the literature review (section 2.5) many interfaces exist that allow abstract interaction in the virtual studio, such as using, but which either do not allow interactions where the actor appears as though they were holding the virtual object, or use methods that are visible to the audience.

Here we discuss an interaction system that has been developed using the skeletal motion capture data from the actor to allow them a high-level of control over the virtual object in a simplified form that conforms to the heuristics.

6.1.1. Virtual Reality Interaction

Interaction is an area of research not limited to the virtual studio. One of the primary domains for creating interaction between human motion and virtual objects is virtual reality, which is a system that allows a user is able to experience a full virtual environment through a virtual reality headset.

Interaction is a key element of virtual reality research, in which the users interact with virtual objects through some device that tracks their movements and infers them as interactions, such as wired gloves that track the pose of the user's hand (surveyed in (Sturman & Zeltzer, 1994) and hand-held tracked wands that can sense how the user is moving them, e.g. (Keefe, et al., 2001)).

However, interacting with virtual objects poses many challenges as humans are not capable of interacting with virtual objects in the same way that they are capable of interacting with real objects, primarily due to a lack of haptic (touch) and visual feedback (Mine, et al., 1997). For example picking up a virtual box is difficult because a person is not able to detect when they are in contact with the sides, allowing potential for misplacement. Consequently, many of the technologies that have been designed to allow humans to interact with virtual objects are pragmatic and indirect, where selection and manipulation tasks do not involve any contact with the virtual object's surfaces themselves. These approaches can be broadly categorised as Ray Casting and Extending Arm.

The ray casting technique (Mine, 1995) uses a ray (a vector) that is cast in the direction that the user is pointing to select a virtual object. Once the virtual object is selected, it will follow the location of the ray as the user points to new locations. This technique allows users to select an object with relative ease, but does not allow them to move the object towards them or further away and is not practical for changing the object's orientation.

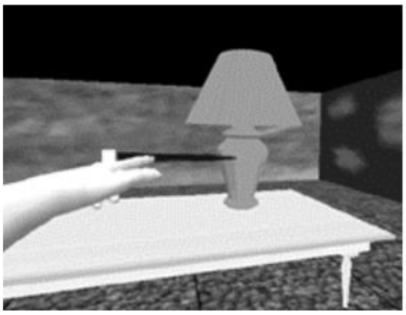


Figure 6-1. Example of ray casting technique, where a ray is cast to the interactive object, in this case the lamp (Image courtesy of Bowman and Hodges **(Bowman & Hodges, 1997)**).

The GoGo technique (Poupyrev, et al., 1996) is one of the standard Extending Arm methods of interaction. In the virtual reality environment the location of a virtual hand is matched to the location of the user's real hand using non-linear scaling. This is to say that when the user's hand is placed close to them the virtual hand is matched 1:1, but as the user moves their hand further away from their body the virtual hand movements become exaggerated, allowing them to reach objects that are far away in the virtual environment. Once the hands reach a virtual object they will be able to select it and will then follow the movement of their hand, allowing them to manipulate it.

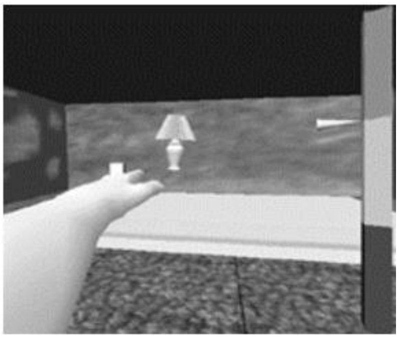


Figure 6-2. Example of the Go-Go technique, where the arm is extended towards the virtual object, in this case the lamp (Image courtesy of Bowman and Hodges **(Bowman & Hodges, 1997)**).

Proprioception has also been used for relaying interaction commands in virtual reality. Proprioception is defined as the awareness of the position of one's own body, meaning that we are able to situate one part of our body in relation to another (e.g. we know how to move our arm so that our hand touches our head). In a virtual reality context this has been used to control virtual objects by allowing the user to directly manipulate the virtual objects by exploiting their knowledge of their relative body positions, or by using body relative controls that allow the user to provide commands by touching a part of his body or making a gesture (Mine, et al., 1997).

Neither the ray casting, extending arm or proprioception methods used for virtual reality are practical for use in the virtual studio, as interaction in this environment would ideally require the actor to appear as though their hands were in contact with the surfaces of the virtual object, as they would with a real object or in a film that involves CGI elements. This violates the requirements of the interaction appearing realistic, as specified in the second heuristic for interaction.

6.2. Development of Interaction

The interaction system was developed in accordance with the findings of the heuristic evaluation and the literature review, whilst remaining compatible with a standard virtual studio system. The heuristic evaluation stipulates that the interaction must allow the actor to manipulate the object to the desired location and this process must be simplified for them to reduce the cognitive load on the actor. The actor must also appear to be holding the virtual object as they would a real one, making the interfaces seen in typical virtual reality systems unworkable in the virtual studio domain (such as those in 6.1.1.).

In this work the hand locations from the real time skeletal motion capture of the actor are obtained using a modern depth-sensitive camera interfaced with OpenNI and are used to create realistic interactions, supporting heuristic #2. We define two broad categories of direct interaction between a person and an object (“single-handed and bimanual”) and one where the interaction is not direct (“abstract”). These are defined as follows:

Single-Handed- single-handed interaction refers to the manipulation of an object with one hand. These are generally pushing or grabbing type manipulations of an object.

Bimanual - Bimanual interaction refers to the manipulation of a virtual object with two hands, usually with symmetric movement between them. An example of this form of interaction is picking up a box by placing the hands on either side of it.

Abstract – Abstract interaction refers to the movement of an object with no direct connection between the hands and surface. Abstract interactions can also have single-handed and bimanual interfaces. An example of an abstract interface is pulling a lever to move the arm of a crane.

The process of interaction is presented as a flow diagram in Figure 6-3, with each stage described in further detail in order of occurrence.

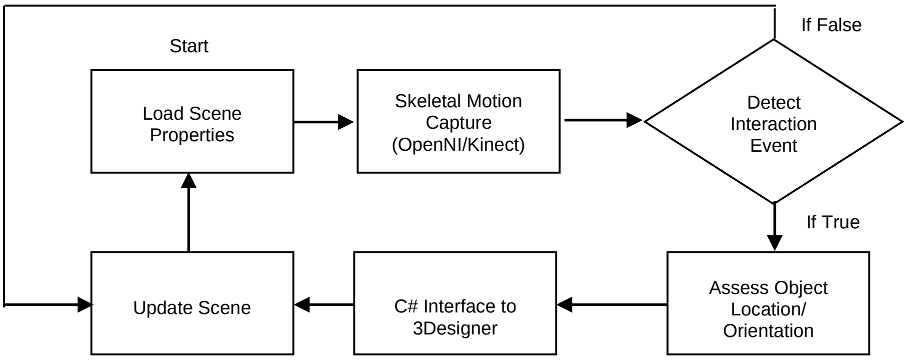

Start
Load Scene Properties
Skeletal Motion Capture (OpenNI/Kinect)
Detect Interaction Event
If False
If True
Assess Object Location/ Orientation
C# Interface to 3Designer
Update Scene

Figure 6-3. Process of interaction

Load scene properties - This step calls the properties of the objects in the virtual set. As previously stated in Section 4.2, Orad 3Designer provides 9 metrics that describe the virtual objects in the set, the location of the virtual object in X, Y, Z, the pan, tilt and yaw of the virtual object and the dimensions of its bounding box

Skeletal motion capture - This step obtains the skeletal motion capture information in 3 Degrees of Freedom; specifically only the locations of the left and right hands are captured and used for interaction.

Detect Interaction event - The *interaction event* is the point where the actor is given control over the motion of the virtual object, allowing them to interact with it. This step determines when the actor intends to interact by identifying whether their hands are near the interactive virtual object's bounding box. The visual hull (the surface) of the object could be used equally and remain as (if not more) valid.

The detection of an active single-handed (section 6.4.2) or bimanual (section 6.4.3) interaction event is defined using the pseudocode in Figure 6-4, where R and L represent the tracked location right and left hands of the actor respectively and BB represents the boundary of the bounding box. Here, while one or both of the actor's hands are detected inside the bounding box (plus an additional tolerance) of the virtual object, an interaction event for that object becomes active until the hands leave the bounding.

```
While R < BB and L < BB:
      Interaction event = True (Bimanual)
      end
While R < BB or L < BB:
      Interaction event = True (Single-handed)
While R > BB and L > BB:
      Interaction event = False
```

Figure 6-4. Pseudocode for detecting an interaction event

Occasionally during a bimanual interaction the actor placed their hands too far away from the bounding box when they intended to interact with it (as evidenced in Chapter 7). In the system described in this chapter, in these bimanual cases a tolerance was included, where the dimensions of the bounding box were expanded by 10% to account for this error. A variation in the distance between the actor's hands where they could drift apart during an interaction was also discovered (again evidenced in Chapter 7), which could lead to an unwarranted end to an interaction event; a case that was found to occur frequently. To account for this, while an interaction event was `True` the tolerance was expanded to 30% instead. To stop the interaction event, the actor moves their hands apart, so that they are outside of the bounding box.

Assess Object Location/Orientation - This stage calculates the new object location and orientation using the interaction modalities discussed throughout section 6.4.

C# Interface to 3Designer - This stage acts a conduit between the previous stages that were programmed using Python, and the next stage, which is to update the scene in 3Designer.

Update Scene - This stage updates the parameters of the virtual set in Orad 3Designer.

6.3. Interaction Modalities

The interaction modalities discussed in this section are Triggered interaction, single-handed interaction and Bimanual interaction.

6.3.1. Triggered Interaction

Triggered interaction would be considered a form of abstract interaction, but with the appearance of a single-handed or bimanual interaction. This allows the actor to interact with a virtual object by moving their hand to a specified location and/or at a specified velocity. When the actor has met these criteria the virtual object moves along a pre-defined motion path. In the example scenario presented in Figure 6-5 the only desired scenario for the scene is that the actor pushes the teapot off its stand. To achieve this, the actor moves their hand towards the location of the teapot and the hand is moving fast enough and in the correct direction the teapot will “fall” off the virtual stand along a predefined path.

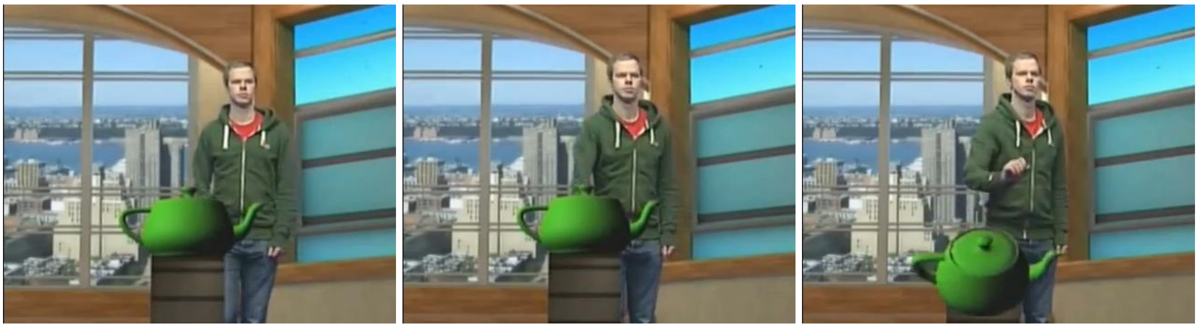


Figure 6-5. Triggered interaction, illustrating the object following a pre-defined path after a push gesture.

6.3.2. Single-handed Interaction

Single-Handed interaction is the movement of a virtual object with one hand, which occurs when one of the actor’s hands meets the bounding box of an interactive virtual object. The virtual object was moved according to Equation 6-1 (page 79). Here V denotes the virtual object 3DOF location on the x, y and z axes, BB the relevant dimensions of the virtual object’s bounding box on the x, y and z axes and A the 3DOF location of either the actor’s left or right hand, so when the hands are detected as touching the bounding box surface, the object moves in the intended direction. The result of this interaction is demonstrated in Figure 6-6 (page 79), where the virtual sphere is moved by the actor touching its surface.

$$V' = BB/2 + A$$

Equation 6-1. Calculation of the virtual object location during a single-handed interaction event

For a more simple form of single-handed interaction a selected location on the virtual object can be fixed to the location of the actor’s hand, supporting heuristic #1. This is demonstrated in Figure 6-7 (page 79) which shows the top of a chain matched to the hand location of the actor, allowing a more intuitive interaction as the actor does not need to consider the location of the object boundaries.

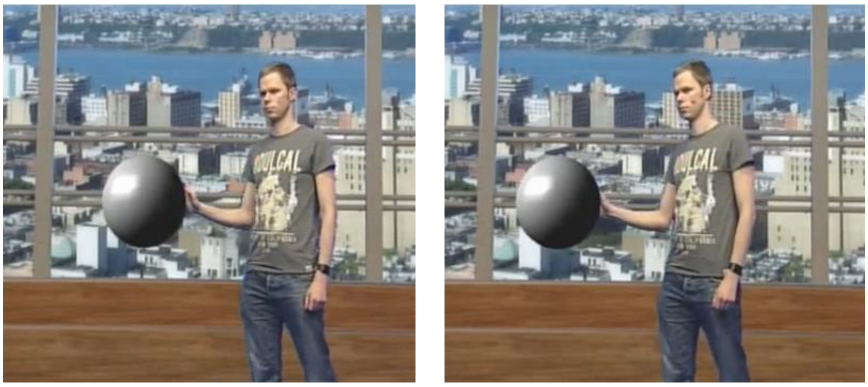


Figure 6-6. Single-handed interaction with virtual object

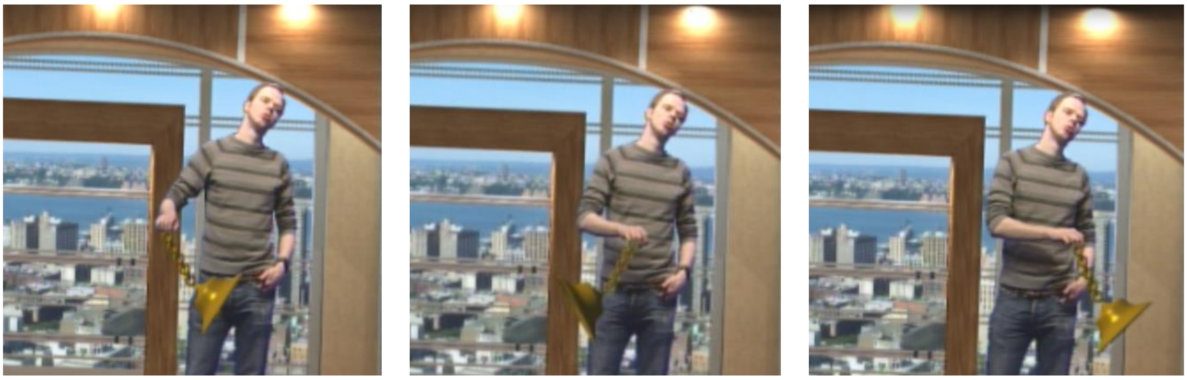


Figure 6-7. Example of a virtual object matched to the location of the actor’s hand

6.3.3. Bimanual interaction

Bimanual interaction is the ability to move the virtual object with two hands. In this case, when the actor places their hands near the surface of the virtual object it is locked to the midpoint of the hands. During an interaction the object will change location and also rotate based on the change in orientation between the hands, relative to its original orientation.

6.3.3.1. Full axis modality

Full axis interaction gives the actor bimanual control over the virtual object in 6DOF, which is shown in Figure 6-8 where the actor moves the virtual tooth down.

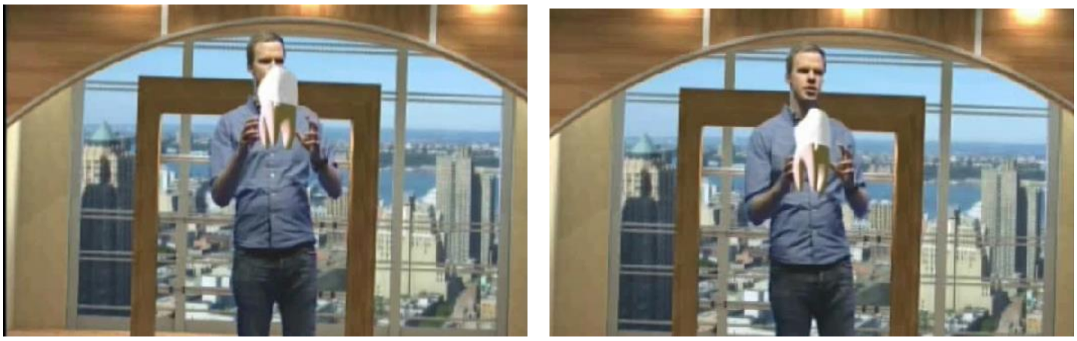


Figure 6-8. Full axis bimanual interaction with virtual object. Actor is moving tooth model down.

The location of the virtual object is defined in Equation 6-2, where V denotes the 3DOF virtual object location, R and L the 3DOF location of the actor's right and left hands respectively and $_{i-1}$ denotes the object's location in the frame before the interaction event occurred. This equation locks the centroid of the virtual object to a constant location relative the midpoint of the actor's hands, defined using the location of the object at the start of the interaction event.

$$V' = \left(V_{i-1} - \left(\frac{R_{-1} + L_{-1}}{2}\right)\right) - \left(\frac{R + L}{2}\right)$$

Equation 6-2. Calculation of the virtual object location during a full axis interaction event

One component of the orientation of the virtual object (in this case the pan) is calculated using Equation 6-3, which is relative to the object's initial orientation (a sum of the object's initial orientation minus the angle between the hands). In this equation θ denotes the angle of the virtual object, θ_{xi-1} denotes the angle of the object in the frame before the interaction event, L and R the locations of the actor's Left and Right hands respectively, O the Origin (as defined in Figure 4-3, page 50), 'a' the distance between the Origin and the Left hand, 'b' the distance between the left and right hands and 'c' the distance between the right hand and the origin and i-1 denoting the object's state in the frame before the interaction event occurred. The equations to find a, b and c are provided in Equation 6-4. The tilt and the roll of the virtual object can be calculated using similar equations. The symbols used in these equations are shown in diagram form in Figure 6-9 (page 81).

$$\theta_x' = \theta_{x_{i-1}} - \left(\cos^{-1}\left(\frac{(a^2 - b^2) - c^2}{2ab}\right) + \tan^{-1}\left(\frac{O_x - L_x}{O_z - L_z}\right)\right)$$

Equation 6-3. Calculation of the virtual object orientation during an interaction event

$$a = \sqrt{(O_x - L_x)^2 + (O_z - L_z)^2} \qquad b = \sqrt{(L_x - R_x)^2 + (L_z - R_z)^2} \qquad c = \sqrt{(O_x - R_x)^2 + (O_z - R_{N_z})^2}$$

Equation 6-4. Calculation of side 'a' length

Calculation of side 'b' length

Calculation of side 'c' length

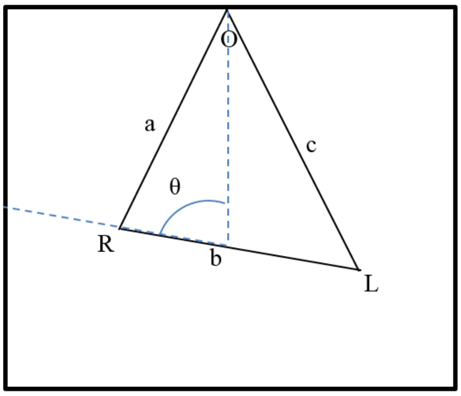


Figure 6-9. Diagram of the trigonometry described in equations 6-3 and 6-4.

6.3.3.2. Locked axis modality

In certain cases where realism may be affected if the actor is able to move a virtual object in a way that would otherwise be constrained in real life, then the axes of motion can be locked. This simplified level of interaction leaves less room for creating unintended object motions along unfeasible axes, which in turn reduces the cognitive load on the actor and supports heuristic #1. In the scenario presented in Figure 6-10 the actor is rotating a globe, an object that would typically only rotate on one axis; hence the actor is only able to rotate the globe around the poles.

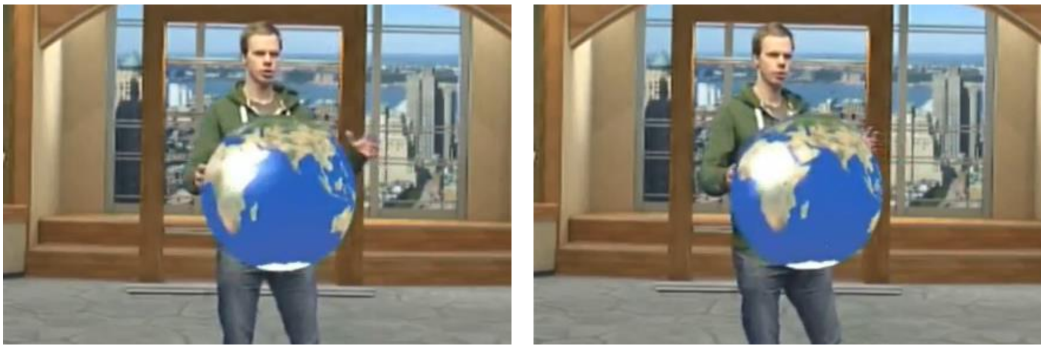


Figure 6-10. Single axis bimanual interaction with virtual object. The actor rotates the globe anti-clockwise

6.4. Discussion

A framework that allows three types of interaction modalities (triggered, single-handed, bimanual) was presented in this chapter, and this framework provided a range of interaction options for the virtual studio. Additional options were also provided for reducing the complexity of the interaction tasks for the actor (i.e. limiting axes and triggering pre-defined events), as stated in the heuristic evaluation.

It was noticed that bimanual interactions did not appear realistic from the perspective of a viewer, as the actor was often likely to overestimate or underestimate the positions of the surfaces of the virtual object. This caused an overlap or gap between their hands and the surface of the virtual object, which appear implausible to the viewer (as evidenced in chapter 8). In addition to this issue, a variation in the distance between the actor’s hands was also present. This issue appears to be one of human kinematics, which would be present in any interaction system that aims to be plausible to a third party viewer, as opposed to an issue related to the interaction system that has been implemented here. Consequently, the ability of actors to use an interaction system similar to this one in a plausible manner is assessed using a novel two-part framework presented across chapters 7 and 8, serving as the analysis of this (and similar) systems.

CHAPTER 7 AND 8 PREFACE

The work in chapter 7 and 8 led to two publications, both presented at IEEE International Symposium on Mixed and Augmented Reality (ISMAR) 2014. These were 'Measurements of Live Actor Motion in Mixed Reality Interaction' (Hough, et al., 2014a) - which was nominated for the best short paper award - and 'Measurement of Perceptual Tolerance for Inconsistencies within Mixed Reality Scenes' (Hough, et al., 2014b). Due to the enthusiastic reception of these publications an extended version of this work has been invited for publication in IEEE Transactions on Visualization and Computer Graphics and is currently under review.

Together Chapter 7 and Chapter 8 will present a novel framework for assessing the plausibility of a bimanual interaction in the virtual studio as it would be perceived by the viewer.

As described in the previous chapters, the focus of research has been towards implementing technology to enable interaction in the virtual studio, in which we sought to produce interactions akin to those of real life. However, the matter of what appears to be a realistic interaction in this environment has not been addressed in the literature. In effect, we would like to answer the question '*what factors make an interaction appear realistic?*' in a manner that accommodates objective comparisons between different interaction modalities and systems.

To achieve this aim, the topic is explored in two stages: one exploring factors affecting the performance of the actor and the other exploring factors that affect the plausibility of the interaction from the perspective of the viewer. These perspectives can then be contrasted to make further conclusions on how they interrelate. The two stages of this method are:

Stage 1: The focus of chapter 7 - The motion of 16 actors when interacting with a virtual object is quantified to identify which conditions impact their performance (i.e. the accuracy and variability of hand placements relative to the object surface).

Stage 2: The focus of chapter 8 - The audience perception of interaction realism is measured by asking a group of observers (test participants who assume the role of a critical audience) to rate a series of videos that each depict a certain magnitude of error measured during chapter 7.

This two stage method will allow the factors affecting the realism of interaction to be identified and will allow a list of recommendations that improve realism to be provided. This method could also be applied to assess and improve the realism of interactions between real and virtual elements for a wide range of environments across the mixed reality spectrum (Milgram, et al., 1995).

Very few studies have been conducted that explore virtual environments from both the perspective of a user and a third party. However, one study presented by Larsson *et al* (Larsson, et al., 2001) sought to analyse how the actor-observer effect (Jones & Nisbett, 1971) influences the “sense of presence” in virtual reality. Presence is defined as the feeling of existing, or seeing someone/something exist, in a certain time and space. The actor-observer effect is a sociological theory that states an actor (a person engaging in an activity) is more likely to attribute their performance to the conditions of the situation, whereas an observer (a person watching the actor) is more likely to attribute the performance of the actor to the actor’s general ability – this theory can be extended to study the feeling of presence of the actor and the observer. These terms are not to be confused with those of “actor” and “observer” throughout the following chapters, where they are used in a more literal context.

In Larsson’s experiment, actors (n=16) completed a task where they had to find the location of four objects in a virtual environment using a headset virtual reality system. The actors completed the task, experiencing the environment from a first person perspective both visually and audibly. The observers (n=16) watched a single video of an actor completing the same task in a fully virtual environment from a third person perspective (the actor was represented by a virtual avatar). The experience of both the actors and the observers were recorded according to the Swedish Viewer-User Presence (SVUP) (Västfjäll, et al., 2000) technique, a method of quantifying the subjective sense of presence in virtual environments.

The results demonstrated that the actors had a greater sense of presence in the scene than the observers. The authors conclude that the actor may have had an enhanced sense of presence due to the fact that the cognitive load required to engage with the task distracted them from unrealistic factors of the environment. Conversely, the observers had a low sense of presence due to a low cognitive load giving them more opportunity to observe the realism of the scene, which allowed them to become more perceptive of detrimental factors such as a lack of photorealism of the virtual environment.

This method was successful in recording the experience of the actors and the observers and comparing them to provide richer detail on how particular factors affect the experience of two groups differently. This is a similar methodology to that adopted in the study presented in these chapters; however, in our study we will use a purely objective measurement of actor errors and find the subjective perception of those errors by the audience (which can then be compared objectively).

Chapter 7 : Actor Motion Study

7.1. Introduction

This chapter presents a study that explores the hand placement accuracy of actors while they complete a series of interaction tasks under varying conditions. From the results of this experiment a profile of the estimation errors is constructed and the impacting conditions analysed. The reason for conducting this experiment is two-fold:

1. To capture and analyse the motion of the actors under various conditions to determine which circumstances are beneficial towards supporting bimanual interaction.

2. To quantify the errors created by the actor for use in the later experiment that explores how adept a viewer is at perceiving the errors created when the actor fails to estimate the size or location of the virtual object correctly.

The novel contributions of this study are to construct a set of guidelines for improving actor movement during a bimanual interaction in the interactive virtual studio and present the results in a manner suitable for perceptual studies.

7.2. Background

7.2.1. Models of Human Motion in Human Computer Interaction

Two models of human motion feature prominently in the literature for describing bimanual human motion for human computer (HCI) interaction systems: Fitts' law and Guiard's model of bimanual skill.

Fitt's Law

Fitts' law was one of the earliest measurements of human motion in relation to HCI (Fitts, 1954). This law provides a model of human movement discovered by psychologist Paul Fitts, who observed that the amount of time a person takes to point to or touch an object could be predicted mathematically. The law can be described using Equation 7-1.

$$T = a + b\log_2\left(1 + \frac{2D}{W}\right)$$

Equation 7-1. Fitts' law

Where T is the time taken to complete the movement, a and b are model parameters, D represents the distance between the starting point and the centre of the target and W represents the width of the target measured along the axis of motion. It is important to note that the height or depth of the object is not taken into consideration when using Fitts' law.

In essence, the law states that the larger and closer an object is to the user, the faster they are able to complete the pointing or touching task. Though Fitts' law was developed outside of the HCI domain, it has come to have a significant influence on the design of many User Interfaces (UIs) (MacKenzie, 1992).

In the context of interaction in virtual environments Fitts' law has been applied in studies that have assessed human motion in virtual environments or using virtual environment interfaces. An example is the work of Johnsgard (Johnsgard, 1994), who conducted a study that used Fitts' law to compare the ability of people in using two HCI devices (a wired glove and a standard PC mouse). 6 Fitts' law style tasks were completed by the participants, 3 for the wired glove and 3 for the mouse; each of the 3 tasks was completed using different levels of motion gain (1x, 2x, 3x).

The wired glove was set to control a cursor on a standard PC monitor. It contained an infrared sensor mounted on the back of the hand that would detect a series of infrared light sources positioned around the monitor to interpret which direction the user was pointing towards. The experiment required the participants (n=18) to place the cursor in a specified start location by pointing. A similar scenario was presented using a standard PC mouse, where the participant clicked to begin the task, moved the mouse to the target location and clicked again. For each task the reaction time, movement time and final acquisition times were recorded. From the results of the Fitts' law style tasks the author was able to demonstrate that the mouse consistently outperformed the wired glove in terms of the time taken to complete the task and that higher gains lowered performance predictability. As such, it was demonstrated that the mouse was a superior input than the wired glove for this task, which demonstrates the ability of Fitts' law in analysing human motion for HCI.

However, the issue with applying Fitts' law to our work is that it is primarily user centric. It can be used to describe the time taken to complete a specific pointing task or the error rate that occurs. While this would be relevant in describing the time taken by an actor to identify the surface location of a virtual object, it is not appropriate to assess hand placement accuracy to the surface. As such, it is not thought that Fitts' law is applicable to this study into actor motion.

Guiard’s model

Guiard’s model of bimanual skill (Guiard, 1987) states that many bimanual tasks are asymmetric and describes the role of each hand in the task. The theory states that the role of the non-preferred hand is to lead the preferred hand when performing tasks, whereas the preferred hand produces fine motor activities. The model is presented in Figure 7-1.

Hand	Role and Action
Non-preferred	• Leads the preferred hand. • Sets the spatial frame of reference for the preferred hand. • Performs coarse movements.
Preferred	• Follows the non-preferred hand. • Works within established frame of reference set by the non-preferred hand. • Performs fine movements.

Figure 7-1. Guiard’s model of bimanual skill

An example of this model’s application in HCI is with interaction using a touchscreen tablet. In this case the non-preferred hand is used to orientate the device (a coarse movement) and offer a frame of reference for the user to produce detailed interactions via the touchscreen using their preferred hand (which is capable of a fine movement).

From Guiard’s model asymmetric bimanual interactions are well understood. However, the bimanual tasks presented in this body of work are symmetric, meaning that although Guiard’s model can be used to assess how people interact with computers, it is not directly applicable in the context of this study.

7.2.2. Performance Metrics for Human Computer Interaction in Virtual Environments

In VEs it is typical to assess the usability of a system with performance metrics such as task completion time and task failure/error rate, as they are indicators of the user’s ease of use (i.e. how consistently and quickly the users are able to perform tasks). They are also robust and portable descriptors, allowing disparate systems to be compared for their effectiveness when a user task is common to both systems. This section discusses how these performance metrics have been used. It is followed by a summary of how these performance metrics relate to measurement of actor motion in the virtual studio.

Task Completion Time

Task completion time describes the average length of time it takes a user to complete a particular task in a virtual environment and can be applied to many different tasks such as navigation and interaction.

An applied example of this performance metric is seen in Bowman et al (Bowman, et al., 1999), who presented a testbed for assessing the efficacy of different selection and manipulation techniques for common interaction tasks in a virtual reality system. An experiment was conducted that compared the average participant (n=48) task completion time using different combinations of Selection (Go-Go (Poupyrev, et al., 1996), Ray Casting, Occlusion), Attachment (Go-Go, Scale User, Move Hand) and

Manipulation techniques (Go-Go, Linear Mapping, Buttons). The authors found that using the Go-Go technique (6.57s), participants took significantly longer to select a virtual object than either the ray-casting (3.28s) or occlusion (3.82s) selection techniques. The results from this experiment demonstrably aided the selection of interaction techniques in virtual environments, with time being one of the strongest indicators of performance.

Error Rate

Error rate measures the average frequency at which a user fails to complete a particular task to within an agreed tolerance. An example of error rate is seen in the work of Schuchardt and Bowman (Schuchardt & Bowman, 2007), who used error rate percentage (alongside completion time) as a metric to determine how adept the participants (n=24) were at finding certain features in a 3D model of a cave (e.g. find the highest point of the cave or the shortest route through the cave) using an immersive and non-immersive display. The immersive display was a 4-wall CAVE (Cruz-Neira, et al., 1992) that displayed the virtual cave[4] from the correct perspective relative to the participant's head in stereoscopic 3D, whereas the non-immersive display was the virtual model projected (non-stereoscopic) onto a single wall from one perspective. The authors found that the immersive display led to a much lower average error rate (15%) than the non-immersive display (33%). This allowed the authors to demonstrate that successful task performance was related to the level of immersion the participants experienced.

Distance Error (from Target Location)

Distance Error measures the total distance the user is from a target location after they complete a task – essentially a measurement of spatial accuracy. An example of this performance metric is seen in the work of Teather and Stuerzlinger (Teather & Stuerzlinger, 2007), who constructed a set of guidelines that allowed user performance (n=12) with 3D and 2D input devices to be quantified and assessed when completing 3D tasks using distance error as one of the performance metrics (the other being completion time). The authors effectively assess the usage accuracy of three input devices (a regular PC mouse, a 2DOF tracked wand and a 3DOF tracked wand) when completing two tasks (assembling a virtual chair and moving a virtual cube to a designated target location). The authors also explored the effect that stereoscopic and head tracked feedback had on the participants. They measured the average errors across the participants for each device, providing a mean total distance error. They were able to demonstrate that for both the chair and the cube tasks, using the 3D wand led to significantly larger distance errors than either the 2D wand or PC mouse methods.

The key difference between this performance metric and one required for assessing the nearness of an actor's hand to an interactive virtual object surface is that the error would need to be measured throughout the interaction, not just at the end. If the average distance error from the surface is to be

[4] For clarity, note that 'cave' is the virtual model and 'CAVE' is the feedback system.

explored, then the amount of deviation from the intended path of the actor’s hands could be considered relevant throughout the interaction. This is a feature that has previously been explored in the field of human kinematics (the study of how the human body moves).

An applied example of a performance metric that measures deviation from an intended path in kinematics can be seen in the work of Atkeson and Hollebach, who investigated the kinematic features of both up and down vertical arm movements (Atkeson & Hollerbach, 1985). They conducted an experiment where participants (n=5) moved their arms along a vertical plane at different speeds (slow, medium and fast), using 3 hand-held weight conditions (0lb, 2lb, 4lb) and 2 direction conditions (up and down). Each permutation of these conditions was experienced by each participant throughout the experiment. The study examined the straightness of the path of the participant’s hands, and the authors were able to identify that a curved motion path is experienced with the upward movement, whereas the motion path of the hands was considerably straighter with a downward direction.

Summary of Performance Metrics

As discussed above, several standard performance metrics exist for describing and assessing usability of interactive virtual environments. Despite the proven suitability for assessing the ease of general VR tasks, we see that neither the task completion time or error rate would be suitable for assessing the fidelity of the interaction in the virtual studio (as these would not capture the actor's accuracy in estimating the location of the object's surfaces), and they are therefore outside the scope of this study.

The metric of distance error is similar to that required by our work. However, two differences between this metric and the one required for this study exist:

1. The errors should be measured continuously throughout the interaction and be represented as an average.
2. It will need to measure the distance error of the right and left hands to the right and left sides of the object respectively – resulting in two distance values.

If distance error is to be used, then it will need to take these into account. For point 2, when designing the performance metric the object centroid can be assumed to be fixed to the midpoint of the actor’s hands (as the interactive object is). This allows the sum of the distance error between each of the hands and the respective object surfaces to be presented in a single and convenient performance metric, where the size of the object is simply subtracted from the distance between the hands (e.g. hands placed at 20cm apart for an object 15cm across will result in a total gap of 5cm; a 2.5cm gap between either hand and the respective object surface). These will be applied when designing the performance metrics (section 7.3.4.3. Performance Metrics) for both the mean distance error from the object surfaces and when measuring the amount of variation between the hands.

7.2.3. Assessing actor motion in the virtual studio

The Department of Media at the University of Applied Sciences in Düsseldorf (as previously mentioned in chapter 4.2) is the only group that has previously conducted comprehensive studies into human motion in the virtual studio, where it assessed the effectiveness of their feedback systems.

One experiment (Woldecke, et al., 2009) compared the haptic belt to more conventional feedback systems in the virtual studio, namely verbal (aural) and visual feedback, in a wayfinding task where participants (n=6) had to navigate four predefined motion paths through a virtual set. The performance of the test participants was measured using the metric of squared positional errors, which allowed average deviation from the predefined path to be quantified during a walking task. The results appeared to indicate that the haptic belt allowed navigational performance at least as accurate as visual and verbal feedback techniques, although this could not be confirmed statistically. This is likely due to the relatively close median results and the small sample size of 6 participants.

This study was followed with a further investigation into the effectiveness of different vibrotactile settings with the haptic belt (Vierjahn, et al., 2009). Specifically, the investigation explored the effects of the angle between the tactors on the haptic belt (60° and 90°) and the signalling method of the haptic feedback (intensity signalling, pulse signalling and pulse-intensity signalling). The description of the signalling method, as described by the authors are:

- Intensity-signalling: linear interpolation of tactors' intensities.
- Pulse-signalling: linear interpolation of tactors' active times.
- Intensity-pulse-signalling: a combination of both.

An experiment was conducted that required participants (n=12) to each complete a series of 6 randomised tests containing each permutation of the conditions described above (spacing of tactors, signalling method), where they had to indicate the direction that the vibrotactile belt was directing them towards. Using Mean Angle Error (the difference between indicated angle to object and actual angle) as a performance metric, the authors were able to quantify the accuracy of the participants. The results demonstrated that the pulse-signalling method outperformed the other two modes, allowing a substantially more accurate performance than intensity-signalling and more accurate performance (albeit slightly) than the intensity-pulse-signalling method. Overall, this experiment demonstrated that a pulsed-signalling approach allowed actors in the virtual studio to identify the direction of a virtual object relative to their own location to a fairly accurate level (Mean=6.46° error with 90° tactor spacings).

Having established that vibrotactile belts could be effective in communicating the relative location of a virtual object to an actor, the group decided to apply this feedback to guide the actor's arm (Woldecke, et al., 2010). An experiment was conducted where participants (n=28) completed a task where they moved one of their hands from a starting location to a final target location using one of two feedback modalities; a video monitor that showed the target location in the form of an orange virtual box for 500ms and a haptic belt device consisting of five tactors placed on the actor's arms that vibrated in the direction of the target. Two performance metrics were used in this experiment, final distance error from target location and movement speed. It was found that visual feedback (median=0.22m/s) allowed a considerably faster movement speed than the haptic belt method (median=0.08m/s). The final distance from target location, which in essence measures the ability of the participants to identify a discrete target location, showed that no statistically significant difference could be identified between the Visual (median=0.07m from target) and Haptic (median=0.08m from target) methods.

In the context of the single-handed identification of object locations in a virtual studio, the authors recognised that an actor would struggle to find discrete locations such as the surface of a virtual object. This bears a resemblance to the study presented in this chapter, where the actors also attempt to place their hands near to the surface of a virtual object. As such the framework section presented in this chapter can be seen as a novel and comprehensive expansion on this work into the bimanual domain, which carries with it an extended range of potential areas for actor errors associated with bimanual interaction into account. For example:

- Having to maintain a fixed distance between the hands,
- Having to focus on where two hands are placed instead of one,
- Having to continuously find the surface location of a moving virtual object.

Our framework also extends on Woldecke's study by exploring the other conditions that impact on performance, such as object size, target interaction speed and the positioning of the hands; as well as exploring these elements in both interactive and non-interactive systems.

7.3. Methodology

16 participants (henceforth called actors) were asked to complete a series of short interaction tasks using an interactive system that replicates the functionality of the interactive virtual studio, as described in chapter 6. The objective of each task was to move a 2D virtual object from a "start" location to a "finish" location.

The scope for measuring actor motion with fully interactive virtual objects that have a complex shape and can be moved in 6 Degrees of Freedom (6DOF) is substantial. Therefore the scope of this study is scaled to that of a 2D interactive system that uses a simple square object shape locked to 2DOF motion along the X and Y axes only.

7.3.1. Experiment Design/Conduct

7.3.1.1. Experiment Procedure

The duration of each experiment session lasted approximately 45 minutes and used the following procedure:

Pre-experiment

Each subject was required to sign a consent form and complete a training routine before the experiment commenced. The training routine lasted for approximately 25 minutes was conducted over two stages – a tutorial stage and a practice stage.

The tutorial stage of the training session presented the actors with instructions on the use of the different interaction modalities they would experience and the conditions they would be presented with, which were presented alongside example tasks for them to complete. This stage lasted 10 minutes. The full set of instructions given to the subjects are in Appendix D. The practice stage of the training session presented the actors with a set of 96 tasks. These tasks contained a comprehensive sample of the full range of conditions and interaction modalities they would experience during the experiment, which ensured they had a minimum level of experience before the experiment proper was conducted. This stage lasted 10 minutes followed by a 5 minute break.

All subjects were required to complete the training routine for four reasons:

- To demonstrate their fitness in completing the tasks.
- To demonstrate that they could comprehend the tasks and conditions that were presented to them and understood the nature of the experiment.
- To train them in the use of the interactive virtual studio and allow them to demonstrate that they are capable of interacting with the virtual objects in a competent manner
- To mitigate carry-over effects that would be associated with experience of the interactive system.

Experiment

Once the experiment proper began, it would last approximately 20 minutes. The actor was presented with the tasks in a randomised order, each of which required them to move the virtual object over a distance of 73cm. Each permutation of the conditions for each of the interaction modalities was tested once over the course of the experiment.

Before each task the examiner waited for actors to prepare themselves and confirm that they were ready, at which point the task was initiated by the examiner who could control the flow of the experiment via a standard PC keyboard. For the few tasks where a false positive in the tracking data was detected, they were repeated at the end of this session.

Post experiment

The actors were asked if they experienced any issues or felt tired throughout the experiment. No issues were reported. The actors were then debriefed and dismissed.

7.3.1.2. Arrangement

The arrangement of the experiment is presented in Figure 7-2 (page 93). The actors stand in a static location with their hands 2 meters away from a Kinect sensor. The Kinect tracked the location of their hands using the depth camera and also provided the “studio camera” video feed, using its inbuilt RGB camera. The Kinect was connected to an adjustable tripod, which was fixed to the height of the actor’s shoulder before the experiment to ensure that all interactions are measured relative to this point, meaning that the movement was comfortable and the experience was similar for all actors.

The feedback monitor was positioned to the left of the Kinect. Ideally it would be central to the actor’s location, but this was not compatible with the arrangements required for the Kinect. Placing the monitor off-centre is also an acceptable design decision however, as it mimics a probable scenario in the virtual studio, where visual feedback is unlikely to be placed directly in front of the actor at all times. For these tasks the actors will have to turn their heads slightly, but they will still have static feedback in their gaze location similar to a robust and stable version of ScaMP.

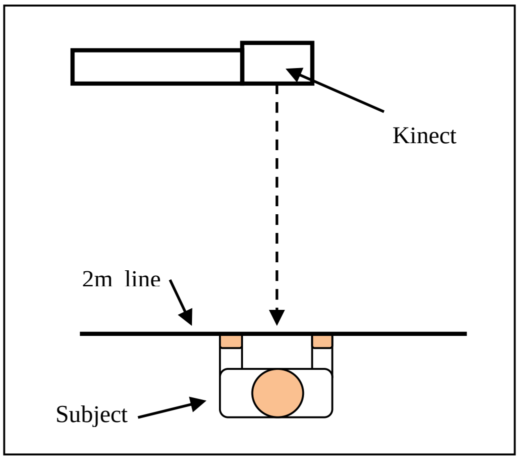


Figure 7-2. Diagram of test arrangement

7.3.1.3. Feedback Presented to Actor

For this experiment feedback was presented to the actor in a specially adapted form from typical virtual studio displays that usually show standard broadcast output. The monitor used was a 32 inch Elonex lumina with a resolution of 1,366x768. Displayed on the feedback monitor was the input from the RGB camera, which had been mirrored horizontally to make orientation based on the feedback video easier for the actor (a common procedure in virtual studios) and presented in a 4:3 format. Also included were on screen graphics to assist the actors and instruct them on the nature of the task they had to complete, which are annotated in Figure 7-3 (page 94); these were partially based on the advice of two actors who had previously used the system. The following on screen graphics were provided:

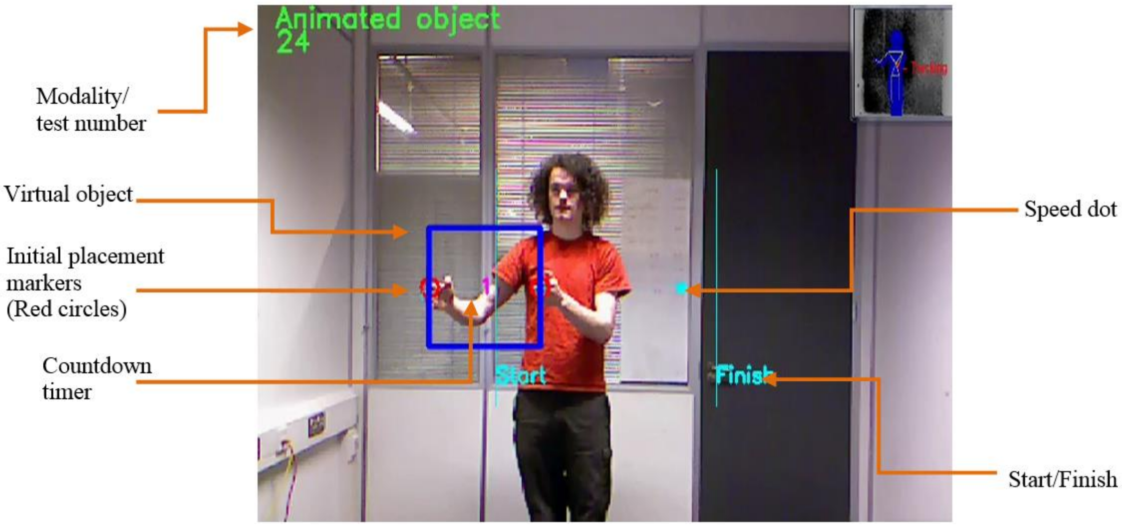


Figure 7-3: Annotation of actor feedback

Modality – The object modality (interactive, animated or no-object) was presented to the actor in the top left corner of the monitor before the test commenced.

Number of task - Each task was assigned a number, which was presented in the top left hand corner of the monitor. This was for the assessor, who used the number to organise any repeats or take notes if necessary.

Speed dot – This dot indicated the movement speed (for the Animated modality) or target speed (for the Interaction modality) of the object motion by travelling across the monitor at the intended velocity. This was shown to the actor before the task commenced and illustrated the speed and direction of movement.

Initial hand placement markers - Before the task started, two red circles were placed on the border of the virtual object. These circles showed the actors where their hands needed to be placed relative to the virtual object for the task. When the task started these circles disappeared.

Hand position markers - The location of the actor's hands, as interpreted by the Kinect, were represented by a yellow circle. The purpose of this was two-fold. Primarily it allowed the assessor to determine when and where false positives occurred. Secondly it allowed the actors to observe the tracked location of their hands, which would aid them when lining their hands up with the initial hand placement markers. These markers were present throughout the entire experiment.

Countdown timer - Before each task commenced a timer was shown inside the virtual object counting down from 5 seconds until the task began. This allowed the actor to prepare for the task.

Start and finish line - A start and finish line were added to show the direction of the interaction. In addition, hand measurements were only taken when the centroid of the object was between these two lines.

Audio – in addition to visual feedback, one audio tool was used to assist the actor. Each time the countdown timer reduced by 1 second, a short beep of 8KHz was played to the actor. A single beep of 10KHz was played when the timer reached 0 and the task started.

7.3.2. Apparatus

Hardware Specifications. The experiment was conducted using a PC with a Quad-core 3.2GHz CPU, 8GB RAM, 2GB Eyefinity6 GPU, with a 32-bit Windows 7 Operating System.

Motion Capture System. From a study of potential motion capture techniques, the Microsoft Kinect (Microsoft, 2011) interfaced with OpenNI (Primesense, 2011) was found to be a sufficiently convenient and accurate method of motion capture. This is likely due to the fact that the ideal requirements of the Kinect closely match the conditions of the experiment (i.e. where the user stands upright ~2m away from the sensor unit under controlled lighting conditions with their hands away from their torso).

The selected tracking point used for both measurement of actor motion and interaction for this experiment is the palm of the actor, which when pointed towards the Kinect becomes the tip of middle finger (as they were asked to do). However, before proceeding with the experiment, to ensure the accuracy of the Kinect was a sufficiently reliable for tracking this point it was calibrated under the conditions of the experiment against a proven marker-based tracking system, Vicasso (Pixoft, 2009). If the Kinect is capable of achieving comparable results to Vicasso, then the accuracy is deemed sufficient for use in the experiment.

The location of the right hand middle fingertip was recorded using both methods as actors completed a series of vertical and horizontal movements with their hand pointing towards the camera (as they would in the experiment). A total of 185 data points representing 7.5 seconds of movement were collected along the X and Y axes. The performance metric used was the mean difference between the two systems, which was calculated using Equation 7-2. Here D represents the mean difference, N is the number of data points, V the motion capture data recorded by Vicasso and K is the motion capture data recorded by the Kinect. The data was captured and described using pixel space, although the conversion to cm with the marker at a 2 metre distance is provided here too.

$$D = \frac{1}{N}\sum_{i=1}^{N} V_i - K_i$$

Equation 7-2. Mean difference between the Kinect and Vicasso.

The motion capture data from both techniques is presented in Figure 7-4, which illustrates how closely the data captured from the Kinect (red line) matches the data from Vicasso (blue line). A two tailed T-

Test, a form of statistical test that describes whether any statistically significant difference exists between two sets of data, was conducted to determine whether any difference exists between the two tracking methods. No statistically significant difference was observed between Vicasso and Kinect for the measurements of the hand on the X-axis ($T_{(184)}$=0.214, p=0.831), where a mean difference of 0.3px (0.11cm) was recorded. No statistically significant difference was observed on the Y-axis either ($T_{(184)}$=0.312, p=0.755), where a mean difference of 0.24px (0.087cm) was measured. The lack of any statistically significant results and the low amount of mean difference between the techniques confirms that the Kinect is as accurate as Vicasso.

As demonstrated, the Kinect is a sufficiently reliable and accurate approach for capturing the motion of an actor under the conditions of this experiment, closely matching the results of an established motion capture technique and thus suitable for use in this experiment.

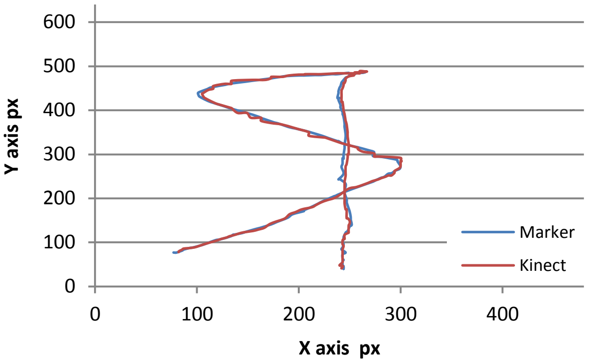


Figure 7-4. A comparison of motion capture using a visual marker (Vicasso – blue line) and the Microsoft Kinect (red line) under the conditions of the experiment. Methods closely match.

Interaction Interface. For the interaction interface, the square virtual object used for the experiment is locked to the midpoint of the actor's hands along the X and Y axes as measured by the Kinect. The rotation of the virtual object was locked to 0° throughout the experiment as including this could interfere when looking at the interaction between the Axis of Hand Placement and the Axis of Motion conditions

7.3.3. Participants

The 16 actors that participated in the experiment were all either staff or students from the faculty of Technology, Engineering and the Environment at Birmingham City University. Those that took part in the experiment had a mean age of 28.75, with an age range of 19 to 46 years. None of the actors had any prior experience with the system used in this experiment, or had any background in the Virtual Studio. The gender of the 16 actors were 14 males and 2 females. The preferred handedness of the actors were 14 that were right handed and 2 that were left handed.

In the pre-experiment stage the actors were asked whether they have been diagnosed with any motion or general fitness issues. None of the actors reported any motion or fitness problems and appeared to be healthy. Each actor further proved their fitness during the training routine, after which they were asked the following questions:

- Do you feel tired or exhausted?
- Did feel any pain or discomfort?
- Did you feel as though you were able to comprehend the nature of the tasks?
- Do you understand how to use each of the modalities?

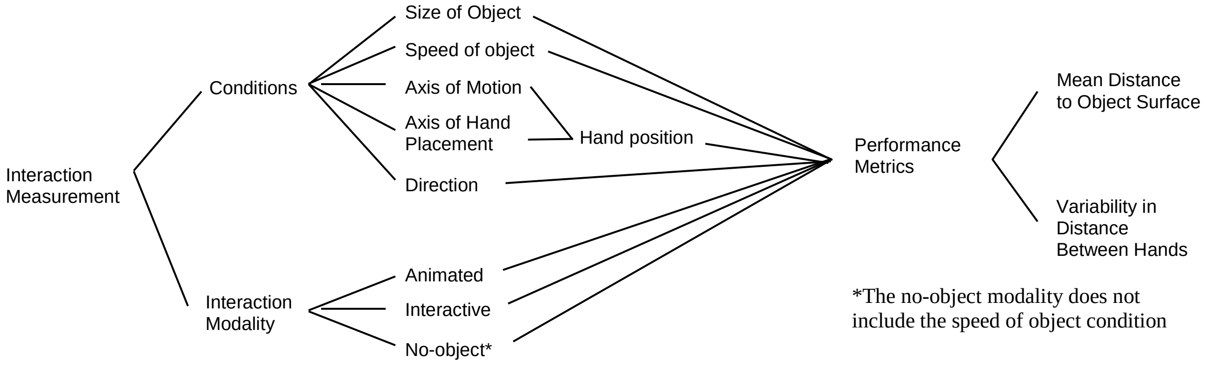
To progress to the experiment, the actor had to respond no to the first two questions and yes to the second two. The assessor had to concur with these answers, with the decision to let the actor proceed to the experiment made at their own discretion. The actor was then asked if they would like to repeat any section of the training routine or had any further queries. The following considerations were also made:

Randomisation - The order of the tasks were presented in a pseudo-random order to mitigate any effects that may come from boredom by forcing the actor's to consider and comprehend the task before they completed it, thereby maintaining their attention. This randomisation also mitigates any carry over effects that would be associated with experience.

Breaks - To combat fatigue, each participant was offered 6 breaks of 30 seconds, which could be taken at any point between tasks. However, due to the general fitness of the participants and the low energy requirements of the tasks, no participant felt the need to use these breaks.

7.3.4. Conditions

Throughout this section the interaction modalities used (section 7.3.4.1.), the experiment conditions (section 7.3.4.2.) and the performance metrics (section 7.3.4.3.) are described. A taxonomy demonstrating how these relate to each other is presented in Figure 7-5, where the permutations of the conditions and interaction modalities can be seen, as well as how they relate to the performance metrics. Throughout the experiment each actor is presented with every permutation of the interaction modalities and experiment conditions once, resulting in a total of 168 tasks for each session.



*The no-object modality does not include the speed of object condition

Figure 7-5. Taxonomy of interaction modalities, conditions and performance metrics

7.3.4.1. Task Modalities

Three interaction modalities were tested in this experiment: animated, interactive and no-object. The three modalities are summarised in Table 7-1 and are followed by a description of the procedure for each in Figure 7-6, Figure 7-7 and Figure 7-8.

Modality	Description
Interactive	The interactive modality represents a basic bimanual interaction system possible in the interactive virtual studio, where the virtual object centroid is locked to the mid-point of the actor's hands.
Animated	The animated modality represents a technique where the actor mimes an interaction with an object that follows a pre-defined path.
No-object	The no-object modality represents a scenario when capturing footage for creating interaction with a CGI object in post-production, where no information on the object's location is provided to the actor.

Table 7-1. Experiment Modalities

Interactive Modality - The purpose of the Interactive modality is to measure the performance of the actor as they bimanually interact with an interactive virtual object. The objective for the actor is to move the interactive virtual object from the starting location to the finishing location, attempting to match the speed indicated to them by the speed dot (described further in page 94).

At the start of the task the hand placement markers and the start and finish lines are shown to the actor. The actor positions their hands ready for the interaction. When they signal to the examiner that they are prepared the countdown timer starts. When the countdown timer reaches 0 the speed dot and the virtual object are then shown to the actor, at which point the virtual object becomes interactive. The virtual object is locked to the midpoint of the actor's hand locations and they will move the virtual object along the path indicated by the speed dot and start/finish lines (these elements are described in page 94). A sequence of images taken from one of these tasks is presented in Figure 7-6.

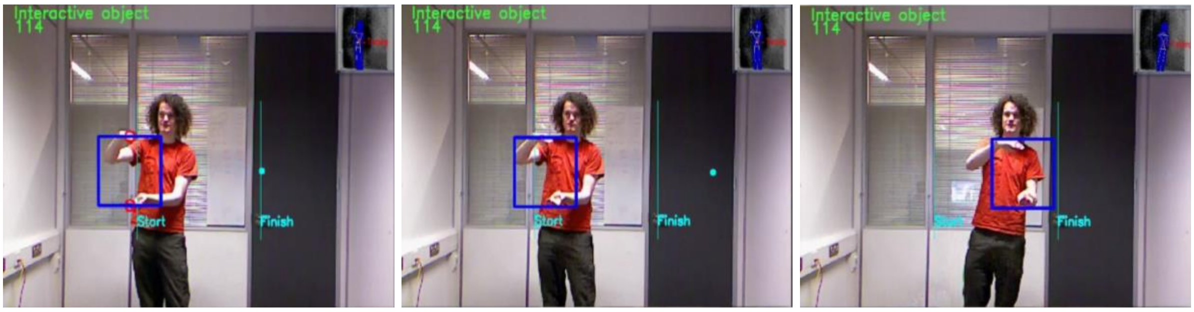


Figure 7-6: Screen captures of an actor completing an Interactive task.

Animated Modality - The purpose of the Animated modality is to measure the performance of the actor if they were to attempt to simulate interaction with a virtual object that travelled along a pre-defined path. The objective for the actor is to attempt to appear as though they are holding (or interacting with) the virtual object as it travels along the path by matching their hands as closely to the object surfaces as possible.

At the start of the task the virtual object, the hand placement markers, the start and finish lines, the speed dot and the countdown timer are shown to the actor (all described in page 94). The actors place their hands inside the hand placement markers and signal to the examiner that they are prepared for the task. The countdown timer will start and the actor will note the velocity of the speed dot. When the countdown timer reaches 0, the virtual object moves along the animated path. A sequence of images taken from one of these tasks is presented in Figure 7-7.

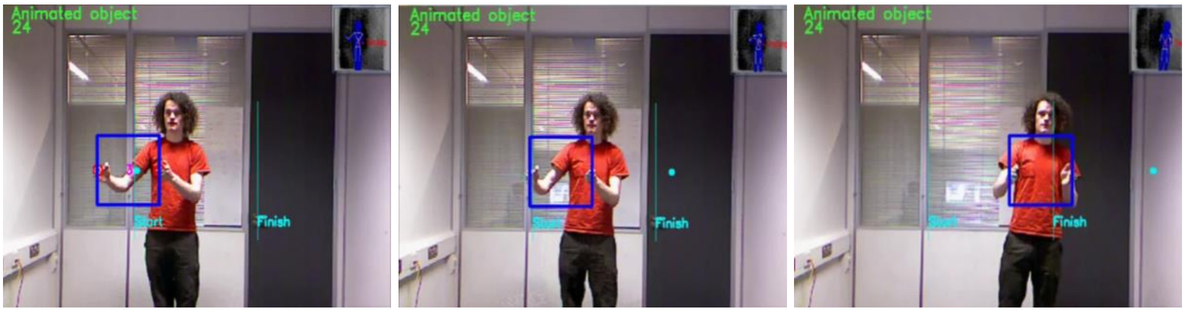


Figure 7-7: Screen captures of an actor completing an Animated task.

No-object Modality - The purpose of the no-object modality is to measure the performance of the actors as they mime an interaction where no virtual object is presented to them in the feedback, allowing their performance when no-object boundaries are presented to them to be assessed. The objective for the actors is to mime interaction with a virtual object from the starting location to the finishing location, attempting to maintain their hands a fixed distance apart. There is no speed dot and the actors are allowed to move at a speed that is comfortable to them.

At the start of the task the hand placement markers and the start and finish lines are shown to the actor. Once the actor signals they are ready the countdown timer will start. When the countdown timer reaches 0, the hand placement markers disappear and the actor has to mime the interaction with the virtual object. A sequence of images taken from one of these tasks is presented in Figure 7-8.

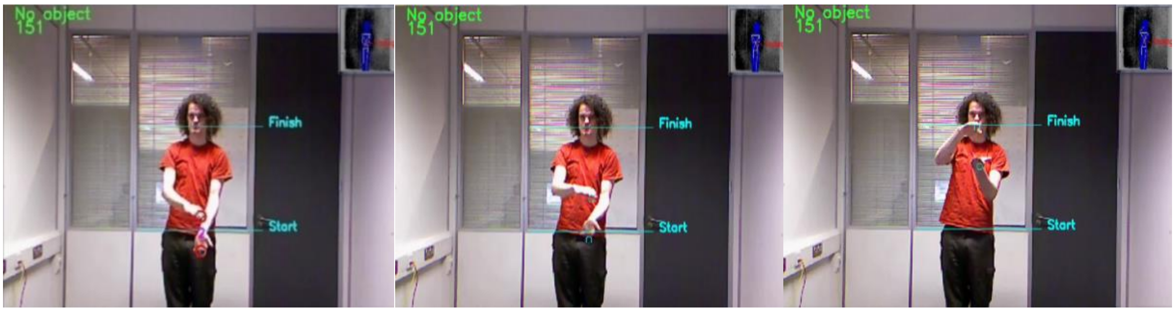


Figure 7-8: Screen captures of an actor completing a no-object task.

7.3.4.2. Experiment Conditions

For the animated and interactive modalities five conditions are used, as presented in Table 7-2. For the no-object modality only four conditions are used, which are the same five conditions present for the animated and interactive modalities, excluding Target Speed.

Condition	Definition	Levels
Size	The size of the virtual object.	18.2cm, 36.4cm, 54.4cm
Target speed	The speed at which the actor aims to complete the interaction.	0.17m/s, 0.26m/s, 0.35m/s
Axis of Motion	The axis along which the interaction takes place.	Horizontal (H), Vertical (V)
Axis of hand placement	The sides of the object the hands are placed on, can either be Top & bottom (vertical) or Left & Right (horizontal)	Left-Right (LR), Top-Bottom (TB)
Direction	The direction of the interaction.	To the Left, To the Right, Upwards or Downwards

Interacting conditions	Definition	Levels
Hand Position	A condition derived from the statistical interaction between the Axis of Motion and Axis of Hand Placement conditions. Represents hand placement relative to axis of motion.	V-LR, V-TB, H-LR, H-TB.

Table 7-2. Experiment conditions

Further to these five conditions the statistical interaction between the Axis of Hand Placement relative to the Axis of Motion is discussed (henceforth known as 'Hand Position'). Hand Position is used to determine how the Axis of Hand Placement and the Axis of Object Motion together impact performance by looking at the results of the statistical interaction between these two conditions. Hand Position is discussed in depth as several important results emerge from it.

This concept is illustrated in Figure 7-9. Here a Left/Right hand placement would be considered a *Horizontal* hand placement, and a Top/Bottom hand placement would be considered a *Vertical* hand placement. If the axis of hand placement and the axis of motion both match (i.e. both are either vertical or horizontal), then it is considered to be motion *With* the axis of hand placement. If they do not match, then they are considered to be *Against* the axis of hand placement.

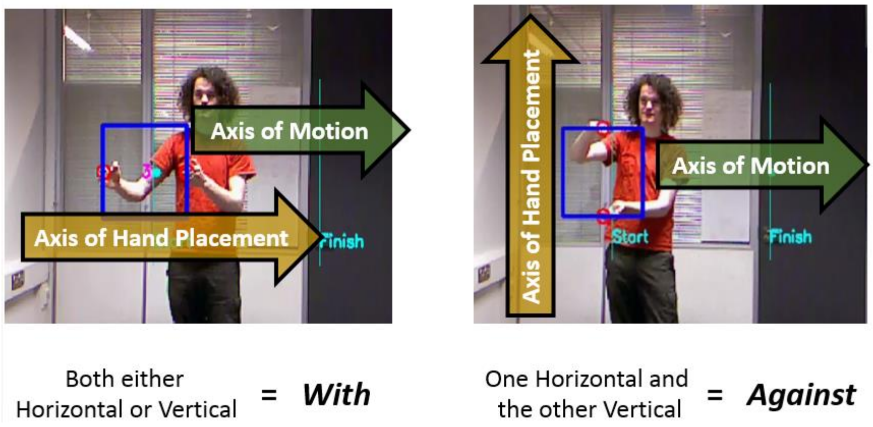
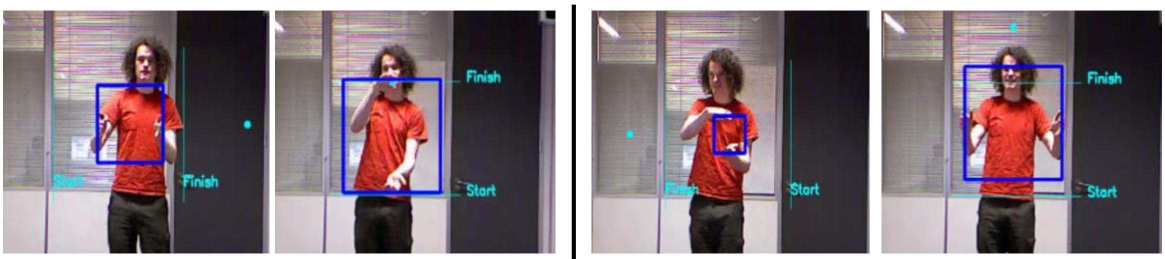


Figure 7-9: Illustration of hand placements *with* and *against* the axis of motion

An example of the kind of effect that would be observed is if the actor moves a virtual object along a horizontal motion path (Axis of Motion). Different levels of performance accuracy might be observed depending on whether they hold the object on the Left and Right sides or the Top and Bottom sides (Axis of Hand Placement).

The levels are represented in initial form (Axis of Motion: H=Horizontal, V=Vertical; Axis of Hand Placement: LR=Left/Right, TB=Top/Bottom), with the first initial representing the axis of motion and the second initial the axis of hand placement. e.g. H-TB represents a horizontal axis of motion and a vertical axis of hand placement. Examples of these Hand Positions and the corresponding initials are presented in Figure 7-10.

All conditions are discussed in the results section, where a particular focus is applied to the Size, Speed and Hand Position conditions, which are analysed in section 7.5.1 to 7.5.3. The Axis of Motion, Direction and Axis of Hand Placement conditions are all analysed in section 7.5.4.



H-LR V-TB H-TB V-LR

Hand positions *With* the axis of hand placement

Hand positions *Against* the axis of hand placement

Figure 7-10: Hand placement relative to the interaction motion. Where H=Horizontal axis of motion, V=Vertical interaction axis of motion, LR=left/right hand placement and TB=top/bottom hand placement

7.3.4.3. Performance Metrics

Two key Performance metrics are reported on to assess the accuracy of the actor's hand placement, namely, the Mean Distance to Object Surface (MDOS) (Equation 7-3, page 102) and the Variability in Distance Between Hands (VDBH) (Equation 7-4 - page 102). These are indicators of the actor's estimation of the object size and their degree of movement around the outline respectively.

The location of the actor's hands in pixel space was used as the unit of measurement for these performance metrics. However in this chapter the results are described in Centimetres (cm). Measurement in cm is calculated by multiplying the measurements in pixels by a constant factor of 0.3636, which is made possible due to the actor's placing their hands at a fixed distance of 200cm from the sensor. Pixel units represent the motion errors as they appear when projected in 2D as a 640x480 image under the conditions of this experiment (which is typical of the audience experience), and are used in the viewer perception study in chapter 8.

$$\mathrm{MDOS} \;\; = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{R}-\mathrm{L}) - \mathrm{W}$$

Equation 7-3. Mean Distance to Object Surface (MDOS) - when measuring along the axis of hand placement

MDOS gives the average measurement of distance between the hands minus the size of the virtual object during the task. This provides sum distance between the hands of the actor and the relevant sides of the virtual object, which informs on the total error of the actor's estimation of object size. The MDOS error can be positive, indicating an overestimation, or negative, indicating an underestimation by the actor. The assumed visual effect of an overestimation is a gap between the hands of the actor and the object surfaces, exposing some of the background. The assumed visual effect of an underestimation is the appearance of the actor's hands intersecting the object's surfaces. The requirement for this performance metric is to be as close to 0cm as possible[5].

For the MDOS equation the following notations are used:

- N represents the number of frames of captured data during the interactive task,
- L and R represents the location of the Left and Right hands of the actor,
- W represents the width of the virtual object (only subtracted when measuring along the axis of hand placement)

$$\mathrm{VDBH} \;\; = \sqrt{\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left((\mathrm{R}-\mathrm{L})-\mathrm{W}\right)^{2}}$$

Equation 7-4. Variability in Distance Between Hands (VDBH)

VDBH represents one standard deviation in distance between the hands of the actor during any particular task (i.e. variation to 68%). This provides the amount of variance in the actor's hand distance to object surface over the course of the interaction to 68%. This informs on the stability of the actor's hand positioning relative to the surface of the virtual object during the interaction, with a large VDBH indicating that there is a high degree of variability in the actor's estimation of the object's surfaces. The assumed visual effect of this is the actor's hands moving in and out from the object surface, creating an apparent disconnect between the hands and the virtual object. The requirement for this performance metric is to be as low as possible with 0cm being the ideal result for each subject across the interaction task. The notations are the same as MDOS (Equation 7-3). This metric does not account for 3D hand motion or diagonal placement, but should be seen as a base on which these metrics can be developed.

[5] For the Animated modality the virtual object cannot be assumed to be fixed to the midpoint of the actor's hands due to the lag/lead error. In this case it just describes how large the actor thinks the object is but does not describe the distance between the nearest surface of the virtual object and their hands. For the purposes of making a simple comparison the MDOS is powerful enough to describe the mean total distance between the actor's hands and the object surface for the Interactive and no-object modalities. This is further discussed in section 7.6 and is taken into account when assessing the performance for question 2.

7.3.4.4. Measurements Along and Perpendicular to Axis of Hand Placement

Each of the performance metrics are presented using two descriptors, Along the Axis of Hand Placement (henceforth referred to as AlongAHP) and Perpendicular to the Axis of Hand Placement (henceforth referred to as PerpendicularAHP). In essence, these descriptors describe hand movement towards and away from the object boundary (AlongAHP) and movement across the object boundary (perpendicularAHP).

Figure 7-11 (page 103) presents an illustration of these descriptors, where the Axis of Hand Placement is indicated by the blue arrow (in this case LR), and the orange arrows show movement that would be considered AlongAHP and PerpendicularAHP.

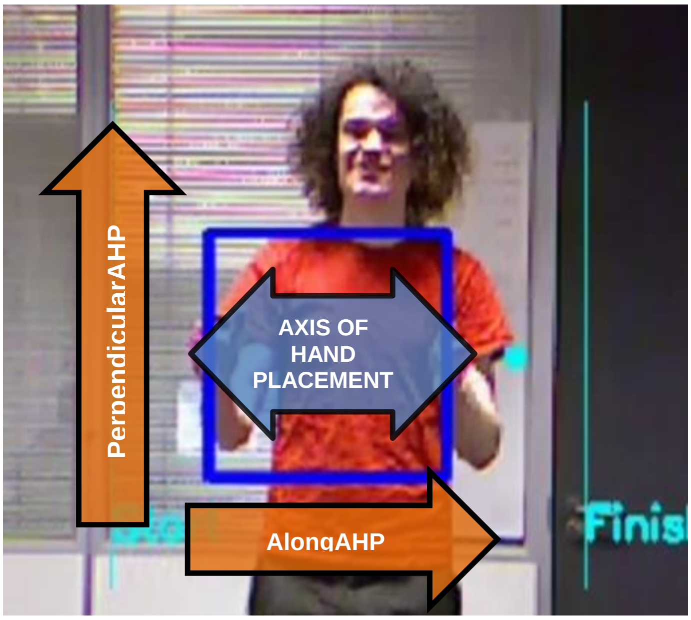


Figure 7-11. Illustration of hand motion relative to the axis of hand placement. The blue arrow shows the Axis of Hand Placement (AHP) and the orange arrows show what would be considered measurements Along and Perpendicular to the Axis of Hand Placement.

7.3.5. Data Analysis

7.3.5.1. Removal of outliers

The removal of outliers was conducted on a task by task basis for each performance metric. To ensure that only the extreme outliers were removed, outlier removal to 3 standard deviations from the mean was conducted, as this level was discovered to be high enough to ensure that only the extreme outlying results would be eliminated. Represented in mathematically, the outlier removal process is:

if $u_{jk} \geq \bar{u}_k + 3\,\delta_{jk}$ *then reject actor for task k*
if $u_{jk} \leq \bar{u}_k - 3\,\delta_{jk}$ *then reject actor for task k*

Where j represents the actor, k represents the task, u the result of the actor (MDOS or VDBH), $\bar{u}$ represents the mean result of all actors and δ represents the standard deviation.

7.3.5.2. Statistical Analysis

The results are analysed for statistical significance using an 'Analysis Of Variance' (hereafter referred to as ANOVA), a type of statistical test that describes whether there is a significant difference between the mean for each level of each condition or whether there is an interaction between the conditions themselves.

The factorial design of the ANOVA, the order in which the conditions are analysed, is conducted in a 3x3x2x2x2 design[6] for the Animated and Interactive modalities and in a 3x2x2x2 design[7] for the no-object modality. The data was analysed using its captured pixels, but the equivalent cm units when converted are provided in the results section.

This analysis uses a 'repeated measures' design, which in the context of this study means the performance of each participant is measured for each combination of conditions. To fulfil the requirements of the repeated measures design, the following assumptions must be satisfied:

1. **Normality:** The data for each task must have a normal distribution. Tests for kurtosis and skew of the data found that the majority of the tasks had normally distributed data.
2. **Sphericity:** To detect violations for sphericity, a Mauchly's test for sphericity is conducted for each condition. Sphericity was found not to be violated throughout the study, except for in a few cases, where a Greenhouse-Geisser correction was applied to correct for this. Violations of sphericity are denoted by the symbol '^' throughout.
3. **Randomness:** Participants should be independent of each other and must represent a random sample of the population. The selection of subjects represents a random sample from an able bodied population.

The ANOVA only reveals whether a difference is present between the levels of each condition, but does not describe where these differences occur. To gain this information, post-hoc analysis must be performed where a statistically significant effect is detected in the ANOVA. The post-hoc analysis method selected for this study is the Tukey HSD, which analyses for statistical significance between each level of each condition, allowing the trends to be identified.

[6] Size x Speed x Axis of Motion x Axis of Hand Placement x Direction

[7] Size x Axis of Motion x Axis of Hand Placement x Direction

For both the ANOVA and Tukey HSD comparison tests an alpha of 5% is used, which means that any p-value result <0.05 is deemed statistically significant.

7.4. Hypotheses

The following three null hypotheses will be assessed both AlongAHP and PerpendicularAHP for the Interaction modality only (to address research question #1 – below):

1. Actor performance is not affected by the size of the object
   a. *The Mean Distance to Object Surface is not related to the size of the object.*
   b. *The amount of variation between the actor's hands is not related to the size of the object.*

2. Actor performance is not affected by the speed of the object
   a. *The Mean Distance to Object Surface is not related to the speed of the object.*
   b. *The amount of variation between the actor's hands is not related to the speed of the object.*

3. Actor performance is not affected by the placement of their hands with regards to the motion of the object
   a. *The Mean Distance to Object Surface is not related to the actor's hand position.*
   b. *The amount of variation between the actor's hands is not related to the actor's hand position.*

The following research questions will be addressed in the discussion section (7.6):

1. What is the Mean Distance to Object Surface (MDOS) and the amount of Variability in Distance Between Hands (VDBH) the actor is likely to achieve when moving a virtual object using the Interactive modality?

The primary research question of this study, this question asks what conditions affect the performance of an actor when moving an interactive virtual object in the virtual studio, and these will be measured using two key performance metrics. The first performance metric measures the ability of the actor to correctly estimate the size of the virtual object (quantified as MDOS). The second measures the ability of the actor to maintain a steady hand placement (quantified as VDBH), which is made difficult by the lack of any rigid surface to support their hands. Using these performance metrics, the effects that Object Size, Interaction Speed, Axis of hand Placement, Axis of Object Motion and Direction of Interaction have on performance will be analysed.

2. Are the MDOS and VDBH of the Interaction modality comparable to those of the Animated modality?

This question asks whether using an Interactive virtual object is preferable to mimicking the interaction using a pre-animated virtual object (as suggested in (Gibbs & Baudisch, 1996)) in terms of actor performance once the lag/lead error of the animated modality is taken into consideration.

In the lag/lead error the centroid of the object and the midpoint of the actor's hands do not match, meaning that the object is either travelling ahead of or behind the actor's hands. The error is the result of the actors incorrectly estimating the speed of a pre-animated virtual object as they try to match their hands to the location of the moving object's surface, which is an error not present for the Interactive modality. If the MDOS and VDBH of the Interactive modality yield similar or comparable results to the Animated modality after the lag/lead error is taken into account, then it can be suggested as a preferred modality for creating the interaction.

3. Are the MDOS and VDBH of the no-object modality comparable to those of the Interactive modality?

This question asks whether the interactive modality is preferable to the no-object modality in terms of actor performance. If the actors were to interact with a virtual object that will be added in post-production they will typically mime the interaction with their hands unconstrained in the studio, with the CGI object mapped to accurately fit their hands in post-production (occasionally the movement of their hands are constrained, but we assume a case where the hand motion is unconstrained as shown in Figure 7-12- which the no-object modality is modelled on).

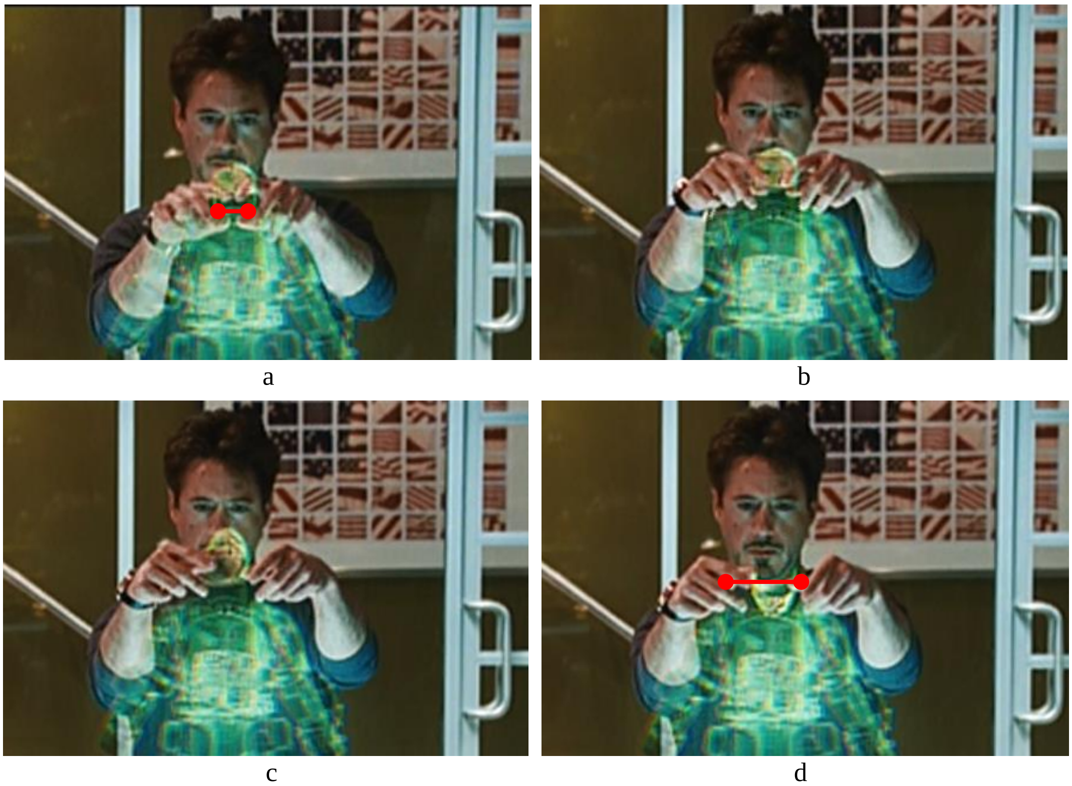


Figure 7-12. A series of screen captures from the film Iron Man (Iron Man, 2008) where Tony Stark is designing the Iron Man suit using JARVIS, a holographic interface created using CGI. In this scene Robert Downey jr removes the helmet from the CGI Iron Man suit, which requires an unconstrained bimanual interaction. Despite the benefit of multiple takes, a variation in the distance between his hands can still be seen where they move apart.

Despite the advantage of being able to fit the object to the hands in post-production, the actor is still liable to vary the distance between their hands or misestimate the intended object size. This is illustrated in Figure 7-12, where from image ‘a’ to image ‘d’ a variation in the distance between Robert Downey jr’s can be observed, where they move apart from each other (indicated by the red lines in these images – which indicate the distance between the index fingers). If the performance of the actor using the Interactive modality can lead to a consistently lower amount of variation between their hands and a better estimation of object size than the no-object modality, it can be recommended for use as an effective tool for aiding accurate actor motion when capturing footage for post-production.

7.5. Results

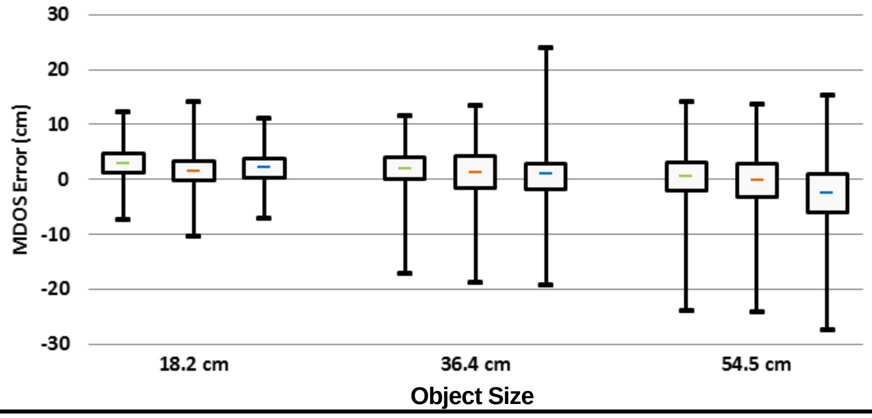
This section will first address the three hypotheses by analysing the Size (section 7.5.1.) and Speed (7.5.2.) conditions of the study, followed by an analysis of the interaction between the Axis of Motion and the Axis of Hand Placement conditions (7.5.3.). For each of these hypotheses the distance between the hands (MDOS) and the amount of variability in the distance between the hands (VDBH) will be analysed both Along (AlongAHP) and Perpendicular (PerpendicularAHP) to the Axis of Hand Placement (as described in Figure 7-11, page 103). This is followed by a summary section that aims to explore the results in a combined and more general manner, discussing the overall observation of trends and guiding the construction of recommendations. Finally all other main effects for the Axis of Motion, Axis of Hand Placement and Direction conditions are discussed in 7.5.4.

7.5.1. Size of virtual object

This section presents our findings on how the virtual object size condition affects an actor's performance. Three object sizes (levels) are used: 18.18cm (which translates to 50px at 2m from the sensor), 36.36cm (100px) and 54.54cm (150px).

7.5.1.1. Analysis of Distance between Hands

Along the axis of hand placement



Size	18.2cm		36.4cm		54.5cm		
Modality	Mean	SD	Mean	SD	Mean	SD	ANOVA
Animated^	2.75	3.39	1.64	4.05	0.12	5.15	$F_{(1.304, 11.737)}$=42.44, p=<0.001
Interactive	1.59	3.53	0.96	5.18	-0.52	5.95	$F_{(2,8)}$=4.327, p=0.027
No-object^	2.14	3.41	0.59	5.56	-2.92	6.75	$F_{(1.284,12.837)}$=21.46, p=<0.001

Figure 7-13. Mean results for all tasks under the size conditions along the axis of hand placement The box plot shows the median, the 1st and 3rd interquartile ranges and the maximum and minimum recorded values. The Animated modality is represented in green, the Interactive modality in orange and the No-object modality in blue (^Greenhouse-Geisser correction for sphericity violation is made).

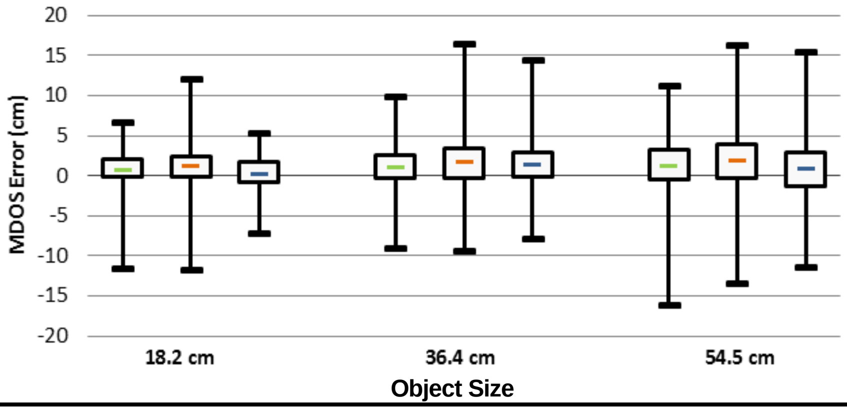
For each modality when measured AlongAHP, the size of the virtual object had a statistically significant effect on the mean distance between the actors' hands normalised to the size of the object (MDOS) for the Animated (p=<0.001), Interactive (p=0.027) and no-object (p=<0.001) modalities. For each modality the trend for the 'mean MDOS' (henceforth referred to as MDOS throughout section 7.5) showed a decrease as larger virtual objects were used, accompanied by an increasing standard deviation. These results are presented in Figure 7-13 (page 108).

These effects were confirmed with post-hoc analysis, where for each statistically significant effect between size levels the larger object yielded a lower MDOS. For the Interactive modality a statistically significant effect was detected between the 36.4cm-54.5cm levels (p=0.03) and the 18.2cm-54.5cm levels (p=0.017), but not between the 18.2cm-36.4cm (p=0.478). The total difference between the 18.2cm and 54.5cm levels was -2.11cm.

For the Animated modality a statistically significant difference was present between each level (18.2cm-36.4cm: p=0.001, 36.4cm-54.5cm: p=<0.001, 18.2cm-54.5cm: p=<0.001). The total difference between the 18.2cm and 54.5cm levels was -2.63cm.

For the no-object modality a statistically significant difference was present between the 36.4cm and 54.5cm levels (p=0.002) and the 18.2cm and 54.5cm levels (p=<0.001), but not between the 18.2cm and 36.4cm levels (p=0.478). The total difference between the 18.2cm and 54.5cm levels was -5.06cm.

Perpendicular to the axis of hand placement



Size	18.2cm		36.4cm		54.5cm		
Modality	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**ANOVA**
Animated	0.87	1.97	1.08	2.61	1.28	3.37	$F_{(2,8)}$=1.45, p=0.264
Interactive	1.16	2.35	1.64	3.27	1.79	3.82	$F_{(2,8)}$=2.084, p=0.148
No-object	0.36	1.95	1.40	2.85	0.90	3.65	$F_{(2,8)}$=1,59, p=0.245

Figure 7-14. Results for all tasks under the size conditions perpendicular to the axis of hand placement

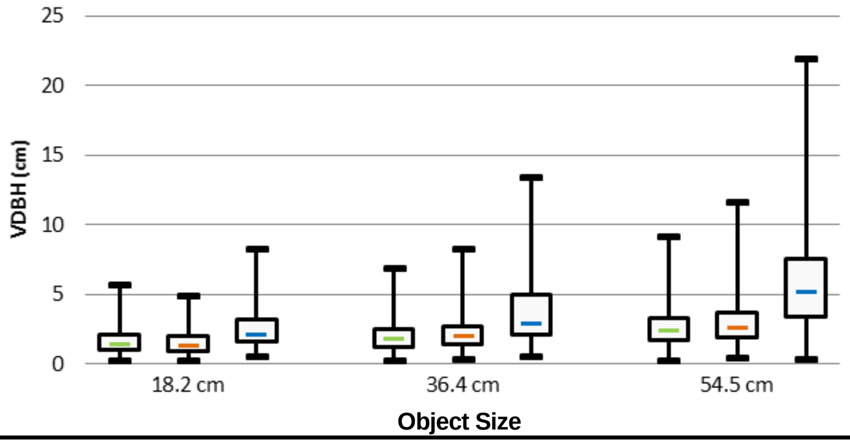
The effects observed for the MDOS when measured PerpendicularAHP[8] demonstrated that the size of virtual object had no statistically significant effect on the distance between the actor's hands for the

[8] **Note:** As previously discussed, with an MDOS error PerpendicularAHP the hands are still in contact with the object surface and should not appear unrealistic to a viewer. As such, the PerpendicularAHP MDOS results are primarily provided here only for completeness and has very little bearing on interaction quality.

Interactive (p=0.148), Animated (p=0.264) and no-object modalities (p=0.245). As no statistically significant effects were detected, post-hoc analysis was not performed. Despite the fact that no statistically significant results were detected between the means of the size levels, it was found that the standard deviation of the MDOS increases as larger virtual objects are used (a factor that the ANOVA gives no information about), which represents a wider distribution of results. These results are shown in Figure 7-14 (page 109).

7.5.1.2. Analysis of Variability in Distance Between Hands

Along the axis of hand placement



Size	18.2cm		36.4cm		54.5cm		
Modality	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**ANOVA**
Animated	1.68	0.96	2.02	1.20	2.59	1.61	$F_{(2,8)}$=25.08, p=<0.001
Interactive^	1.56	0.88	2.19	1.34	2.96	1.93	$F_{(1.31,14.38)}$=47.16, p=<0.001
No-object	2.56	1.61	3.80	2.96	5.84	4.37	$F_{(2,8)}$=23.36, p=<0.001

Figure 7-15. Mean results for all tasks under the size conditions along the axis of hand placement (^Greenhouse-Geisser correction for sphericity violation is made)

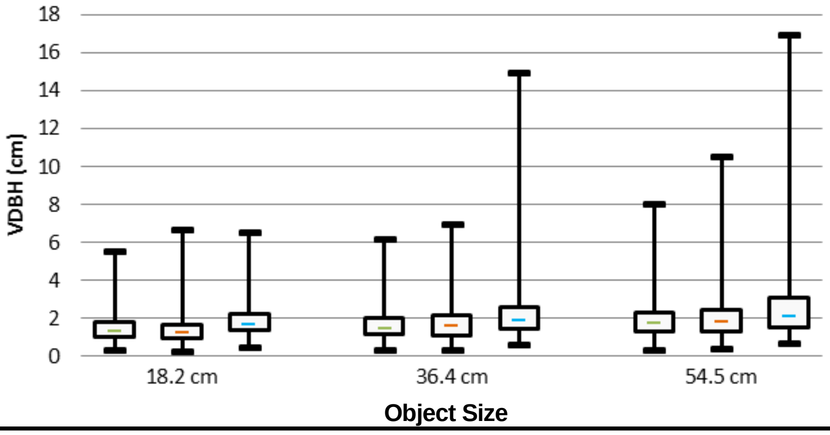
When measuring the amount of variability in the distance between the actor’s hands (VDBH) AlongAHP, the size of the virtual object had a statistically significant effect on performance for the Interactive (p=<0.001), Animated (p=<0.001) and no-object (p=<0.001) modalities. These results are presented in Figure 7-15 and show that when using larger virtual objects there was a larger ‘mean VDBH’ (henceforth referred to as VDBH throughout this section), meaning that the positioning of the actor’s hands became less stable.

Post-hoc analysis confirmed this trend. For the Interactive modality a statistically significant difference was again detected between the 18.2cm and 36.4cm levels (p=<0.001; difference=0.63cm), the 36.4cm and 54.5cm levels (p=<0.001; difference=0.77cm) and the 18.2cm and 54.5cm levels (p=<0.001; difference=1.4cm).

For the Animated modality no statistically significant difference was detected between the 18.2cm and 36.4cm levels (p=0.159), but one was between the 36.4cm and 54.5cm levels (p=<0.001; difference=0.57cm) and the 18.2cm and 54.5cm levels (p=<0.001; difference=0.91cm).

For the no-object modality no statistically significant effect was detected between the 18.2cm and 36.4cm (p=0.461), but one was detected between the 36.4cm and 54.5cm levels (p=0.011; difference=2.04cm) and the 18.2cm and 54.5cm levels (p=0.006; difference=3.28cm).

Perpendicular to the axis of hand placement



Size	18.2cm		36.4cm		54.5cm		
Modality	Mean	SD	Mean	SD	Mean	SD	ANOVA
Animated	1.51	0.85	1.67	0.96	1.92	1.20	$F_{(2,8)}$=10.67, p=0.002
Interactive	1.44	0.90	1.74	1.13	1.99	1.42	$F_{(2,8)}$=32.78, p=<0.001
No-object	1.85	1.03	2.27	1.99	2.52	2.08	$F_{(2,8)}$=1.81, p=0.205

Figure 7-16. Mean results for all tasks for the size conditions perpendicular to the axis of hand placement

The size of the virtual object had a statistically significant effect on the VDBH PerpendicularAHP for the Interactive (p=<0.001) and Animated modalities (p=0.002). No statistically significant result was observed for the no-object mode (p=0.205).

Post-hoc analysis revealed that for the Interactive modality a statistically significant effect was detected between the 18.2cm and 36.4cm levels (p=<0.001; difference=0.3cm), the 36.4cm and 54.5cm levels (p=0.016; difference=0.25cm) and the 18.2cm and 54.5cm levels (p<0.001; difference=0.55cm). For the Animated modality a statistically significant effect was detected between the 18.2cm and 36.4cm levels (p=0.016; difference=0.16cm), the 36.4cm and 54.5cm levels (p=<0.001; difference=0.25cm) and the 18.2cm and 54.5cm levels (p=<0.001; difference=0.41cm).

These results have demonstrated that larger virtual objects caused an increase in the amount of variability of the PerpendicularAHP between the actors' hands for the Animated and Interactive virtual modalities. The results are presented in Figure 7-16 (page 111).

7.5.1.3. Section Summary and Recommendations

Presentation of Data

Throughout sections 7.5.1.3, 7.5.2.3 and 7.5.3.3 the results of the mean and standard deviation for the MDOS and VDBH performance metrics are represented using tables alongside box plots of the data. A final summary for each condition is also presented at the end of each section, where the results from all the performance metrics for each modality are discussed together and compared. In this section the data is presented using plots that show the mean and standard deviation for both the MDOS and VDBH of a single level (with Figure 7-17 showing an annotated example). The data is presented in this manner to show both the mean distance of the hands from the surface of the object and the amount of variation in a way that allows quick visual comparison between the levels of the conditions and between the interaction modalities.

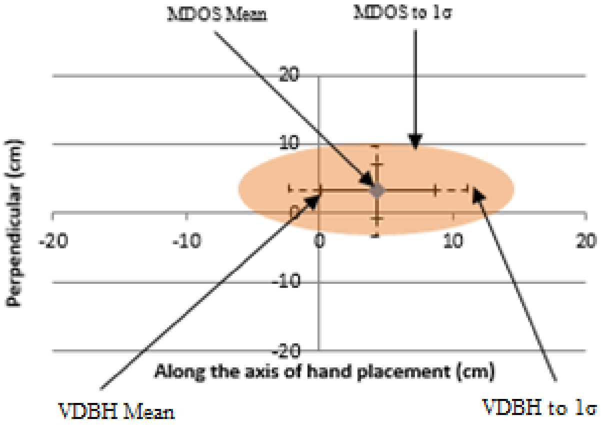


Figure 7-17. Annotated diagram of hand placement value graphs

In these graphs the measurements PerpendicularAHP are represented along the Y axis and measurements AlongAHP are presented along the X axis of the graph. The mean of the MDOS is represented by the diamond marker and the shaded oval represents MDOS to one standard deviation. The solid whiskers emanating from the diamond marker represent the mean VDBH either side of MDOS mean (which on the horizontal axis in Figure 7-17 would represent a mean VDBH of ~4cm), with the dashed whiskers showing VDBH to 1 standard deviation (which in Figure 7-17 would represent a standard deviation of ~2cm).

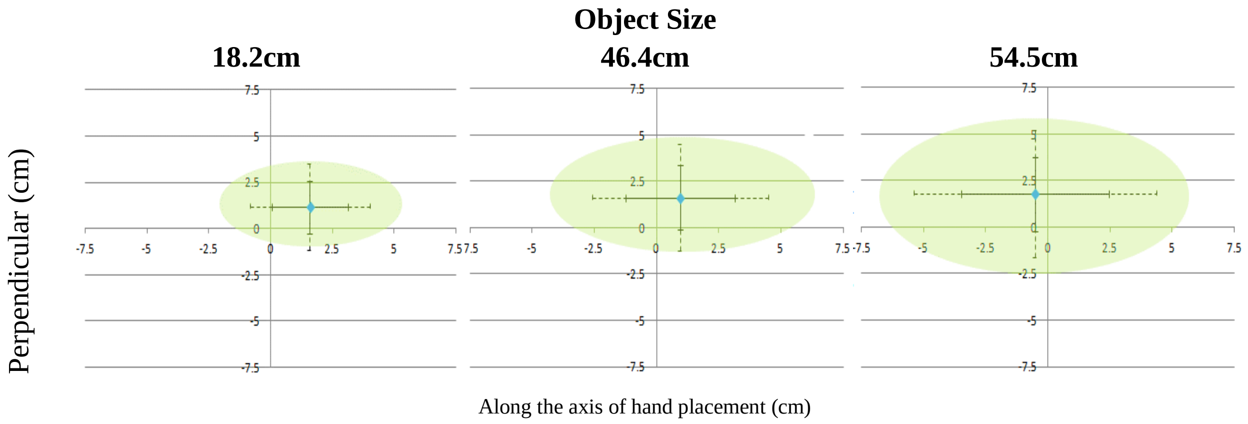


Figure 7-18. Combined results for each Interactive mode size condition. (a) 18.2cm, (b) 36.4cm and (c) 54.5cm sizes

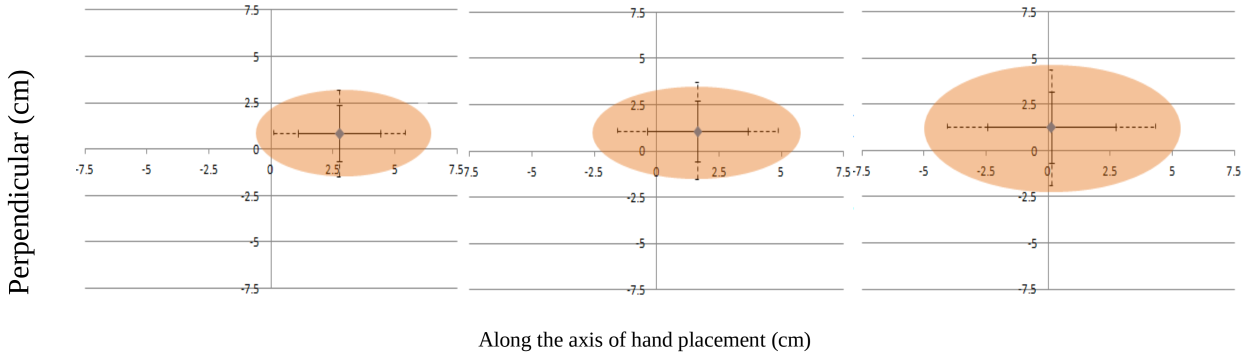


Figure 7-19. Combined results for each Animated mode size condition. (a) 18.2cm, (b) 36.4cm and (c) 54.5cm sizes

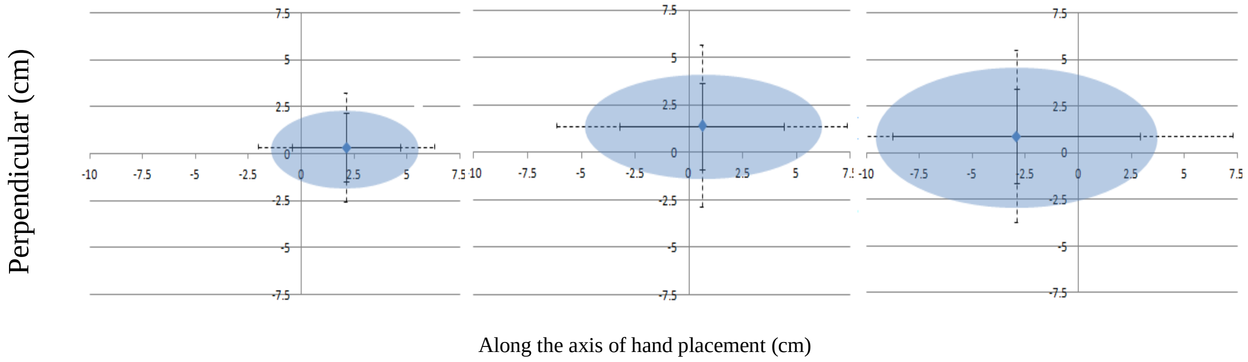


Figure 7-20. Combined results for each no-object mode size condition. (a) 18.2cm, (b) 36.4cm and (c) 54.5cm sizes

A considerable amount of evidence was present to show that the size of the virtual object had a significant impact on the actor's ability to perform a realistic interaction. The combined results are presented in Figure 7-18 for the Interactive mode, Figure 7-19 for the Animated mode and Figure 7-20 for the no-object mode.

Using larger virtual objects appeared to negatively impact the VDBH increasing both AlongAHP and PerpendicularAHP. However, a large object appeared to benefit the MDOS AlongAHP, which decreased when larger objects were used, which could be exploited to achieve a hand placement closer to the ideal of 0cm; although this was counteracted by an increasing standard deviation, indicating that an accurate hand placement is less likely (resulting in an increased range of potential MDOS errors).

It was observed that as a result of this reducing mean and increasing standard deviation that the overestimation to one standard deviation for each object size appeared to remain static at around ~+5.5cm (the exception being +3.83cm for the 54.5cm level in the no-object modality).

Conversely, the underestimation to one standard deviation ranged from -1.93cm to -6.74cm between the 18.2cm and 54.5cm levels for the Interactive modality, from -0.64cm to -5.03cm for the Animated modality and from -1.27cm to -9.67cm for no-object modality. Therefore with larger virtual objects the actor becomes more likely to underestimate the object's size than overestimate it.

Hypothesis Tests

The null hypothesis for the effect that object size has on the MDOS was:

"1a. The Mean Distance to Object Surface is not related to the size of the object."

As a statistically significant decrease in average MDOS was detected between the 36.4cm-54.5cm and 18.2cm-54.5cm levels AlongAHP, the hypothesis must be rejected and revised to:

"As the size of the virtual object increases the mean distance between the actor's handsdecreases along the axis of hand placement."

The null hypothesis for the effect that object size has on the VDBH was:

"1b. The amount of variation in the distance between the actor's hands is not related to the size of the object."

A statistically significant increase in the average VDBH was detected both AlongAHP and PerpendicularAHP as larger object sizes were used. Therefore, the null hypothesis is concluded to be incorrect and must be rejected. The revised hypothesis is:

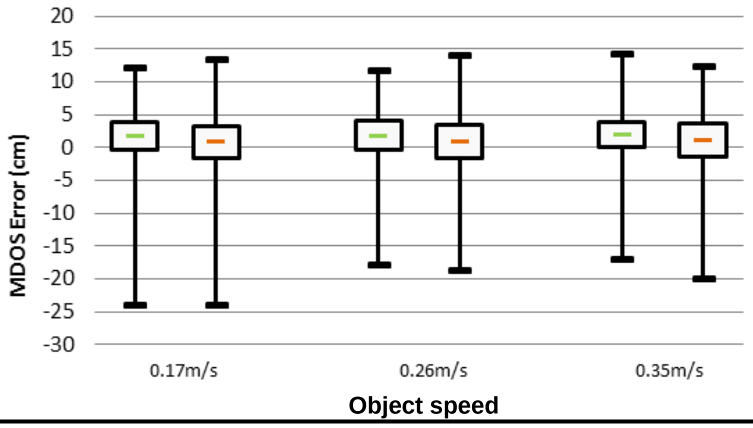
"The use of larger virtual objects results in a larger amount of variation between the hands of the actor and the relevant object surfaces both along and perpendicular to the axis of hand placement"

7.5.2. Speed of virtual object

This section presents the findings concerning the effect that the target speed of the interaction has on an actor's performance. Three levels are used: 0.17m/s, 0.26m/s and 0.35m/s. The speed condition was omitted for the no-object modality, so it will not be reported on.

7.5.2.1. Analysis of Distance between Hands

Along the axis of hand placement

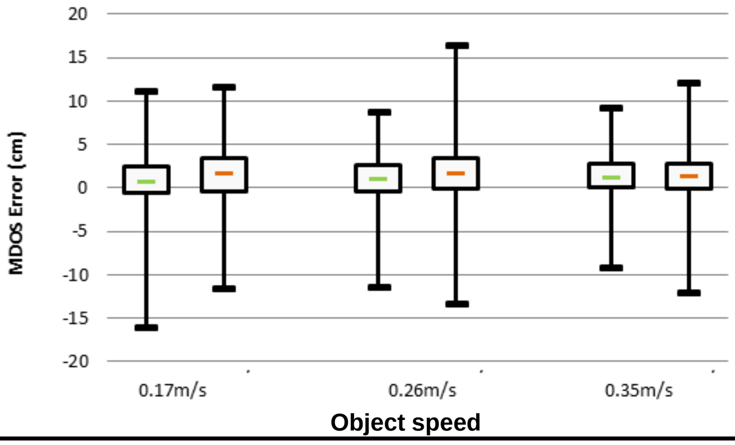


Speed	0.17m/s		0.26m/s		0.35m/s		
Modality	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**ANOVA**
Animated	1.28	4.51	1.56	4.56	1.66	4.09	$F_{(2,8)}$=2.344, p=0.125
Interactive	0.53	5.23	0.68	5.14	0.80	4.84	$F_{(2,8)}$=0.526, p=0.559

Figure 7-21. Mean results for all tasks under the speed conditions along the axis of hand placement

When measuring the MDOS AlongAHP for the Speed condition, no statistically significant effect was detected for either the Interactive modalities (p=0.599) or Animated (p=0.125). This means that Speed of the interaction does not affect the actor's ability to accurately estimate the size of the virtual object. These results are presented in Figure 7-21.

Perpendicular to the axis of hand placement



Speed	0.17m/s		0.26m/s		0.35m/s		
Modality	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**ANOVA**
Animated	0.82	3.07	1.06	2.72	1.35	2.30	$F_{(2,8)}$=4.047, p=0.038
Interactive	1.54	3.32	1.66	3.49	1.38	2.79	$F_{(2,8)}$=0.16, p=0.852

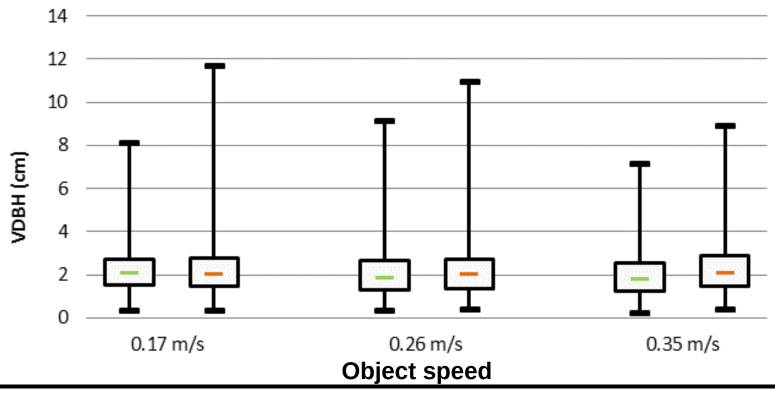
Figure 7-22. Mean results for all tasks under the speed conditions perpendicular to the axis of hand placement

When measuring the distance between the hands and the surface of the virtual object PerpendicularAHP for the Speed condition, no statistically significant effect was found for Interactive mode (p=0.852). However, a statistically significant effect (p=0.038) was detected for the Animated mode, which manifested itself as a small increase of 0.53cm between the slowest and fastest target speeds.

Post-hoc analysis performed on the Animated modality demonstrated that this effect was statistically significant between the 0.17m/s and 0.26m/s levels (p=<0.001; difference=0.24cm), the 0.26m/s and 0.35m/s levels (p=0.016; difference=0.29cm) and the 0.17m/s and 0.35m/s levels (p=<0.001, difference=0.53cm). These results are presented in Figure 7-22 (page 116).

7.5.2.2. Analysis of Variability in Distance Between Hands

Along the axis of hand placement



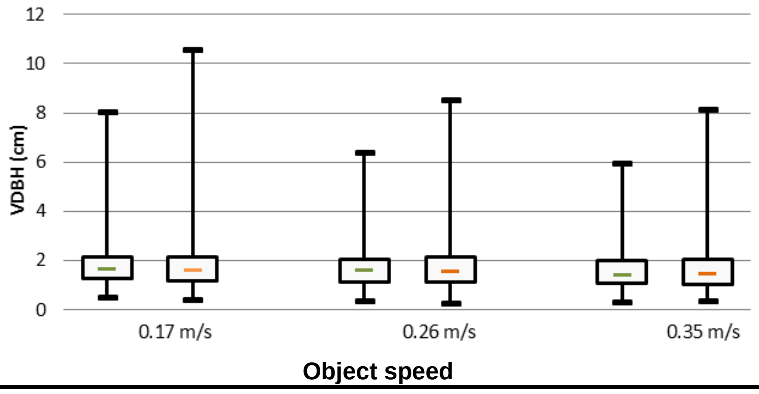
Speed	0.17m/s		0.26m/s		0.35m/s		
Modality	Mean	SD	Mean	SD	Mean	SD	ANOVA
Animated	2.20	1.29	2.11	1.40	1.97	1.32	$F_{(2,8)}$=2.051, p=0.155
Interactive	2.23	1.64	2.21	1.45	2.27	1.58	$F_{(2,8)}$=0.007, p=0.993

Figure 7-23. Mean results for all tasks under the speed condition along the axis of hand placement

The Speed of the interaction was shown to have no statistically significant impact on the VDBH AlongAHP for either the Interactive object modality (p=0.993) or the Animated object modality (p=0.155), indicating that target Speed does not affect the amount of variability in the distance between the actor's hands. These results are presented in Figure 7-23.

Perpendicular to the axis of hand placement

Speed was not shown to have a statistically significant effect on the VDBH PerpendicularAHP for the Animated modality (p=0.081), but it did have a statistically significant effect for the Interactive modality (p=0.043). These results are presented in Figure 7-24.



Speed	0.17m/s		0.26m/s		0.35m/s		
Modality	Mean	SD	Mean	SD	Mean	SD	ANOVA
Animated	1.80	1.11	1.69	1.02	1.60	0.92	$F_{(2,12)}$=3.113, p=0.081
Interactive	1.78	1.17	1.71	1.15	1.68	1.24	$F_{(2,14)}$=3.978, p=0.043

Figure 7-24. Mean results for all tasks under the speed conditions perpendicular to the axis of hand placement

This effect appeared to manifest itself in a decreasing amount of VDBH between the 0.17m/s and the 0.35m/s target speeds. However, the total amount of decrease between the 0.17m/s and 0.35m/s levels is negligible (0.1cm). Despite this small effect, post-hoc analysis found significant effects for the Interactive modality between the 0.17m/s and 0.26m/s levels (p=<0.001), the 0.26m/s and 0.35m/s levels (p=0.019) and the 0.17m/s and 0.35m/s levels (p<0.001).

7.5.2.3. Section summary and Recommendations

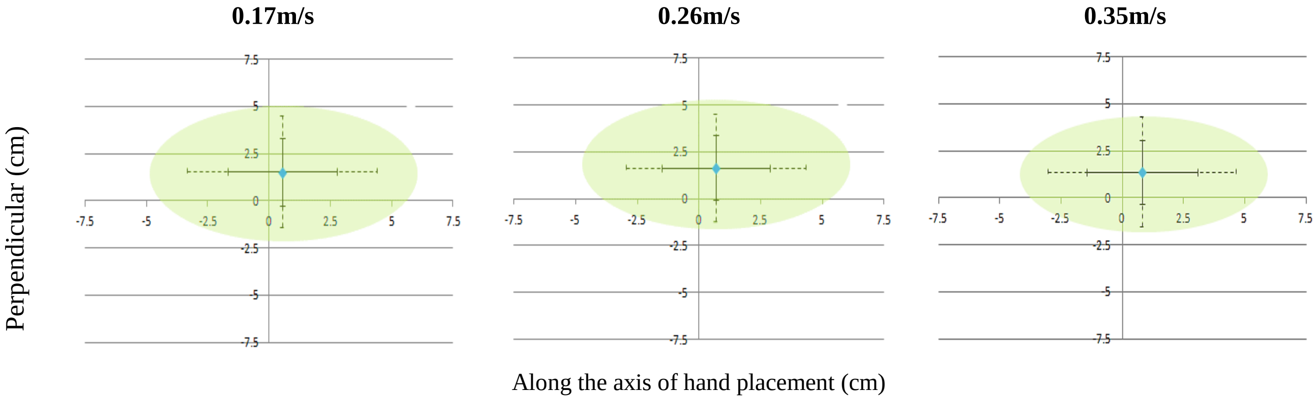


Figure 7-25. Results for each movement speed with the animated object. (a) 0.17m/s, (b) 0.26m/s and (c) 0.35m/s

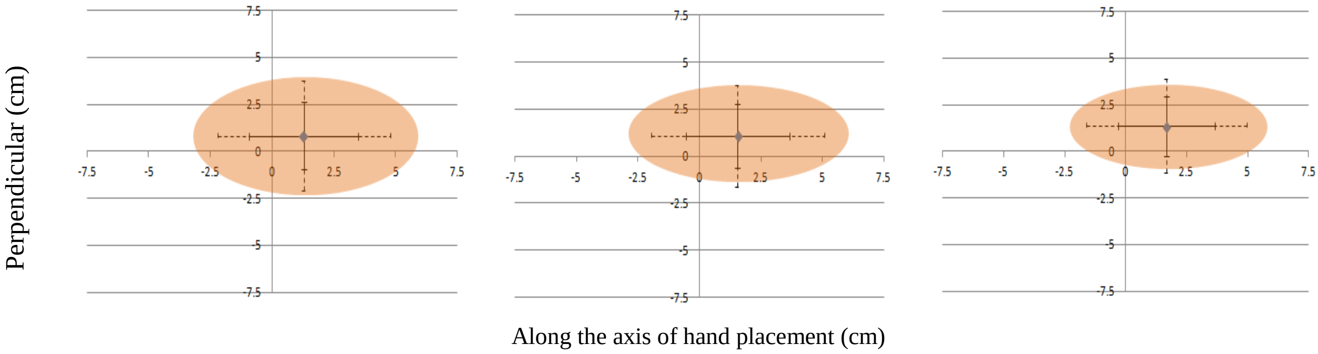


Figure 7-26. Results for each target speed with the interactive object. (a) 0.17m/s, (b) 0.26m/s and (c) 0.35m/s

Almost no evidence is present to show that the target speed has any considerable impact on the actor's ability to estimate the boundaries of the virtual object. Any statistically significant results that were observed resulted in an almost negligible impact. This held true across all modes. Therefore it is recommended that any target interaction speed (in the range tested) can be used, as it does not affect performance in any significant way.

Hypothesis tests

The null hypothesis for the effect that target interaction speed has on the MDOS was:

> "*2a. The Mean Distance to Object Surface is not related to the speed of the object.*"

As no statistically significant result was detected the null hypothesis remains true. The null hypothesis for the effect that target interaction speed has on the VDBH was:

> "*2b. The amount of variation in the distance between the actor's hands is not related to the speed of the object*

A statistically significant effect was detected for the MDOS PerpendicularAHP, although the actual effect was negligible and was likely due to a Type I[9] error. Therefore, the hypothesis was not revised.

7.5.3. Hand Placement Relative to Axis of Motion (Hand Position)

This section presents the findings on how Hand Position affects the actor's performance. To recap, 'Hand Position' is a 'derived condition' that is analysed in the ANOVA by looking at the statistical interaction between the Axis of Motion and Axis of Hand Placement conditions. The two Axis of Motion levels analysed are along the "Horizontal" and the "Vertical" axes. The Hand Placement levels used are described as "Left/Right" and "Top/Bottom".

The notation of this Hand Position condition is as follows. The two Axis of Motion conditions were a Horizontal path of interaction (H) and a Vertical path of interaction (V), with the Axis of Hand Placement conditions described as Left/Right (LR) and Top/Bottom (TB). For example, with this notation a horizontal interaction path with a hand placement on the top and bottom is represented by "H-TB".

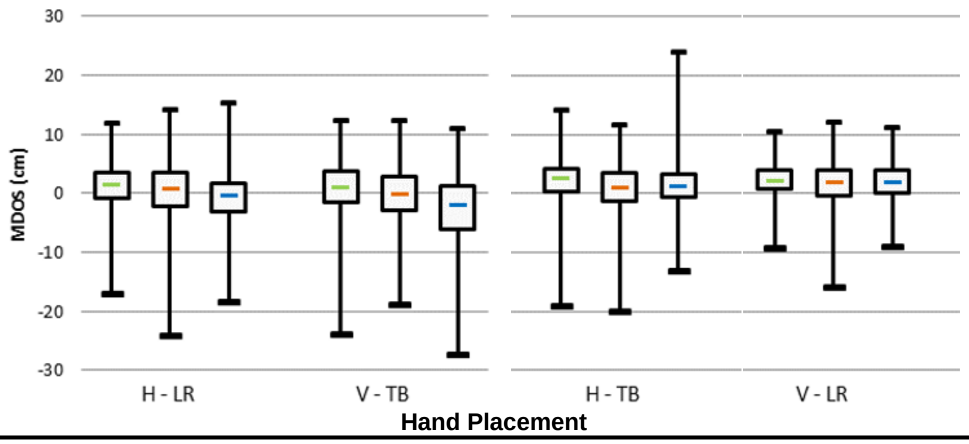
In cases where the Axis of motion and the Axis of Hand Placement and the Axis of Motion directions match (e.g. both are horizontal), it is considered a *with* hand position (LR = Horizontal, TB = Vertical). If they do not match (i.e. one is vertical and the other horizontal), it is considered an *against* hand position. Using these definitions H-LR and V-TB are considered *with* hand positions, and H-TB and V-LR *against*. Refer back to page 101 for a descriptions and examples.

[9] A type of error occasionally present in statistical analysis, where a statistically significant effect is detected - even though it is not present.

7.5.3.1. Analysis of Distance between Hands

Along the axis of hand placement

A statistically significant effect was detected with Hand Position AlongAHP for the Interactive (p=0.039), Animated (p=0.006) and no-object modalities (p=0.002). Each modality appeared to present two common features. These results are presented in Figure 7-27.



Hand	With				Against				
Position	H-LR		V-TB		H-TB		V-LR		
Modality	Mean	SD	Mean	SD	Mean	SD	Mean	SD	ANOVA
Animated	1.11	4.55	0.85	5.24	1.97	4.28	2.09	3.08	$F_{(1,4)}$=12.97 p =0.006
Interactive	0.61	5.40	-0.28	5.31	0.51	5.07	1.83	4.20	$F_{(1,4)}$=5.61 p=0.039
No-object	-0.47	5.73	-2.85	7.29	1.25	4.64	1.93	3.72	$F_{(1,4)}$=17.53 p=0.002

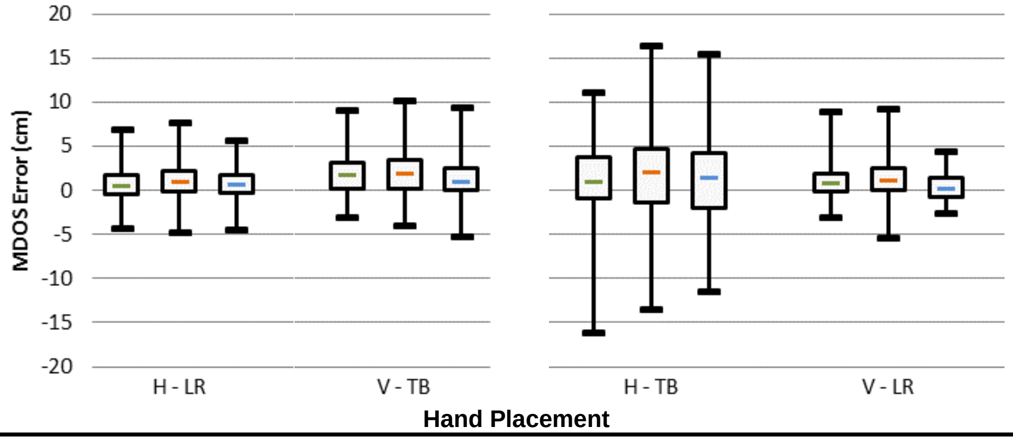
Figure 7-27. MDOS results for the Hand Position conditions along the axis of hand placement

The first common feature was that for each mode the V-LR level held the largest MDOs, ranging from 1.83cm (Interactive) to 2.09cm (Animated). For the Interactive modality post-hoc analysis confirmed that V –LR was significantly larger than all other Hand Positions (H-LR-V-LR: p=0.01 (difference=1.22cm), H-TB-V-LR: p=0.018 (difference=1.32cm), V-TB-V-LR: p=0.019 (difference=2.11cm)). For the Animated modality there was only a statistically significant difference between the V-TB and V-LR levels (p=0.008, difference=1.24cm) and for the no-object modality there was only a statistically significant difference between the V-TB and V-LR levels (p=0.024, difference=4.78cm) and the H-TB and V-LR levels (p=0.029, difference=0.68cm). Thus, for each modality the MDOS of the V-LR level was statistically significantly larger than the V-TB level and was frequently statistically larger than the other Hand Positions.

The second common feature was that the V-TB hand position held the lowest MDOS for all three modalities, and for the Animated and Interactive modalities it also yielded the result closest to the ideal of 0cm for the Animated and Interactive modalities (-0.28cm and +0.85cm respectively). For the no-object mode the V-TB level resulted in a large underestimation of -2.85cm accompanied by a considerable standard deviation of 7.29cm.

However post-hoc analysis could only confirm that this low MDOS for V-TB was statistically significant from that for the other hand positions for each object in a few cases. A statistically significant difference was found between the V-LR and V-TB hand positions for the Interactive (p=0.019), Animated (p=0.008) and no-object modalities (p=0.024); in each case the V-TB condition was lower by 2.11cm, 1.24cm and 4.78cm respectively. A statistically significant effect was detected between the V-TB and the H-TB hand placements (p=0.005) for the Animated modality, with a total difference of 1.12cm (H-TB scoring an MDOS of 1.25cm). There were no other statistically significant effects detected.

Perpendicular to the axis of hand placement



Hand	With				Against				
Position	H-LR		V-TB		H-TB		V-LR		
Modality	Mean	SD	Mean	SD	Mean	SD	Mean	SD	ANOVA
Animated	0.64	1.72	1.72	2.24	0.98	4.22	0.97	1.76	$F_{(1,8)}$=0.63 p=0.451
Interactive	1.09	2.00	1.92	2.55	1.83	5.09	1.28	2.12	$F_{(1,11)}$=0.48 p=0.831
No-object	0.85	1.88	1.27	2.14	1.12	4.90	0.32	1.53	$F_{(1,6)}$=1.84 p=0.223

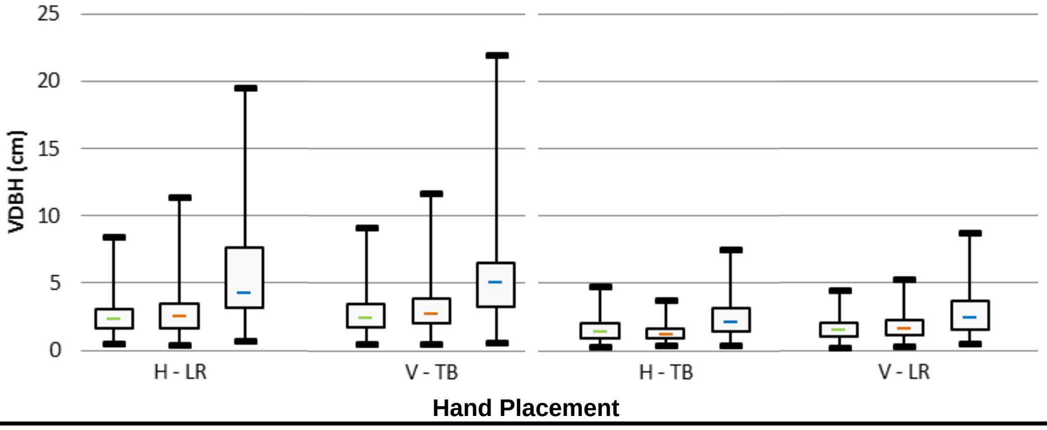
Figure 7-28. MDOS results for the Hand Position conditions perpendicular to the axis of hand placement

For MDOS no statistically significant effect could be detected for Hand Position PerpendicularAHP for either the Interactive (p=0.831), Animated (p=0.451) or no-object modalities (p=0.223). However, it was noted that for each modality the H-TB level yielded an abnormally large standard deviation compared to the other hand positions, ranging from 4.22cm (Animated) to 5.09cm (Interactive). As the ANOVA only informs on the significance of the difference between means and not on the standard deviation, it is unknown whether this effect was significant or not. However, the fact that the same feature appeared with all three modalities strongly suggests that this was not a coincidence.

As such, Hand Position probably does have an effect on MDOS PerpendicularAHP, in that a H-TB hand position would be more likely to lead to an MDOS error. As previously stated in section 7.5.1, MDOS errors PerpendicularAHP are consistent with real life interactions, so this would not be an issue when attempting to create a realistic interaction. Because this effect cannot be confirmed statistically it cannot be included in the revised hypothesis. These results are presented in Figure 7-28.

7.5.3.2. Analysis of Variability in Distance Between Hands

Along the axis of hand placement



Hand	With				Against				
Position	**H-LR**		**V-TB**		**H-TB**		**V-LR**		
Modality	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**ANOVA**
Animated	2.52	1.41	2.72	1.64	1.53	0.83	1.60	0.85	$F_{(1,4)}$=42.91 p =<0.001
Interactive	2.75	1.76	3.06	1.84	1.36	0.67	1.77	0.93	$F_{(1,4)}$=52.39 p =<0.001
No-object	5.51	3.92	5.53	4.21	2.37	1.64	2.80	1.80	$F_{(1,4)}$=35.14 p =<0.001

Figure 7-29. VDBH results for the Hand Position conditions Along the axis of hand placement

Hand Position was shown to have a statistically significant effect on the amount of variation in the distance between the actor's hands AlongAHP for the Interactive (p=<0.001), Animated (p=<0.001) and no-object (p=<0.001) modalities.

For all modes, when the hands were placed *with* the axis of motion (H-LR, V-TB) the amount of variation in the distance between them was considerably larger than when they were placed *against* the axis of motion (H-TB, V-LR). These effects were confirmed with post hoc analysis. When comparing the *with* hand positions to the *against* hand positions it was observed that the differences between them were statistically significant for all modalities. These results are presented in Figure 7-29.

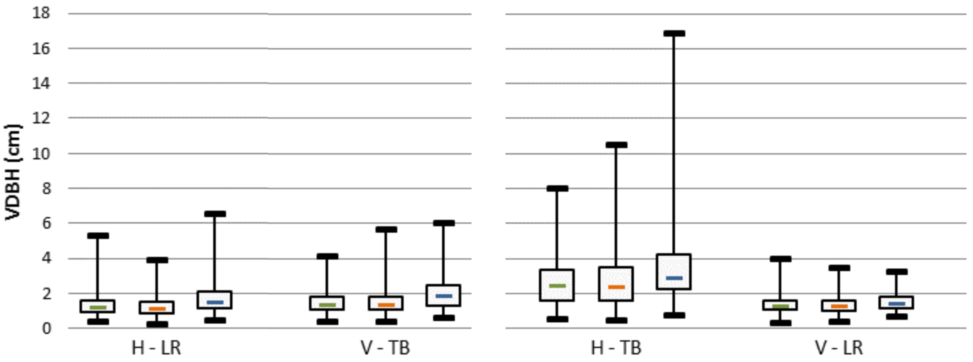
A statistically significant difference was detected between H-LR (a *with* hand position) and both of the *against* hand positions for each modality. Between the H-LR and H-TB hand placements the effects were significant for the Interactive: (p=<0.001, difference=-1.39cm) and Animated (p=<0.001, difference=-0.99cm) and the no-object modalities (p=<0.001, difference=-3.14cm). Between the H-LR and V-LR hand positions the effects were also significant for the Interactive (p=<0.001, difference= -0.98cm), Animated (p=<0.001, difference=-0.92cm) and no-object modalities (p=0.05, difference= -2.71cm). In each case H-LR resulted in a larger VDBH than either of the against hand positions.

Similarly, statistically significant effects were discovered when comparing V-TB (the other *with* hand position) to the *against* hand positions for each modality. A statistically significant effect was present between

V-TB and H-TB (*against*) hand placements for the Interactive (p=<0.001, difference=-1.7cm), Animated (p=<0.001, difference=-1.19cm) and no-object modalities (p=0.025, difference=-3.16cm). Similarly, a statistically significant effect was present between V-TB and the V-LR (*against*) hand positions for the Interactive (p=<0.001, difference=-1.29cm) and no-object modalities (p=0.015, difference=-2.73cm), although not for the Animated modality (p=0.989). From this it is possible to conclude that the *with* hand positions lead to a much larger amount of VDBH than the *against* hand positions.

No statistically significant effect was detected between the two *with* levels (H-LR and V-TB) for either modality (Interactive: p=0.351, Animated: p=0.388, no-object: p=0.325). Conversely, A statistically significant difference was detected between the *against* hand positions (H-TB and V-LR) for the Interactive modality (p=0.009), which presented itself as an increase in variation of 0.41cm for the V-LR hand position. This was also true for the Animated modality (p=<0.001) where an increase of 0.07cm was present, though negligible. However, no statistically significant result was detected between the H-TB and V-LR hand placements for the no-object modality (p=0.984).

Perpendicular to the axis of hand placement



Hand	**With**				**Against**				
Position	**H-LR**		**V-TB**		**H-TB**		**V-LR**		
Modality	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**ANOVA**
Animated	1.30	0.61	1.48	0.65	2.61	1.40	1.39	0.55	$F_{(1,6)}$=1.51 p=0.008
Interactive	1.26	0.67	1.50	0.66	2.75	1.72	1.36	0.58	$F_{(1,7)}$=32.5 p=0.001
No-object	1.70	0.91	1.99	1.04	3.65	2.82	1.57	0.59	$F_{(1,6)}$=13.61 p=0.01

Figure 7-30. VDBH results for the Hand Position conditions Perpendicular to the axis of hand placement

For the VDBH in hand positions PerpendicularAHP a statistically significant effect was detected for the Animated (p=0.008), Interactive (p=0.001) and no-object (p=0.01) modalities. For each mode, this manifested itself in a considerably larger amount of variation in the H-TB hand position than the other three. The results are found in Figure 7-30.

This effect was confirmed in post-hoc analysis for both the Animated and Interactive modalities, where the statistically significant difference was detected between the H-TB and the other three hand positions, each time yielding a larger VDBH:

Interactive: H-TB-H-LR p=<0.001 (diff.=1.49cm); H-TB-V-LR p=<0.001 (diff.=1.39cm);
H-TB-V-TB p=<0.001 (diff.=1.25cm)
Animated: H-TB-H-LR p=<0.001 (diff.=1.31cm); H-TB-V-LR p=<0.001 (diff.=1.22cm);
H-TB-V-TB p=<0.001 (diff.=1.13cm)

However, this was found to not be statistically significant for the no-object modality as no significant effect could be found between H-TB and the other three hand placements; despite the fact that H-TB is larger by 1.66cm (compared to V-TB) to 2.08cm (compared to V-LR) and that a statistically significant effect was detected in the ANOVA. This is likely due to a type II error[10] in the post-hoc analysis, caused by the considerably larger standard deviation (2.82cm) of the data for the H-TB hand position. This error can be detected by calculating a Beta error value, which describes the probability of a type II error occurring, where a value of >0.3 means that an error is likely to be present. Compared to the H-TB level, H-LR had an alpha of 0.8686, V-LR one of 0.9032 and V-TB one of 0.7611, meaning that in each case a Type II error was likely present and that a statistically significant effect was in fact present between these levels for the no-object modality also.

7.5.3.3. Section summary and Recommendations

A significant amount of evidence was present to show that Hand Position has an impact on the actor’s ability to perform an accurate interaction. The combined results of MDOS and VDBH are presented in Figure 7-31 for the Animated mode, Figure 7-32 for the Interactive mode and Figure 7-33 for the no-object mode.

The key result for Hand Position AlongAHP was that the *With* Hand Positions (H-LR, V-TB) had an extremely negative effect on the performance of the actors, leading to a much greater VDBH. This observation held true across each mode.

It is observed that the MDOS overestimation is approximately +5.5cm for each level. However, the underestimation result for V-LR is -2.37cm, which is significantly closer to 0px than the other levels, which range from -4.56cm for H-TB to -5.59cm for V-TB. Therefore the actor is far less likely to underestimate the size of the object for the V-LR level.

[10] A type of error occasionally present in statistical analysis, where a statistically significant effect is not detected - even though it is present.

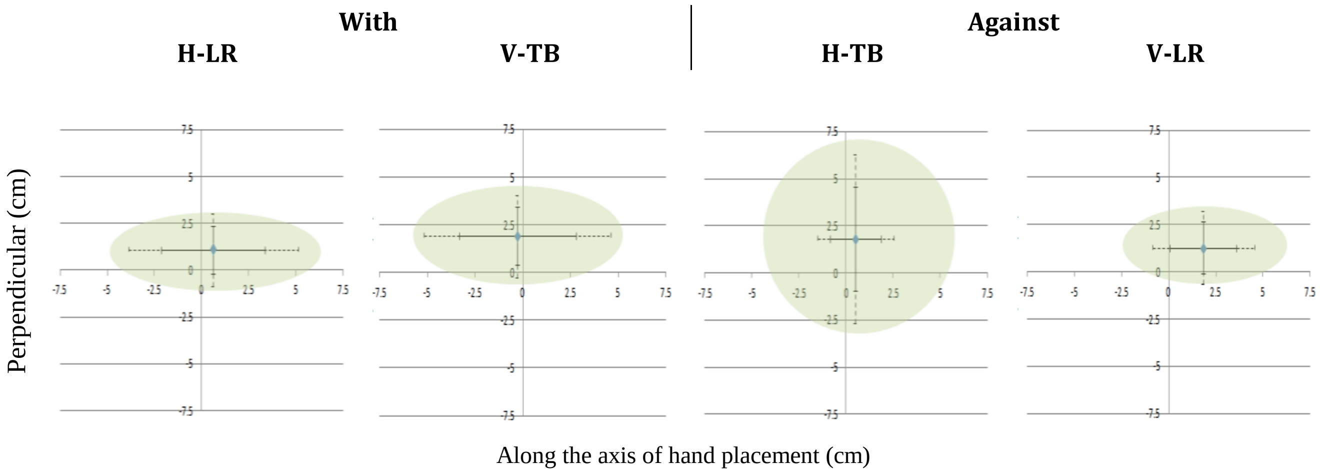


Figure 7-31. Combined results for each Hand Position for the animated mode

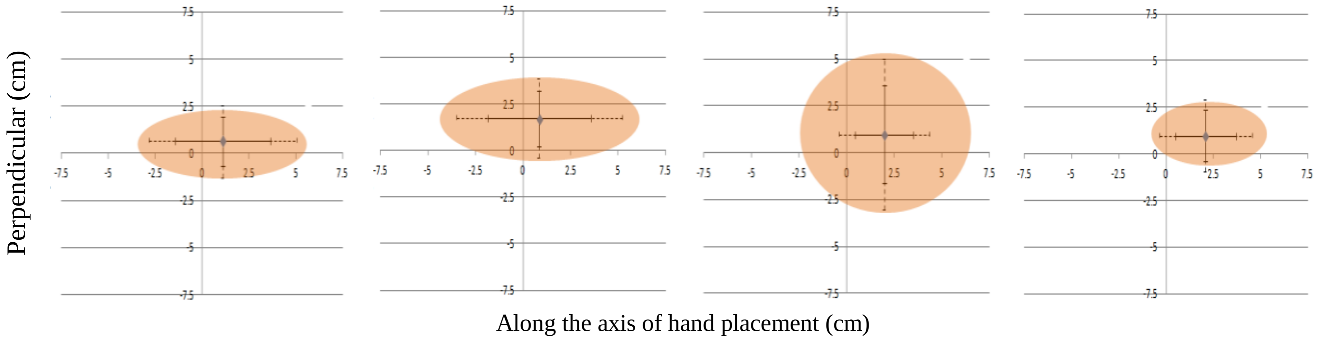


Figure 7-32. Combined results for each Hand Position for the interactive mode

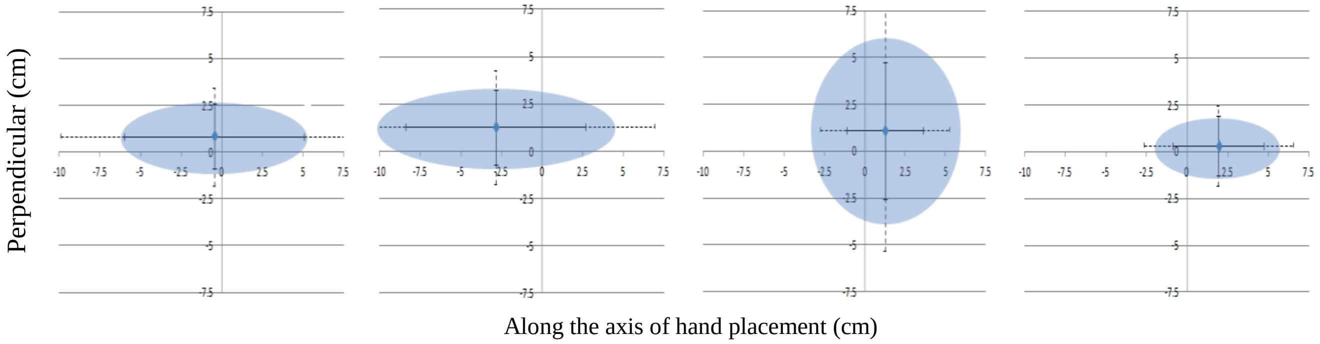


Figure 7-33. Combined results for each Hand Position for the no-object mode.

It was observed that the H-TB *Against* level had an extremely negative impact on the performance of the actors for VDBH PerpendicularAHP. In addition, the H-TB Hand position resulted in a large standard deviation for MDOS. This observation held true across each modality, in Figure 7-32 to Figure 7-33.

Taking these factors into account, it is recommended that interactions that conform to the *With* Hand Positions (H-LR, V-TB) be avoided due to the negative effects they cause AlongAHP. Similarly the H-TB method should also be avoided where possible due to the large VDBH experienced PerpendicularAHP.

This leaves the most reliable hand position as V-LR, which as shown in Figure 7-32 and Figure 7-33 yields the lowest amount of Variability in Distance Between Hands and a small region of MDOS error (to one standard deviation). In practice this Hand Position is not always possible as it only allows the actor to move an object along the vertical axis.

If the actor is to move an object along the horizontal axis, the H-TB level is preferable despite the aforementioned issues. This is because any variation in the distance between the hands PerpendicularAHP will produce smaller gaps than the same variation AlongAHP, which would be the case with the H-LR level.

Hypothesis Tests

The null hypothesis was:

"3a. The Mean Distance to Object Surface is not related to the actor's hand position."

As a statistically significant result was detected for the Interactive modality along the axis of hand placement, where the V-LR Hand Position led to a significantly larger average MDOS than the other Hand Positions, the null hypothesis must be rejected in favour of a new hypothesis:

"*The mean distance between the actor's hands to object surface is related to the Hand position. Along the axis of hand placement the V-LR Hand Position results in the largest Mean Distance to Object Surface and V-TB results in the smallest."*

The null hypothesis concerning the effect that hand placement relative to the axis of motion has on the VDBH was:

"3b. The amount of variation between the actor's hands is not related to the actor's hand position"

Statistical analysis confirmed that for the Interactive modality Hand Position had a statistically significant effect on the amount of variation between the actor's hands both AlongAHP (where the average VDBH was larger for the *with* positions) and PerpendicularAHP (where the H-TB level leads to a higher VDBH). As such, the null hypothesis must be rejected and revised to:

"*The amount of Variability in Distance Between the actor's Hands is related to the actor's Hand Position. Along the Axis of Hand Placement this results in a significantly larger amount of variation when the actor's hands are in the 'With' positions (i.e. H-LR &V-TB). Perpendicular to the Axis of Hand Placement this leads to a larger amount of variation in the H-TB Hand Position."*

7.5.4. Other Main Effects

This section briefly describes the main effects for each of the remaining conditions. As the comparison is only between conditions that contain two levels, post-hoc analysis is not required for this section.

7.5.4.1. Axis of Motion

Analysis of Distance between Hands. For the MDOS AlongAHP for the Axis of Motion condition, no statistically significant effect was detected for the Interactive modality ($F_{(1, 10)}$=0.005, p=0.944), the Animated modality ($F_{(1,9)}$=0.275, p=0.613), or the no-object modality ($F_{(1,10)}$=0.342, p=0.571).

For the MDOS PerpendicularAHP no statistically significant effect was observed for either the Interactive modality ($F_{(1,11)}$=1.015, p=0.335) or the no-object modality($F_{(1,6)}$=0.495, p=0.508). A statistically significant effect was detected for the Animated modality ($F_{(1,8)}$=5.520, p=0.047), which manifested itself as a slight increase on the vertical axis of 0.54cm (Vertical=1.35cm, Horizontal=0.81cm).

Analysis of Variability between Hands. For the VDBH AlongAHP no statistically significant effect was observed for either the Animated modality ($F_{(1,10)}$=1.407, p=0.263) or the no-object modality ($F_{(1,11)}$=0.028, p=0.871). A statistically significant effect was observed was observed with the Interactive modality ($F_{(1,11)}$=6.407, p=0.028), which manifested itself as a slight increase of 0.35cm in variability between hands for the Vertical axis (Vertical=2.41cm, Horizontal=2.06cm).

For the VDBH PerpendicularAHP, statistically significant effects were detected for both the Interactive ($F_{(1,7)}$=25.148, p=0.002) and Animated modalities ($F_{(1 ,6)}$=10.391, p=0.018). In both cases, the amount of variability between the actor's hands was slightly larger on the Horizontal axis, with a difference of 0.58cm for the Interactive modality (Vertical=1.403cm, Horizontal=2.01cm) and 0.53cm for the Animated modality (Vertical=1.434cm, Horizontal=1.96cm). No statistically significant result was detected for the no-object modality ($F_{(1, 6)}$=4.447, p=0.080).

7.5.4.2. Axis of Hand Placement

Analysis of Distance between Hands. For the MDOS AlongAHP for the Axis of Hand Placement, no statistically significant effect was detected for either the Interactive modality ($F_{(1, 10)=}$3.178, p =0.105) or the Animated modality ($F_{(1,9)}$ =1.737, p=0.220). A statistically significant result was detected for the no-object modality ($F_{(1,10)}$ =6.336, p =0.031), which manifested itself as an increase of 1.57cm distance of hands from the object surface for the LR (Left/Right) hand placement (LR=0.71cm, TB=-0.86cm).

For the MDOS PerpendicularAHP no statistically significant effect was detected for either the Interactive ($F_{(1,11)}$=2.545, p=0.139), Animated ($F_{(1,8)}$=0.539, p=0.484) or no-object modalities ($F_{(1,6)}$=0.159, p=0.704).

Analysis of Variability between Hands. For the VDBH AlongAHP no statistically significant effect was observed for either the Interactive ($F_{(1,11)}$=0.172, p=0.686), Animated ($F_{(1, 10)}$=0.001, p=0.974) or no-object modalities ($F_{(1,11)}$=0.252, p=0.626).

For the VDBH PerpendicularAHP a statistically significant effect was observed for the Interactive modality ($F_{(1,7)}$ =58.561, p=<0.001), the Animated modality ($F_{(1 ,6)}$=63.878, p=<0.001) and the no-object modality ($F_{(1, 6)}$=57.912, p=<0.001). In each case this manifested itself as a slight increase in VDBH when in the TB (Top/Bottom) arrangement, with a difference of 0.82cm for the Interactive modality (TB=2.13cm, LR=1.31cm), 0.71cm for the Animated modality (TB=2.05cm, LR=1.34cm) and 1.18cm for the no-object modality (TB=2.81cm, LR=1.63cm).

7.5.4.3. Direction

Analysis of Distance between Hands. For the MDOS AlongAHP for the Direction condition no statistically significant effect was detected for either the Interactive ($F_{(1,10)}$=0.107, p =0.751), Animated ($F_{(1,9)}$ =<0.001, p=0.991) or no-object modalities($F_{(1,10)}$ =0.067, p =0.800).

For the MDOS PerpendicularAHP no statistically significant effect was detected for either the Interactive ($F_{(1,11)}$=1.330, p=0.273) or Animated modalities ($F_{(1,8)}$=<0.001, p=0.986). A statistically significant result was detected for the no-object modality ($F_{(1,6)}$=16.195, p=0.007), which manifested itself as an increase of 0.42cm for the Left/Down level (Left/ Down=1.1cm, Right/Up=0.68cm).

Analysis of Variability between Hands. For the VDBH AlongAHP no statistically significant effect was detected for either the Interactive ($F_{(1,11)}$=1.727, p=0.216), Animated ($F_{(1,10)}$=0.607, p=0.454), or no-object modalities ($F_{(1,11)}$=0.066, p=0.802).

For the VDBH PerpendicularAHP, no statistically significant effect was observed for either the Animated modality ($F_{(1 ,6)}$=0.280, p=0.616) or the no-object modality($F_{(1, 6)}$=0 .265, p=0.625). A statistically significant effect was detected for the Interactive modality ($F_{(1,7)}$ =7.961, p=0.026), which manifested itself as an increase of 0.22cm for the Left/Down level (Left/Down=1.83cm, Right or Up=1.61cm).

7.6. Discussion and Conclusion

This chapter presented a taxonomy for measuring the problems of actor motion in MR interaction systems, serving as the first part of the framework to assess the plausibility of interactions. The tests applied within this work assess the motion of test subjects when completing a controlled series of interactive tasks and define two measurements for interaction accuracy, namely Mean Distance to Object Surface (MDOS) and the Variation in Distance Between Hands (VDBH). These measures were shown to be successful in determining the magnitude of common errors with interaction in the interactive virtual studio and are used to test a set of hypotheses. Using the results obtained from these measures we discuss and answer the three research questions presented at the start of this chapter.

Question 1. What is the Mean Distance to Object Surface (MDOS) and the amount of Variability in Distance Between Hands (VDBH) the actor can achieve when moving a virtual object using the Interactive modality?

The results for interaction are presented throughout the comprehensive review of actor motion discussed in section 7.5. A summary of the hypotheses tested and revised is presented:

1. Actor performance is not affected by the size of the object
   a. *As the size of the virtual object increases the mean distance between the actor's hands decreases along the axis of hand placement.*
   b. *The use of larger virtual objects results in a larger amount of variation between the hands of the actor both along and perpendicular to the axis of hand placement.*

2. Actor performance is not affected by the speed of the object
   a. *The Mean Distance to Object Surface is not related to the speed of the object.*
   b. *The amount of variation between the actor's hands is not related to the speed of the object.*

3. Actor performance is not affected by the placement of the hands with regards to the motion of the object
   a. *The mean distance between the actor's hands to object surface is related to the Hand position. Along the axis of hand placement the V-LR Hand Position results in the largest Mean Distance to Object Surface and V-TB results in the smallest."*
   b. *The amount of Variability in Distance Between the actor's Hands is related to the actor's Hand Position. Along the Axis of Hand Placement this results in a significantly larger amount of variation when the actor's hands are in the 'With' positions (i.e. H-LR &V-TB). Perpendicular to the Axis of Hand Placement this leads to a larger amount of variation in the H-TB Hand Position.*

From the results of the experiment the following key trends were identified and presented for the Interaction modality. It was also noted that in many cases these same trends were observed for the Animated and no-object modalities too, which suggested that these are real effects.

1. Actors are more likely to underestimate the size of large virtual objects.
2. Larger virtual objects result in a larger amount of variability between the actor's hands.
3. The target speed of the virtual object does not affect actor performance.
4. The amount of variability between the actor's hands and their estimation of the object is considerably larger for the *with* Hand Positions (H-LR, V-TB).
5. The amount of variability between the actor's hands PerpendicularAHP is considerably larger for the H-TB Hand Position.

From these trends, the following recommendations were made:

1. To reduce the amount of Variability in Distance Between Hands (VDBH) a smaller virtual object should be preferred,
2. To reduce the amount of Variability in Distance Between Hands (VDBH) with a V-LR Hand Position for interaction along a vertical motion path and H-TB for interaction along a horizontal motion path (both are the *against* Hand Positions).

Recommendations on how to improve performance in estimating the size of the object (MDOS) using data found from the Size conditions will not be made at this stage, as further important evidence will be presented during the analysis of the viewer perception study in chapter 8, where the viewer perception of errors that are measured for the object size conditions are explored. This evidence from the perceptual study changes the nature of what would be recommended if only the errors measured in this chapter were considered, ultimately validating the use of this framework.

Question 2. Are the MDOS and VDBH of the Interaction modality comparable to those of the Animated modality?

This section addresses whether the Interactive modality is preferable to the Animated modality. As discussed earlier in this thesis, one possible method for creating interaction in a conventional virtual studio would involve a virtual object travelling along a pre-defined path, usually triggered by a timed event or a signal by the actor (Gibbs & Baudisch, 1996). In the scenario of direct bimanual interaction using this method, to create the illusion the actor would move their hands along the same path as the virtual object, trying to match them to the surfaces of the object as closely as possible.

For the Animated modality the actor was provided with information on the speed of the object and countdown to the moment that the object would start to move along its path. This procedure approximately matches the experience of an actor in a live studio production, provided they are aware of the virtual object speed from rehearsal or on-screen feedback and that the animation is only triggered when they either expect it (for example, provide a signal to an operator).

Because the motion of the actor is unlikely to match the motion of the object perfectly, this method of interaction can result in a "lag" or "lead" between the mid-point of the actor's hand and the centroid of the virtual object. In these cases the actor either underestimates the speed of the object (lag) or overestimates (lead) the speed of the virtual object. Hereafter, this misestimation of the object's speed is referred to as the 'lag/lead error'. The Interactive modality has no lag/lead error associated with it by definition, as the centroid of the virtual object is matched to the mid-point of the actor's hands.

The magnitude of the lag/lead error for the animated conditions in the experiment are measured, followed by a discussion of whether it is a viable approach towards creating a convincing interaction illusion when compared to the Interaction modality.

If the Animated modality has a significant lag/lead associated with it and the performance of the actor in terms of MDOS and VDBH is similar or worse than with the Interactive modality, then the recommendation can be made that an Interactive virtual object is a superior method of creating the appearance of a bimanual interaction than the animated method.

Methodology. For each actor in each task the lag/lead error was calculated along the axis of motion using Equation 7-5, where E is the lag/lead error, N is the number of frames of captured data, O the location of the virtual object centroid, and L and R represent the location of the Left and Right hands respectively. Essentially this equation compares the centroid of the object to the midpoint of the actor’s hands measured along the Axis of Motion.

$$E = \frac{1}{N}\sum_{i=1}^{N} O_i - \big((L_i + R_i)/2\big)$$

Equation 7-5. Calculation for the lag/lead error for each actor in each task

When analysing the average lag/lead error for each condition the arithmetic mean could not be used as the lag (negative) and lead (positive) values cancel each other out, providing a result that is not representative of the error's true value. Therefore Root Mean Square (RMS) will be used to assess the average performance. Using RMS means that it is not possible to interpret whether the average result is a lag or a lead, but it will give a more accurate interpretation of the average magnitude of the error.

The RMS of the lag/lead was calculated using Equation 7-6, where N represents the number of lag/lead measurements, E the mean lag/lead error, j the actor and k the tasks associated with the desired condition (e.g. the 18.2cm size condition will include all results, after outlier removal, from every task where an 18.2cm object size was used). Outlier removal was conducted to two standard deviations for each task using a similar method to that described in 7.3.5.2.

$$RMS_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {E_{jk}}^2}$$

Equation 7-6. Calculation of the RMS lag/lead for each condition

Results and discussion. The RMS lag/lead error across all tasks was 5.45cm. The results for each condition are presented in Table 7-3, alongside the difference between the two modalities in MDOS and VDBH *along the Axis of Motion* only, so the measurements are based on the direction the object is moving.

To assess which method can achieve an MDOS closer to 0, the absolute value of the results are used, then the results of the Animated modality for each condition is subtracted from those of the Interactive modality. To assess which method can achieve the lowest VDBH, the results of the Animated modality for each condition are subtracted from the results of the Interactive modality. For either of these calculations, when a negative value is present the Interactive modality is superior to the Animated modality for that performance metric and condition.

Condition	18.2cm	36.4cm	54.5cm	0.17m/s	0.26m/s	0.35m/s	H-LR	H-TB	V-LR	V-TB
RMS lag/lead	4.46	5.42	6.30	2.74	4.50	7.84	6.83	6.83	1.19	4.87
MDOS diff.	-1.16	-0.68	0.4	-0.75	-0.88	-0.86	-0.5	-1.46	-0.26	-0.57
VDBH diff.	-0.12	0.17	0.37	0.03	0.1	0.3	0.23	-0.17	0.17	0.34

Table 7-3. Route Mean Square Lag/lead error result (cm) and difference in MDOS and VDBH between the Interactive and Animated modalities for each condition (cm).

Lag/lead error. The results presented in the top row of Table 7-3 demonstrate that a sizeable lag/lead error is present in all conditions.

The size of the virtual object appeared to have impact on the lag/lead, with larger errors appearing when a larger virtual object was used, increasing from 4.46cm for the 18.2cm condition to 6.3cm for the 54,5cm condition.

For the speed conditions a larger lag/lead error was detected as faster speeds were used, ranging from 2.74cm for the 0.17m/s condition to a relatively large 7.84cm for the 0.35m/s condition.

Hand Position also affected the lag/lead error. Motion along the Horizontal axis yielded a lag/lead error of 6.83cm for both H-LR and H-TB Hand Positions. Despite the fact that such similar results are detected for both of these conditions (only a negligible difference of 0.002cm), a statistically significant effect was found. This is because RMS was used, where negative results become positive; if the arithmetic mean was used, the difference would be 1.76cm between the H-LR (4.45cm) and H-TB (2.69cm) conditions.

An interesting result was observed for the V-LR hand position, where a lag/lead error of 1.19cm was observed, which is considerably lower than for the other Hand Positions. It is unclear why this specific hand position condition yielded such a low lag/lead error, although it could be due to the relative simplicity of the arm movement, where unlike other tasks, the V-LR condition only requires the actor to place their arms straight in front of their shoulders and move them down or up.

In summary, regardless of the condition a sizeable lag/lead error is present for the Animated modality.

Comparison of MDOS and VDBH. The MDOS difference between the modalities shows that in all but one of the conditions the Interactive modality allowed the actors to better estimate the size of the virtual object. Generally the improvement the Interactive modality allows is <-1.5cm. A repeated measures ANOVA (2x72) conducted between the tasks of the Animated and Interactive modalities yielded a result of $F_{(1,8)}$=2.638, p=0.143, indicating that there is no statistical significance between them.

The VDBH difference between modalities reveals that in all but two conditions the Animated modality yields a lower amount of variability between the actor's hands. However, this effect is not large as the best improvement in performance is only 0.37cm. So while the Animated modality appears consistently superior to the Interactive modality for VDBH, the improvement is negligible, which was reflected in the results of a one-way repeated measures ANOVA conducted between the tasks of the Animated and Interactive modalities, yielding no statistically significant result ($F_{(1,9)}$=0.632, p=0.447).

In each case, the difference between the Animated and Interactive modalities for the MDOS and VDBH is negligible when compared to the magnitude of the lag/lead error that would be introduced by the Animated modality. From these findings the recommendation made is:

3. Due to the presence of sizeable lag/lead errors for the Animated modality and the similar level of MDOS and VDBH performance between the Animated and Interactive modalities, the Interactive modality is preferable to the Animated modality when trying to create an accurate interaction.

Question 3. Are the MDOS and VDBH of the no-object modality comparable to those of the Interactive modality?

When capturing footage for post-production the actor will typically mime an interaction and the CGI object will be added into the scene later. Where possible, the motion of the actor will typically be constrained using some device. However, some devices that constrain the motion of the actor may be too difficult to either conceal from the audience or remove in post-production, so the actor would have to complete an unconstrained hand motion with live visual feedback that gave no information on the current object location either (as shown in Figure 7-12, page 106); the no-object modality was designed to replicate this experience. In these cases real time feedback on the location of the virtual object, as provided by the Interactive modality, may help improve the performance of the actor. Here the performance of the actors when using the Interactive and the no-object modalities are compared.

The average speed of the actor's motion during the no-object modality was 0.26m/s with a standard deviation of 0.1m/s. As the average speed is comparable to the 0.26m/s speed condition of the Interactive modality, the results of the no-object test can be directly compared with the results from the Interactive tasks that used this speed only.

Results. As with question 2, the absolute values of the results are used to assess which modality can achieve an MDOS closer to 0. For each condition the results for the no-object modality are subtracted from those of the Interactive modality. With VDBH, for each condition the results of the no-object modality are subtracted from the results of the Interactive modality. As before, a negative difference value indicates that the Interactive modality yields a superior performance. The results are presented in Table 7-4 and were assessed for statistical significance using a series of repeated measures ANOVA of a 2x3x2x2x2[11] design.

	Condition	18.2cm	36.4cm	54.5cm	H-LR	H-TB	V-LR	V-TB
Along AHP	**MDOS diff.**	-0.25	0.31	-2.16	0.02	-0.80	-0.03	-2.73
	VDBH diff.	-0.87	-1.67	-3.03	-2.77	-1.00	-1.00	-2.67
Perp. AHP	**MDOS diff.**	0.86	0.71	0.75	0.32	0.99	1.07	0.72
	VDBH diff.	-0.44	-0.55	-0.52	-0.48	-0.89	-0.16	-0.52

Table 7-4. Difference between Interactive and no-object modalities for each condition (cm)

Discussion and conclusion. The difference in MDOS AlongAHP indicates that the Interactive modality allows a consistently more accurate hand placement than the no-object modality, except in two cases (36.4cm and H-LR). These results were found to be statistically significant ($F_{(1,9)}$=11.385, p=0.008). The improvements generally range between -0.03cm and -0.80cm; but in two cases, 54.5cm and V-TB, the improvements are a more substantial -2.16cm and -2.73cm respectively. In the two cases where the no-object modality outperforms the Interaction modality, the benefit is minor (≤0.31cm).

[11] Modality x Size x Axis of Motion x Axis of Hand Placement x Direction

It is noted that the no-object modality consistently outperforms the Interactive modality for each condition when measured PerpendicularAHP for the MDOS, with performance improvement ranging from 0.32cm to 1.07cm, although his is not statistically significant ($F_{(1,5)}$=4.638, p=0.084). However, as discussed in section 7.3.4.4 the MDOS PerpendicularAHP should not affect realism, as this “error” describes something one would commonly see with an interaction with a real square or cube object, so the results for this are not considered significant.

The difference in VDBH sees the Interactive modality outperform the no-object modality considerably and consistently both AlongAHP and PerpendicularAHP (Along: $F_{(1,10)}$=21.02, p=0.001; Perpendicular: $F_{(1,5)}$=28.017, p=0.003). For the results AlongAHP, performance is improved by -0.87cm (18.2cm condition) to -3.03cm (54.5cm condition), which represents a massive reduction in the VDBH during the task. For the results PerpendicularAHP, performance is improved by -0.16cm (V-LR) to -0.89cm (H-TB).

From these findings it is concluded that when compared to the no-object modality, the Interactive modality would improve the overall ability of the actor in estimating the size of the virtual object AlongAHP and significantly reduce the amount of VDBH during an interaction.

These improvements made possible by the Interactive modality would benefit the motion of the actor when obtaining footage for post-produced interaction where an unconstrained hand placement technique is required. Therefore, the following recommendation is made:

4. The no-object modality is worse than the Interactive modality for the VDBH both AlongAHP and PerpendicularAHP. This holds true across all conditions. Therefore the Interactive modality is a more effective choice for unconstrained hand movement when recording material for post-production.

Summary of Chapter

This chapter presented the first step of a framework where the problems of actor motion in MR interaction systems were measured. Using MDOS and VDBH the actor’s interactive capabilities were measured, from which it was discovered that the size of the object impacts on actor performance, the target speed of the virtual object does not appear to affect actor performance and the VDBH is considerably affected by the actor’s hand positions. These results can considerably inform the conditions that yield effective performance for interaction using a system replicating that in chapter 6, in a system that accounts for the occlusion technique described in chapter 5 and an analogue to the feedback system described in chapter 4.

Chapter 8 : VIEWER PERCEPTION STUDY

8.1. Introduction

In this chapter the second part of the framework is presented, where a method to measure how adept an audience will be at perceiving inconsistencies between virtual objects and a real environment is demonstrated. In this particular case the framework is used to measure how well an audience can perceive a Mean Distance to Object Surface (MDOS) error created by the actor when interacting with a virtual object in the virtual studio.

This stage is based on the ITU-R BT.500 recommendations for Video Quality Assessment (VQA) (ITU-R, 1990), where a group of observers rate a series of video sequences based on their quality, with each video containing some form of deviation from an ideal (alongside a hidden reference). As no previous studies have applied this methodology to assessing mixed reality, a pilot study was conducted to assess the suitability of the method and the procedure and controls needed for the main experiment.

First, the proposed method is used to model how perceptible the actor MDOS errors are to viewers by collecting subjective data. Second, this method also allows the efficacy of different interaction systems in terms of plausibility to the average viewer to be evaluated. Plausibility in the context of this study is defined as the belief that an actor is accurately placing their hands on the surface of a virtual object that conforms to real world familiarity. This subjective data is used as a base against which the efficacy of any solutions can be *objectively* measured by collecting similar subjective data. Two potential solutions are proposed to mitigate the perceived imperfections.

One is the adaptation solution made to the virtual environment which involves the virtual object's size being adjusted to match the distance between the actor's hands, eradicating any gap or overlap that may arise from misestimation. The second is adaptation made to the real scene to see if adjusting the colour of the background exposed in the overestimation gap can be exploited to obscure the imperfections from the viewer.

These results will allow a list of guidelines to be produced that describe how effective solutions are in quantifiable terms and will be compared and contrasted to the motion capture data from chapter 7 to make further conclusions.

8.2. Background

In our proposed scenario the plausibility of a scene is reliant on the belief that the actor is making contact with the sides of the virtual object as they would with a real one. This is a case of a human interacting with a virtual object. However most previous studies into analysing plausibility of MR scenes have focused on assessing the discernibility of virtual objects as opposed to real objects.

Scene manipulation to reduce photorealism has been shown to be an effective method to reduce the viewer's tendency to perceive objects as non-real, in an augmented reality environment. Non-Photorealistic Rendering (NPR) (e.g. Haller (Haller, 2004) and Hertzman (Hertzman, 1998)) is a method of applying filters that produce "stylised" non-photorealistic effects in an attempt to normalise the real and the virtual objects in a scene. These filters typically place a thick border around detected edges and homogenise the textures in the scene, creating a painted effect. Conventionally this has been tested by presenting participants with real and virtual objects in an augmented reality scene and asking them to identify which ones are virtual.

Fischer et al (Fischer, et al., 2006) tested the effectiveness of NPR using 60 video sequences displayed on a monitor to 18 participants, who were required to identify whether the object in the scene was real or virtual. 15 real objects and 15 virtual counterparts were shown using 2 render modes, conventional AR and stylized. Participants were able to discern between the objects in 94% of conventional cases presented, but only in 69% of cases with the stylised mode. Steptoe et al (Steptoe, et al., 2014) also examined real time NPR, here using an augmented reality headset. 30 participants had to identify which of 10 objects placed in an area in front of them (5 real, 5 virtual) were real using 3 modes of rendering, conventional, stylized and virtualized, using 10 subjects for each mode. Virtualised mode presented the outlines of the virtual and real objects only. Stylised (56% accuracy) proved to be more successful than conventional augmented reality rendering (73%), with virtualised outperforming both (38%).

Whilst the above studies have relied on binary identification of objects in mixed reality scenes, the assessment of scene plausibility would be more effective when reviewed on a scale as the errors themselves exist at different values and so different value errors may only have a partial impact.

It is also noted that while NPR allows improved results, it is not explored in this chapter as the method of stylisation would not be applicable in the case of a realistic TV studio production.

8.2.1. Assessment of Video Quality

The plausibility as perceived by a viewer, of an interaction between an actor and a virtual object is subjective. When an actor fails to correctly estimate the size of a virtual object the principal factor is not the value of the error itself, but whether the viewer is able to recognize it or not. It is possible that the viewer may not be able to perceive when the actor has misestimated the size of the virtual object by a small amount, allowing a tolerance for estimation errors.

If the resulting visual artefacts such as gaps or overlap between the actor's hands and the virtual object's surface are perceived by the viewers it would break their "suspension of disbelief" and could make the interaction appear unrealistic. Such an effect is far from ideal for television as it would negatively impact on the viewer's experience; therefore it is necessary to measure how adept viewers are at perceiving the mistestimation of a virtual object by the actor.

This concern is analogous to one regularly encountered in the field of Video Quality Assessment (VQA), an example application being video compression. When encoding a video, the compression techniques used produce visible spatial artefacts (e.g. blockiness or posterizing (Winkler, et al., 2001), (Wang, et al., 2000) and (Wu & Yuen, 1997)) or temporal artefacts (e.g. dropped frames (Pastrana-Vidal, et al., 2004)), which can manifest themselves in varying degrees of magnitude. However, as with interaction in the interactive virtual studio these errors are unimportant unless the viewer is able to perceive them, which is what the VQA methodologies test for.

Misalignment between the hand of an actor and the surface of a virtual object would present itself as a type of spatial artefact analogous to those arising from compression. In this sense, they are both errors that occur to a varying degree that occupy a certain space on the screen, and that may be visible to an observer.

From this, the commonality between assessing the quality of video and the quality of interaction in the virtual studio becomes clear. Both take into consideration spatial artefacts that may affect the quality of a scene as it is perceived by a viewer into consideration. It is believed that for this reason VQA methods have a high degree of relevance in the interactive virtual studio domain, and by extension other related mixed reality domains. Until now the VQA methods have never been used to assess the quality of interaction in the interactive virtual studio.

The standard method for assessing the perception of compression artefacts is to depict them in a sequence of video segments shown to a set of human observers. The video segments present one or multiple source videos with no distortion, alongside a number of other videos that have been adapted from the source videos to replicate a distortion to some extent. The observers are asked to rate the quality of each video segment using a set of discrete categories that are described adjectively, but can be represented numerically (e.g. 1=Terrible ... 5=Excellent). The average result of the equivalent numerical values for each video is presented as a Mean Opinion Score (MOS), a standard way of representing the mean that can be used to objectively measure the perception of errors. This allows evaluators to draw compare the performances of different groups and conditions. The results of the studies in this chapter are presented using the MOS, which in the context of this work is calculated using Equation 8-1.

$$\bar{\mathrm{u}}_{\mathrm{jk}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i=l}}^{\mathrm{N}} \mathrm{u}_{\mathrm{ijk}}$$

Equation 8-1. Calculation of Mean Opinion Score

Where, N is the number of observers after the removal of outliers and u_{ijk} is score of the observer, with i representing the observer, j representing the hand distance condition (i.e. the 18.2cm (50px), 36.4cm (100px) and 54.5cm (150px) distances – described in page 149) and k the error condition (e.g. the value of the MDOS error or object growth replicated). Graphically each MOS is presented with a 95% confidence interval.

8.2.2. The ITU-R Recommendations

VQA is standardised in the ITU-R BT.500-13 Recommendations for the Subjective Assessment of Television Pictures [ITU, 2012] (henceforth BT.500), which outlines the methods of presenting videos for subjective assessment. These methods are broadly categorised as Single Stimulus and Double Stimulus, which are further divided into the following sub-categories:

Single Stimulus –

Single Stimulus (SS): Observers are presented with each video segment once, which they rate as an independent entity on an adjectival or numerical scale.

Single Stimulus with Multiple Repetitions (SSMR): Observers are presented each single video segment multiple times, each time they rate it as an independent entity on an adjectival or numerical scale.

Single Stimulus Continuous Quality Scale (SSCQS): Observers are presented with a single video segment, which they rate on a Continuous Quality Scale. This scale tracks the observers' experience over time, where they are assessing the transient quality of a video using a slider type input.

Double Stimulus –

Double Stimulus Impairment Scale (DSIS): Observers are presented with two video segments, in which they rate the quality of the second on a comparative scale to the first (i.e. negative to positive comparative scores).

Double Stimulus Continuous Quality Scale (DSCQS): Observers are presented with two video segments, one an unimpaired reference and the other an impaired video, which they compared and rated on a Continuous Quality Scale.

When viewers are watching television in real life it is unlikely they would be presented with a situation where they are able to directly compare two interactions between an actor and virtual object and this experience should be replicated in the experiment. Accordingly, this experiment will use a Single Stimulus method over Double Stimulus methods to make the observer rate each video segment as its own entity, as they would when watching television. This is not possible with Double Stimulus methods as the observers would make comparisons between the videos presented to them and this could alter their perception of plausibility (i.e. they may realise they are being deceived when one video shows no manipulations made to a virtual object and the other does).

The basic form of Single Stimulus proved to be a logical choice when selecting the type of Single Stimulus method to be used for this experiment. Continuous Quality Evaluation (SSCQS) was not selected as the temporal nature of artefacts of the video do not need to be measured in order to understand the overall quality of the interaction at this stage, as the value of the errors presented are constant. Multiple repetitions (SSMR) were not included as part of this experiment as this method is not relevant to a television setting, where typically the viewer would not see multiple repetitions of the same interaction.

Types of Single Stimulus perceptual ratings

Three types of Single Stimulus methods are typically used; each one requires the observers to assess the video sequences in different ways:

Numerical Categorical Judgement methods. For the Numerical Categorical Judgement method observers use an 11 category numerical scale to rate the quality of the video sequences. A study has shown that this method has a high degree of sensitivity and stability when assessing the quality of videos (ITU-R, 1990). However, the video segments are only scored using a number, meaning that it is more difficult to assess the deviation from the ideal in terms of quality,

Non-Categorical Judgement methods. Two forms of Non-Categorical judgement methods are used, Continuous Scaling and Numerical Scaling. For Continuous Scaling an observer will rate the quality of a video sequence by marking its quality at some point along a line that is bounded by two adjective descriptions and can be focused to a particular feature of interest (e.g. for VQA one end of the line says very poor and the other says very good, the observer marks a point on the line where they think the quality lies). The Numerical Scaling method is similar except the scale is quantised (e.g. the observer selects a value between 0-100, for example 0 representing a blurred image and 100 representing a sharp image).

Adjectival Categorical Judgement methods. For the Adjectival Categorical Judgement the observers rate each video sequence presented to them using a set of adjectives, which are linked to numerical values. The ITU present two sets of adjectives that can be used for this particular type of study, 'Quality' and 'Impairment', as presented in Table 8-1. The Quality scale is used to assess how the observers rate the overall quality of the video. This was successfully used by Sazzad *et al* (Sazzad, et al., 2009) for assessing the quality of stereoscopic images. The Impairment scale is used to determine whether the observers are able to detect an artefacts that arise from an impairment from an ideal. This was successfully used by Klaue *et al* (Klaue, et al., 2003) who demonstrated a framework for assessing compression artefacts in MPEG4 videos streamed over a network.

Five-grade Adjectival Categorical Judgement scale			
Quality		**Impairment**	
5	Excellent	5	Imperceptible
4	Good	4	Perceptible, but not annoying
3	Fair	3	Slightly annoying
2	Poor	2	Annoying
1	Bad	1	Very annoying

Table 8-1. The Adjectival Categorical Judgement scales as defined by the ITU

For this study the Impairment Scale of the Adjectival Categorical Judgement method will be used to measure the perception of misalignment between the hands of the actor and the virtual object. This is because a gap between the actor's hand and the surface of the virtual object is considered to be a specific impairment from the ideal of a perfect alignment, which is not something that could be described by assessing the overall quality of the video.

8.2.3. Use outside of Video Quality Assessment for Video Codecs

Demonstrating the transferability of the ITU-R BT.500 methodology is important for the work presented throughout this chapter. This is because measuring a misalignment between an actor's hands and the surfaces of a virtual object would be considered and an atypical and novel application of the BT.500 recommendations. Here, we present examples that demonstrate that although the methodology is designed for assessing video codecs, it is a flexible methodology that can be used in studies that are removed from this original purpose.

Deshpande (Deshpande, 2009) presented a method for assessing the perception of mismatched synchronisation between the monitors in tiled video displays, a form of display where multiple monitors are placed together and show a single video image, which imitates a larger display. Using the BT.500 recommendations as a base for their methodology, Deshpande was able to successfully analyse the effect that mismatched synchronisation between the tiles of the display had on the observers. The author was able to conclude that the observers could perceive mismatched synchronisation more easily for the 3x3 arrangement than the 1x2 or 2x1 arrangements, and also that a larger mismatched synchronisation was more likely to be perceived by the observers. The perception of mismatched synchronisation between displays is a far-removed and atypical use of the BT.500 recommendations, yet it allowed the author to successfully draw significant conclusions. As such, this demonstrates that the method is flexible enough for use in atypical studies.

IJsselsteijn *et al* (IJsselsteijn, et al., 1998) conducted a study addressing the issue of "presence" when viewing material presented with a stereoscopic (3D) display. The concept of presence relates to the feeling that one exists in a time and space, which the authors argue is a feature that is more apparent in stereoscopic displays compared with conventional 2D displays in certain settings. In this study 12 observers (and 18 in a second experiment) viewed three 8-minute videos in an SSCQE format, continuously rating their sense of presence in the scene. From these results the authors were able to identify the impact that certain visual artefacts associated with stereoscopy have on presence. For example, during the video presentation a particular scene had inconsistent depth and occlusion cues, both of which are the effect of an incorrect quality stereoscopic recording of the material. From the sudden drop in the perceptual ratings during this scene it was possible to identify that the observers' sense of presence was considerably reduced, particularly when compared to scenes where these cues were consistent with reality. The fact that the authors were able to draw this and similar conclusions about *presence* in this study highlights that the BT.500 recommendations can be used to measure even abstract concepts.

As demonstrated using the studies presented, the BT.500 recommendations are flexible and can remain useful outside their original sphere of application: for example, in measuring how interaction errors are perceived by an audience.

However, to support this claim a pilot study was also conducted (presented in Appendix E, page 208). The Pilot study used the proposed methodology of this experiment to address the following objectives:

- To confirm that the Single Stimulus method of for rating perceived quality is transferable to assessing the perception of interaction errors in the virtual studio.
- To inform on the conditions of the final test.

The pilot study confirmed that the Single Stimulus approach was transferable for use in this experiment. It also allowed conclusions to be made on the quality of the videos produced (notably using a standard broadcast resolution, using appropriately detailed 3D objects, ensuring that the videos have a consistent level of quality) and the test arrangement (notably informing on the number of videos presented and viewing distance from the monitor). Full details of the findings are presented in Appendix E.

8.3. Experimental Methodology

This methodology the experiment procedure and controls (8.3.1.), the manner of video production (8.2.2.), the method of data analysis (8.3.3.) and a description of the experiments that will be conducted in this experiment (8.3.4.).

8.3.1. Experiment Procedure/Controls

The experiment design and conduct was based on the Single Stimulus method as suggested in BT.500 and confirmed in the pilot study. The observers were presented with a sequence of video segments shown one of two pseudo-random orders with no repetition[12]. The observers did not have prior knowledge about the nature of the experiment.

A total of two experiment sessions were conducted. The reason two were conducted is that from the first session a sudden and considerable degradation in the MOS was detected when a gap between the hands and the object surface of around +10 to +15px was present (Reported on in page 153). In the first session the Speed of Growth and the Background Colour studies used a replicated MDOS of +20px. It was decided that the presence of such a large gap would be too distracting, masking the potential positive or negative impact that these features could have. Consequently, these studies were conducted again in the second experiment session with the gap was reduced to +10px to ensure a more accurate result. The first session was conducted over a period of 5 days and the second session over 4 days. The first and second sessions will henceforth be referred to as 20px session and 10px session respectively.

[12] The BT.500 specifies a preference for presenting video sequences in a unique pseudo-random order for each session. However, it also states that it is acceptable to present the sequence in a set number of pseudo-random sequences across the sessions provided that the same picture or test sequence is not presented on two successive occasions, which was ensured throughout these experiments.

The experiment procedure is as follows:

Pre-experiment. Each observer was asked to complete an Ishihara test for colour blindness, a Snellen chart test to measure their visual acuity and sign a consent form. The observers were each read the following statement before the experiment:

"You are about to see a series of videos, each depicting an interaction between an actor and a virtual box. You will rate each video from 1 – Very annoying to 5 – Imperceptible based on how realistic the interaction appears. Each video will be played once followed by a 5 second blank, during which time you will rate the video by ticking the appropriate box on the form you have been given. The session will start off with a short sample, showing the types of video and the structure of the experiment. This will be followed by a question session. The experiment will commence after that"

Experiment. The first experiment session presentation shown to the observers used the following format with 21 observers each watching 73 videos (9 stabilising videos and 64 test videos):

- A training sequence of 7 video segments encompassing the range of conditions the observers would be presented with throughout the presentation (including a reference). Observers were told to familiarise themselves with the content and not to rate them
- A Question and Answer session.
- Presentation of 73 video segments for subjective assessment by observers. The video segments were presented in the following order:
    - 9 video segments encompassing a wide range of conditions to help stabilise the observer's results. The ratings for these videos were discarded.
    - 64 randomised video segments showing the full range of conditions for assessment.
    - Between each video segment a 5 second mid-grey post exposure field was shown for the observers to rate the video.

The second experiment session presentation was conducted using the same format, with 15 observers watching 20 videos (3 stabilising videos and 17 test videos).

- A training sequence of 7 video segments encompassing the range of conditions the observers would be presented with throughout the presentation. Observers were told to familiarise themselves with the content and not to rate them
- A Question and Answer session.
- Presentation of 20 randomised video segments for subjective assessment by observers. The video segments were presented in the following order:
    - 3 video segments encompassing a wide range of conditions to help stabilise the observer's results. The ratings for these videos were discarded.
    - 17 videos segments showing the full range of conditions for assessment.
    - Between each video segment a 5 second mid-grey post exposure field was shown for the observers to rate the video.

Post-experiment. Each subject was given a debriefing, where they were told the aims of the experiment and types of videos that were presented. They were then asked for any comments they had.

Arrangement

The video demonstration was shown to the observers in compliance with the set-up described in BT.500. The experiment was conducted in a low luminance environment, using a 14 inch calibrated JVC (TM-H140PN) video monitor adjusted to the BT.500 recommended specifications. The high quality CRT monitor was selected to avoid the low refresh rates and motion blur of LCD monitors that could interfere with the results of the experiment, as with (Seshadrinathan, et al., 2010). The approximate distance between each observer and the video monitor was 150cm, around the Preferred Viewing Distance (ITU-R, 1990) for a monitor of this size.

For the recording of the results method a computer-based interface was initially considered but rejected in favour of a scoring card method, similar to a multiple-choice exam form where the appropriate box is filled in to indicate the selection.

Observers were asked to rate each video using 5 discrete quality categories as described by the Adjective Categorical Judgement system using the impairment phrasing. These quality categories are assigned a numerical value, in this case:

5 = Imperceptible.
4 = Perceptible, but not annoying.
3 = Slightly Annoying.
2 = Annoying.
1 = Very Annoying.

Participants

Observers. A total of 22 observers were used for the first experiment and 15 for the second experiment. All observers had at least an undergraduate education, typically from a computer science background. None were experts in this particular field.

Visual Acuity. Each observer's visual acuity was measured using a Snellen chart, with average left eye acuity of 0.92 and right eye acuity of 0.89 (where 1 is equivalent to 20/20); the observers used any visual aid they would typically wear for viewing television (i.e. Glasses and contact lenses). Observers with a visual acuity of <0.8 in either eye were not allowed to participate. A total of 3 observers were rejected under these criteria.

Colour blindness. Each observer was required to score 6 out of 6 on a standard Ishihara test for colour blindness, indicating that they are able to perceive all colours correctly. All observers passed this requirement.

8.3.2. Video production

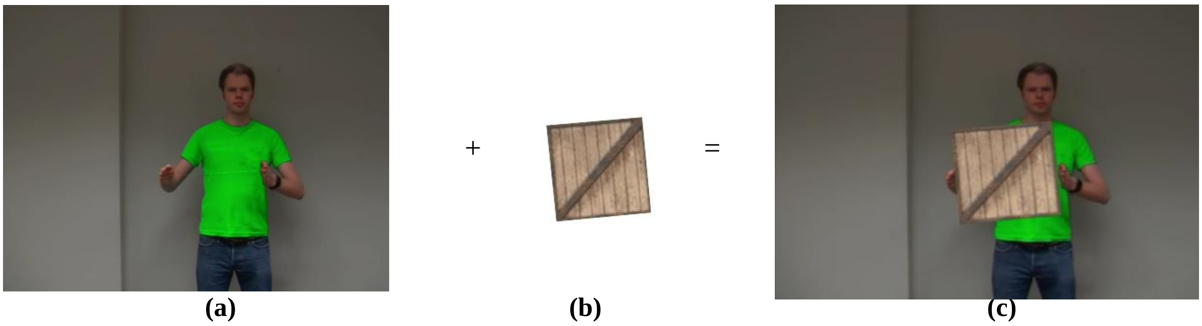
A single source video was recorded for the 50px, 100px and 150px hand distance conditions, which was used across all studies; as well as the separate videos required for the Matched adaptation study. Each video segment was shown to the observers using the native specifications of the video monitor and were approximately 4 seconds in length.

The video footage was recorded in a high bit rate H.264 format at a high data rate of >4Mbs using a film industry level High Definition camera (Panasonic DVC-30) in a professional level film studio with appropriate lighting. To match the specifications of the monitor used in the experiment, the captured footage was down-sampled from 1080*1920 to 480*640 in post-production. Down-sampling the footage required changing the aspect ratio from 16:9 to 4:3, where the footage is cropped so no potentially distracting "squeezing” effects would occur. These specifications were approximate to that of a standard 4:3 PAL broadcast.

When recording the footage it was important to constrain the motion of the actor to control motion errors that may impact the quality of the videos. A telescopic apparatus was used that could be set to hold the hands at fixed distance apart, or could be set to increase in length by a specified amount throughout the interaction. The (thin) telescopic apparatus was removed from the video in post-production without leaving any significant visual artefacts, as can be seen in Figure 8-1a where it has been removed from the image.

For the first experiment session a total of 70 video segments replicating the interactive virtual studio were produced using Adobe After Effects CS5, an industry standard special effects software package. 17 videos were produced for the second experiment session. In each video the base of the actor's middle fingers were tracked using the ‘Track Motion’ feature, from which the location where the apparatus meets the hand for both hands could be measured for each frame. From these two points the midpoint and angle between the hands could be calculated, as well as any scaling that might occur.

The virtual object followed the midpoint between the actor's hands and maintained the correct angle throughout the video, as a real object would. The virtual object used for this experiment is a simple cube with a wooden crate texture, which is selected as it represents a basic and widely used shape for virtual objects. An example of this process is shown in Figure 8-1.



(a) (b) (c)

Figure 8-1. Composition of the video, where (a) is the recorded footage (b) is the virtual object and (c) is the composited output.

8.3.3. Data Analysis

8.3.3.1. Removal of Outliers

The data was processed based on the rejection criteria suggested in BT.500, which is designed to remove frequently outlying observers. This removal outlier process identifies observers that frequently rated videos significantly far from the mean.

For each presentation the Kurtosis of the results is calculated using the β2 test, where β2 represents the kurtosis coefficient. If the presentations yielded a kurtosis coefficient of $2 \leq \beta 2 \leq 4$ the data was assumed to be normally distributed. In the cases where data for the presentation was normally distributed, outliers were identified as being two standard deviations from the mean. In cases where the data for the presentation was non-normally distributed, outliers were identified as being √20 (i.e. 4.47) standard deviations from the mean.

For each experiment, each time a participant's score is found to be $u \geq \bar{u}+2\sigma$ or $u \geq \bar{u}+\sqrt{20}\sigma$ depending on normality, a counter associated with that participant, Pi, is incremented; For each time a participant's score is found to be $\bar{u}-2\sigma \leq u$, a counter associated with that participant, Qi, is incremented.

For each participant two ratios are calculated. $(P_i+Q_i)/L$, where L is the total number of video sequences shown to the participant and $(P_i-Q_i)/(P_i+Q_i)$. If the first ratio is >5% and the second ratio is also <30%, then the participant is removed from the data. Using this outlier removal criteria 1 frequently outlying subject was removed under these criteria, leaving 21 suitable observers. Represented mathematically the outlier removal process is:

```
if $2 \leq \beta 2_{jkl}$
        if $u_{jkl} \geq \bar{u}_{jkl} + 2\, S_{jkl}$              then $P_i = Pi + 1$
        if $u_{jkl} \leq \bar{u}_{jkl} + 2\, S_{jkl}$              then $Q_i = Qi + 1$
if $2 \geq \beta 2_{jkl}$
        if $u_{jkl} \geq \bar{u}_{jkl} + \sqrt{20}\, S_{jkl}$      then $P_i = Pi + 1$
        if $u_{jkl} \leq \bar{u}_{jkl} + \sqrt{20}\, S_{jkl}$      then $Q_i = Qi + 1$
if $\frac{P_i+Q_i}{JK} > 5\%$   or   if $\frac{P_i-Q_i}{P_i+Q_i} < 30\%$     then reject observer $i$
```

Where, N is the number of observers after the removal of outliers and u_{ijk} is score of the observer, with i representing the observer, j representing the hand distance (i.e. the 50px, 100px and 150px hand distances) and k the error condition (e.g. the value of the misestimation or object growth replicated).

8.3.3.2. Statistical Analysis

Statistical Analysis of Experiments

The data collected was ordinal, non-parametric, used repeated measures and had not been normalised. With these elements considered, analysis for statistical significance of each condition is conducted using a Friedman's test with an alpha of 5%. This is an alternative type of ANOVA that is used for the same format of the data collected. If the p values is <0.05, then a significant effect has been detected across the conditions.

Post-hoc analysis is conducted using a pairwise series two-tailed Wilcoxon Signed Ranks test with an alpha of 5% with a Bonferroni adjustment applied. The two-tailed Wilcoxon Signed ranks test allows two conditions to be directly compared for statistical significance, which is conducted for every pair in study condition. The Bonferroni adjustment changes the value of the alpha using the formula α/N, where α is the original value of the alpha and N is the number of pairwise tests conducted. If the *p* value of the Wilcoxon Signed Ranks test is lower than the value provided by the Bonferroni adjustment, then the result is determined to be statistically significant.

Comparing Static Size study to Tweened adaptation and Matched adaptation studies

When comparing the performance of two different studies (i.e. comparing the static size object to an adapting size object), a standard two-way repeated measures ANOVA would not provide an accurate result due to the format of the data collected in this experiment being non-parametric and ordinal. No alternative statistical test compatible with these requirements could be found. Therefore statistical analysis will be performed as a series of 2-tailed Wilcoxon signed ranks test with an alpha of 5%, using a pairwise comparison between the corresponding conditions of each condition. Again, a Bonferroni adjustment is made, with the result being statistically significant if it is lower that the value it provides.

8.3.3.3. Units used

Throughout this chapter the distances involved in the measurement of the replicated errors as they appear in the video image are described using pixel units, which is a single picture element in an image. Although a pixel is a single point on an image with no spatial dimension, when shown on a video monitor it takes up a portion of space that can be considered a 2D spatial element. This makes it a useful unit for this experiment.

However, pixel units only provide valid results within the context of this test arrangement, with the observers sat a distance of 150cm from a video monitor with the same size and resolution. This does not describe what portion of the observer's field of view an artefact fills, which is important if these results are to be compared to other experiments with different arrangements. An arcminute is a unit that measures the angle subtended at the origin (in this case the eye) by an object (in this case an object in the visual field). A single arc minute is the equivalent of 1/60th of a degree.

Although measurements in arcminutes are not given in the results of this experiment, they can be calculated using Equation 8.2, where S represents the real size of the pixels as they appear on screen (e.g. in millimetres) and D represents the distance from the observer to the monitor. Table 8-2 (page 148) presents a conversion between these units. In addition to arc-minutes, the equivalent between the pixel units and the cm units as used in chapter 7 are also presented for reference.

$$Arcminutes = \tan^{-1}\left(\frac{S}{D}\right)60$$

Equation 8-2. Calculation of Arcminutes

Number of pixels (px)	**Size in mm on screen**	**Arcminutes**	**Motion error in cm from chap. 7**
5	2.35	5.39`	1.82
10	4.7	10.77`	3.64
15	7.05	16.16`	5.45
20	9.4	21.54`	7.27
30	14.1	32.32`	10.91
40	18.8	43.09`	14.54
50	23.5	53.86’	18.18
100	47	107.72’	36.36
150	70.5	161.58’	54.54

Table 8-2. Conversion table for the size of a visual artefact in terms of number of pixels, onscreen size in mm, arcminutes at 1500mm from the monitor and the equivalent error from Chapter 7’s motion study.

8.3.4. Experiments

Across the two experiment sessions a total of 7 studies were conducted, each of which is briefly described in Figure 8-2. Some studies were assessed multiple times using different distances between the actor's hands to detect whether any effects on the viewer's perception of errors are proportional or absolute. These sizes are 50px, 100px and 150px[13], which correspond to the object sizes of 18.2cm, 36.4cm and 54.5cm used in chapter 7 respectively.

Study		**Description**	**Hand Distances**
Static Size study (Section 8.4.1)		The object is the same size throughout the video, replicating the MDOS errors observed in chapter 7 without any variability between the hands (VDBH). This will be used as a base for comparing the solutions tested.	50px, 100px, 150px
Virtual Environment Scene Adaptation	**Tweened adaptation (Section 8.4.2.1)**	The actor is overestimating the size of the object at the start of the video. Throughout the interaction the size of the object will grow to fit the distance between the actor's hands, cancelling out any MDOS error. The growth occurs over a period of 1.42s.	100px, 150px
	Speed of Tween (Section 8.4.2.1)	In additional to the tweened adaptation study , the speed of the tween was also considered. Using a starting MDOS overestimation of +20px, the object grows to fit the hands at different rates for each level. This study was repeated with the size of the gap between the actor's hands and the virtual object reduced.	150px
	Matched adaptation (Section 8.4.2.2)	The hands of the actor drift further apart during the interaction, replicating a type of VDBH error. The virtual object continuously adapts to match the changing distance between the actor's hands	100px, 150px
Real Scene Adaptation	**Background colour (Section 8.4.3)**	A gap is seen between the hands of the actor and the surface of the box exposing the colour of the actor's shirt, which is digitally altered to represent a range of background colour conditions. The study is repeated with the size of the gap between the actor's hands and the virtual object reduced.	150px
Occlusion	**Changing Occlusion (Section 8.4.4)**	The occlusion changes during an interaction in a way that violates the design of the occlusion system discussed in chapter 5 (hands move from in front of the box to behind and vice versa), which is included to further test the viability of that system.	150px

Figure 8-2. Studies conducted

[13] **A note on the nature of the object sizes and hand distance**

Per the findings in the pilot study (section 8.4.3) a single source video had to be used where possible to ensure uniformity across the video sequences. As such, instead of recording many videos with different distances between the actor's hands, one source video was recorded for each sub-condition where the distance between the actor's hands was constant. In total three source videos were recorded, one with a 50px hand distance, one with a 100px hand distance and one with a 150px hand distance.

To replicate the MDOS errors detected in the previous chapter the size of the object became the variable feature. This is to say that if an overestimation is to be presented, the size of the virtual object is reduced (replicating a positive MDOS). Similarly the size of the object is enlarged if an underestimation is desired (replicating a negative MDOS). Therefore these videos do not exactly replicate the MDOS errors from chapter 7, but *approximate* them. This concession is required, as recording enough videos to directly replicate them would produce many unstable artefacts between and thus rendering the results unreliable.

Three of the studies fell under the heading of ‘Virtual Environment Scene Adaptation’, which explores the effect that changing the size of the object to meet the distance between the actor’s hands has on the observers’ experience. This adaptation takes place in two stages: The first stage is ‘tweened adaptation’, where the object grows from its original size to fit the distance between the actor’s hands; the second stage is matched adaptation, where after the object has changed to fit the distance between the actor’s hands it will continue to change size to match any subsequent variation between them. No third stage is present, as the remaining study was a supplementary study that explores the effects of varying the speed of object growth.

A brief description for each of these studies will be presented at the start of each discussion session, including the conditions assessed (and their derivation) and the hypotheses that will be tested.

8.4. Results and Discussion

8.4.1. Static Size study

8.4.1.1. Experiment Summary

This experiment explored the impact that an (Actor’s) MDOS error, as described in chapter 7, had on the observers. No Variability between the hands (VDBH) was shown.

An error in the Static Size (object) study is equivalent to the MDOS error measured in Chapter 7 (assuming a VDBH of 0), where the actor misestimates the size of the object by a fixed amount throughout the interaction and the virtual object is a constant size. From the viewer’s perspective, an MDOS error would result in the actor appearing to place their hands inside (underestimation) or outside (overestimation) the virtual object's surface. Examples of these errors are shown in Figure 8-3.

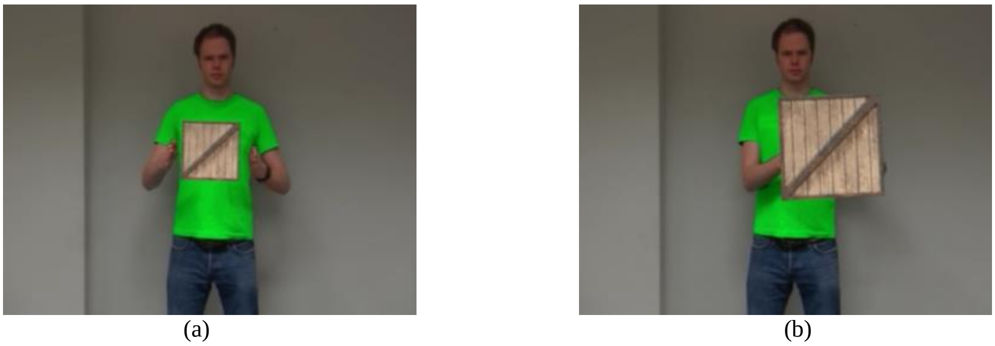


Figure 8-3. (a) Example of an actor overestimating the size of a virtual object by 20px, causing a gap between the surface of the virtual object and their hand and (b) the actor underestimating the size of a virtual object by 20px, causing their hands to appear behind the virtual object

For the Static Size study the following hypothesis will be assessed:

Static Size error - *As the actor's misestimation of the virtual object size becomes more extreme, the observers will be more likely to perceive the error.*

Object Size	Mean(px)	1σ (px)	-2 σ	+2σ
50px	4.37	9.32	-14.27	23.01
100px	2.64	14.24	-25.84	31.12
150px	-1.43	16.09	-33.61	30.75

Table 8-3. Results from the actor motion study showing the MDOS errors (px) to 2 Standard Deviations for each virtual size condition. Here 50px, 100px and 150px represent the object sizes from chapter 7.

For this study the range of Static Size conditions that will be shown are based on the MDOS errors AlongAHP for the Interactive modality to two standard deviations (from section 7). This range is presented in Table 8-3, where they have been converted into pixel units.

The range of replicated Static Size errors is split into equal divisions for all three hand distances. The base unit selected for these divisions is 10% of the smallest hand placement distance, which is 5px. The conditions (the size of the virtual object) will be incremented by the base unit of 5px starting from a perfect fit of a 0px error, which is also shown as a reference video. From ±20px onwards the conditions are incremented by 10px instead for the sake of brevity in the experiment.

Also included are two outliers that replicate errors to 4 standard deviations from the mean, which are used as anchors for each condition. The resulting conditions that will be replicated for the Static Size study in are shown in Table 8-4 (page 151). Screen shots of the videos used in each study are presented in Figure 8-4 (page 151).

Hand Distance	**MDOS levels to be replicated in the perception experiment (Px)**												
	Underestimation increments							Overestimation increments					
	-4σ	-6	-4	-3	-2	-1	0	+1	+2	+3	+4	+6	+4σ
50px	-45		-20	-15	-10	-5	0	5	10	15			35
100px	-65	-30	-20	-15	-10	-5	0	5	10	15	20	30	60
150px	-65	-30	-20	-15	-10	-5	0	5	10	15	20	30	70

Table 8-4 The levels presented for the Static Size condition, based on the MDOS errors in chapter 7

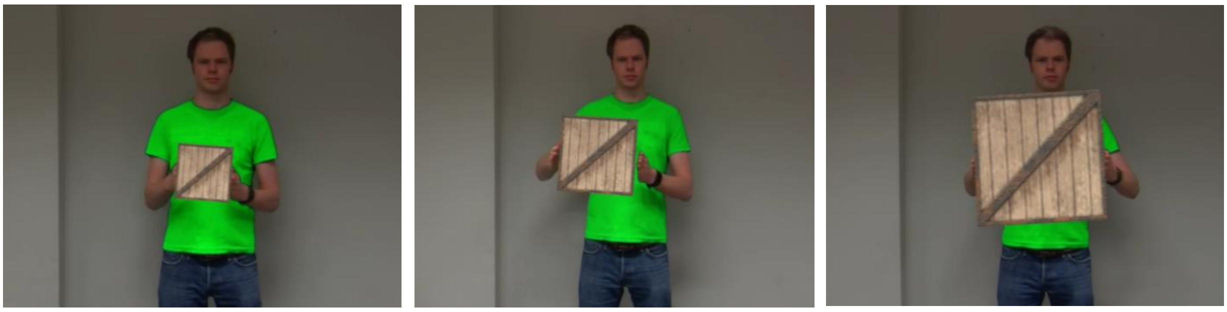
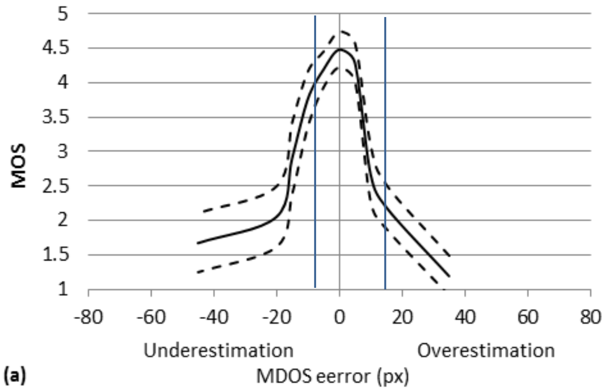

(a) (b) (c)

Figure 8-4. Screen shots from the static error experiment. (a) from the 50px hand distance with an MDOS error of -10px. (b) from the 100px hand distance condition with an MDOS of +20px virtual object size. (c) from the 150px hand distance condition with a 0px MDOS error.

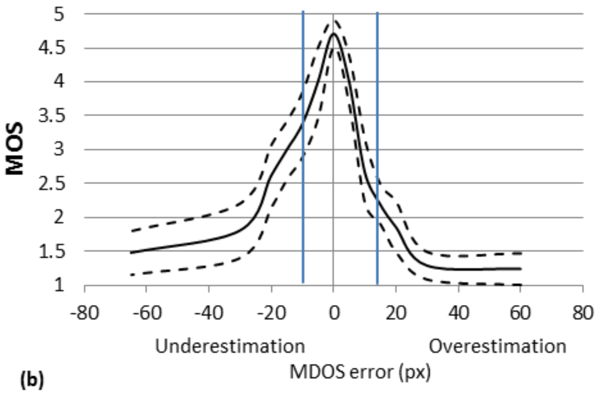
8.4.1.2. Results

The data plotted in Figure 8-5 (page 152) shows the results for the Static Size error for the 50px, 100px and 150px hand distance conditions. As hypothesised, as the MDOS errors became more extreme from 0px, the MOSs degraded. The results of the Friedman's ANOVA for this study were – 50px: χ^2=134.76, *p*=<0.001; 100px: χ^2=181.625, *p*=<0.001; 150px: χ^2=177.425, *p*=<0.001. These results indicate that this outcome is statistically significant for each hand distance condition.



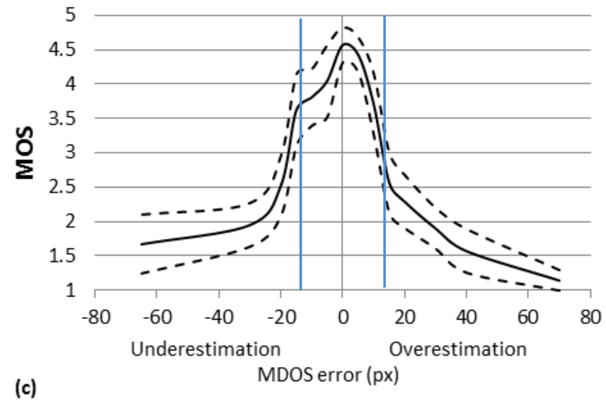
MDOS(px)	Total votes	Rating value frequency 1	2	3	4	5	MOS	CI	Std Dev
-45	21	12	6	2	0	1	**1.67**	**0.43**	**1.02**
-20	21	7	9	2	3	0	**2.05**	**0.44**	**1.02**
-15	21	3	5	6	5	2	**2.9**	**0.52**	**1.22**
-10	21	0	2	6	8	5	**3.76**	**0.4**	**0.94**
-5	21	0	0	2	13	6	**4.19**	**0.26**	**0.6**
0	21	0	0	1	9	11	**4.48**	**0.26**	**0.6**
5	21	0	0	2	11	8	**4.29**	**0.28**	**0.64**
10	21	2	8	6	5	0	**2.67**	**0.41**	**0.97**
15	21	3	12	5	1	0	**2.19**	**0.32**	**0.75**
35	21	19	1	0	1	0	**1.19**	**0.29**	**0.68**

(a)



MDOS(px)	Total votes	Rating value frequency 1	2	3	4	5	MOS	CI	Std Dev
-65	21	14	4	3	0	0	**1.48**	**0.32**	**0.75**
-30	21	9	9	1	2	0	**1.81**	**0.4**	**0.93**
-20	21	3	8	4	6	0	**2.62**	**0.46**	**1.07**
-15	21	1	6	7	6	1	**3**	**0.43**	**1**
-10	21	1	4	5	8	3	**3.38**	**0.48**	**1.12**
-5	21	1	2	3	5	10	**4**	**0.52**	**1.22**
0	21	0	0	0	6	15	**4.71**	**0.2**	**0.46**
5	21	0	1	4	10	6	**4**	**0.36**	**0.84**
10	21	3	7	6	4	1	**2.67**	**0.47**	**1.11**
15	21	2	14	4	1	0	**2.19**	**0.29**	**0.68**
20	21	8	9	3	1	0	**1.86**	**0.37**	**0.85**
30	21	15	6	0	0	0	**1.29**	**0.2**	**0.46**
60	21	17	3	1	0	0	**1.24**	**0.23**	**0.54**

(b)



MDOS(px)	Total votes	Rating value frequency 1	2	3	4	5	MOS	CI	Std Dev
-65	21	13	4	2	2	0	**1.67**	**0.43**	**1.02**
-30	21	5	13	2	1	0	**1.95**	**0.32**	**0.74**
-20	21	4	6	7	4	0	**2.52**	**0.44**	**1.03**
-15	21	1	3	6	4	7	**3.62**	**0.53**	**1.24**
-10	21	0	2	6	7	6	**3.81**	**0.42**	**0.98**
-5	21	2	0	3	6	10	**4.05**	**0.53**	**1.24**
0	21	0	0	1	7	13	**4.57**	**0.26**	**0.6**
5	21	0	0	1	10	10	**4.43**	**0.26**	**0.6**
10	21	1	1	6	8	5	**3.71**	**0.45**	**1.06**
15	21	3	7	8	2	1	**2.57**	**0.44**	**1.03**
20	21	3	12	3	3	0	**2.29**	**0.39**	**0.9**
30	21	6	11	4	0	0	**1.9**	**0.3**	**0.7**
70	21	18	3	0	0	0	**1.14**	**0.15**	**0.36**

(c)

Figure 8-5. Results from the static error experiment, showing the MOS for the 50px (a), the 100px (b) and the 150px hand distance conditions (c). The blue bars show the MDOS error measured in chapter 7 to 1 SD from the mean. The dashed lines represent a 95% confidence interval.

The most striking feature of the data plotted in Figure 8-5 was the asymmetry. For each size condition, as the replicated MDOS error decreased (representing an underestimation) the MOS degradation was seen to decrease steadily and began to plateau as the MDOS errors became more extreme.

However, as the replicated MDOS error increased (representing an overestimation) a severe and sudden degradation in the MOS was observed. This feature indicated that a gap between the actor's hands and the virtual object surface was particularly noticeable to the observers and once it reached a particular threshold they become more aware of the false looking interaction and responded poorly to it; more so than with the equivalent underestimation. These results are presented in Table 8-5.

Hand distance	+5px MDOS	+10px MDOS	Total degredation
50px	4.29	2.76	-1.53
100px	4	2.67	-1.33
	+10px MDOS	**+15px MDOS**	**Total degredation**
150px	3.71	2.57	-1.14

Table 8-5. MOSs of virtual objects between the +5px and +10px MDOS for the 50px/100px hand distance and +10px and +15px for the 150px hand distance. A significant degradation in MOS is detected between these levels.

The sudden drop occurred between the first and second intervals for the 50px and 100px hand placement distances, which was between +5px to +10px (a gap of 5.4’ to 10.8’ arc minutes between the actor’s hand and the surface of the virtual object). For the 150px hand distance condition this sudden drop occurred between +10px to +15px (a gap of 10.8’ to 16.2’ arc minutes).

To confirm this effect post-hoc analysis is conducted with an alpha of 5%, with the Bonferroni adjustment providing new significance levels of 0.00179 for the 50px hand distance condition and 0.00091 for the 100px and 150px hand distance conditions.

		MDOS (px)							
			Underestimation				Overestimation		
Hand distance	**Statistical Measurements**	**-20:-15**	**-15:-10**	**-10:-5**	**-5:0**	**0:+5**	**+5:+10**	**+10:+15**	**+15:+20**
50px	Z	-3.491	-2.425	-2.066	-1.303	-1.069	-3.972	-4.062	
	p	<0.001	0.015	0.039	0.193	0.285	<0.001	<0.001	
100px	Z	-1.867	-1.886	-2.629	-2.511	-3.116	-3.553	-2.178	-1.807
	p	0.062	0.059	0.009	0.012	0.002	<0.001	0.029	0.071
150px	Z	-2.956	-0.68	-1.272	-1.704	-0.775	-3.274	-3.624	-1.732
	p	0.003	0.497	0.203	0.088	0.439	0.001	<0.001	0.083

Table 8-6. Results from the Pairwise Wilcoxon SR test. Blue Highlight confirms statistically significant degradation in MOS.

The results from post-hoc analysis are provided in Table 8-6, which shows the Wilcoxon Signed Ranks test results between neighbouring MDOS conditions only (statistically significant differences are highlighted in blue). These results confirmed that, as observed, the sudden degradation in MOS was statistically significant between the +5px and +10px for the 50px (Z=-3.972, p=<0.001) and 100px hand distances (Z=-3.553, p=<0.001) and +10px and +15px for the 150px hand distance (Z=-3.624, p=<0.001). The sudden degradation is not observed between the equivalent MDOS conditions for the underestimation (-10:-5 or -15:-10).

8.4.1.3. Conclusion

The initial Hypothesis for this section was:

Static error hypothesis - As the actor's misestimation of the virtual object size becomes more extreme, the observers will be more likely to perceive the error.

The result of the Friedman's test demonstrated that as the magnitude of the MDOS error became more extreme, the observers became more likely to perceive the error. A more sophisticated version of the initial hypothesis would include the heightened ability of the observer's to detect an overestimation of the object size. This modified hypothesis is revised to:

Static Size error - As the actor's misestimation of the virtual object size becomes more extreme, the observers will be more likely to perceive the error. The observers are more adept at detecting an overestimation than an underestimation, tolerating an overestimation until around 10.8' arc-minutes.

8.4.1.4. Comparison with motion analysis results

The results of this perceptual study were compared to the MDOS results measured in chapter 7 to one standard deviation (these results are represented by the vertical blue lines presented in Figure 8-5 and are presented numerically in Table 8-7). It was observed that the 1 standard deviation underestimation yields an MOS between 3.3 (100px) to 4.2 (150px), which would be perceptible but not overly distracting. Consequently the likelihood of actors underestimating the size when interacting with larger virtual objects, as discussed in the previous chapter, was not as important as it initially appeared because the observers are tolerant of this error. This tolerance can be exploited by using large objects to cause the actor to underestimate their size: a less noticeable error.

However, due to the asymmetry of the results if the actor overestimates the size of the virtual object to only one standard deviation from the mean, then the corresponding MOS score is remarkably low, lying between 2.2 (100px) to 2.7 (150px). This outcome demonstrates that under present circumstances the chance of a distracting overestimation occurring is high.

	MOS compared to motion analysis result		
MDOS interval	**Mean-1SD**	**Mean MDOS**	**Mean+1SD**
18.2cm/50px	4.2	4.4	2.3
36.4cm/100px	3.3	4.5	2.2
54.5cm/150px	3.8	4.5	2.7

Table 8-7. MOS values at intervals of the MDOS results recorded in the motion study

8.4.2. Virtual Environment Scene Adaptation

This results presented in this section are ordered in the following manner:

Stage 1: Tweened Adaptation - Adapt the virtual object from an initial size to match the distance between the actor’s hands – A study to compare the effect of adapting the virtual object from an initial size, to the equivalent static error that would be produced if no adaptation occurred. In addition the effects of adaptation speed are also considered which could be an important factor in viewer experience.

Stage 2: Matched Adaptation - Matched adaptation to (actor's) Mean Distance to Object Surface during interaction – A study to investigate the results of continuously adapting the size of the virtual object and to compare these with the results of the equivalent static error with no adaptation.

We believe that it is more valid to represent the two steps of the technique separately, so that the impact that the stages have can be modelled individually. Two size conditions are used for Stage 1 and 2, a 100px and a 150px hand distance; only a 150px hand distance is used for the addendum to stage 1. Should the results presented in this chapter demonstrate that this method provides an effective alternative in comparison to the Static Size results, then the method can be proposed as an effective alternative mode for achieving realistic interaction.

8.4.2.1. Experiment summary

The virtual environment adaptation study explores a proposed solution of adapting the virtual object in in two stages:

Stage 1 - Tweened adaptation: *Object growth from an initial size to a final size that matches the distance between the actor’s hands.*

Stage 2 - Matched adaptation: *The object continuously adapts to the distance between the actor’s hands as the distance varies.*

For this study the following hypotheses will be assessed:

Static error vs. Object Size adaptation - *Observers will find the result of the Object Size Adaptation significantly less plausible than the equivalent static size error, given that a and b are true:*

> a. *The perception of the tweened adaptation to the MDOS is not lower than the corresponding static error.*
>
> *b. The perception of the matched adaptation is higher than the corresponding static error.*

Addendum: Speed of adaptation - *A slower adaptation would be more distracting to the observers, as it provides more time for them to recognise that an adaptation is taking place.*

Tweened adaptation. As an overestimation was the most common type of error found in chapter 7, this study will explore adapting a virtual object to the distance between actor's hands starting from an initial overestimation. The conditions replicated for the tweened adaptation study will be based on the same overestimation conditions as the Static Size study so that a direct comparison can be made, which are presented in Table 8-8. Images from the tweened adaptation video are presented in Figure 8-6.

Hand Distance		**Initial object sizes to be replicated in the perception experiment (Px)**					
		1	2	3	4	6	4σ
100px	Initial Object Size	95	90	85	80	70	40
	Equivalent MDOS	5	10	15	20	30	60
150px	Initial Object Size	145	140	135	130	120	80
	Equivalent MDOS	5	10	15	20	30	70

Table 8-8 The initial object size levels presented for the Tweened adaptation condition.

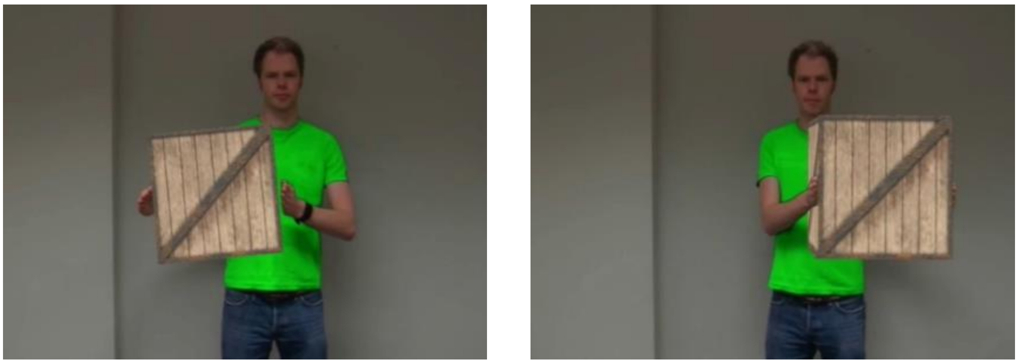


Figure 8-6. Screen captures from Tweened adaptation video, with the left image showing the initial gap and the right image showing the object after it has expanded in size, eliminating the gap.

In addition to this study, the effect that speed of the object growth has on the observer was also investigated, using videos depicting the object growing from an initial 20 px MDOS to a 0px error, over different time periods from an initial 20px MDOS to a 0px error were shown. This study was repeated in the second experiment session with the initial error reduced to a 10px MDOS. The selection of the periods of adaptation is based on the duration of the interaction in the source video, which took place over approximately 1.42s. The different time periods at which this growth occurs for the two experiment sessions are:

First experiment - 0.16s, 0.38s 0.79s, 1.21s, 1.42s
Second Experiment - 0.16s, 0.38s, 0.58s, 0.79s, 1s, 1.21s, 1.42s

Matched adaptation. As discussed in chapter 7, when the actor completes an interaction the distance between the hands will also vary, either becoming further apart or closer together. To stay with the theme of exploring the effects of overestimation, this study explores this effect when the actor's hands drift further apart. With the Matched adaptation videos, when this variation in hand distance occurs, the object changes size accordingly. Images from a Matched adaptation video are presented in Figure 8-7 (page 157).

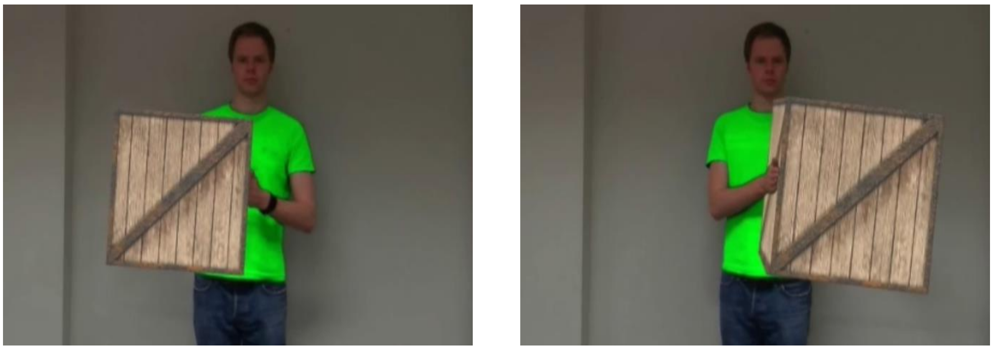
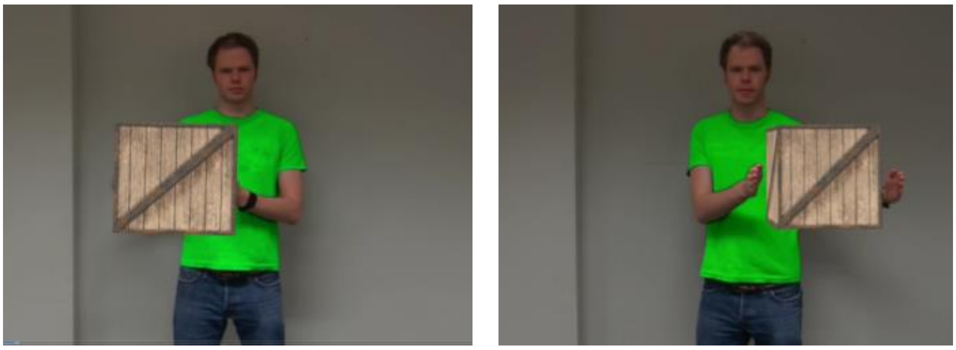


Figure 8-7. Screen captures from Matched adaptation video. The left image shows a 130px object size expanding to 150px in the right image, matching the actor's hands.

Here the actor is correctly estimating the object size at the start of the interaction, as the tweened adaptation is assumed to have happened. When their hands do drift apart the result would be a gap that is similar to an overestimation in the Static Size study, an example of which is shown in Figure 8-8 (page 151). The errors that will be replicated are matched to those used in the Static Size study, so that again a direct comparison can be drawn between them.

To maintain comparability between the Matched adaptation study and the Static Size study, in each video the starting hand distance will be equal to the corresponding object size in the Static Size study, and after the hands have drifted apart the final hand distance will be equal to the distance between the hand distances in the Static Size study.

For example, with the 100px size condition the actor might start off with a distance of 70px between their hands, which throughout the interaction will drift outwards to a resting size of 100px, representing a total growth of 30px; this will be comparable to the equivalent Static Size condition which uses a 100px hand distance and an object size of 70px is, which represents an MDOS error of 30px.



(a) (b)

Figure 8-8. (a) Example of an actor approximately estimating the size of the virtual object at the start of an interaction and (b) the actor's hands after they have moved further apart at the end of the interaction. The figures show the visual result if no adaptation occurs.

Unlike other experiments presented in this chapter where a single source video is used for every condition in each study, each condition presented in this experiment required a separate video to be recorded as the starting hand positions were different each time.

Unfortunately, due to constraints of recording and producing a large amount of individual video segments that satisfied the requirements of the experiment (i.e. maintained consistent speed and pose across videos), only 7 conditions could be presented. The conditions that had videos that satisfied the requirements are presented in Table 8-9.

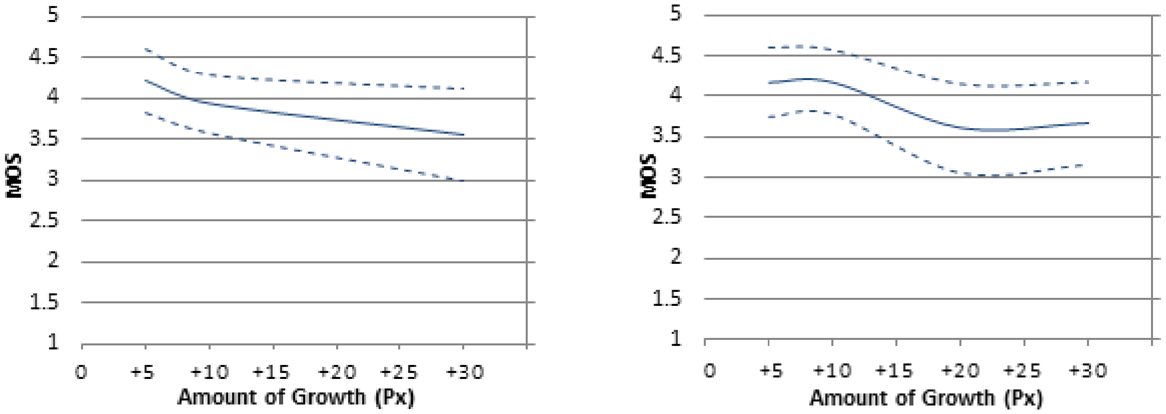
		Matched adaptation sizes to be replicated (Px)			
	Increment	1	2	4	6
100px	Initial Object Size	95	90		70
	Equivalent MDOS	5	10		30
150px	Initial Object Size	145	140	130	120
	Equivalent MDOS	5	10	20	30

Table 8-9. The initial size levels presented for the Matched adaptation condition

8.4.2.2. Stage 1: Tweened Adaptation

8.4.2.2.1. Results

Figure 8-9 presents the results for the adaptation from an initial size to 100px size object (a) and 150px size object (b) studies. In brief, larger MDOS errors resulted in a lower MOS.



Amount of growth (px)	**Total votes**	**Rating value frequency**					**MOS**	**CI**	**Std Dev**
		1	**2**	**3**	**4**	**5**			
+5	21	0	0	7	9	5	4.00	0.33	0.77
+10	21	1	3	9	6	2	3.33	0.43	1.00
+20	21	5	10	6	0	0	2.17	0.32	0.74
+30	21	7	8	5	1	0	2.11	0.38	0.89
+60	21	19	1	1	0	0	1.17	0.20	0.48

(a)

Amount of growth (px)	**Total votes**	**Rating value frequency**					**MOS**	**CI**	**Std Dev**
		1	**2**	**3**	**4**	**5**			
+5	21	1	0	1	6	13	4.44	0.42	0.98
+10	21	0	2	2	5	12	4.44	0.43	1.01
+20	21	1	8	7	4	1	2.94	0.42	0.98
+30	21	3	10	7	1	0	2.33	0.34	0.78
+70	21	11	6	4	0	0	1.67	0.34	0.80

(b)

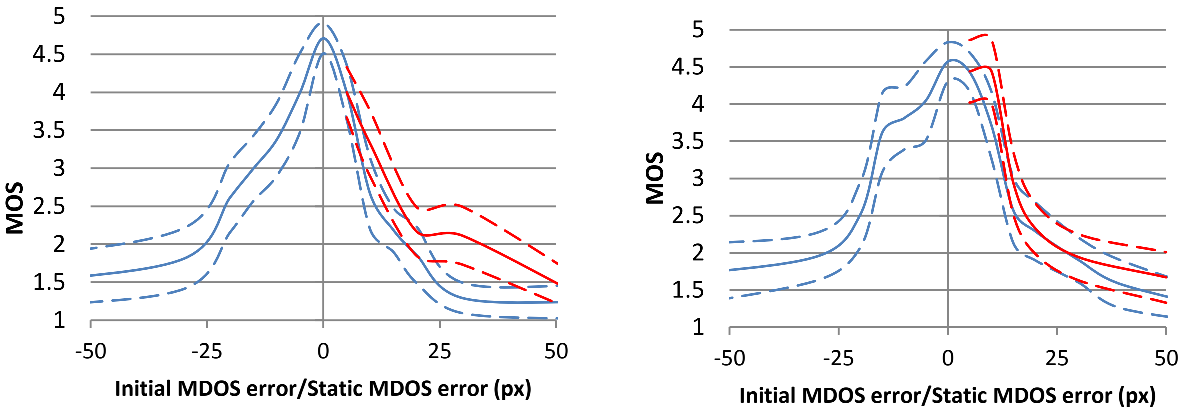
Figure 8-9. Results from the matched adaptation experiment, showing the MOS for (a) the 100px hand distance and (b) the 150px hand distance

8.4.2.2.2. Comparison with corresponding Static Size results

To test the hypothesis, the results were compared with those from the Static Size experiment. Statistical significance was tested for using a pairwise comparison between the corresponding conditions of the static size and tweened adaptation studies (i.e. a gap of 5px in the Static Size results was matched to growth of 5px in the Tweened adaptation results). A Bonferoni adjustment provided a new significance level of 0.0011 for the 100px hand distance condition and 0.00076 for the 150px hand distance condition.

Figure 8-10 presents the comparison between the results of the Static Size and Tweened adaptation studies, alongside the results of the pairwise Wilcoxon Signed Ranks test. For both object size conditions the MOS trend of the tweened adaptation study closely tracked that of the Static Size study and no statistically significant effect ('sig' in table) was measured between any of the corresponding conditions.

In conclusion, there is no significant difference in recorded MOS scores between the corresponding conditions of the Tweened adaptation study and the Static Size study.



Amount of growth (px)	Adapt MOS	Static MOS	Diffe-rence	Wilcoxon SR results Z	p	Sig?
+5	4.00	4.00	0	-0.63	0.527	No
+10	3.33	2.67	0.66	-2.2	0.028	No
+20	2.17	1.86	0.31	-0.92	0.356	No
+30	2.11	1.29	0.82	-2.95	0.003	No
+60	1.17	1.24	0.07	-1	0.317	No

(a)

Amount of growth (px)	Adapt MOS	Static MOS	Diffe-rence	Wilcoxon SR results Z	p	Sig?
+5	4.44	4.43	0.01	-0.44	0.66	No
+10	4.44	3.71	0.73	-2.13	0.034	No
+20	2.94	2.29	0.65	-2.84	0.005	No
+30	2.33	1.90	0.43	-2.14	0.033	No
+40	1.94	1.57	0.37	-1.35	0.177	No
+70	1.67	1.14	0.53	-2.81	0.005	No

(b)

Figure 8-10. Comparison of results between the tweened adaptation and the corresponding static errors for the 100px hand distance

Stage 1: Speed of Tween

Experiment

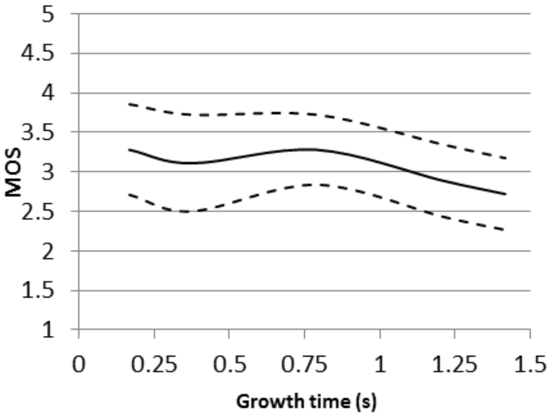
This goal of this study was to investigate the effect that the speed of object growth had on the plausibility of an interaction during the Tweened adaptation. The hypothesis was that a slower adaptation would be more distracting to the viewer, as it would provide more time for them to recognise that an adaptation was taking place. This study used a 150px hand distance size and a starting MDOS of +20px, representing an overestimation. It was repeated in the second experiment session with a +10px MDOS.

Five video segments were produced for the first experiment session using the source video from the 150px hand distance condition and seven were produced for the second experiment session. Each video showed the virtual object adapting over a particular of time.

Results

20px session

The data plotted in Figure 8-11 presents the results for the speed of adaptation. The differences between conditions was small with MOSs ranging between 3.28 (0.16s) and 2.72 (1.42s), a difference of 0.56.



Condition	Total votes	Rating value frequency 1	2	3	4	5	MOS	CI	Std Dev
0.16s	21	3	1	7	5	5	3.28	0.57	1.32
0.38s	21	5	3	4	6	3	3.11	0.61	1.43
0.79s	21	1	4	8	6	2	3.28	0.44	1.03
1.21s	21	3	4	9	4	1	2.89	0.46	1.08
1.42s	21	3	7	5	6	0	2.72	0.46	1.06

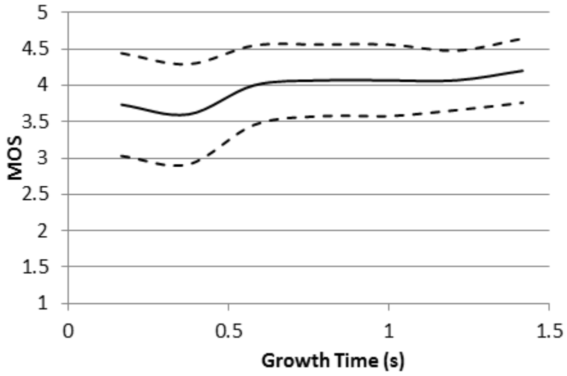
Figure 8-11. MOS scores for adapting the virtual object from an initial MDOS of +20px to a final MDOS of 0px

The result from the Friedman's determined that a statistically significant effect was present (χ^2=9.469, *p*=0.05). As the p-value lies exactly on the pre-determined significance level of 0.05 a post-hoc analysis is conducted. A two-tailed Wilcoxon signed-ranks test was conducted between each pair, with a Bonferroni adjustment providing a new significance level of p<0.005. No statistically significant difference between any conditions was detected.

Although the findings of this study appeared to show that the speed of the tweened adaptation had no significant impact on the plausibility of interaction. It must be noted that as +20px MDOS scored poorly in the static size study, which meant any effects due to speed of adaptation could have been masked by the distracting size of the gap.

10px Session

The data plotted in Figure 8-12 shows that as before the differences between conditions were minimal, with MOSs ranging from 3.6 (0.375s) to 4.2 (1.417s). These results presented a small difference of 0.6 between the two most extreme results, which again suggested that the rate of virtual object growth had no impact on the plausibility of the interaction.



Condition	Total votes	Rating value frequency 1	2	3	4	5	MOS	CI	Std Dev
0.16 s	15	2	0	4	3	6	3.73	0.70	1.39
0.37 s	15	2	1	2	6	4	3.6	0.68	1.35
0.58 s	15	0	2	2	5	6	4	0.54	1.07
0.79 s	15	0	1	3	5	6	4.07	0.49	0.96
1 s	15	0	1	3	5	6	4.07	0.49	0.96
1.21 s	15	0	1	1	9	4	4.07	0.40	0.8
1.42 s	15	0	1	1	7	6	4.2	0.44	0.86

Figure 8-12. MOS scores for adapting the virtual object from an initial MDOS of +10px to a final MDOS of 0px

The results of the Friedman's test (χ^2=1.916, *p*=0.927) demonstrates that no statistically significant result was present when a reduced gap was used.

Suggested Guidelines

As the impact that the rate of growth had on the plausibility of the interaction was not statistically significant, it is suggested that any rate of object growth within the range of the experiment can be used.

Conclusion

The initial Hypothesis for this study was:

A slower adaptation would be more distracting to the observers, as it provides more time for them to recognise that an adaptation is taking place.

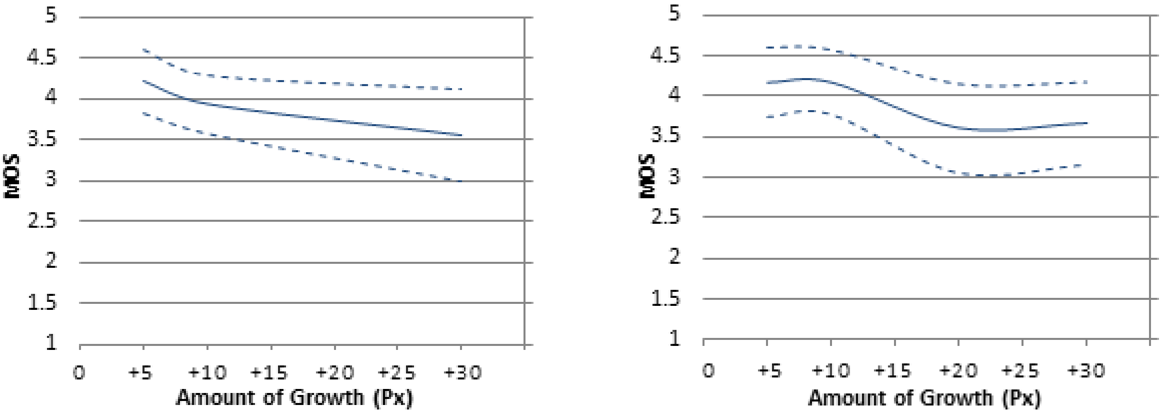
The Friedman test and subsequent post-hoc analysis indicated that there was no statistically significant difference between the adaptation speed conditions for either of the experiments. In addition, no observers commented affirmatively or negatively on the speed of the adaptation. Consequently the hypothesis must be rejected in favour of the alternative hypothesis:

Within the range used in the experiment, the speed of adaptation does not have an impact on the plausibility of an interaction between an actor and a virtual object.

8.4.2.3. Stage 2: Matched adaptation to (actor's) estimation during interaction

8.4.2.3.1. Results

The results of this experiment are shown in Figure 8-13. For the 100px size hand distance (Figure 8-13a), between the +5px and +30px conditions the MOS dropped from 4.22 to 3.56, a fall of 0.66. For the 150px (Figure 8-13b) hand distance study the total MOS degradation was 4.17 to 3.67, a fall of 0.5.



Amount of growth (px)	Total votes	Rating value frequency					MOS	CI	Std Dev
		1	2	3	4	5			
5	21	0	1	3	6	11	4.22	0.39	0.90
10	21	0	1	4	10	6	3.94	0.36	0.84
30	21	2	2	3	6	8	3.56	0.57	1.34

(a)

Amount of growth (px)	Total votes	Rating value frequency					MOS	CI	Std Dev
		1	2	3	4	5			
5	21	0	2	2	6	11	4.17	0.43	1.00
10	21	0	2	1	8	10	4.17	0.40	0.94
20	21	1	4	1	7	8	3.61	0.55	1.29
30	21	1	4	0	11	5	3.67	0.51	1.19

(b)

Figure 8-13. Results from the matched adaptation experiment, showing the MOS for (a) the 100px hand distance and (b) the 150px hand distance

Comparison to corresponding static values

For both size conditions the results appeared to confirm that continuously adapting the size of the object provides a significantly better outcome than the corresponding conditions of the Static Size study.

The 2-tailed Wilcoxon Signed Ranks test using an alpha of 5% was conducted across all conditions, but analysis was focused on corresponding pairs between the Matched adaptation and Static Size results. A Bonferoni adjustment provided a new significance level of 0.0033 for the 100px hand distance and 0.0017 for the 150px hand distance. The results of these experiments and the statistical analysis between the Adapt and the Static MOS are presented in Figure 8-13.

Statistical analysis demonstrates that there is no statistically significant difference between the matched adaptation and static size techniques until the sudden drop that was observed in the Static Size study occurs (+10px for the 100px hand distance and +15px for the 150px hand distance). After this point the results became statistically significant and the Matched adaptation technique consistently outperformed the Static Size technique. The benefit this technique offered was considerable when considering it yielded a 2.27 and 1.77 improvement in MOS for an error of +30px with the 100px and 150px hand distance studies respectively, effectively eliminating the sudden degradation observed in the Static Size study.

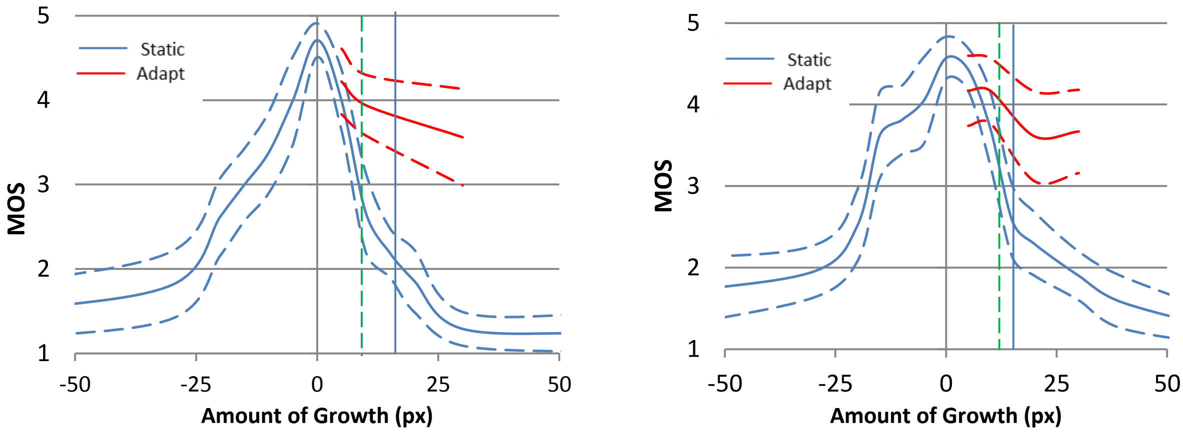
It is important to note that when asked for comments after the presentation 6 observers reported they were aware of the growth of the virtual object. The reason observers scored each video segment highly compared to the corresponding Static Size conditions was that they assumed the actor was moving the virtual object towards the camera slightly, instead of the virtual object growing in size. This natural tendency of humans to attribute an object's change in size to it's moving towards or away from the viewer could be exploited to enhance the plausibility of an interaction.

8.4.2.3.2. Comparison to motion analysis results

The solid blue vertical lines in Figure 8-16 represents the mean MDOS overestimation to one standard deviation from the motion analysis experiment (Chapter 7) for the Interactive Modality AlongAHP for the results of the object size condition (where results for the 36.4cm size condition corresponds to the 100px hand placement condition and the 54.5cm size condition corresponds to the 150px hand placement condition). In both cases adapting the size of the virtual object was shown to mitigate the error effectively at this point. The adaptation technique shows ~1.4 MOS improvement for the 100px hand distance size and ~0.8 MOS for the 150px hand distance size.

The dashed green lines represent 1 standard deviation from the mean for the amount of variability between the actor's hands (VDBH). Again it was possible to see that adapting the size of the virtual had a positive effect at this point. The matched adaptation allowed an MOS improvement of ~1.2 for the 100px hand distance and ~0.7 for the 150px hand distance.

Due to the viewer's improved tolerance of the overestimation, it was possible to determine that the actor's misestimations could be compensated for by continuously adjusting the size of the virtual object.



Amount of growth (px)	Adapt MOS	Static MOS	Diffe-rence	Wilcoxon SR results Z	p	Sig?
5	4.22	4.00	0.22	-1.015	0.31	No
10	3.94	2.67	1.27	-3.091	0.002	Yes
30	3.56	1.29	2.27	-3.684	<0.001	Yes

(a)

Amount of growth (px)	Adapt MOS	Static MOS	Diffe-rence	Wilcoxon SR results Z	p	Sig?
5	4.17	4.43	-0.26	-0.655	0.512	No
10	4.17	3.71	0.46	-1.573	0.116	No
20	3.61	2.29	1.32	-3.573	0.001	Yes
30	3.67	1.9	1.77	-3.593	< 0.001	Yes

(b)

Figure 8-14. Comparison of results between the matched adaptation and the equivalent static errors for the (a) 100px hand distance and (b) the 150px hand distance

8.4.2.4. Conclusion

The initial hypothesis was:

Static error vs. Object Size adaptation hypothesis - Observers will find the result of Object Size Adaptation significantly less distracting than the equivalent static error, given that:

> *The plausibility of the tweened adaptation is not lower than the corresponding static error, and*
>
> *The plausibility of the matched adaptation is higher than the corresponding static error.*

A series of Wilcoxon Signed Ranks tests confirmed that no statistically significant difference was detected between the Tweened adaptation and the Matched adaptation. Therefore, the first section of the hypothesis has met its requirement.

Another series of Wilcoxon Signed Ranks tests confirmed that the Matched adaptation was either as good as, or better than, the equivalent Static Size error (depending on the size of the error/growth), resulting in a significantly improved plausibility of the interaction. Therefore, the second section of the hypothesis has met its requirement. As both requirements have been met, the original hypothesis is deemed to be correct.

8.4.3. Real Scene Adaptation

8.4.3.1. Experiment summary

The real scene adaptation study explores a solution where an element of the real environment is manipulated, in this case the actor's shirt seen through the gap created by an overestimation (henceforth referred to as 'background colour' for generalization purposes). The shirt's colour is a bright green, which contrasted greatly with the colour of the box and the actor's hands. The decision was made to investigate the effect that background colour has on the perception of an MDOS error during interaction, as using certain background colours could be exploited to reduce the visual impact of the estimation error if an effect is found. Using a video from the Static Size condition (150px hand distance, +20px MDOS) a range of videos were created where the colour of the actor's shirt was digitally altered. This study is repeated in the second experiment session using a 150px hand distance with a +20px MDOS.

For this study the following hypothesis will be assessed:

Background colour - *The observers are less likely to perceive an overestimation by the actor if the background colour contrasts strongly with the skin colour of the actor.*

Six different shirt colours were used, representing the background colour. In RGB (Red, Green, Blue) they were: Green (0, 255, 0), Black (0, 0, 0), Blue (0, 0, 255), Red (255, 0, 0), the actor's skin tone (134, 89, 74) and White (255, 255, 255). The colours were selected to cover a range of matching and contrasting conditions. Samples of some of the colours used are presented in Figure 8-15 (page 165).

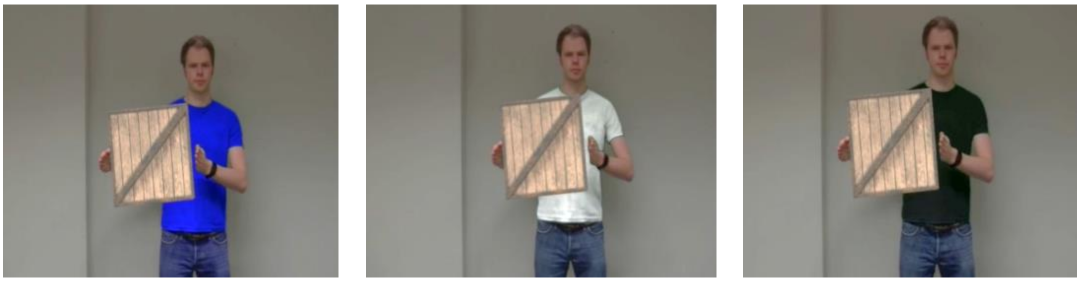


Figure 8-15. Selection of screenshots from Background Colour videos

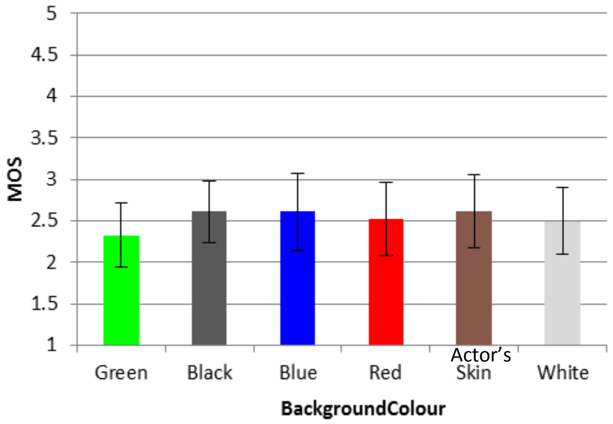
8.4.3.2. Results

20px session

The results presented in Figure 8-16 showed that there was little difference between each condition. This suggests that background colour had no statistically significant effect on the interaction plausibility.

A Friedman's test was conducted using an alpha of 5% to determine whether there was any statistical significance among the results. The result χ^2=3.958, *p*=0.556 confirmed that background colour had no statistically significant effect on the plausibility of interaction.

Although the findings of this experiment appeared to show that background colour had no significant impact on plausibility of the interaction, the results were potentially inconclusive due to the distracting size of the MDOS error, as previously discussed. This meant any impact background colour may have had could have been masked by the size of the gap.



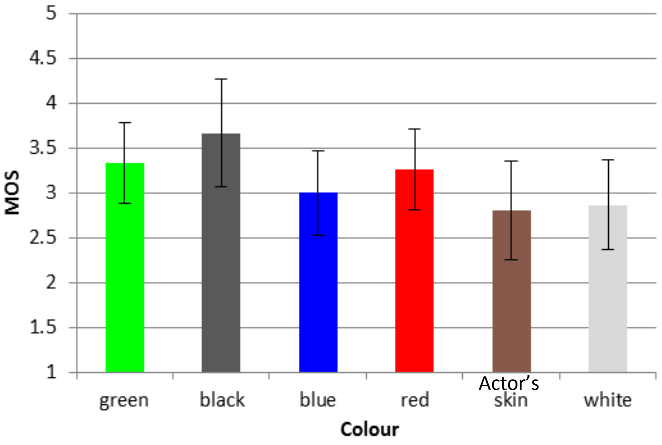
Condition	Total votes	Rating value frequency					MOS	CI	Std Dev
		1	2	3	4	5			
Green	21	3	12	3	3	0	2.33	0.39	0.90
Black	21	2	8	8	3	0	2.61	0.37	0.87
Blue	21	2	10	4	4	1	2.61	0.46	1.07
Red	21	4	6	7	4	0	2.52	0.44	1.03
Skin	21	4	9	4	4	0	2.61	0.44	1.02
White	21	3	8	7	3	0	2.50	0.40	0.93

Figure 8-16. The effect of background colour on the plausibility of interaction: an overestimation of 20px. The MOS for each condition shows that background colour has no significant effect on the plausibility of interaction

10px session

The data plotted in Figure 8-17 (page 166) showed that as before there was little difference between each condition.

The results of the Friedman's test (χ^2=13.256, p=0.021) revealed that a statistical significant result is present. Post hoc-analysis was conducted using a Wilcoxon Signed Ranks test, with a Bonferroni adjustment providing a new significance threshold of 0.0023. No statistically significant effects were detected between any of the conditions.



Condition	Total votes	Rating value frequency 1	2	3	4	5	MOS	CI	Std Dev
Green	15	0	3	5	6	1	3.33	0.46	0.9
Black	15	1	1	4	5	4	3.67	0.59	1.18
Blue	15	1	3	6	5	0	3	0.47	0.93
Red	15	1	0	9	4	1	3.27	0.45	0.88
Skin	15	2	4	4	5	0	2.8	0.55	1.08
White	15	1	5	4	5	0	2.87	0.50	0.99

Figure 8-17. The effect of background colour on the plausibility of interaction: 150px hand distancewith an MDOS of +10px

8.4.3.3. Conclusion

The initial hypothesis for this section was:

The observers are less likely to perceive an overestimation by the actor if the background colour contrasts greatly against the skin colour of the actor

No statistically significant difference was discovered in the Friedman's test between any of the background colour conditions for the first experiment (p=0.561). However, when a reduced gap was used a statistically significant effect was detected (p=0.021), but no effect could be confirmed with post-hoc analysis. As no observers commented affirmatively or negatively on the background colour conditions either, there was no evidence to show that background colour affects the perception of the MDOS error. Therefore the initial hypothesis must be rejected in favour of an alternative hypothesis:

Background colour does not have an impact on the perception of an MDOS error between an actor and a virtual object.

8.4.4. Occlusion Study

8.4.4.1. Experiment summary

This study explores the effects of switching the occlusive properties of the scene during an interaction/occlusion event to determine whether it has an impact on the perceived quality of the scene, as reported on in the case study (chapter 3). This essentially tests the validity that feature preventing occlusion switching built into the occlusion system (chapter 5). For this study the following hypothesis will be assessed:

Switching Occlusion – *Observers will notice when the occlusion properties of the scene change during an interaction.*

Each video segment presented to the observers for scoring used a single source video, using a 150px hand distance size condition with an MDOS of 0px. Midway through the interaction the occlusive properties between the actor's hands and the virtual object suddenly change. The two videos presented showed the hands moving from behind (B) to in front (F) of the virtual object and vice versa.

To allow the videos to be presented incognito only the occlusion properties of the hands will change, so they appear similar to the other conditions. An example from one of the videos is presented in Figure 8-18, where the actor's hands change from appearing F of the object to B. The three videos that were presented are:

- A video depicting correct occlusion properties throughout the video.
- A video where the actor's hand suddenly changed from appearing in front of the object to behind the object.
- A video where the actor's hand suddenly changed from appearing behind the object to in front of the object.

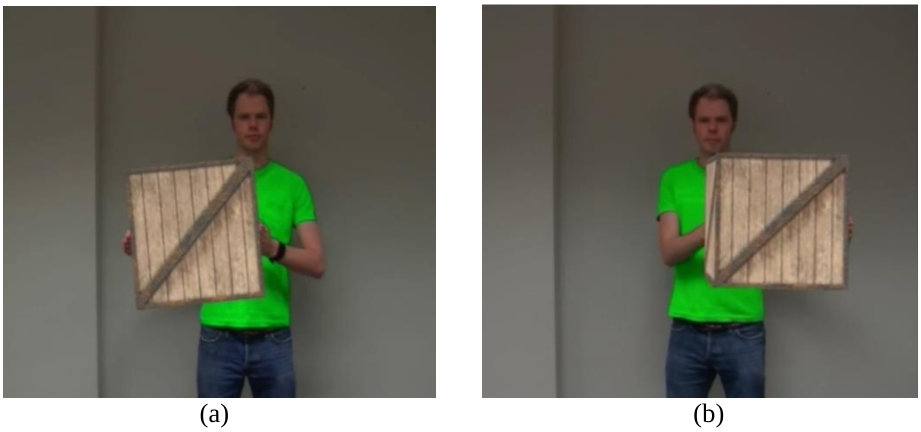
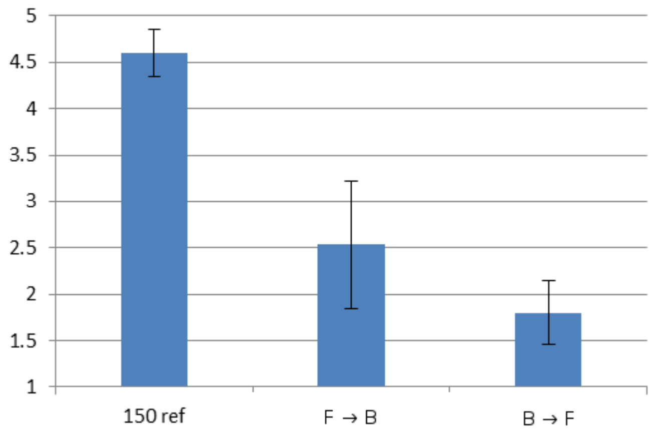

(a) (b)

Figure 8-18. Screenshots from video where the actor's hands switch from (a) in front of the virtual object to (b) behind the virtual object

8.4.4.2. Results

Figure 8-19 presents the results that switching occlusion during an occlusion event/interaction had on the interaction plausibility. As shown, if the occlusion between the actor and the virtual object changed mid-interaction the impact that it had on the viewer was devastating, as the MOS drops significantly. A Wilcoxon Signed Ranks was conducted to confirm whether this effect was statistically significant with a Bonferroni adjustment providing a significance threshold of 0.0167.

The reference video sequence scored an MOS of 4.6, which was 2.07 higher than the F→B condition (Z=-3.134, *p*=0.002) and 2.8 higher than the B→F condition (Z=-3.443, *p*=0.001), demonstrating that switching occlusion properties of the scene during an interaction had a negative effect on the quality of the interaction.



Condition	Total votes	Rating value frequency 1	2	3	4	5	MOS	CI	Std Dev
150 reference	15	0	0	0	6	9	4.60	0.26	0.51
In front to behind (F⟶B)	15	3	7	1	2	2	2.53	0.69	1.36
Behind to in front (B⟶F)	15	5	8	2	0	0	1.80	0.34	0.68

Figure 8-19. Results of changing occlusion

8.4.4.3. Conclusion

The initial hypothesis for this experiment was:

Observers will notice when the occlusion properties of the scene changed during an interaction.

The results from the Wilcoxon Signed Ranks test demonstrated that the switching of occlusion properties of the scene were perceptible and had a statistically significant negative effect. Consequently, the initial hypothesis is correct.

8.4.5. Post Experiment Interview

The following comments were made during the post-experiment interview across both experiment sessions, where the experiment was explained to the observer and they were asked for any observations they had:

- 2 observers made comments about whether a noticeable misestimation is forgivable or unforgivable:
    - One observer from a Chinese background noted that in Chinese cinema an object floating between an actor's hands (the visual consequence of an overestimation by the actor) is a common special effect and found it forgivable as a result.
    - Conversely, one observer claimed they would have felt "short-changed" if they had seen such a poor interaction on television.
- The matched adaptation of the virtual object was noticed by 6 observers, who each mistook it for the scaling one would experience when the actor moves an object closer to the camera. This lead to the conclusion: *The perceived high plausibility of the matched adaptation is due to the observer believing the object is not growing, but scaling due to actor moving it closer to the camera.*
- 3 observers note that plausibility of the interaction was more difficult to judge when the object was in motion, than when it was relatively static (before and after the movement of the object) and so these portions of the video had a salient impact on their awarded scores.
- 3 observers claimed they had spotted multiple repetitions of certain videos, when in reality only the stabilizing videos would have been repeated later in the main test. This indicates that although they had scored interactions with similar conditions differently (i.e. videos depicting a +5px and a +10px MDOS), in many cases they were not aware that a different error had been presented to them.
- 4 observers reported that they became more forgiving after they had seen the first few videos (potentially within the stabilising set). This could indicate that once the observer has viewed several interactions they may be willing to accept that errors will occur in an interactive virtual studio and become more tolerant as a result.
- 1 observer said the box looked too light and the ease at which the actor moved the object was the most unrealistic element of video.

8.5. Conclusions

Presented in this chapter were the results of a Single Stimulus experiment that explored the viewer perception of interaction realism between actors and virtual objects in an interactive virtual studio. The aim was to identify how well an audience could perceive the errors caused by the actors and build a results base against which solutions to mitigate the effect of these errors can be tested. This process was demonstrated by testing a proposed solution where the virtual object was adapted to the distance between the actor’s hands in two steps. The results showed that this solution was more capable of creating a plausible interaction than if the object size remained static, thus justifying this method. The level of occlusion sophistication and the colour of the background were also examined to determine whether they had a significant impact on the perception of an interaction.

A summary of the hypotheses generated from the results section is presented:

1. *Static Size error - As the actor's misestimation of the virtual object size becomes more extreme, the observers will be more likely to perceive the error. The observers are more adept at detecting an overestimation than an underestimation, only tolerating an overestimation until around 10.8’ arc-minutes.*
2. *Static error vs. Object Size adaptation hypothesis - Observers will find the result of Object Size Adaptation significantly less distracting than the equivalent static error, given that:*
    a. *The plausibility of the tweened adaptation is not lower than the corresponding static error, and*
    b. *The plausibility of the matched adaptation is higher than the corresponding static error.*
3. *The speed of adaptation does not have an impact on the plausibility of an interaction between an actor and a virtual object.*
4. *Background colour does not have an impact on the perception of an MDOS error between an actor and a virtual object.*
5. *Observers will notice when the occlusion properties of the scene changed during an interaction.*

From the results gained in the experiment the following findings are summarised:

- The observers are more adept at identifying an overestimation of the object size by the actor than an underestimation. The observers become aware of gaps at around an overestimation of +10px MDOS.
- Adapting an object from an initial size to fit the distance between the actor’s hands yields approximately the same MOS as when size of the object remains static.
- The rate at which an object grows from an initial size does not have an impact on the plausibility of the interaction.
- Matched adaptation of the virtual object to match the varying distance between the actor’s hands yields superior plausibility to a situation where the size remained static. This is probably due to the belief of the observer that the object is not actually growing, but moving towards them.
- The background colour of the scene that appears in the gaps between the actor’s hands and the virtual object surface does not affect the plausibility of the interaction.
- Switching the occlusion properties of the scene is highly distracting for the observers.

From these trends the following set of guidelines can be populated, which suggests how the perception of interaction can be improved when an error occurs. The current suggested guidelines are:

- Overestimation of the object by the actor of >+10px is considerably more likely to result in a perceptible error; therefore, all other factors being equal, underestimation of the object is preferable as it is more likely to result in a more plausible interaction.
- The Object Size Adaptation solution where the object's size is adapted should be used in preference to a static-size object due to the following conclusions.
    - Adapting the virtual object from an initial size to the MDOS is approximately as distracting as leaving the error.
    - The object can be adapted from an initial size to the MDOS at any speed with negligible impact.
    - Continuously scaling the virtual object to the MDOS as it varies produces a significantly more plausible interaction than the static error
- The occlusion properties of the scene must remain consistent throughout. Switching occlusion has a devastating effect on the plausibility of the interaction and the scene in general.

With regards to the inter-relation of the results for the motion experiment and the perception experiment, the following observations were made:

- A noted effect from the results of the motion experiment in Chapter 7 was that the likelihood of the actor overestimating the object size did not appear to increase with larger objects, but the likelihood of them underestimating the object size did. When compared to the perception results, it was found that the viewers were particularly tolerant of the actor underestimating the virtual object. As such, the initial error of the actor tending towards the underestimation of the virtual object is not as significant as originally thought, as this error is less likely to be noticed.
- The actors were likely to have some variation between their hands during an interaction, which would lead to a misestimation in the object's size. This effect was particularly noticeable when the actors placed their hands in line with the axis of motion. This chapter demonstrated that adapting the size of the object using the two-stage method was powerful in mitigating the effect that variability between hands could have, as the observers were tolerant of the object changing in size.

Application of these guidelines can aid both actors and content producers. Actors could be advised to bias towards object underestimation, as they tend to overestimate the size of (smaller) virtual objects and an underestimation error was found to be less perceptible than an equivalent overestimation. For content producers, in the case of an overestimation error, using a virtual object with an adaptable size continuously matching the distance between the actor's hands would be a demosntrably effective technique for mitigating the impact of the error.

These conclusions demonstrate that measuring the motion of the actor alone is insufficient when trying to improve the realism of an interaction, as the data from a perceptual experiment changes how the outcomes of the motion data are viewed.

The success of this framework lies in its ability to highlight the impacting conditions and its ability to assess techniques to improve interaction plausibility. This framework would be beneficial for research studying fidelity in MR or for content developers looking to create a more plausible MR scene. Future research could be applied to the assessment of other techniques, for example NPR (Haller, 2004) or the addition of lighting effects (Steptoe, et al., 2014).

Due to the effectiveness of this methodology in highlighting the conditions affecting motion and allowing the assessment of solutions, it may also be transferable for assessing realism in similar interactive Mixed Reality or Special Effects environments.

Summary of Chapter

This chapter presented the second step of the framework where the perception of actor errors identified throughout chapter 7 was assessed. Using MOS the ability of a home viewer to detect any errors was measured, alongside which videos depicting solutions designed to mitigate the error were also presented and tested. It was discovered that viewers are less likely to perceive underestimations than overestimations and that overestimations could be mitigated by adapting the size of the virtual object. These results inform on how errors produced using an interaction system similar to the one presented in chapter 6, using the occlusion technique described in chapter 5 and an analogue to the feedback system described in chapter 4, are perceived.

Chapter 9 : CONCLUSION

In this work an interaction system for the virtual studio was developed followed by the creation of a framework for analysing the impact of actor motion errors on interaction plausibility in a virtual studio or similar mixed reality environment.

9.1. Review of Research

9.1.1. Development of Interaction

The first portion of this work was the development of three elements using heuristic analysis to guide their design. These three elements were: visual feedback using gaze directed projection, occlusion systems compatible with a standard virtual studio and interaction based on skeletal tracking data. To support these developments a depth-sensitive camera was used to compute the location of 15 skeletal points of the actor using OpenNI, which was identified as being the most compatible and sufficient method for this study. These were implemented into an adapted virtual studio architecture. The following is a summary of the three elements:

Directed Visual Feedback

A gaze-directed projector for use on planar surfaces in the virtual studio was presented, which provided a real-time visual representation of the virtual set at the gaze point of the actor in either first or third person perspectives. This system was constructed using a standard projector with a servo-guided mirror mounted in front of the lens and a camera for head tracking. The projected image was corrected in real time for warping caused by the keystone effect.

Layer-based Occlusion

An occlusion system compatible with the standard layer-based design of virtual studios was presented. This system was able to replicate three common forms of occlusion that would occur, namely: absolute occlusion, object intersection occlusion and actor intersection occlusion. These occlusions were based on the skeletal motion capture of the actor, where the closest skeletal point to each virtual object was used to define the occlusion properties of the scene. While the actor appeared either directly in front of or behind a virtual object, the object was locked to the layer it appeared on so that a sudden change in occlusion would not occur. Subsequent perceptual tests (chapter 8) tests did confirm that locking the occlusion was a vital component of an occlusion system, giving improved perceptual performance.

Interaction

A method for direct single-handed and bimanual interaction between the actor and objects in the virtual set was presented. The virtual object was manipulated using the motion capture data of the actor, where for a bimanual interaction the location of the virtual object was matched to the midpoint of the actor's hands when they were detected close to its surface. The object could be moved with full orientation and location and were released once the actor’s hands were detected far from the surface. The interaction could be simplified in a number of ways to make the interaction task easier.

9.1.2. Assessment of Interaction Quality

The second portion of this work presented a framework for examining what errors actors make when attempting to estimate the size of the virtual object during a bimanual interaction and how these errors impact on the plausibility of the interaction from the perspective of the viewer. This work contained two elements: The first was an analysis of actor motion error during interaction and the second was a study to find the viewer's perception of the errors.

Analysis of Actor Motion during Interaction

This stage was a method for measuring the magnitude of errors made by an actor during the bimanual manipulation of a virtual object and a study of the impacting conditions. 16 actors completed 168 interaction tasks each under varying conditions, which were Size of Object, Speed of interaction and Hand position with relation to object motion. Their motion was measured using two novel performance metrics which are unique to this work: Mean Distance to Object Surface (MDOS) and Variability in Distance Between Hands (VDBH). The experiment also used three types of object modality: Animated, where the object followed a predefined path that the actor had to follow; Interactive, where the object was fixed to the midpoint of the actor’s hands; and no-object, where no-object was present and the actor mimed the intended interaction.

From the results of this study the following conclusions were made:

- The size of the virtual object had a significant effect on the error created by the actor’s estimation. Larger object sizes resulted in decreased assessment of the object size (MDOS) and an increased amount of variance between the hands (VDBH). It was also demonstrated that the actor tends to underestimate larger virtual objects, but not overestimate them.
- The speed at which the interaction occurred had no impact on the actor’s MDOS or VDBH.
- Hand placement in relation to the axis of object motion had a significant effect on the variation in distance between the actor’s hands. It was demonstrated that if the actor positioned their hands *with* the axis of motion it lead to an increased amount of VDBH.
- The Interactive modality yielded comparatively similar results for MDOS and VDBH for each condition as the Animated modality, but did not have a "lag/lead" error associated with it. Consequently, the Interactive modality is better for creating live interaction.
- The Interactive modality yielded better results than the no-object modality, where the actor was able to perform with a lower MDOS and VDBH along the axis of hand placement.

Viewer Perception of Actor Errors

A method for measuring the ability of an audience to perceive the MDOS errors created by the actor was presented. To achieve this, the range of errors recorded for each object size for the Interactive modality were replicated in a series of videos that were shown to 21 observers who rated how well they were able to identify the errors. The observers measured the videos using the performance metric Mean Opinion Score (MOS), which used an adjective-based impairment scale. This introduces a new application for the of the ITU-R standards showing their applicability to Mixed Reality performance assessment.

It was also demonstrated that this method could be used to identify whether manipulations made to the scene would have an effect on the perception of errors. Virtual environment adaptations were tested, where videos were presented throughout the experiment in which the size of the virtual object was altered during the interaction to match the distance between the actor’s hands, removing any estimation errors. Real scene adaptations were tested in the form of videos that depicted different colours in the gap between the actor’s hands and the object surface.

This method was also used to test the suitability of the occlusion system presented in chapter 5, where videos depicting idealised occlusions and those made possible by the occlusion system were tested; as well as two videos that depict an occlusion changing partway through the interaction. From the results of this work the following conclusions were made:

- The viewer was particularly aware of when the actor overestimated virtual object size, where even a small gap was perceptible to them. They were more tolerant of when actor underestimated the size of the virtual object.
- Continuously adapting the size of the virtual object to the varying distance between the actor’s hands throughout the interaction produced a considerably better quality interaction than if the virtual object size had remained static. Therefore adapting the object size is a more effective way of achieving realistic interaction.
    - Post-experiment interviews indicated that the observers were aware of the change in object size, but assumed that it was because the actor's was moving the object towards the camera.
- The colour of the background exposed to the viewer in the gap between the hands of the actor and the surface of the virtual object had no effect on the perceived quality of the interaction.
- Changing the occlusion properties of a scene during an "occlusion event" was distracting to the observers.

These results also demonstrated that interaction errors cannot be assessed solely on the measurement of the actor's motion. This was because the perceptual experiment results illustrated that the increased likelihood of the actor to underestimate the size of larger virtual objects was not actually a detriment as originally believed and could even be beneficial, as the viewer was particularly tolerant of this type of error and it allowed a lower MDOS.

It was also possible to determine that adapting the interactive virtual studio system to so that the size of the virtual object changed to match the distance between the actor's hand was beneficial for improving interaction realism, which should be a feature implemented into future virtual studio systems.

This work has presented an effective framework for designing realistic interaction in a virtual studio environment using heuristic and analytical methods. Results from this have demonstrated how both the reliability of interaction system and the actor performance are both vital elements in the design of an interactive virtual studio. In addition, the end-user QoE can also be used to inform the design, with measurements of perceptual realism applied in solutions for creating improved actor-virtual object interaction.

9.2. Future work

The following are proposed as future directions for further development of an interactive virtual studio.

Re-examination of Case Study and Heuristic Analysis

In chapter 3 a case study was presented where interactions were practiced in the virtual studio and analysed to create a set of heuristic requirements for future technical developments towards achieving interaction. With future developments made to meet the requirements of these heuristics it would be beneficial for future researchers to review then, either designing their own extension to these or reject existing ones that they find to conflict with the new information gained from their further analysis.

One example would be the need for an occlusion system to be compatible with the existing layer based format of most virtual studios. In retrospect, this heuristic could act as an artificial barrier that limits the effective use of modern technologies, such as Z-mixing of real and virtual scenes using a depth sensitive camera where layers aren't necessary to achieve a convincing level of occlusion. If new information finds a compelling argument for this to be the case, then that heuristic should be rejected.

Extension of ScaMP to multiple planar surfaces

In chapter 4 ScaMP was demonstrated as being able to provide ever-present feedback to an actor on a single planar surface. To realise the potential of this system, future work should focus on extending it to the multiple planar surfaces of the virtual studio. A proposal for this is presented in chapter 4.3.4 followed by a more detailed method in appendix #C.

Solution to the Painter’s problem like issue for the occlusion system

One of the limitations with the occlusion system presented in chapter 5 was that the actor cannot both occlude and be occluded by the same virtual object (e.g. they cannot stand behind a virtual object and place their hand in front of it). In Appendix #D a proof of concept system that could achieve this and would be compatible with a layer-based virtual studio system is presented. Here the issue is solved using a subtractive ray casting technique. In this proposed system the motion of the actor is relayed to an invisible 3D “avatar” in the virtual set, which is used alongside ray casting to identify which portions of a virtual object are occluded from the perspective of the studio camera. The portion of the virtual object occluded by the avatar is rendered as transparent, creating in a “hole”. The object is presented on a foreground layer, with the hole exposing the occluding portion of the actor on the video layer.

Object size adaptations in Bimanual Interaction

From the results presented in chapter 8 it was demonstrated that adapting the size of the virtual object to match the distance between the actor’s hands was an effective way of mitigating the viewer’s perception of motion errors. This feature should be implemented into future interactive virtual studio designs as it improves the quality of the interaction as seen from the perspective of the audience.

Design of a breakaway procedure for object interactions

Current research is exploring a breakaway procedure for interactions, so that actors are able to end an interaction with a virtual object in a naturally appearing manne. As the object is locked to the centroid of the actor’s hands while they are sufficiently close, the actor will not currently be able to break the interaction unless they move their hands far away from the object surface. More practical and realistic breakaway procedures could be found in either gesture control (we are currently implementing a “fist” gesture to stop all interactions) or velocity control (where the actor moves their hands apart above a certain velocity away from an object).

Extension of the MDOS and VDBH metrics

The MDOS and VDBH metrics allow hand placement around an object’s surfaces to be analysed in a broad way in a 2D setting. In this thesis many conclusions about actor hand placement could be made using these metrics. However, there is room for extending these metrics to make further conclusions on the accuracy of actor hand placement. It is anticipated that expending these metrics will reduce the limitations of the framework. Proposed directions for expansion of these metrics are:

- Separation of distance from hands to surface for MDOS, measuring the distance of the left hand to the relevant left surface and right hand to the relevant right surface individually.
- Extension into 3D. The current 2D analysis provides detailed information about the how hand placement appears projected onto a 2D screen (i.e. how it appears to a home viewer), but 3D metrics could provide details on the actor’s interaction itself.
- Diagonal extension. The current equations are not optimised for measuring diagonal errors.

Further Analysis on how Viewers Perceive Actor Errors

In chapter 8 it was discovered that observers were aware of the virtual object changing in size, but also that they assumed this was because the actor was moving the virtual object towards the camera. This demonstrated that the viewer perception study was capable of identifying psychological factors that could be exploited to improve the apparent quality of interaction. Future work should use the methods demonstrated in this work to explore other types of manipulation that could be made to the scene in order to exploit these psychological factors in ways that positively impact the observers' experience, such as showing busy scenes that provide distractions. It is also believed that these methods could also be applied to more general Mixed Reality applications.

Some factors related to plausibility that are recommended for further explorations are:

- Objects of different shapes (spherical, complex, etc)
- Objects of different transparencies.
- Objects with different textures.
- Different hands placements to H-LR (e.g. those used in chapter 7)
- Interactions along vertical or horizontal meridians.

9.3. Summary of Research

This research presented in this thesis has focused on the creation of frameworks that contribute towards the development of an interactive virtual studio, which has yielded significant contributions to knowledge. These frameworks guided the development of an interactive virtual studio through heuristic means and refined the plausibility of interactions that take place. These frameworks were successfully implemented to an existing virtual studio infrastructure and can continue to be used to further improve the quality of an interactive virtual studio and potentially other interactive AR systems, which has led to recognition from the wider AR community.

REFERENCES

Allard, J. et al., 2007. Grimage: Markerless 3D Interactions. *ACM SIGGRAPH,* Volume Article 9.

Appel, A., 1968. Some Techniques for Shading Machine Renderings of Solids. *ACM spring joint computer conference,* pp. 37-45.

Ascention, 2010. *Ascension 3D Guidance trakSTAR.* [Online]
Available at:
http://www.inition.co.uk/inition/product.php?URL_=product_mocaptrack_ascension_trakstar&SubCatID_=18 [Accessed 08 06 2011].

Atkeson, C. & Hollerbach, J., 1985. Kinematic Features of Unrestrained Vertical Arm Movement. *Journal of Neuroscience,* September, 5(9), pp. 2318-2330.

Baird, J. L., 1929. *Apparatus for transmitting views or images to a distance.* U.S., Patent No. 1,699,270.

Blender Foundation, 2011. *Blender.* [Online]
Available at: http://www.blender.org [Accessed 12 11 2011].

Blonde, L. et al., 1996. A virtual studio for Live Broadcasting: The Mona Lisa Project. *IEEE MultiMedia,* 3(2), pp. 18-29.

Bowman, D. & Hodges, L., 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *ACM Proceedings of the 1997 symposium on Interactive 3D graphics,* pp. 35-38.

Bowman, D., Johnson, D. & Hodges, L., 1999. Testbed Evaluation of Virtual Environment Interaction. *Proceedings of the ACM symposium on virtual reality software and technology,* pp. 26-33.

Brooks, A. & Czarowicz, A., 2012. Markerless motion tracking: MS Kinect and Organic Motion OpenStage. *Proc. 9th Intl Conf. Disability, Virtual Reality & Associated Technologies,* pp. 435-437.

Chandaria, J. et al., 2007. *Real-Time Camera Tracking in the MATRIS project,* London: British Broadcasting Corporation.

Cruz-Neira, C. et al., 1992. The CAVE: Audio Visual Experience Automatic Virtual Environment. *Communications of the ACM,* 35(6), pp. 64-72.

Cutting, J. E., 1997. How the eye measures reality and virtual reality. *Behavior Research Methods, Instruments, & Computers,* 29(1), pp. 27-36.

Daemen, J., Haufs-Brusberg, P. & Herder, J., 2013. Markerless Actor Tracking for Virtual (TV) Studio Applications. *International Joint Conference on Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA),* pp. 790-796.

Daemen, J., Haufs-Brusberg, P. & Herder, J., 2013. *VRON: Virtual Reality over Networks.* [Online]
Available at: http://vsvr.medien.fh-duesseldorf.de/productions/vron2013/
[Accessed 23 05 2014].

Deshpande, S., 2009. A method for synchronization mismatch perception evaluation for large ultra high resolution tiled displays. *International Workshop on Quality of Multimedia Experience,* pp. 238-243.

Digital Broadcasting, 2001. *Orad Reveals an Array of Technological Innovations for Broadcasters.* [Online]
Available at: http://www.digitalbroadcasting.com/doc/orad-reveals-an-array-of-technological-innova-0001 [Accessed 29 07 2014].

Digital Rune, 2011. *Project FineSkills - Physically-Based Interaction in Virtual Reality.* [Online]
Available at: http://www.digitalrune.com/Misc/FineSkills/tabid/356/Default.aspx [online] [Accessed 12 08 2011].

ECE, University of Texas at Austin, 2011. [Online]
Available at: http://www.bu.edu/ece/2011/04/26/itec-competition

Edison, T., 1891. *Kinetographic camera.* U.S., Patent No. US589168 A.

Ehnes, J., 2010. A Precise Controllable Projection System for Projected Virtual Charachters and its calibration. *International Symposium on Mixed and Augmented Reality,* 9(1), pp. 221-222.

Ehnes, J. & Hirose, M., 2006. Projected reality – content delivery right onto objects of daily life. *The International Journal of Virtual Reality,* 5(3), pp. 17-23.

Ehnes, J., Hirota, K. & Hirose, M., 2004. Projected Augmentation - Augmented Reality using Rotatable Video Projectors. *IEEE and ACM International Symposium on Mixed and Augmented Reality,* Volume 3, pp. 26-35.

Elson, B., Athwal, C. & Reynolds, P., 2009. Creating the World of Augmented Dental Training. *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education,* pp. 2547-25.

Farnsworth, P. T., 1927. *Television system.* U.S., Patent No. US1773980 A.

Fischer, J. et al., 2006. Measuring the Discernability of Virtual Objects in Conventional and Stylized Augmneted Reality. *Eurographic Symposium on Virtual Environments,* pp. 53-61.

Fitts, P., 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology,* 47(6), pp. 381-391.

Flasko, M., Pogscheba, P., Herder, J. & Vonolfen, W., 2012. Heterogeneous binocular camera-tracking in a Virtual Studio. *Workshop Virtuelle und Erweiterte RealitŁt der GI-Fachgruppe VR/AR,* Volume 8.

Fukaya, T. et al., 2003. An effective Interaction Tool for Performance in the virtual Studio - Invisible Light Projection System. *In Proceedings of the International Broadcasting Conference,* pp. 389-396.

Gibbs, S. et al., 1996. Virtual Studios: An Overview. *IEEE Mutlimedia,* 5(1), pp. 18-35.

Gibbs, S. & Baudisch, P., 1996. Interaction in the Virtual Studio. *Computer Graphics,* November.

Google, 2013. *Google Glass.* [Online]
Available at: https://www.google.co.uk/intl/en/glass/start/ [Accessed 04 10 2015].

Google, 2014. *Google Cardboard.* [Online]
Available at: https://www.google.co.uk/get/cardboard/ [Accessed 04 10 2015].

Grau, O. et al., 2005. The ORIGAMI Project: Advanced tools for creating and mixing real and virtual content in film and TV production. *IEE Proc.-Vis. Image Signal Process,* 152(4), pp. 454-469.

Greene, N., Kass, M. & Miller, G., 1993. Hierarchical Z-buffer visibility. *ACM Proceedings of the 20th annual conference on Computer graphics and interactive techniques,* Volume 27, pp. 231-238.

Guiard, Y., 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior,* Volume 19, pp. 486-517.

Haller, M., 2004. Photorealism or/and Nonphotorealism in Augmented reality. *In proceedings of the 2004 ACM SIGGRAPH internation conference on the virtual reality continuum and applications in industry,* pp. 189-196.

Hertzman, A., 1998. Painterly Rendering with Curved Brush Strokes of Multiple Sizes. *in Proceedings of 25th annual conference on computer graphics and interactive techniques,* pp. 453-460.

Hilton, A. et al., 1999. Virtual People: Capturing human models to populate virtual worlds. *IEEE Conference on Computer Animation,* pp. 174-185.

Hough, G., Athwal, C. & Williams, I., 2012a. ScaMP: A Head Guided Projection System. *ACM Designing Interactive Systems 2012.*

Hough, G., Athwal, C. & Williams, I., 2012b. Advanced occlusion handling for virtual studios. *Convergence and Hybrid Information Technology, Lecture Notes in Computer Science,* Volume 7425, pp. 287-294.

Hough, G., Williams, I. & Athwal, C., 2014a. Measurements of Live Actor Motion in Mixed Reality Interaction. *IEEE International Symposium on Mixed and Augmented Reality,* pp. 99-104.

Hough, G., Williams, I. & Athwal, C., 2014b. Measurement of Perceptual Tolerance for Inconsistencies within Mixed Reality Scenes. *IEEE International Symposium on Mixed and Augmented Reality,* pp. 343-344.

Hughes, D., 1996. *Virtual studio technology: The 1996 Eurovision Song Contest,* Geneva: European Broadcasting Union.

IJsselsteijn, W. et al., 1998. Perceived depth and the feeling of presence in 3DTV. *Musings on telepresence and virtual presence, Presence: Teleoperators and Virtual Environments,* pp. 207-214.

Interactivos, 2014. *Interactivos? projects.* [Online]
Available at: http://www.interactivosbham.co.uk/projects [Accessed 24 06 2014].

Iron Man 2. 2010. [Film] Directed by Jon Favreau. United States of America: Marvel Studios.

Iron Man. 2008. [Film] Directed by Jon Favreau. United States of America: Marvel Studios.

ITU-R BT.500-13, 2012. *Methodology for the Subjective Assessment of the Quality of Television Pictures,* Geneva: International Telecommunication Union.

ITU-R, 1990. *Studies toward the unification of picture assessment methodology,* Geneva: International Telecommunication Union.

Jacobs, D., 2004. *Department of Computer Science and UMIACS, University of Maryland.* [Online]
Available at: http://www.cs.umd.edu/~djacobs/CMSC427/VisibilityNonDiscrete.pdf [Accessed 2013 08 14].

Johnsgard, T., 1994. Fitts' Law with a virtual reality glove and a mouse: Effects of gain. *Canadian Information Processing Society,* pp. 8-15.

Jones, E. & Nisbett, R., 1971. *The actor and the observer: Divergent perceptions of the causes of behavior.* New York: General Learning Press.

Jurassic Park. 1993. [Film] Directed by S Spielberg. United States: Amblin Entertainment.

Kanade, T. et al., 1995. *Video-rate Z keying: A new method for merging images,* Pittsburgh, PA: The Robotics Institute, Carnegie Mellon University.

Katahira, R. & Soga, M., 2015. Development and Evaluation of a System for AR enabling Realistic Display of Gripping Motions using Leap Motion Controller. *19th International Conference on Knowledge Based and Intelligent Information and Engineering systems,* Volume 60, pp. 1595-1603 .

Kato, H. & Billinghurst, M., 1999. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. *2nd IEEE and ACM International Workshop on Augmented Reality,* pp. 85-94.

Kato, H. et al., 2000. Virtual object manipulation on a table-top AR environment. *International Symposium on Augmented Reality,* pp. 111-119.

Keefe, D. F. et al., 2001. CavePainting: a fully immersive 3D artistic medium and interactive experience. *In Proceedings of the 2001 ACM Symposium on Interactive 3D graphics,* pp. 85-93.

Khattak, S. et al., 2014. A real-time reconstructed 3D environment augmented with virtual objects rendered with correct occlusion. *IEEE Games Media Entertainment (GEM),* pp. 1-8.

Kim, N., Woo, W., Kim, G. & Park, C.-M., 2006. 3D virtual studio for Natural Inter-Acting. *IEEE trans. on Systems, Man and Cybernetics,* 36(4), pp. 758-773.

Kim, S. et al., 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. *In Proceedings of the 25th annual ACM symposium on User interface, software and technology,* pp. 167-176.

Klaue, J., Rathke, B. & Wolisz, A., 2003. Evalvid–A framework for video transmission and quality evaluation. *In Computer Performance Evaluation. Modelling Techniques and Tools,* pp. 255 - 272.

Koch, R. et al., 2009. MixIn3D: 3D mixed reality with ToF-camera. *Dynamic 3D imaging. Springer Berlin Heidelberg,* pp. 126-141.

Kuchera-Morin, J. et al., 2014. Immersive Full-Surround Multi-User System Design. *IEEE Computers & Graphics,* pp. 1-14.

Lalioti, V. & Woolard, A., 2003. *Mixed reality productions of the future,* Amsterdam, The Netherlands: BBC R&D White Paper, published in the conference publication of the Int. Broadcasting Convention.

Larsson, P., Vastfjall, D. & Kleiner, M., 2001. The Actor-Observer Effect in Virtual Reality Presentations. *CyberPsychology & Behaviour*, 4(2), pp. 239-246.

Leap, 2014. *Leap Motion.* [Online]
Available at: https://www.leapmotion.com/ [Accessed 05 10 2015].

Lee, J. et al., 2009. Anamorphosis Projection by Ubiquitous Display in Intelligent Space. *Lecture Notes in Computer Science,* pp. 209-217.

Lee, P. et al., 2015. TranSection: Hand-Based Interaction for Playing a Game within a Virtual Reality Game. *Computer Human Interaction,* pp. 73-76.

Lee, T. & Höllerer, T., 2008.. Hybrid Feature Tracking and User Interaction for Markerless Augmented Reality. *Proc. IEEE Conf. Virtual Reality,* pp. 145-152.

MacKenzie, S., 1992. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human Computer Interaction,* 7(1), pp. 91-139.

Mammoth Graphics and Kenziko Ltd, 2014. *Kinetrak.* [Online]
Available at: www.kinetrak.tv [Accessed 15 05 2014].

Marinos, D., Geiger, C. & Herder, J., 2012. Large-area moderator tracking and demonstrational configuration of position based interactions for. *10th European Interactive TV Conference,* pp. 105-114 .

Marinos, D., Geiger, C., Schwirten, T. & Göbel, S., 2010. Multitouch Navigation in Zoomable User Interfaces for Large Diagrams. *ACM International Conference on Interactive Tabletops and Surfaces,* pp. 275-276.

Martin, G., 2012. *Notes on willing Suspension of Disbelief,* New York: December7th.org.

Mendes, D. et al., 2014. Mid-Air Interactions Above Stereoscopic Interactive Tables. *IEEE Symposium on 3D User Interfaces 2014,* pp. 3-10.

Metamotion, 2010. *Gypsy 6 motion capture system.* [Online]
Available at: http://www.metamotion.com/gypsy/gypsy-motion-capture-system.htm [Accessed 2011 06 08].

Microsoft, 2010-a. *Kinect.* [Online]
Available at: http://www.xbox.com/en-us/live/kinect [Accessed 05 November 2011].

Microsoft, 2011. *Kinect.* [Online]
Available at: www.xbox.com/en-us/live/kinect. [Accessed 18 11 2011].

Microsoft, 2015. *Microsoft Hololens.* [Online]
Available at: https://www.microsoft.com/microsoft-hololens/en-us [Accessed 05 10 2015].

Milgram, P., Takemura, H., Utsumi, A. & Kishino, F., 1995. Augmented Reality:A class of displays on the reality-virtuality contnuum. *International Society for Optics and Photonics,* pp. 282-292.

Mine, M., 1995. *Virtual Environment Interaction Techniques,* Chapel Hill: University of North Carolina Computer Science Technical Report TR95-018.

Mine, M. R., Brooks Jr, F. P. & Sequin, C. H., 1997. Moving objects in space: exploiting proprioception in virtual-environment interaction. *In Proceedings of the 24th annual conference on Computer graphics and interactive techniques,* pp. 19-26.

Minoh, M. et al., 2007. Direct Manipulation of 3D Virtual Objects by Actors for Recording Live Video Content. *Second International Conference on Informatics Research for Development of Knowledge Society Infrastructure,* pp. 11-18.

Mitsumine, H., Fukaya, T., Komiyama, S. & Yamanouchi, Y., 2005. Immersive Virtual Studio. *ACM SIGGRAPH Sketch,* p. 121.

Morimoto, C. H. & Mimica, M., 2005. Eye gaze tracking techniques for interactive applications. *Computer vision and image understanding,* 98(1), pp. 4-24.

Murphy-Chutorian, E. & Trivedi, M., 2009. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 31(4), pp. 607-626.

Nielsen, J., 1994. *Usability Engineering.* San Diego: Academic Press.

Nielsen, J. & Molich, R., 1990. Heuristic Evaluation of User Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* pp. 249-256.

Oculus Rift, 2013. *Oculus Rift.* [Online]
Available at: https://www.oculus.com/en-us/ [Accessed 04 10 2015].

OpenCV 2.3.1., 2011. [Online]
Available at: http://opencv.willowgarage.com/wiki/ [Accessed 19 12 2012].

Orad, 2004. *Camera Tracking.* [Online]
Available at: http://www.nexus-set.tv/pdf/CameraTracking.pdf [Accessed 2015 01 16].

Orad, 2010. *Orad.tv.* [Online]
Available at: http://www.orad.tv/Data/Uploads/Tracking_1.pdf [Accessed 14 05 11].

Orad, 2012a. *Orad 3Designer.* [Online]
Available at: http://www.orad.tv/3designer [Accessed 19 07 2012].

Orad, 2012b. *Orad HDVG.* [Online]
Available at: http://www.orad.tv/products/hdvg [Accessed 19 07 2012].

OrganicMotion, 2012. *Radium Motion Capture.* [Online]
Available at: http://www.radiusmotioncapture.com/OrganicMotion_OpenStage_Brochure.pdf [Accessed 18 05 2014].

Pastrana-Vidal, R. R., Gicquel, J. C., Colomes, C. & Cherifi, H., 2004. Sporadic frame dropping impact on quality perception. *In Electronic Imaging 2004, International Society for Optics and Photonics,* pp. 182-193.

Pece, F., Kautz, J. & Weyrich, T., 2011. *Three Depth-Camera Technologies Compared,* London: Department of Computer Science, University College London.

Petit, B., Lesage, J. & C., M., 2010. Multicamera Real-Time 3D Modeling for Telepresence and Remote Collaboration. *International Journal of Digital Multimedia Broadcasting,* Volume Article ID 247108.

Pinhanez, C. 2., 2001. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces.. *Ubiquitous Computing ,* pp. 12-17.

Pixoft, 2009. *[Vicasso information.* [Online]
Available at: http://www.pixoft.co.uk/Vicinfo.htm

Polhemus, 2011. *Biometric applications.* [Online]
Available at: http://www.polhemus.com/?page=Motion_Applications_Biomechanics [Accessed 26 08 2011].

Poupyrev, I., Billinghurst, M., Weghorst, S. & Ichikawa, T., 1996. The Go-Go Interaction Technique: Non-linear Mapping for Direct Manipulation in VR. *ACM Symposium on User Interface Software and Technology (UIST),* pp. 79-80.

Price, M. & Thomas, G., 2000. *3D virtual production and delivery using mpeg-4,* London: British Broadcasting Corporation.

Primesense, 2011. *OpenNI.* [Online]
Available at: openni.org [Accessed 19 12 2011].

Raskar, R. & van Baar, J., 2003. ilamps: Geometrically aware and self-conjuring projectors. *ACM Transactions on Graphics (TOG),* 22(3), pp. 809 - 818.

Rekimoto, J., 2014. Traxion: A Tactile Interaction Device with Virtual Force Sensation. *SIGGRAPH 2014,* p. Poster.

Roth, S., 1982. Ray Casting for Modeling Solids. *Computer Graphics and Image Processing,* 2(18), pp. 109-144.

Sazzad, P., Yamanaka, S., Kawayoke, Y. & Yuukou, H., 2009. Stereoscopic image quality prediction. *International Workshop on Quality of Multimedia Experience*, pp. 180-185.

Schuchardt, P. & Bowman, D., 2007. The benefits of immersion for spatial understanding of complex underground cave systems. *In Proceedings of the 2007 ACM symposium on Virtual reality software and technology,* pp. 121-124.

Seeing Machines, 2013. *FaceAPI.* [Online]
Available at: www.seeingmachines.com/product/faceapi/

Sensor products inc, 2009. *Improving performances of data gloves based on bend sensors.* [Online]
Available at: http://www.sensorprod.com/news/white-papers/dgb/index.php [Accessed 26 08 2011].

Seshadrinathan, K., Soundararajan, R., Bovik, A. & Cormack, L., 2010. Study of subjective and objective quality assessment of video. *IEEE transactions on Image Processing,* 19(6), pp. 1427-1441.

Shimode, S., Hayashi, M. & Kanatsugu, Y., 1989. New Chromakey Vision Technique with Hi-Vision background. *IEEE Transactions on Broadcasting,* 35(4), pp. 357-361.

Simsch, J. & Herder, J., 2014. SpiderFeedback: visual feedback for orientation in virtual TV studios. *ACE '14 Proceedings of the 11th Conference on Advances in Computer Entertainment Technology,* p. Article no. 11.

Steptoe, W., Julier, S. & Steed, A., 2014. Presence and Discernability in Conventional and Non-Photorealistic Immersive Virtual Reality. *IEEE International Symposium on Mixed and Augmented Reality,* pp. 213-218.

Sturman, D. & Zeltzer, D., 1994. IEEE Computer Graphics and Applications. *A survey of glove-based input,* 14(1), pp. 30-39.

Sun, G. et al., 2014. An Advanced Computational Intelligence System for Training of Ballet Dance in a CAVE Virtual Reality Environment. *2014 IEEE International Symposium on Multimedia,* pp. 159-166.

Sutcliffe, A. & Gault, B., 2004. Heuristic evaluation of virtual reality applications. *Interacting with Computers,* Volume 16, pp. 831-849.

Teather, R. & Stuerzlinger, W., 2007. Guidelines for 3D Positioning Techniques. *Proc. of the 2007 conference on Future Play,* pp. 61-68.

Terminator 2: Judgement Day. 1991. [Film] Directed by James Cameron. United States: Carolco Pictures.

The Museum of the City of San Francisco, 2013. [Online]
Available at: http://www.sfmuseum.org/hist3/sallie.html

The Thief of Baghdad. 1940. [Film] United Kingdom: London Film.

Thomas, G. A., Jin, J., Niblett, T. & Urquhart, 1997. A versatile camera position measurement system for virtual reality tv production. *In Proceedings of Conference on International Broadcasting Conventions,* pp. 284-289.

Thomas, G. & Grau, O., 2002. 3D image sequence acquisition for TV & film production. *in Proc. of 1st Symp. on 3D Data Processing Visualization Transmission.*

van den Bergh, F. & Lalioti, V., 1999. Software chroma keying in an immersive virtual environment. *South African Computer Journal,* pp. 155-162.

Västfjäll, D., Larsson, P. & Kleiner, M., 2000. Development and validation of the Swedish viewer-user presence questionnaire.

Vierjahn, T., Woldecke, B., Geiger, C. & Herder, J., 2009. Improved Direction Signalization Technique Employing Vibrotactile feedback. *VRIC'09,* pp. 169-174.

vizrt, 2013. *Interactive election coverage.* [Online]
Available at: http://www.vizrt.com/products/viz_trio/39222/Interactive_election_coverage [Accessed 27 5 2014].

VizRT, 2014. *viz virtual studio.* [Online]
Available at: http://www.vizrt.com/products/viz_virtual_studio/# [Accessed 14 05 2014].

Vlachos, T., 2000. Detection of blocking artifacts in compressed video. *Electronic letters,* pp. 1106-1108.

Wang, Z., Bovik, A. & Evans, B., 2000. Blind Measurement of Blocking Artifacts in Images. *2000 International Conference on Image Processing,* Volume 3, pp. 981-984.

Weber, M., 2008. A hybrid approach towards fully automatic 3D marker tracking. *ACM Symposium on Virtual Reality Software and Technology*, pp. 243-244.

Westworld. 1973. [Film] Directed by Michael Crichton. USA: Metro-Goldwyn-Mayer.

White, I., 2010. *Vizrt.com/casestudies.* [Online]
Available at: http://www.vizrt.com/casestudies/36228/ITNs_development_of_a_virtual_touch_screen
[Accessed 15 05 2014].

Whitted, T., 1980. An Improved Illumination Model for Shaded Display. *Communications,* pp. 343-347.

Winkler, S., Sharma, A. & McNally, D., 2001. Perceptual Video Quality and Blockiness Metrics. *In proceedings of International Symposium on Wireless Multimedia communications,* pp. 547-552.

Woldecke, B., Marinos, D., Herder, J. & Geiger, C., 2010. Vibrotactile pitfalls: Armguidence for moderators in virtual TV studios. *In Proceedings of the 13th International Conference on Humans and Computers,* pp. 72-80.

Woldecke, B. et al., 2009. Steering Actors Through A Virtual Set Employing Vibrotactile Feedback. *TEI'09, Proc. of the Third International conference on Tangible Embedded Interaction,* pp. 169-174.

Woop, S., Schmittler, J. & Slusallek, P., 2005. RPU: A Programmable Ray Processing Unit for Realtime Ray Tracing. *Siggraph,* 24(3), pp. 434-444.

Wu, H. & Yuen, M., 1997. A generalized block-edge impairment metric for video coding. *IEEE Signal Processing Letters,* pp. 317-320.

Yang, P., Wu, W., Moniri, M. & Chibelushi, C., 2008. A Sensor-based SLAM Algorithm for Camera Tracking in Virtual Studio. *International Journal of Automation and Computing,* 5(2), pp. 152-162.

Young, A. D., Ling, M. J. & Arvind, D., 2010. Distributed estimation of linear acceleration for improved accuracy in wireless inertial motion capture. *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks,* pp. 256-267.

Zerroug, A., Cassinelli, A. & Ishikawa, M., 2009. Virtual Haptic Radar. *In ACM SIGGRAPH ASIA 2009,* p. 9.

zLense, 2014. *Interactive environment.* [Online]
Available at: http://zlense.com/features/#interactiveenvironment [Accessed 2014 09 19].

Appendix A. SELECTION OF MOTION CAPTURE METHOD

A.1. Introduction

Unlike most systems that require motion capture for interaction, the motion capture system used in the virtual studio should be concealed from the viewer or it will break the audience's suspension of disbelief. This is a requirement unique to the virtual studio.

While motion capture technology in fields such as virtual reality and medical research are advanced, they typically rely on methods that are either visible to a third party (e.g. mechanical tracking) or tethered in manner that restricts movement (e.g. magnetic tracking). Imperceptibility is not generally accounted for in existing motion capture methods in these fields as it is not important and only required for niche applications, such as the virtual studio.

In this section the existing motion capture technologies are analysed and their suitability for the needs of the virtual studio are assessed. From the heuristic analysis we describe a set of requirements for a motion capture system in the virtual studio and discuss how established motion capture techniques meet these requirements, with the most compatible being selected. This chapter only reviews the state of the art in motion capture that was available in 2011, the beginning of this study.

Previously, Zerroug *et al* (Zerroug, et al., 2009) presented a list of requirements for the development of tracked haptic devices for use in the virtual studio, of which reliable motion capture and imperceptibility were significant factors. Although conducted independently, our evaluation came to many of the same conclusions as Zerroug's work. The requirements stated by him were:

1. *Each tracking module must know its own position in the room, independently from the others and without the need for a centralized computer*
2. *The tracker must be inexpensive in order to enable scalability*
3. *The modules must be invisible to the cameras (in particular, they should not emit or reflect visible light)*
4. *Interference should be minimal or nonxistent if one wants tens of modules to work simultaneously (this rules out most non-sophisticated magnetic trackers)*

It is important to note that in #1 of Zerroug's requirements they did not use a centralised computer for tracking as each haptic module was designed to function independently, a constraint that does not apply to our work.

The heuristic investigation of the motion capture method yielded the following:

#1: Must be imperceptible to the viewer

Heuristic: The method of motion capture must not be visible to the viewer.
Problem: If tracking devices are placed on the actor's hands, head or over their clothes, they would be visible to the viewer and would break their suspension of disbelief.
Requirement: The motion capture system must be imperceptible to the audience.

#2: Must have low impedance on actor motion

Heuristic: The actor must be able to move as they would normally do so in an existing virtual studio.
Problem: Tethered or occludable motion capture systems are likely to interfere with the performance of the actor.
Requirement: The method of motion capture must not impede the motion of the actor, nor the natural motion of the actor interfere with the quality of the tracking (e.g. issues caused by occlusion).

#3: Must provide sufficiently detailed motion capture data

Heuristic: The motion capture method should provide the location of the actor's joints in 3DOF (Location of X, Y, Z).
Problem: Some motion capture systems provide data that is not suitable for use with interaction or occlusion (e.g. provides orientation, but not location)
Requirement: If interaction or occlusion is desired, the 3D motion of the actor is needed to instruct the objects how to behave. The motion capture system must be able to provide this data for all joints that are relevant for interaction with sufficient detail; primarily the hands.

#4: Must be low latency

Heuristic: To ensure plausibility of the interaction the tracking method must have a low latency.
Problem: Actors provided with high latency feedback will probably struggle to place their hands in the appropriate location during an interaction, as it will make it difficult for them to determine the current location of their hands relative the virtual object.
Requirement: An artificial delay in the video can be added to the final composite of the scene to ensure that the interaction remains consistent for the viewer. However, for the benefit of the actor the end-to-end latency of the system needs to be as low as possible to provide them with immediate feedback. As the motion capture is a large contributor to the end-to-end latency, it should be processed within 4-5 frames (160-200ms).

#5: Tracking system must accommodate acting space

Heuristic: Motion capture systems should have a suitable range for acting in the virtual studio and must not be affected by any of the equipment.

Problem: Any sensory equipment from the motion capture system must not interfere with the equipment of the virtual studio and must allow the actor to travel.

Requirement: The motion capture system must also support the range of the acting space the actor requires with reliable accuracy. Results of the investigation suggest that a tracking area over a floor space of 9m^2 (3mx3m) is sufficient for most interactions.

A.2. Summary of motion capture techniques and suitability

Many forms of tracking have already been developed and implemented for other interactive virtual environment systems. Figure A-1 and Figure A-2 provide a summary of motion capture methods that could be used for virtual object manipulation at the start of this investigation from a range of related fields (optical and non-optical).

The resolution field refers to the increments in which the measurements are made. The accuracy field refers to the potential error margin of the measurements. The latency field is the amount of time required to determine the motion capture measurements. For some motion capture systems the state of the art capabilities are not clearly stated, in this case a description is given.

A.2.1. Optical tracking

Optical tracking uses a single or an array of camera to capture the motion of the user for interaction. There are two applicable forms, marker-based and markerless

	Tracking method	State of the art	Resolution	Accuracy	Latency	Advantages	Disadvantages
Marker based	Marker Based optical	(Weber, 2008)	High – Location of markers can detected with high accuracy	High – 3DOF location of markers can be found accurately.	<10ms	Accurate tracking from a two camera set up.	Visible markers: Subject to occlusion.
Markerless	Markerless optical (volumetric)	GrImage (Allard, et al., 2007) (Petit, et al., 2010)	Low - Volumetric model loses visual fidelity.	High – can detect many points in 3DOF accurately.	>100ms (Lower with large computer resources)	No visible equipment.	Large camera installation.
	Depth-sensitive camera tracking	Kinect (Microsoft, 2011)	1cm at 2m, increasing with distance.	High – Can find location in 3DOF to <2cm	<40 ms	No visible equipment.	Difficult to track fine details such as fingers. 2.5D data subject to occlusion.

Figure A-1. Optical tracking

Marker Based Optical Motion Capture. Marker-based optical tracking typically uses high-visibility markers which are detected by multiple cameras from which the pose of the actor is reconstructed. The markers can be active (where the markers are LEDs that are illuminated in sequence so they can each be identified) or passive (where the markers are semi-spheres that reflect an external light source). Marker based motion capture has a low impedance on the motion of the actor, supporting Requirement #2. It allows up to 6DOF tracking of joints which can be captured with low latency (<10ms), fulfilling Requirement #3 and #4 respectively. The tracking would also be sufficiently accurate over a range suitable for the virtual studio, fulfilling requirement #3. However, by their nature the markers are highly visible and thus violate Requirement #1 and due to this, this method of motion capture is unsuitable for the virtual studio.

Markerless Optical Motion Capture. Markerless optical tracking methods typically infer information about the actor's pose from the video images, without requiring any marker present in the image. This method usually requires multiple camera setups, where an image of the actor is captured from many different angles. Single camera methods do exist, but are not as powerful as multiple camera methods. The motion of the actor is then reconstructed by finding their visual hull, where the images from each of the cameras are compared and from this the actor's body can be constructed as a 3D model in a manner that suits Requirement #3.

This method is not perceptible to the viewer as no markers or body-worn tracking devices are used, satisfying requirement #1. For the same reason, the actor's motion is also not impeded using this method, satisfying Requirement #2. This method of motion capture also supports the range required, fulfilling Requirement #5. However, this process is computationally expensive and does not offer a low latency motion capture, violating Requirement #4. This method appears to be almost ideal, but the latency of the system would provide a significant hurdle before any interaction could be implemented.

Depth-sensitive Cameras. Depth-sensitive camera tracking uses a single camera that is able to infer the depth of each pixel in a captured image, allowing the distance of elements in the scene from the camera to be calculated. Many approaches towards obtaining the distance of each pixel in an image exist, namely stereo camera triangulation, Time-of-Flight and Light Coding.

Although Time-of-Flight and Structured Light cameras use Infrared projections to calculate the image depth, the projected patterns are not visible to a studio camera, particularly when an infrared filter is used on the lens, meaning that motion capture would be imperceptible to the viewer, satisfying Requirement #1. A possible concern is that the infrared projection could suffer from interference via the studio lights; however infrared light has been successfully used for tracking in the virtual studio, for example with the Xync camera tracking system (Orad, 2004), with little interference.

Depth cameras do not impede the motion of the actor, but the 2.5D nature of the captured image means that occluded areas of the actor cannot be tracked. However, typically in the virtual studio the actor will be facing the sensor which allows skeletal tracking to be accomplished with little risk of occlusion, which fulfils Requirement #2.

The tracking of joint locations for motion capture is not handled by the depth cameras alone, but uses software to infer the pose of the actor. We measured this process using OpenNI, a standard open platform for this task, and a Kinect. The location of the actor's joints could be tracked in 3DOF at a resolution of <3mm X and Y, 1cm Z at 2 meters fulfilling Requirement #3. This would be improved with TOF depth cameras, which are higher resolution and more accurate. Depth-sensitive cameras support motion capture within a range of 0.5 to 4m from the camera fulfilling Requirement #5. The latency of the systems differ, but some can possess a latency of <40ms, satisfying Requirement #4.

A.2.2. Non-optical tracking

Non-optical tracking uses a range of devices to measure the location of certain body parts or measure the angle of joints for interaction. Current uses exist in virtual reality systems and motion tracking in medicine (Polhemus, 2011). There are several forms, the main being Inertial Measurement Units (IMUs), mechanical tracking, wired gloves and magnetic tracking.

Tracking method	State of the art	Resolution	Accuracy	Latency	Advantages	Disadvantages
Inertial Measurement Units	Young's wireless IMU mo-cap system (Young, et al., 2010)	1°	~3°, but decreases with inertial drift.	<10ms	Trackers can be discretely placed: rapidly advancing technology.	Hybrid system needed for fine detail: Inertial drift.
Mechanical tracking	Gypsy 6 (Metamotion, 2010)	High – Limited to potentiometer	High – Measures to within fraction of a degree.	<10ms	Allows robust skeletal tracking	Visible equipment. Exoskeleton form. Requires location tracking.
Wired gloves	Project fineskills (Digital Rune, 2011)	High – Limited to potentiometer	High– Measures to within fraction of a degree.	<10ms	Allows complex object manipulation (Minoh, et al., 2007).	Visible equipment. Needs to be part of a hybrid system. Only tracks hand pose.
Magnetic tracking	TrakSTAR (Ascention, 2010)	5mm at 30.5 cm, lowering with distance (up to 3.3m)	Sub-centimetre at 30.5 cm, poorer with distance	<10ms	Small size.	Affected by ferrous materials. Tecthered

Figure A-2. Non-Optical tracking

Inertial Measurement Units. An IMU is a physical tracking device that principally captures rotational motion and, in some cases, limited location motion. Comprehensive inertial tracking devices that use a combination of Gyroscopic (devices that sense orientation), Accelerometer (devices that sense velocity) and Magnetic (compass) sensors are able to track motion in 6 Degrees of Freedom (6DOF), allowing changes in orientation and location to be tracked. A popular example of an IMU is the Wiimote, a handheld games controller developed by Nintendo that relays the user's hand motion to the Wii games console.

One of the principle issues with IMUs is that any errors that are produced are cumulative because the motion is tracked relative to an initial starting point, which renders data received after prolonged use unreliable. This means that the location of the device in a known space is very unreliable and that these devices can only be relied upon for describing the general direction of motion. This lack of accuracy violates Requirement #3.

Inertial devices can be hidden discretely in some cases, but for tracking hand and head motion they would likely need to be placed in a location on the actor's body that would be visible to the viewer, violating requirement #1. They are low latency at <10ms meeting Requirement #4, although wireless inertial tracking systems are limited to the speed of communication (typically Bluetooth™ or WI-FI). They allow motion to be tracked over a large area suitable for the virtual studio, only limited to the range of the wireless communication system, meeting Requirement #5. Wireless inertial tracking devices do not typically impede the motion of the actor, meeting Requirement #2.

Mechanical tracking. Mechanical tracking is typically constructed using potentiometers arranged in the form of an exoskeleton or placed on the joints of the user. These potentiometer devices are physical devices that are large and need to be placed on the actor's body in locations that would be visible to the viewer such as the hands, violating requirement #1. Mechanical tracking is also only capable of tracking the orientation of the actor's joints and in some systems their location relative to each other. The location of the actor's joints in a known space cannot be resolved using mechanical tracking alone, and would at least need to be supplemented by a system that can find the location of the devices too. Therefore a mechanical motion capture system would violate Requirement #3 as alone it would not provide sufficient tracking data suitable for use in the virtual studio. The nature of many potentiometer devices would add resistance to the joints of the actor or if an exoskeleton system was required would impede their movement, thereby violating Requirement #2. The range possible using mechanical tracking is limited to that permitted by the communications system, which meets requirement #5. At <10ms the latency of mechanic tracking is low, meeting Requirement #4, but is again limited to the speed permitted by the communication system used.

Wired gloves. Wired gloves again use potentiometers, but in the form of a glove with flex sensitive potentiometers along the fingers to track how bent they are. The glove only captures the pose of the actor’s hand and typically the location of the hand itself has to be captured using other means of tracking. As such, the wired glove violates Requirement #3 unless supplemented with further location tracking. The wired glove would also be visible to the viewer, violating Requirement #1. However, the wired glove does not impede actor motion (fulfilling requirement #2), is low latency (fulfilling

Requirement #3) and would work over a suitable range for the virtual studio (fulfilling requirement #5); the last two again primarily impeded by the communication system assuming that the glove is wireless. **Magnetic motion capture.** Magnetic motion capture uses devices placed on the actor's body that can sense a low frequency magnetic field produced by a transmitter. The sensors can infer their position in a 3D space in 6DOF, allowing the orientation and location of the body part they have been placed on to be tracked. Magnetic motion capture provides sufficiently detailed tracking and is typically low latency (<10ms) and, satisfying Requirements #3 and #4 respectively.

Magnetic motion capture systems use visible hardware and are susceptible to occlusion, which violate Requirements #2 and #3. Magnetic sensors are also highly susceptible to interference from magnetic fields and metal objects within the acting space, an issue that is inescapable in virtual studio environments (particularly as the actor moves further away from the sensor). Consequently, the tracking itself in the studio environment would be unreliable, violating Requirement #5. Due to the unreliability of the motion capture data in the virtual studio and the visible nature of the system, magnetic motion capture is not adequate for use in the development of an Interactive Virtual Studio.

A.2.3. Summary

A summary of the motion captured systems that have been reviewed and their compatibility with the requirements described in 3.1 are presented in **Figure A-3**. We produced a brief feasibility report for each of the compatible methods; the markerless optical method, the depth camera method and the wired glove method. Although Marker based optical tracking scored 4/5, due to their inherent visibility in the scene they will not be investigated further.

		Heuristics/Met Requirements					
		#1 Imperceptible to the viewer	#2 Low impedance on actor motion	#3 Sufficient tracking detail	#4 Low latency	#5 Sufficient range	Total
Motion capture method	Optical (Marker)	x	✓	✓	✓	✓	4/5
	Optical (Markerless)	✓	✓	✓	x	✓	4/5
	Depth camera	✓	✓	✓	✓	✓	5/5
	Inertial tracking	x	x	x	✓	x	1/5
	Mechanical tracking	x	x	✓ *	✓	✓	3/5
	Wired gloves	x	✓	✓ *	✓	✓	4/5
	Magnetic tracking	x	x	✓	✓	x	2/5
	Magnetic tracking	x	x	✓	✓	x	2/5

Figure A-3. Suitability of Motion Capture Systems According to the Heuristics

Markerless optical tracking. A review of the literature found that a markerless optical tracking approach similar to the requirements of the virtual studio had been implemented in a project named GrImage (Allard, et al., 2007), which used multiple cameras to reconstruct the visual hull of a user. The visual hull was treated as a virtual model inside a virtual environment and in real time the virtual model would match the movements of the user, allowing them to push virtual objects using a simple physics engine. In this project some of the visual fidelity of the user was lost when converted into a 3D model, which means it would be unsuitable for television. For this reason we decided that it would be risky to implement a similar system in the virtual studio.

Depth Cameras. Three forms of depth-sensitive camera are available. The performance of each of these systems is as follows:

Stereo Camera: Large latency, large computational costs, low accuracy and resolution (due to Correspondence Problem).

Time-of-Flight camera: Large latency (although reduced considerably since initial study), large computational cost, high accuracy and fine resolution.

Structured light: Low latency, low computational cost, reasonable accuracy and resolution.

As discussed in chapter 2, a depth camera approach towards motion capture was investigated by Kim *et al* (Kim, et al., 2006) who used a stereo camera to track the motion of the actor, as well as other elements of the scene. This work did not track the location of the actor's joints, but the depth data was used to identify and segment the actor. The segmentation process was inexact and the actor was treated as a cloud of points. Recent advances in depth-sensitive camera technology mean that tracking the skeletal motion capture of the actor has become a simple and common task.

A Microsoft Kinect was used as a representative approach for assessing the suitability of a depth-sensitive camera. The Kinect is a common commercial depth-sensitive camera that uses structured light to assess the depth of each pixel in an image; it is well supported in terms of software, with many applications existing that support skeletal tracking. Other structured light or Time-of-Flight cameras may offer higher resolutions, greater tracking ranges.

The Kinect, interfaced with OpenNI, provided 3DOF skeletal data of an actor at a rate of 30FPS, with a resolution of 3mm on the X and Y axes and 20mm on the Z axis at a distance of approximately 2m from the device. At the time of this study, depth camera technology was rapidly advancing and starting to enter the consumer market.

Wired glove. A previous implementation of the wired glove into the virtual studio by Minoh *et al* yielded some success (Minoh, et al., 2007), so further investigation into the viability of this method was conducted.

The work of Minoh *et al* (Minoh, et al., 2007) demonstrated that using a wired gloved in the virtual studio was an effective approach for achieving interaction, although at the cost of realism. In summary, Minoh's approach was to use a wired glove to track the motion of the actor's hand for interaction and improve the realism by removing the visually perceptible glove from the scene by rendering a less distracting virtual hand over it. Figure A-4 shows an example of this method in use.

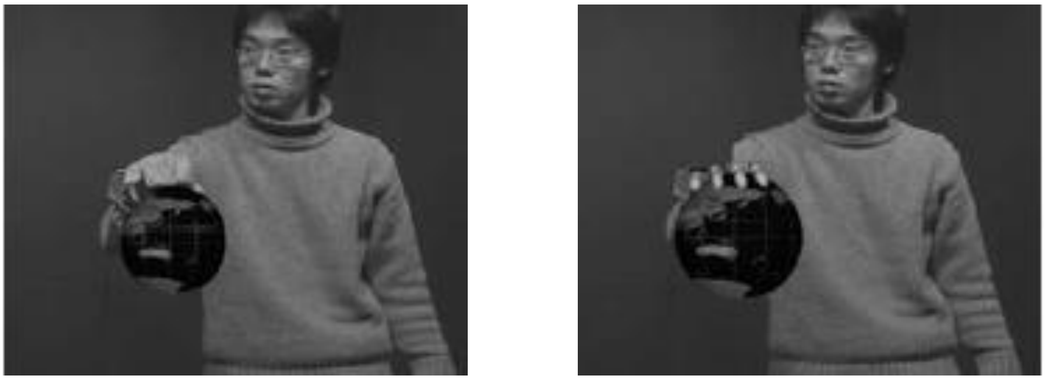
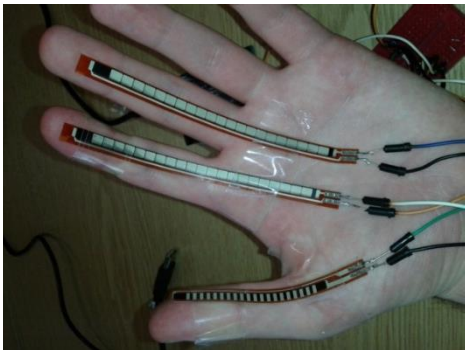


Figure A-4a. Interaction using wired glove

Figure A-4b. Virtual hand rendered over the wired glove



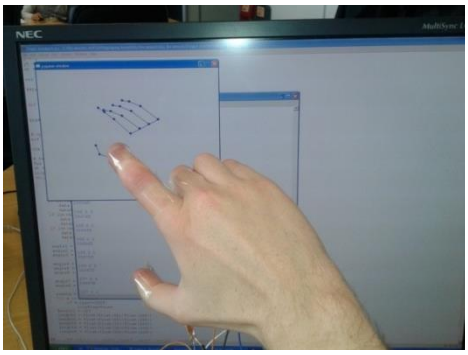


Figure A-5. Prototype imperceptible glove

Figure A-6. Example of prototype imperceptible glove displayed on monitor

Figure A-4. Wired glove interaction in the virtual studio with virtual hand rendered over the actor's hand (from Minoh (Minoh, et al., 2007))

Minoh's system allowed a good quality single-handed interaction, but the issue of visibility and the CGI hand meant that it was not an acceptable tracking system as it would violate the suspension of disbelief. This led us to develop a near-imperceptible wired glove that may improve the realism of the interaction and remove the need for a virtual hand as part of a feasibility study.

In the prototype, the bend of each finger is measured by discretely placed flex sensors (Sensor products inc, 2009) on the underside of the actor's hand, which in the final design would be blended in to the texture and colour of the actor's skin. The operation of each flex sensor is to measure the bend of each finger and if the dimensions of the actor's hand are known then inverse kinematics can be employed to reconstruct the pose of each finger. Figure A-5 shows a prototype of the near imperceptible wired glove from the underside of the hand. Figure A-6 shows the glove in operation, accurately capturing and reconstructing the pose of a hand on a computer monitor; this image also shows the glove from behind, where the components are near imperceptible.

Since this investigation and development of a prototype imperceptible glove, Microsoft have developed Digits (Kim, et al., 2012), a device that would offer the same level of hand pose estimation as a wired glove. A module is placed on the wrist that contains a camera and an infrared laser projector. The camera captures the deformation of the laser from which the pose of the fingers is calculated. Concealing this device under the sleeve of the actor's shirt would offer a better approach towards capturing the pose of the hand in an imperceptible manner

Although the prototype wired glove was able to accurately reconstruct the hand pose, it was not used. The imperceptible wired glove could be taken further and used in conjunction with the method of motion capture presented here to capture the pose of the hand, something that is difficult to achieve using the Kinect.

Appendix B. DETAILS OF IMAGE TRANSFORMATION TECHNIQUE

Step 1: Define a constant target image size and shape.

In this step a target image size is defined, which is the size that the image will be corrected to when projected at arbitrary angles. A shape is also defined, but in convention should be a rectangle. The first stage is to calculate the location of the corners of the unwarped projection area relative to the centroid of the projection area, when ScaMP is projecting perpendicular to the projection surface. The second step of this process is to define the target image size as a constant fraction of the unwarped projection area, which is expressed as a fraction of 1. Figure B-1 demonstrates how these stages are presented in the equations used in this step. Figure B-2 shows this process, where the corners of the target image size are found as a percentage of the projection area.

$$corner\ on\ x\ axis = \underbrace{\boxed{\frac{Z}{\cos\left(\frac{Py}{2}\right) * \sin\left(\frac{Px}{2}\right)}}}_{\text{Location of Corner from centroid}} * \underbrace{\boxed{A\%}}_{\text{Fraction}}$$

Figure B-1. Annotated sample equation for calculating the corner locations of the target **image size.**

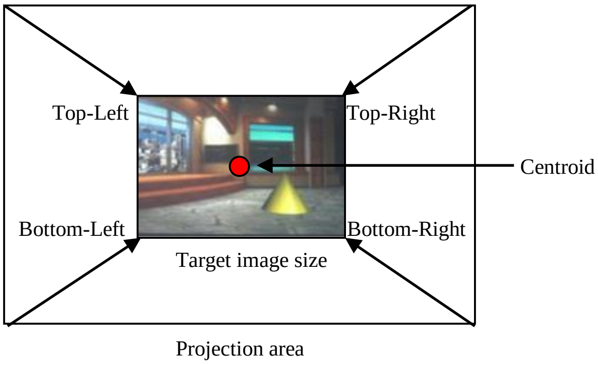


Figure B-2. Defining the target image size from the unwarped projection size (example shown is a reduction to 43%)

The purpose for setting the target image size as a fraction of the projection area is because once projected at arbitrary angles a full size may not fit into the warped projection area fully. It was found that a target image size set at a fraction of 0.33 (33%) of the projection area was sufficient to avoid this. Equation B-1 to Equation B-4 show how the location of the corners for the target image size in x and y are calculated. The first part of this equation calculates the location of the relevant corner for the projection area relative to the centre of the projection area and the second part reduces the position of the corner relative to the centroid by a certain percentage.

In these equations Z represents the distance of the ScaMP unit from the planar surface, Px and Py represent the height and width of the angle of image projection respectively and A is the factor by which the image size is reduced, which should be constant for all corners in both X and Y.

Top-Right: $x = \frac{z}{\cos\left(\frac{Py}{2}\right)*\sin\left(\frac{Px}{2}\right)} * A$ $y = \frac{z}{\cos\left(\frac{Px}{2}\right)*\sin\left(\frac{Py}{2}\right)} * A$

Equation B-1. Calculation of Top-Right corner of target image size

Top-Left: $x = \frac{z}{\cos\left(\frac{Py}{2}\right)*\sin\left(\frac{Px}{2}\right)} * A$ $y = \frac{z}{\cos\left(\frac{Px}{2}\right)*\sin\left(\frac{Py}{2}\right)} * A$

Equation B-2. Calculation of Top-Left corner of target image size

Bottom-Right: $x = \frac{z}{\cos\left(\frac{Py}{2}\right)*\sin\left(\frac{Px}{2}\right)} * A$ $y = \frac{z}{\cos\left(\frac{Px}{2}\right)*\sin\left(\frac{Py}{2}\right)} * A$

Equation B-3. Calculation of Bottom-Right corner of target image size

Bottom-Left: $x = \frac{z}{\cos\left(\frac{Py}{2}\right)*\sin\left(\frac{Px}{2}\right)} * A$ $y = \frac{z}{\cos\left(\frac{Px}{2}\right)*\sin\left(\frac{Py}{2}\right)} * A$

Equation B-4. Calculation of Bottom-Left corner of target image size

Step 2: Calculate the area of the warped projection.

The second step of the correction process is to calculate the location of each corner of the warped projection area relative to the centroid of projection area. This process is illustrated in Figure B-3 and is accomplished using Equation B-5 to Equation B-8 (page 202).

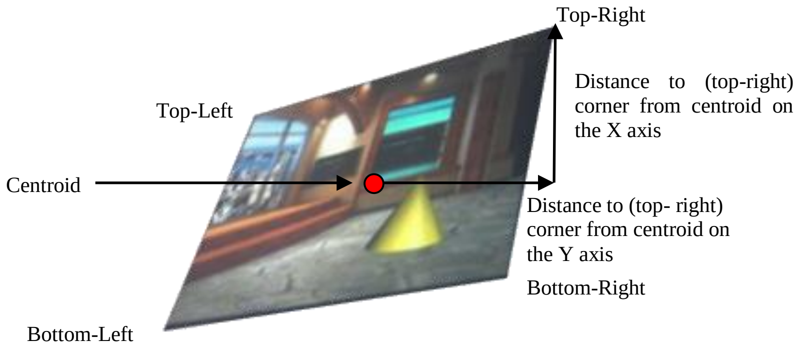

Top-Right
Distance to (top-right) corner from centroid on the X axis
Top-Left
Centroid
Distance to (top- right) corner from centroid on the Y axis
Bottom-Right
Bottom-Left

Figure B-3. The corner locations of the warped projection relative to the centroid of the projection area

In these equations Z represents the distance of the ScaMP unit from the planar surface, Panθ and Tiltθ represent the angle of the mirror in the pan and tilt respectively. Px and Py are the height and width angle of projection respectively.

Top-Right: $x = \frac{Z}{\cos\left(Tilt\theta+\left(\frac{Py}{2}\right)\right)*\sin\frac{Px}{2}} + \frac{Z}{\cos(pan\theta)}$ $y = \frac{Z}{\cos\left(Pan\theta+\left(\frac{Px}{2}\right)\right)*\sin\frac{Py}{2}} + \frac{Z}{\cos(Tilt\theta)}$

Equation B-5. Calculation for Top-Right corner of projection area after warp

Top-Left: $x = \frac{Z}{\cos\left(Tilt\theta-\left(\frac{Py}{2}\right)\right)*\sin\frac{Px}{2}} + \frac{Z}{\cos(pan\theta)}$ $y = \frac{Z}{\cos\left(Pan\theta+\left(\frac{Px}{2}\right)\right)*\sin\frac{Py}{2}} + \frac{Z}{\cos(Tilt\theta)}$

Equation B-6. Calculation for Top-Left corner of projection area after warp

Bottom-Right: $x = \frac{Z}{\cos\left(Tilt\theta+\left(\frac{Py}{2}\right)\right)*\sin\frac{Px}{2}} + \frac{Z}{\cos(pan\theta)}$ $y = \frac{Z}{\cos\left(Pan\theta-\left(\frac{Px}{2}\right)\right)*\sin\frac{Py}{2}} + \frac{Z}{\cos(Tilt\theta)}$

Equation B-7. Calculation for Bottom-Right corner of projection area after warp

Bottom-Left: $x = \frac{Z}{\cos\left(Tilt\theta-\left(\frac{Py}{2}\right)\right)*\sin\frac{Px}{2}} + \frac{Z}{\cos(pan\theta)}$ $y = \frac{Z}{\cos\left(Pan\theta-\left(\frac{Px}{2}\right)\right)*\sin\frac{Py}{2}} + \frac{Z}{\cos(Tilt\theta)}$

Equation B-8. Calculation for Bottom-Left corner of projection area after warp

Step 3: Calculate how to warp the projected image to fit the target size defined in step 1.

The third and final step of the correction process is to apply the affine transformation. This is achieved in two stages.

The first stage is to calculate to what percent each corner of the warped project needs to be reduced by to bring it to the corresponding corner of the target image. For example, on the x axis if the top-right corner of the warped projection area is 100cm from the centroid and the corner of the target image size is 40cm from the centroid, then the distance of the corner from the centroid has to be reduced by 60%.

The second stage is to apply these percentage corrections to the actual image being projected. This process is presented in Figure B-5, where the outline represents the shape of the image that would normally be projected and the image inside that outline is the image after the correction has been applied. This figure shows how each corner of the image is brought closer to the centroid by a certain percentage on both the x and y axes (demonstrated on the Top-Right corner, using the same percentage reduction present in Figure B-5).

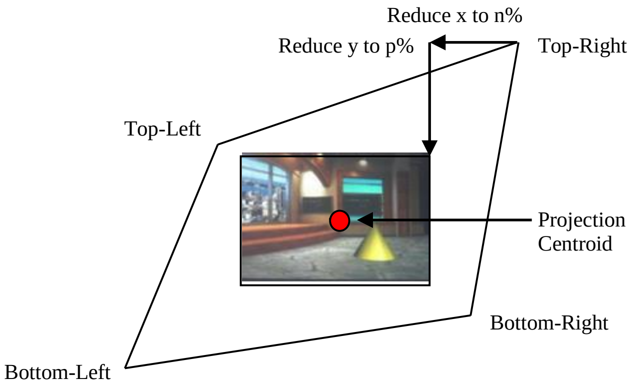


Figure B-5. Illustration of corrected image inside warped projected area

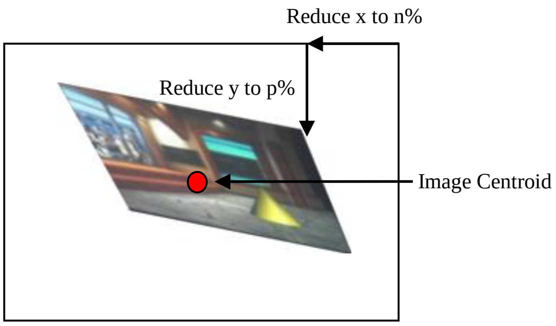


Figure B-5. The image warped for correction.

The calculations are presented in Equation B-9 to Equation B-12, where the corner locations of the warped projection area are denoted as TR = Top-Right, TL = Top-Left, BR = Bottom-Right and BL = Bottom-Left. When the suffix ‘t’ is placed after the corner notations, it represents the corner location of the target image. $\frac{\mathrm{Imgx}}{2}$ and $\frac{\mathrm{Imgy}}{2}$ describe the centre of the image on the X and Y axes respectively.

Top-Right: $\mathrm{x} = \frac{\mathrm{scrx}}{2} * \frac{\mathrm{TRx}}{\mathrm{TRtx}}$ $\mathrm{y} = \frac{\mathrm{scry}}{2} * \frac{\mathrm{TRy}}{\mathrm{TRty}}$

Equation B-9. Calculation of the X and Y co-ordinates for the Top-Right corner

Top-Left: $\mathrm{x} = \frac{\mathrm{scrx}}{2} * \frac{\mathrm{TLx}}{\mathrm{TLtx}}$ $\mathrm{y} = \frac{\mathrm{scry}}{2} * \frac{\mathrm{TLy}}{\mathrm{TLty}}$

Equation B-10. Calculation of the X and Y co-ordinates for the Top-Left corner

Bottom-Right: $\mathrm{x} = \frac{\mathrm{scrx}}{2} * \frac{\mathrm{BRx}}{\mathrm{BRtx}}$ $\mathrm{y} = \frac{\mathrm{scry}}{2} * \frac{\mathrm{BRy}}{\mathrm{BRty}}$

Equation B-11. Calculation of the X and Y co-ordinates for the Bottom-Right corner

Bottom-Left: $\mathrm{x} = \frac{\mathrm{scrx}}{2} * \frac{\mathrm{BLx}}{\mathrm{BLtx}}$ $\mathrm{y} = \frac{\mathrm{scry}}{2} * \frac{\mathrm{BLy}}{\mathrm{BLty}}$

Equation B-12. Calculation of the X and Y co-ordinates for the Bottom-Left corner

Appendix C. EXTENDING SCAMP TO MULTIPLE PLANAR SURFACES

To determine which surface the actor is looking towards, a profile for each surface must be constructed. This profile defines each planar surface as the angle between the location of the actor and the two corners either side of it (e.g. S1 is the surface that exists between Corner1 (C1) and Corner2 (C2) in Figure C-1).

To achieve this, the geometry presented in Figure C-1 is used. Here each corner is represented as C_N (C $_{1,}$ C_2...), the planar surfaces are represented by S_N, the location of the tracking camera is represented by T, the actor is represented by A.

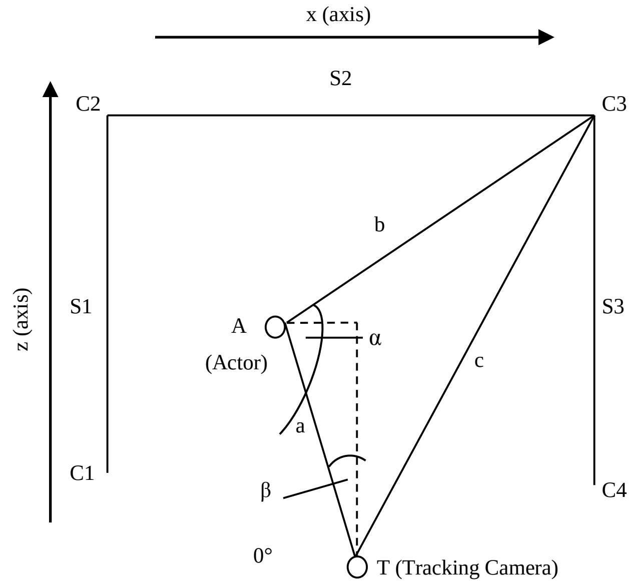


Figure C-1. Geometry for multiple planar surfaces (an example of measuring the angle to C3 is presented here)

The method of calculating the angle between the actor and each corner is essentially the sum of two angles, denoted as α and β in this case (both presented in Equation B-10 and Equation B-11 respectively). α is the angle between the actor and the target corner from the tracking camera. β is the angle between the actor and the tracking camera.

To calculate α the lengths of sides a, b and c must first be calculated using Equation C-3. Then using Equation C-2 angle α can be calculated. To calculate angle β, Equation C-2 is used.

$$a = \sqrt{(T_x - A_x)^2 + (T_z - A_z)^2}$$

Equation C-3. Calculation of side 'a' leng

$$b = \sqrt{(A_x - C_{N_x})^2 + (A_z - C_{N_z})^2}$$

Calculation of side 'b' length

$$c = \sqrt{(T_x - C_{N_x})^2 + (T_z - C_{N_z})^2}$$

Calculation of side 'c' length

$$\propto = \cos^{-1}\left(\frac{(a^2 + b^2) - c^2}{2ab}\right)$$

Equation C-2. Calculation of α

$$\beta = \tan^{-1}\left(\frac{T_x - A_x}{T_z - A_z}\right)$$

Equation C-2. Calculation of β

The angle between the actor and the corner is calculated using Equation C-4. If C_N is to the left of the actor (as defined by the x axis), the angle to the corner from the actor is the sum of α and β. If the C_N is to the right of the actor, then the sum of α and β is subtracted from 360 to create a continuous geometry.

```
If $C_{Nx} < Ax$:
    $C_N\theta = \propto + \beta$
If $C_{Nx} > Ax$:
    $C_N\theta = 360 - (\propto + \beta)$
```

Equation C-4. Calculation of angle between the actor and any corner.

The surface that the actor is looking towards is then determined using the logic presented in Figure C-2, where each planar surface is defined as being between two of the corners and the angle that actor is facing (Aθ). If the actor is facing within the angle of two corners the corresponding surface can be found.

```
If $C_1 < A\theta < C_2$:
    ScaMP projects to S1
If $C_2 < A\theta < C_3$:
    ScaMP projects to S2
If $C_3 < A\theta < C_4$:
    ScaMP projects to S3
```

Figure C-2. Logic for determining which surface the actor is looking towards.

By following this method, it should be possible to detect which planar surface the actor is looking towards and orientate ScaMP to project onto that surface, using the method proposed in chapter 4.3.4. There the head location is measured using a single tracking camera placed off set, and the various geometric equations are based on a central location of the relevant planar surface instead of the tracking camera location (these geometric equations are presented in section 4.3.2.1. Head Tracking and Calculation of Gaze Point, page 55).

Appendix D. TRAINING ROUTINE

The training routine took each actor through the full range of interaction tasks they would experience. The training task proceeded in the following stages.

1. A 2D square virtual object is placed on the monitor with two red circles on the left and right surfaces, with a message that reads "Please place your hands in the red circles when you see them during the training and test"
2. The actor was instructed on how to complete the animated task using the following steps.
    - They were taught the objectives of the task, with the following three messages displayed:
        - "The first test will feature a box that will move along an animated path from start to finish"
        - "You need to keep your hands on the side of the box"
        - "Try to keep your hands on the sides of the box, as though you are holding it"
    - The actor was informed that the task starts when the countdown animation reaches 0 with a message that read "The animation task will begin when the counter reaches zero"
    - The actor had to complete two tasks where they had to follow the virtual object along a horizontal path, first right and then left
3. The actor was introduced to the animated dot that indicates the velocity of the virtual object.
    - The actor was shown the message "The dot shows the object's speed"
    - The actor had to complete a task where they aimed to match the speed of the virtual object.
4. The actor was introduced to the vertical axis that the virtual object can travel in.
    - The actor was shown the message "The box will also move vertically"
    - The actor had to complete two tasks where they followed the animated box along a vertical path, first down and then up.
5. The actor was introduced to the different speeds that the virtual object can travel at.
    - The actor was shown the message "The virtual object will move at different speeds indicated by the animated dot"
    - The actor had to complete two tasks where they followed the animated box along a horizontal path from left to right at the slowest and then the fastest speed conditions.
6. The actor was introduced to the "top/bottom" hand placement.
    - The actor was shown the message "If the circles appear above and below you must hold the top with the right hand and bottom with left hand"
    - The actor had to complete two tasks where they followed the animated box along a horizontal path from left to right at the slowest speed and the highest speed.
7. The actor was introduced to the interactive virtual object tasks.

- The actor was shown the message "The second test is interaction placing your hands inside box will allow you to move it.", followed by "You must match the speed of the dot"
- The actor had to complete 8 tasks using the interactive virtual object that covered the range of conditions.

8. The actor was introduced to the no-object tasks.
    - The actor was shown the message "The third test features no-object you will mime the movement of an animated object path"
    - The actor had to complete 8 tasks with no virtual object that covered a range of conditions
9. The actor completed a comprehensive set of 96 sample tasks that represented many permutations of the conditions they would be exposed to in the final experiment

Appendix E. Viewer Perception Pilot Study

The pilot test established that the Single Stimulus method of perceived image quality evaluation as described in the BT.500 guidelines is suitable for measuring the perception of MDOS errors in the virtual studio. The objectives of this study were:

- To confirm that the Single Stimulus method of for rating perceived quality is transferable to assessing the perception of interaction errors in the virtual studio.
- To inform on the conditions of the final test.

E.1. Methodology

6 observers were shown a video presentation of 18 videos in a Single Stimulus format, consisting of a series of 3 stabilising videos followed by 15 video sequences shown in one of two pseudo-random orders. Each video sequence depicts an actor moving a basic 2D rectangular virtual object with a plain texture from the left side of the screen to the right side. An example of this interaction is shown in Figure E-1.

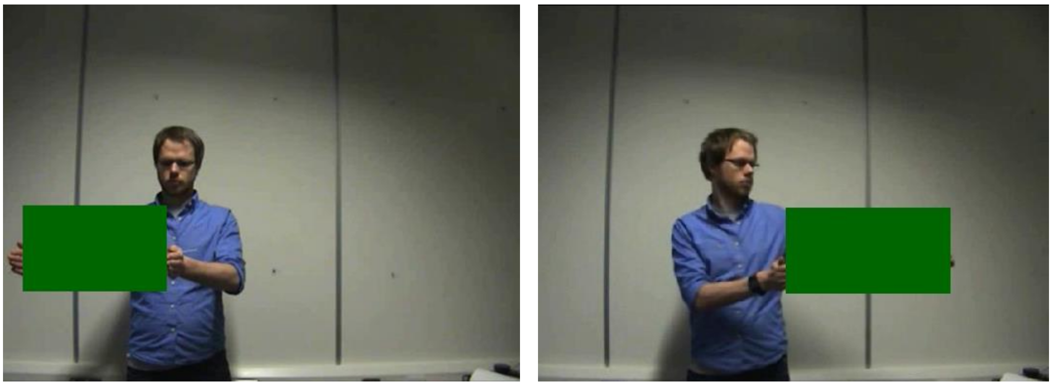


Figure E-1. Example of an interaction video from the pilot test.

The test is conducted at a standard PC work station, with the observers situated approximately 50cm from the monitor. It is important to note this does not reflect the Preferred Viewing Distance (PVD) (ITU-R, 1990) for audiences watching visual media (which is a suggested arrangement for this method), but it is applicable when considering it as a typical arrangement in the consumption of online media.

Video production

The source videos were recorded using a Logitech C120 camera and the video sequences were produced using Adobe Flash CS4. Two types of video sequences were included as part of the pilot test:

Static Object Size:

Video – These video sequences depict an MDOS error that does not change throughout the interaction. The distance between the hands is static, as is the size of the virtual object, thus the error itself remains constant. To maintain consistency all the video sequences use a single source video. The errors are depicted for different magnitudes of overestimation and underestimation by using different size objects. These videos also include one reference where the object fits the hands exactly.

Observation - Results from this pilot experiment has allowed the trends in the perception of errors in the interaction to be identified.

Adapting Object Size:

Video – These videos depict an alternative system. During the interaction, the distance between the actor's hands expands from an initial distance to a resting distance of 150px, during which time the object changes in size to match. Consequently, any gaps that would otherwise be caused by the deviation are no longer present (as they would be with the Static Object Size). The videos depict 5 different magnitudes of object growth and require a unique source video to be recorded for each condition presented (similar to the Matched adaptation method described in section 8.3.1).

Observation – Results from this experiment will be compared to the results of the equivalent errors for the static size object (i.e. the amount of growth to the equivalent overestimation size). If a comparison between the trends can be drawn, it will help confirm that this approach is effective in testing whether two different interaction systems can be compared using this method.

It is important to provide justification for using a single source video where possible when creating the video sequences used for testing. As stated, the Static Object Size videos were created using a single source video and the size of the object was the variable factor. Although this is not precisely representative of the error measured in Chapter 7, where the variable factor is the distance between the hands and the object was one of three fixed sizes, it will present a more viable approach to replicating the overlap and underlap errors as it removes the inconsistencies that would be present if many videos were recorded.

Producing video sequences where the distance between the hands is the variable factor would require multiple source videos to be recorded. This is not practical as the source videos ideally require a high degree of uniformity between them and producing multiple source videos would result in numerous differences between them that would compromise the uniformity and consequently the analysis of results across those conditions (i.e. differences in interaction speed, slight inaccuracies in hand motion, etc).

For the Adapting size error it was not possible to use a single source video for each video sequence, as the change in distance between the actor's hands was different for each video sequence. Therefore multiple sources had to be used, although care was taken to ensure a high degree of uniformity by selecting the most similar videos between many takes. This process was found to be time-consuming and error prone, so it could not be implemented for the production of all video sequences in the experiment proper.

E.2. Results

The results of the Static Size and Adapting Size object studies are presented in Figure E-2. The results from the Static Object Size (brown line) demonstrate that as the size of the MDOS errors becomes more extreme the likelihood of the observers perceiving them increases, as reflected by the lower MOS. This provides a strong indication that this method is transferable to the assessment of estimation errors in the interactive virtual studio, as it is possible to detect that observers became more aware of the MDOS error as the magnitude of it increased.

The results from the Adapting Size study (blue line) illustrate that when the object is subjected to more extreme amounts of growth the MOS is also lower, indicating that the observers are aware of the object size change. One anomalous result that does not follow this trend was recorded for the 20px object growth. From a later inspection of the video sequences, the origin of this outlying result is believed to be due to an irregular source video, which was a consequence of the need to record multiple source videos.

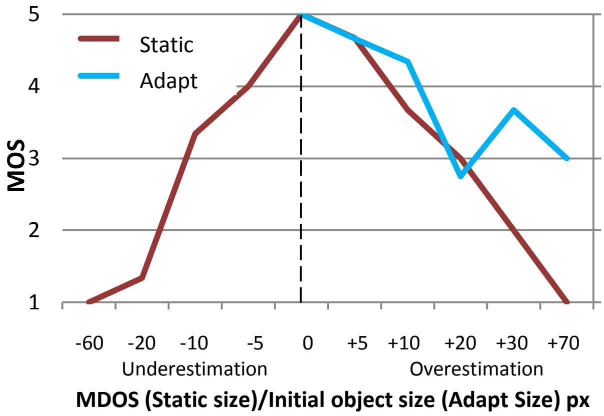


Figure E-2. Comparison of Static and Adapting Object Size

When comparing the two studies it is assumed that if the size of the objects did not change in the Adapting Object Size videos, then it would produce the equivalent visual result of the MDOS for the Static Size Object. It is observed that as the magnitude of the errors become more extreme, the degradation in MOSs for the Adapting Size Object happens at a considerably lower rate than the equivalent errors of the Static Size object. From this observation it is possible to conclude that the Adapting the size of the object produces a more convincing interaction. The ability to make this comparison indicates that the Single Stimulus method is a viable way of comparing two systems.

E.3. Discussion

The first objective was to confirm that the Single Stimulus method is transferable to assessing interaction errors in the virtual studio. As demonstrated in the results section (8.4.2), the pilot test confirms that this method does appear to provide meaningful data on the perception of the MDOS error in the virtual studio and allows the trends to be analysed. In addition, the method also allows the two modalities to be compared. Consequently we believe this method is transferable for use in the main test, both for the assessment of trends and the comparison of different systems.

The production quality of the videos for the pilot test is a key concern, where two issues exist:

- The source videos were low resolution and suffered from compression artefacts. This is not representative of standard television broadcasts, which do not suffer from these issues to the same extent. For the main test the source videos will be raised to that of broadcast quality and minimal compression will be applied to them throughout all stages of production.
- The virtual object is not sufficiently detailed. Virtual objects in the virtual studio often mimic the visual properties of real objects (i.e. Have a texture and exist in three dimensions). This is not reflected in nature of the object used for the pilot test, which was 2D and possessed no texture. A 3D object with a photorealistic texture will be used for the main test.

Recording of unique source videos. Video sequences in the main test where the object size adapts to the actor's hands will require unique video sources to be recorded. As seen by the effect the poor quality video had on the results, a high degree of uniformity between the video sequences is required for meaningful conclusions to be drawn. Videos that do not conform are at danger of receiving a misleading MOS, thereby obscuring the observed trends (as shown in Figure E-2). When producing source videos for the main test some solutions will be implemented to ensure a high degree of uniformity, many interaction attempts will be recorded for each condition and the ones that conform closely to each other will be selected. If no appropriate source video is recorded for a video sequence, it would be appropriate to forgo presenting that particular video sequence altogether as the results may obscure the trend.

Assessment of occlusive properties. From further analysis of the videos, it is believed that the quality of occlusion could also play a crucial role in the perception of interaction plausibility, so a realistic occlusion will be presented in the main test.

Arrangement. With regards to the arrangement of the main test, two changes will be made:

- More videos sequences will be presented to obtain a richer depth of information. These include using multiple hand distances (reflecting the object sizes from chapter 7) for the Static Size video sequences to analyse whether any observed trends are proportional to the hand distance or absolute.
- The viewing distance will be based on the PVD recommendations. Using a desktop based arrangement was valid for the pilot test, although the main test will use an arrangement that will take the PVD into account as it better represents a home viewing environment.