

Trabajo Fin de Grado

Percepción *crossmodal* en entornos inmersivos
Crossmodal perception in immersive environments

Autor

Marcos Allué Visa

Directora: Ana Serrano Pacheu

Ponente: Belén Masiá Corcoy

Escuela de Ingeniería y Arquitectura
2016



DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

D./D^a. Marcos Allué Visa

con nº de DNI 18055185-R en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado _____, (Título del Trabajo)

Percepción crossmodal en entornos inmersivos

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 24 de noviembre de 2016

Fdo: Marcos Allué Visa

Resumen

Durante los últimos años, la realidad virtual y aumentada han cobrado un gran interés. La reaparición de estas tecnologías ha sido posible debido a la proliferación de dispositivos de realidad virtual (HMDs por sus siglas en inglés) de bajo coste, que ahora son accesibles al público gracias a los recientes avances en hardware y el abaratamiento de la potencia computacional.

El concepto de *realidad virtual* hace referencia a la creación y visualización de contenido virtual intentando simular la manera en la que observamos el mundo real. Estos entornos generan en el usuario una sensación de inmersión, contribuyendo a enriquecer su experiencia. Por otro lado, el concepto de *realidad aumentada* hace referencia a la visualización de contenido virtual en el mundo real, con el objetivo de hacer el contenido real y virtual indistinguible. Sin embargo, todavía hay importantes obstáculos en ambas tecnologías que impiden el desarrollo de más aplicaciones y reducen la calidad visual de los resultados como, por ejemplo, baja resolución espacial, aberraciones cromáticas, o capacidades de procesamiento limitadas que producen desfases en la visualización (*lag*), lo que ocasiona mareos.

En este proyecto se propone el uso de la percepción humana para intentar aliviar los problemas mencionados anteriormente y mejorar la inmersión. Dentro del área de la percepción en entornos inmersivos, se ha escogido estudiar el área de la percepción *crossmodal*, esto es, la interacción de diferentes sentidos cuando perciben el entorno. En particular, se analiza la *influencia del sonido en la percepción de estímulos visuales*.

Este trabajo contiene un estudio de la teoría de la percepción *crossmodal* aplicada a entornos inmersivos, incluyendo la implementación de un sistema completo para realizar estudios de usuario con distintos estímulos, el diseño de dichos estímulos (entornos virtuales), y el subsecuente análisis del comportamiento de los usuarios. Finalmente, también se ha implementado un sistema de realidad aumentada integrado en un HMD para estudiar la influencia de entornos reales en la percepción *crossmodal*.

Abstract

In the last few years, virtual and augmented reality have gained a great interest. The reappearance of these technologies has been possible due to the proliferation of low-cost head-mounted displays (HMDs) now accessible in a consumer-level thanks to recent advances in hardware and lowering costs of computational power.

The term *virtual reality* refers to the creation and display of virtual environments as if they were the real world. These environments generate in the user the feeling of *immersion*, which greatly enriches user experience. On the other hand, the term *augmented reality* refers to displaying virtual content in the real world, with the goal of making the real and the virtual content indistinguishable. However, there are still important stumbling blocks in both technologies that hinder the development of more applications and reduce the visual quality of the results; examples include limited spatial resolution, chromatic aberrations, or limited processing capabilities leading to lag, and subsequent motion sickness.

In this project, we propose to use human perception in order to try to overcome the aforementioned problems and improve immersion. Within the much-studied area of perception in virtual environments, we chose to look into the less explored area of *crossmodal* perception, that is, the interaction of different senses when perceiving the environment. In particular, we looked at *the influence of sound on visual perception in a virtual reality scenario*.

This work contains a study of the *crossmodal* perception theory applied to immersive environments, including the implementation of the entire pipeline for carrying out user studies with different stimuli, the design of such stimuli (virtual environments), and the subsequent analysis of user behavior. It also includes the implementation of an augmented reality system integrated in a HMD for studying the influence of real environments in the *crossmodal* perception.

Agradecimientos

Me gustaría agradecer a Ana Serrano y a Belén Masiá la paciencia que han tenido conmigo y la ayuda que me han prestado a la hora de realizar este proyecto, además de la oportunidad que me han dado para aprender multitud de cosas nuevas, no solo de manera académica sino también personal. Me gustaría dar las gracias a la gente del Graphics and Imaging Lab por ayudarme a realizar este proyecto y acogerme como a uno más. También a Diego Gutiérrez por introducirme en el mundo de la informática gráfica. Además me gustaría dar las gracias a Manuel Bedia por su colaboración en este proyecto.

Me gustaría agradecer de manera especial a todos los compañeros y amigos que participaron en los experimentos.

Por otro lado agradecer a mis padres y mis amigos por apoyarme durante toda la carrera.

Índice

1. Introducción	9
1.1. Objetivos, alcance y contexto	9
1.2. Organización del proyecto	10
1.3. Planificación	11
2. Marco contextual	12
2.1. Realidad virtual y aumentada	12
2.2. Percepción <i>crossmodal</i>	13
2.3. Elección de los estímulos	14
2.3.1. Introducción	14
2.3.2. Estudio de posibles estímulos	14
2.3.3. Estímulos escogidos	15
3. Entorno de trabajo	18
3.1. Oculus Rift Development Kit 2	18
3.1.1. Introducción	18
3.1.2. Características	18
3.2. Diseño de los experimentos en Unity3D	20
3.2.1. Introducción	20
3.2.2. Consideraciones generales	20
3.2.3. Creación del entorno	23
3.2.4. Controlador	24
3.3. Realidad virtual	26
3.3.1. Integración de la realidad virtual en Unity3D	26
3.3.2. Experimento 1	27
3.3.3. Experimento 2	27
3.4. Realidad aumentada	27
3.4.1. Características de las cámaras	28
3.4.2. Montaje estéreo de las cámaras	28
3.4.3. ARToolkit	30
3.4.4. Integración de ARToolkit en Unity3D	31
3.4.5. Calibración de las cámaras	34
3.4.6. Experimento 3	36
4. Desarrollo de los experimentos	38
4.1. Diseño de los estímulos	38
4.1.1. Descripción de los estímulos	38

4.1.2.	Traslado de los estímulos a Unity3D	39
4.2.	Experimento 1	42
4.2.1.	Objetivo del experimento	42
4.2.2.	Bloques del experimento	42
4.2.3.	Procedimiento	43
4.3.	Experimento 2	44
4.3.1.	Objetivo del experimento	44
4.3.2.	Bloques del experimento	45
4.3.3.	Procedimiento	47
4.4.	Experimento 3	47
4.4.1.	Objetivo del experimento	47
4.4.2.	Procedimiento	48
5.	Análisis y resultados	51
5.1.	Pre-procesado de los datos	51
5.1.1.	Recolección de los datos	51
5.1.2.	Descarte de espurios	51
5.1.3.	Análisis de los datos	52
5.2.	Experimento 1	53
5.3.	Experimento 2	54
5.4.	Experimento 3	55
6.	Conclusiones y trabajo futuro	57
	Bibliografía	59
A.	Trabajo presentado en el Congreso Español de Informática Gráfica (CEIG)	61

Índice de figuras

1.1. Diagrama de Gantt con las tareas realizadas	11
3.1. Oculus Rift Development Kit 2 y el sensor infrarrojo	19
3.2. Estructura genérica de un motor gráfico	22
3.3. Diagrama de clases de los objetos de Unity3D	23
3.4. Diagrama del controlador del experimento	25
3.5. Montaje en los tres modos de las cámaras	30
3.6. Soporte de las cámaras	30
3.7. Marcador de realidad aumentada	31
3.8. Ejemplo del funcionamiento de la aplicación de realidad aumentada	31
3.9. Figura de las cámaras que siguen el marcador	33
3.10. Diferencia entre la calibración manual y ARToolkit para seguir un marcador	33
3.11. Esquema de la aplicación de realidad aumentada	34
3.12. Imagen del tablero de calibración distorsionado y con la distorsión corregida	35
4.1. Descripción gráfica de ángulo visual	38
4.2. Percepción del movimiento de las esferas rebotando o cruzándose	39
4.3. Proyección de la escena generada por Unity3D en la pantalla	40
4.4. Representación de las medidas de los estímulos en un entorno virtual	41
4.5. Estado inicial del primer experimento	42
4.6. Ejemplo de una escena del entrenamiento	43
4.7. Diapositiva que aparece cuando se debe escoger en el test	44
4.8. Todas las escenas del experimento 2	45
4.9. Justificación del aumento lineal de la velocidad	46
4.10. Ejemplo de estímulo visual en realidad aumentada	48
4.11. Diapositiva que guía al participante en el experimento de realidad aumentada	49
5.1. Porcentajes agregados del Experimento 1	54
5.2. Porcentajes no agregados del Experimento 1	54
5.3. Porcentajes agregados del Experimento 2	55
5.4. Porcentajes agregados del Experimento 3	56

1. Introducción

1.1. Objetivos, alcance y contexto

En los últimos años ha resurgido el concepto de realidad virtual. Esto es debido a una serie de factores que han permitido la creación de dispositivos de tipo HMD (*Head Mounted Display*) a nivel de consumidor como Oculus Rift, PlayStation VR o HTC Vive. Uno de los principales factores que han permitido la creación de estos dispositivos es el gran incremento en potencia computacional y su abaratamiento, incluyendo la posibilidad de tener pantallas alta resolución y capaces de mostrar una gran cantidad de fotogramas por segundo, o la posibilidad de integrar pequeños y precisos sensores en los dispositivos, entre otras.

El término realidad virtual hace referencia a la creación y visualización de entornos virtuales que se perciben como si fueran el mundo real. El término utilizado para definir que un usuario está dentro de un entorno virtual y lo percibe como real se denomina *inmersión*.

A la par que la realidad virtual también ha resurgido el término de realidad aumentada, que consiste en integrar elementos virtuales en el mundo real. Este es un entorno mucho más complejo ya que el reto radica en que el usuario sea incapaz de distinguir objetos reales y virtuales.

Tanto la realidad virtual como la aumentada tienen múltiples utilidades, no solo de entretenimiento, también pueden ser útiles en campos como la medicina, la arquitectura, ayudando a visualizar cualquier tipo de prototipo que no se ha creado todavía, incluso para simulación, siendo el caso de los simuladores de vuelo que se usan para entrenar pilotos. La realidad virtual actual se apoya en la potencia del hardware, pero esto es muy costoso económicamente para los usuarios aun con el aumento notable de la potencia hardware en los últimos años. Además, la realidad virtual todavía tiene bastantes problemas derivados de la tecnología. En este proyecto se estudia la implicación de la percepción humana en la generación de contenidos para realidad virtual y aumentada para mejorar la inmersión en estos entornos sin aumentar el coste computacional de generarlos. En el pasado, la percepción ha sido útil en multitud de problemas de informática gráfica como mejora de render basado en percepción [1], edición y adquisición de materiales [2], o visualización de contenidos [3]. Una buena guía de la aplicación de la percepción en informática gráfica se puede encontrar en el curso de McNamara et. al. [4].

El objetivo en este proyecto es explorar la percepción en entornos inmersivos, para en el futuro poder solucionar algunos problemas de los entornos inmersivos derivados de la potencia hardware necesaria para crearlos, intentando mejorar la inmersión. Para ello se tienen en cuenta conceptos perceptuales del sistema visual humano. En concreto este proyecto se apoya en efectos

derivados de la percepción intermodal o *crossmodal* (terminología comúnmente aceptada en la literatura que se utilizará a lo largo del proyecto). La percepción *crossmodal* es un fenómeno que describe interacciones entre varias modalidades sensoriales. Los sentidos involucrados en este estudio son el sentido de la vista y el sentido auditivo, y la influencia que puede tener este último sobre el primero en determinados escenarios. En este proyecto se desarrolla un experimento de percepción *crossmodal* en varios entornos inmersivos y se hacen estudios de usuario para comprobar el efecto de las interacciones entre los sentidos visuales y auditivos, y cómo afectan estas interacciones a la hora de percibir una serie de estímulos. El alcance de este proyecto incluye:

- Estudio del estado del arte en realidad virtual y sus limitaciones.
- Estudio del estado del arte en percepción *crossmodal*. Análisis de posibles efectos de interacción *crossmodal* que puedan contribuir a aliviar las limitaciones actuales en las tecnologías de realidad virtual.
- Elección y adaptación del diseño de un experimento de percepción *crossmodal* donde interactúan los sentidos visual y auditivo. Implementación en un entorno de realidad virtual, realización del experimento mediante un estudio de usuario, y análisis de los resultados.
- Diseño de varios escenarios en realidad virtual para analizar la influencia del realismo del entorno en la percepción del efecto *crossmodal*. Realización de estudios de usuario y análisis de los resultados.
- Adaptación del diseño e implementación del experimento en un entorno de realidad aumentada (incluyendo modificaciones hardware) para analizar la influencia de presentar estímulos en entornos reales en el efecto de percepción *crossmodal*. Realización de estudios de usuario y análisis de los datos.

Una parte del trabajo desarrollado en este proyecto ha sido presentada al Congreso Español de Informática Gráfica (CEIG), y ha sido seleccionada uno de los dos mejores trabajos e invitado a someter una versión extendida en la revista “Computer Graphics Forum” (JCR Q1 en el campo).

1.2. Organización del proyecto

Esta memoria se divide en varias partes o capítulos, en el Capítulo 2 se muestra el marco contextual de los conceptos de realidad virtual, aumentada y percepción *crossmodal*. Además también se hace un estudio de posibles experimentos de percepción *crossmodal* y se justifica la selección del experimento realizado en este proyecto. En el Capítulo 3 se describen los dispositivos hardware que se usan en este proyecto, además del software usado y la implementación que se ha desarrollado para lograr diseñar los experimentos que aparecen en este proyecto. En el Capítulo 4 se describen las características de todos los experimentos desarrollados. En el Capítulo 5 se detalla cómo se han analizado los datos, y se discuten los resultados de los estudios de usuario realizados en los experimentos. Finalmente en el Capítulo 6 se encuentran las conclusiones que se sacan de este proyecto y el posible trabajo futuro.

1.3. Planificación

Durante la duración de este proyecto, desde marzo de 2016 hasta noviembre de 2016, se han realizado múltiples tareas entre ellas destacan: investigación sobre realidad virtual para aprender los conceptos básicos, investigación sobre percepción *crossmodal*, implementación de los experimentos que se realizaron y análisis los resultados de los experimentos. En la Figura 1.1 se puede ver el diagrama de Gantt de todo el trabajo que se ha realizado a lo largo del proyecto.

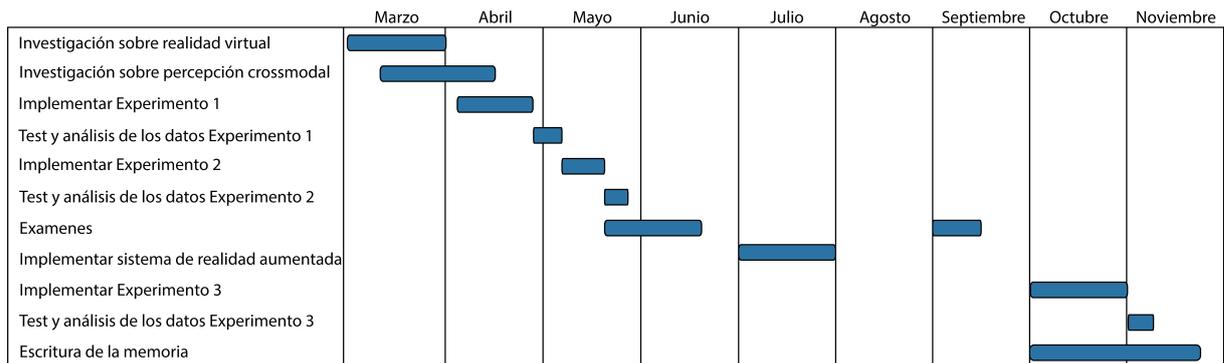


Figura 1.1: Esta figura muestra el diagrama de Gantt de las tareas realizadas.

2. Marco contextual

En esta sección se detallan los conceptos básicos de la realidad virtual, la realidad aumentada y la percepción *crossmodal*. Además se hace un estudio de posibles estímulos para realizar los experimentos que se proponen en este proyecto.

2.1. Realidad virtual y aumentada

A lo largo de la historia de la informática y la tecnología los conceptos de realidad virtual y aumentada han ido apareciendo en múltiples ocasiones. El concepto de realidad virtual hace referencia a entornos generados por ordenador donde el usuario tiene una sensación de estar inmerso. En cambio la realidad aumentada hace referencia a combinar la percepción del mundo real con objetos virtuales generados por ordenador. Esto dos campos han surgido con fuerza en la actualidad debido al aumento de potencia del hardware y sobre todo a su reducción de tamaño, que permite tener dispositivos con mucha potencia computacional en la palma de la mano, véase el caso de los smartphones o teléfonos inteligentes.

Uno de los puntos importantes de la realidad virtual es intentar simular el sistema de visión humano. Para reproducir contenidos en tres dimensiones se debe tener una disparidad binocular, esto significa que cada ojo debe tener una vista distinta de un objeto para tener la sensación de que éste tiene profundidad. En el caso de la realidad virtual esto se hace presentando una imagen diferente a cada ojo.

Actualmente la mayor parte de dispositivos que han aparecido para realidad virtual son dispositivos de tipo HMD (*Head Mounted Display*). Estos dispositivos son cascos de realidad virtual, y todos ellos tienen una serie de componentes comunes que ayudan a simular un entorno de realidad virtual. Estos dispositivos intentan simular el sistema de visión de los seres humanos, para ello colocan una pantalla muy cerca de los ojos del usuario y la dividen por la mitad, permitiendo que cada ojo solo vea una mitad de la pantalla, de esta manera se puede mostrar a cada ojo una imagen distinta. Con la introducción de estas dos imágenes se puede simular la visión estereoscópica introduciendo tres dimensiones al sistema visual. Actualmente todos los HMD introducen dos lentes entre los ojos y la pantalla para que se vea la pantalla curvada, esto se hace para simular el sistema de visión humano. Una de las características importantes de los HMD actuales es la capacidad de saber si el usuario gira la cabeza, para ello se utilizan una serie de sensores que recogen la información de los movimientos.

La realidad aumentada se puede visualizar de varias formas. Por ejemplo en un dispositivo

como un teléfono móvil o una tablet, se captura la realidad con la cámara y luego se superponen los objetos virtuales. También existen gafas de realidad aumentada que consisten en unas gafas en las que en ciertas partes de las lentes se colocan los objetos virtuales con un efecto estereoscópico, intentando tener una disparidad entre las dos lentes, para poder visualizarlo en el mundo real en tres dimensiones. También en algunos artículos de investigación o proyectos como en el caso de Steptoe en su proyecto AR-Rift [5] implementan un dispositivo de realidad aumentada con un HMD y dos cámaras.

Colocar los objetos virtuales en un entorno de realidad aumentada puede realizarse de diversas formas. Una de ellas es el seguimiento de marcadores como códigos de barras o imágenes 2D sencillas, en las cuales las cámaras identifican los patrones y marcadores de realidad aumentada y colocan allí el objeto virtual. Otra forma es utilizar el GPS y la brújula del dispositivo donde se quiera reproducir la realidad aumentada y con eso poder colocar objetos en puntos de interés. En la actualidad se están desarrollando elementos como gafas de realidad aumentada que colocan los objetos donde el usuario quiera haciendo que encajen, esto es gracias a que son capaces de detectar los objetos del mundo real para luego colocar allí los objetos virtuales, un ejemplo de estos dispositivos son las Hololens de Microsoft.

2.2. Percepción *crossmodal*

Hoy en día, una corriente popular en neurociencia mantiene que el cerebro humano está estructurado en un gran número de áreas con la información muy separada [6]. Ésta perspectiva asume que el procesamiento mental y la percepción, como también las emociones e intenciones, están limitadas por el proceso neuronal dentro del cerebro y confinadas en áreas particulares. En el mismo camino, a menudo se asume que la entrada de diferentes modalidades perceptuales es procesada en el cerebro de manera independiente y en diferentes regiones de éste [7].

Sin embargo, la afirmación de que la percepción que se tiene de un objeto por diferentes modalidades sensoriales se unifica también está extendida. Esto sugiere que la información procedente de diferentes modalidades sensoriales debe unirse de alguna manera en el cerebro para representar una sola percepción [8]. Ésta suposición es clave en las más recientes alternativas de las vistas neurodinámicas (como, por ejemplo, los enfoques corporal y sensoriomotor) con el fin de proponer sólidas alternativas explicativas a las perspectivas tradicionales e internalistas de la organización del cerebro [9, 10]. En estos modelos alternativos, los procesos de la percepción multisensorial y las diferentes modalidades sensoriales se entienden como estrechamente relacionadas a través de la integración flexible de las dinámicas del cerebro por medio de la aparición conjuntos transitorios de sincronización neuronal cuando surge una percepción unificada [11]. Por lo tanto, una completa comprensión de la percepción requiere el conocer las diferentes formas en las cuales una de las modalidades sensoriales impacta en otra creando ilusiones *crossmodales* [12]. Si se logra comprender las interacciones entre las diferentes modalidades sensoriales, se puede arrojar luz sobre los verdaderos mecanismos que soportan los procesos perceptuales.

Cabe destacar que, hasta hace poco, los principios neuronales de la integración multisensorial y las ilusiones *crossmodales* no han sido investigados en profundidad. La idea de que el cerebro está dividido en bloques dependiendo de las modalidades sensoriales ha sido tan fuerte que siempre se ha considerado que el sentido visual era independiente del resto. Sin embargo, en los

últimos años el interés en entender los fenómenos *crossmodales* y las ilusiones se ha incrementado sustancialmente [13]. Algunos de los estudios más profundos son los que involucran alteraciones entre los sentidos auditivos y visuales. Uno de los más conocidos es el efecto del ventrilocuismo (conocido como *ventriloquism effect* en inglés) que se refiere a la percepción de sonido que viene de una fuente distinta a la que realmente viene, forzado por la influencia del estímulo visual de un objeto que aparentemente habla [14]. Otro ejemplo bien conocido es el efecto McGurk (conocido como *McGurk effect* en inglés) [15] donde el movimiento de los labios de un sujeto implica cambios en la percepción del sonido que emite dicho sujeto.

En este proyecto se estudia los efectos de la percepción *crossmodal* en entorno inmersivos, los sentidos que se quiere involucrar son los sentidos auditivo y visual, y cómo afecta el sentido auditivo a la hora de percibir una serie de estímulos visuales.

2.3. Elección de los estímulos

2.3.1. Introducción

En este proyecto se busca combinar los entornos inmersivos introducidos anteriormente (realidad virtual y aumentada) y el concepto de percepción *crossmodal*. Para ello se ha decidido modelar un experimento de percepción *crossmodal* en estos entornos inmersivos. Los sentidos con los que se trabaja en este proyecto son el sentido auditivo y el visual. Además el hecho de usar un dispositivo HMD implica que al sentido visual se le añaden ciertos indicadores, como de profundidad y de sensación 3D. Este hecho hace pensar que el tipo de experimento con percepción *crossmodal* que más encaja en este proyecto es uno donde lo que se percibe por el sentido auditivo (a partir de ahora estímulo auditivo) influye en cómo se percibe el estímulo visual. Para encontrar los estímulos y que éstos tuvieran una base científica se recurre a buscar experimentos de percepción *crossmodal* en la literatura científica (en revistas, artículos y libros científicos sobre percepción y psicología).

En la búsqueda de posibles estímulos se encontraron varios experimentos que contenían estímulos que podían ser reproducibles en los entornos inmersivos. Todos los experimentos están generados por ordenador para un entorno no inmersivo (monitor estándar de ordenador), esto implica que se deben volver a diseñar para un entorno inmersivo.

2.3.2. Estudio de posibles estímulos

A lo largo de la literatura sobre percepción *crossmodal* aparecen multitud de experimentos que exploran diferentes consecuencias de la interacción entre los sentidos visual y auditivo, como el sonido provocando la aparición de ilusiones visuales, influenciando en la percepción de un movimiento ambiguo, ayudando a detectar patrones de movimiento o ayudando a determinar el orden temporal de dos estímulos visuales. A continuación se van a detallar algunos estímulos que podrían ser interesantes para ser reproducidos en un entorno inmersivo.

Una influencia de un estímulo auditivo en lo que una persona percibe visualmente que se puede considerar es la llamada ilusión del doble flash. Esto está relatado en el trabajo de Shams

et. al. [16], en esta interacción el estímulo visual que aparece es un flash de luz, y el estímulo auditivo es un pitido corto. Cuando se muestra el estímulo visual se puede acompañar de uno o varios estímulos auditivos, que pueden provocar que una persona perciba que el estímulo visual aparece más veces de las que ha aparecido realmente. Estos estímulos se descartaron ya que mostrarlos en un entorno inmersivo no implicaba un cambio significativo a mostrarlos en un entorno no inmersivo, ya que aprovecha pocas de las diferencias que se introducen al usar un HMD. En este caso solo se utilizarían la profundidad que se percibe y la corta distancia a la pantalla.

Otra posible interacción entre estímulos a valorar es la influencia que tiene el sonido a la hora de determinar el orden temporal de dos estímulos visuales, esto se relata en el trabajo de Hairston en [14]. En este artículo se hace un estudio de cuándo un estímulo auditivo ayuda a discernir qué estímulo visual apareció antes. Además demuestra que un estímulo auditivo colocado en un momento concreto entre los dos estímulos ayuda a mejorar el tiempo de reacción de los participantes para contestar. Estos estímulos también se descartaron porque realmente aportan lo mismo que los anteriores al reproducirlos en un entorno inmersivo generado en un HMD.

La influencia del sonido también es significativa para determinar si existen patrones de movimiento en un conjunto de objetos sencillos como relatan Shams et. al [12]. En este trabajo los autores demuestran que de entrada introducir sonido mejora levemente la detección de estos patrones. Si se entrena a los participantes, unos con sonido y otros sin él, durante varias pruebas en varios días, entonces esta diferencia aumenta más. Esto también lo exploran sin ningún entrenamiento Kim et. al. [17].

Se puede explorar también influencias como la que tiene el sonido a la hora de percibir la intensidad de una fuente de luz como Stein et. al. [18]. En este trabajo los autores estudian la influencia que tiene un estímulo auditivo a la hora de percibir la intensidad lumínica de un estímulo visual. Aquí demuestran que el estímulo auditivo hace que aumente intensidad percibida por el participante al visualizar el estímulo visual, pero un aumento pequeño, que solo aparece en determinadas situaciones, en concreto cuando el usuario está con la atención fijada en los estímulos. Este experimento se ha descartado porque la aparición del estímulo auditivo no implica un gran cambio en la percepción de la intensidad lumínica.

2.3.3. Estímulos escogidos

En el trabajo de Sekuler et. al. [19] se trata la influencia que tiene el sonido en un movimiento ambiguo. Para ello tienen un estímulo visual y uno auditivo. El estímulo visual consiste en dos discos que van uno hacia el otro a una velocidad determinada, se cruzan y siguen moviéndose en la misma dirección. El estímulo auditivo consiste en un sonido breve. Tanto el estímulo auditivo como el estímulo visual aparecen bajo diferentes condiciones. El estímulo auditivo tiene cuatro posibles condiciones que consisten en no aparecer, aparecer en el momento de coincidencia de los discos, 150 milisegundos antes o 150 milisegundos después. El estímulo visual tiene tres posibles condiciones, que consisten en si los discos tienen un movimiento continuo, si tienen una pausa de un fotograma, o tienen una pausa de dos fotogramas. Las pausas del experimento medidas en fotogramas hacen referencia una medida que existe a la hora de mostrar videos, que es la tasa de fotogramas por segundo (fps), teniendo en cuenta que esto también depende del dispositivo

2. Marco contextual

donde se muestra. En este experimento el estímulo visual se puede percibir que los discos rebotan y cambian la dirección de movimiento, o que siguen el movimiento como si se cruzaran. Cuando aparece el estímulo auditivo aumenta el número de rebotes percibidos por los participantes en el experimento, introduciendo la idea de que el sonido puede influir en la percepción de un movimiento ambiguo.

Este experimento en un entorno de realidad virtual es interesante de reproducir debido a que introducir indicadores de profundidad pueden influenciar la percepción del movimiento. En particular, en este proyecto se implementa estos estímulos en tres experimentos distintos, dos en realidad virtual y uno en realidad aumentada.

3. Entorno de trabajo

Para la implementación de los experimentos que se quiere modelar con los estímulos descritos en la Sección 2.3.3 son necesarios varios dispositivos hardware. Uno de ellos es el Oculus Rift Development Kit 2, un dispositivo de tipo HMD preparado para la realidad virtual. También se usan dos cámaras para recrear un entorno de realidad aumentada. Para diseñar y modelar los experimentos se usa un motor gráfico, que es Unity3D. En este motor gráfico se diseñan e implementan los experimentos y los entornos de realidad virtual y realidad aumentada que se reproducen en el dispositivo HMD.

3.1. Oculus Rift Development Kit 2

3.1.1. Introducción

Para reproducir contenidos en realidad virtual hace falta un dispositivo compatible con estos. En los últimos años han aparecido multitud de prototipos y dispositivos comerciales capaces de reproducir estos contenidos. El dispositivo con el que se trabaja en este proyecto es el Oculus Rift Development Kit 2, que a partir de ahora se denominará Oculus.

El Oculus es un dispositivo de realidad virtual desarrollado por la compañía Oculus VR, propiedad de Facebook. Es un dispositivo de tipo HMD. Es la segunda versión de desarrollo de la compañía, las primeras unidades aparecieron en julio de 2014, y mejora bastante las características de la primera versión de desarrollo. En la Figura 3.1 se puede ver una imagen del dispositivo. Las especificaciones del Oculus aparecen descritas en la web del producto [20]. En esta memoria aparecen descritas aquellas especificaciones relevantes para el proyecto.

3.1.2. Características

El Oculus es un dispositivo es de tipo HMD que está compuesto por una pantalla, unas lentes, una cámara que sigue los movimientos del dispositivo y una serie de sensores.

La pantalla que usa el Oculus es una pantalla de resolución 1920 x 1080 píxeles, una resolución FullHD en un formato de 16:9, de tamaño 5,7 pulgadas. Se divide la mitad para cada ojo, teniendo resolución de 960 x 1080 píxeles y un formato de 8:9. Esta pantalla tiene un ratio de refresco de 75 Hz y se conecta al equipo donde está funcionando el Oculus mediante un cable HDMI. Justo delante de la pantalla del Oculus, hay dispuestas dos lentes, una para cada ojo. Estas lentes



Figura 3.1: En esta figura se puede ver el dispositivo HMD de realidad virtual llamado Oculus Rift Development Kit 2 que es el que se usa en este proyecto. Debajo a la derecha del Oculus, con un tamaño más pequeño, se puede ver el sensor infrarrojo con forma de cámara que ayuda a seguir los movimientos del Oculus.

distorsionan la imagen que reciben, por eso cuando se envía una imagen para que se muestre en el dispositivo se debe aplicar una distorsión de barril, que introduce una curvatura a la imagen más notable en las partes más alejadas del centro de la imagen. Cada ojo en el Oculus tiene un campo de visión (FOV por sus siglas en inglés) limitado por un valor vertical de 106 grados y 95 grados en el valor horizontal, manteniendo las proporciones de 8:9 para cada ojo.

El Oculus cuenta con varios sensores, un giroscopio que permite saber cuando el dispositivo rota, un acelerómetro que permite calcular la aceleración del dispositivo y un magnetómetro que ayuda a calcular la orientación con respecto al campo magnético de la tierra. Adicionalmente cuenta con una cámara que se coloca en frente del Oculus, que se puede ver en la Figura 3.1. La cámara es un sensor que se usa para saber la posición del dispositivo, es un sensor infrarrojo que sigue una serie de micro leds que están distribuidos por toda la parte frontal del Oculus, el rango de distancias ideal entre el Oculus y la cámara va de 0.5 metros a 2.5 metros. Todos estos sensores y la cámara sirven para calcular eficientemente y de manera precisa los movimientos del dispositivo. La cámara y los sensores están conectados y transmiten información al equipo donde está funcionando el Oculus mediante dos cables USB.

El Oculus es un dispositivo que necesita un equipo potente para funcionar correctamente ya que requiere una imagen por ojo y a una tasa de fotogramas por segundo (fps) de 75, además del seguimiento de los múltiples sensores que tiene. Necesita una tarjeta gráfica de gama media alta, en este proyecto se usa una Nvidia GeForce GTX 970, y un procesador de al menos 3.5 ghz por núcleo, el que se usado en este proyecto es un Intel i7-950 con 4 núcleos a 3.06 ghz por núcleo. Además se necesitan dos puertos USB y una salida de HDMI.

El Oculus necesita un software específico instalado en el equipo para funcionar correctamente, este software es el llamado Oculus Runtime, que contiene los drivers necesarios para la comunicación entre el hardware del Oculus y el equipo.

Por otro lado, para poder modelar aplicaciones para el Oculus es necesario disponer del Oculus SDK. Este software contiene funciones que permiten programar aplicaciones con las librerías de gráficos por ordenador OpenGL y DirectX. El núcleo de este SDK es la librería LibOVR que contiene las funciones necesarias para recibir la información de los sensores del

Oculus. También tiene las funciones necesarias para enviar las imágenes que se deben visualizar en cada ojo al Oculus.

3.2. Diseño de los experimentos en Unity3D

3.2.1. Introducción

En este proyecto se ha escogido usar Unity3D para generar contenidos para realidad virtual. Esta elección viene dada en gran parte por la facilidad de uso de Unity3D y su fácil configuración para generar y reproducir contenido en realidad virtual. Una opción a Unity3D podría ser Unreal Engine, un motor de videojuegos que también permite el uso de realidad virtual, pero posee una curva de aprendizaje más pronunciada que Unity3D. También se pueden generar gráficos para realidad virtual a más bajo nivel con librerías como OpenGL o DirectX usando directamente el OculusSDK. Sin embargo esto es mucho más costoso que usar cualquier motor gráfico, ya que en los motores gráficos hay muchos elementos y funcionalidades ya implementados.

Unity3D es un motor gráfico que apareció en 2005. Está disponible para generar contenido para muchas plataformas como PC, Linux, Mac, Android, iPhone, y contenido web entre otras. Unity3D es usado para generar imágenes por ordenador en tiempo real. Uno de los mayores usos de este tipo de motores gráficos, y por el que es más conocido, es la creación de videojuegos. Unity3D es muy utilizado gracias a que es fácil y cómodo de usar. Una de las premisas con las que se desarrolló Unity3D es la de dar acceso al desarrollo de videojuegos a la mayor parte de gente posible. Actualmente Unity3D se encuentra en la versión de desarrollo 5. En las últimas actualizaciones de esta versión se permite la creación de contenido para realidad virtual de la misma manera que se crean el resto de contenidos.

Para realizar este trabajo ha sido muy útil el uso de la documentación que proporciona Unity3D, disponible en su página web [21].

3.2.2. Consideraciones generales

La versión de Unity3D que se ha usado para este proyecto es la versión 5.3.4 de 64 bits, funcionando sobre un ordenador de sobremesa con Windows 7 también de 64 bits. En Windows Unity3D puede utilizar DirectX 11 para generar contenidos gráficos. Esta herramienta es una API desarrollada para ayudar a generar contenido multimedia, en concreto Unity3D usa Direct3D que facilita el uso de formas geométricas para crear gráficos, además de proporcionar soporte para usar procesadores gráficos. Al utilizar este tipo de tecnologías para renderizar contenido, Unity3D puede dar distintos tipos de apariencia a los objetos, usar texturas, shaders, y tecnologías de generación de gráficos en tiempo real que intentan conseguir una apariencia realista.

Tiene dos vertientes de desarrollo de contenido, una es la vertiente de desarrollo en dos dimensiones, y otra es la de desarrollo en tres dimensiones, variando entre estas dos casi todos los elementos que generan las interacciones físicas y geométricas del motor, como por ejemplo el detector de colisiones, el sistema de coordenadas, y muchas otras. En este proyecto se usa la vertiente de desarrollo en tres dimensiones.

3. Entorno de trabajo

Como la mayor parte de motores gráficos que generan contenido en tres dimensiones, la geometría con la que trabaja Unity3D está compuesta por mallas poligonales, es decir trata todos los objetos que tiene que renderizar (proceso de crear gráficos por ordenador) como si estuvieran compuestos por polígonos, en la mayoría de los casos triángulos.

Unity3D tiene una estructura interna que es propietaria, esto quiere decir que no es pública, pero para describir la estructura que puede tener se puede tomar como referencia la Figura 3.2, en esta figura aparece en el trabajo de Gregory et. al. [22]. Esta estructura específicamente es de un motor de videojuegos, pero el uso principal de los motores gráficos y de Unity3D en concreto es como motor de videojuegos. Como se representa en la figura el motor está dividido por secciones, y cada una se encarga de una tarea. Hay una sección dedicada a los componentes específicos de cada aplicación que se desarrolla con ese motor, otra se encarga de dar efectos a las imágenes, otra de gestionar la creación de gráficos a bajo nivel (como texturas, materiales y shaders), también hay secciones como las que implementan las funciones online, la parte de sonido del motor, las animaciones, las físicas y colisiones entre los objetos. El motor gráfico también accede a funciones específicas y librerías del propio sistema en que se ejecuta. Además la mayor parte de los motores no tienen implementados por los desarrolladores todos los componentes, sino que usan componentes de otros desarrolladores, como por ejemplo las librerías que generan gráficos a bajo nivel (OpenGL o DirectX), las físicas, el sistema de ficheros, los protocolos de conexión a Internet y otros componentes.

Unity3D divide su contenido en escenas, para representar internamente los objetos que componen las escenas Unity3D se utiliza una representación basada en la programación orientada a objetos. Todos los objetos que componen una escena son del tipo *GameObject*. Todos los *GameObject* están formados por una lista de componentes, objetos de la clase *Component*. Estos componentes pueden ser cosas como detectores de colisiones, cuerpo físico para que actúen sobre él las físicas del motor, entre otros. Entre estos componentes están los renderizables (*Renderer*) que determinan qué se renderiza de un objeto, estos renderizables se componen por la malla poligonal del objeto y el material del que está compuesto. Entre los componentes también destacan las transformaciones, estas son del tipo *Transform* y determinan la posición, la rotación y la escala que tiene el objeto en la escena. Otro tipo de componentes de un *GameObject* son los que podrían calificarse como comportamientos, estos se llaman *Behaviour*, los cuales se pueden activar y desactivar. De este tipo de objetos de la clase *Behaviour* destacan tres. El primero de ellos es la clase *MonoBehaviour* del cual heredan los scripts que modelan el comportamiento de los *GameObject*, estos tienen varios métodos, el método “Start()” que se ejecuta cuando se inicializa el script y el método “Update()” que se ejecuta cada vez que se renderiza un fotograma en Unity3D mientras este el script activo. Estos scripts se pueden programar en varios lenguajes de programación. En este proyecto se ha escogido C# ya que es más fácil de manejar si se tienen los conocimientos adecuados de programación orientada a objetos. Otro tipo de componentes que heredan de *Behaviour* son los componentes de tipo *Camera* que determina que parte de la escena se renderiza. Y por último las luces u objetos de la clase *Light*, que determinan la iluminación de la escena. Un diagrama de clases de los objetos y componentes más usados de Unity en este proyecto se puede ver en la Figura 3.3.

Una de las características de los *GameObject* es que pueden anidarse. Estos objetos que están contenidos en otros tienen la transformación en función del objeto que los contiene, siendo una transformación local.

3. Entorno de trabajo

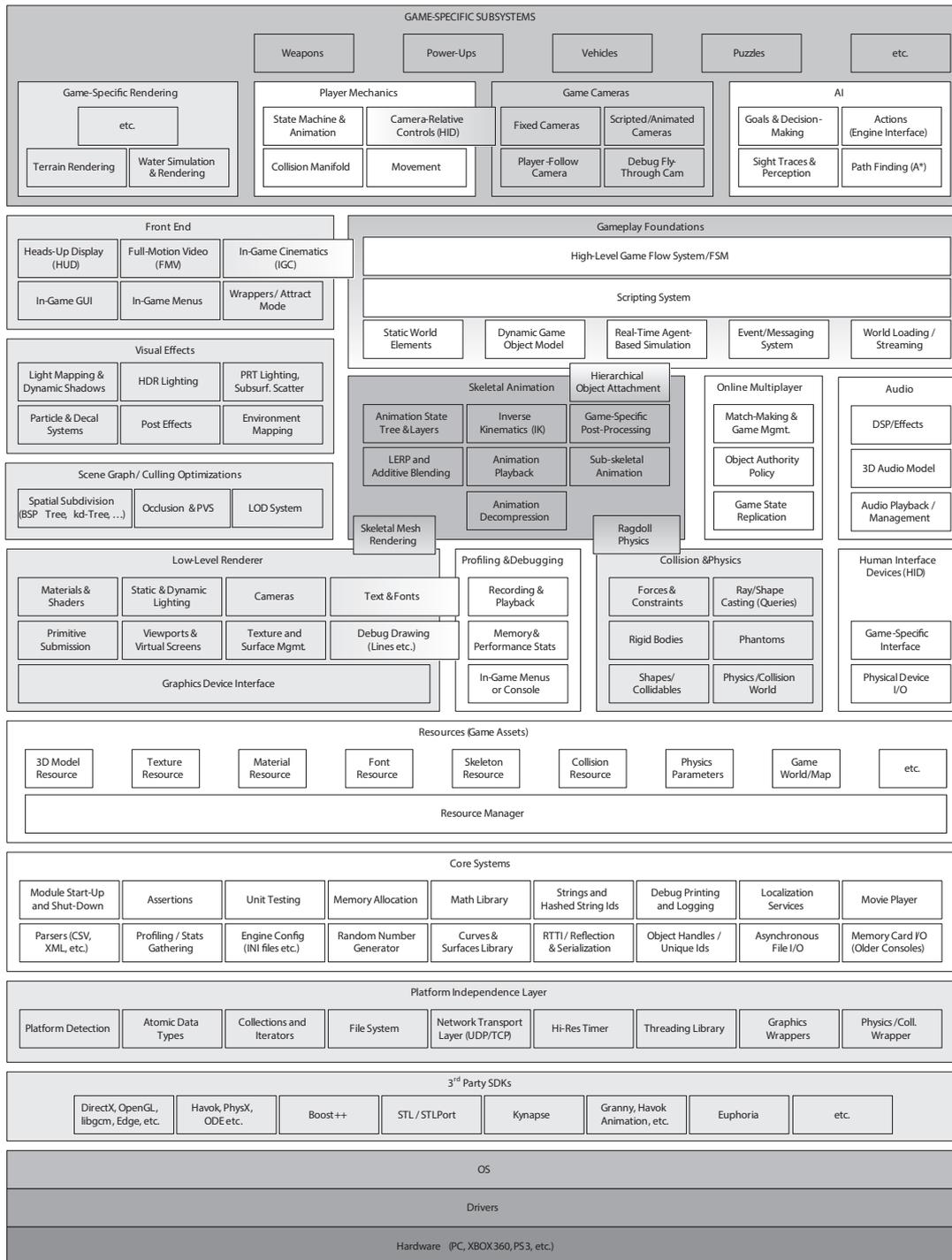


Figura 3.2: Esta figura muestra la estructura genérica que puede tener un motor gráfico, en este caso uno dedicado específicamente a videojuegos. En la figura se puede ver las distintas secciones en las que se agrupan los componentes más característicos de un motor gráfico. Fuente [22].

El funcionamiento de Unity3D cuando renderiza una escena se puede dividir en dos partes, la

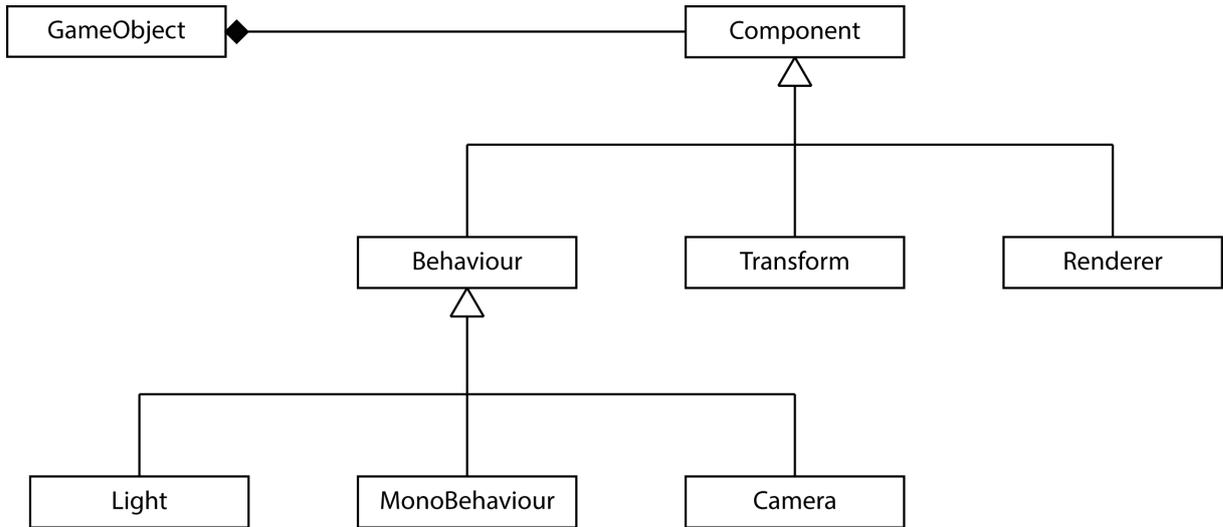


Figura 3.3: Esta figura muestra un diagrama de clases simplificado de los objetos y los componentes de Unity3D que más se usan en este proyecto.

parte que renderiza los objetos de la escena y la parte que hace que el comportamiento asociado a los objetos de la escena se ejecute. Para renderizar la escena Unity3D necesita una cámara, desde esa cámara se renderizan los objetos que estén en su campo de visión y tengan asociado un renderizable, teniendo en cuenta los objetos que tienen como componente luces para dar iluminación a la escena. Por otro lado el comportamiento de Unity3D se modela con los objetos que tienen asociados scripts, cada vez que se renderiza un fotograma Unity3D ejecuta el método “Update()” de todos los scripts que están activos en todos los objeto de la escena.

3.2.3. Creación del entorno

Para modelar un entorno en Unity3D se usan los objetos (*GameObject*) mencionados anteriormente. Estos objetos se colocan en la escena y se les añade los componentes (*Component*) que se deseen. Con estos componentes se les puede dar varios tipos de propiedades a los objetos tanto a la hora de renderizar como a la hora de que los objetos tengan un comportamiento mientras el motor está generando los gráficos.

La iluminación se aplica a las mallas poligonales que componen los objetos de la escena, teniendo en cuenta las propiedades que tienen los objetos que se quieren renderizar. En tiempo real se usan principalmente dos tipos distintos de cálculo de la iluminación, la que se calcula en tiempo real que se centra en iluminar los objetos dinámicos que se mueven en tiempo real, e iluminación precalculada. Esta iluminación es más precisa pero más cara computacionalmente que la iluminación calculada en tiempo real, y está centrada en mostrar objetos estáticos de manera realista.

A los objetos que se renderizan en Unity3D se les puede dar propiedades visuales como iluminación difusa, destellos especulares, propiedades metálicas o texturas entre otros. Las propiedades de los objetos en Unity3D se tratan mediante shaders o sombreadores, que son fragmentos sencillos de código que trabajan sobre los vértices de los polígonos que componen las mallas po-

ligonales y los píxeles que se ven en pantalla de dicho objeto. Entre estos shaders se encuentra el *standard shader* que es un shader complejo que permite aplicar multitud de efectos a un objeto.

En Unity3D, como en todo motor de gráficos, existe un elemento que se denota como “cámara”. Esta cámara es lo que determina la vista de lo que el motor gráfico va a renderizar. En Unity3D la cámara tiene dos modos de proyección. El primer modo es la proyección perspectiva: en este modo se puede configurar el FOV y las proporciones que se ven en la pantalla. El FOV que se configura en las cámaras es el vertical, el horizontal se ajusta automáticamente para encajar con la resolución en la que se está renderizando. El segundo tipo de modo de proyección es el modo de proyección ortográfica, en este tipo de proyección se puede cambiar el tamaño de la proyección y las proporciones de esta. La principal diferencia entre ambos modos está en que mientras en proyección perspectiva un objeto con la misma forma y tamaño se ve de diferente forma y tamaño dependiendo de su posición, en proyección ortográfica el objeto no varía de forma en la que se ve, ignorando profundidad y desplazamiento. Además, en ambos modos de proyección se pueden configurar parámetros como la distancia más cercana y más lejana que se renderizan. Un punto importante que permiten las cámaras de Unity3D y que es clave en este proyecto es que se les puede asociar una textura dinámica (*render texture* es su nombre en Unity3D), en la cual se captura la vista de la cámara, y luego asociarla a un objeto como un plano, y que en éste se vea lo que está capturando la cámara en ese momento. En este proyecto este tipo de texturas son necesarias para solventar problemas que se introducen a la hora de integrar la aplicación de realidad aumentada. La documentación de estas texturas se puede encontrar en la documentación de Unity3D, en concreto en [23].

3.2.4. Controlador

En este proyecto se diseñan varios experimentos con los estímulos que aparecen en la Sección 2.3.3. En particular, se debe modelar el comportamiento del experimento (esferas avanzando en direcciones opuestas y cruzándose), así como un sistema de guiado de los participantes mediante texto durante el experimento. Finalmente, se deben grabar y almacenar las respuestas del usuario. Para todo esto, se ha creado un script de Unity3D. Este script se encarga también de tareas como generar el orden en que aparecen las condiciones de los estímulos y almacenar en un fichero las respuestas de los usuarios. El código del script tiene dos partes importantes. Una es la que se ejecuta al inicializarse, el método “Start()”. La otra es la que se ejecuta cada vez que se genera un fotograma en Unity3D, el método “Update()”. Los componentes sobre los que actúa este controlador caracterizados en un diagrama de clases se pueden ver en la Figura 3.4.

El método de inicialización del script, “Start()”, contiene el código necesario para crear el archivo donde se registran los datos del experimento y las respuestas de cada participante. También genera el orden en que se mostrarán las condiciones del experimento, e inicializa todos los valores que son necesarios para empezar con el experimento.

El método “Update()” es el que modela el comportamiento de los estímulos. Al comenzar el experimento este método se encarga de activar los carteles de explicación que aparecen y registrar las pulsaciones del ratón que permiten continuar. Para modelar el comportamiento de las condiciones de los estímulos se recurre a una variable que controla el tiempo que deben moverse las esferas. En dos de las condiciones del estímulo visual las esferas deben pausarse durante un determinado número de fotogramas. Para el cálculo del momento en el cual las

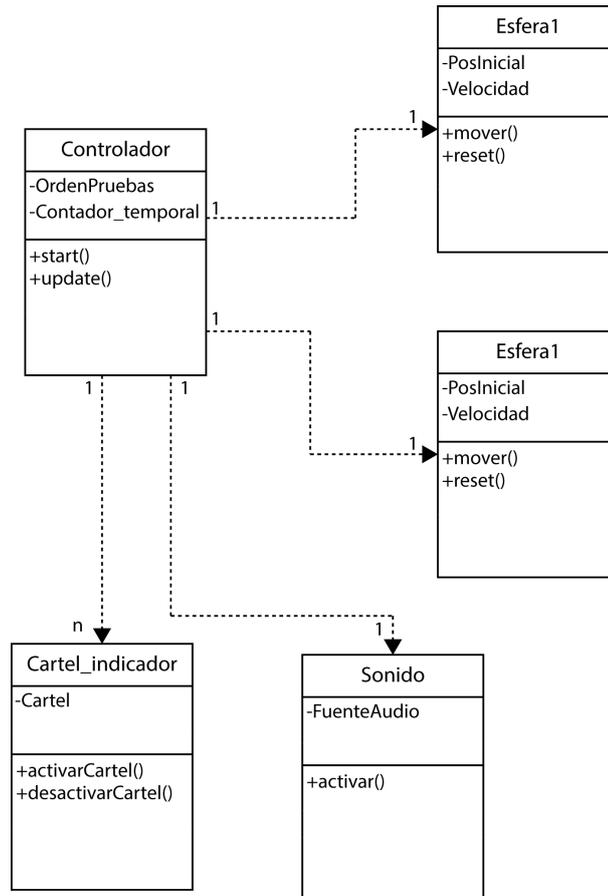


Figura 3.4: Representación como diagrama de clases del controlador, los estímulos y los carteles indicadores. Las flechas indican la asociación del controlador con los objetos que controla. Los atributos son los parámetros que tienen los objetos. Los métodos del controlador son el método “Start()”, que es el que inicializa todos los parámetros necesarios, y el método “Update()”, que es el que modela el comportamiento del resto de objetos. El controlador es el que accede a los métodos del resto de objetos para activar o desactivar cada uno de los carteles, mover las esferas o colocarlas en su posición inicial y hacer que el sonido se active.

esferas deben detenerse se tiene en cuenta la posición de las esferas y el momento en el que coinciden o están cerca de hacerlo, entonces se pausa el movimiento durante los fotogramas que correspondan. El caso de las condiciones del estímulo auditivo es distinto ya que existen las condiciones de aparición de 150 milisegundos antes del punto de coincidencia o 150 milisegundos después. Para esto se ha usado la posición local de las esferas y se ha calculado con la Ecuación 3.1 los tiempos donde debe aparecer el sonido. Esta ecuación es la de movimiento rectilíneo uniforme.

$$x = x_0 + vt \tag{3.1}$$

donde x_0 es el punto inicial, x es el punto actual, v es la velocidad a la que se mueve la esfera, y t es el tiempo que ha pasado. Para calcular cual es la posición de una esfera 150 ms antes de coincidir con la otra y 150 ms después se calcula cuánto tiempo tarda en llegar a la mitad del espacio por el que están separadas las esferas, luego se resta o suma 150 ms a ese tiempo, y se logra el punto en el cual las esferas están 150 ms antes y después del punto de coincidencia.

Cuando termina el movimiento se activa un cartel preguntando si se han visto las esferas cruzarse o rebotar, se registra la respuesta del usuario a través del ratón y se guardan las condiciones de pausa y de sonido, además del tiempo que ha tardado en responder desde la aparición del cartel y la respuesta en el fichero de registro. Una vez el participante responde se pasa a la siguiente ejecución escogiendo las condiciones de los estímulos siguientes generadas aleatoriamente en el método “Start()”. Cuando se acaban de mostrar todas las condiciones de los estímulos el total de veces que se determine ver cada una, se muestra un cartel indicando la finalización.

Este controlador aparece a lo largo de los tres experimentos que se describen en este proyecto, y en todos el comportamiento es el que se ha descrito aquí. En la Figura 3.4 se puede ver una representación de las relaciones entre los estímulos y los carteles con el controlador caracterizado como un diagrama de clases. Está programado como un script de Unity3D en C#.

3.3. Realidad virtual

En este proyecto se trabaja en un entorno de realidad virtual. Para generar dicho entorno se usa Unity3D el cual integra el soporte para dispositivos de realidad virtual. En este entorno se modelan dos experimentos con los estímulos descritos en la Sección 2.3.3. Para modelar el comportamiento de los estímulos en estos experimentos se usa el controlador descrito en la Sección 3.2.4.

3.3.1. Integración de la realidad virtual en Unity3D

A partir de la versión 5.1 de Unity3D la realidad virtual está integrada, siendo necesario activarla en la configuración del proyecto. Unity3D soporta varios tipos de dispositivos de realidad virtual: los de toda la familia de Oculus, tanto las versiones de desarrollo como la versión comercial, el Samsung Gear VR, los dispositivos que funcionan mediante OpenVr como por ejemplo el HTC Vive, y también soporta el dispositivo Playstation VR.

El principal cambio de activar la función de realidad virtual en Unity3D consiste en que en lugar de una cámara se utilizan dos para poder renderizar diferentes vistas para el ojo derecho y el ojo izquierdo. Se puede escoger el modo de renderizado para el ojo izquierdo, el ojo derecho, la pantalla del ordenador o ambos ojos. En este último caso activa el render estéreo con dos cámaras, manteniendo ambas cámaras a la distancia media interpupilar en Unity3D. También se adapta el FOV de las cámaras al mismo que tiene el dispositivo de realidad virtual.

Al activar la función de realidad virtual Unity3D también activa para todas las cámaras el seguimiento posicional que tiene el Oculus, variando la rotación de las cámaras dependiendo de la información que procuren los sensores y la cámara posicional del Oculus, Unity3D no permite desactivar esta función sin desactivar el resto de las funciones de realidad virtual. Unity3D también se encarga de dar la distorsión de barril necesaria para el Oculus.

En resumen, Unity3D se encarga de usar las funciones y valores que proporciona el OculusSDK y automatiza muchos procesos como el rotar las cámaras e introducir la distorsión necesaria antes de enviar las imágenes al Oculus.

3.3.2. Experimento 1

Este experimento está dividido en dos bloques, uno que se ejecuta en la pantalla de ordenador y otro que se ejecuta en el Oculus, en un entorno de realidad virtual. Uno de los puntos clave de la implementación de este experimento es pasar de la pantalla de ordenador al Oculus de manera sencilla. Para ello se tienen cuatro escenas de Unity3D diferentes: una de inicio, otra un entrenamiento (que se ve siempre justo antes del bloque de realidad virtual), otra el propio bloque de realidad virtual, y una más que es el bloque que se ve en la pantalla. En la primera escena se genera el orden en el que aparecerán los bloques del experimento (realidad virtual y pantalla), se muestran los carteles indicadores de si el primer bloque se reproduce en el Oculus o en la pantalla y se genera el orden de aparición de condiciones en cada bloque. Pasar de escena en Unity3D es simple mediante una instrucción que se introduce en un script que permite cargar otras escenas del proyecto.

En la parte del experimento que se reproduce en la pantalla se desactiva el soporte de realidad virtual en Unity3D mediante una instrucción que se añade en el método “Start()” del controlador (*VRSettings.enabled=false*). Antes del bloque del experimento que se reproduce en el Oculus, al comienzo del entrenamiento previo, se activa la opción para que funcione correctamente la realidad virtual (*VRSettings.enabled=true*), de esta manera se puede llevar a cabo todo el experimento solamente guiando al participante, sin tener que hacer ninguna configuración intermedia.

El comportamiento de los estímulos, los carteles que explican el experimento, y el registro de las respuestas se hace a través del controlador descrito en la Sección 3.2.4. En este caso el entorno que rodea al experimento en los dos bloques es bastante sencillo, un plano de color blanco debajo de los estímulos visuales y varias luces puntuales encima de los estímulos iluminando la escena.

3.3.3. Experimento 2

Este experimento tiene tres bloques y los tres se reproducen en un entorno de realidad virtual. Tiene cuatro escenas: un entrenamiento, y los tres bloques del experimento. En el entrenamiento se determina el orden, que es aleatorio, de las fases del experimento.

El comportamiento de los estímulos se ha modelado con el controlador que se ha descrito en la Sección 3.2.4 teniendo en cuenta factores como el aumento de las proporciones del estímulo visual.

3.4. Realidad aumentada

En este proyecto, aparte de trabajar en un entorno de realidad virtual, se trabaja en un entorno de realidad aumentada. La realidad aumentada consiste en superponer objetos virtuales en un entorno real. Para ello en este proyecto se usan dos cámaras para capturar la realidad y fusionarla con objetos virtuales. Las cámaras se montan en un soporte, el que aparece en la Figura 3.6, en el que van separadas a la distancia que corresponde a la media de la distancia interpupilar de una persona adulta, este soporte se coloca en la parte frontal del Oculus. Para

crear el entorno de realidad aumentada se usa una aplicación llamada ARToolkit que se integra en Unity3D. Esta aplicación se modifica para funcionar con el Oculus y dos cámaras para poder modelar un experimento con los estímulos que aparecen en la Sección 2.3.3 en un entorno de realidad aumentada.

3.4.1. Características de las cámaras

Las cámaras que se necesitan para este proyecto deben tener capacidad para capturar a lo largo de un gran FOV, cercano al que puede reproducir el Oculus, para que las imágenes encajen correctamente y mantengan las proporciones reales al reproducirlas en el Oculus. Para poder encajar las cámaras en un soporte, que luego se colocara en el Oculus, estas deben ser de pequeño tamaño. Las cámaras deben transmitir en tiempo real a una tasa de fps aceptable para que la realidad aumentada se vea correctamente en el Oculus. Por eso se escogió usar unas cámaras USB ELP ya que son pequeñas, fáciles de montar en un soporte y no requieren ninguna instalación extra en el equipo para funcionar.

Estas cámaras tiene un tamaño de sensor de 1/3 pulgadas, con dos megapíxeles de resolución. Además tienen distintos formatos de imagen como 16:9, 5:4 o 4:3, y también distintas resoluciones como 1080p (1920 píxeles por 1080 píxeles, con un formato de 16:9, comúnmente conocido como Full HD), 720p (1280 píxeles por 720 píxeles, con un formato de 16:9), o 1280 píxeles por 1024 píxeles con formato 4:3, entre otras resoluciones.

Estas cámaras son capaces de capturar diferentes tasas de fps dependiendo de la resolución de las imágenes obtenidas. Por ejemplo, son capaces de capturar 30 fps en las resoluciones con más cantidad de píxeles como FullHD o 1280 por 1024 píxeles, y a 60 fps con las imágenes de menor resolución como 1280 por 720 píxeles. Las cámaras tienen unas lentes con una distancia focal de 2.1 mm. La distancia focal de una lente es la separación de esta y el punto focal, que es donde coinciden todos los rayos paralelos que atraviesan la lente. Con el tamaño del sensor y la distancia focal de las lentes se puede calcular el FOV máximo que ofrece la cámara, el FOV horizontal máximo de la cámara es de 97.63 grados y el vertical de 81.20 grados. Este FOV es algo menor al que se reproduce en el Oculus (106 grados vertical y 95 horizontal). Esto se calcula con la Ecuación 3.2:

$$FOV = 2 * \arctan\left(\frac{d}{2 * f}\right) \quad (3.2)$$

siendo d el tamaño del sensor en milímetros, en la dirección en que se quiere medir el FOV (el ancho del sensor si se quiere el FOV horizontal y el alto si se quiere el vertical), y f la distancia focal de la lente en milímetros.

3.4.2. Montaje estéreo de las cámaras

Para poder producir un efecto tridimensional, las cámaras deben estar colocadas a una distancia específica una de la otra. En este proyecto las cámaras están montadas en un montaje en paralelo (comúnmente denominado *parallel*). Este tipo de montaje consiste en colocar las cámaras a una cierta distancia una de la otra y orientarlas de forma paralela, como en la Figura 3.5. De esta manera se consigue lograr un efecto tridimensional al tener una referencia

de un objeto para ambas cámaras. En este tipo de montaje el llamado plano de convergencia de las cámaras (el punto donde la vista de los objetos coincide para ambas cámaras) está en el infinito. El hecho de tener todos los objetos por delante del plano de convergencia quiere decir que a la hora de representarlos en el Oculus todos se verán como si estuvieran por delante de la pantalla.

Existen otros tipos de montaje de cámaras que colocan el plano de convergencia en un punto del espacio, no el infinito. El poder fijar el plano de convergencia en algún punto que no sea el infinito implica que hay dos nuevas posiciones con respecto a tenerlo en el infinito, en el plano de convergencia y por detrás de él. Los objetos que estén en el plano de convergencia se verán como si estuvieran en la pantalla del Oculus, a la misma distancia que ésta, y los objetos que están por detrás del plano de convergencia se verán como si estuvieran más alejados de la pantalla. El primero de ellos es el llamado *toed-in*, que consiste en montar las cámaras de manera que estén orientadas con una inclinación hacia el lado donde está la otra cámara, esta inclinación debe ser la misma en ambas y determina, junto con la separación de las cámaras, el lugar del plano de convergencia. En la Figura 3.5 se puede ver la representación gráfica del montaje *toed-in*. Este método tiene problemas derivados de la orientación de las cámaras, ya que el plano de convergencia de cada una de las cámaras es diferente, como se muestra en la Figura 3.5. Esto se puede corregir pero requiere hacer transformaciones a las imágenes que las cámaras capturan, generando otros problemas como que las imágenes no tengan forma rectangular, sino trapezoidal, esto se denomina *keystone distortion* y para solucionarlo requiere un recorte de la imagen que limita el FOV vertical.

Otra forma que combina las anteriores es la llamada *off-axis*. En este tipo de montaje las cámaras están en paralelo como en *parallel* pero se introduce un recorte en el campo de visión horizontal en el lado izquierdo para la cámara izquierda, y en el lado derecho para la cámara derecha. Para lograr este recorte las cámaras deben tener una lente asimétrica o se debe recortar parte de la imagen, perdiendo FOV. El ángulo de FOV que se recorta fija el lugar donde se encuentra el plano de convergencia. Este método también logra tener un plano de convergencia en algún punto que no está en el infinito, con sus consiguientes ventajas. En la Figura 3.5 se puede ver la representación gráfica del montaje *off-axis*.

Se ha escogido un montaje *parallel* ya que es bastante sencillo de implementar en el soporte, no es necesario recortar FOV de las cámaras, y a la hora de montar el sistema de realidad aumentada es útil para corregir diversos fallos que aparecen. Tampoco requiere fijar en ningún punto el plano de convergencia, lo cual podría provocar diferentes cambios en el experimento al variar la posición.

Para fijar las cámaras al Oculus de manera precisa se ha creado un soporte que se ajusta a las cámaras, las mantiene a la media de la distancia interpupilar humana (63.5 mm) y mantiene su orientación paralela. Para que las cámaras se adapten mejor al FOV que tiene el Oculus las cámaras se han rotado 90 grados. Esto se hace para proporcionar un FOV vertical mayor que el horizontal, como ocurre en el Oculus. Este soporte se fija al Oculus y se mueve conforme se mueve la cabeza del usuario que lleva el Oculus, capturando lo que veía el usuario si no llevara el Oculus. El soporte se muestra en la Figura 3.6.

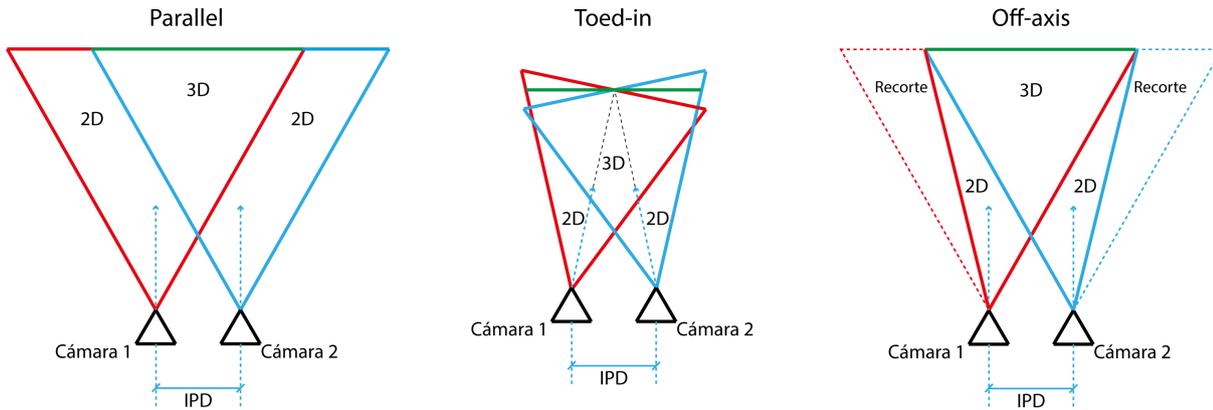


Figura 3.5: Izquierda: En esta figura se ve el montaje en modo *parallel*, las secciones marcadas con la palabra 2D son secciones donde solo hay una vista de esa parte, mientras que en la sección marcada con 3D convergen las imágenes de las dos cámaras formando un efecto tridimensional. El plano de convergencia (verde) está donde coinciden totalmente los campos de visión de las cámaras, en este caso el infinito. Las cámaras están separadas a la media de la distancia interpupilar humana o IPD como aparece en la figura. Centro: En esta figura se puede ver una representación gráfica del montaje *toed-in*. El plano donde coinciden los vectores de orientación es el plano de convergencia (verde). La rotación de los planos de ambas cámaras (rojo y azul) producen una vista errónea. Derecha: En esta figura se puede ver una representación gráfica del montaje *off-axis*. Se puede ver que el plano de convergencia (verde) está donde coinciden los campos de visión de la cámara, la parte indicada por la línea de puntos es la que se recorta para lograr la coincidencia de los campos de visión.

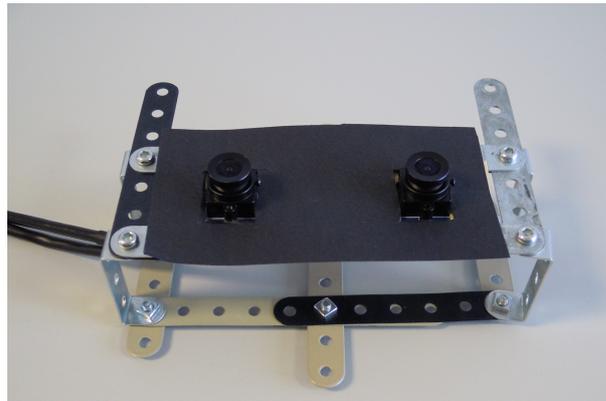


Figura 3.6: En esta figura se puede ver una imagen del soporte que se usa para sustentar las cámaras orientadas en modo *parallel*. El soporte está fabricado con varillas de metal y se fija a la parte frontal del Oculus.

3.4.3. ARToolkit

Uno de los puntos clave del proyecto es mostrar un experimento de percepción *crossmodal* en varios entornos inmersivos, uno de ellos es un entorno de realidad aumentada. Para ello se van a usar las cámaras y el soporte explicados en la Secciones 3.4.1 y 3.4.2. La aplicación elegida para desarrollar esta parte es ARToolkit, una aplicación de código abierto que es integrable con Unity3D a través de varios objetos y scripts especiales que proporciona la aplicación.

ARToolkit es una aplicación de realidad aumentada basada en la búsqueda de marcadores, como el de la Figura 3.7, que es el que se usa en este proyecto. Esta aplicación usa una librería de visión por computador llamada openCV, que es la que trata las imágenes y da la posición del

marcador con respecto a la imagen donde aparece. Esta aplicación permite coger las imágenes directamente desde una cámara USB.



Figura 3.7: En esta figura se puede ver una imagen de uno de los marcadores que usa la aplicación ARToolkit para colocar encima objetos virtuales.

La aplicación ARToolkit permite hacer seguimiento de marcadores de una manera sencilla, en Unity3D funciona mediante scripts que configuran todos los objetos necesarios para crear la realidad aumentada. Una vez que todos los objetos necesarios están inicializados la aplicación puede comenzar a ejecutarse. Cuando se detecta un marcador la aplicación hace que una cámara virtual varíe su transformación con respecto a la escena virtual dependiendo de la posición del marcador en la imagen. Para generar la imagen realidad aumentada, la aplicación combina la vista de la cámara virtual que renderiza la escena virtual que se coloca encima del marcador de realidad aumentada y la imagen del mundo real que captura la cámara. Un ejemplo del funcionamiento de esta aplicación se puede ver en la Figura 3.8.

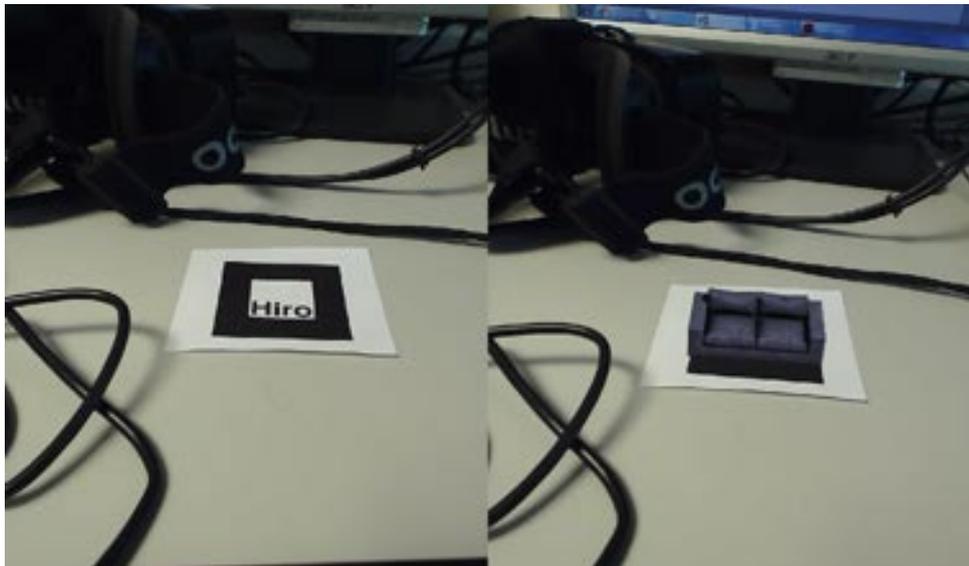


Figura 3.8: A la izquierda de esta figura se puede ver una imagen de lo que graba la cámara de video cuando no está activa la aplicación de realidad aumentada. A la derecha se puede ver lo que se ve en Unity3D cuando está activa la aplicación de realidad aumentada y se coloca un objeto virtual (un sofá) encima del marcador.

3.4.4. Integración de ARToolkit en Unity3D

En este proyecto se quiere representar un entorno de realidad aumentada en el Oculus, para ello se usa la aplicación ARToolkit, que se integra cómodamente en Unity3D. Al integrar esta

3. Entorno de trabajo

aplicación con el Oculus y las cámaras aparecen dos problemas graves. El primero derivado de la activación de la rotación automática de las cámaras virtuales en Unity3D cuando el Oculus está conectado, y el segundo debido a un problema que hace que al activar el uso de dos cámaras para seguir los marcadores no funcione correctamente la aplicación. En este proyecto se ha implementado una solución a ambos problemas.

El primer problema que aparece es causado por la rotación de las cámaras virtuales que renderizan contenidos para el Oculus, variando la rotación de cualquier cámara virtual cuando varía la rotación del Oculus. Esto es un problema, ya que la aplicación ARToolkit muestra los contenidos capturados por la cámara en una textura que se reproduce en malla poligonal frente a una cámara virtual, y al rotar el Oculus varía la posición de la cámara virtual y de la cámara física (al estar colocadas las cámaras en un soporte sobre la parte frontal del Oculus) pero no de la malla poligonal. Para el correcto funcionamiento de la aplicación ARToolkit en el Oculus la cámara virtual no debe variar ni su posición ni rotación con respecto a la malla poligonal donde se reproduce el video. Para solventar este problema se va a usar una *render texture* que se asocia a la cámara virtual que renderiza la malla poligonal donde se reproduce lo que captura la cámara física (al usar ésta *render texture* automáticamente Unity3D no aplica la rotación que se introduce al renderizar contenidos para el Oculus), y ésta *render texture* se coloca en plano que tiene una transformación local con respecto a una cámara virtual que es la que genera el contenido para el Oculus, de esta manera cuando rota la cámara virtual también rota el plano con la *render texture* y el contenido de realidad aumentada siempre se ve en la misma posición en la vista renderizada por esta cámara virtual. La cámara virtual se configura para que renderize la parte que corresponde a un solo ojo del Oculus, ya que el otro problema que aparece al integrar la aplicación ARToolkit con dos cámaras físicas implica que se debe configurar solo una de ellas con la aplicación.

El otro problema viene dado por un error de la aplicación que provoca que al usar dos cámaras el contenido de éstas no se reproduzca correctamente y solo se muestre una imagen totalmente negra para cada cámara. Un punto a tener en cuenta a parte de este error es que el coste computacional de trabajar con dos cámaras es bastante alto, ya con una sola siguiendo un marcador el ordenador no es capaz de renderizar a los 75 fps necesarios para el Oculus con la cámara en una resolución FullHD, solo siendo capaz de llegar a apenas 35 fps. La solución encontrada para este problema pasa por usar solo una cámara a través de la aplicación ARToolkit y usar la otra a través de Unity3D como *WebCamTexture*, una textura que reproduce el video que captura una webcam. Se crea un plano al que se le coloca la textura y se asocia a una cámara virtual para que tengan la misma rotación. Para el objeto virtual visto desde la cámara que no sigue el marcador se crea una cámara virtual que se asocia a la que genera la vista de la escena virtual, y se calibra manualmente para que los objetos virtuales aparezcan justo encima del marcador. Para lograr esto de una manera correcta se aprovecha el montaje *parallel* de las cámaras, colocando la cámara virtual asociada a la que genera la vista de la escena de modo que tengan la misma orientación, en la Figura 3.9 se puede ver una representación gráfica de estas figuras. A ésta cámara virtual se le asocia una *render texture* y ésta se coloca justo por delante del plano donde se reproduce el contenido de la cámara que no sigue al marcador. En la Figura 3.10 se puede ver como apenas hay una diferencia significativa entre la cámara que se calibra manualmente y la que depende de ARToolkit a la hora de mostrar una escena virtual.

Con estas dos soluciones se tienen dos cámaras virtuales, una para cada ojo, teniendo en cuenta que tiene que corresponder la cámara que está a la derecha del soporte con lo que se

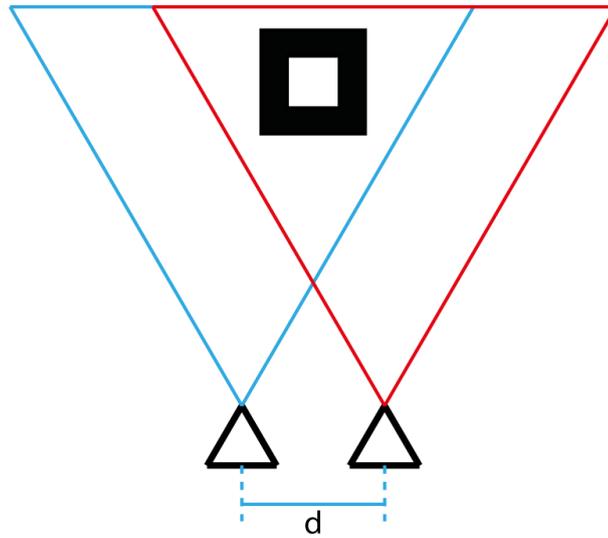


Figura 3.9: En esta figura aparece una representación gráfica de las cámaras para seguir el marcador, siendo una solo la que sigue el marcador y la otra se mantiene siempre a la misma distancia d , esta distancia d es la que se calibra manualmente para que la vista del objeto virtual coincida en la posición del marcador. Al tener dos cámaras se genera una vista 3D del objeto virtual.

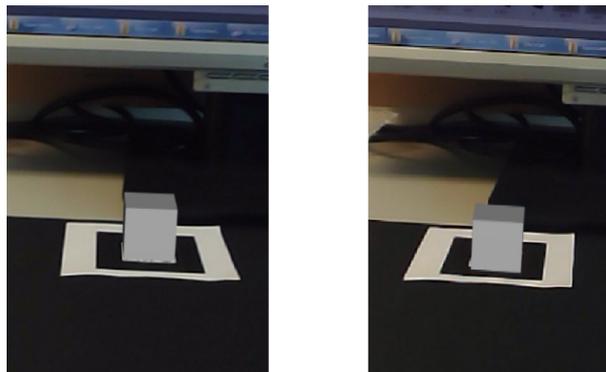


Figura 3.10: En la imagen de la izquierda se puede ver lo que muestra la cámara que es controlada totalmente por ARToolkit y a la derecha lo que muestra la cámara que es calibrada manualmente para seguir el marcador.

envía al ojo derecho del Oculus y lo mismo con la cámara izquierda.

Después de todos estos cambios y añadidos que se le ha introducido a la aplicación ARToolkit ha aumentado el número de objetos de Unity3D que están implicados en crear un sistema de realidad aumentada en el Oculus. En la Figura 3.11 se puede ver un esquema de los objetos de Unity3D que forman parte de la aplicación de realidad aumentada. Se pueden distinguir dos bloques, el primero muestra los elementos originales que vienen dados por la aplicación ARToolkit y que se modifican para funcionar en el Oculus, y el segundo muestra los elementos añadidos en este proyecto para solucionar los problemas que se han detallado anteriormente. Además el código del script del controlador asociado a Controlador_RA ha sido modificado para el correcto funcionamiento de la aplicación. Textura_video es la imagen del mundo real que captura la cámara, Camara_RA es la cámara virtual que genera la vista de la escena virtual (Escena_Virtual), y Escena_RA es la combinación de lo que captura la cámara real y lo que

3. Entorno de trabajo

captura la cámara virtual, es decir la escena de realidad aumentada. Para solucionar el problema de la rotación que introduce el Oculus se captura Escena_RA con una *render texture* y se muestra en RA_Ojo_Derecho, que es la escena de realidad aumentada para el ojo derecho del Oculus. El objeto Sub_Camara_RA es la cámara virtual que se asocia a Camara_RA para generar la vista aproximada de la escena virtual para la cámara que no es capaz de seguir el marcador. Textura_video.2 es donde se reproduce lo que captura dicha cámara y RA_Ojo_Derecho es donde se combinan Textura_video.2 y lo que captura Sub_Camara_RA para formar la escena de realidad aumentada para el ojo izquierdo del Oculus.

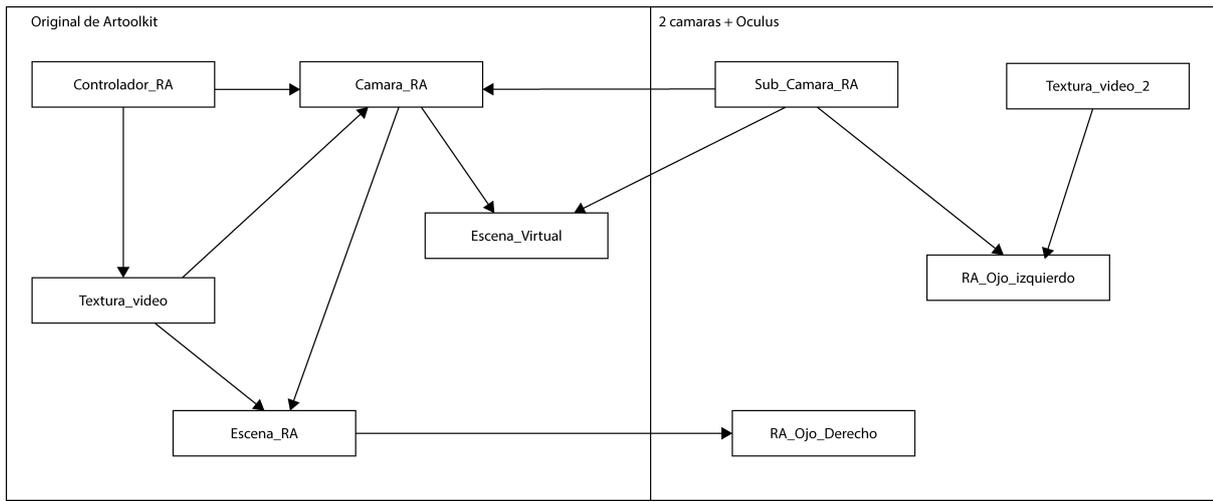


Figura 3.11: En esta figura se puede ver un esquema de los componentes que forman parte de la aplicación de realidad aumentada dividido en dos bloques, el primero de ellos, a la izquierda, corresponde con los elementos originales de ARToolkit y el segundo, a la derecha, corresponde a las partes añadidas para solucionar los problemas que se han detallado anteriormente. Para más información sobre los distintos elementos, el lector puede referirse al texto.

3.4.5. Calibración de las cámaras

Las cámaras que se usan en este proyecto introducen una distorsión a la imagen, en este proyecto se corrige dicha distorsión para lograr una representación más realista del sistema de realidad aumentada.

Las lentes con una distancia focal baja producen un efecto que se llama distorsión radial, que deforma la imagen curvando la posición que deberían tener los píxeles y mostrando las líneas rectas como curvas, aumentando dicho efecto conforme la posición del píxel se aleja del centro de la imagen. En las cámaras también aparece otro tipo de distorsión, la distorsión tangencial, que aparece debido a deformidades en la construcción de las cámaras debido a un desalineamiento entre el centro de la lente y el sensor, esto ocurre debido al uso de cámaras de bajo coste y en las cámaras modernas no es muy significativa, pero en este proyecto el coste de corregirla es mínimo y también se implementa la corrección a esta distorsión. Esta distorsión se puede ver en la Figura 3.12 y se puede caracterizar mediante los parámetros que aparecen en la Ecuación 3.3.

$$\begin{aligned}
 x_d &= x_u(1 + K_1r^2 + K_2r^4) + (P_2(r^2 + 2x_u^2) + 2P_1x_uy_u) \\
 y_d &= y_u(1 + K_1r^2 + K_2r^4) + (P_1(r^2 + 2y_u^2) + 2P_2x_uy_u) \\
 r &= \sqrt{(x_u - x_c)^2 + (y_u - y_c)^2}
 \end{aligned}
 \tag{3.3}$$

los parámetros x_u e y_u corresponden a la posición del píxel en la imagen sin distorsionar todavía; x_c e y_c corresponden al píxel central de la imagen; y x_d e y_d a la posición del píxel cuando se le aplica la distorsión. Los parámetros K_1 y K_2 caracterizan la distorsión radial, y los parámetros P_1 y P_2 caracterizan la distorsión tangencial de las imágenes. La forma de caracterizar la distorsión de las lentes que aparece en esta ecuación se conoce como el modelo de *Brown-Conrady* [24]. Este modelo de distorsión se usa porque es aplicable en un shader ya que no tiene operaciones complejas y las fórmulas que se usan son independientes por cada píxel donde se calcula la distorsión. En este proyecto para corregir la distorsión se aplica una contraria a la que generan las lentes de la cámara. Los parámetros que se necesitan son obtenidos mediante una calibración que consiste en tomar fotos de un tablero (checkerboard) como el que aparece en la Figura 3.12 desde distintas posiciones, y con ello calcular los parámetros de distorsión que introducen las lentes. Para calcular estos parámetros se usa un software que proporciona la aplicación ARToolkit, que se usará más adelante para generar un entorno de realidad aumentada. A este software se le introducen las imágenes con el tablero desde distintas posiciones y devuelve los parámetros para generar la distorsión inversa de la cámara acorde a la Ecuación 3.3.

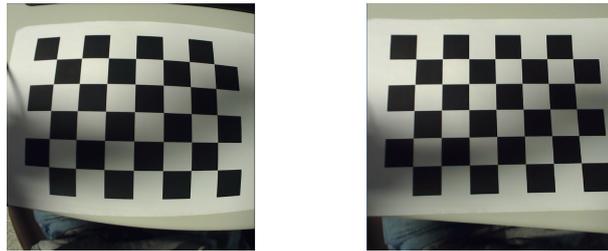


Figura 3.12: A la izquierda en esta figura se puede ver una imagen del tablero de calibración distorsionado y a la derecha se puede ver otra imagen del mismo tablero con la distorsión corregida.

Corregir la distorsión es un proceso que se debe ejecutar por cada píxel de cada imagen que genera cada una de las cámaras, ejecutar esto en el procesador del equipo disminuiría notablemente el rendimiento de la aplicación de realidad aumentada, por ello se implementa una solución que corrige esta distorsión se usa un *shader* sobre las texturas que están en los planos donde se reproduce el video. Utilizar un shader para corregir la distorsión es una idea que utilizan Steptoe et. al. [5], además el uso de este shader no tiene apenas impacto computacional, ya que es procesado por la tarjeta gráfica que es capaz de calcular muchas operaciones a la vez, y una de las virtudes de los shaders es que son capaces de computar cada píxel en uno de los muchos microprocesadores de la tarjeta gráfica. El *vertex shader* (fragmento de código se encarga de trabajar con los vértices de los polígonos de las mallas poligonales) y el *fragment shader* (fragmento de código que trabaja con los píxeles de la malla poligonal) que se usan para corregir la distorsión están listados a continuación.

```

//Vertex shader simple que coloca los vértices en el punto donde les corresponde
v2f vert (appdata_full v){
    v2f o;
    o.pos = mul(UNITY_MATRIX_MVP,v.vertex);
    o.uv_MainTex = TRANSFORM_TEX(v.texcoord, _MainTex);
    return o;
}
//Fragment shader que corrige la distorsión de las cámaras sustituyendo el color
de un pixel por el que le correspondería si no tuviera distorsión
sampler2D _MainTex;
float4 frag (v2f IN) : COLOR {
    float k1; //Primer parámetro de distorsión radial
    float k2; //Segundo parámetro de distorsión radial
    float k3; //Primer parámetro de distorsión tangencial
    float k4; //Segundo parámetro de distorsión tangencial
    float f1; //Distancia focal horizontal
    float f2; //Distancia focal vertical
    float ppx; //Posición x del pixel central
    float ppy; //posición y del pixel central
    //En este fragmento de código se convierte las coordenadas de textura UV
    (que va de 0 a 1) al tamaño que tendría en píxeles
    float xtext=IN.uv_MainTex.x;
    float ytext=IN.uv_MainTex.y;
    int xu=ppx*2*xtext;
    int yu=ppy*2*ytext;
    float x = (xu-ppx)/f1;
    float y = (yu-ppy)/f2;
    //A partir de aquí se aplican las ecuaciones de distorsión para generar
    una distorsión inversa capaz de corregir la que tiene la imagen
    float r2=(x)*(x)+(y)*(y);
    float r4=r2*r2;
    float dx=((2*k3*x*y)+(k4*(r2+(2*x*x))));
    float dy=(k3*(r2+(2*y*y)))+(2*k4*x*y);
    float xd=(1+k1*r2+k2*r4)*x+dx;
    float yd=(1+k1*r2+k2*r4)*y+dy;
    // A partir de aquí se vuelve a transformar todo en coordenadas
    de textura UV
    float xp = xd*f1 + ppx;
    float yp = yd*f2 + ppy;
    float xuv = xp/(ppx*2);
    float yuv = yp/(ppy*2);
    //Aquí se calcula el color que corresponde a la nueva posición del pixel y
    se devuelve
    float2 corrected=float2(xuv,yuv);
    float4 c=tex2D(_MainTex,corrected);
    return c;
}

```

3.4.6. Experimento 3

En este experimento se muestran los estímulos en un entorno de realidad aumentada. Para ello se ha usado la aplicación ARToolkit adaptada para funcionar con el Oculus como se describe en la Sección 3.4.4 y el controlador que modela el comportamiento de los estímulos descrito en la Sección 3.2.4.

3. Entorno de trabajo

La resolución ideal para a la que debería funcionar esta aplicación debería ser mayor o igual a la que tiene asignada para cada ojo el Oculus (960 x 1080 píxeles), pero debido a que el Hardware no es capaz de reproducir el contenido a la tasa de fps necesaria para que el Oculus funcione correctamente se ha fijado la resolución de 1024 x 768 píxeles con una proporción de 4:3, teniendo en cuenta que las cámaras están rotadas 90 grados para adaptarse mejor al FOV del Oculus. Al aplicar esta resolución a las cámaras, éstas recortan el FOV de manera significativa, teniendo 55-60 grados de FOV horizontal y 75-80 de vertical. Este FOV es menor que el que muestra el Oculus, provocando un efecto que hace que todo parezca más grande cuando se está dentro de la aplicación de realidad aumentada.

Se ha desplazado el estímulo visual colocándolo delante del marcador para que el color blanco de los bordes del marcador no interfieran con el contraste entre el plano y las esferas del estímulo visual.

En este experimento existe el riesgo de que se pierda de vista el marcador y entonces no se pueda visualizar el estímulo visual que corresponde en ese momento. Para evitar que el participante pierda el estímulo visual se modifica la parte que modela el movimiento de las esferas para que si se pierde referencia del estímulo visual el movimiento de las esferas se reinicie si todavía no han terminado el movimiento, con las mismas condiciones visuales y auditivas que tenía antes de perder de vista el marcador.

A lo largo de los experimentos, las esferas que componen al estímulo visual tienen un tamaño fijo, en este experimento es difícil fijar el tamaño ya que depende de la posición desde donde se observa el marcador. Para calcular el tamaño que tienen las esferas, la distancia a la que tienen que estar separadas y la velocidad a la que deben ir se usa la cámara virtual que captura la escena virtual. Ésta varía de posición y rotación cuando en la imagen que captura la cámara física se detecta el marcador. Con la posición de esta cámara virtual y su FOV, además de la posición y el tamaño de las esferas que componen la escena virtual se calculan las proporciones que va tener el estímulo visual en la pantalla del Oculus. Con esto se fija el marcador a una distancia del participante para que estas proporciones sean aproximadamente las mismas que en el resto de experimentos.

4. Desarrollo de los experimentos

En este proyecto se han planteado una serie de experimentos que involucran la teoría de la percepción *crossmodal*. Estos experimentos se han realizado en varios entornos inmersivos, en concreto la realidad virtual y la realidad aumentada. El diseño del experimento está basado en el trabajo de Sekuler et. al. [19], y busca analizar si la influencia *crossmodal* del sonido sobre estímulos visuales se produce cuando los estímulos se visualizan en un entorno inmersivo en lugar de en una pantalla tradicional. En esta sección se describe el diseño de los estímulos, así como el desarrollo de los experimentos.

4.1. Diseño de los estímulos

4.1.1. Descripción de los estímulos

En los experimentos realizados se ha tratado el tema de la influencia que tiene el sonido en un movimiento ambiguo como parte de la teoría de percepción *crossmodal*. Para ello se cuenta con una serie de estímulos visuales y auditivos, estos estímulos se basan en los que aparecen en el primer experimento planteado por Sekuler et. al. [19]. Estos estímulos, el experimento de Sekuler y el porqué de su elección se detalla en la Sección 2.3.3. El objetivo es llevar estos estímulos a entornos inmersivos para comprobar cómo evoluciona la influencia del sonido en el movimiento en dichos entornos. Las medidas y proporciones de los estímulos son las mismas que se usan en el experimento de Sekuler. Estas medidas vienen dadas en una unidad llamada ángulo visual. Este ángulo se corresponde al ángulo que tiene un objeto con respecto al punto desde donde observa el participante el experimento, esto se puede ver descrito gráficamente en la Figura 4.1.

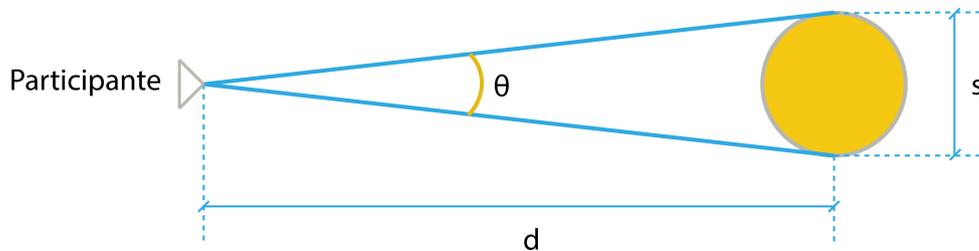


Figura 4.1: En esta figura se puede ver una descripción gráfica que facilita la comprensión del término ángulo visual. El término d es la distancia a la que está del participante lo que se mide, y el término s es el tamaño de lo que se está midiendo. Con estos dos términos se calcula θ que es el tamaño en unidades de ángulo visual del objeto que se mide.

4. Desarrollo de los experimentos

El estímulo visual consiste en dos discos de color marrón oscuro de 0.5 grados de ángulo visual, separados por una distancia de 4.2 grados uno de otro. Los discos al principio están detenidos sobre un plano, luego comienzan a moverse uno hacia el otro a una velocidad de 6 grados por segundo. El estímulo auditivo consiste en un breve sonido de 3 milisegundos, con una forma sinusoidal y una frecuencia de 2000 hz.

Tanto el estímulo auditivo como el estímulo visual aparecen bajo diferentes condiciones. El estímulo auditivo tiene cuatro posibles condiciones que consisten en no aparecer, aparecer en el momento de coincidencia de los discos, 150 milisegundos antes o 150 milisegundos después. El estímulo visual tiene tres posibles condiciones, que consisten en si los discos tienen un movimiento continuo, si tienen una pausa de un fotograma, o tienen una pausa de dos fotogramas. Las pausas del experimento medidas en fotogramas hacen referencia una medida que existe a la hora de mostrar videos, que es la tasa de fotogramas por segundo (fps), teniendo en cuenta que esto también depende del dispositivo donde se muestra. El experimento de Sekuler es del año 1997 y por eso se realizó en una pantalla analógica que reproduce contenidos a 25 fps, es decir que las pausas corresponden a 40 milisegundos la de un fotograma y 80 milisegundos la de dos fotogramas. Aunque en los experimentos y en los resultados las pausas se definan como de uno o dos fotogramas están adaptadas a los fps que corresponden en cada experimento.

El objetivo principal de usar estos estímulos es evaluar cómo influye el estímulo auditivo en la percepción del estímulo visual. El estímulo visual se puede percibir de dos formas distintas, que los discos se cruzan y no cambian la dirección de movimiento o que los discos rebotan y cambian la dirección de movimiento, esto se puede ver explicado de forma gráfica en la Figura 4.2.



Figura 4.2: En esta figura se muestra un esquema del movimiento de los discos que percibe un participante en el experimento. A la izquierda el movimiento que percibe el participante si cree que los discos se cruzan y a la derecha el movimiento que percibe el participante si cree que los discos rebotan.

4.1.2. Traslado de los estímulos a Unity3D

El objetivo de este proyecto es mostrar los estímulos visuales en un entorno inmersivo. Este entorno se modela con Unity3D. Para darle una componente tridimensional se modifica la forma de los discos que componen el estímulo visual por forma de esferas. Se modela una escena para colocar los estímulos.

La escena se compone de varias partes que son los estímulos, el entorno y la cámara. Los estímulos son los que componen el experimento, son los descritos anteriormente y son la parte esencial del experimento. El entorno es el resto de objetos que aparecen en la escena. La cámara es el objeto que une a la escena y al participante, este objeto es el que determina lo que será mostrado en el Oculus o en la pantalla del ordenador.

Para calcular las proporciones del experimento en la pantalla se deben conocer el tamaño

4. Desarrollo de los experimentos

de ésta y el campo de visión de la cámara que se usa en Unity3D para trasladar las medidas lo más precisamente posible a la pantalla. En la Figura 4.3 se puede ver una visión general de la traslación del experimento a la pantalla.

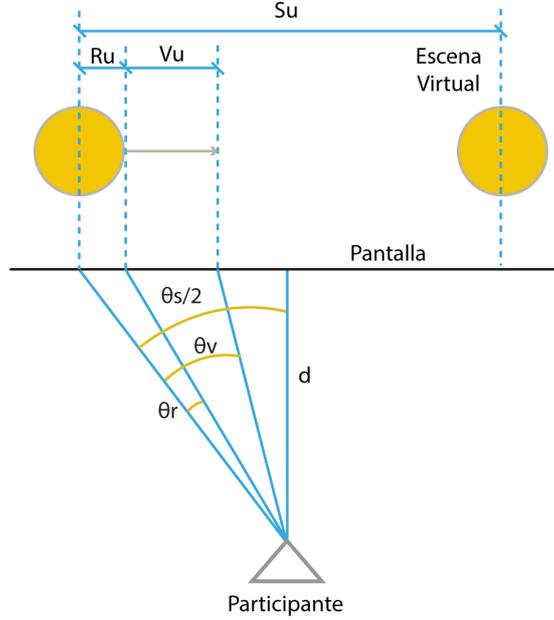


Figura 4.3: En esta figura se muestra la traslación de la escena de Unity a la pantalla, y los grados de ángulo de visión que tienen los estímulos visuales (θ_r), la separación de estos (θ_s) y la velocidad que llevan (θ_v).

Primero se busca el tamaño de cada unidad de Unity3D a la distancia de la cámara donde están los estímulos visuales en la escena, para ello usamos la Ecuación 4.1:

$$\tan(FOV/2) = \frac{h_u}{d_u} \quad (4.1)$$

$$(Unity \rightarrow World) = \frac{h_{pantalla/2}}{h_u}$$

donde d_u es la distancia a la que están los estímulos de la cámara, y h_u es la distancia entre el centro y la parte superior de la pantalla. Con esto se puede sacar la relación entre la distancia en Unity3D y el mundo real al saber las medidas del monitor que se usa.

El siguiente paso es calcular el tamaño que deberían tener los estímulos en pantalla, acorde a los tamaños del experimento original, asumiendo una distancia del usuario a la pantalla de 70 cm y la siguiente ecuación:

$$\tan(\theta/2) = \frac{t_{pantalla/2}}{d} \quad (4.2)$$

donde θ es el ángulo visual del que se quiere calcular el tamaño, d es la distancia que a la pantalla, y $t_{pantalla}$ es el tamaño que debe tener en pantalla para corresponderse con el ángulo visual.

4. Desarrollo de los experimentos

Con la relación que se ha calculado en la Ecuación 4.1 entre la pantalla y la escena de Unity3D, y el tamaño en la pantalla de un ángulo visual que se calcula en la Ecuación 4.2, se pueden generar los estímulos visuales que tengan las proporciones correctas en la pantalla.

Trasladar el experimento al Oculus es más complejo ya que al ser un dispositivo de tipo HMD provoca que no se pueda calcular de manera efectiva la medida del ángulo visual de las proporciones del estímulo visual, esto es debido a la distorsión que producen las lentes que hay dentro del Oculus. Debido a esto el ángulo visual es aproximado y se calcula desde el punto donde está situado el participante en la escena virtual donde se reproducen los estímulos, es decir desde el punto donde está situada la cámara. La forma en que están colocados los estímulos se puede ver en la Figura 4.4.

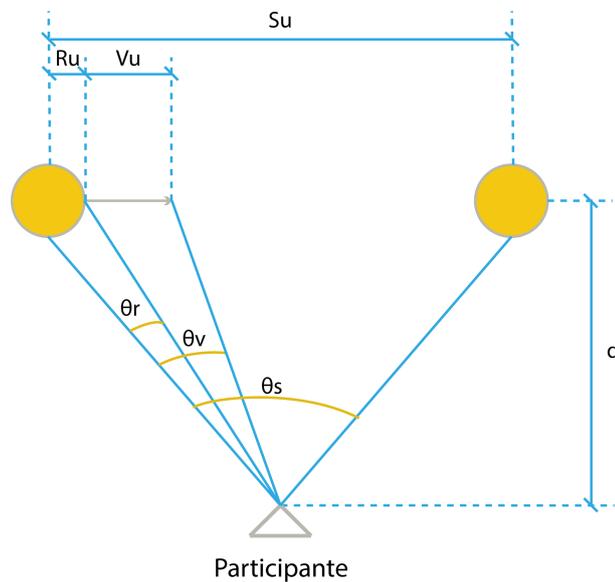


Figura 4.4: En esta figura se muestra cómo se mide el ángulo visual de las proporciones del estímulo visual, y la relación entre las medidas en ángulo visual (θ_r , θ_s y θ_v) y las medidas en unidades de Unity 3D (S_u , R_u y V_u). También se muestran los grados de ángulo visual que tienen las esferas que componen el estímulo visual (θ_r), la separación de estas (θ_s) y la velocidad que llevan (θ_v).

El primer paso para modelar la escena donde están los estímulos es colocar la cámara. Una vez colocada la cámara se fija una distancia d , que es la distancia de el estímulo visual a la cámara. Una vez colocado el estímulo, con la distancia d se calculan las medidas que debe tener el estímulo en unidades de Unity3D para tener las unidades en ángulo visual deseadas. El estímulo auditivo se modela como un sonido que se escuchará en el momento de colisión de las esferas que componen el estímulo visual. Los estímulos son mostrados combinando sus condiciones de aparición de manera aleatoria.

4.2. Experimento 1

4.2.1. Objetivo del experimento

En este proyecto se estudia la influencia un estímulo auditivo sobre una serie de estímulos visuales, es decir si se existe un efecto *crossmodal* entre ellos. Para ello se ha desarrollado este primer experimento donde hay varios estímulos visuales y varios estímulos auditivos. Los estímulos visuales están compuestos por dos esferas que se mueven una hacia la otra, se cruzan, y continúan moviéndose hasta que desaparecen. Los estímulos auditivos consisten en un sonido que aparece cuando las esferas coinciden, antes, después, o no aparece. Estos estímulos trabajan sobre la cuestión de si influye el sonido en la percepción de un movimiento ambiguo. Se puede percibir el estímulo visual de dos formas distintas, las esferas se perciben rebotando o cruzándose, como se ha descrito en detalle en la Sección 4.1.

Este experimento se ha dividido en dos bloques distintos, variando en éstos el dispositivo donde se muestra el experimento. Uno de los bloques se muestra en un monitor de ordenador, y el otro bloque se muestra en un dispositivo capaz de visualizar entornos inmersivos en 3D, en este caso en un HMD, el Oculus. El objetivo de dividir el experimento en estos dos bloques es estudiar si existe la influencia del sonido sobre la percepción del movimiento en un entorno inmersivo simple, y si existe una diferencia de esta influencia, con respecto a la visualización en una pantalla tradicional.

Para lograr este experimento se modela una escena en Unity3D donde se colocan los estímulos. El entorno que rodea los estímulos únicamente es un plano de color blanco debajo de éstos. La escena con el estímulo visual detenido y el entorno se puede ver en la Figura 4.5.

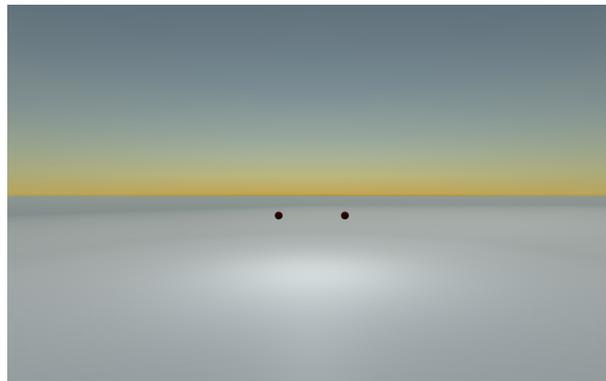


Figura 4.5: La figura muestra el estado inicial del primer experimento antes de que el estímulo visual, las esferas, comience a moverse.

4.2.2. Bloques del experimento

Este experimento se divide en dos bloques, la principal diferencia entre éstos es el dispositivo en el que se reproduce cada bloque. En ambos bloques se modela una escena con Unity3D como se describe en la Sección 4.1.

4. Desarrollo de los experimentos

En el primer bloque se reproduce el experimento en un monitor de ordenador estándar (Acer AL2216W TFT 22”), para poder calcular correctamente las medidas del estímulo visual se debe conocer las características de resolución, altura y anchura del monitor. Este bloque del experimento se reproduce con una tasa de 60 fps, entonces para cuadrar las pausas con las descritas en la Sección 4.1 se ha limitado la tasa de fotogramas por segundo a 50 mientras dura la pausa. Con esta limitación se pueden multiplicar los fotogramas que está pausado el experimento por dos y lograr las mismas pausas, en cuestión temporal, que el experimento original.

Para el bloque que se reproduce en el Oculus el experimento se reproduce con una tasa de 75 fps. En este caso se han multiplicado los pausas de las condiciones del estímulo visual por tres.

4.2.3. Procedimiento

Para la realización del experimento se guía al usuario mediante una serie de diapositivas que aparecen a lo largo de todo el experimento. Estas diapositivas explican al participante qué debe hacer. El experimento consta de dos bloques (pantalla tradicional y realidad virtual), que son los que se describen en la Sección 4.2.2, y aparecen en un orden aleatorio.

Antes del bloque de realidad virtual se lleva a cabo un entrenamiento, que consiste en ver dos esferas a diferente distancia como muestra la Figura 4.6. El usuario debe elegir cuál está más cerca, pulsando con el ratón el clic derecho si piensa que es la derecha, o pulsando con el ratón el clic izquierdo si piensa que es la izquierda. Las esferas aparecen durante 4 segundos y después aparece una diapositiva explicando cómo escoger la esfera más cercana. Este entrenamiento se hace para dos cosas. Para ver si el participante es capaz de distinguir la esfera más lejana, si no lo hace podría tener un problema de visión estereoscópica (que no es capaz de distinguir profundidad en una imagen en tres dimensiones). También es útil para que los participantes se acostumbren al uso del ratón sin verlo (el hecho de llevar el Oculus impide ver el mundo real).

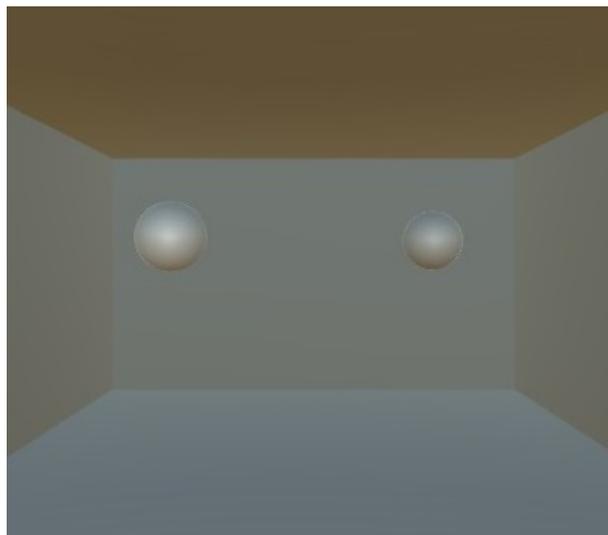


Figura 4.6: En esta imagen se muestra una escena del entrenamiento que se realiza antes del bloque de realidad virtual.

4. Desarrollo de los experimentos

En cada bloque del experimento se presentan las combinaciones de condiciones del estímulo auditivo y el estímulo visual explicadas en la sección 4.1 un total de 10 veces cada combinación posible, en total los estímulos aparecen 120 ocasiones, con un orden totalmente aleatorio. Las condiciones auditivas que aparecen en este experimento son en las cuales no aparece el estímulo auditivo, en las que aparece antes del momento de coincidencia de las esferas, cuando coinciden y 150 ms después. Las visuales consisten en las cuales las esferas no se detienen en el momento de coincidencia y se detienen durante uno y dos fotogramas. Después de ver cada combinación de estímulos del experimento aparece una diapositiva como en la Figura 4.7 en el que aparece la frase *did the spheres bounce or stream?*, esto indica a los usuarios que deben responder si han percibido las esferas rebotar o cruzarse.

Cada bloque dura aproximadamente 10 minutos. En el bloque de realidad virtual aparece una diapositiva indicando al participante que puede descansar en mitad del experimento para evitar la fatiga visual asociada al uso de HMDs.

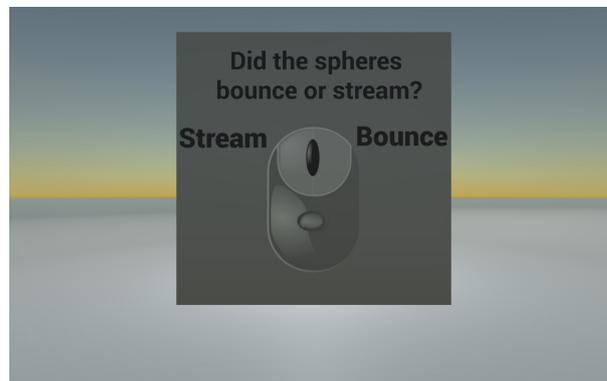


Figura 4.7: En esta imagen se muestra un ejemplo de la diapositiva que aparece cuando el usuario debe escoger si le ha parecido si las esferas rebotaban o se cruzaban.

4.3. Experimento 2

4.3.1. Objetivo del experimento

Los resultados del Experimento 1 (detallados en la Sección 5.2) indican que el sonido sigue influyendo en la percepción de un movimiento ambiguo aun estando en un entorno inmersivo. Con esta información se ha desarrollado otro experimento. Los estímulos que aparecen siguen siendo los detallados en la Sección 4.1, pero en este experimento se han modificado ciertas características del estímulo visual. También se han eliminado alguna de las condiciones de los estímulos para acortar la duración del experimento. Además se añade más complejidad al entorno que rodea la escena. Al contrario que el primer experimento, este segundo experimento se ha realizado solo en un entorno de realidad virtual, que se reproduce en el Oculus.

El objetivo de los cambios realizados en el estímulo visual es analizar la influencia de la complejidad del entorno en la percepción del efecto *crossmodal*. El estímulo visual se ha modificado variando alguna de sus propiedades como el material, el tamaño, la velocidad, la separación o

4. Desarrollo de los experimentos

la dirección de movimiento de las esferas. Estas modificaciones se han dividido en tres bloques. Además, al entorno que rodea el experimento se le ha añadido más complejidad, haciendo que lo que rodea a las esferas parezca una habitación. También se ha modificado el plano que sobre el que se mueven los estímulos visuales, ahora tiene forma de mesa, conservando el contraste entre las esferas y el plano existente en el primer experimento.

Con respecto a las condiciones de los estímulos descritas en la Sección 4.1 en este experimento se ha eliminado una condición del estímulo auditivo y otra del estímulo visual. La condición del estímulo auditivo eliminada es en la que el estímulo auditivo aparece 150 milisegundos antes de la coincidencia de las esferas. La condición del estímulo visual eliminada es en la que aparece una pausa de un fotograma en el momento de coincidencia de las esferas. Este descarte de condiciones se ha hecho analizando los resultados del primer experimento y dada la necesidad de reducir la duración de este experimento debido a la fatiga visual provocada por el Oculus.

Para crear la escena compleja que rodea al experimento se ha modelado en Unity3D una habitación simple. La habitación se ha rodeado con paredes a las que se les ha añadido una textura para que parezca compuesta por láminas de madera, se ha añadido una textura embaldosada al suelo, las esferas se mueven sobre una mesa a la que no se le ha añadido textura y se ha dejado con el mismo color del plano que se tenía en el experimento anterior. Además en la habitación se han colocado una serie de objetos decorativos como cuadros, en concreto tres, un sofá de dos plazas y un sillón. También se ha cambiado la forma de mostrar las preguntas e indicaciones al usuario, en este caso se ha colocado un cuadro bastante amplio en frente del usuario en el que su contenido será los carteles que se mostraban en el experimento anterior, si en algún momento no corresponde mostrar un cartel, véase el momento en el cual las esferas se mueven, se ve un cartel sin texto en el interior. Todo esto se puede ver en la Figura 4.8.



Figura 4.8: Izquierda: Esta figura muestra la escena que rodea al experimento, se pueden ver las texturas de las paredes y el suelo, también los sofás que rodean la escena. Esta figura también muestra un ejemplo de una de las modificaciones del estímulo visual, en concreto, en la que se aumentan sus proporciones únicamente. Centro: Esta figura muestra el estímulo visual con las proporciones aumentadas, con un color más vivo, reflejos especulares y sombras en el plano de movimiento de las esferas. Derecha: Esta figura muestra el estímulo visual con el plano de movimiento rotado, el movimiento que las esferas describen es paralelo a los bordes más largos del plano, moviéndose una hacia a la otra.

4.3.2. Bloques del experimento

Mientras que en el primer experimento se analiza si el sonido influye en la percepción de un movimiento ambiguo en un entorno inmersivo, en este experimento se busca analizar el comportamiento de esta influencia cuando se modifican varios parámetros del estímulo visual, ya sea las proporciones, propiedades del material, o la forma en que el participante observa el

4. Desarrollo de los experimentos

estímulo visual. En todos los bloques se modela una escena 3D con los objetos que contienen el experimento, en este caso toda la habitación que rodea el experimento y las modificaciones que corresponden al estímulo visual en cada bloque.

En el primer bloque se ha modificado las proporciones del experimento teniendo en cuenta que el ángulo visual y el valor en unidades de Unity3D. Se ha multiplicado por dos todas las medidas del estímulo visual manteniendo el color marrón oscuro del primer experimento, las dos esferas tienen de 1 grado de ángulo visual, separadas por una distancia de 8.4 grados una de otra. Las esferas al principio están detenidas sobre un plano, luego comienzan a moverse una hacia la otra a una velocidad de 12 grados por segundo. Este aumento de velocidad puede hacerse linealmente con respecto al radio porque con ángulos visuales pequeños y por tanto se puede aproximar, como se ve en la Figura 4.9 y la Ecuación 4.3,

$$\tan(\theta_v) = \frac{r+v}{d} \quad (4.3)$$

$$\theta_v \approx 0 \implies \tan(\theta_v) \approx \theta_v$$

para un valor pequeño de θ_v , el incremento de θ_v es lineal cuando se aumenta r . El primer bloque del experimento se puede ver en la Figura 4.8.

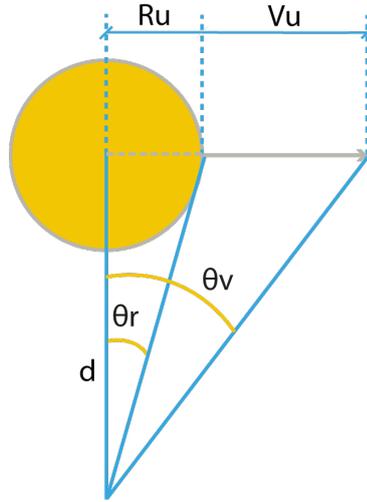


Figura 4.9: La figura muestra el incremento de θ_v cuando se aumenta r . Para pequeñas θ_v , el incremento θ_v es lineal cuando se incrementa r , como se ve en la ecuación 4.3.

En el segundo bloque del experimento se ha mantenido el tamaño del primer bloque y además se le ha añadido a las propiedades del material de las esferas brillos especulares, que se puede ver mejor porque se ha aumentado el número de luces en la escena y se ha cambiado el color de las esferas a rojo. También se han aumentado la distancia al plano sobre el que se mueven las esferas para que se puedan ver sombras claras debajo de éstas. El segundo bloque del experimento se puede ver en la Figura 4.8 (centro).

En el tercer bloque del experimento también se ha mantenido el tamaño del primer bloque, pero no se ha mantenido ninguna de las propiedades del bloque anterior. En este bloque el cambio que se ha realizado ha sido rotar el plano sobre el que se mueven las esferas en el eje Y,

variando también la forma en la que ve el participante observa el movimiento. La rotación del plano de movimiento de las esferas se puede ver en la Figura 4.8 (derecha).

4.3.3. Procedimiento

Como en el primer experimento, después de construir el experimento como se ha descrito en las secciones anteriores, se han realizado estudios de usuario. En este caso, se realizó el mismo entrenamiento que en el primer experimento, pero al comenzar el estudio de usuario, ya que todo se realiza en el dispositivo de realidad virtual.

Los bloques del experimento se muestran en orden aleatorio y en cada bloque del experimento se muestran las combinaciones de condiciones del estímulo auditivo y del estímulo visual explicadas en la Sección 4.1 un total de 10 veces cada combinación posible, con un orden totalmente aleatorio, en total se muestran 60 veces los estímulos por bloque. Las condiciones auditivas que aparecen en este experimento son en las cuales no aparece el estímulo auditivo, en las que aparece en el momento de coincidencia de las esferas y en las que aparece 150 ms después. Las visuales consisten en las cuales las esferas no se detienen en el momento de coincidencia y se detienen durante dos fotogramas. Después de ver cada una de estas combinaciones aparece el mismo cartel que en el experimento anterior, esta vez ve en la pared de enfrente del participante, con la frase *did the spheres bounce or stream?*, que indica al participante que debe responder si ha percibido las esferas rebotar o cruzarse. El cartel en el marco que se puede ver en la Figura 4.8. Las explicaciones e indicaciones del experimento aparecen también en el mismo marco. Cada bloque del experimento tiene una duración aproximada de 6-7 minutos y el participante puede realizar un descanso entre cada bloque, indicado por una diapositiva. La duración del experimento está entre 20 y 25 minutos.

4.4. Experimento 3

4.4.1. Objetivo del experimento

En este experimento se han trasladado los estímulos a un entorno totalmente realista. Se ha modelado un experimento con los estímulos que aparecen en la Sección 4.1 en un entorno de realidad aumentada, este experimento está modelado con Unity3D y se reproduce en su totalidad en el Oculus. La construcción y características del sistema de realidad aumentada aparecen detallados a en Sección 3.4.

Las principales diferencias entre este experimento y los otros dos realizados es el entorno de realidad aumentada. El estímulo auditivo se ha mantenido igual que el que aparece en la Sección 4.1, pero al estímulo visual se le ha aumentado el tamaño como en el segundo experimento y también se han cambiado el resto de proporciones del experimento, teniendo duplicadas todas las proporciones como en el segundo experimento. El tamaño de las esferas es de 1 grado de ángulo visual, separadas por una distancia de 8.4 grados una de otra. Las esferas al principio están detenidas sobre un plano, luego comienzan a moverse una hacia la otra a una velocidad de 12 grados por segundo. Este tamaño es aproximado ya que depende de la distancia a la que el usuario observe el marcador de realidad aumentada, para eso se fija al usuario a una distancia

4. Desarrollo de los experimentos

aproximada de 70 cm de la mesa y se configura el tamaño de las esferas para que tengan las proporciones que se han descrito anteriormente.

En este experimento en vez de tener unas esferas oscuras moviéndose sobre un plano de un color más claro se han intercambiando los colores, quedando un plano más oscuro y unas esferas más claras. Se ha intentado mantener el mismo contraste que hay entre el plano y las esferas del primer experimento. Se ha optado por cambiar el color del plano a un color muy oscuro por la dificultad de simular las sombras de las esferas sobre una imagen capturada para la realidad aumentada. El experimento se puede ver en la Figura 4.10.



Figura 4.10: Esta figura muestra el estímulo visual en el entorno de realidad aumentada el cual rodea al tercer experimento, también se puede ver el marcador que se usa para colocar en este entorno el estímulo visual.

Las condiciones que hay en este experimento son las mismas que aparecen en el segundo experimento. Se mantienen las condiciones en las que no aparecía pausa en el momento de coincidencia de las esferas y en las que aparecía una pausa de dos fotogramas. En cuanto a las condiciones auditivas se mantienen en las cuales el sonido no aparece, aparece en el momento de coincidencia de las esferas o aparece 150 milisegundos después del momento de coincidencia.

4.4.2. Procedimiento

Una vez modelado el experimento, se realizan estudios de usuario. En este caso no hay entrenamiento pero sí que hay al principio del experimento una fase en la cual los participantes pueden explorar el entorno de realidad aumentada, pudiendo mirar alrededor y experimentar con la realidad aumentada. Después de esta fase comienza el experimento.

En este experimento se muestran cada combinación de condiciones un total de 10 veces cada posible combinación, con un orden totalmente aleatorio, apareciendo 60 veces los estímulos. Las condiciones auditivas que aparecen en este experimento son en las cuales no aparece el estímulo auditivo, en las que aparece en el momento de coincidencia de las esferas y en las que aparece 150 ms después. Las visuales consisten en las cuales las esferas no se detienen en el momento de coincidencia o se detienen durante dos fotogramas. Para mostrar los carteles indicadores se muestra un entorno como el que se ve en la Figura 4.11, cada vez que termina el movimiento de

4. Desarrollo de los experimentos

las esferas, éstas se mantienen unos segundo quietas y después desaparece el entorno de realidad aumentada para que aparezca el entorno donde se muestra el cartel que en el que aparece la frase *did the spheres bounce or stream?*, que indica al participante que debe responder si ha percibido las esferas rebotar o cruzarse. La duración de este experimento es de 10 minutos aproximadamente. En este experimento no se realiza entrenamiento previo porque los usuarios pueden ver el ratón, pero antes de realizar el experimento se comprueban las capacidades de visión estéreo de los usuarios.

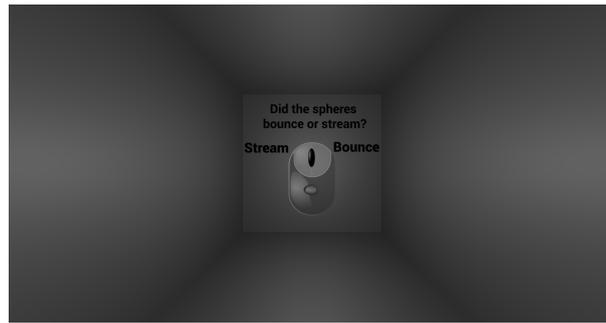


Figura 4.11: Esta figura muestra una diapositiva de las que aparecen en el experimento de realidad aumentada, son similares a las que aparecen en el resto de experimentos.

5. Análisis y resultados

5.1. Pre-procesado de los datos

5.1.1. Recolección de los datos

Al realizar los estudios de usuario, en los tres experimentos, los usuarios deben contestar a una misma pregunta una serie de veces después de cada estímulo del experimento, *Did the spheres bounce or stream?*, es decir a si han percibido que las esferas rebotaban o se cruzaban, variando las condiciones de sonido y de movimiento de las esferas cada vez. Las respuestas a la pregunta se almacenan en un fichero. Además a esta respuesta se le añaden códigos que definen la condición de movimiento y la condición auditiva que han aparecido cada vez que aparecen los estímulos, y el tiempo desde que aparece el cartel para responder a la pregunta hasta que hace click con el ratón. Esto se almacena en un fichero distinto por participante, experimento y bloque de ese experimento. También se almacenan las respuesta registradas en el entrenamiento. Teniendo para cada participante en un experimento un fichero que son los resultados del entrenamiento y tantos ficheros como bloques tenga el experimento realizado.

Con todos estos datos almacenados, se han unido los resultados de todos los participante en cada bloque, estando estos ordenados, así se ha logrado tener un fichero para cada bloque del experimento con el que se hará más sencillo calcular los datos.

Después de esto los ficheros se han importado con el software matemático llamado Matlab, donde se ha desarrollado el análisis de los datos.

5.1.2. Descarte de espurios

Un descarte de datos espurios consiste en eliminar aquellos valores que no son correctos con algún criterio específico. En este proyecto eliminar datos espurios puede ayudar a descartar valores de participantes que no han entendido en que consistía el experimento, o que podían tener problemas de visión estereoscópica.

En este proyecto se han descartado datos espurios de dos formas distintas, la primera basándose en los resultados del entrenamiento realizado en cada experimento para descartar usuarios que no sean capaces de percibir estéreo correctamente. Se ha calculado el porcentaje de veces que el participante selecciona correctamente la esfera más cercana y si este es menor o igual al 70 % se han eliminado las muestras de los datos de ese participante para el experimento,

este descarte se ha hecho para el Experimento 1 y el Experimento 2, el caso del Experimento 3 ha sido diferente ya que no era necesario realizar un entrenamiento previo para adaptarse al uso del ratón sin verlo, por lo que se ha usado directamente un test de estereopsis (*test de Titmus*). Con esto se descarta a participantes que pueden tener problemas de visión estereoscópica, es decir, que no son capaces de distinguir profundidad en un entorno en 3D.

La otra forma de eliminar datos espurios se ha realizado calculando los porcentajes de veces que cada participante ver rebotar las esferas para cada condición en cada bloque del experimento del que se calculan los datos. Se han usado los cuartiles primero y tercero (Q_1 y Q_3), y la diferencia entre estos, para encontrar datos espurios para cada condición. Si encontramos que una condición no pasa las siguientes inecuaciones se eliminan los datos de esa condición en el participante:

$$\begin{aligned} condicion &< (Q_1 - K_d * Q_d) \\ condicion &> (Q_3 + K_d * Q_d) \end{aligned} \tag{5.1}$$

con $Q_d = Q_3 - Q_1$ y $K_d = 1,5$. Además si un participante tiene marcada más de una condición como espuria, se eliminan todos los datos de ese participante.

5.1.3. Análisis de los datos

Para analizar los datos se han seguido tres líneas distintas de mostrar los datos. Dos de ellas han sido crear gráficos de barras, y otra, más compleja, ha sido realizar un análisis de varianza que permite sacar conclusiones sólidas entre factores del experimento.

La primera forma de crear gráficos consiste en crear un gráfico de barras por cada condición de movimiento en cada bloque del experimento, teniendo así múltiples gráficos de barras que contienen todas las condiciones de sonido. En estas barras se indica la media del porcentaje veces que ve cada participante rebotar las esferas. A estas barras se le ha añadido el SEM (*standard error of the mean* o *error estándar de la media* en castellano), que se calcula de la siguiente manera:

$$SEM = \frac{s}{\sqrt{n}} \tag{5.2}$$

con s como la desviación estándar de la muestra de datos y n el número de datos que contiene la muestra.

La siguiente forma de crear gráficos muestra los datos agregados, es decir una gráfica para cada bloque, otra gráfica para cada condición de sonido, y otra para cada condición de movimiento. Para ello se ha calculado el porcentaje de veces que ha visto el participante rebotar las esferas en cada uno de los bloques para las gráficas de bloques, en cada una de las condiciones de movimiento para las condiciones de movimiento y en cada una de las condiciones de sonido para cada condición de sonido. A esto también se le ha añadido el SEM para cada una de las barras de la gráfica.

Con los datos de los resultados de los experimentos también se ha realizado un análisis más complejo, que consiste en un análisis de varianza, ANOVA por sus siglas en inglés, que ayuda

contrastar hipótesis como si un factor del experimento tiene influencia sobre otro. Para mostrar los resultados de este análisis se contrastan distintos factores con el porcentaje de veces que se ha visto rebotar las esferas, los resultados que se muestran son los valores del F-test y la significancia, con un valor del F-test alto y un valor de significancia menor que 0.05 quiere decir que el factor en cuestión influye sobre el porcentaje veces que los participantes han percibido rebotar las esferas. En concreto el análisis de varianza que se realiza aquí es uno específico que se tiene en cuenta que los datos son de los mismos participantes en distintas situaciones, este tipo de análisis de varianza se llama *repeated measures ANOVA*. En el tercer experimento se ha descartado hacer un análisis de varianza debido a los pocos factores que se tienen.

5.2. Experimento 1

El primer experimento que se realiza en este proyecto busca encontrar si existe la influencia del sonido sobre la percepción del movimiento en un entorno inmersivo simple, y si existe una diferencia de esta influencia entre este entorno y uno no inmersivo, el experimento se desarrolla conforme a la Sección 4.2. En este primer experimento han participado 13 usuarios. Todos los participantes realizaron el experimento en ambos entornos, con un orden aleatorio. En primer lugar se busca mediante la realización de un análisis de varianza si el sonido, la pausa en el movimiento, y el dispositivo tienen influencia sobre la percepción de rebote de las esferas, se analizan los valores uno a uno con respecto al porcentaje de rebote. Los resultados se pueden ver en la Tabla 5.1. Para marcar un factor como influyente en porcentaje de veces que se perciben las esferas rebotando debe tener un valor de significancia menor que 0.05.

Tabla 5.1: Resultados (*F-test* y *significancia*) del análisis de los datos con *repeated measures ANOVA* para el Experimento 1. Se ha comprobado la influencia de los tres factores en el porcentaje de rebotes percibidos por los participantes.

	F	Sig.
Sonido vs porcentaje de rebotes	83.664	0.000
Pausa vs porcentaje de rebotes	63.528	0.000
Dispositivo vs porcentaje de rebotes	13.176	0.000

Por otro lado se pueden observar las gráficas de datos agregados por factor sobre el porcentaje de veces que se han percibido las esferas rebotar, esto se puede ver en la Figura 5.1, en estas gráficas se puede observar una reducción de este porcentaje al reproducir el experimento en el Oculus con respecto a cuando se reproduce en la pantalla, esto puede ser debido a la inclusión de indicadores de profundidad al estar en un entorno de 3D. También se ve que la aparición de sonido aumenta el porcentaje de rebotes que se perciben, siendo el aumento más grande cuando aparece el sonido justo en el momento de coincidencia y un poco menor en los otros dos casos donde aparece el sonido antes y después. Por último también aparece que la pausa también influye en el porcentaje de percepción de rebote de las esferas, aumentando dicho porcentaje en los casos donde las pausas corresponden a más fotogramas.

La tendencia de los datos no agregados que aparecen en la Figura 5.2 es similar a la tendencia de los resultados del trabajo de Sekuler et. al. [19]. Sin embargo se puede ver en que en el bloque que se reproduce en el Oculus disminuyen los porcentajes de rebotes percibidos.

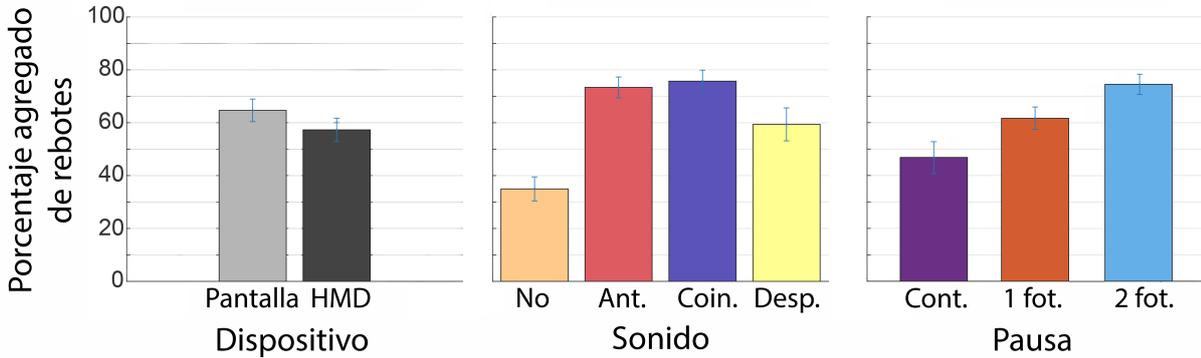


Figura 5.1: Porcentajes agregados de los rebotes percibidos y sus barras de error correspondientes para el Experimento 1. De izquierda a derecha: Porcentajes para los dos dispositivos donde se reproduce el experimento (pantalla o HMD), porcentajes para las cuatro condiciones auditivas (sin sonido, sonido 150 ms antes, sonido en el punto de coincidencia y sonido 150 ms después) y porcentajes para las tres condiciones visuales (movimiento continuo, pausa de un fotograma o pausas de dos fotogramas en el punto de coincidencia de las esferas).

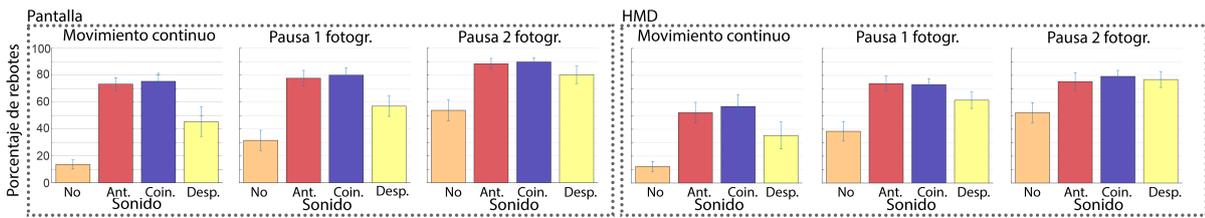


Figura 5.2: Porcentajes (no agregados) y barras de error para el Experimento 1. Izquierda: Resultados para la pantalla. Derecha: Resultados para el HMD (Oculus). En ambos casos de izquierda a derecha: Gráficos para cada una de las tres condiciones de pausa (movimiento continuo, pausa de un fotograma y pausa de dos fotogramas en el punto de coincidencia de las esferas). Cada uno de los tres gráficos muestra el porcentaje de rebotes reportado por los participantes para cada una de las condiciones auditivas (sin sonido, sonido 150 ms antes, sonido en el punto de coincidencia y sonido 150 ms después).

5.3. Experimento 2

En el segundo experimento se busca estudiar si el efecto *crossmodal* que aparece en el primer experimento cuyos datos se analizan en la Sección 5.2 se mantiene en una escena más compleja. La construcción de este experimento, sus bloques y condiciones aparecen relacionados en la Sección 4.3. De este experimento se tienen muestras de 27 participantes.

Como en el experimento anterior también se realiza un análisis de varianza sobre los datos teniendo en cuenta tres factores, en este caso la influencia de la escena, la influencia del sonido cuando coinciden las esferas y la influencia de la pausa en el punto de coincidencia de las esferas. Con este análisis de varianza se busca determinar si estos tres factores influyen en la percepción de las esferas rebotando. Los resultados de este análisis de varianza se pueden ver en la Tabla 5.2.

En el caso del factor de sonido, el valor de significancia del análisis de varianza indica que influye en la percepción de las esferas rebotando, como el factor de pausa, en cambio no se puede determinar que el factor de escena influya o no en el hecho de que se perciban las esferas rebotar. En la Figura 5.3 se puede ver que en el caso de la escena el porcentaje rebotes percibido está en

5. Análisis y resultados

Tabla 5.2: Resultados (*F-test* y *significancia*) del análisis de los datos con *repeated measures ANOVA* para el Experimento 2. Se ha comprobado la influencia de los tres factores en el porcentaje de rebotes percibidos por los participantes.

	F	Sig.
Sonido vs porcentaje de rebotes	124.137	0.000
Pausa vs porcentaje de rebotes	845.386	0.000
Escena vs porcentaje de rebotes	0.220	0.977

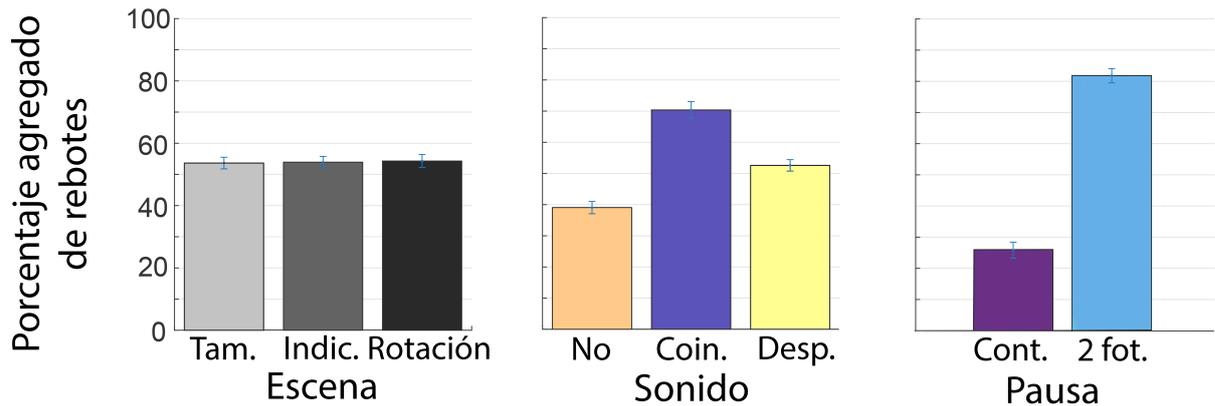


Figura 5.3: Porcentajes agregados de los rebotes percibidos y sus barras de error correspondientes para el Experimento 2. De izquierda a derecha: Porcentajes para las tres escenas diferentes (esferas más grandes, esferas especulares y con más sombras, esferas con el movimiento rotado); porcentajes para las tres condiciones auditivas (sin sonido, sonido en el punto de coincidencia y sonido 150 ms después) y porcentajes para las dos condiciones visuales (movimiento continuo o pausas de dos fotogramas en el punto de coincidencia de las esferas).

casi el mismo nivel para todas las escenas.

Si se comparan los resultados que aparecen en la Figura 5.3 con los que aparecen en el experimento anterior en la Figura 5.1 se puede decir que el efecto *crossmodal* se mantiene con la misma tendencia, pero en este experimento aparece un leve descenso del porcentaje rebotes percibidos. Esto puede ser debido a la aparición de indicadores adicionales al aplicar más realismo a la escena. Sin embargo el valor de la significancia del factor de la escena no permite decir si hay una diferencia significativa en el efecto *crossmodal* en ninguna de las tres escenas, es decir no se puede afirmar que ningún indicador de los introducidos afecte al efecto *crossmodal* de manera significativa.

5.4. Experimento 3

En este experimento se llevan los estímulos que aparecían en los anteriores a un entorno todavía más realista, por ello se ha decidido que este experimento transcurra en un entorno de realidad aumentada. La construcción de este experimento se relata en la Sección 4.4. Este experimento se realizó con 12 participantes.

Los resultados agregados de este experimento se pueden ver en la Figura 5.4. En estos

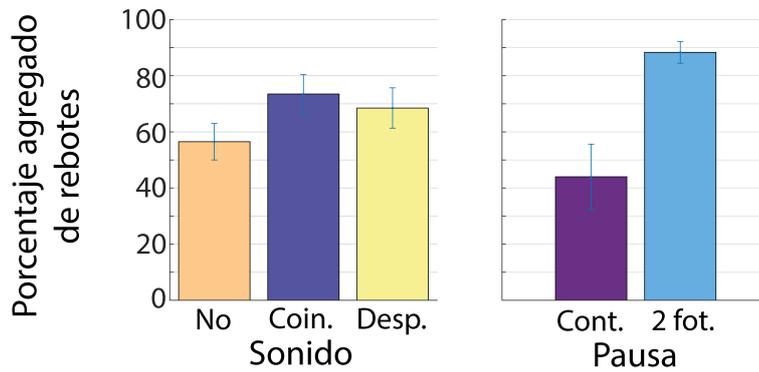


Figura 5.4: Porcentajes agregados de los rebotes percibidos y sus barras de error correspondientes para el Experimento 3. Porcentajes para las tres condiciones auditivas (sin sonido, sonido en el punto de coincidencia y sonido 150 ms después) a la izquierda y porcentajes para las dos condiciones visuales (movimiento continuo o pausas de dos fotogramas en el punto de coincidencia de las esferas) a la derecha.

gráficos se puede ver que, si bien la tendencia que aparece en los experimentos anteriores se mantiene, aumentan en gran medida los porcentajes de rebotes percibidos en todos los casos, siendo bastante significativo el aumento de en las condiciones donde no aparece sonido. De esta manera se pierde parte de la diferencia entre la condición sin sonido y el resto de condiciones auditivas. Esto puede deberse a que, aunque el tamaño del estímulo visual no cambie, es necesario ver el marcador de realidad aumentada con cierta inclinación, en este caso 30 grados, lo que provoca que la inclinación con la que se observa el plano que está debajo de las esferas sea diferente a la del resto de experimentos, en los cuales la inclinación era de aproximadamente 7 grados. A consecuencia de esta inclinación se podría deducir que es más fácil ver que las esferas que forman parte del estímulo visual están en el mismo plano y que físicamente no pueden cruzarse. Esto puede tener todavía un efecto más fuerte en realidad aumentada ya que al estar en un entorno real (y no virtual) el cerebro interprete que los objetos virtuales tienen propiedades físicas e induzca al participante a percibir las esferas rebotar. Adicionalmente, el hecho de que los porcentajes de la Figura 5.4 cuando no hay pausa no sea muy alto podría deberse a que cuando en la realidad se ven dos esferas rebotando se espera algún tipo de cambio en el movimiento, como el que se introduce en el caso de la condición donde se introduce una pausa, o la aparición del estímulo auditivo.

6. Conclusiones y trabajo futuro

En este proyecto se llevado a cabo una exploración de la percepción *crossmodal* en varios entornos inmersivos, todos ellos se visualizan a través de un HMD. Se ha estudiado la influencia de las señales auditivas en la percepción visual de un movimiento. Primero se ha replicado un experimento existente para demostrar la existencia de la interacción *crossmodal* entre ambos sentidos con unos estímulos simples en una pantalla estándar. Al replicarlo y realizar estudios de usuario se encontraron las mismas tendencias en los resultados que en el experimento original. Con estos resultados se replicó el experimento en un entorno inmersivo, utilizando el Oculus. Se encontraron unas tendencias similares en el Oculus (los factores explorados tienen la misma influencia en el efecto *crossmodal*), pero hay una reducción del efecto *crossmodal*. Esta reducción indica que hay un cambio en los resultados por el hecho de percibir un porcentaje mayor de veces el comportamiento real de las esferas que componen el estímulo visual, es decir que los participantes perciben un porcentaje mayor de veces las esferas cruzándose. Esto puede ser debido a indicadores adicionales, como indicadores de profundidad incluidos en la disparidad binocular que genera el Oculus. Una conclusión similar se puede obtener del Experimento 2. En este experimento se ha repetido el Experimento 1 (solo en el Oculus), con nuevos sujetos, y con unos estímulos más complejos para comprobar que el efecto *crossmodal* se mantiene con un escenario más realista. Se puede observar una reducción del efecto *crossmodal* (los sujetos analizar mejor el comportamiento del experimento, es decir perciben un porcentaje mayor de veces las esferas cruzándose), se puede suponer que esto puede producirse debido a la presencia de indicadores adicionales, en este caso pictóricos (sombras, perspectiva, texturas). Por último se ha desarrollado el Experimento 3 en un entorno de realidad aumentada, en este experimento se ha vuelto a reproducir el Experimento 2 en un entorno de realidad aumentada, para el cual también se ha usado el Oculus. Los resultados muestran que el efecto *crossmodal* sigue con tendencias similares, pero se ha visto reducido. Esta reducción es completamente distinta a la que aparece en los experimentos anteriores ya que aumentan considerablemente el porcentaje de rebotes percibidos, esto puede ser debido a que en un entorno tan realista como la realidad aumentada el cerebro interprete que las leyes de la física se aplican a los objetos virtuales. También esta reducción puede ser debida a que el ángulo con el que percibe el sujeto los estímulos cambia debido a limitaciones del entorno de realidad aumentada lo que provoca que ambas esferas se vean en el mismo plano, y por tanto la percepción de rebote aumente.

En conclusión, los efectos *crossmodales* de la interacción entre dos modalidades sensoriales se siguen manteniendo en entornos inmersivos, en este caso la influencia del sonido en un movimiento ambiguo. Pero en concreto esta influencia se reduce en los entornos inmersivos debido a la introducción de indicadores, ya sea de profundidad, pictóricos o físicos.

Por otra parte los estudios realizado sobre la percepción *crossmodal* en este proyecto son solo

6. Conclusiones y trabajo futuro

un primer paso, y todavía es necesaria mucha más investigación para poder aplicar la percepción *crossmodal* en entornos inmersivos. En el futuro, se pueden expandir estos experimentos de varias maneras, como por ejemplo incluyendo otros factores o efectos influyentes, e incrementado la complejidad de los estímulos. Comprender mejor las interacciones sensoriales es un requisito indispensable para hacer que las conclusiones sean realmente utilizables en entornos inmersivos. Adicionalmente se requiere un análisis adicional de las interacciones de primer orden de los factores estudiados para entender mejor su influencia en la percepción *crossmodal*.

Adicionalmente, parte de este proyecto (Experimento 1 y Experimento 2) se presentó como un artículo científico al Congreso Español de Informática Gráfica 2016, donde fue aceptado y galardonado como uno de los mejores artículos. Esto conlleva una invitación a publicar una ampliación del artículo en la revista “Computer Graphics Forum” (JCR Q1 en el campo).

Desde una perspectiva personal, para mí este proyecto de investigación ha resultado muy instructivo, ya que he podido comprobar desde dentro cómo trabaja un grupo de investigación dentro de la universidad. También ha aportado a mi enseñanza una nueva visión, ya que en este proyecto me he enfrentado a problemas reales. Además este proyecto ha implicado la adquisición de una gran cantidad de conceptos nuevos que todavía no conocía.

Bibliografía

- [1] Ganesh Ramanarayanan, James Ferwerda, Bruce Walter, and Kavita Bala. Visual equivalence: Towards a new standard for image fidelity. *ACM Trans. Graph.*, 26(3), July 2007.
- [2] François X. Sillion, Holly Rushmeier, and Julie Dorsey. *Digital Modeling of Material Appearance*. Morgan Kaufmann/Elsevier, 2008.
- [3] Belen Masia, Gordon Wetzstein, Piotr Didyk, and Diego Gutierrez. A Survey on Computational Displays: Pushing the Boundaries of Optics, Computation, and Perception. *Computers & Graphics*, 37(8):1012 – 1038, 2013.
- [4] Ann McNamara, Katerina Mania, and Diego Gutierrez. Perception in graphics, visualization, virtual environments and animation. SIGGRAPH Asia Courses, 2011.
- [5] William Steptoe. Ar-rift, 2014. <http://willsteptoe.com/post/66968953089/ar-rift-part-1>.
- [6] J. Fodor. *The mind doesn't work that way: The scope and limits of computational psychology*. Cambridge, MA: MIT Press, 2000.
- [7] R. Samuels. *Massively modular minds: Evolutionary psychology and cognitive architecture*. P. Carruthers and A. Chamberlain, eds., Evolution and the Human Mind. Cambridge: Cambridge University Press, 2000.
- [8] J. Prinz. *Is the Mind Really Modular?*. Stainton, Robert J. (Ed), (2006). Contemporary debates in cognitive science. Contemporary debates in philosophy. Malden: Blackwell Publishing, 2006.
- [9] G. Tononi and G. M. Edelman. Consciousness and complexity. *Science*, 282:1846–1851, 1998.
- [10] Lachaux J. P. Rodriguez E. Varela, F. and J. Martinerie. The brainweb: phase synchronization and large-scale integration. *Nature reviews. Neuroscience*, 2:229–239, 2001.
- [11] M. Le Van Quyen. The brainweb of cross-scale interactions. *New Ideas in Psychology*, 29:57–63, 2011.
- [12] Kim R. Shams L. Crossmodal influences on visual perception. *Physics of Life Reviews*, 2010.

- [13] Nijhawan R Shams L Kamitani Y Watanabe K. Shimojo S, Scheier C. Beyond perceptual modality: Auditory effects on visual perception. *Acoustical Science and Technology*, 22(2):61–67, 2001.
- [14] David W Hairston, Donald A Hodges, Jonathan H Burdette, and Mark T Wallace. Auditory enhancement of visual temporal order judgment. *NeuroReport*, 17(8):791–5, 2006.
- [15] MacDonald J. McGurk H. Hearing lips and seeing voices. *Nature*, 264:746–8, 1976.
- [16] Ladan Shams, Yukiyasu Kamitani, and Shinsuke Shimojo. Visual illusion induced by sound. *Cognitive brain research*, 14:147–152, 2002.
- [17] Shams L. Kim R, Peters M.A.K. 0 + 1 \neq 1: How adding noninformative sound improves performance on a visual task. *Psychological Science*, 23(1):6–12, 2012.
- [18] Barry E Stein, Nancy London, Lee K Wilkinson, and Donald D Price. Enhancement of perceived visual intensity by auditory stimuli: A psychophysical analysis. *Journal of cognitive Neuroscience*, 8(6):497–506, 1996.
- [19] Robert Sekuler, Allison B Sekuler, and Renee Lau. Sound alters visual motion perception. *Nature*, 385(6614):308, 1997.
- [20] LLC Oculus VR. Oculus documentation, 2016. <https://developer3.oculus.com/documentation/>.
- [21] Unity Technologies. Unity documentation, 2016. <https://docs.unity3d.com/Manual/index.html>.
- [22] Jason Gregory. *Game Engine Architecture*. A.K.Peters Ltd., 2009.
- [23] Unity Technologies. Render texture unity documentation, 2016. <https://docs.unity3d.com/Manual/class-RenderTexture.html>.
- [24] Duance C Brown. Decentering distortion of lenses. *Photometric Engineering*, 32(3):442–462, 1966.

Anexo A. Trabajo presentado en el Congreso Español de Informática Gráfica (CEIG)

En este anexo se va adjuntar una copia del trabajo sometido al Congreso Español de Informática Gráfica (CEIG) del año 2016, donde fue aceptado y galardonado como uno de los mejores artículos.

Crossmodal perception in immersive environments

Marcos Allue Ana Serrano Manuel G. Bedia Belen Masia

Universidad de Zaragoza

Abstract

With the proliferation of low-cost, consumer level, head-mounted displays (HMDs) such as Oculus VR or Sony's Morpheus, we are witnessing a reappearance of virtual reality. However, there are still important stumbling blocks that hinder the development of applications and reduce the visual quality of the results. Knowledge of human perception in virtual environments can help overcome these limitations. In this paper, within the much-studied area of perception in virtual environments, we chose to look into the less explored area of crossmodal perception, that is, the interaction of different senses when perceiving the environment. In particular, we looked at the influence of sound on visual motion perception in a virtual reality scenario. We first replicated a well-known crossmodal perception experiment, carried out on a conventional 2D display, and then extended it to a 3D head-mounted display (HMD). Next, we performed an additional experiment in which we increased the complexity of the stimuli of the previous experiment, to test whether the effects observed would hold in more realistic scenes. We found that the trend which was previously observed in 2D displays is maintained in HMDs, but with an observed reduction of the crossmodal effect. With more complex stimuli the trend holds, and the crossmodal effect is further reduced, possibly due to the presence of additional visual cues.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

1. Introduction and related work

With the proliferation of low-cost, consumer level, head-mounted displays (HMDs) such as Oculus VR or Sony's Morpheus, we are witnessing a reappearance of virtual reality. New applications are developed every day, going far beyond entertainment and gaming, and including advertising, virtual tourism, prototyping, medicine, scientific visualization, or education, to name a few. There are still important stumbling blocks that hinder the development of more applications and reduce the visual quality of the results; examples include limited spatial resolution, significant chromatic aberrations, tracking issues, limited processing capability leading to lag, subsequent motion sickness, or content generation. A relevant area which has received quite some interest but remains full of unanswered questions and open problems is how our perception is modified or altered when immersed in a virtual environment. Knowledge of human perception in virtual environments can help overcome the aforementioned current limitations; in the past, perception has helped tremendously in many computer graphics-related areas such as rendering [RFB07], material modeling and acquisition [SRD08], or display [MWDG13]; a good review of applied perception in graphics can be found in the course by McNamara and colleagues [MMG11].

In this paper, within the much-studied area of perception in vir-

tual environments, we chose to look into the less explored area of crossmodal perception, that is, the interaction of different senses when perceiving the environment. In particular, we looked at the *influence of sound on visual perception in a virtual reality scenario*.

Nowadays, a popular view in neuroscience holds that the human brain is structured into a large number of areas in which information is highly separated [Fod00]. This perspective assumes that mental processes such as perception -but also emotions and intentions- are limited to neural processes inside the brain and confined to particular areas. In the same way, it is often assumed that inputs coming from different perceptual modalities are processed in the brain independently and in different brain regions [Sam00].

However, the feeling of unified perceptions of objects and events is an ordinary experience. It suggests that information from different sensory modalities must somehow be bounded together in the brain in order to represent a single object or event [Pri06]. This assumption is cornerstone in most recent alternative neurodynamic views (as for example, bodily and sensorimotor approaches) in order to propose solid explanatory alternatives to traditional and internalist perspectives of brain organization [TE98, VM01]. In these alternative approaches, multisensory perception processes and different sensory modalities are understood as closely related through flexible integrations of the dynamics of brain by means of the emer-

gence of transient assemblies of neural synchronization when a unified perception arises [LVQ11]. Thus, a complete understanding of perception would require to know the different ways in which one sense modality is able to impact another, creating crossmodal illusions [SL10]. If we understood the interactions among perceptual modalities, we could shed light on the true mechanisms that support perceptual processes.

It is worth highlighting that, until very recently, the neural principles of multisensory integration and crossmodal illusions have remained unexplored. The modular view of the brain has been so strong with respect to the visual stimuli that it has been considered in the past as independent from other modalities. However, in recent years the interest in understanding crossmodal phenomena and illusions has increased substantially [Shi01]. Some of the deeper studies are those involved in alterations between auditory and visual senses. The best known example amongst these is the *ventriloquism effect* which refers to the perception of speech sounds as coming from a different direction than its real source, forced by the influence of visual stimuli from an apparent speaker [HHBW06]. Another well-known example is the *McGurk effect* [MH76] where lip movements of a subject are integrated with different but similar speech sounds.

The aim of the present paper is to investigate the effect of auditory spatial information on the perception of moving visual stimuli. We focus on the case of *motion perception* because previous studies have suggested that there should exist common neural substrates between the visual and auditory modalities [SL00]. The work is inspired in a classical experiment developed in the 90s where sound influenced ambiguous visual motion perception as proposed by [SSL97]. The authors found that when two objects -in a virtual and ambiguous simulation- moving along crossing trajectories reached the same position and then moved apart, they would be sometimes perceived by participants in the study as if they followed the same trajectory. However, in other cases, they reported that the objects reversed their direction as they would do following a collision. Sekuler et al. [SSL97] discovered that this ambiguity was solved when a sound emerged at the moment of coincidence of the objects, as this would show that the sensory information perceived in one modality (audition) could modulate the perception of events occurring in another modality (visual motion perception). Although the crossmodal effect reported by Sekuler and collaborators was accused of simply showing a cognitive limit rather than a genuine crossmodal perceptual effect, the authors opened the debate regarding the perceptual nature of many other crossmodal illusions between visual and auditory stimuli. For instance, the effect known as sound-induced flash illusion [SL00,SKS02] showed how the perception of a brief visual stimuli could be altered by concurrent brief sounds. When a single flash of light was showed together with two beeps, the perception changed from a single flash to two flashes. The reverse illusion could also occur when two flashes were accompanied by a single beep (which would be then perceived as a single flash).

These results revealed that unified and integrated perceptual constructs cannot be simply an assemblage of modality-specific components and that the traditional conception of perceptual experiences as an aggregate of ingredients (different type of sensorial

stimuli) is not accurate. No particular modality of sensorial perception can be characterized entirely in isolation from the others. We consider that these conclusions on how crossmodal illusions are constituted could contribute significantly to current progress in perceptual research. Here, we take steps towards further understanding this phenomenon by performing two experiments: First, we reproduced the experiment of Sekuler et al. [SSL97] on a conventional, 2D display; and conducted, the same experiment on a HMD, with the aim of discovering whether the same trends in crossmodal perception are observed in HMDs as the ones observed on conventional displays (Experiment 1). Next, in Experiment 2, we extended the original experiment by modifying the original stimuli, to check whether the effects observed by Sekuler et al. still hold in the presence of more complex stimuli. With the exception of the replication of Sekuler et al.'s experiment in a conventional display (first part of Experiment 1), it is the first time, to our knowledge, that these experiments are performed with a virtual reality scenario. We describe both experiments in Section 2 and analyze and discuss the results in Section 3; final conclusions are drawn in Section 4.

2. Experimental procedure

We have performed two experiments in order to determine how much an immersive environment interferes with the crossmodal interaction between the visual and auditory systems. Our experiment is based in the work of Sekuler et al. [SSL97], where they explore the perceptual consequences of sound altering visual motion perception. In their experiments, they showed two identical disks that moved steadily towards each other, coincided, and then continued in the same direction. This scenario is consistent with two different interpretations: either the two spheres did not collide and continued in their original directions (they *streamed*), or they collided and *bounced*, changing their traveling direction. The goal of the experiment is to analyze whether a sound at the moment of the impact can affect the interpretation of the scenario.

We built upon Sekuler's work, and extended his experiment to virtual reality, aiming to explore the consequences on crossmodal interactions of introducing the user inside a more realistic and complex environment presented with a *head mounted display* (HMD). In order to ensure that the replication is accurate, we performed both the original 2D experiment on a conventional display, and the experiment with a 3D environment on an HMD. So that our 3D environment reproduced the 2D one faithfully, we projected our 3D scene on the screen, and ensured that all the measurements of the original 2D scene are maintained. We show the transformation between the two spaces in Figure 1.

2.1. Experiment 1

We first reproduced the experiment described in Sekuler's work both in a regular screen and in a HMD (*Oculus Rift DK2*) to determine if introducing a sound at the moment of the collision between the two spheres would promote the perception of bouncing. We sought to analyze the influence of presenting the action on an HMD on perception.

Stimuli The visual stimuli were rendered with *Unity*. They consisted of two spheres with radius 0.5 degrees, placed over a white

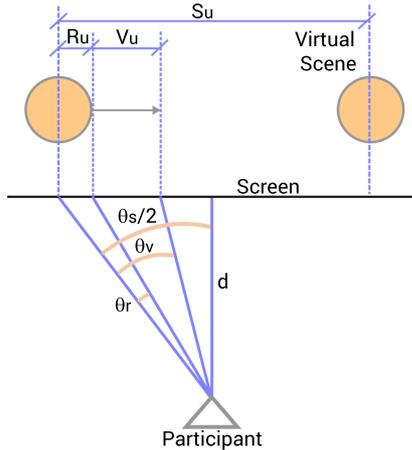


Figure 1: Projection of our 3D scene on the 2D screen plane for reproducing the experiment described in the work of Sekuler et al. [SSL97]. θ_r is the angle between the center and the radius of the sphere (0.5 degrees), θ_v represents the speed of sphere (6 degrees per second), θ_s is the angle between the centers of the spheres, and d is the distance to the screen. These angles are translated to distances in our 3D scene: R_u is the radius of the spheres, S_u is the distance between the spheres, and V_u is their speed when moving.

plane. The material of the spheres was brown and very diffuse to avoid introducing additional visual cues. The two spheres were initially separated by a distance of 4.2 degrees, and moved towards each other at a constant speed of 6 degrees per second. After they coincided, they continued moving without changing their original direction. We show in Figure 2 the initial layout of the scene. In this scenario we presented three different visual conditions: the spheres moved continuously, paused one frame at the point of their coincidence, or paused two frames at the point of their coincidence[†]. These three visual conditions were presented together with one of

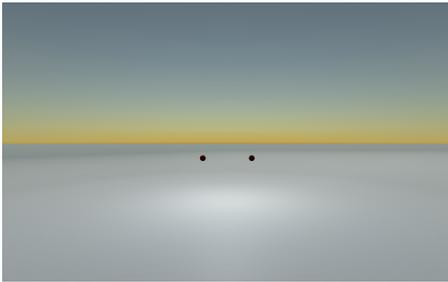


Figure 2: Initial layout of the scene for Experiment 1.

[†] The original experiment [SSL97] reported frames in a regular analog screen whose typical framerate is 25 frames per second. Since the framerate of our screen and the HMD (*Oculus Rift*) were very different, we adjusted the pause to last 1/25 seconds. Therefore, throughout the paper the terminology is as follows: one frame is equivalent to 1/25 seconds, and two frames are equivalent to 2/25 seconds.

the four following auditory conditions: no sound, accompanied by a brief click sound (frequency of 2000 Hz, duration of 3 milliseconds) triggered 150 milliseconds before or after the coincidence, or accompanied by a brief click sound at the point of coincidence.

Participants Thirteen participants took part in the experiment, three women and ten men, with ages ranging from 18 to 28 years. They all had normal or corrected-to-normal vision.

Procedure During the experiment we presented a total of twelve different conditions to each participant, three visual (continuous movement, pause one or two frames at the coincidence) and four auditory (no sound, sound at, before, or after the coincidence). Each of these conditions was presented ten times, making a total of 120 trials that appeared in a random order. We performed two blocks of the same experiment ordered randomly: one displayed on a regular screen (*Acer AL2216W TFT 22"*), and the other one displayed on an HMD (*Oculus Rift DK2*).

Before the HMD block, the lenses of the *Oculus Rift DK2* were adjusted to the participant eyes. We additionally introduced a training session before this block, where we showed two spheres at different depths (see Figure 3) and the participant had to choose which one was closer. We presented ten trials of the training with spheres at random depths. With this training the user gets used to the device, setup, and answering procedure.

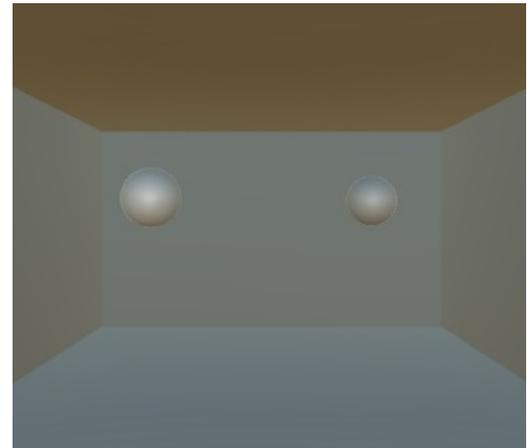


Figure 3: Training for the HMD experiment, in the scene the participants saw two spheres at different depths, and had to indicate which sphere was closer, clicking the corresponding (right or left) mouse button.

We guided the participants through the test by showing several slides with descriptions of each phase of the experiment. After each trial, a slide was displayed with the question "Did the spheres bounce or stream?", and a visual aid indicating the participant to answer with a mouse click (right or left). The supplementary material shows screenshots of the test, including explanation and questions.

Data processing We first processed the collected data by rejecting those users with stereo vision problems. In order to do this, we discarded a user if during the training the percentage of successful answers was equal or under 70%. We further processed the data by rejecting outliers. To do this, we first calculated for each participant and for each of the twelve conditions the percentage of *bouncing* answers over the ten trials. Then we used the first and third quartiles (Q_1 and Q_3), and the interquartile difference (Q_d) to find outliers for each condition [HI87]. We discarded a condition if it fulfilled any of the following inequalities:

$$\begin{aligned} condition &< (Q_1 - K_d * Q_d) \\ condition &> (Q_3 + K_d * Q_d) \end{aligned} \quad (1)$$

with $Q_d = Q_3 - Q_1$ and $K_d = 1.5$. Additionally, if a participant was marked as an outlier for more than one condition, all the answers of the participant were discarded.

2.2. Experiment 2

For the second experiment we sought to further analyze the effect of a more realistic environment in the crossmodal interaction between the visual and auditory systems. In order to do this, we increased the realism of the scene in three different ways (we term them three *blocks*) while keeping the proportions between distances and speed of the spheres of the original experiment.

Stimuli The visual stimuli were rendered once again with *Unity*. We designed a new scene where the spheres are placed on a white table, inside a furnished room, and with a more realistic illumination. With respect to the first experiment we also increased the size of the spheres to *1 degree* of radius, and the distance between them to *8.4 degrees*, to make them more visible. In order to keep proportions, we needed to increase the speed accordingly. This increase of speed is shown in Figure 4 and described by Equation 2.

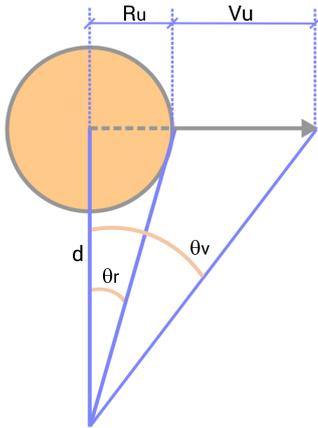


Figure 4: Scheme showing the increase of θ_v when increasing r . For a small θ_v , the increase of θ_v is linear when increasing r , as shown in Equation 2.

$$\begin{aligned} \tan(\theta_v) &= \frac{r + v}{d} \\ \theta_v \approx 0 &\implies \tan(\theta_v) \approx \theta_v \end{aligned} \quad (2)$$

For a small θ_v , the increase of θ_v is linear when increasing r . A screenshot of the initial layout of the scene for the first block of the experiment is shown in Figure 5. For the second block of the ex-



Figure 5: Initial layout of the scene with increased radius of the spheres (block 1) for Experiment 2.

periment, starting from the scene in the first block, we additionally introduced two more visual cues to the spheres. First, we increased the glossiness of the material of the spheres, and second, we slightly lifted the spheres over the table in order to have more visible shadows (see Figure 6). Finally, for the third block of the experiment,



Figure 6: Initial layout of the scene with increased radius of the spheres and additional visual cues (block 2) for Experiment 2.

starting from the scene in the first block, we also rotated the plane of the collision between the spheres. We show a screenshot of the initial layout for this block in Figure 7.



Figure 7: Initial layout of the scene with increased radius of the spheres and rotated plane of the collision (block 3) for Experiment 2.

Participants Twenty seven participants took part in the experiment, two women and twenty five men with ages ranging from 19 to 32 years. They all had normal or corrected-to-normal vision.

Procedure During the experiment we presented a total of six different conditions, two visual (continuous movement, pause two frames at the coincidence), and three auditory (no sound, *click sound* at, or after the coincidence). Based on the results of the first experiment we removed the visual condition with a pause of one frame because the percentage of bouncing perceived was similar to the one perceived with the pause of two frames, and the auditory condition corresponding to the sound before the coincidence, also because of its similarity with the sound after the coincidence. Each of these conditions was presented *ten* times, making a total of 60 trials that appeared in a random order. All the blocks of the experiment were presented in the *HMD*, and each participant performed three randomly ordered blocks that corresponded to the three scenes described in the *Stimuli* section, totalling 180 trials per subject. Before starting the test, the participants performed the same training described in Experiment 1.

Finally, in this experiment the slides with instructions about the test were shown on a frame on the back of the room striving to preserve as much as possible the realism of the environment.

Data processing We followed the same methodology as in Experiment 1 for rejecting outliers.

3. Results and analysis

In this section we analyze the results collected in our experiments, in particular we calculate for every user and condition the percentage of times subjects observed the spheres *bouncing* in the *ten* trials. We use repeated measures ANOVA to test whether each of the conditions have influence in the observed percentage of bounce responses. We need the repeated measures scheme because we measure the same independent variables (e.g., frames paused) under different conditions performed by the same subjects. We fix a significance value (p-value) of 0.05 in all the tests, and in those cases in which results from Mauchly’s test of sphericity indicate that variances and covariances are not uniform, we use adjusted measures (Greenhouse Geisser correction [CW11]). Previous to the analysis, we perform outlier rejection as detailed in Section 2.

3.1. Experiment 1

The goal of this experiment was to test whether the effect of sound altering visual motion perception as reported in the experiments carried out by Sekuler et al. [SSL97] is also observed when reproduced in a virtual environment with an HMD. We wanted to test three factors: (i) the overall influence of the display (2D scene presented on a *screen*, or 3D environment presented on an *HMD*); (ii) the influence of the *sound* when the spheres collide; and (iii) the influence of the length of the *pause* at the point of coincidence between the spheres. We aggregate the percentages of all trials for every condition and perform a repeated measures ANOVA; results are presented in Table 1. We can conclude that all three factors have a significant effect in the percentage of bounce responses, since

Table 1: Results (F-test and significance) of the analysis of the data with repeated measures ANOVA for Experiment 1. We test the influence of three factors in the perceived percentages of bounce responses.

	F	Sig.
Sound vs percent. bounce	83.664	0.000
Pause vs percent. bounce	63.528	0.000
Display vs percent. bounce	13.176	0.000

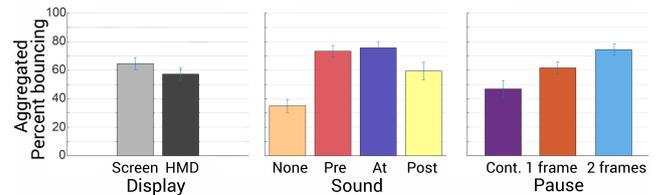


Figure 8: Aggregated percentages of bounce responses and corresponding error bars (standard error of the mean) for the Experiment 1. From left to right: Percentages for two display conditions (screen or HMD), percentages for four auditory conditions (no sound, sound at, before, or after the moment of coincidence of the spheres), and percentages for three visual conditions (continuous movement, pause one, or two frames at the point of coincidence of the spheres).

all the p-values are below 0.05. We show in Figure 8 the mean percentages of bounce responses for the tested factors (error bars represent the standard error of the mean). We observe that the percentage of bounce responses decreases when using the HMD display. However, the main findings of Sekuler’s work hold: a sound at the moment of coincidence, and a pause of two frames at the point of coincidence promote the perception of bouncing. We believe that the decrease in perceived bouncing in the tests with the HMD comes from the increase in the amount of visual cues due to the stereoscopic view. We additionally show in Figure 9 simple (non-aggregated) mean percentages of bouncing and error bars for each condition. Sound promotes perception of bouncing when compared with the absence of sound; however, it has significantly less effect when reproduced after the point of coincidence. Still, there is a high tolerance for asynchrony between the sound and the visual input: even when the sound is delayed, the percentage of bounce responses increases. Also, as reported previously by Sekuler and others [SSL97, BBB93, SSB95], the overall percentage of bounce responses increases with the duration of the pause.

3.2. Experiment 2

The goal of this experiment was to test whether a more complex scene could influence the effect of sound altering visual motion perception. Again, we wanted to test three factors: the influence of each of the three scenes (three blocks) described in Section 2, the influence of the *sound* when the spheres collide, and the influence of the *pause* at the point of coincidence between the spheres. We aggregate the percentages for every condition and perform a repeated

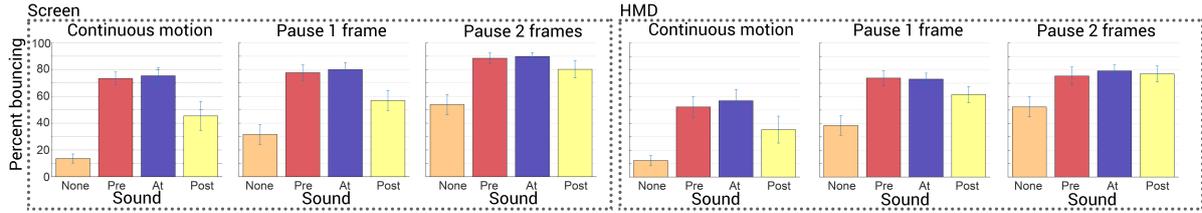


Figure 9: Percentages (non-aggregated) and error bars (standard error of the mean) for Experiment 1. Left: Results for display on a 2D screen. Right: Results for display on an HMD. In both cases, from left to right: Plots for each of the three conditions of the pause factor (continuous movement, pause one, or two frames at the point of coincidence of the spheres). Each of the three plots shows the percentage of bounce responses for each of the four auditory conditions (no sound, sound at, before, or after the moment of coincidence of the spheres).

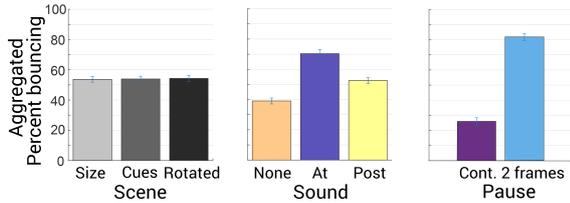


Figure 10: Aggregated percentages and error bars (standard error of the mean) for the Experiment 2. From left to right: Percentages for the three different scenes or blocks (increase in the size of the spheres, additional visual cues in the spheres, or rotated plane of the movement); percentages for three auditory conditions (no sound, sound at, or after the moment of coincidence of the spheres); and percentages for two visual conditions (continuous movement, or pause two frames at the point of coincidence of the spheres).

measures ANOVA; results are presented in Table 2. In Figure 10 we show the mean percentages of bounce responses for the tested factors, and the associated error bars representing the standard error of the mean. The analysis with the ANOVA reveals that, as be-

Table 2: Results (F-test and significance) of the analysis of the data with repeated measures ANOVA for Experiment 2. We test the influence of three factors in the perceived percentages of bounces.

	F	Sig.
Sound vs percent. bounce	124.137	0.000
Pause vs percent. bounce	845.386	0.000
Scene vs percent. bounce	0.220	0.977

fore, there is a significant effect of the *sound*, and the *pause* in the perceived percentage of bounces. However, the p-value for the test with different scenes is very high, therefore we cannot draw any significant conclusion about the relationship between the three different scenes and the observed percentage of bouncing. When comparing Experiments 1 and 2 we can see that even when increasing the level of realism of the scene, the crossmodal effect of the sound altering the perceived motion still holds, although there is a general shift downwards of the percentage of bounce responses which can be observed by comparing the corresponding percentages of Figures 8 and 10. This shift downwards is possibly due to the presence

of additional cues; however the high p-value of the scene factor, further indicates that there is no significant difference on the effect on crossmodal interaction between the three scenes (blocks) tested (i.e., no cue has proven to be significantly stronger or weaker in the detection of bouncing).

4. Discussion and conclusions

In this paper, we have performed an exploration of crossmodal perception in virtual reality scenarios, in particular using an HMD. We have studied the influence of auditory signals in the perception of visual motion. To do so, we first replicated an existing experiment which demonstrated the existence of a crossmodal interaction between both senses with simple stimuli on a 2D conventional display. We were able to successfully replicate it, obtaining the same trends in the results, and then extended it to virtual reality with a HMD. We found that the same trends hold on an HMD (i.e., the factors explored had the same influence on the crossmodal effect), but that there is a reduction in the crossmodal effect. This reduction essentially means that there is a shift in the results towards a better accuracy of subjects in performing the tasks assigned in the HMD setup. This can be due to the presence of additional cues, in particular depth cues including binocular disparity and possibly motion parallax. A similar conclusion can be drawn in our second experiment: We repeated the first experiment (only on the HMD), with new subjects, and with more complex stimuli (we had three different variations of the initial stimulus) to see whether the effect would still hold with more realistic scenery. We once again observed a reduction of the crossmodal effect (subjects were better at detecting the correct behavior of the stimuli), which we hypothesize is due to the presence of additional cues, in this case pictorial cues (shading, perspective, texture).

Overall, we believe these are just a few steps in the exploration of crossmodal perception in virtual reality. In the future, we would like to expand these experiments by including other potentially influencing factors or effects, and by further increasing the complexity of the stimuli, which is required for the conclusions to be usable in a real virtual reality application. Additionally, further analysis of the first-order interactions of the factors studied is required.

5. Acknowledgments

The authors would like to thank Diego Gutierrez for fruitful insights and discussion. Ana Serrano was supported by an FPI grant from the Spanish Ministry of Economy and Competitiveness (project Lightslice).

References

- [BBB93] BERTENTHAL B. I., BANTON T., BRADBURY A.: Directional bias in the perception of translating patterns. *Perception* 22, 2 (1993), 193–207. doi:10.1068/p220193. 5
- [CW11] CUNNINGHAM D., WALLRAVEN C.: *Experimental Design: From User Studies to Psychophysics*, 1st ed. A. K. Peters, Ltd., Natick, MA, USA, 2011. 5
- [Fod00] FODOR J.: *The mind doesn't work that way: The scope and limits of computational psychology*. Cambridge, MA: MIT Press, 2000. 1
- [HHBW06] HAIRSTON D. W., HODGES D. A., BURDETTE J. H., WALLACE M. T.: Auditory enhancement of visual temporal order judgment. *NeuroReport* 17, 8 (2006), 791–5. 2
- [HI87] HOAGLIN D. C., IGLEWICZ B.: Fine-tuning some resistant rules for outlier labeling. *Journal of the American Statistical Association* 82, 400 (1987), 1147–1149. 4
- [LVQ11] LE VAN QUYEN M.: The brainweb of cross-scale interactions. *New Ideas in Psychology* 29 (2011), 57–63. 2
- [MH76] MCGURK H M. J.: Hearing lips and seeing voices. *Nature* 264 (1976), 746–8. 2
- [MMG11] MCNAMARA A., MANIA K., GUTIERREZ D.: Perception in graphics, visualization, virtual environments and animation. SIGGRAPH Asia Courses, 2011. 1
- [MWDG13] MASIA B., WETZSTEIN G., DIDYK P., GUTIERREZ D.: A Survey on Computational Displays: Pushing the Boundaries of Optics, Computation, and Perception. *Computers & Graphics* 37, 8 (2013), 1012 – 1038. 1
- [Pri06] PRINZ J.: *Is the Mind Really Modular?*. Stainton, Robert J. (Ed), (2006). Contemporary debates in cognitive science. Contemporary debates in philosophy. Malden: Blackwell Publishing, 2006. 1
- [RFWB07] RAMANARAYANAN G., FERWERDA J., WALTER B., BALAK.: Visual equivalence: Towards a new standard for image fidelity. *ACM Trans. Graph.* 26, 3 (July 2007). 1
- [Sam00] SAMUELS R.: *Massively modular minds: Evolutionary psychology and cognitive architecture*. P. Carruthers and A. Chamberlain, eds., Evolution and the Human Mind. Cambridge: Cambridge University Press, 2000. 1
- [Shi01] SHIMOJO S. S. C. N. R. S. L. K. Y. . W. K.: Beyond perceptual modality: Auditory effects on visual perception. *Acoustical Science and Technology* 22, 2 (2001), 61–67. 2
- [SKS02] SHAMS L., KAMITANI Y., SHIMOJO S.: Visual illusion induced by sound. *Cognitive brain research* 14 (2002), 147–152. 2
- [SL00] SHAMS L KAMITANI Y S. S.: What you see is what you hear. *Nature* 408 (2000), 788. 2
- [SL10] SHAMS L K. R.: Crossmodal influences on visual perception. *Physics of Life Reviews* (2010). 2
- [SRD08] SILLION F. X., RUSHMEIER H., DORSEY J.: *Digital Modeling of Material Appearance*. Morgan Kaufmann/Elsevier, 2008. 1
- [SSB95] SEKULER R., SEKULER A., BRACKETT T.: When visual objects collide: Repulsion and streaming. *Investigative Ophthalmology and Visual Science* 36, 50 (1995). 5
- [SSL97] SEKULER R., SEKULER A. B., LAU R.: Sound alters visual motion perception. *Nature* 385, 6614 (1997), 308. 2, 3, 5
- [TE98] TONONI G., EDELMAN G. M.: Consciousness and complexity. *Science* 282 (1998), 1846–1851. 1

- [VM01] VARELA F. L. J. P. R. E., MARTINERIE J.: The brainweb: phase synchronization and large-scale integration. *Nature reviews. Neuroscience* 2 (2001), 229–239. 1