Universidad Zaragoza
1542

Trabajo Fin de Grado

Edición cinematográfica y continuidad narrativa en realidad virtual

Movie editing and narrative continuity in virtual reality

Autor/es

Jaime Ruiz-Borau Vizárraga

Directora: Ana Serrano Pacheu

Ponente: Diego Gutiérrez Pérez

Escuela de Ingeniería y Arquitectura
2017

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

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).

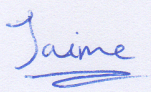
D./Dª. Jaime Ruiz-Borau Vizárraga,

con nº de DNI 77218350-J 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) Edición cinematográfica y continuidad narrativa en realidad virtual,

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

Zaragoza, 20 de abril de 2017



Fdo: Jaime Ruiz-Borau Vizárraga

Resumen

Con el reciente auge de la realidad virtual, en los últimos años se están creando experiencias de todo tipo para este nuevo formato, con el fin de exprimir al máximo este nuevo entorno virtual. La posibilidad de ubicar al espectador directamente en el mundo que el creador ha hecho para él resulta especialmente atractiva, tanto a nivel de juegos y aplicaciones como de experiencias narrativas.

Es en estas *experiencias narrrativas* donde los creadores de contenidos, en una primera aproximación, tratan de trasladar lo que ya saben de la cinematografía tradicional a la realidad virtual, pero por las características inherentes de ésta, no todas las técnicas son igualmente aplicables y no tienen la misma efectividad.

En este proyecto se llevan a cabo una serie de experimentos empíricos sobre la atención de los usuarios en narrativa en realidad virtual, con el fin de comprobar la efectividad de las técnicas de cinematografía tradicional en un entorno inmersivo. El objetivo es el de proporcionar unas pequeñas guías sobre lo que se puede hacer y lo que no se debe hacer cuando se habla de narrativa en realidad virtual. En primer lugar, se investiga si en un entorno virtual los usuarios tienen una percepción de continuidad similar a la de la narrativa tradicional, para lo cual este proyecto se apoya en varios estudios sobre teorías acerca de la cognición humana y la continuidad. Después, se procede a analizar bajo diferentes circunstancias cómo se ve influenciada la atención de los usuarios en la narrativa de un entorno inmersivo.

Este trabajo contiene: un estudio del estado del arte en cinematografía en realidad virtual, un estudio de la teoría cognitiva de segmentación de eventos aplicada a un entorno de realidad virtual, implementación de un sistema completo para llevar a cabo estudios de usuario con distintos estímulos, diseño de los experimentos a llevar a cabo, creación y generación de los diferentes estímulos a utilizar en los experimentos con usuarios y análisis del comportamiento de los usuarios.

Abstract

Along with the recent rise of virtual reality, in the last few years experiences of all kinds are being created for this new format, in order to squeeze the most of this new virtual environment. The possibility of placing an user directly into the world that the creator has made for him is especially attractive, both at the level of games and applications and also narrative experiences.

It is in these *narrative experiences* where the content creators, in a first attempt, try to apply what they already know about traditional cinematography to virtual reality, but due to the inherent features of this virtual environment, not all the techniques are equally applicable and therefore have not the same effectiveness.

In this project we empirically test the users' attention in narrative virtual reality, in order to check the effectiveness of the traditional cinematography techniques in a virtual environment. Our final goal is to provide some small guidelines of what you can do and what you should not do when we talk about narrative in virtual reality. First, we investigate if in a virtual environment users have a perception of continuity similar to the one perceived in traditional narrative. To do so, this work takes into account several studies about human cognitive theories and continuity. After this, we proceed to analyze how users' behaviour is affected under different circumstances in virtual reality's narrative.

This work contains: a study of the state of the art in cinematography in virtual reality, a study of the cognitive theory of event segmentation applied to a virtual reality environment, implementation of a complete system to carry out user studies with different stimuli, the design of the experiments to be carried out, creation and generation of the different stimuli to be used in the experiments with users, and the analysis of users' behavior.

Agradecimientos

Me gustaría dar las gracias especialmente a mi directora de trabajo fin de grado, Ana Serrano, por la inestimable ayuda que me ha brindado durante todo el desarrollo del proyecto; así como a Belén Masia y Diego Gutiérrez por su ayuda en el proyecto, por la oportunidad de trabajar en su grupo y por introducirme y enseñarme el mundo de la investigación. También quiero agradecer a todas las personas del Graphics and Imaging Lab por su participación en los experimentos y por acogerme como a uno más.

También quiero dar las gracias a Abaco digital por su ayuda a la hora de grabar los vídeos que forman parte de los estímulos utilizados en los experimentos, así como a Sandra Malpica y Victor Arellano por su excelente actuación en dichos vídeos; y a Paz Hernando y Marta Ortín por su ayuda con los experimentos y los análisis.

A nivel personal, me gustaría agradecer a todos los compañeros y amigos que me han apoyado a lo largo de la carrera, pero en especial a Sara Cerón, David Vergara, Alejandro Royo, Alejandro Márquez y Patricia Lázaro, ya que sin su ayuda y apoyo no hubiese llegado hasta aquí. También me gustaría agradecer a mi familia, en especial a mis padres y a mi hermano, por su apoyo incondicional.

Por último, también me gustaría agradecer a todos los profesores que me han formado a lo largo de la carrera.

Índice

Índice de figuras

1. Introducción

1.1. Objetivos

La cinematografía tradicional se basa en un conjunto de reglas y técnicas que se han ido utilizando y perfeccionando a lo largo de más de cien años. Este conjunto de reglas y técnicas conforman lo que se denomina *edición en continuidad* (del inglés *continuity editing*), y lo que consiguen es dar una sensación de coherencia a un flujo de eventos discontinuos tanto en espacio como en tiempo, de tal forma que el espectador que visualiza dicho conjunto de eventos consigue percibirlos con continuidad a pesar de los enormes cambios visuales que puede haber entre ellos.

Con el resurgimiento reciente de la realidad virtual, creadores de contenidos de todo el mundo se han lanzado a generar experiencias de todo tipo para este nuevo formato, incluidos contenidos cinematográficos, los cuales reciben el nombre de vídeos 360º (por su característica inherente de proyectar el vídeo alrededor del punto de visualización, 360º alrededor de él). Sin embargo, cuando se traslada al campo de la realidad virtual la edición en continuidad de la cinematografía tradicional, muchas de las técnicas que se emplean no funcionan de la misma forma, ya que se basan en el uso y modificación de la cámara (su posición, orientación, zoom, etc), lo cual en realidad virtual no es posible, puesto que es el propio espectador el que controla qué y hacia dónde está mirando.

El objetivo de este proyecto es el de comprobar empíricamente bajo qué circunstancias es posible aplicar técnicas de edición en continuidad tradicionales en el nuevo formato de la realidad virtual, así como comprobar si la sensación de continuidad que se daba en cinematografía tradicional se conserva también en realidad virtual.

Para ello el proyecto se apoya en la teoría cognitiva de segmentación de eventos (Kurby and Zacks [1]; Reynolds et al. [2]; Zacks and Swallow [3]), la cual expone que el cerebro humano segmenta las acciones continuas que percibe en una serie de eventos discretos representativos, de tal forma que es capaz de elaborar pequeñas predicciones sobre lo que puede ocurrir a continuación. Esta teoría según otros estudios recientes (Zacks et al. [4]) está estrechamente relacionada con la continuidad percibida en la cinematografía tradicional, por esto el primer experimento desarrollado trata de comprobar que las bases de la teoría cognitiva de segmentación de eventos también se cumplen en realidad virtual.

Por otro lado, se han estudiado y escogido unos tipos concretos de cortes de la edición en continuidad tradicional, así como un conjunto de parámetros circunstanciales relativos a las tomas de vídeo que conforman los experimentos. Estos parámetros circunstanciales se apoyan

en un concepto existente en los vídeos 360º: las *regiones de interés* o ROIs por sus siglas en inglés (Regions Of Interest). En el contexto de la realidad virtual como entorno inmersivo en el que un usuario es capaz de mirar a donde quiera y cuando quiera, estas ROIs representan la o las regiones de ese entorno inmersivo donde es más probable que un usuario esté mirando en un momento determinado. Normalmente en una película tradicional es el propio director el que elige qué entra dentro del encuadre de la cámara y el espectador ve lo que el director quiere que vea; sin embargo, en realidad virtual el director tiene que convencer al usuario de que mire a donde él quiere que mire: un coche, dos personas, una casa, etc. Todas esas cosas que el director quiere que el espectador vea y que tiene que convencerlo para que esté atento a ellas son ROIs.

Así pues, los parámetros circunstanciales escogidos para el estudio tienen que ver directamente con estas ROIs, concretamente, con el número de ROIs antes y después del corte y con la alineación entre ellas antes y después del corte (la alineación viene definida por la diferencia de distancia horizontalmente entre las ROIs principales antes y después del corte). Otro parámetro circunstancial escogido es el tipo de corte cinematográfico empleado.

Este proyecto se enmarca dentro de un proyecto de investigación, el cual ha sido desarrollado en el *Graphics and Imaging Lab* de la Universidad de Zaragoza y en colaboración con la Universidad de Stanford. Dicho proyecto de investigación ha sido recientemente aceptado en la conferencia *ACM SIGGRAPH 2017*, que publica en la revista *ACM Transactions on Graphics* (Q1, 1/106). La versión sometida a SIGGRAPH puede consultarse en el anexo A.

Durante este trabajo fin de grado se ha trabajado activamente en las siguientes fases del proyecto de investigación:

- Estudio del estado del arte en realidad virtual y sus características.
- Estudio del estado del arte en la cinematografía en realidad virtual.
- Diseño de un experimento para validar las bases de la teoría cognitiva de segmentación de eventos en realidad virtual
- Diseño de un experimento para comprobar el comportamiento de los usuarios ante distintos tipos de cortes y ROIs en vídeos en realidad virtual.
- Grabación y montaje de los vídeos 360º para proyectar durante los experimentos
- Implementación de un entorno de realidad virtual capaz de proyectar los vídeos 360º y recopilar la información de los usuarios para su posterior análisis.

Adicionalmente, se incluye la parte final de análisis por completitud, y para describir adecuadamente las conlusiones obtenidas en el trabajo.

1.2. Organización y planificación del proyecto

La memoria parte del planteamiento del problema de la cinematografía en realidad virtual; en el Capítulo 2 se muestra el contexto de los conceptos de la realidad virtual, la segmentación

de eventos y la cinematografía, y la implicación del uso de la realidad virtual en ésta última. En el Capítulo 3 se describen en detalle el diseño y desarrollo de los dos experimentos que se llevaron a cabo así como los objetivos que se buscan con ellos. En el Capítulo 4 se describe el entorno de trabajo utilizado para el desarrollo de los experimentos, todos los aspectos relacionados con el software y el hardware utilizados así como sus interrelaciones. En el Capítulo 5 se detalla cómo se han analizado los datos, en el Capítulo 6 se exponen los resultados y en el último capítulo se detallan las conclusiones obtenidas de dichos resultados.

A lo largo de la duración de este proyecto, desde septiembre de 2016 hasta enero de 2017 se han realizado múltiples tareas, entre las que se destacan: investigación sobre realidad virtual en general e investigación sobre cinematografía aplicada a realidad virtual, estudio de los parámetros a considerar en los experimentos, implementación de los experimentos y análisis de los resultados. En la Figura 1.1 se puede ver el diagrama de Gantt del trabajo realizado a lo largo del proyecto.

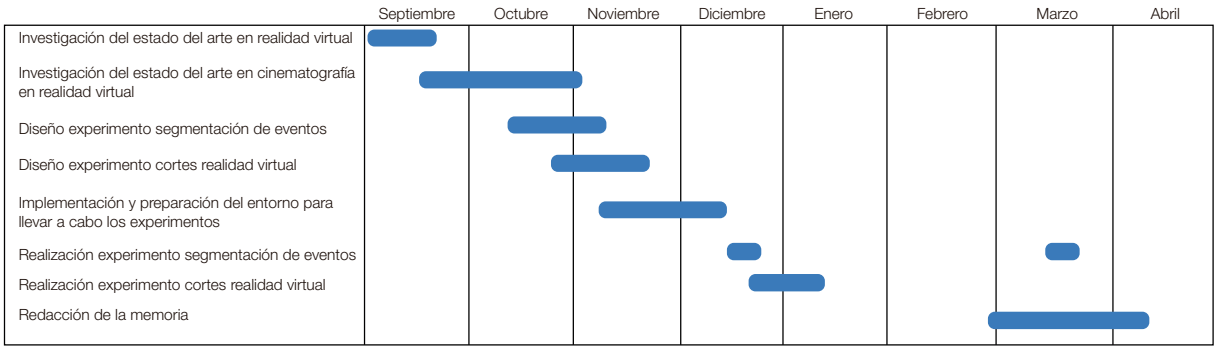


Figura 1.1: Diagrama de Gantt de las tareas realizadas en el proyecto

2. Contexto y trabajo relacionado

En este Capítulo se detallan los conceptos básicos de la realidad virtual, así como los fundamentos de la teoría de segmentación de eventos en la que se apoya este proyecto y las implicaciones que tiene de cara a la cinematografía tradicional y aplicada a la realidad virtual.

2.1. Realidad Virtual

En los últimos años, el concepto de realidad virtual ha ido ganando cada vez más y más presencia en el mundo de la informática. La realidad virtual hace alusión a entornos generados por ordenador donde el usuario tiene la sensación de estar plenamente inmerso, de tener la capacidad de percibirlo e incluso de interactuar con él desde dentro de ese entorno. Este resurgimiento reciente de la realidad virtual tiene su origen en el considerable aumento de la potencia del hardware a la par que se ha reducido su tamaño, lo cual ha hecho posible fabricar dispositivos capaces de presentar entornos virtuales con una calidad suficiente como para proporcionar una sensación de inmersión aceptable.

La mayor parte de los dispositivos actuales creados para la realidad virtual son dispositivos de tipo HMD (*Head Mounted Display*), aunque también existen adaptadores de distintos tipos que permiten el uso de un teléfono móvil como un visualizador de contenidos para realidad virtual. Todos ellos se caracterizan por tener más o menos una forma de casco: se colocan en la cabeza y envuelven de una u otra forma la parte frontal de la misma, concretamente la zona de los ojos, y delante de ella ubican una pantalla típicamente dividida en dos partes (una para cada ojo), donde se muestra el entorno inmersivo.

Envolviendo la parte frontal de la cabeza y limitando la visión de los ojos a lo que muestra la pantalla se consigue el efecto deseado: se deja de percibir visualmente el entorno real para pasar a percibir únicamente el entorno virtual, dando una sensación de inmersión en ese entorno generado por ordenador.

En cuanto a trabajo reciente relacionado sobre realidad virtual, existen unos cuantos que dan unos primeros pasos que analizan la trayectoria de la mirada y la saliencia en realidad virtual. El trabajo de Van et al. [5] utiliza filtrado predictivo para una predicción en tiempo real de de las trayectorias de la mriada, con el fin de reducir la latencia y las imprecisiones en el seguimiento en realidad virtual. Sin embargo, su trabajo está pensado para predicciones a corto plazo y no tiene en cuenta lo que está viendo el usuario, por lo que no se puede usar para analizar la trayectoria de la mirada en función de lo que se le está proyectando al usuario. En el trabajo de Kollenberg

et al. [6] se propone un análisis muy básico de la interacción entre la cabeza y la trayectoria de la mirada en realidad virtual. También en otros trabajos recientes se proponen herramientas para visualización de trayectoria de la mirada en tiempo real en escenarios 3D [7], o de trayectoria de la mirada en escenas 360º [8].

2.2. Cinematografía y segmentación de eventos

Como ya se ha dicho anteriormente, este proyecto se apoya en la teoría cognitiva de segmentación de eventos (Kurby and Zacks [1]; Reynolds et al. [2]; Zacks and Swallow [3]). Esta teoría postula que los seres humanos no almacenamos en nuestra memoria a corto plazo la totalidad del flujo continuo de acciones y eventos que percibimos en el día a día, sino que en su lugar, almacenamos series de eventos más pequeños representativos, a partir de los cuales creamos predicciones sobre lo que ocurrirá a continuación. Cuando esas predicciones se cumplen, la sensación que producen es de continuidad: el evento actual que se ha registrado sigue su curso. Cuando estas predicciones se violan, quiere decir que algo nuevo e inesperado ha ocurrido, luego es necesario registrar un nuevo evento.

Apoyándose en esta teoría, otros estudios recientes (Zacks et al. [4]) trataron de averiguar si la misma segmentación de eventos que se produce en el día a día se mantiene cuando un sujeto visualiza un corto cinematográfico o un largometraje. La motivación de estos estudios residía en la diferencia entre la percepción del mundo real y la de un metraje cinematográfico, puesto que en cualquier vídeo existen discontinuidades de todos los tipos, a las que llamamos cortes cinematográficos. Estas discontinuidades cinematográficas vienen a ser equivalentes a lo que ocurre en la vida real cuando a un sujeto le ocurre algo inesperado: siguiendo la teoría de segmentación de eventos, las predicciones sobre lo que ocurrirá a continuación no se corresponden con lo que ha ocurrido y eso marca el final del evento registrado y el inicio de uno nuevo.

Lo que se descubrió en estos estudios es que, tal y como se esperaba, algunas regiones del córtex cerebral mostraban cambios sustanciales cuando percibían discontinuidades en los metrajes, de la misma forma que ocurría con percepciones de la vida real; y más interesante, que ese proceso de elaborar predicciones es consistente con las prácticas habituales de los editores de cine profesionales, quienes utilizan los cortes para apoyar o romper las predicciones de los espectadores, o lo que es lo mismo, para favorecer o deshacer la sensación de continuidad percibida.

Cuando un corte introduce un cambio sustancial en el contenido visual, el cerebro no trata de explicar la discontinuidad percibida, sino que se adapta al cambio, crea una nueva representación mental y comienza a rellenarlo con detalles. Este mecanismo automático podría ser un proceso clave que explicase por qué la edición en continuidad funciona en la cinematografía tradicional.

2.3. Implicaciones del uso de realidad virtual en cinematografía

La cinematografía tradicional es diferente a la cinematografía en realidad virtual de la misma forma que la cinematografía tradicional es diferente a una obra de teatro. Todos ellos (teatro, cinematografía y realidad virtual) son formatos para un mismo objetivo: contar una historia.

De la misma forma que el teatro tiene sus elementos particulares, como pueden ser el escenario, las butacas para los espectadores, el telón que define el comienzo y el final de los actos... la cinematografía tradicional se ha valido de sus elementos particulares para transmitir una narrativa a su manera. Una de las técnicas concretas y más básicas de la cinematografía tradicional es la de los cortes: cambios totales o parciales en el contenido visual presentado a los espectadores a través de la pantalla. En una película tradicional el director tiene la capacidad de decidir qué imagen mostrar a los espectadores antes y después del corte. Puede cambiar la cámara de posición, hacer zoom en una parte concreta de la escena, rotar la cámara, etc. Y una de las características más importantes de este formato es que, ocurra el cambio que ocurra, el director sabe que el espectador siempre notará el cambio y verá lo que él (el director) quiere que vea, puesto que solo puede mirar a la pantalla y observar la imagen que le es presentada.

Sin embargo, en el formato de la realidad virtual esta característica del cine tradicional no se aplica. En un entorno inmersivo el espectador tiene total libertad de observar el mundo que le rodea, el director no tiene poder alguno sobre la decisión del espectador de la zona del metraje a la que el usuario está mirando. No solo eso, sino que el espectador forma parte del mundo que está visualizando, lo cual implica que los cambios en la posición de la cámara no son tan sencillos como lo eran en la cinematografía tradicional. Cambiar de posición a la cámara implica mover al usuario de la posición en la que estaba y colocarlo en otro sitio. Un corte cinematográfico en la realidad virtual implica cambiar todo el mundo alrededor del espectador, de forma más o menos abrupta, en contraposición con la cinematografía tradicional, donde un corte solo representa un cambio en la imagen. Por último, algunas técnicas de cámara de la cinematografía tradicional no tienen sentido en realidad virtual, como por ejemplo el zoom.

Todas estas características inherentes de la realidad virtual hacen que sea difícil el crear experiencias narrativas en este nuevo formato, ya que no se pueden trasladar directamente todas las técnicas de la cinematografía tradicional.

Por ello, este proyecto tiene como objetivo llevar a cabo experimentos empíricos para determinar la efectividad y bajo qué circunstancias se pueden aplicar las metodologías de la cinematografía tradicional en realidad virtual.

3. Desarrollo de los experimentos

En este proyecto se plantean dos experimentos: uno para validar las bases de la teoría cognitiva de segmentación de eventos en un entorno de realidad virtual; y otro para comprobar si se mantiene la sensación de continuidad cinematográfica en realidad virtual con distintos tipos de cortes cinematográficos y circunstancias de las tomas. En esta sección se describe en detalle el desarrollo de estos experimentos.

3.1. Experimento 1: Segmentación de eventos en realidad virtual

3.1.1. Objetivo

En este primer experimento lo que se busca es comprobar si en un entorno inmersivo los sujetos segmentan los eventos del mismo modo que en cinematografía tradicional, según Magliano y Zacks [9]. De esta forma, se puede asegurar que la percepción de continuidad narrativa en realidad virtual es similar a la percepción de continuidad en la cinematografía tradicional.

3.1.2. Desarrollo del experimento

El planteamiento de este experimento consiste en replicar el mismo experimento que realizan Magliano y Zacks en su trabajo, solo que extrapolado a un entorno inmersivo. Este experimento consiste en que un conjunto de usuarios visualicen en realidad virtual un total de 4 vídeos 360º de duración suficiente y con los 3 tipos distintos de cortes cinematográficos que se mencionan en dicho estudio. Adicionalmente durante la visualización de los vídeos los usuarios deben pulsar un botón para señalar cuándo piensan ellos que el evento actual ha terminado y ha comenzado un nuevo evento. Cabe mencionar que los participantes deben realizar este experimento dos veces, una para registrar eventos más *amplios* y otra para registrar eventos más *finos*. La definición de evento que se les proporciona es la siguiente:

- Amplios - *La mayor unidad de acción que tenga sentido para el usuario*
- Finos - *La menor unidad de acción que tenga sentido para el usuario*

La definición de evento fino o amplio es intencionadamente ambigua para evitar sesgar al usuario. Un ejemplo de evento amplio podría ser una conversación, y de esa misma conversación

eventos finos pueden ser: el personaje A habla, el personaje B le contesta, el personaje A se levanta, el personaje A habla de nuevo, etc.

Los estímulos escogidos para este experimento son: un cortometraje 360º de Star Wars[1] de 8 minutos de duración, otro cortometraje 360º titulado Always[2] de 6 minutos de duración, y dos capítulos de una serie de YouTube en 360º de 6 minutos de duración[3] y 5 minutos de duración[4], respectivamente.

El desarrollo de este experimento sigue la misma dinámica que el empleado en Magliano y Zacks [9]. En primer lugar, los participantes del experimento deben realizar un entrenamiento para comprobar que han comprendido suficientemente bien la mecánica del experimento. El estímulo escogido para el entrenamiento es un fragmento de un cortometraje[5] 360º de YouTube de 50 segundos de duración. Tras una breve explicación inicial de lo que deberán hacer durante la visualización del vídeo del experimento, se les proyecta el vídeo de entrenamiento y se les pide que registren los eventos de un determinado tipo. El tipo de eventos (finos o amplios) que se les pide registrar la primera vez es elegido al azar.

Tras visualizar el vídeo de entrenamiento se comprueba que el número de eventos registrados por el participante se ajusta aproximadamente al número de eventos calculados de antemano, y en caso de que así sea, el participante procede a visualizar los vídeos del experimento y al mismo tiempo registra el tipo de eventos que se le asignó en el entrenamiento previo. Si en el entrenamiento el participante registra un número inusualmente alto o bajo de eventos en comparación con los calculados de antemano, se le pide que repita el entrenamiento tratando de ajustar la granularidad de los eventos que registra. El número de veces que el participante repite el entrenamiento queda registrado para tenerlo en cuenta más adelante.

Una vez visualizado el vídeo del experimento, el participante repite el entrenamiento de la misma forma que la vez anterior, pero cambiando el tipo de evento que debe registrar. Si en el primer entrenamiento se dedicó a registrar eventos finos, esta segunda vez registrará eventos amplios y viceversa. De la misma forma, una vez terminado el entrenamiento, visualizará el vídeo registrando el nuevo tipo de eventos, si todo fue bien en el entrenamiento; o repetirá el entrenamiento si el número de eventos registrados en él fueron inusualmente bajos/altos.

3.1.3. Resultados

Para poder relacionar correctamente los cortes y los eventos señalados por los usuarios se buscaron los cortes y su tipo en cada uno de los vídeos y, para cada uno de los cortes, se asociaron los eventos cercanos en una ventana de ± 3 segundos, centrada en el momento del corte.

No se encontró una correlación suficientemente fuerte entre la segmentación de eventos finos y los cortes en los vídeos 360º. Considerando que la escala temporal de los eventos finos es de al menos un orden de magnitud menor que la duración promedio de una toma en realidad virtual

[1] Enlace a *Star Wars: Hunting of the Fallen*: https://youtu.be/SeDOoLwQQGo
[2] Enlace a *Always*: https://youtu.be/Tn_V8sVSnoU
[3] Enlace a *Invisible - Episode 2 - Back In The Fold*: https://youtu.be/M3FO3j2z5Tk
[4] Enlace a *Invisible - Episode 5 - Into the Den*: https://youtu.be/qYxNCB678WQ
[5] Enlace a *Invisible - Episode 6 - Into The Fire*: https://youtu.be/fz88kpRNTqM . El fragmento escogido de este vídeo comienza en el minuto 0:11 y termina en el minuto 1:01

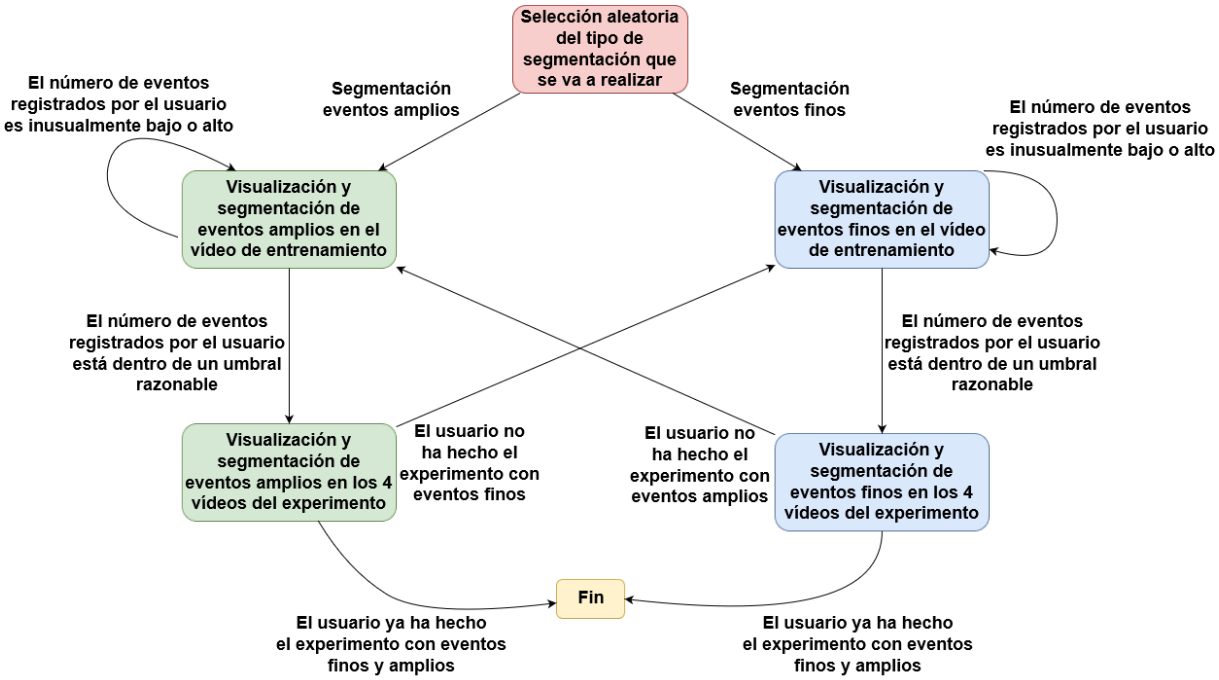


Figura 3.1: Diagrama aclaratorio con el flujo de sucesos en el experimento de segmentación de eventos

(segundos para el caso de los eventos finos y decenas de segundos para el caso de las tomas en realidad virtual) esto es comprensible. Por tanto solo se analizaron los datos para los eventos amplios. La Figura 3.2 muestra los resultados, agrupados por el tipo de corte. Los tipos de corte siguen el mismo esquema de Magliano y Zacks[9], el cual define la continuidad a lo largo de las dimensiones del tiempo, el espacio y la acción; y clasifican los cortes en tres tipos distintos, E_1, E_2, y E_3:

- E_1: cortes que son discontinuos en espacio, tiempo y acción (discontinuidades en acción).
- E_2: cortes que son discontinuos en espacio y tiempo, pero que son continuos en acción (discontinuidades espacio/temporales).
- E_3: cortes que son continuos en espacio, tiempo y acción (cortes en continuidad).

Los resultados muestran similitudes con los resultados de otros estudios recientes de cinematografía tradicional (Zacks, Magliano; [4, 9]). En primer lugar, las discontinuidades en acción son las que más consiguen que se produzca la segmentación de eventos, y por tanto son los predictores de los límites entre un evento y otro más fiables. En segundo lugar, los cortes en continuidad consiguen mantener la sensación de continuidad, incluso antes y después del corte, ya que el número de eventos asociados a este tipo de cortes es significativamente menor.

Por tanto, se puede asegurar que incluso en el nuevo formato de la realidad virtual la sensación de continuidad se mantiene.

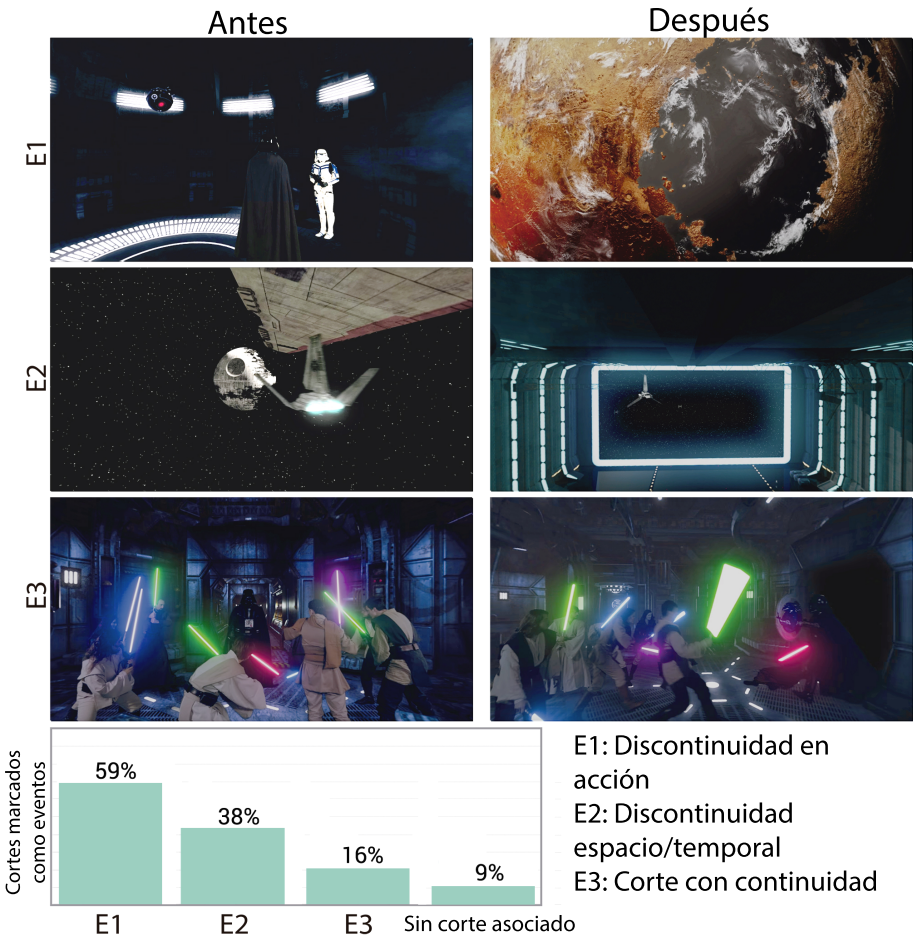


Figura 3.2: Arriba: Fotogramas representativos de *Star Wars - Hunting of the Fallen*, antes (izquierda) y después (derecha) del corte para cada uno de los tres tipos de corte, de arriba a abajo, E1 - discontinuidad en acción, E2 - discontinuidad espacial y E3 corte en continuidad. Abajo, resultados del experimento de segmentación de eventos para los eventos amplios, mostrando los porcentajes de los cortes de cada tipo marcados como nuevos eventos por los sujetos del experimento. Los resultados están normalizados por el número de apariciones de cada tipo de corte.

3.2. Experimento 2: Atención en cinematografía en realidad virtual

3.2.1. Objetivo

El objetivo de este segundo experimento es el de cuantificar la influencia en la percepción de continuidad de distintos parámetros característicos de los vídeos 360º. Dichos parámetros se apoyan en el concepto de las regiones de interés o ROIs tal y como se ha mencionado anteriormente. Dichas ROIs son regiones del vídeo 360º donde es más probable que un usuario esté mirando en un instante de tiempo determinado. Dada la elevada magnitud de parámetros que pueden influir en un metraje, se ha optado por reducir la dimensionalidad del espacio de parámetros a 3: el desalineamiento entre ROIs antes y después del corte (tomando como desalineamiento la distancia horizontal entre ellos en grados del vídeo 360º), la posición y número de ROIs antes

y después del corte, y los distintos tipos de cortes de la cinematografía tradicional. Todos ellos están descritos en detalle en la Sección 3.2.2.

3.2.2. Desarrollo del experimento

El planteamiento de este experimento consiste en que un conjunto de usuarios visualicen en realidad virtual una serie de vídeos 360º de 12 segundos de duración. Los vídeos en cuestión tienen un único corte justo a mitad del mismo, en el segundo 6, y han sido generados teniendo en cuenta las variables descritas anteriormente: desalineación de ROIs antes y después del corte, número y disposición de ROIs antes y después del corte, y tipos de corte cinematográficos.

Los vídeos que se utilizaron para generar estos clips de 12 segundos fueron grabados por un equipo profesional especializado en grabación de vídeos 360º, *Abaco digital*, quienes se encargaron de poner el equipo y de llevar a cabo la grabación de las tomas. Se trata de diversas tomas grabadas en 4 lugares diferentes, variando el número de ROIs y su distribución en cada una. Los ROIs de las grabaciones son personas del *Graphics and Imaging Lab* llevando a cabo distintas tareas en esos 4 lugares diferentes. La resolución de los vídeos es de 3840 píxeles de ancho por 1920 píxeles de alto y poseen una tasa de fotogramas por segundo (abreviado *fps*) de 59,97 fps.

Tras grabar los vídeos base, se llevó a cabo la edición y montaje de los clips de vídeo que se mostrarían a los sujetos del experimento. Los vídeos originales fueron editados y cortados de diversas formas, atendiendo a los parámetros circunstanciales del experimento que se deseaban probar, resultando en un total de 216 clips editados y montados manualmente.

El montaje se llevó a cabo con el software *Adobe Premiere Pro CC 2017*, donde se utilizaron diversos efectos y distorsiones de vídeo para producir los desalineamientos entre tomas. Para producir todas las combinaciones de tipos de corte y configuraciones de ROIs antes y después del corte, se cortaron y editaron manualmente los vídeos originales combinándolos de todas las formas posibles para producir los 216 clips que conforman el experimento.

Figura 3.3: Fotogramas de ejemplo de los vídeos 360º grabados mostrados mediante una proyección equirrectangular. Se pueden observar diferentes configuraciones de ROIs en ellos.

A continuación se describen en detalle los parámetros circunstanciales de los vídeos.

Tipos de desalineamiento entre ROIs

Hay 3 tipos de desalineamiento entre ROIs antes y después del corte:

- A_0 - Desalineamiento de 0º en el eje horizontal (ROIs alineados)
- A_{40} - Desalineamiento de 40º en el eje horizontal (ROIs desalineados levemente)

- A_{80} - Desalineamiento de 80º en el eje horizontal (ROIs muy desalineados)

Tipos de configuración de ROIs antes y después del corte

En lo que respecta a la configuración de ROIs antes y después del corte, hay 9 configuraciones diferentes. Estas configuraciones dependen tanto del número de ROIs antes y después del corte como de su posición relativa en el vídeo en cuestión. Esta posición relativa se basa en el campo de visión del dispositivo de realidad virtual, más conocido como FOV (por sus siglas en inglés, *Field Of View*). De esta forma, cuando se dice que hay *2 ROIs dentro del mismo FOV*, lo que quiere decir es que dentro del rango de visión del usuario se puede visualizar al mismo tiempo 2 ROIs; y del mismo modo, *2 ROIs en distinto FOV* significa que es imposible ver los dos ROIs al mismo tiempo.

Para describir las configuraciones posibles, se introducen dos variables: R_b (*ROIs before*) para describir la configuración de ROIs antes del corte; y R_a (*ROIs after*) para describir la configuración de ROIs después del corte.

Los valores que pueden tomar estas variables, atendiendo al número y disposición de ROIs, pueden ser 3:

- 1 ROI: El valor de $R_{a|b}$ es 0.
- 2 ROIs en el mismo FOV: El valor de $R_{a|b}$ es 1.
- 2 ROIs en distinto FOV: El valor de $R_{a|b}$ es 2.

Combinando la configuración de ROIs antes y después del corte da lugar a 3 combinaciones antes x 3 combinaciones después = 9 posibles combinaciones de ROIs diferentes.

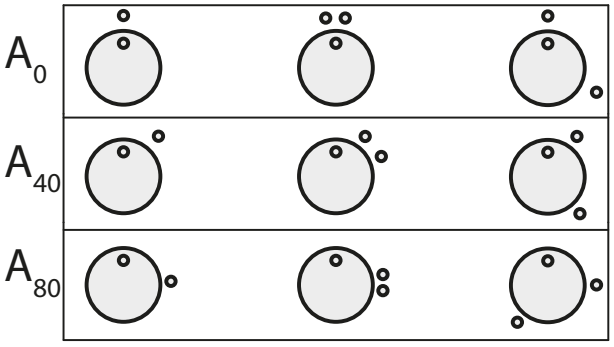


Figura 3.4: Gráfico ilustrativo sobre la disposición de los ROIs en relación con la visualización del usuario. El círculo grande representa el espacio 360º visto desde arriba, mientras que el círculo pequeño dentro del círculo grande representa la dirección hacia la que el usuario mira al comienzo de cada vídeo. Los círculos pequeños dispuestos en la parte externa del círculo grande representan la posición de los ROIs en el espacio 360º que rodea al usuario para cada uno de los casos. De arriba a abajo: A_0, A_{40} y A_{80}. De izquierda a derecha: $R_{a|b} = 0$, $R_{a|b} = 1$ y $R_{a|b} = 2$

Tipos de cortes cinematográficos

Por último, en relación con los tipos de corte cinematográficos, siguiendo el esquema de Magliano y Zacks (2011) [9] hay 3 tipos:

- E_1 - Corte con discontinuidad en acción
- E_2 - Corte con continuidad en acción pero discontinuidad en espacio o tiempo
- E_3 - Corte con continuidad en acción, tiempo y en espacio

Sin embargo, tras analizar múltiples vídeos narrativos 360º, se descubrió que el tipo de cortes E_3 solo se utiliza un 2,3 % de las veces, frente a un 73 % del uso del tipo de cortes E_1 y un 24,6 % de uso del tipo de cortes E_2, por lo que el tipo de cortes E_3 se descartó del experimento. Esto se puede deber en parte a que los cortes del tipo E_3 en cinematografía tradicional suelen ser cambios del punto de vista dentro de una misma escena/marco contextual de acción, sin embargo, estos cambios del punto de vista en realidad virtual no son tan utilizados ya que los creadores de contenidos pueden conseguir el mismo efecto con la posibilidad de que el usuario pueda mirar 360º alrededor de él.

Dentro de los tipos de corte E_1 y E_2, se utilizaron las siguientes técnicas cinematográficas para cada uno de ellos:

- Para E_1 - Jump cut (corte con salto): Aunque se suelen evitar en general cuando la toma sigue en la misma escena (Arev et al. 2014) [10], se usan a menudo para crear una transición abrupta de una escena a otra.
- Para E_2 - Compressed time (tiempo comprimido): Representa el paso de un periodo largo de tiempo montando conjuntamente dos tomas clave (por ejemplo, un personaje haciendo café en la cocina. En la vida real este evento puede durar de 2 a 3 minutos, pero se puede resumir rápidamente en un par de segundos con dos tomas mostrando al personaje cogiendo el agua, moliendo el café, dejándolo hervir y sirviéndolo en una taza)
- Para E_2 - Match-on-action (acción coincidente): Un corte en el que la acción de la segunda toma coincide con la acción de la primera toma (ejemplo, un personaje desplazándose hacia una puerta; conforme la puerta empieza a abrirse, se hace un corte a una toma con el personaje atravesando la puerta)

Adicionalmente, para cada triada de tipo de desalineamiento, configuración de ROIs y tipo de corte, se montaron y editaron 4 vídeos diferentes, para eliminar la influencia de la escena en la visualización de esa triada concreta. Así pues, el número de estímulos totales asciende a 3 x 9 x 2 x 4 = 216 vídeos diferentes.

Durante el experimento, cada participante visualizó 36 vídeos diferentes del conjunto de 216 vídeos totales. Los 36 vídeos que visualizó cada participante fueron elegidos al azar, salvo por una única restricción: que cada participante no pudiera visualizar el mismo vídeo con configuraciones de alineamiento diferentes. Esto se hizo para eliminar la posible influencia que pudiera tener el visualizar la misma escena con los mismos ROIs variando únicamente el desalineamiento entre antes y después del corte.

Para poder conocer la influencia de cada parámetro en la percepción de continuidad del vídeo, mientras el sujeto visualiza un vídeo, se está grabando constantemente la posición del vídeo hacia la que está mirando, mediante el uso de 2 cámaras que hacen un seguimiento ocular al sujeto,

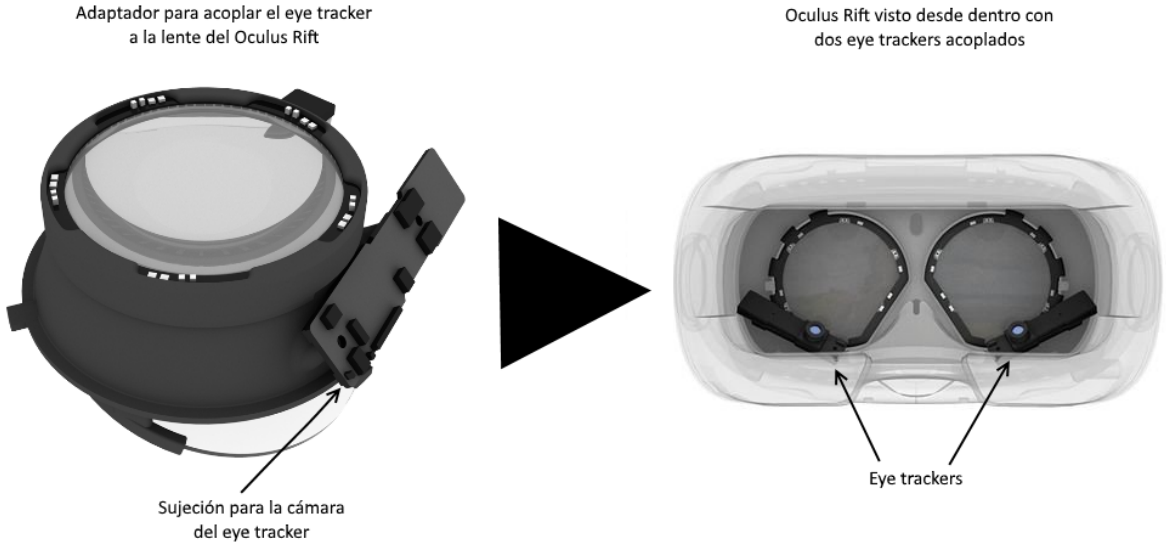


Figura 3.5: A la izquierda, el módulo porta lentes del HMD Oculus Rift DK2, en el cual se ha acoplado un adaptador que se encarga de sujetar el eye tracker. A la derecha, una visión del interior del HMD Oculus Rift DK2 con ambas lentes insertadas, con eye trackers incluidos

más conocidas por su nombre en inglés *eye trackers*, y el uso del propio dispositivo de realidad virtual, que aporta información de la posición y orientación de la cabeza durante el visionado.

De esta forma, conociendo en todo momento hacia dónde miran los usuarios, se puede averiguar qué parámetro influye más o menos en la percepción de continuidad y de qué forma lo hace. La Figura 3.5 ilustra visualmente el acoplamiento de los eye trackers al HMD Oculus Rift DK2 empleado en este experimento.

El análisis de los datos obtenidos durante el experimento se describe en el Capítulo 5.

4. Entorno de desarrollo

Para llevar a cabo los experimentos descritos en el Capítulo 3 son necesarios distintos componentes hardware y software. En primer lugar, para que los sujetos del experimento puedan percibir los estímulos se necesita un dispositivo de realidad virtual del tipo HMD. A través de él, el usuario puede visualizar los vídeos de realidad virtual y al mismo tiempo se pueden obtener los datos de rotación en los tres ejes de coordenadas de la cabeza del usuario; esto último se utiliza para saber hacia dónde está mirando en todo momento. El dispositivo de realidad virtual de tipo HMD elegido para este fin es el *Oculus Rift Development Kit 2*.

De la misma forma que el HMD proporciona información acerca de la orientación de la cabeza del usuario, es necesario un hardware y un software que permitan saber a qué posición está mirando el usuario dentro de la pantalla del dispositivo HMD. Combinando esta información, la de la orientación de la cabeza y la de la posición a la que está mirando el usuario dentro de la pantalla, podemos saber con certeza a qué coordenada del vídeo 360 está mirando el usuario, y por ende analizar posteriormente la influencia de los distintos parámetros circunstanciales en la trayectoria de la mirada de los usuarios. El hardware elegido para este fin son *dos cámaras USB* instaladas en los huecos para los ojos dentro del Oculus Rift DK2; y el software elegido que se encarga de procesar hacia dónde están mirando los usuarios es *Pupil Capture*, de la empresa *Pupil Labs*[11].

Por otro lado, para la implementación de los experimentos se necesita un software que sea capaz, no solo de renderizar y mostrar por la pantalla el contenido de los vídeos 360 a los usuarios del experimento, sino que también sea capaz de procesar y guardar la información relativa a las rotaciones del HMD y la posición a la que está mirando el usuario que generan el Oculus Rift y Pupil Capture, respectivamente. También hay que tener en cuenta que una de las formas más comunes de visualizar vídeos 360 es mapear la proyección equirrectangular del vídeo 360 en una esfera 3D y visualizar la esfera desde dentro de la misma, concretamente desde el centro.

Teniendo en cuenta estos aspectos, el software elegido para llevar a cabo estas tareas es *Unity3D*, el cual es un motor gráfico capaz de renderizar una esfera, proyectar los vídeos 360 en la misma y al mismo tiempo procesar y almacenar la información proveniente tanto de Pupil Capture como del HMD Oculus Rift DK2.

A continuación se describen más en detalle cada uno de estos componentes software/hardware y la integración que se ha llevado a cabo con los experimentos desarrollados.

4.1. Oculus Rift Development Kit 2

El Oculus Rift DK2 (en adelante *Oculus*) es un dispositivo de tipo HMD que está compuesto por una carcasa que cubre la parte frontal de la cara del usuario y aloja los componentes internos del mismo, unas lentes, una pantalla y unos sensores internos que detectan las rotaciones y movimientos del mismo de la cabeza del usuario. También posee una cámara externa USB completamente independiente de la carcasa que sigue los movimientos del dispositivo desde una posición alejada (normalmente desde el monitor del ordenador en el que funciona).

La pantalla del Oculus posee una resolución de 1920 x 1080 píxeles en formato 16:9, con un tamaño de 5,7 pulgadas y una tasa de refresco de 75 Hz. Esta pantalla se divide en dos partes, una para cada ojo, resultando en una pantalla menor para cada ojo de resolución de 960 x 1080 píxeles y de formato 8:9.

Del contenido mostrado a través del Oculus el usuario tiene un campo de visión (de ahora en adelante FOV por las siglas en inglés de *Field Of View*) de dicho contenido de 106 grados verticalmente y 95 grados horizontalmente, manteniendo la proporción 8:9 para cada ojo.

Entre los sensores internos del Oculus para detectar los movimientos del mismo están: un giroscopio para detectar cuándo el dispositivo rota; un acelerómetro que permite calcular la aceleración del dispositivo; y un magnetómetro para calcular la orientación con respecto al campo magnético terrestre.

Adicionalmente el Oculus provee una cámara USB que funciona a modo de sensor para saber la posición relativa del dispositivo respecto de la pantalla. Esta cámara se coloca encima del monitor de la máquina donde se está usando el Oculus, y es un sensor infrarrojo que hace un seguimiento de una serie de micro leds que están distribuidos por la parte frontal del Oculus, de tal forma que sirve para calcular eficientemente y de forma precisa los movimientos del dispositivo. De cara al usuario esto se traduce en una mayor precisión en la detección del movimiento de la cabeza y un pequeño extra que da sensación de *profundidad*: al acercar/alejar la cabeza a la cámara la imagen del Oculus da la misma sensación de estar acercándose/alejándose de dicho punto frontal.

Figura 4.1: En esta imagen del prototipo del Oculus Rift DK2 se pueden ver los leds infrarrojos que emplea la cámara infrarroja para detectar la posición relativa del Oculus respecto de la pantalla donde se sitúa la cámara

Sin embargo cabe mencionar que en este proyecto el sensor de la cámara no ha sido utilizado.

Esto se debe a que el formato de los experimentos consiste en la proyección de vídeos 360 en una esfera y el visionado de dichos vídeos proyectados desde el centro de la misma. Lo único que puede hacer el usuario para visionarlos es rotar su cabeza, ya que si se acercase/alejase de su posición central la imagen podría distorsionarse ligeramente y no mostrar fielmente el contenido original del vídeo. También podría producir mareos en personas sensibles.

Para que el visionado de imágenes de realidad virtual resulte cómodo a los usuarios debe proporcionar una tasa de fps ideal de 90, mínimo 60. Por ello se recomienda utilizar junto al Oculus un equipo con una tarjeta gráfica de gama media-alta además de un procesador de al menos 3.5GHz por núcleo, un puerto USB y una salida de HDMI. El equipo utilizado para este proyecto consta de:

- Procesador Intel Core i5-2500K a 3.30GHz, capaz de alcanzar los 3.60GHz.
- 32GB de RAM.
- Tarjeta gráfica Nvidia GeForce GTX 970..

Para que el Oculus funcione correctamente en el equipo es necesario tener instalado el software Oculus Runtime, que contiene los drivers necesarios para la comunicación entre el hardware Oculus y el equipo; así como el Oculus SDK, que provee la librería LibOVR que permite recibir información de los sensores y enviar imágenes a la pantalla del Oculus, así como desarrollar aplicaciones utilizando el Oculus. También permite modelar aplicaciones para el Oculus con las librerías de gráficos por ordenador OpenGL y DirectX.

4.2. Características de Unity3D

Unity3D es un software del tipo de motor gráfico diseñado principalmente para desarrollo de videojuegos pero que también ofrece la posibilidad de generar contenidos para realidad virtual. Se ha elegido este software de motor gráfico frente a otros como Unreal Engine o librerías de bajo nivel (OpenGL o DirectX) por su facilidad de uso y porque posibilita la integración de la realidad virtual en sus proyectos de forma sencilla y directa.

La estructura interna de Unity está basada en secciones, y cada una se encarga de una tarea. Existen secciones encargadas de gestionar la creación de gráficos a bajo nivel, otras de dar efectos a las imágenes, algunas se encargan de las físicas, etc... Sin embargo, los detalles de su estructura interna son propietarios, por lo que no son públicos. La Figura 4.2 muestra un ejemplo de la estructura interna típica de un motor de juegos como Unity3D para tomarla como referencia.

Los proyectos modelados en Unity3D se componen principalmente de *escenas*. Una escena es un agregado de otros componentes de Unity3D, llamados *GameObject*, los cuales se distribuyen a lo largo de los 3 ejes del espacio 3D que posee. Los GameObject, a su vez, están formados por una lista de componentes, los cuales son objetos de la clase *Component*. Estos componentes pueden ser de diversos tipos: existen componentes para detectar colisiones, para emitir sonidos, para definir qué parte se renderiza de un objeto, etc.

Un tipo destacable de componentes que permite definir el comportamiento de los GameObjects es el componente *Behaviour*, concretamente un subtipo de éste que se denomina *MonoBeha-*

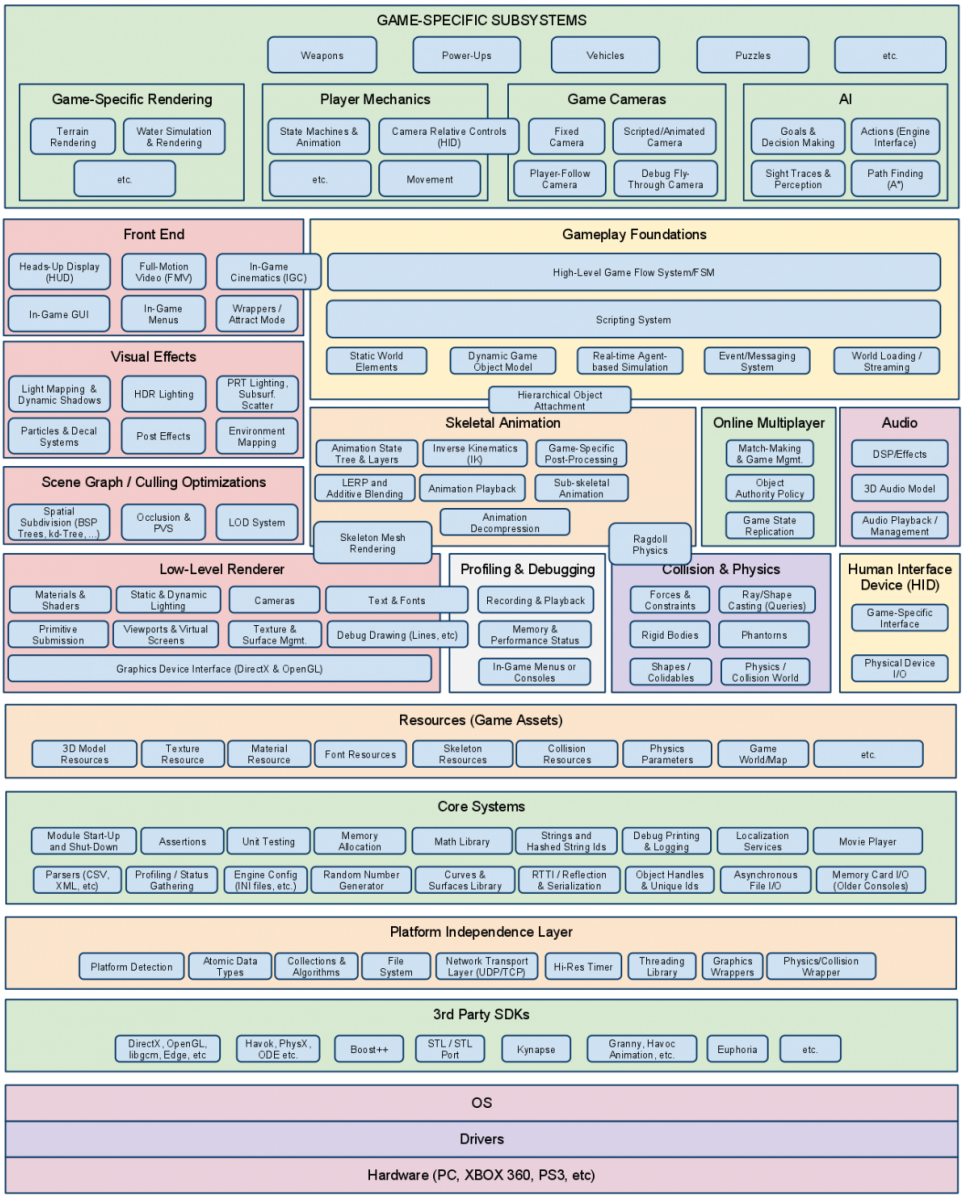


Figura 4.2: Ejemplo de la estructura interna típica de un motor de juegos como Unity3D. Más arriba en el diagrama representa un mayor nivel de abstracción y más abajo representa operaciones a más bajo nivel

viour. Este es el componente principal de los scripts que permiten programar el comportamiento de los GameObjects desde lenguajes como JavaScript o C#. Para este proyecto se ha elegido C# por su facilidad de manejo de cara a una programación orientada a objetos.

La renderización de una escena en Unity3D se realiza en dos partes, por un lado está la parte que renderiza los objetos de la escena y muestra la imagen final por pantalla para un fotograma concreto; y por otro lado está la parte que se encarga de hacer funcionar el comportamiento asociado a los objetos que hay en la escena. Para la primera parte, Unity3D necesita de al menos una cámara en la que basarse para renderizar la escena. La segunda parte simplemente recorre la lista de todos los objetos de la escena y va llamando a todos los métodos de los distintos

componentes en un orden determinado. Dicho orden se puede ver en la Figura 4.3.

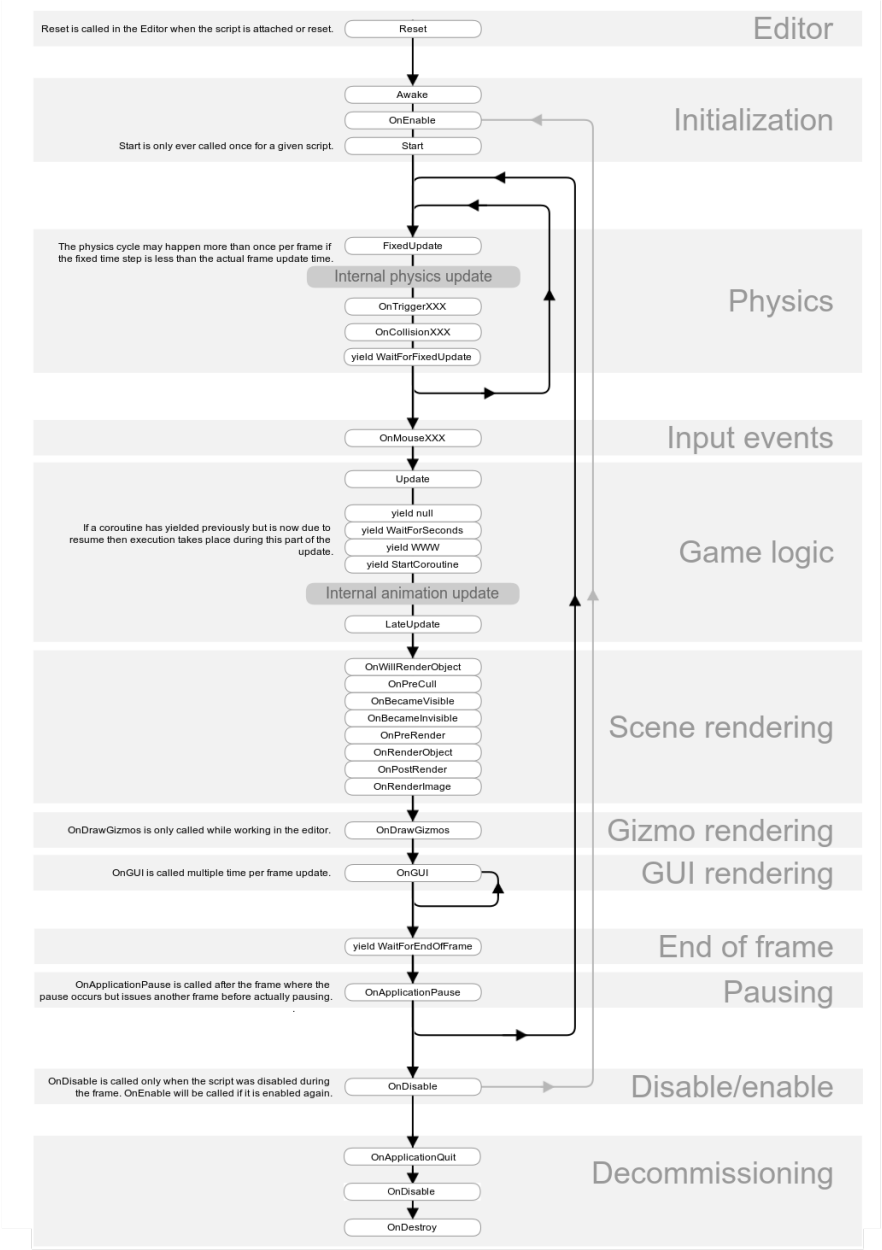


Figura 4.3: De arriba a abajo: orden de ejecución de los distintos métodos contenidos en scripts en Unity3D, primero llama al método Reset desde el Editor, para evaluar los cambios hechos en la escena (inclusión de nuevos Scripts), luego llama a una serie de scripts de inicialización y después a los scripts encargados de gestionar las físicas. A continuación, evalúa el input proporcionado por el usuario y tras ello, hace llamadas a distintas funciones para gestionar la lógica del juego. Una vez tratadas las físicas y la lógica del juego, renderiza la escena y la interfaz gráfica de usuario. Todo esto se repite en bucle a no ser que se seleccione otra ventana del sistema operativo o se cierre el juego, en cuyo caso la ejecución de los scripts se desvía a los métodos de la parte de "Disable/Enable.º "Decommisioning", respectivamente

4.3. Características de Pupil Capture

Tal y como se ha mencionado anteriormente, durante el experimento sobre la atención en narrativa en realidad virtual se hace un seguimiento continuo de la trayectoria de la mirada de los sujetos participantes en el experimento. Dicho seguimiento ocular, se lleva a cabo mediante dos cámaras alojadas en el interior del HMD Oculus Rift, las cuales reciben el nombre de eye trackers.

El software encargado de realizar este seguimiento ocular es *Pupil Capture*.

Pupil Capture es un software para realizar y grabar un seguimiento ocular mediante eye trackers. Es propiedad de Pupil Labs, una empresa que se encarga de desarrollar y lanzar el código de todo el software así como el hardware relacionado con los eye trackers. Todo el código de Pupil Labs y su software es open source, lo cual lo hace fácilmente modificable y ampliable mediante plugins. Tiene dos vertientes de funcionamiento, una con aplicaciones directamente en el mundo real, utilizando una tercera cámara para capturar el mundo real y de esta forma saber a qué parte del mundo real está mirando el sujeto; y otra vertiente más orientada al desarrollo de aplicaciones para otros tipos de entornos, en los que no se utiliza la tercera cámara para capturar el mundo real.

La versión de Pupil Capture empleada en este proyecto es la 0.8.7 y su funcionamiento es el siguiente:

- En primer lugar, se debe configurar Pupil Capture para trabajar con una cámara para capturar el mundo real o, por el contrario, si se quiere utilizar en otros entornos distintos al mundo real, como por ejemplo el de la realidad virtual.

- Antes de comenzar a grabar una sesión de seguimiento ocular es necesario calibrar los eye trackers. La calibración es un proceso mediante el cual Pupil Capture establece una relación entre la posición de la pupila que está percibiendo a través de los eye trackers y la posición de un marcador que se desplaza por la pantalla del HMD al cual el usuario tiene que mirar. Hay distintos tipos de marcadores para realizar el calibrado, y concretamente el utilizado en este proyecto es el marcador en forma de cruz.

  Una vez calibrados los eye trackers, Pupil Capture es capaz de saber a qué posición de la pantalla está mirando el usuario observando la pupila de éste.

- Una vez calibradas las cámaras, el dispositivo y el software están listos para capturar y grabar información de seguimiento ocular.

Como detalles adicionales, para este proyecto Pupil Capture corre en una máquina Linux y no requiere de un equipo especialmente potente para funcionar, por lo que las especificaciones de dicho equipo se han omitido. El motivo por el cual corre en una máquina Linux es que en el momento del desarrollo del proyecto Pupil Labs no daba soporte para Pupil Capture en Windows. Ahora, sin embargo, parece ser que sí es posible hacer funcionar Pupil Capture en Windows.

4.4. Integración de Pupil Capture y Unity 3D

Uno de los problemas de utilizar Pupil Capture y Unity3D al mismo tiempo es el sistema operativo en el que funciona cada uno.

Pupil Capture, como se menciona en la Sección 4.3, en el momento del desarrollo del proyecto solo daba soporte para Linux; mientras que el kit de desarrollo de aplicaciones del Oculus Rift DK2 empleado en Unity está en pausa para Linux desde hace más de un año, por lo que en estos momentos no es posible desarrollar adecuadamente aplicaciones para el Oculus Rift en Linux. Además, Unity en Linux presenta aún ciertos bugs, ya que el desarrollo está focalizado principalmente para Windows. Esto implica que para que haya una comunicación entre Unity y Pupil Capture es preciso utilizar dos máquinas, una con Windows para hacer funcionar Unity y el Oculus Rift DK2, y la otra con Linux, Pupil Capture y los eye trackers; y conectarlas mediante un cable ethernet. Así se puede conseguir que ambos software se envíen paquetes con la información necesaria para llevar a cabo los experiimentos

Por la parte de Unity 3D se ha utilizado una interfaz implementada en C# del protocolo *Open Sound Control* (OSC) para llevar a cabo este intercambio de mensajes, aunque concretamente se encarga de la parte de enviar órdenes a Pupil Capture. Este protocolo es un protocolo de comunicaciones que permite comunicar instrumentos de música, computadoras y otros dispositivos multimedia (por ejemplo móviles o PDA's equipados con bluetooth) pensado principalmente para compartir información musical en tiempo real sobre una red, pero que se puede utilizar con otros fines. Entre sus características principales se hallan:

- Es un protocolo ampliable, dinámico, con un esquema de nombres simbólicos tipo URL.
- Datos numéricos simbólicos y de alta resolución.
- Posee un lenguaje de coincidencia de patrones (pattern matching) para especificar múltiples receptores de un único mensaje.
- Marcas de tiempo (time tags) de alta resolución.
- Mensajes *empaquetados* para aquellos eventos que deben ocurrir simultáneamente.
- Sistema de *interrogatorio* para encontrar dinámicamente las capacidades de un servidor OSC y obtener documentación.

Por el lado de Pupil Capture hay dos componentes encargados de gestionar la intercomunicación. Por un lado, existe un servidor OSC, que funciona a través de una implementación en un script de Python y que se encarga de recibir las órdenes que Unity emite a través de su cliente de OSC en C#. Dada la naturaleza Open Source del software de Pupil Labs, es posible manipular el comportamiento de Pupil Capture a través de un script Python. Y es este mismo script, el del servidor OSC, el que se encarga de traducir las órdenes que llegan a través de Unity en comandos que Pupil Capture comprenda.

Así pues, para enviar órdenes de comenzar a grabar o calibrar, Unity utiliza esta interfaz del protocolo OSC para enviar comandos al servidor OSC de la máquina que corre Pupil Capture, y es éste servidor el que se encarga de poner a Pupil Capture a grabar o calibrar, respectivamente.

En lo que respecta a la transmisión de datos desde Pupil Capture a Unity, por el lado de Pupil Capture se utiliza un plugin incorporado llamado *Pupil Remote*, que basa su funcionamiento en la utilización de la librería *Zero MQ*, la cual, dada una dirección IP envía paquetes con toda la información relativa al seguimiento ocular que está capturando Pupil Capture.

Para que Unity reciba esta información, necesita un cliente de la librería Zero MQ. Este cliente está implementado en el script *Pupil Listener*, el cual desempaqueta y registra la información relevante del ojo capturada con Pupil Capture. Como detalles a destacar de este cliente implementado, es un cliente que utiliza un hilo de ejecución en paralelo al hilo de ejecución principal de scripts de Unity, ya que al parecer existe un bug que hace que bloquee el método *Update()* de un componente MonoBehaviour; y que los datos que recibe son fácilmente leídos mediante un parser del tipo JSON.

En resumen, cuando un experimento comienza, Unity se vale del protoclo OSC para enviar órdenes a Pupil Capture, el cual modifica su comportamiento y comienza a calibrar o grabar en función de la orden recibida. Si además está grabando, Pupil Capture envía paquetes a través de la red a Unity para que este pueda registrar en cada instante de tiempo la posición a la que están mirando los ojos, Unity los desempaqueta y escribe en los ficheros de log la información correspondiente. La Figura 4.4 muestra un diagrama de despliegue que trata de ilustrar gráficamente el funcionamiento de el sistema del proyecto al completo.

4.5. Diseño e implementación de los experimentos con Unity3D

En lo que respecta a la integración de un sistema de realidad virtual en Unity3D, desde la versión 5 el propio Unity3D integra directamente esta opción desde la configuración del proyecto, como se puede apreciar en la Figura 4.5. Activando esta función, los principales cambios que se aprecian es que en lugar de utilizar solo una cámara para renderizar la escena se cambia para que se usen 2 cámaras, de esta forma, se puede renderizar una imagen para cada ojo. Se puede escoger el modo de renderizado de cada cámara para ajustarlo al ojo izquierdo, al derecho, a la pantalla del ordenador o para ambos ojos al mismo tiempo. En este proyecto se han utilizado dos cámaras, con la opción de asignación de una para cada ojo, de tal forma que se activa el renderizado estéreo de la escena. Sin embargo, dada la naturaleza monoscópica de los vídeos 360º, al visualizarlos no da sensación de estar viendo una imagen estéreo.

Dado que los dos experimentos de los cuales se compone este proyecto requieren de la visualización de vídeos 360º en un entorno de realidad virtual, es necesario implementar en Unity3D un sistema de proyección de vídeos 360º. Concretamente el sistema utilizado se basa en un modelo 3D de una esfera con la particularidad de tener las normales de sus caras invertidas.

Esta particularidad se debe a que normalmente en un motor de juegos 3D se aplican una serie de optimizaciones para que el juego que se está desarrollando vaya lo más rápido posible. Una de estas optimizaciones es el llamado *back face culling* (determinación de caras ocultas). Es una técnica que consiste en determinar qué polígonos de la escena 3D que estamos renderizando deben ser considerados para renderizar y cuáles no es necesario renderizarlos, ya sea porque no es posible verlos desde la posición de la cámara o porque la cara del polígono que estamos viendo es una cara interna del modelo 3D (una cara que se supone que no se puede ver por ser la cara interna).

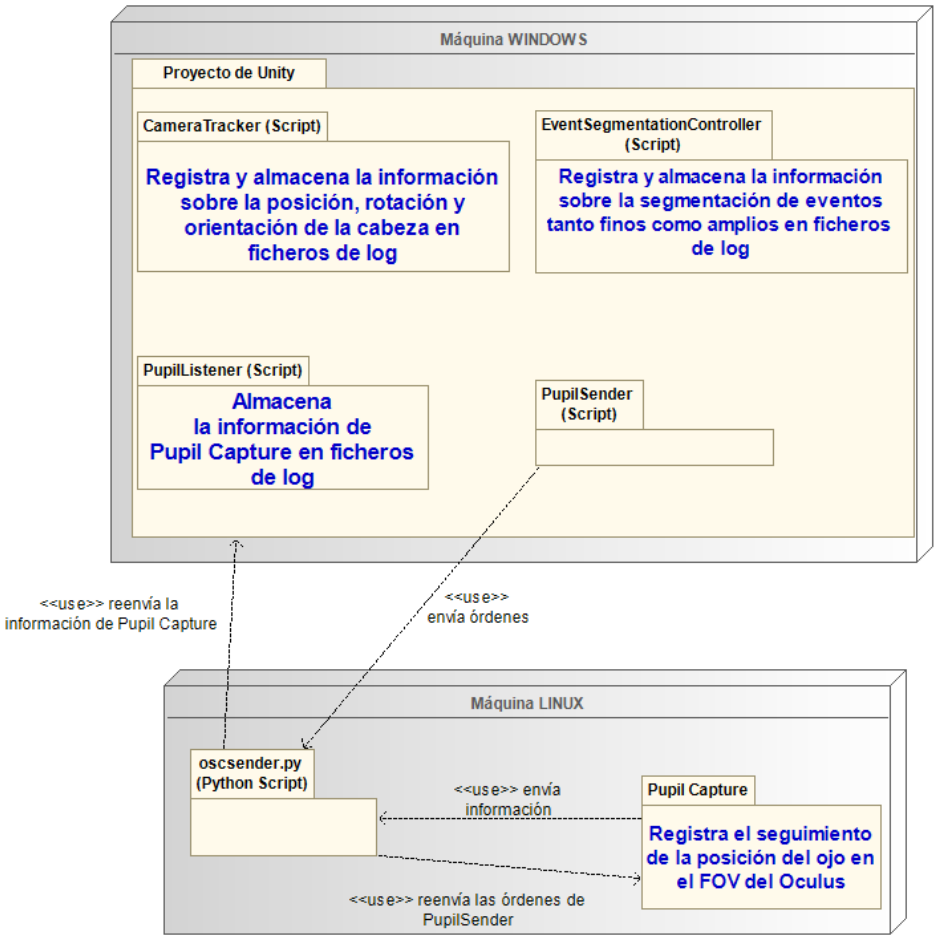


Figura 4.4: Diagrama de despliegue del sistema implementado en el proyecto al completo

Esta técnica se puede implementar de diversas formas. Una de ellas es la que utiliza Unity3D: se calcula el ángulo existente entre el vector de visualización de la cámara y el vector normal del polígono en el espacio 3D que estamos determinando si debemos renderizar o no. Si este ángulo sale mayor de 90º quiere decir que desde la cámara no podemos visualizar el polígono y por tanto el motor de juegos (Unity en este caso) no lo renderiza.

Una esfera predeterminada generada con un programa de modelado 3D (o por el propio Unity3D) posee todos sus polígonos con las normales apuntando hacia el exterior de la esfera. Esto hace que una cámara situada fuera de la esfera renderice correctamente la esfera, pero una cámara colocada *en el interior* de la esfera hará que la esfera no se vea, el ángulo entre el vector de visualización de la cámara y las normales de la esfera siempre será mayor que 90º.

La solución para poder proyectar los vídeos 360º en el interior de la esfera y poder visualizarlos desde dentro con una cámara es invertir las normales de la esfera de tal forma que apunten hacia el centro de la esfera. Esto hace que, al colocar una cámara dentro de la esfera, esta renderice correctamente desde dentro de la esfera el contenido proyectado.

Otra opción de implementación de este experimento consistiría en utilizar dos esferas, una

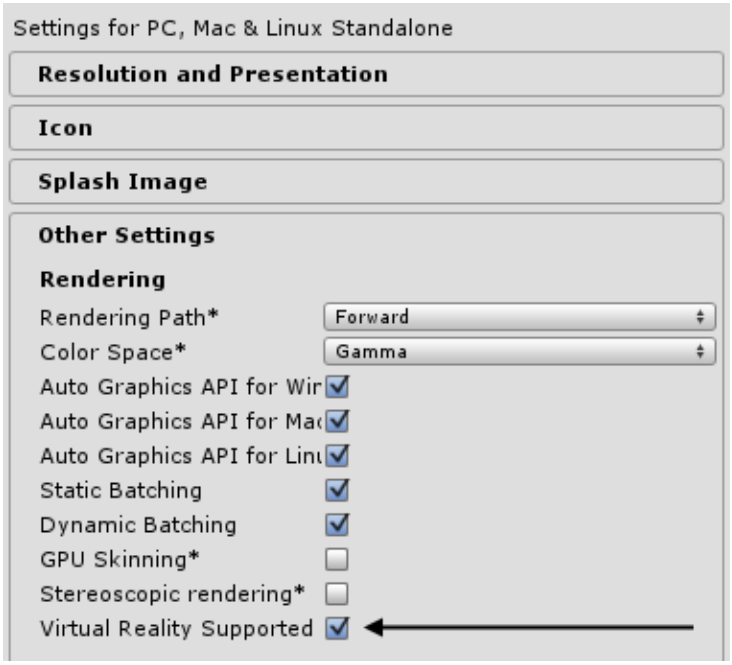


Figura 4.5: En el apartado de propiedades del proyecto de Unity es posible activar automáticamente el soporte para realidad virtual marcando la casilla señalada. Como ya se ha mencionado, para que funcione correctamente con el HMD Oculus Rift DK2 hace falta instalar el Oculus Runtime y el Oculus SDK

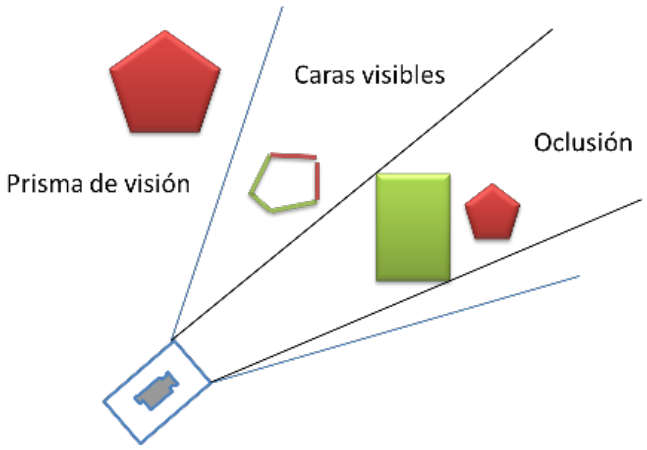


Figura 4.6: Ejemplo visual del efecto del back face culling. Los polígonos marcados en verde son renderizados, mientras que los marcados en rojo no se renderizan. La cámara se encuentra abajo a la izquierda y apunta hacia arriba a la derecha. Se incluye un ejemplo también de back face culling por oclusión, el cuadrado verde tapa al pentágono rojo de la derecha y es por ello que este no se renderiza a pesar de que posee caras en teoría visibles (aquellas cuya normal forma un ángulo de menos de 90° con el vector de visión de la cámara). El pentágono rojo de la izquierda no se renderiza por estar fuera del cono de visión de la cámara.

para cada ojo, de tal forma que si se dispusiese de un vídeo para el ojo derecho y otro vídeo para el ojo izquierdo se podrían visualizar los vídeos 360º en estéreo. Sin embargo, dado que en la actualidad la mayoría de contenido de vídeo para realidad virtual es monoscópico, se optó por grabar y utilizar vídeos en formato monoscópico, para lo cual es solo necesaria una esfera.

La escena en Unity3D se compone pues de los siguientes objetos:

- *Objeto CameraNegater*: Se trata de un agregado de varios objetos entre los que se incluye la *cámara principal* de la escena. Lleva unidos a él 3 scripts, de los cuales 2 gestionan el funcionamiento de la cámara y otras funcionalidades; y el último script, que es el que gestiona prácticamente todo el funcionamiento del pipeline de los experimentos: inicializa un experimento, gestiona que se escriba en los ficheros la información correspondiente, etc.

  - Script *CounterCamTransform*: Ofrece un par de métodos para modificar/corregir la posición/rotación de la cámara y una vez por fotograma a través del método *Update* aplica una corrección de la posicíon al objeto CameraNegater. Esto se debe a que el Oculus aplica un desplazamiento en cada fotograma según cómo se mueva el usuario, y para mantener al usuario en el centro de la esfera y limitar únicamente sus movimientos a rotaciones y no desplazamientos, este script corrige el movimiento que realiza el Oculus (lo anula).
  - Script *CameraTracker*: Este script es que el gestiona todo el funcionamiento del pipeline de los experimentos. Al inicializar el proyecto realiza las llamadas necesarias a otros scripts u objetos con el fin de prepararlo todo para llevar a cabo un experimento y durante el mismo también es el encargado de hacer las llamadas necesarias para realizar primero una calibración de los eye trackers y luego ir proyectando los vídeos que compongan el experimento.

  También posee diversos parámetros modificables desde el editor de Unity para que no sea necesario abrir el editor de código para cambiar el comportamiento del pipeline de los experimentos. Esto se hace porque el mismo pipeline para el experimento de segmentación de eventos se reutiliza para el experimento de atención de usuarios en realidad virtual, solo que al cambiar estos parámetros, se modifica el comportamiento del proyecto para adecuarse a uno u otro escenario.

- *Objeto PupilListener*: Contiene un único script, el script *PupilListener*. Este script se encarga de recibir y desempaquetar la información recibida desde Pupil Capture y, en caso de estar a mitad de un experimento, almacenar y escribir la información del eye tracker en el fichero de log. Para recibir información de Pupil Capture emplea una implementación en C# de un cliente de tipo *NetMQ*, el cual se explica más en detalle en la sección 4.5. También posee funciones para recibir órdenes desde otros scripts, como CameraTracker.

- *Objeto PupilSender*: Contiene un único script, el script *PupilGazeTracker*. Este script posee funciones para enviar órdenes a la máquina donde esté funcionando Pupil Capture, tales como que empiece a grabar o que comience la calibración de los eye trackers. Para ello dispone de funciones que pueden ser llamadas desde otros scripts, como CameraTracker. En la sección 4.5 se explica más en detalle la tecnología empleada para comunicarse con Pupil Capture.

- *Objeto Randomizer*: Contiene un único script, el script *Randomizer*. Su única funcionalidad es la de proveer un método para que otros scripts puedan obtener una lista de n vídeos aleatorios contenidos en una carpeta, dando prioridad a los que aún no han salido tantas veces como el resto. También permite filtrar la lista de vídeos devueltos según su alineamiento (para evitar que un mismo sujeto del experimento sobre atención de usuarios

en cinematografía en realidad virtual vea el mismo vídeo pero con distinta alineación). El resto de funciones son internas para el funcionamiento del script.

- *Objeto EventSegmentation*: Posee un único script, el script *EventSegmentationController*, que es el que controla el funcionamiento de la segmentación de eventos y la escritura de dichos eventos registrados en el fichero de log. Posee también un método público accesible desde otros scripts para activar el registro de la segmentación de eventos y que Unity registre las pulsaciones de la barra espaciadora como eventos, ya sean finos o amplios. El tipo de eventos registrados se configura en un parámetro del script CameraTracker del objeto CameraNegater, y este script modifica el fichero escrito en función de dicho parámetro.

- *Objeto Sphere*: Contiene un único script, el cual, especificándole un archivo con formato .ogg proyecta en la esfera el vídeo utilizando el motor de renderizado de texturas y la superficie del objeto como pantalla.

  Cabe mencionar que tras implementar este sistema y hacer diversas pruebas, se comprobó que carecía de las cualidades necesarias para poder proporcionar una experiencia de realidad virtual óptima. Dados los altos costes computacionales de las funciones por defecto de Unity3D para proyectar vídeos en modelos 3D, los vídeos proyectados nunca alcanzaban la tasa de fotogramas por segundo mínima en realidad virtual, que es 60fps, por lo que se optó para este proyecto relegar la proyección de vídeos a un plugin de Unity3D de terceros, concretamente, el plugin AvPro Video, de Renderheads. Este plugin es capaz de proyectar los vídeos eficientemente y de adaptarse a cualquier archivo de vídeo y formato sin necesidad de convertirlos al formato .ogg.

- *Objeto 360SphereVideo*: Está compuesto de otros dos objetos, el modelo 3D de la esfera (objeto *Sphere*), donde se proyectan los vídeos 360, y un objeto que es el que contiene el plugin para reproducir vídeos 360 (objeto *AVPro Video Media Player*). Dicho objeto posee muchos componentes enlazados a él, pero el más relevante es el script *MediaPlayer*, que permite la reproducción de vídeos utilizando el plugin proporcionado por AVPro, además de permitir configurar otras opciones.

La Figura 4.7 ilustra mediante un diagrama de clases la estructura de la escena de Unity. Nótese que en la enumeración anterior se han descrito los objetos y scripts más relevantes de la escena, pero en el diagrama aparecen todos los componentes del proyecto.

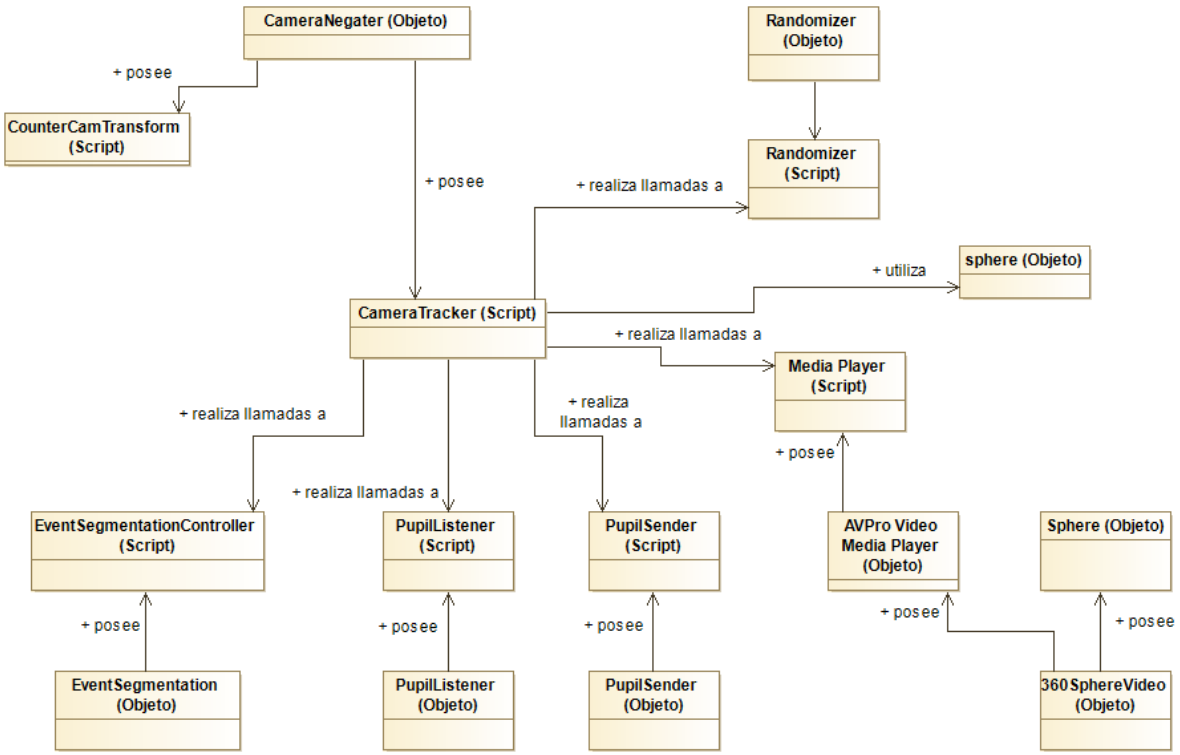


Figura 4.7: Diagrama de clases que representa la arquitectura software del proyecto Unity

5. Análisis

En esta sección se describen los procedimientos realizados para llevar a cabo el análisis de los datos obtenidos en el experimento sobre la atención de los usuarios en cinematografía en realidad virtual. Los resultados del primer experimento están expuestos en la Sección 3, ya que se trata únicamente de una validación del proceso cognitivo de segmentación de eventos.

5.1. Recolección y pre-procesado de datos

5.1.1. Recolección de datos

Tal y como se ha descrito anteriormente, mientras los sujetos visualizan los experimentos el software Unity3D guarda, para cada instante de tiempo, un registro de la posición y orientación de la cabeza y de la posición de la pantalla a la que están mirando los sujetos mediante la información proporcionada por los eye trackers. Este registro es distinto para cada usuario y para cada vídeo mostrado, dejando tantos ficheros como visualizaciones por usuario de un vídeo determinado se hayan hecho.

5.1.2. Pre-procesado: Datos de referencia

Antes de procesar los datos recogidos, se han recogido otros datos adicionales para utilizarlos como datos de referencia. Esto se hace para comparar los datos recogidos durante el experimento con los datos de referencia, de tal forma que se pueda analizar y eliminar la influencia de los cortes en la alteración de la mirada en los sujetos.

Estos datos de referencia consisten en 10 visualizaciones adicionales para cada uno de los vídeos grabados sin editar. De esta forma, se puede comparar la trayectoria de la mirada de referencia obtenida de los vídeos originales con la trayectoria de la mirada de los fragmentos de los vídeos del experimento. El orden de las visualizaciones es puramente aleatorio. Una vez obtenidas las 10 trayectorias de las miradas para cada uno de los vídeos originales se calcula la trayectoria de la mirada media entre usuarios para cada vídeo. La figura 5.1 muestra un ejemplo de esta trayectoria media.

Una vez calculada la trayectoria media, para asegurarse de que estos datos se pueden usar como referencia se tiene que certificar la congruencia de dichos datos entre sujetos. Para ello se utiliza una curva ROC (de sus iniciales en inglés *Receiver Operating Characteristic*), la cual

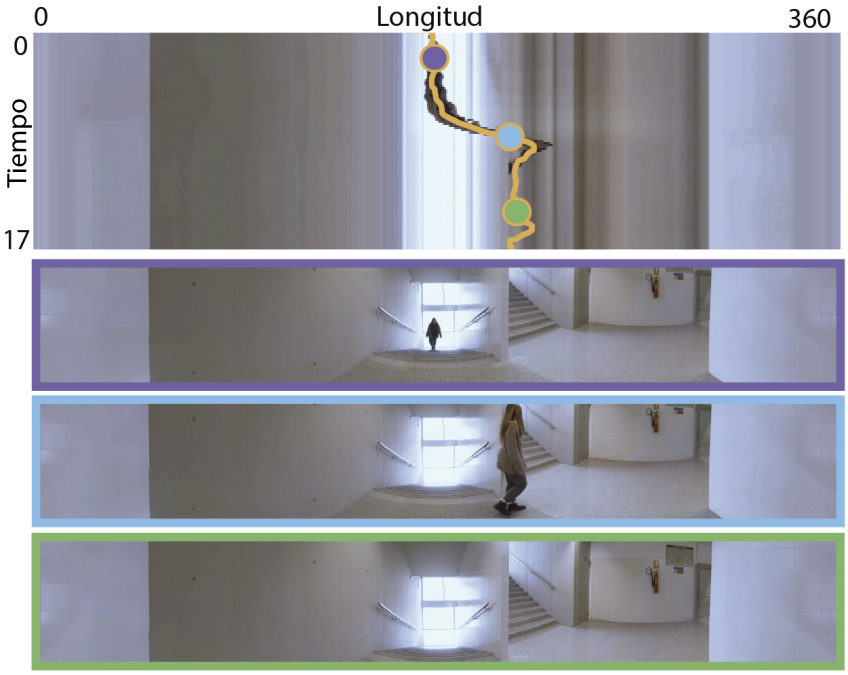


Figura 5.1: Trayectoria media de los sujetos para un vídeo de ejemplo. La primera imagen muestra en el eje X la longitud del vídeo, y en el eje Y el tiempo transcurrido. La zona negra representa las trayectorias de la mirada de los 10 usuarios recogidas. La línea amarilla representa la trayectoria media calculada. Los círculos de colores representan los instantes de tiempos correspondientes a los fotogramas mostrados debajo de la primera imagen. Se puede ver que la posición de la trayectoria media coincide con el punto de interés de los fotogramas.

proporciona una medida de la congruencia entre observadores para cada instante de tiempo.

En primer lugar, se agregan las fijaciones de todos los usuarios en ventanas de tiempo de dos segundos, y se convolucionan mediante una gaussiana bidimensional con $\sigma = 1$ de grado de ángulo visual, lo cual proporciona un *mapa de saliencia* para cada ventana de tiempo. La correspondiente curva ROC se calcula usando un enfoque uno-contra-todos dejando fuera al i-ésimo sujeto. Entonces, se calcula, para cada mapa de saliencia, las regiones más salientes dado un porcentaje de saliencia k; y a continuación, se procede a calcular el porcentaje de fijaciones de ese sujeto i-ésimo que se encuentran dentro de esas regiones salientes.

Este proceso se repite para una serie de umbrales de la variable k, con valores oscilantes entre el 0 % y el 100 %, y los puntos resultantes son los que definen cada curva. Para comprender mejor la evolución de la congruencia entre observadores a lo largo del tiempo, se calcula el área debajo de la curva para cada ventana de tiempo. Dicha área oscila entre los valores 0, que significa que hay incongruencia entre los sujetos, y 100, que significa que la congruencia es completa entre los sujetos.

Tal y como muestra la Figura 5.2, la congruencia entre observadores a lo largo del tiempo permanece muy elevada, alcanzando un valor de 1 con solo el 2 % de las regiones consideradas salientes por el resto de sujetos, y permanece constante para valores crecientes de k. En la parte derecha, está la misma interpretación desde la perspectiva del área bajo la curva, todas las fijaciones del sujeto caen de media dentro del 2 % de las regiones consideradas salientes. Esto indica que todos los sujetos consistentemente consideraron las mismas regiones salientes, y por

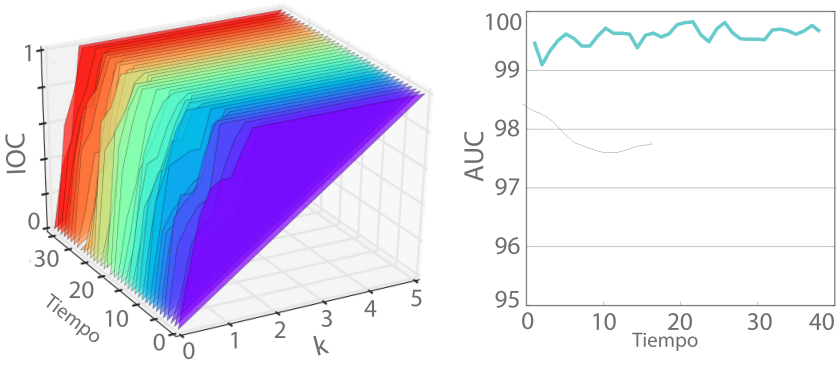


Figura 5.2: A la izquierda, la congruencia entre observadores a lo largo del tiempo para uno de los vídeos sin editar. Se calcula la curva ROC para cada segundo del vídeo. A la derecha, la evolución temporal del área bajo la curva calculada para cada una de las curvas ROC. Los altos valores de la congruencia entre observadores y de la área bajo la curva indican que los sujeatos consideraron consistentemente las mismas regiones como salientes.

tanto, los datos de referencia son válidos para su uso.

5.2. Métricas y análisis de datos

5.2.1. Métricas

En el siguiente apartado se describen las medidas empleadas para analizar el comportamiento de la mirada de los sujetos tras un corte. Adicionalmente, para descubrir patrones subyacenctes en el comportamiento de los usuarios que estas métricas no puedan detectar se ha utilizado un *análisis de secuencias de estados*, el cual se describe más adelante. Las métricas empleadas son:

- *Fotogramas en alcanzar una ROI*: La más sencilla de las métricas, indica el número de fotogramas tras el corte que el sujeto tardó en fijarse de nuevo en la ROI o las ROIs de la toma tras el corte. Es indicativo del tiempo que se tarda en converger la mirada de nuevo a la acción principal tras el corte.
- *Porcentaje de las fijaciones totales dentro de la ROI*: Este porcentaje se comienza a calcular después de la primera fijación en la ROI que se produce después del corte. Es, por tanto, independiente de la métrica de fotogramas en alcanzar una ROI. Dadas las configuraciones distintas de ROIs, es posible que el número de fijaciones dentro de una ROI varíe enormemente de una configuración a otra (ya que varía el número de ROIs entre configuraciones). Para compensar esto, se calcula el porcentaje relativo al porcentaje medio de fijaciones dentro de la ROI para cada configuración de ROIs, *antes del corte*. Esta métrica indica el grado de interés del sujeto en las ROIs.
- *Error en la trayectoria de la mirada*: Se calcula el error RMSE para cada trayectoria de mirada respecto a la trayectoria de mirada de referencia correspondiente. Esta métrica aporta información sobre cómo influye el tipo de corte en el comportamiento de la mirada. Una vez más, esta métrica se calcula después de la primera fijación en la ROI después del corte, para que sea independiente de los fotogramas en alcanzar una ROI.

- *Número de fijaciones*: Para calcular esta métrica, se calcula la relación entre el número de fijaciones y el número total de muestras de mirada después del corte después de alcanzar una ROI. De esta forma, se eliminan los posibles incrementos en los movimientos oculares (no fijaciones) mientras se busca una ROI después del corte. Un valor bajo en esta métrica corresponde con un alto número de movimientos oculares, lo cual sugiere un comportamiento más explorativo ante el corte, produciendo menos fijaciones en una región particular o acción concreta.

- *Secuencias de estados*: Para esta métrica, se clasifican las fijaciones de los sujetos a lo largo del tiempo en 4 posibles estados: fijaciones en la ROI primaria, fijaciones en la ROI secundaria (para aquellos casos con 2 ROIs. La clasificación de una ROI como primaria o secundaria es al azar), fijaciones en el resto de la escena que no es una ROI y un estado llamado *inactivo*, que representa un estado en el cual no hay fijaciones para registrar (se produce cuando hay más movimientos oculares que fijaciones). Mediante esta clasificación, se puede describir el comportamiento visual de los sujetos como si de una secuencia de estados se tratase, así como el tiempo invertido por cada sujeto en cada estado. Para representar el patrón general de secuencias de estado para cada configuración se utiliza un *análisis de distribución de estados*, el cual provee una vista agregada de la frecuencia de cada estado para cada intervalo de tiempo. Se ha utilizado para este análisis la librería para R *TraMineR*.

5.2.2. Análisis

Dado que no se puede asumir que las observaciones sean independientes, se ha utilizado *modelado multinivel* en el análisis, el cual es adecuado dada la naturaleza los datos. El modelado multinivel permite especificar ciertos factores aleatorios entre los predictores, es decir, contempla la posibilidad de que el modelo difiera para diferentes valores de estos factores aleatorios. Para este caso, el factor aleatorio es el sujeto concreto visualizando el vídeo.

En la regresión se incluyen los 4 factores de los que se ha hablado anteriormente: el desalineamiento entre ROIs antes y después del corte, el tipo de corte cinematográfico empleado, la configuración de ROIs antes del corte y la configuración de ROIs después del corte. Dada la naturaleza categórica de algunas de las variables entre los predictores, se recodifican como variables binarias temporales durante la regresión.

Para dos de las métricas calculadas (porcentaje de fijaciones dentro de la ROI y número de fijaciones) se ha detectado que la influencia del sujeto es significativa, lo cual indica que no podemos tratar las muestras para esas métricas como independientes. Para las otras dos métricas (frames en alcanzar ROI y error de la trayectoria de la mirada) el sujeto no tenía efecto. Por tanto las muestras pueden considerarse independientes.

Tras esto, se lleva a cabo un análisis ANOVA, junto a un análisis Bonferroni *post hoc* para buscar efectos significativos en los datos. A lo largo del análisis se utiliza un nivel de significancia de 0.01.

6. Resultados

En el siguiente capítulo se listan los efectos y la influencia que posee cada uno de los parámetros considerados en el experimento sobre cinematografía en realidad virtual.

6.1. Influencia del desalineamiento entre ROIs antes y después del corte

Una de las primeras cosas que se observan es que el factor del alineamiento entre ROIs antes y después del corte influye claramente en las 4 variables (métricas) estudiadas. En el caso de los frames hasta alcanzar la ROI, el análisis de Bonferroni *post hoc* muestra una diferencia significativa entre los 3 desalineamientos diferentes, siendo mayor el número de frames cuanto más desalineado está el vídeo; lo cual, por otro lado, se ajusta a lo esperado, ya que cuanto más lejos se halla la ROI después del corte más se tarda en encontrarlo. La métrica también muestra una tendencia exponencial cuanto mayor es el grado de desalineamiento. Esto se puede apreciar en la Figura 6.1, gráfico izquierdo, el cual incluye la curva ajustada y el intervalo de confianza del 95 %.

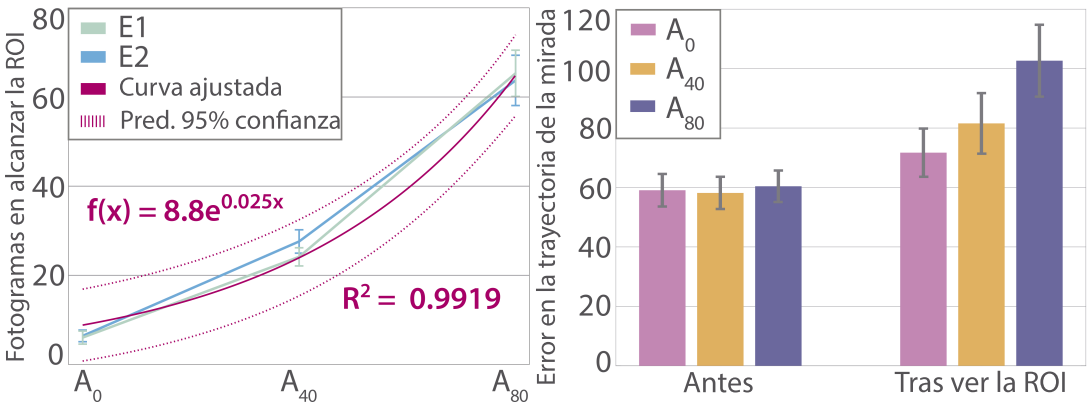


Figura 6.1: A la izquierda, la media de los fotogramas en alcanzar una ROI para cada alineamiento. Las curvas verde y azul muestran los datos promedio para los dos tipos de corte (E_1 y E_2 respectivamente). Se muestra asimismo una curva ajustada a una exponencial con el intervalo de confianza del 95 % asociado. A la derecha: Media del error RMSE respecto a antes del corte, y después del corte después de ver la ROI (error en la trayectoria de la mirada) para las diferentes condiciones de alineamiento incluidas en el experimento. En ambas gráficas, las barras de error muestran un intervalo de confianza del 95 % para la media

La métrica del error de la trayectoria de la mirada muestra en un análisis *post hoc* que no hay

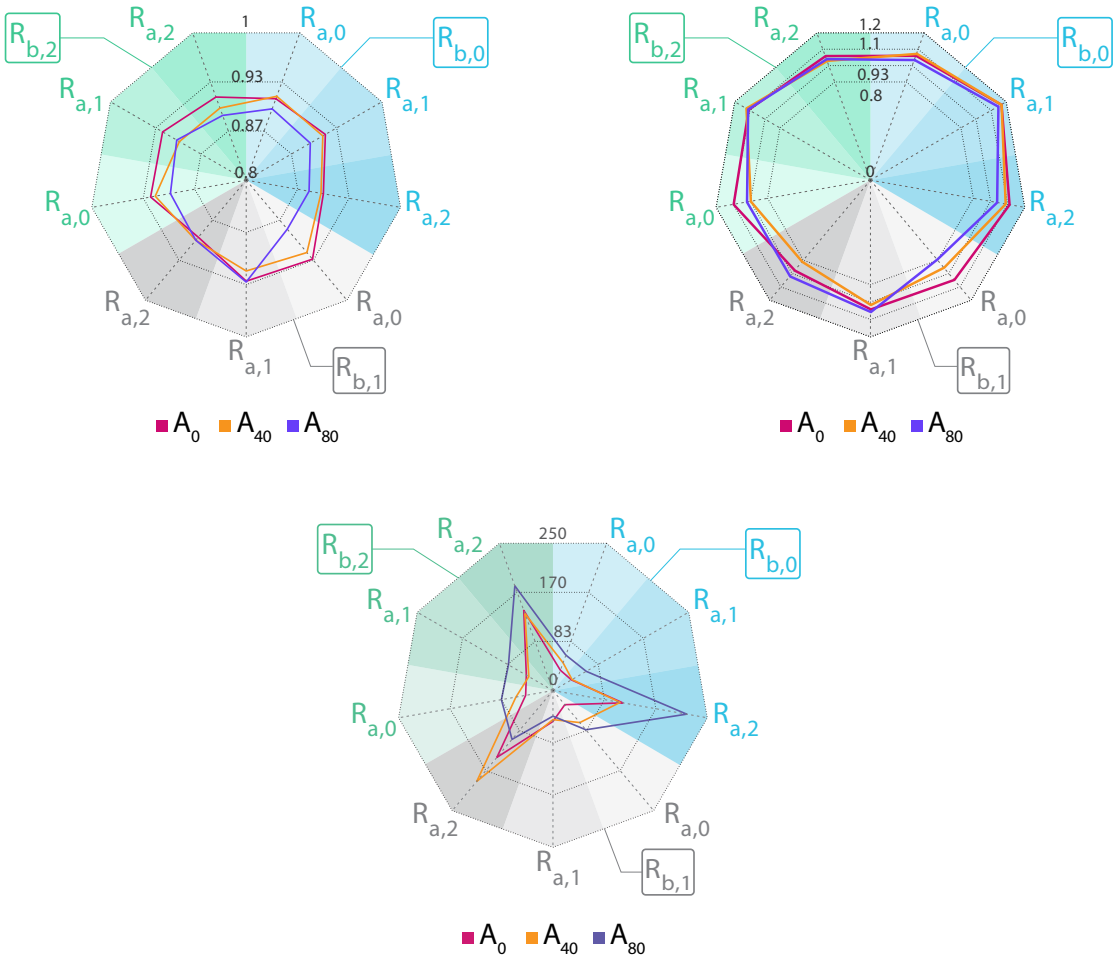


Figura 6.2: Gráficos de radar que muestran la variación de las tres métricas de la alineación (A) y las configuraciones de ROIs antes y después del corte (R_a y R_b. La variación sobre la variable E no se incluye. En cada gráfico, las tres curvas corresponden a cada una de las condiciones de los alineamientos, tal y como se muestra en la leyenda. Los *radios* del gráfico corresponden con las diferentes combinaciones de R_a y R_b. Los valores de R_a están escritos en los límites del gráfico, mientras que las tres secciones coloreadas de azul, gris y verde corresponden a los valores de $R_b = 0$, $R_b = 1$ y $R_b = 2$ respectivamente. *Arriba izquierda:* Métrica del número de fijaciones después del corte después de ver la ROI, la cual se puede apreciar que muestra una diferencia significativa en A_{80} respecto de A_{40} y A_0. La escala del eje radial está agrandada con el propósito de una mejor visualización. *Arriba derecha* Valor del porcentaje de fijaciones totales dentro de la ROI para cada una de las condiciones; este valor se ve afectado significativamente por la configuración de las ROIs tanto antes como después del corte (detalles en el texto). **Abajo:** Error RMSE medio con respecto a después del corte tras fijarse en la ROI (*error en la trayectoria de la mirada*). Más detalles en el texto.

tanta diferencia entre el alineamiento perfecto y el desalineamiento leve, sin embargo, cuando el desalineamiento es grande es mucho mayor. Esto indica que a los sujetos les cuesta mucho más encontrar la ROI con una diferencia de alineamiento grande, aunque con una desalineación leve se recuperan significativamente pronto. Con la métrica del porcentaje de fijaciones dentro de la ROI sucede algo similar, la diferencia con el desalineamiento grande es mucho mayor con el alineamiento perfecto y el desalineamiento leve. Esto se puede apreciar en la Figura 6.1, gráfico derecho, comparando directamente con los valores equivalentes de antes del corte (antes del corte no hay una diferencia significativa entre desalineamientos ya que el corte aún no se ha producido).

La métrica del número de fijaciones presenta la misma diferencia significativa entre el alineamiento perfecto y desalineamiento leve comparado con el desalineamiento grave (ver Figura 6.2, gráfica superior izquierda). Este efecto indica que los desalineamientos grandes alteran el comportamiento de los sujetos, no solo en el tiempo que se tarda en percibir la ROI después de la toma si no en las fijaciones después de haber sido encontrado. Esto sugiere que los sujetos muestran un comportamiento más exploratorio cuando hay un gran desalineamiento en el corte.

6.2. Influencia del tipo de corte cinematográfico empleado

En cuanto al tipo de corte cinematográfico empleado, no se ha detectado una influencia significativa en el porcentaje de fijaciones totales dentro de las ROIs, ni en el número de fijaciones ni tampoco en el error en la trayectoria de la mirada. Sin embargo sí que se percibe un efecto de interacción entre el tipo de corte con las configuraciones de las ROIs.

Sorprendentemente, el tipo de corte empleado no influye significativamente en la métrica de los fotogramas en alcanzar una ROI, tal y como sugiere la Figura 6.1, en el gráfico izquierdo, curvas azul y verde.

6.3. Influencia de las distintas configuraciones de ROIs en la escena antes y después del corte

Según los resultados del análisis, no se percibe una influencia notable entre las configuraciones de ROIs en la escena en la métrica del número de fijaciones totales, lo cual indica que no influye el número de ROIs antes o después del corte en lo que a número de fijaciones se refiere.

Sin embargo, la configuración del número de ROIs antes del corte sí tiene influencia en el porcentaje de fijaciones dentro de la ROI después del corte y después de ver la ROI. Hay que recordar que, aunque por la propia naturaleza de estas configuraciones de ROIs esta influencia en el porcentaje de fijaciones podría deberse simplemente al número de ROIs antes y/o después y no a un efecto real, tal y como se ha explicado en el apartado 5.2.1, se calcula el porcentaje relativo al porcentaje medio de fijaciones dentro de la ROI para cada configuración de ROIs, con lo cual, se compensa cualquier posible influencia *inherente* a la configuración de las ROIs.

Concretamente, hay una diferencia notable de comportamiento cuando hay dos ROIs dentro del mismo FOV antes del corte frente a cuando hay una ROI antes del corte y dos ROIs en distinto FOV: el número de fijaciones es mucho menor después del corte después de ver la ROI cuando antes del corte hay dos ROIs en el mismo FOV, es decir, favorece una conducta exploratoria. Esto se puede deber a que tras el corte se busca la ROI que *falta* de la toma de antes del corte. Esto se puede visualizar en la Figura 6.2, en el gráfico superior derecho.

También hay una influencia significativa de la configuración de las ROIs después del corte en la desviación de la trayectoria de la mirada comparada con la trayectoria de la mirada de referencia (ver Figura 6.2, gráfico inferior). Concretamente los análisis revelan que los sujetos tienden a variar más su trayectoria de la mirada para el caso de dos ROIs en distinto FOV después del corte, ya que no pueden ser visualizados las dos al mismo tiempo.

Por último, para la métrica de frames hasta alcanzar la ROI, los resultados de los análisis mostraron que tanto la configuración de ROIs antes del corte como después del corte tienen una influencia similar y significativa en dicha métrica.

6.4. Otros efectos observables

Atendiendo a la distrubución de las secuencias de estados, se pueden observar otros efectos interesantes. En concreto, se puede observar un pico de comportamiento exploratorio al comienzo de cada toma, tanto al comienzo del vídeo como justo después del corte, el cual tiende a durar de 1 a 2 segundos. Este pico de comportamiento exploratorio siempre viene seguido de un pico de atención, que vuelve a durar de 1 a 2 segundos. Este efecto se puede observar independientemente de la configuración de las ROIs y de la alineación entre tomas de los mismos. Esto sugiere que los usuarios siempre necesitan un cierto tiempo para comprender el entorno que se les presenta y estabilizar su mirada al comienzo y tras un corte. Tras ese estado transitorio, sin embargo, su trayectoria de la mirada se ve fuertemente atraída por las ROIs (ver Figura 6.3).

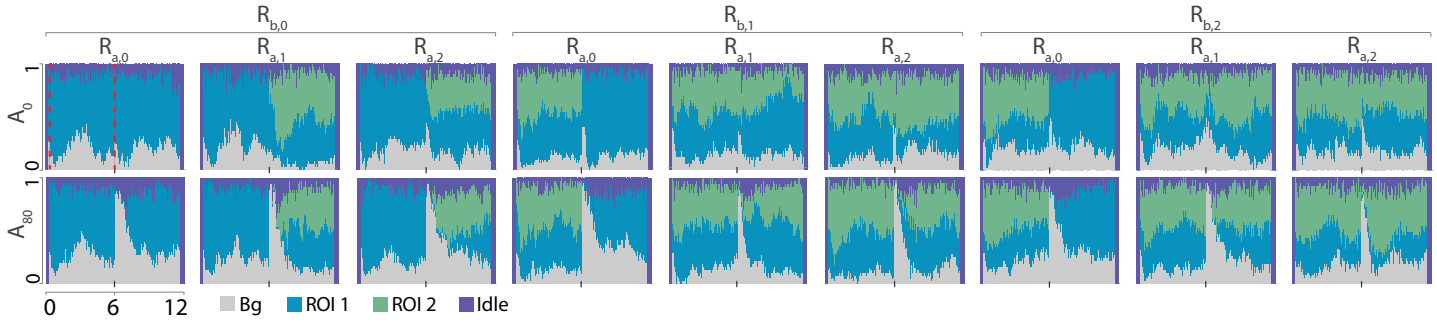


Figura 6.3: *Distribución de estados* para todas las diferentes combinaciones de R_b y R_a, y para los alineamientos de A_0 (primera fila) y A_{80} (segunda fila). Los diferentes tipos de cortes están agregados para cada una de las condiciones mencionadas. El eje de abscisas muestra el tiempo en segundos (el corte se produce en el segundo $t = 6$), mientras que el eje de ordenadas muestra el porcentaje agregado de usuarios en cada estado. Por tanto, cada gráfica muerstra el porcentaje de los usuarios en cada estado para cada instante de tiempo. Al inicio y al final de cada gráfica existe un corto período en los que solo existe el estado *idle*, esto se debe a la existencia de fotogramas negros al principio y al final de cada clip (desvanecimiento a negro), y por tanto no son relevantes para el análisis. Las líneas de puntos rojas de la gráfica de arriba a la izquierda ilustran los picos de exploración y de atención respectivamente, que se describen en el Capítulo 6, Sección 4

Por último, se han analizado más en profundidad los efectos de los dos tipos de cortes cinematográficos empleados en los experimentos para el caso particular de 1 ROI antes del corte y 1 ROI después del corte. A pesar de ser el caso más sencillo, se ha optado por analizar este caso en profundidad por ser el caso más habitual en la escasa narrativa en realidad virtual existente actualmente. La figura 6.1 ilustra la distribución de los diagramas de estados para este caso mostrando para cada tipo de corte los casos de alineamiento perfecto y desalineamiento grave.

Como se puede observar en la figura, existe un pequeño *valle* en el estado de fijación en la ROI para el caso de los cortes con continuidad en acción. Esto sugiere que este tipo concreto de cortes atraen más la atención de los sujetos y que este efecto es consistente entre alineamientos diferentes. Una posible explicación es que la continuidad en acción actúe a modo de conexión o puente entre la toma antes del corte y la toma después del corte, lo cual favorece la atención de

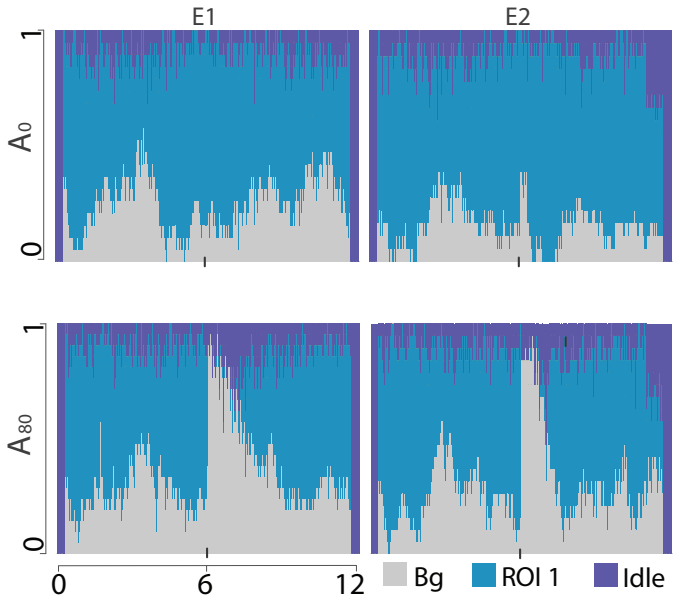


Figura 6.4: Distribución de estados para 1 ROI antes del corte y 1 ROI después del corte. Aunque las métricas no percibieron influencia alguna de los cortes cinematográficos en el comportamiento de los usuarios, la distribución de estados para estos casos sí que sugiere que los cortes con discontinuidad en acción son más difíciles de entender que los que poseen continuidad en acción

los usuarios.

7. Conclusiones y trabajo futuro

En este proyecto se ha llevado a cabo un experimento empírico sobre la influencia de distintos factores en el comportamiento de los usuarios cuando se les presenta una narrativa en un entorno de realidad virtual. Los factores que han formado parte de este estudio han sido el alineamiento entre regiones de interés antes y después del corte, el tipo de corte cinematográfico empleado y el número y disposición de regiones de interés antes y después del corte. En concreto el objetivo del análisis del comportamiento de los usuarios es el de comprobar cómo de bien perciben los sujetos la sensación de continuidad narrativa en un entorno inmersivo

Previo a este experimento, se ha llevado a cabo otro experimento de menor alcance que, bajo un enfoque cognitivo y apoyándose en la teoría de segmentación de eventos, corrobora que el proceso de segmentación de eventos y elaboración de pequeñas predicciones que el cerebro humano realiza tanto en el día a día como cuando visualiza un metraje de cinematografía tradicional, también se aplica a un entorno inmersivo entre cortes a pesar de las enormes diferencias que éste guarda con la cinematografía tradicional.

Al llevar a cabo el experimento empírico y comprobar sus resultados, se encontraron efectos interesantes en la influencia del comportamiento visual de los sujetos para los 4 factores considerados. Es importante señalar que dada la alta dimensionalidad del problema y la influencia de muchas otras variables discretas y categóricas, éste se ha reducido intencionadamente a los 4 factores señalados para favorecer una exploración sistemática del problema.

Los efectos interesantes que se encontraron indican:

- Para el factor del desalineamiento: Que un desalineamiento notable entre regiones de interés antes y después del corte puede alterar notablemente el comportamiento del sujeto incluso cuando este ya ha localizado la región de interés relevante después del corte. Es decir, que desalineamientos más grandes favorecen un comportamiento más exploratorio del nuevo entorno que le es presentado al sujeto tras el corte.
- Para el factor del número y disposición de regiones de interés antes y después del corte: Cuando hay dos regiones de interés antes de un corte y se pasa a una sola región de interés después del corte, esto propicia un comportamiento más exploratorio del entorno, posiblemente en búsqueda de la región de interés faltante.
- Para el factor de los tipos de corte empleados: las métricas no revelaron ningún tipo de información relevante. Sin embargo, analizando la distribución de las secuencias de estados de los sujetos, sí que se percibe que las regiones de interés atraen más atención ante un tipo de corte con continuidad en acción.

- Adicionalmente, analizando la distribuición de las secuencias de estados de los sujetos, se ha encontrado que existe un pico de comportamiento exploratorio en los 2 primeros segundos del vídeo y los 2 primeros segundos después del corte, lo cual sugiere que se requiere un cierto tiempo de adaptación ante los cambios del entorno siempre que hay un corte.

Como en todos los proyectos de similares características, estos resultados solo son aplicables para estímulos similares a los escogidos. Se ha escogido el formato de los vídeos 360º por varios motivos: para tener más control sobre dichos estímulos, para facilitar y permitir un análisis más sistemático de los datos, y para mostrar acciones simples en los estímulos. Es por ello que algunos resultados podrían no resultar aplicables para otras condiciones distintas a las de este proyecto. Sin embargo, el estudio de los mecanismos cognitivos del ser humano en conjunción con las técnicas cinematográficas existentes proporcionan una base firme para continuar sobre esta línea de investigación en este nuevo medio.

Una de las principales líneas de trabajo futuro podría ser el experimentar con más variables para determinar la influencia de cada una de ellas en la atención en narrativa de realidad virtual. Estas variables pueden ser otros tipos de cortes cinematográficos, vídeos más largos, contenido visual más complejo o el sonido como guía de los puntos de interés.

Personalmente, este proyecto me ha resultado muy interesante y me ha enseñado muchas cosas. He aprendido cómo funciona un grupo de investigación y cómo abordan un problema real: desde el planteamiento del problema hasta la búsqueda de soluciones en todo tipo de campos y el desarrollo e implementación de los experimentos. Todo el desarrollo del proyecto ha sido un constante e intenso aprendizaje de todo lo que rodea al mundo de la investigación, que para mi era totalmente nuevo; y aún ahora, después de haber terminado el proyecto, siento que me falta mucho por aprender.

Bibliografía

[1] Christopher A Kurby and Jeffrey M Zacks. Segmentation in the perception and memory of events. *Trends in cognitive sciences*, 12(2):72–79, 2008.

[2] Jeremy R. Reynolds, Jeffrey M. Zacks, and Todd S. Braver. A computational model of event segmentation from perceptual prediction. *Cognitive Science*, 31(4):613–643, 2007.

[3] Jeffrey M Zacks and Khena M Swallow. Event segmentation. *Current directions in psychological science*, 14(2):80–84, 2007.

[4] Jeffrey M Zacks, Nicole K Speer, Khena M Swallow, and Corey J Maley. The brain's cutting-room floor: Segmentation of narrative cinema. *Frontiers in human neuroscience*, 4:168, 2010.

[5] Arjen Van Rhijn, Robert van Liere, and Jurriaan D Mulder. An analysis of orientation prediction and filtering methods for vr/ar. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.*, pages 67–74. IEEE, 2005.

[6] Tobit Kollenberg, Alexander Neumann, Dorothe Schneider, Tessa-Karina Tews, Thomas Hermann, Helge Ritter, Angelika Dierker, and Hendrik Koesling. Visual search in the (un) real world: how head-mounted displays affect eye movements, head movements and target detection. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, pages 121–124. ACM, 2010.

[7] Thies Pfeiffer and Cem Memili. Model-based real-time visualization of realistic three-dimensional heat maps for mobile eye tracking and eye tracking in virtual reality. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, pages 95–102. ACM, 2016.

[8] Thomas Löwe, Michael Stengel, Emmy-Charlotte Förster, Steve Grogorick, and Marcus Magnor. Visualization and analysis of head movement and gaze data for immersive video in head-mounted displays. In *Proc. Workshop on Eye Tracking and Visualization (ETVIS)*, volume 1, Oct 2015.

[9] Joseph Magliano and Jeffrey M. Zacks. The impact of continuity editing in narrative film on event segmentation. *Cognitive Science*, 35(8):1489–1517, 2011.

[10] Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica K. Hodgins, and Ariel Shamir. Automatic editing of footage from multiple social cameras. *ACM Trans. Graph.*, 33(4):81:1–81:11, 2014.

[11] Pupil Labs. Documentación: https://docs.pupil-labs.com/introduction.

[12] Antoine Coutrot and Nathalie Guyader. How saliency, faces, and sound influence gaze in dynamic social scenesshort title?? *Journal of vision*, 14(8):5–5, 2014.

[13] Thomas C Kübler, Katrin Sippel, Wolfgang Fuhl, Guilherme Schievelbein, Johanna Aufreiter, Raphael Rosenberg, Wolfgang Rosenstiel, and Enkelejda Kasneci. Analysis of eye movements with eyetrace. In *International Joint Conference on Biomedical Engineering Systems and Technologies*, pages 458–471. Springer, 2015.

Anexo A. Trabajo presentado en la conferencia ACM SIGGRAPH 2017

En este anexo se adjunta una copia del trabajo sometido a la conferencia *ACM SIGGRAPH 2017*, que publica en la revista *ACM Transactions on Graphics* (Q1, 1/106).

Movie Editing and Cognitive Event Segmentation in Narrative Virtual Reality

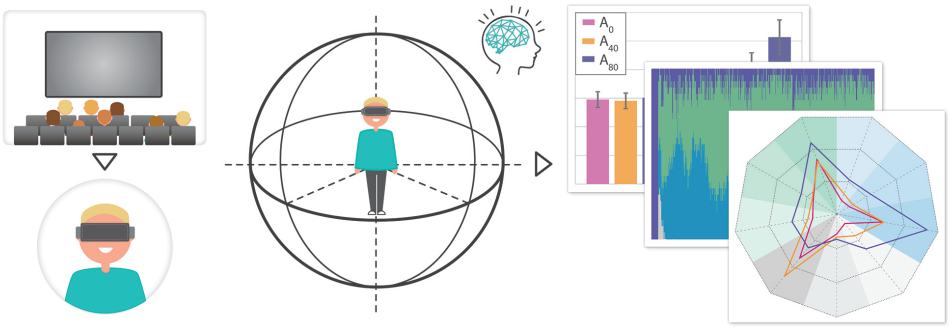


Figure 1: *Different from traditional cinematography, watching a VR movie offers viewers control over the camera. This poses many questions as to what editing techniques can be applied in this new scenario. We investigate the perception of continuity while watching edited VR content, gathering eye tracking data from many observers. We rely on recent cognitive studies, as well as well-established cinematographic techniques, to provide an in-depth analysis of such data, and to understand how different conditions affect viewers' gaze behavior.*

Abstract

Traditional cinematography has relied for over a century on a well-established set of editing rules, called continuity editing, to create a sense of situational continuity. Despite massive changes in visual content across cuts, viewers in general experience no trouble perceiving the discontinuous flow of information as a coherent set of events. However, Virtual Reality (VR) movies are intrinsically different from traditional movies in that the viewer controls the camera orientation at all times. As a consequence, common editing techniques that rely on camera orientations, zooms, etc., cannot be used. In this paper we investigate key relevant questions to understand how well traditional movie editing carries over to VR, such as: Does the perception of continuity hold across edit boundaries? Under which conditions? Do viewers' observational behavior change after the cuts? To do so, we rely on recent cognition studies and the event segmentation theory, which states that our brains segment continuous actions into a series of discrete, meaningful events. We first replicate one of these studies to assess whether the predictions of such theory can be applied to VR. On a next stage, we gather gaze data from viewers watching VR videos containing different edits with varying parameters, and provide the first systematic analysis of viewers' behavior and the perception of continuity in VR.

Keywords: Immersive environments, virtual reality

1 Introduction

Movies are made up of many different camera shots, usually taken at very different times and locations, separated by cuts. Given that the resulting flow of information is usually discontinuous in space, time, and action, while the real world is not, it is somewhat surprising that the result is perceived as a coherent sequence of events. The key to maintaining this illusion lies in how these shots are edited together, for which filmmakers rely on a system called *continuity editing* [Bordwell et al. 1997; O'Steen 2009]. Although other techniques exist to link shots together, such as the fade-out, fade-in, or dissolve, approximately 95% of editing boundaries are cuts [Cutting 2004], which directly splice two camera frames.

The goal of continuity editing in movies is then to create a sense of situational continuity (a sequence of shots perceived as a single event), or discontinuity (transitions from one event or episode to another). Professional editors have developed both a strong sense of rhythm and a solid intuition for editing together camera shots, making the cuts "invisible". For instance, spatial continuity is maintained largely by the 180° rule, stating that the camera should not cross the axis of action connecting two characters in a shot, while continuity of action is achieved by starting the action in one shot and immediately continuing it in the shot after the cut. Some authors have proposed partial theories to explain why these edited shots are perceived as a continuous event. For example, the 180° rule creates a virtual stage where the action unfolds [Bordwell et al. 1997], while mechanisms to process and track biological motion may mask an action cut [Smith and Henderson 2008].

However, the higher level cognitive processes that make continuity editing work are not yet completely understood. What is understood, though, is that a core component of spatial perception is our ability to segment a whole into parts [Biederman 1987]. Recent cognitive and neuroscience research indicates that a similar segmentation also occurs in the temporal domain, breaking up a continuous activity into a series of meaningful events. This has lead to the development of the *event segmentation* theory [Zacks and Swallow 2007; Reynolds et al. 2007; Kurby and Zacks 2008], which postulates that our brains use this discrete representation to predict the immediate course of events, and to create an internal, interconnected representation in memory. New events are registered whenever a change in action, space, or time, occur. Based on this theory, recent works have explored how continuity is perceived in movies, across different types of cuts [Zacks et al. 2010; Magliano and Zacks 2011; Cutting 2014]. Interestingly, it seems that the predictive process suggested by event segmentation theory is consistent with common practice by professional movie editors.

In this work, we investigate continuity editing for *narrative virtual reality*[1]. Virtual reality (VR) content is intrinsically different from traditional movies in that viewers now have partial control of the camera; while the position of the viewer within the scene is decided during acquisition, the orientation is not. This newly-gained freedom of users, however, renders many usual techniques, such as camera angles and zooms, ineffective when editing the movie. Neverthe-

[1] In this work we deal with 360° movies; throughout the text we will use the terms VR and 360° movies interchangeably.

less, new degrees of freedom for content creators are enabled, and fundamental questions as to what aspects of the well-established cinematographic language apply to VR should be revisited. In particular, we seek to investigate answers to the following key questions:

- Does continuity editing work in VR, i.e., is the perception of events in an edited VR movie similar to traditional cinematography?
- A common, safe belief when editing a VR movie is that the regions of interest should be aligned before and after the cut. Is this the only possible option? What are the consequences of introducing a misalignment across the cut boundaries?
- Are certain types of discontinuities (cuts) in VR favored over others? Do they affect viewer behavior differently?

We use a head mounted display, equipped with an eye tracker, and gather behavioral data of users viewing different VR videos containing different types of edits, and with varying parameters. We first perform a study to analyze whether the connections between traditional cinematography and cognitive event segmentation apply to immersive VR (Sec. 4). To this end, we replicate a recent cognitive experiment, previously carried out using traditional cinematographic footage [Magliano and Zacks 2011], using instead a VR movie. Our results show similar trends in the data, suggesting that the same key cognitive mechanisms come into play, with an overall perception of continuity across edit boundaries.

We further analyze continuity editing for VR, exploring a large parameter space that includes the type of edit from the cognitive point of view of event segmentation, the number and position of regions of interest before and after the cut, or their relative alignment across the cut boundary (Sec. 5). We propose and analyze a series of metrics that allow us to assess viewer behavior in the different conditions tested (Sec. 6). We make a series of relevant findings, including the following:

- Our data suggests that all types of edits tested are equally well understood in terms of continuity
- The misalignment at the edits introduces a delay in the time it takes viewers to fixate on a new region of interest. This delay seems to follow an exponential trend with respect to the amount of misalignment.
- The viewers' gaze behavior varies with this misalignment: larger misalignments favor a more exploratory behavior, even *after* viewers have fixated on a new region of interest.

We believe our work is the first to empirically test the connections between continuity editing, cognition, and narrative VR, as well as to look into the problem of editing in VR, in a systematic manner. This will help understand and provide guidelines for content creation and content editing in cinematic VR. We will make eye tracking data, videos, and code publicly available for researchers to build upon it.

2 Related Work

Tools for editing Creating a sequence of shots from raw footage while maintaining visual continuity is hard, especially for novice users [Davis et al. 2013]. Automatic cinematography for 3D environments was proposed by He at al. [1996], encoding film *idioms* as hierarchical finite state machines, while Christianson et al. [1996] proposed a declarative camera control system. Many other different tools have been devised to help in the editing process, usually leveraging the particular characteristics of specific domains such as 3D animations [Galvane et al. 2015], interview videos [Berthouzoz et al. 2012], narrated videos [Truong et al. 2016], classroom lectures [Heck et al. 2007], group meetings [Ranjan et al. 2008], egocentric footage [Lu and Grauman 2013], or multiple social cameras [Arev et al. 2014]. Jain et al. [2014] proposed a gaze-driven, re-editing system for retargeting video to different displays. More recently, Wu and Christie [2015] created a language to define camera framing and shot sequencing. Other methods focusing on camera placement and planning can be found in [Christie et al. 2005]. All these tools have been designed for traditional, two-dimensional viewing experiences, where the spectator sits passively in front of a screen. In contrast, our goal is to analyze continuity editing for narrative virtual reality.

Continuity and cognition Several works have analyzed the effects of edits or cuts from a computer vision perspective (e.g., [Carroll and Bever 1976; Hochberg and Brooks 2006; Smith and Henderson 2008]). Closer to our approach, a few works have analyzed the perception of continuity from a cognitive science point of view. For instance, Cohn's analyses of comic strips [2013] suggests that viewers can build links between frames while maintaining a global sense of the narrative; however, rearranging elements can quickly lead to confusion. Some researchers argue that our perception of reality is a very flexible process, and this flexibility allows us to adapt and perceive edited film as a continuous story [Anderson 1996; Cutting 2004]. Smith [2012] performed an empirical study to understand how continuity editing aligns with our perceptual abilities, identifying the role of visual attention in the perception of continuity between edits. In our work, we explore the recent theory of *event segmentation* [Zacks and Swallow 2007; Reynolds et al. 2007; Kurby and Zacks 2008; Zacks 2010], and analyze its connections with continuity editing for VR.

3 Background on event segmentation

We present here a brief summary of the event segmentation theory, and refer the reader to the original publications for a more thorough explanation [Zacks and Swallow 2007; Reynolds et al. 2007; Kurby and Zacks 2008; Zacks 2010]. Recent research suggests that event segmentation is an *automatic* key component of our perceptual processing, reducing a continuous flow of activity into a hierarchical, discrete set of events. The advantages of this strategy are twofold: First, it is very efficient in terms of internal representation and memory. Second, it provides a much easier way to think about events in relation to one another. It can be seen as the time equivalent to the well-known spatial segmentation in vision, where we segment an object (e.g., a car) into many components such as wheels, chassis, engine, etc.

This discrete mental representation is used as a basis for predicting the immediate course of events: a person walking down the street will continue to do so, or somebody will answer a question when asked. When these predictions are violated, it is an indication of a new (discrete) event; in other words, it seems that unexpected changes lead to the perception of an event boundary. More precisely, the event segmentation theory assumes that new events are registered when *changes in action, space, or time*, occur; when this happens, the mechanisms of event segmentation update the mental representation of the event, storing the old one in long-term memory.

This event segmentation theory has recently been tested in the context of film understanding. Some experiments have even recorded brain activity with functional magnetic resonance imaging (fMRI) while watching a movie, and showed that many regions in the cortex underwent substantial changes in response to the situational discontinuities (unexpected changes) introduced by some movie cuts [Zacks et al. 2010; Magliano and Zacks 2011]. An interesting observation

follows: *the predictive process suggested by event segmentation theory is consistent with common practice by professional movie editors*, who place cuts to support or break the expectations of event continuity by the viewers [Bordwell et al. 1997]. When a cut introduces a major change, the brain does not try to explain the perceived discontinuity; instead, it adapts to the change, creates a new mental representation, and begins populating it with details [Magliano and Zacks 2011]. This automatic mechanism might be a key process to explain why continuity editing works. The next section explores whether this connection between even segmentation and continuity editing studied in traditional cinema carries over to VR movies, a key question before we can dive into a more detailed investigation. Note that, in the following, we use the term *edit* to refer to a discontinuity between two shots, while *cut* refers to the actual cinematographic implementation (match-on-action, jump cut, etc.) of the edit.

4 Does continuity editing work in VR?

As we have seen in Sec. 3, there is considerable evidence that continuity editing performed in traditional movies may be related to how our brains process events and situational changes, and that this may be the cause why continuity editing has been so successful in conveying the narrative. Therefore, before we analyze specific aspects related to editing in VR movies, we first want to assert that continuity editing applies to VR scenarios. For this purpose, we check whether the connections between event segmentation and edits, which have been identified and analyzed in traditional movies [Magliano and Zacks 2011] also holds in VR movies, where the viewing conditions and the perception of immersion change significantly. This is the goal of the experiment described in this section. We aim to replicate the methodology of recent cognition studies, sharing a similar goal in the contexts of event segmentation [Zacks and Swallow 1976], and film understanding [Zacks et al. 2010; Magliano and Zacks 2011]. We introduce such works and our own experiment in the following paragraphs.

Types of edits Following common practice in film editing, Magliano and Zacks [2011] define a continuity domain along the dimensions of space, time, and action. They then classify edits into three different classes, which we call here E_1, E_2, and E_3:

- E_1: edits that are discontinuous in space or time, and discontinuous in action (action discontinuities);
- E_2: edits that are discontinuous in space or time, but continuous in action (spatial/temporal discontinuities);
- E_3: edits that are continuous in space, time, and action (continuity edits).

We adopt the same taxonomy for edits in this experiment, and in the rest of the paper.

Cognition studies with traditional movie content In these works [Zacks et al. 2010; Magliano and Zacks 2011], participants watched *The Red Balloon* (a 33-minute, 1956 movie by A. Lamorisse), and were asked to segment the movie into meaningful events by pressing a button. They were asked to do this twice, once defining the *"largest units of natural and meaningful activity"* (coarse segmentation), and once defining the smallest units (fine segmentation); the order of this division was randomized between participants, who first practiced the task on a different, 2.5-minute movie. *The Red Balloon* was presented in 7-to-10-minute sections, to avoid fatigue. Previous to this task, the authors additionally identified all the locations where edits occurred in the movie, and coded each one according to the above categorization: E_1, E_2, or E_3. Based on the principles of film editing discussed in Sec. 3, action discontinuities

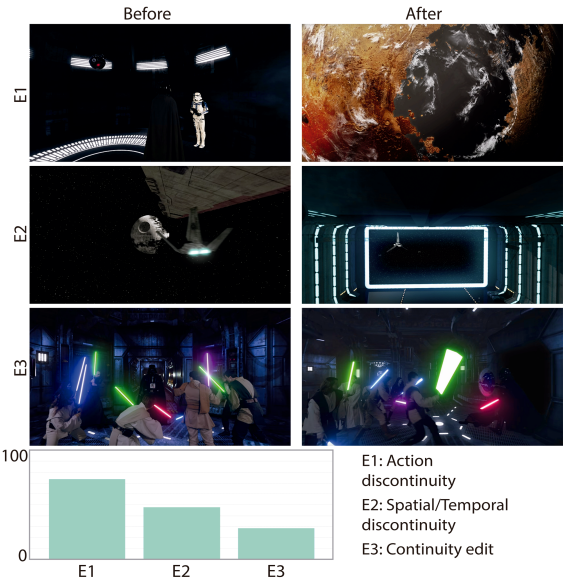


Figure 2: ***Top:*** *Representative frames of the 360° movie* Star Wars - Hunting of the Fallen*, before (left) and after (right) an edit for each of the three types of edits. (Original video property of CubeFX (http://cubefx.cz); video and images used with permission).* ***Bottom:*** *Results of our coarse segmentation test, showing the percentage of edits of each type marked as an event boundary by subjects. E_1 action discontinuities dominate event segmentation, while E_3 continuity edits maintain the perceived continuity of the event. These findings match results of similar studies in traditional cinematography, and suggest that movie editing rules and common practice can be in general applied to narrative VR as well.*

E_1 should have the largest influence on perceived discontinuities, whereas continuity edits E_3 should mostly maintain the perceived continuity. The analysis of the data discretized in five-second bins, along with fMRI information, confirmed this predicted trend.

Replication of the study with VR content We followed the same methodology in our VR study. Specifically, we asked eight participants (ages between 21 and 31, three female) to watch *Star Wars - Hunting of the Fallen*, a publicly available film[2], while indicating perceived events at coarse scale, similar to the original experiments. Participants watched the movie while seated, wearing an Oculus Rift. We chose this particular movie among several candidates since: i) like *The Red Balloon*, it is a narrative movie with a rich enough structure; ii) it lasts eight minutes, which falls within the range fixed by previous studies to avoid fatigue; iii) its average shot lasts about 40 seconds, close to the average we obtained from analyzing several 360° movies; and iv) it contains edits of all three kinds: seven action discontinuities (E_1), three spatial/temporal discontinuities (E_2), and two continuity edits (E_3). Fig. 2 (top) shows representative frames of all three types of edits.

Insights To identify the relation between the edits and the perceived event segmentation, we identified the location and type of each edit, and binned all perceived events from the test within a ± 3 second window, centered at the edit. We did not find strong correlations between the fine event segmentation and the edits in the VR movie. This is expected, considering that, different from traditional movies, the time scale of such fine perceived events is

[2]https://www.youtube.com/watch?v=SeDOoLwQQGo

about one order of magnitude smaller than the average VR shot (seconds vs. tens of seconds). We thus analyze coarse segmentation data only in the rest of the paper. Fig. 2 (bottom) shows the results, grouped by edit category. Our findings show similarities with the previous studies on traditional cinematography [Zacks et al. 2010; Magliano and Zacks 2011]. First, action discontinuities dominate event segmentation, and are therefore the strongest predictors of event boundaries. Second, continuity edits succeed in maintaining a sense of continuity of action, even across the edit boundaries. It thus appears that the key cognitive aspects of traditional movie editing that make it work so well carry over to 360° immersive narrative movies. To our knowledge, this is the first time that these findings are empirically tested in immersive 360° movies.

5 Measuring continuity in VR

After confirming in the previous section that the perception of continuity is maintained across edit boundaries in VR narrative content, we now perform a second, in-depth study to assess *how the different parameters that define an edit in VR affect the viewers' behavior after the edit takes place*. Given the high dimensionality of this space, we focus on four main parameters (or *variables of influence*), which are: The type of edit, for which we follow the cognitive taxonomy described in previous sections; the degree of misalignment of the regions of interest (ROIs) before and after the edit; and the number and location of such ROIs, both before and after the edit boundaries. In the following we describe our stimuli, variables of influence, and procedure. Additional details can be found in the supplemental material.

5.1 Stimuli

Our stimuli are created from 360° monocular videos, professionally captured and stitched by a local company. We choose to use monocular (and not stereo) footage since it is more common among existing capture devices and public repositories (e.g., YouTube 360). The videos depict four different scenarios (*Stairs, Kitchen, Living Room, Study*), with four different actions in each one, totalling sixteen videos ranging from 13 seconds to 2 minutes in length. Fig. 3 shows some representative frames in equirectangular projection. They were captured using two different rigs: a GoPro Omni (a 360-video rig consisting of six GoPro Hero4 cameras), and a Freedom360 3× rig (with three GoPro Hero4 cameras with modified Entaniya 220 lenses). Sound was recorded using a Zoom F8 recorder with wireless microphones.

From these videos, we created a total of 216 *clips*, sampling our parameter space as explained in the following subsection. Each clip is made up of two shots, separated by an edit. Shots are taken from short sequences both within and across the four different scenarios, to maximize variety. Each shot lasts six seconds, to provide enough time to the viewers to understand the actions being shown.

5.2 Variables of influence and parameter space

Type of edit We rely on the event segmentation theory, and initially consider the three different types of edits $\{E_1, E_2, E_3\}$, defined along the dimensions of space, time, and action, as introduced in Sec. 3. However, we observe that type E_3 (which essentially refers to a change of viewpoint within the same scene) is rarely used in narrative VR; we thus remove it from our conditions, and focus on the two most prominent types of edits: $\mathbf{E} = \{E_1, E_2\}$. For the actual implementation of these edits, we revise traditional cinematography techniques and analyze existing VR videos, and select the most common cuts for each type of edit: For type E_1 (discontinuous in action, and in time or space) we use jump cuts, while for E_2 (continuous in action, discontinuous in time *or* space) we use compressed time cuts, and match-on-action cuts (see e.g., [Dmytryk 1984; Chandler 2004]; please refer to the supplemental material for a brief explanation of each one). To keep a balanced number of clips for each type of edit, we include twice as many jump cuts (type E_1) as match-on-action and compressed time cuts (type E_2).

Alignment of ROIs We define the regions of interest (ROIs) as the areas in the 360° frame in which the action takes place[3]. Since the point of view of the camera cannot be controlled by the filmmaker in VR, a common practice among content creators is to simply align ROIs before and after an edit, to make sure that the viewer does not miss important information. However, the exploration of controlled (mis)alignments is interesting for the following reasons: First, the director may want to introduce some misalignment between ROIs for artistic or narrative purposes (e.g., to create tension). Second, the viewer may not be looking at the predicted ROI before the cut, thus rendering the alignment after the edit useless. Third, there might be multiple ROIs within a scene. We therefore test three different alignment conditions: (i) perfect alignment before and after the edit (i.e., 0° between ROIs); (ii) a misalignment that is just within the field of view (FOV) of the HMD; we chose 40° since it is close to the average misalignment in 360° videos found in public repositories[4]; and (iii) a misalignment that is outside the FOV; we chose 80°, since we found that larger values are very rare. We name these conditions $\mathbf{A} = \{A_0, A_{40}, A_{80}\}$.

ROI configuration Given the viewer control over the camera also makes the disposition and number of ROIs in the scene play a key role in gaze behavior. To analyzing the ROI configuration before and after the edit, we introduce two variables $\mathbf{R_b}$ and $\mathbf{R_a}$. The space of possible configurations is infinite, so to keep the task tractable we test three possibilities for each one: a single ROI ($R_{\{b|a\},0}$), two ROIs both falling within a single FOV ($R_{\{b|a\},1}$), and two ROIs not within the same FOV, i.e. more than 95° apart ($R_{\{b|a\},2}$). Examples of the three configurations are shown in Fig. 3. The possible combinations of $\mathbf{R_b}$ and $\mathbf{R_a}$ yield a total of nine conditions.

Summary This sampling leads to 2 (types of edit) × 3 (alignments) × 9 (ROI configurations) = 54 different conditions. For each one, we include four different clips, to minimize the effect of the particular scene shown, yielding our final number of 216 stimuli.

5.3 Hardware and procedure

We used an Oculus DK2 HMD equipped with a binocular eye tracker from pupil-labs[5], which records data at 120 Hz with a spatial accuracy of 1 degree. We also used a pair of headphones to reproduce stereo sound. Subjects stood up while viewing the video. A total of 49 subjects (34 male, 15 female, $\mu_{age} = 25.4$ years, $\sigma_{age} = 7.7$ years) participated in the experiment. All of them reported normal or corrected-to-normal vision. Each subject first carried out the eye tracker calibration procedure. Then, they were shown 36 stimuli from the total of 216, in random order. The randomization was such that no subject viewed two alignment conditions of the same clip, while guaranteeing that each clip was viewed by at least five people. Following Sitzmann et al. [2016], in order to ensure that the starting

[3]We manually label ROIs as continuous regions at several keyframes, creating the rest through interpolation. We define the center of each ROI as the centroid of its pixels.

[4]Our Oculus DK2 HMD has a horizontal FOV of 95°, so a 40° misalignment falls just in the periphery of the FOV.

[5]https://pupil-labs.com/

Figure 3: *Representative frames of three of the scenes depicted in our clips: Kitchen, Stairs, and Study (refer to the supplemental for full videos). From left to right, examples corresponding to the following region of interest (ROI) configurations: 1 ROI, 2 ROIs in the same FOV, and 2 ROIs in different FOV. For clarity, ROIs are marked by a blue box.*

condition was the same for all subjects, a gray environment with a small red box was displayed between clips; users had to find it and align their head direction with it, which would launch a new clip after 500 ms. The Unity game engine was used to show the videos, and to record head orientation on the same computer, while eye tracking data was recorded on a second computer. After viewing the clips the experimenter did a debriefing session with the subject. The total time per experiment was around 15 minutes. From the raw gathered data, we performed outlier rejection and then computed *scanpaths*, defined as a temporal sequence containing one gaze sample per frame. More details on these aspects (debriefing, outlier rejection, gaze data processing) can be found in the supplemental. From this data, we define, compute and analyze a series of metrics, as described in the next section.

6 How do edits in VR affect gaze behavior?

To obtain meaningful data about viewers' gaze behavior across event boundaries in VR, we first gather additional *baseline data* to compare against. We make the assumption that the higher the gaze similarity between the edited clips and the corresponding (unedited) baseline videos, the higher the perception of continuity; this assumption is similar to previous works analyzing gaze to assess the impact of retargeting and editing operations in images and video [Castillo et al. 2011; Jain et al. 2014]. In the following, we first describe how this baseline data is obtained, then introduce our continuity metrics, and describe the results of our analysis.

6.1 Baseline data

In order to capture baseline eye tracking data from the unedited videos, we gathered ten new subjects (nine male, one female, μ_{age} = 28.1 years, σ_{age}= 5.2 years) and collected head orientation and gaze data following the procedure described in Sec. 5.3. Videos were watched in random order. We compute the baseline scanpaths for each video from the obtained gaze data as the mean scanpath across users. We show in Fig. 4 the mean scanpath corresponding to one of our videos: we display the temporal evolution of the longitudinal gaze position (0°-360°), and it shows how viewers' attention is driven towards the ROI moving across the scene.

To ensure that this data can be used as baseline for our subsequent analyses, we need to ascertain the congruency between subjects. To do so, we rely on a *receiver operating characteristic* curve (ROC) metric, which provides a measure of the *Inter Observer Congruency* (IOC) [Le Meur et al. 2011] over time. First, we aggregate all the users' fixations[6] in two-second windows, and convolve them with a 2D Gaussian of $\sigma = 1$ degree of visual angle [Le Meur and Baccino 2013], yielding a saliency map for each time window. The

[6]Please refer to the supplemental for a description of how fixations are computed from gaze data.

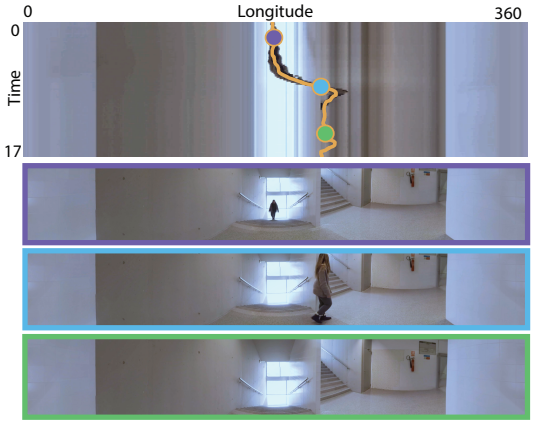


Figure 4: *Subjects mean scanpath for one example video. We plot one scanline per frame of the video, the x-axis showing longitudinal position (0°-360°), and the y-axis time. Superimposed (orange line) we plot the scanpath, showing the temporal evolution of the longitudinal position of the gaze. We also plot the full frame (equirectangular projection) at three key instants that correspond to the three marked temporal instants. Viewers' gaze is clearly directed by the movement of the ROI along time.*

corresponding ROC curve is then obtained using a one-against-all approach by leaving out the i_{th} subject: we compute, for each saliency map, the $k\%$ most salient regions, and then calculate the percentage of fixations of the i_{th} subject that fall within those regions. This process is performed for a set of thresholds $k = 0\%..100\%$, and the resulting points define each curve. Additionally, we compute the *Area Under the Curve* (AUC) for each window, which provides an easier interpretation of the evolution of the IOC along time (Fig. 5, right). The AUC takes values between 0 (incongruity between users) and 100 (complete congruency). As displayed in Fig. 5, the congruency between subjects remains very high along time. On the left of the figure, the IOC rapidly reaches a value of 1 with $k = 2\%$ most salient regions, and remains constant for increasing values of k. On the right, the same interpretation from an AUC perspective: all the viewer's fixations fall on average within the 2% regions considered most salient by the rest of the viewers, yielding a very high AUC. This indicates that all the viewers consistently considered the same regions salient. Please, refer to the supplemental material for the results for all our videos.

6.2 Metrics

Measuring the perceived continuity across edit boundaries in an objective manner is not a simple task, since no predefined metrics

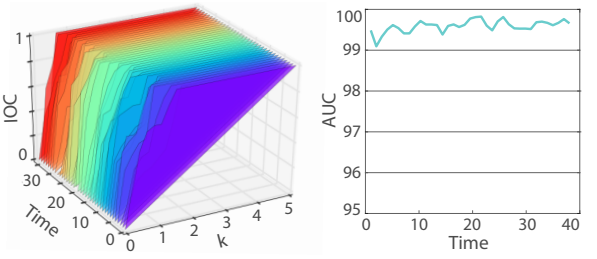


Figure 5: *Left: Inter Observer Congruency (IOC) for one of our videos. We compute a ROC curve for each second of the video. Right: Temporal evolution of the Area Under the Curve (AUC) calculated for each of the ROC curves. The high values of the IOC and AUC indices indicate that all the viewers consistently considered the same regions salient (refer to the main text for details).*

exist. We describe here our four different metrics used to analyze gaze behavior after an edit. In addition, to further look for underlying patterns in the users' behavior that our metrics may not capture, we also introduce a *state sequence* analysis.

Frames to reach a ROI (*framesToROI*) This is the simplest of our metrics, simply indicating the number of frames after the occurence of the edit before the observer fixated on a ROI. It is indicative of the time taken to converge again to the main action(s) after the edit.

Percentage of total fixations inside the ROI (*percFixInside*) This percentage is computed after fixating on a ROI after the edit. It is thus independent of *framesToROI*. Different configurations of the ROIs may imply, by nature, different number of fixations inside the ROI. To compensate for this, we compute *percFixInside* relative to the average percentage of fixations inside a ROI, for each ROI configuration, *before* the edit. This metric is indicative of the interest of the viewer in the ROI(s).

Scanpath error (*scanpathError*) We compute the RMSE of each scanpath with respect to the corresponding baseline scanpath (see Sec. 6.1). This metric indicates how gaze behavior is altered by the edit; again, we compute this metric after fixating on a ROI after the edit, to make it independent of *framesToROI*.

Number of fixations (*nFix*) We compute the ratio between the number of fixations, and the total number of gaze samples after the edit after fixating on a ROI; this way, we eliminate the possible increase in saccades while searching for the ROI after the edit. This metric is therefore indicative of how many fixations and saccades the subject performs. A low value corresponds to a higher quantity of saccades, which in turn suggests a more exploratory behavior, fixating less on any particular region or action.

State sequences We classify users' fixations along time in four different states, corresponding to the ROIs (each clip having one, or two), the background, and a so-called idle state where saccadic eye movements take place and no fixations are recorded. With this classification we are able to describe users' behavior as a state sequence, observing the succession of states with time, as well as the time spent in each of them. In particular, we use a *state distribution analysis* to represent the general pattern of state sequences for each condition, which provides an aggregated view of the frequency of each state for each time interval. We use the R library TraMineR [Gabadinho et al. 2011] for this analysis.

6.3 Analysis

Since we cannot assume that our observations are independent, we employ multilevel modeling [Browne and Rasbash 2004; Raudensbush and Bryk 2002] in our analysis, which is well-suited for grouped or related data like ours. Multilevel modeling allows the specification of random effects among the predictors, i.e., it contemplates the possibility that the model might differ for different values of these random effects. In our case, the random effect is the particular subject viewing the stimuli, for which we considered a random intercept.

We include in the regression all four factors ($\mathbf{A}$, $\mathbf{E}$, $\mathbf{R_b}$ and $\mathbf{R_a}$), as well as the first-order interactions between them. Since we have categorical variables among our predictors, we recode them to dummy binary variables for the regression. For two of our metrics (*percFixInside* and *nFix*), the effect of the subject was significant ($p = 0.002$ and $p = 0.005$, respectively, in Wald's test), indicating that we cannot treat the samples as independent; we therefore report significance values given by multilevel modeling. For the other two metrics (*framesToROI* and *scanpathError*), the effect of the subject was found to be non-significant ($p = 0.201$ and $p = 0.046$, respectively). Therefore, samples can be considered independent, and we perform factorial ANOVA, together with Bonferroni post hoc analyses to further look for significant effects in our data. Throughout the analysis we use a significance level of 0.01.

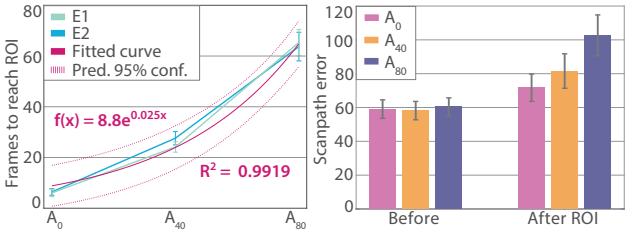


Figure 6: ***Left:*** *Average* framesToROI *for each alignment. The green and blue curves show average data for the two types of edit (E_1 and E_2, respectively). We also show a fit to an exponential function, with the associated 95% confidence interval.* ***Right:*** *Mean RMSE with respect to the baseline before the edit, and after the edit after seeing the ROI (*scanpathError*) for the different alignment conditions tested. In both plots, error bars show a 95% confidence interval for the mean.*

Influence of alignment A The first thing we observe is that there is a clear effect of the alignment factor on the four dependent variables (metrics) under study. In the case of the *framesToROI* ($F(2, 787) = 198.059, p < 0.001$), the Bonferroni post hoc further shows a significant difference ($p < 0.001$) between all three levels (A_0, A_{40} and A_{80}). As expected, the further away the ROI is, the longer it takes viewers to find it. Interestingly, the metric suggests an exponential trend with the degrees of misalignment. This is shown in Fig. 6 (left), which includes the goodness of fit, and the 95% confidence interval. Fig. 7 also illustrates this, with strong peaks and larger tails of background fixations after the edit (t = 6 secs.) for A_{80} (bottom row) than A_0 (top row).

Our *scanpathError* metric ($F(2, 787) = 14.511, p < 0.001$) allows us to dig deeper into this finding, showing in the post hoc analyses that there is no significant difference between A_0 and A_{40} ($p = 0.277$), while A_{80} is significantly different to both of them ($p \leq 0.001$ in both cases). This is shown in Fig. 6 (right), comparing directly with the equivalent values before the edit (where, as expected, no significant difference was found). A similar trend

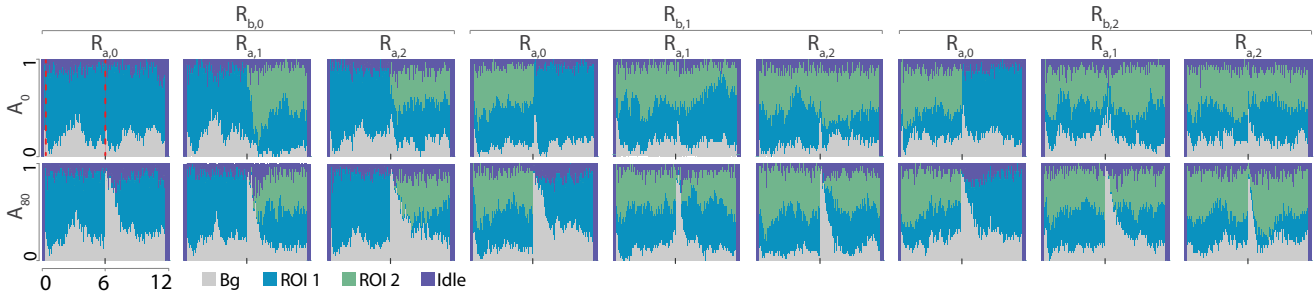


Figure 7: State distribution *for all the different combinations of* $\mathbf{R_b}$ *and* $\mathbf{R_a}$*, and for alignments* A_0 *(first row) and* A_{80} *(second row). The different types of edits* E *are aggregated in each of the aforementioned conditions. The abscissae show time in seconds (the edit takes place at* $t = 6$*), while the ordinates show aggregated percentage of users. Each plot shows the percentage of users in each state at each time instant. The short Idle periods at the beginning and the end are due to black frames before and after the movie, and thus are not significant to our analysis. The two red lines on the top-left image illustrate the exploration and attention peaks, respectively, reported in the paper.*

can be seen in *percFixInside*: A_{80} is significantly different to A_0 ($p < 0.001$), but A_{40} is not ($p = 0.138$).

This effect seems to indicate that the large misalignment alters viewer behavior not only in the time it takes to fixate on the ROI, but also *after* it is found. A closer look reveals that the same significant difference holds for *nFix*: the number of fixations is significantly lower ($p = 0.003$) for A_{80} compared to A_0, but this is not the case for A_{40} ($p = 0.954$). This, also shown in Fig. 8 (top) as a radar plot, is a very interesting finding, suggesting that viewers could be more inclined to explore the scene when there is a high misalignment across the edit boundary.

Influence of type of edit E Interestingly, the type of edit (E) has no effect on the fixational behavior after the edit after fixating on the ROI ($p = 0.674$ and $p = 0.430$ for *percFixInside*, and *nFix*, respectively). The type of edit did not have a significant effect on *scanpathError* ($F(1, 787) = 0.038, p = 0.846$) either, but the interactions of the type of edit with both ROI configurations did ($p = 0.002$ in both cases). Surprisingly, the type of edit had no significant effect on the *framesToROI* either ($F(1, 787) = 1.373, p = 0.242$), as hinted in Fig. 6.

Influence of ROI configurations $\mathbf{R_b}$ and $\mathbf{R_a}$ We observe no significant influence of these factors on *nFix*, indicating that ROI configuration does not influence the exploratory behavior (how much viewers fixate, in general) of the viewers after the edit once they see one of the ROIs. Interestingly, however, $\mathbf{R_b}$ has an effect on *percFixInside*, i.e., on how much viewers fixate on the ROI(s) after the edit after fixating, compared to the total number of fixations in that time period. Note that, while different ROI configurations may imply by nature different number of fixations inside, we are compensating for this effect in the computation of *percFixInside* (Sec. 6.2). Specifically, $\mathbf{R_b}$ reveals a difference between two ROIs in the same FOV, and one ROI ($R_{b,1}$ vs. $R_{b,0}$, $p = 0.015$), but not in case of two ROIs in different FOVs ($R_{b,2}$ vs. $R_{b,0}$, $p = 0.792$). This can be seen in Fig. 8 (middle): two ROIs in the same FOV before the edit lead to less fixations on the ROI(s) after the edit. We hypothesize that this is because multiple ROIs before the edit elicit a more exploratory behavior after the edit, in search for more ROI(s) even after having fixated on one.

We also found a significant influence of the ROI configuration after the edit $\mathbf{R_a}$ on the deviation of the scanpath wrt. the baseline, *scanpathError* ($F(2, 787) = 168.569, p < 0.001$ for $\mathbf{R_a}$); meanwhile, $\mathbf{R_b}$ had no significant influence ($F(2, 787) = 1.660, p = 0.191$ for $\mathbf{R_b}$). Bonferroni post hocs show that $R_{a,2}$ is significantly different to the other two ($p < 0.001$), while $R_{a,0}$ and $R_{a,1}$ are not significantly different between them ($p = 0.804$). Fig. 8 (bottom) shows this effect: the *scanpathError* is significantly higher for $R_{a,2}$ (two ROIs in different FOVs), indicating that there is more variability in the scanpaths since the two ROIs cannot be looked at simultaneously. Finally, both $\mathbf{R_b}$ and $\mathbf{R_a}$ had also a significant effect on *framesToROI* ($F(2, 787) = 6.478, p = 0.002$ for $\mathbf{R_b}$, and $F(2, 787) = 10.300, p < 0.001$ for $\mathbf{R_a}$).

Other effects Additionally, we can observe some new effects in the state distribution sequences. In particular, we find an *exploration* peak right at the beginning of each clip, both when the video starts and right after the edit; this peak usually lasts around 1-2 seconds. It is followed by an *attention* peak, again lasting around 1-2 seconds. This effect appears regardless of the ROI configurations and the alignment, and can be observed in Fig. 7. This suggests that users require some time to understand their environment and stabilize their gaze patterns when a change of scenario occurs; after that transitory state, however, their gaze is strongly attracted to the actions being performed (the ROIs).

Last, we analyze more in depth the effect of the two types of edits (E_1 and E_2) in the particular case of ($R_{b,0}$,$R_{a,0}$) (edits from one ROI to one ROI). This is one of the simplest cases, but also one of the most relevant, since many current VR film-making strategies are commonly based on a single ROI across scenes. In Fig. 9 we show the *state distribution* for this particular case ($R_{b,0}$,$R_{a,0}$) for alignments A_0 and A_{80}, and for the two types of edits. Even though we found no significant effect of the type of edit in our metrics, the graphs suggest a difference that our metrics are not capturing. In particular, it seems that E_2 attracts more attention to the ROI after the edit than E_1, as seen in the deeper blue valley after the edit in the right column), and this effect is consistent across all alignments. A potential explanation is that the continuity in action before and after the E_2 edit acts as an anchor.

7 Discussion and Conclusions

To our knowledge, our work is the first to attempt a systematic analysis of viewer behavior and perceived continuity in narrative VR content. A systematic exploration of this topic is challenging for two main reasons: (i) the extreme high dimensionality of its parameter space; and (ii) that it involves many discrete, categorical (as opposed to interval or ordinal) variables of influence. Moreover, other basic issues need to be addressed, such as: how does one measure continuity, or viewer behavior? Which are the best metrics to use? Are our observations independent of the subjects? We have

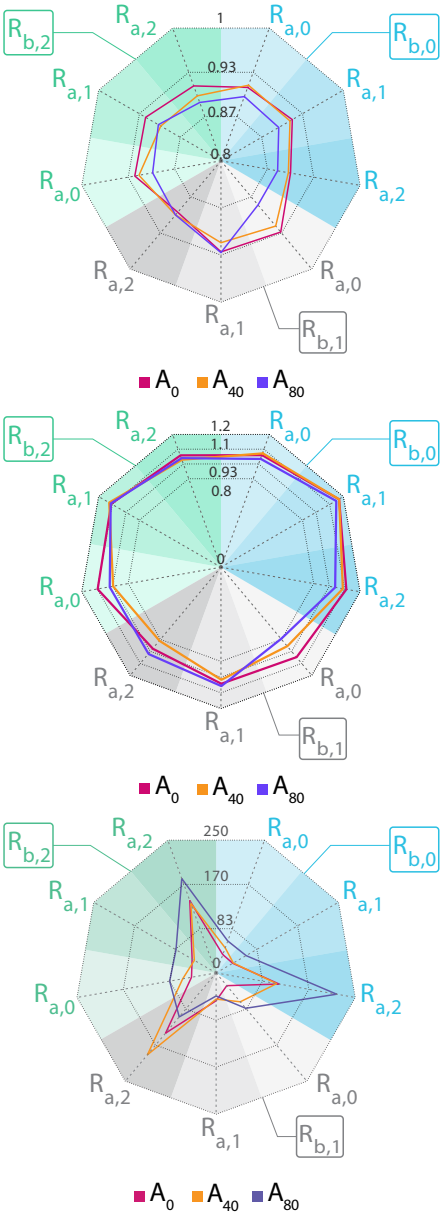


Figure 8: *Radar graphs showing variation of three of our metrics with* $\boldsymbol{A}$, $\boldsymbol{R_b}$ *and* $\boldsymbol{R_a}$. *Variation with* $\boldsymbol{E}$ *is not shown. In each graph, the three curves correspond to the three alignment conditions, as labeled in the legend. The radii of the graph correspond to the different combinations between* R_a *and* R_b. R_a *values are written at each point in the perimeter, while the large colored sectors (blue, gray and green), correspond to* R_b ($R_{b,0}$, $R_{b,1}$ *and* $R_{b,2}$ *respectively, as indicated).)* ***Top:*** *Number of fixations after the edit after fixating seeing the ROI (*nFix*), which is significantly different for* A_{80} *than for* A_{40} *and* A_0. *The scale of the radial axis is enlarged for visualization purposes.* ***Middle:*** *Value of* percFixInside *for the different conditions;* percFixInside *is significantly affected by the ROI configuration both before and after the edit (see text for details).* ***Bottom:*** *Mean RMSE with respect to the baseline after the edit after fixating on the ROI (*scanpathError*). Please see the text for details.*

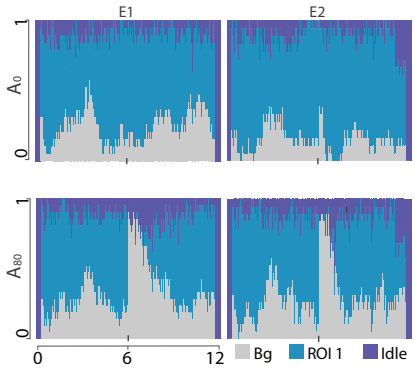


Figure 9: State distribution *for* ($R_{b,0}$,$R_{a,0}$)*, for alignments* A_0 *and* A_{80}*, and for the two different types of edits* E_1 *(left) and* E_2 *(right). Although our metrics did not capture this effect, it appears that* E_1 *edits might be harder to understand than* E_2*, as indicated by the deeper blue valley after the edit in the right column.*

relied on the event segmentation theory, which has provided us with some solid ground to carry out our research, and have analyzed previous related studies on traditional cinematography.

Our results may have direct implications in VR, informing content creators about the potential responses that certain edit configurations may elicit in the audience. For instance, for a fast-paced action movie our results suggest that ROIs should be aligned across edits, while to evoke a more exploratory behavior, misalignments are recommended. Additionally, from all the narrative 360° movies we have explored, we have found an interesting trend in the number and classification of edits: while in VR movies the great majority of edits are type E_1 (action discontinuity), they are by far the least frequent in traditional cinematography, where E_3 continuity edits are the most prominent. For example, *The Red Balloon* has 85 continuity edits, 67 spatial/temporal discontinuities, and only 18 action discontinuities. We believe this is due to the immersive nature of narrative VR, where an excessive number of continuity edits would reduce opportunities for free exploration. In the rest of the section, we summarize our main findings, and outline interesting areas of future work ahead.

Cognition and event segmentation in VR We have first replicated an existing cognitive study carried out on the *The Red Balloon* movie, and found many similarities in VR. Like in traditional cinematography, action discontinuities dominate event segmentation in VR, becoming the strongest predictors of event boundaries. Continuity edits do succeed in maintaining the perceived continuity also in VR, despite the visual discontinuity across edit boundaries. This suggests that viewers build a mental model of the shown event structure that is similar to watching a traditional movie, despite the drastically different viewing conditions.

Measuring continuity effects Our analysis has revealed several other interesting findings. Moreover, most of our reported findings have significant values of $p < 0.01$; this minimizes the risk of false positives in our conclusions.

The relation between how misaligned a ROI appears after an edit, and how long it takes viewers to fixate on it, seems to be exponential; this could be used as a rough guideline when performing edits. Even more importantly, large misalignments across edit boundaries do alter the viewers' behavior *even after they have fixated on the new ROI*. A possible interpretation is that the misalignment fosters a more exploratory behavior, and thus could be used to control attention.

Two ROIs in the same FOV before an edit seem to elicit a more exploratory behavior as well, even after having located one ROI after the edit.

Other effects not caught by our metrics can be inferred by visual inspection of the state distributions. There seems to be at exploration peak at the beginning of each clip, and a similar attention peak right after the edit, independent of the type of edit. Both suggest that users require some time to adapt to new visual content, before their gaze fixates on ROIs. Also, it appears that the ROI attracts more attention after an E_2 edit than after a type E_1, perhaps because the consistent action before and after the edit acts as an anchor.

Limitations and future work As in all studies of similar nature, our results are only strictly valid for our chosen stimuli. We have focused on short 360° videos for several reasons: to isolate simple actions, avoiding confounding factors; to gain control over the stimuli, enabling a systematic exploration of the parameter space; and to facilitate the analysis of the gathered data. Some of our findings may therefore not generalize to conditions outside our study.

Of course, many other variables and parameters can be explored in future work, such as other types of cinematographic cuts, longer movies, more complex visual content, or the influence of sound. More comprehensive subjective data may also be a valuable source of information, together with our objective gaze data. We believe that the joint study of cognitive mechanisms and cinematographic techniques provides a solid ground to carry out this research.

In summary, we believe that our work is a timely effort, since narrative VR is a fast-growing new medium still in its initial exploratory phase, with many content creators testing ways to communicate stories through it. We hope that our findings will be useful as guidelines for VR content creators, especially amateurs, across a reasonable range of situations.

References

ANDERSON, J. D. 1996. *The Reality of Illusion: An Ecological Approach to Cognitive Film Theory*. Southern Illinois University Press.

AREV, I., PARK, H. S., SHEIKH, Y., HODGINS, J. K., AND SHAMIR, A. 2014. Automatic editing of footage from multiple social cameras. *ACM Trans. Graph. 33*, 4, 81:1–81:11.

BERTHOUZOZ, F., LI, W., AND AGRAWALA, M. 2012. Tools for placing cuts and transitions in interview video. *ACM Trans. Graph. 31*, 4, 67:1–67:8.

BIEDERMAN, I. 1987. Recognition-by-components: A theory of human image understanding. *Psychological Review 94*, 115–147.

BORDWELL, D., THOMPSON, K., AND ASHTON, J. 1997. *Film art: An introduction*, vol. 7. McGraw-Hill New York.

BROWNE, W., AND RASBASH, J. 2004. Multilevel modelling. In *Handbook of data analysis*, Hardy, M. and Bryman, A. (Eds.). Sage Publications, 459–478.

CARROLL, J. M., AND BEVER, T. G. 1976. Segmentation in cinema perception. *Science 191*, 4231, 1053–1055.

CASTILLO, S., JUDD, T., AND GUTIERREZ, D. 2011. Using eye-tracking to assess different image retargeting methods. In *Symposium on Applied Perception in Graphics and Visualization (APGV)*, ACM Press.

CHANDLER, G. 2004. *Cut by cut*. Michael Wiese Productions.

CHRISTIANSON, D. B., ANDERSON, S. E., HE, L.-W., SALESIN, D. H., WELD, D. S., AND COHEN, M. F. 1996. Declarative camera control for automatic cinematography. In *AAAI/IAAI, Vol. 1*, 148–155.

CHRISTIE, M., MACHAP, R., NORMAND, J., OLIVIER, P., AND PICKERING, J. H. 2005. Virtual camera planning: A survey. In *Int. Symposium on Smart Graphics*, 40–52.

COHN, N. 2013. Visual narrative structure. *Cognitive Science 37*, 3, 413–452.

CUTTING, J. 2004. Perceiving scene in film and in the world. In *Moving image theory: ecological considerations*, J. D. Anderson and B. F. Anderson, Eds. ch. 1, 9–26.

CUTTING, J. E. 2014. Event segmentation and seven types of narrative discontinuity in popular movies. *Acta psychologica 149*, 69–77.

DAVIS, N. M., ZOOK, A., O'NEILL, B., HEADRICK, B., RIEDL, M., GROSZ, A., AND NITSCHE, M. 2013. Creativity support for novice digital filmmaking. In *Proc. ACM SIGCHI*, 651–660.

DMYTRYK, E., 1984. On film editing. an introduction to thè art of film construction.

GABADINHO, A., RITSCHARD, G., MLLER, N., AND STUDER, M. 2011. Analyzing and visualizing state sequences in R with TraMineR. *Journal of Statistical Software 40*, 1.

GALVANE, Q., RONFARD, R., LINO, C., AND CHRISTIE, M. 2015. Continuity editing for 3d animation. In *AAAI Conference on Artificial Intelligence*.

HE, L.-W., COHEN, M. F., AND SALESIN, D. H. 1996. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, 217–224.

HECK, R., WALLICK, M. N., AND GLEICHER, M. 2007. Virtual videography. *TOMCCAP 3*, 1.

HOCHBERG, J., AND BROOKS, V. 2006. Film cutting and visual momentum. *In the mind?s eye: Julian Hochberg on the perception of pictures, films, and the world*, 206–228.

JAIN, E., SHEIKH, Y., SHAMIR, A., AND HODGINS, J. 2014. Gaze-driven video re-editing. *ACM Transactions on Graphics*.

KURBY, C. A., AND ZACKS, J. M. 2008. Segmentation in the perception and memory of events. *Trends in cognitive sciences 12*, 2, 72–79.

LE MEUR, O., AND BACCINO, T. 2013. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods 45*, 1, 251–266.

LE MEUR, O., BACCINO, T., AND ROUMY, A. 2011. Prediction of the inter-observer visual congruency (iovc) and application to image ranking. In *Proc. ACM Multimedia*, ACM, 373–382.

LU, Z., AND GRAUMAN, K. 2013. Story-driven summarization for egocentric video. In *Proc. IEEE CVPR*, 2714–2721.

MAGLIANO, J., AND ZACKS, J. M. 2011. The impact of continuity editing in narrative film on event segmentation. *Cognitive Science 35*, 8, 1489–1517.

O'STEEN, B. 2009. *The Invisible Cut*. Michael Wiese Productions.

RANJAN, A., BIRNHOLTZ, J. P., AND BALAKRISHNAN, R. 2008. Improving meeting capture by applying television production

principles with audio and motion detection. In *Proc. ACM SIGCHI*, 227–236.

RAUDENSBUSH, S., AND BRYK, A. 2002. *Hierarchical Linear Models*. Sage Publications.

REYNOLDS, J. R., ZACKS, J. M., AND BRAVER, T. S. 2007. A computational model of event segmentation from perceptual prediction. *Cognitive Science 31*, 4, 613–643.

SITZMANN, V., SERRANO, A., PAVEL, A., AGRAWALA, M., GUTIERREZ, D., AND WETZSTEIN, G. 2016. Saliency in vr: How do people explore virtual environments? *arXiv preprint arXiv:1612.04335*.

SMITH, T. J., AND HENDERSON, J. M. 2008. Edit blindness: The relationship between attention and global change blindness in dynamic scenes. *Journal of Eye Movement Research 2*, 2.

SMITH, T. J. 2012. The attentional theory of cinematic continuity. *Projections 6*, 1, 1–27.

TRUONG, A., BERTHOUZOZ, F., LI, W., AND AGRAWALA, M. 2016. Quickcut: An interactive tool for editing narrated video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST 2016, Tokyo, Japan, October 16-19, 2016*, 497–507.

WU, H.-Y., AND CHRISTIE, M. 2015. Stylistic patterns for generating cinematographic sequences. In *Workshop on Intelligent Cinematography and Editing*.

ZACKS, J. M., AND SWALLOW, K. M. 1976. Foundations of attribution: The perception of ongoing behavior. vol. 1, 223–247.

ZACKS, J. M., AND SWALLOW, K. M. 2007. Event segmentation. *Current directions in psychological science 14*, 2, 80–84.

ZACKS, J. M., SPEER, N. K., SWALLOW, K. M., AND MALEY, C. J. 2010. The brain's cutting-room floor: Segmentation of narrative cinema. *Frontiers in human neuroscience 4*, 168.

ZACKS, J. M. 2010. How we organize our experience into events. *Psychological Science Agenda 24*, 4.

Anexo B. Tabla con todas las condiciones puestas a prueba en los experimentos

En la siguiente tabla se detallan todas las combinaciones de las condiciones de los clips puestas a prueba en el experimento. A modo de breve recordatorio:

Condiciones acerca del desalineamiento

- A_0 - ROIs alineadas entre tomas
- A_{40} - ROIs desalineadas 40º horizontales entre tomas
- A_{80} - ROIs desalineadas 80º horizontales entre tomas

Condiciones acerca de la configuración de ROIs antes del corte

- $R_{b,0}$ - 1 ROI
- $R_{b,1}$ - 2 ROIs dentro del mismo FOV
- $R_{b,2}$ - 2 ROIs en distinto FOV

Condiciones acerca de la configuración de ROIs después del corte

- $R_{a,0}$ - 1 ROI
- $R_{a,1}$ - 2 ROIs dentro del mismo FOV
- $R_{a,2}$ - 2 ROIs en distinto FOV

Condiciones acerca del tipo de corte cinematográfico empleado

- E_1 - Corte con discontinuidad en acción
- E_2 - Corte con continuidad en acción pero discontinuidad en espacio o tiempo

B. Tabla con todas las condiciones puestas a prueba en los experimentos

Tabla B.1: Set de condiciones puestas a prueba en los experimentos. Desde A_0 hasta A_{40}

A_0	$R_{b,0}$	$R_{a,0}$	E_1	$(A_0, R_{b,0}, R_{a,0}, E_1)$
			E_2	$(A_0, R_{b,0}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_0, R_{b,0}, R_{a,1}, E_1)$
			E_2	$(A_0, R_{b,0}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_0, R_{b,0}, R_{a,2}, E_1)$
			E_2	$(A_0, R_{b,0}, R_{a,2}, E_2)$
	$R_{b,1}$	$R_{a,0}$	E_1	$(A_0, R_{b,1}, R_{a,0}, E_1)$
			E_2	$(A_0, R_{b,1}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_0, R_{b,1}, R_{a,1}, E_1)$
			E_2	$(A_0, R_{b,1}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_0, R_{b,1}, R_{a,2}, E_1)$
			E_2	$(A_0, R_{b,1}, R_{a,2}, E_2)$
	$R_{b,2}$	$R_{a,0}$	E_1	$(A_0, R_{b,2}, R_{a,0}, E_1)$
			E_2	$(A_0, R_{b,2}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_0, R_{b,2}, R_{a,1}, E_1)$
			E_2	$(A_0, R_{b,2}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_0, R_{b,2}, R_{a,2}, E_1)$
			E_2	$(A_0, R_{b,2}, R_{a,2}, E_2)$
A_{40}	$R_{b,0}$	$R_{a,0}$	E_1	$(A_{40}, R_{b,0}, R_{a,0}, E_1)$
			E_2	$(A_{40}, R_{b,0}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{40}, R_{b,0}, R_{a,1}, E_1)$
			E_2	$(A_{40}, R_{b,0}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{40}, R_{b,0}, R_{a,2}, E_1)$
			E_2	$(A_{40}, R_{b,0}, R_{a,2}, E_2)$
	$R_{b,1}$	$R_{a,0}$	E_1	$(A_{40}, R_{b,1}, R_{a,0}, E_1)$
			E_2	$(A_{40}, R_{b,1}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{40}, R_{b,1}, R_{a,1}, E_1)$
			E_2	$(A_{40}, R_{b,1}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{40}, R_{b,1}, R_{a,2}, E_1)$
			E_2	$(A_{40}, R_{b,1}, R_{a,2}, E_2)$
	$R_{b,2}$	$R_{a,0}$	E_1	$(A_{40}, R_{b,2}, R_{a,0}, E_1)$
			E_2	$(A_{40}, R_{b,2}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{40}, R_{b,2}, R_{a,1}, E_1)$
			E_2	$(A_{40}, R_{b,2}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{40}, R_{b,2}, R_{a,2}, E_1)$
			E_2	$(A_{40}, R_{b,2}, R_{a,2}, E_2)$

Tabla B.2: Set de condiciones puestas a prueba en los experimentos para A_{80}

A_{80}	$R_{b,0}$	$R_{a,0}$	E_1	$(A_{80}, R_{b,0}, R_{a,0}, E_1)$
			E_2	$(A_{80}, R_{b,0}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{80}, R_{b,0}, R_{a,1}, E_1)$
			E_2	$(A_{80}, R_{b,0}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{80}, R_{b,0}, R_{a,2}, E_1)$
			E_2	$(A_{80}, R_{b,0}, R_{a,2}, E_2)$
	$R_{b,1}$	$R_{a,0}$	E_1	$(A_{80}, R_{b,1}, R_{a,0}, E_1)$
			E_2	$(A_{80}, R_{b,1}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{80}, R_{b,1}, R_{a,1}, E_1)$
			E_2	$(A_{80}, R_{b,1}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{80}, R_{b,1}, R_{a,2}, E_1)$
			E_2	$(A_{80}, R_{b,1}, R_{a,2}, E_2)$
	$R_{b,2}$	$R_{a,0}$	E_1	$(A_{80}, R_{b,2}, R_{a,0}, E_1)$
			E_2	$(A_{80}, R_{b,2}, R_{a,0}, E_2)$
		$R_{a,1}$	E_1	$(A_{80}, R_{b,2}, R_{a,1}, E_1)$
			E_2	$(A_{80}, R_{b,2}, R_{a,1}, E_2)$
		$R_{a,2}$	E_1	$(A_{80}, R_{b,2}, R_{a,2}, E_1)$
			E_2	$(A_{80}, R_{b,2}, R_{a,2}, E_2)$

Anexo C. Clips representativos del conjunto de todos los clips mostrados en el experimento sobre atención de los usuarios en realidad virtual

En este anexo se incluye un enlace a 9 clips representativos de todos los mostrados en el experimento sobre atención de los usuarios en realidad virtual. Los fotogramas están en baja resolución, e incluyen superpuesta la trayectoria de la mirada de diferentes usuarios codificadas en distintos colores y representadas como puntos; así como una aproximación del FOV de cada usuario representada como un cuadrado. Esta representación del FOV no tiene en cuenta la distorsión generada por la proyección equirrectangular del vídeo.

Enlace a los clips representativos: *https://goo.gl/QE8REZ*

Anexo D. Recogida y procesado de datos

D.1. Recogida de datos

En esta sección se incluye el cuestionario que los usuarios tuvieron que rellenar antes de llevar a cabo el experimento. El cuestionario se muestra en la Figura D.1 e incluye información demográfica, información sobre corrección visual si se tiene e información sobre salud ocular en general. La última sección del cuestionario añade preguntas que se les hace a los usuarios después del test.

D.2. Procesado de datos

En esta sección se incluyen más detalles sobre el procesado de los datos después de recolectar los puntos de la mirada con los eye trackers y las posiciones de la cabeza con el Oculus Rift DK2.

D.2.1. Trayectorias de la mirada

En primer lugar se procesaron las muestras de los eye trackers. Se descartaron los resultados enteros de aquellas posiciones de los ojos en las que la certeza de la posición de ambos ojos era menor que 0.6 (los valores oscilan entre 0 y 1, siendo uno certeza absoluta de la posición del ojo). Después se interpolaron las medidas de aquellas posiciones de los ojos cuya certeza de la posición de ambos ojos era menor de 0.9.

Debido a que la posición de la cabeza del Oculus Rift DK2 tiene una frecuencia de muestreo menor que la de los eye trackers, se emparejó cada medida de la posición de los ojos con la medida más cercana en el tiempo correspondiente de la posición de la cabeza del Oculus Rift DK2.

En segundo lugar, se emparejaron las medidas de las posiciones de los ojos con los fotogramas de los clips. Dado que los clips tenían una frecuencia de 60 fps y los eye trackers grababan a 120Hz, se habían grabado 2 posiciones de los ojos para cada fotograma. Se calculó la media de las 2 posiciones de los ojos para cada fotograma y se asignó dicha media a cada fotograma de los clips (Coutrot and Guyader 2014) [12].

Por último, se define una trayectoria de la mirada como la secuencia temporal resultante de dichas posiciones de los ojos en cada fotograma.

D.2.2. Detección de fijaciones

Para detectar fijaciones en los clips, se utiliza un detector de fijaciones basado en la velocidad de las posiciones de los ojos. Se considera que una posición de los ojos en un fotograma es una fijación cuando su velocidad relativa está por debajo de un cierto umbral. Se calcula este umbral para cada trayectoria de la mirada como el 20 % de la velocidad máxima, tras descartar el segundo percentil de las velocidades más altas (Kübler et al. 2015) [13].

D.2.3. Descarte de espurios

Se descartan espurios atendiendo a dos criterios. En primer lugar, se descartan trayectorias de la mirada que tengan menos del 40 % de las fijaciones antes del corte dentro de la ROI. Consideramos que en dichos casos los usuarios no estaban prestando atención o no entendieron bien lo que debían hacer. En segundo lugar, descartamos aquellas trayectorias de la mirada que difieren significativamente del comportamiento de otros usuarios. Esto se lleva a cabo siguiendo un descarte de espurios estándar conservador, tal y como se describe en las siguientes condiciones:

$$TrayectoriaMirada < (Q_1 - K_d * Q_d)$$

$$TrayectoriaMirada > (Q_3 + K_d * Q_d)$$

Donde Q_1 y Q_3 son el primer y el tercer cuartil respectivamente, $Q_d = Q_3 - Q_1$ y $K_d = 1 : 5$.

EXPERIMENT ID _____

Demographics

Age _____
Gender _____

Have you used a VR headset before? YES / NO

If YES: Do you use VR regularly (more than once a month)? YES / NO
Do you have eye strain, headaches and/or nausea in VR? YES / NO

Other information? _____

Information about visual correction

Have you had Lasik eye surgery? YES / NO
Do you have surgically implanted intraocular lenses (IOL)? YES / NO
Do you typically wear glasses or contacts? YES / NO

If YES: For near vision? YES / NO
For far vision? YES / NO
What are you currently wearing? GLASSES / CONTACTS / NEITHER

Other information? _____

Presbyopia

Presbyopia is the inability to focus on nearby objects associated with normal aging, often corrected with bifocals, progressive lenses, or monovision.

Are you aware of having a diagnosis of presbyopia? YES / NO

If YES: Do you wear monovision correction? YES / NO
If YES, which of your eyes you use for near vision? LEFT / RIGHT

If not monovision, how is your presbyopia corrected? _____

Other information? _____

Ocular Health

Are you aware of having a diagnosis of:

Amblyopia	Strabismus/lazy eye	Glaucoma	Diabetic Retinopathy
Cataracts	Light sensitivity	Nystagmus	Macular Degeneration

Other information?

_____

After the test

Did you experience visual discomfort (such as headache, eye strain...)?
Did you experience dizziness or motion sickness?
Did you see any artifacts in the videos?
Did you understand the actions being performed?
Did you at any point feel lost when trying to follow the actions?
Were the videos comfortable to watch (e.g., they did not require too much head movement)?

Figura D.1: Cuestionario previo realizado a los usuarios en los experimentos sobre atención de los usuarios en realidad virtual. La última sección se realizaba después del experimento

Anexo E. Resultados de la distribución de secuencias de estados, congruencia entre observadores (IOC) y área bajo la curva (AUC)

En el siguiente anexo se incluyen gráficas con los resultados obtenidos para la distribución de secuencias de estados de los experimentos, la congruencia entre observadores (IOC) y para el área bajo la curva (AUC).

E.1. Distribución de secuencias de estados

Las figuras E.1, E.2, E.3, E.4, E.5 E.6 muestran los resultados del análisis de secuencias de estados para diferentes condiciones.

E.2. Congruencia entre observadores (IOC) y área bajo la curva (AUC)

Las figuras E.7, E.8, E.9, E.10, E.11 E.12 muestran los resultados del análisis de la congruencia entre observadores (IOC) y área bajo la curva (AUC).

E. Resultados de la distribución de secuencias de estados, congruencia entre observadores (IOC) y área bajo la curva (AUC)

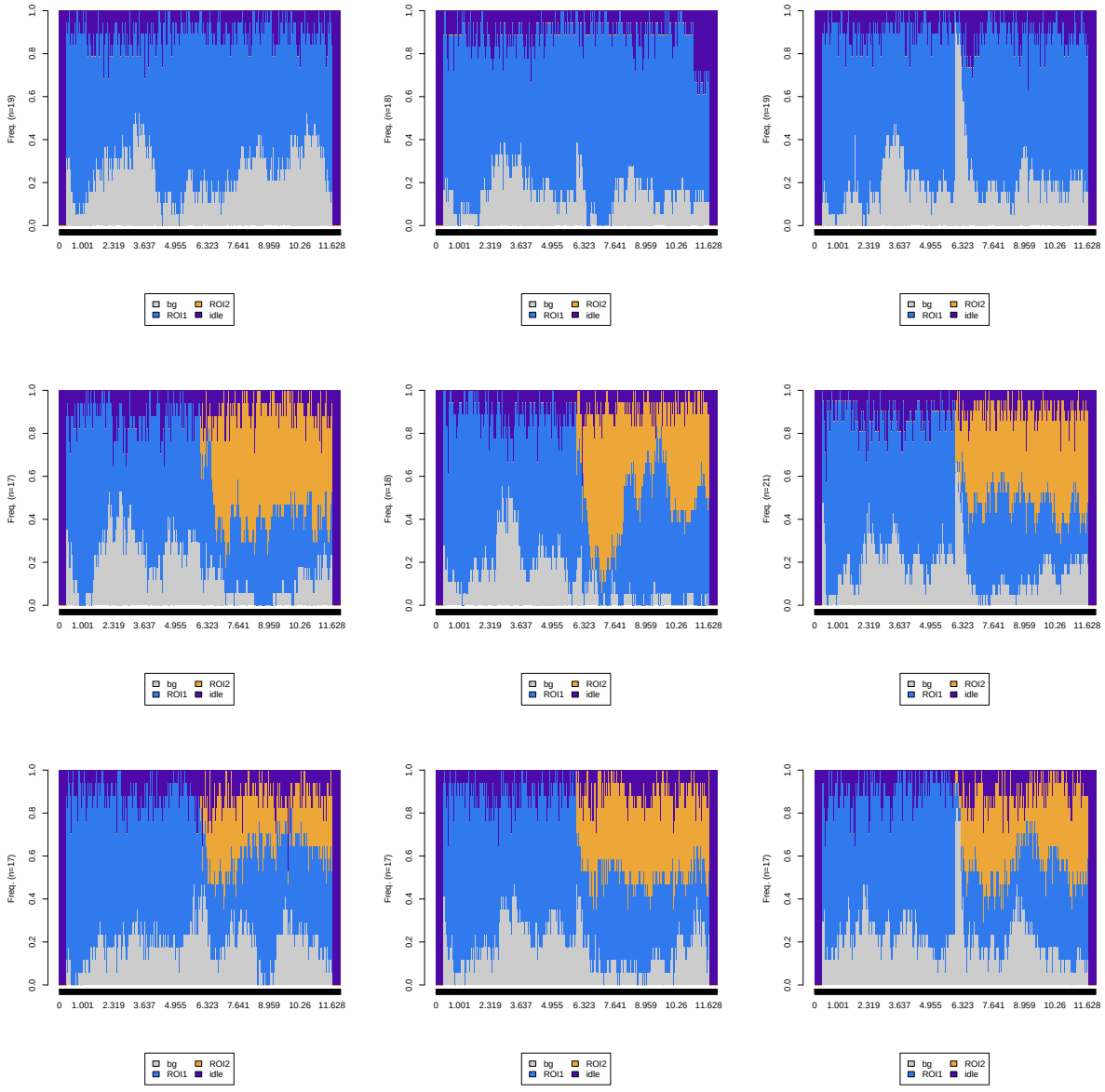


Figura E.1: Análisis de secuencias de estados para diferentes condiciones. Cada fila corresponde a una combinación diferente de R_b y R_a. De arriba a abajo: $(R_{b,0}R_{a,0})$, $(R_{b,0}R_{a,1})$, $(R_{b,0}R_{a,2})$. De izquierda a derecha: (A_0E1), (A_0E2), $(A_{40}E1)$

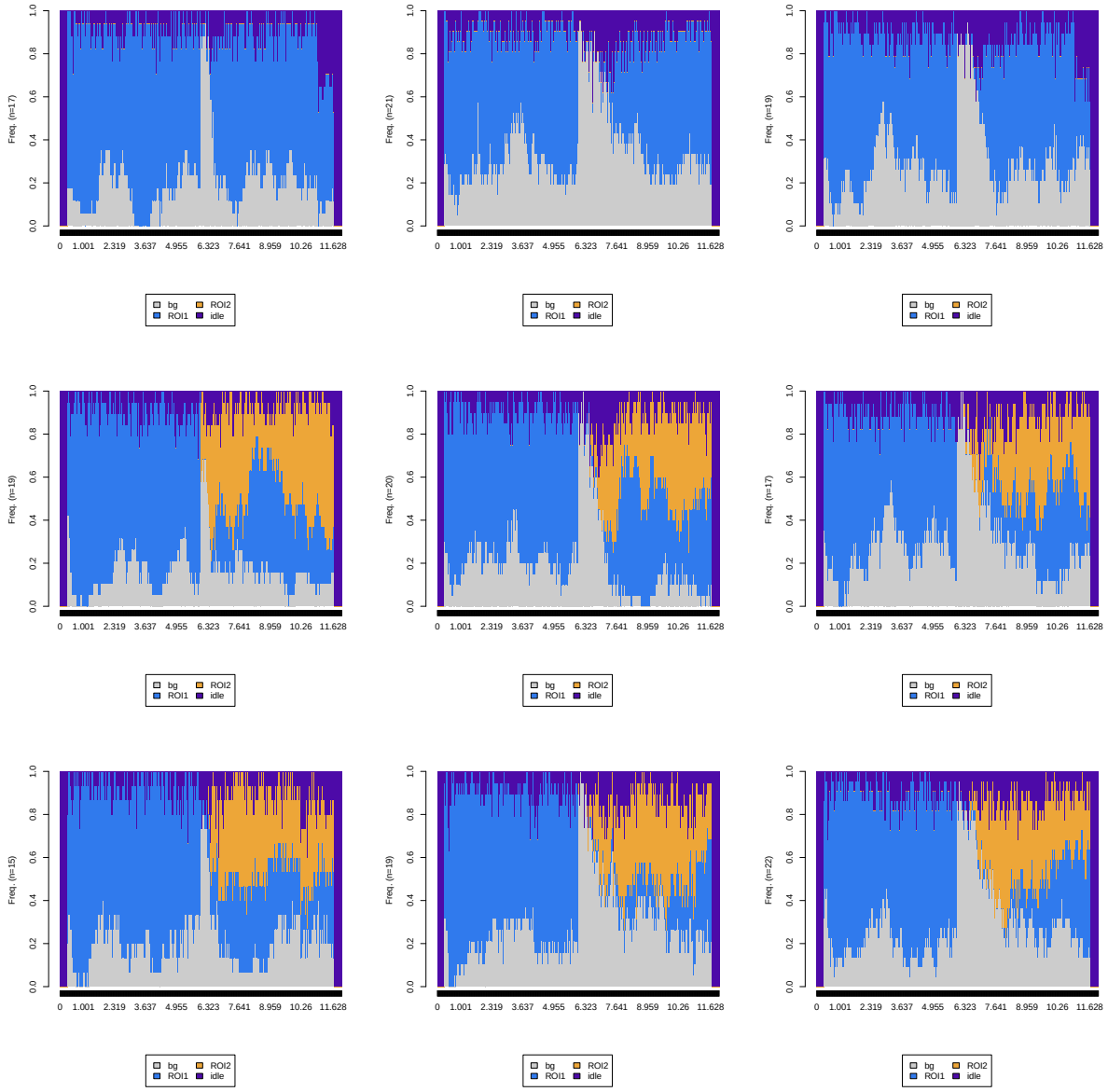


Figura E.2: Análisis de secuencias de estados para diferentes condiciones. Cada fila corresponde a una combinación diferente de R_b y R_a. De arriba a abajo: $(R_{b,0}R_{a,0})$, $(R_{b,0}R_{a,1})$, $(R_{b,0}R_{a,2})$. De izquierda a derecha: $(A_{40}E2)$, $(A_{80}E1)$, $(A_{80}E2)$.

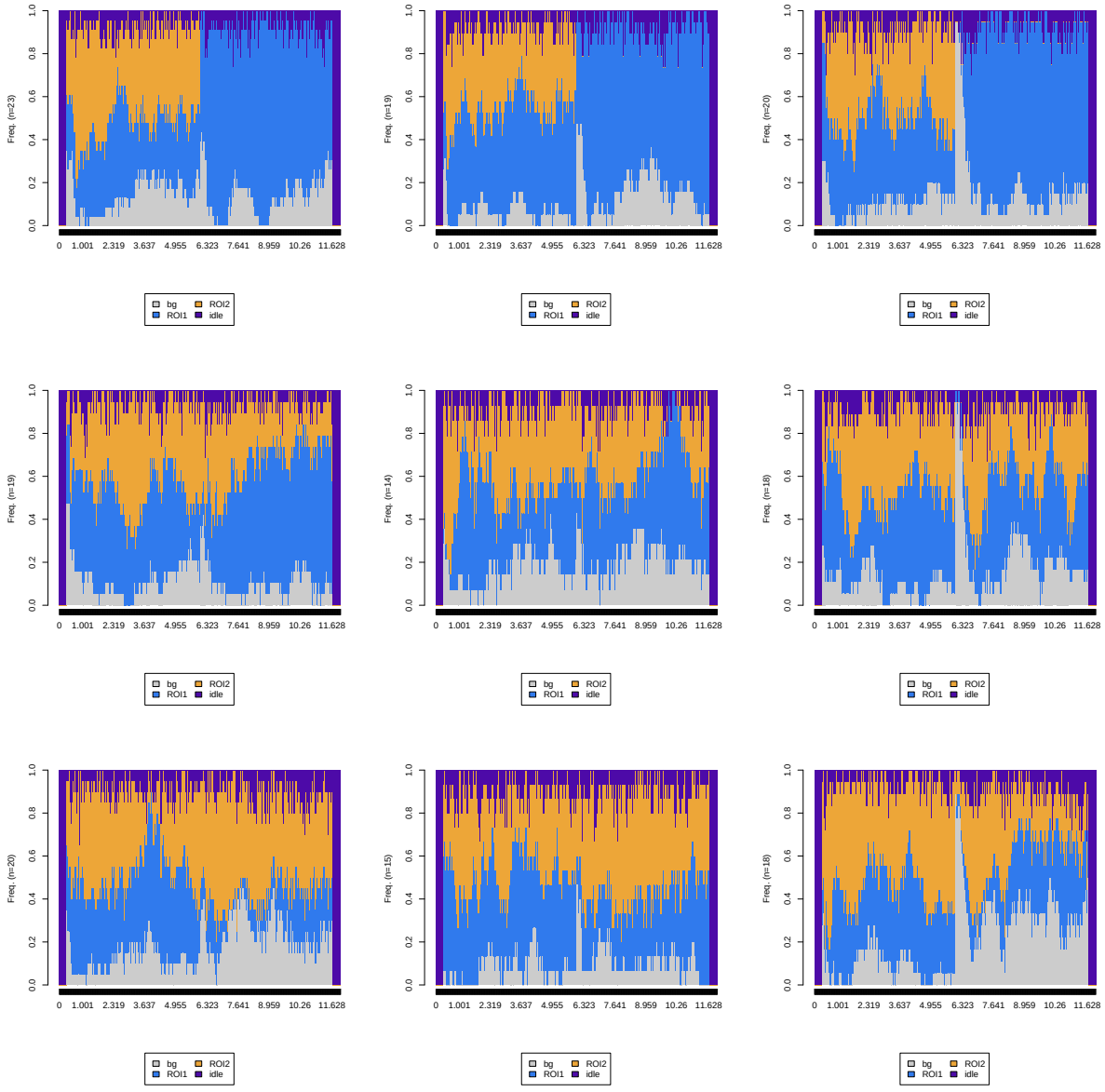


Figura E.3: Análisis de secuencias de estados para diferentes condiciones. Cada fila corresponde a una combinación diferente de R_b y R_a. De arriba a abajo: $(R_{b,1}R_{a,0})$, $(R_{b,1}R_{a,1})$, $(R_{b,1}R_{a,2})$. De izquierda a derecha: (A_0E1), (A_0E2), $(A_{40}E1)$.

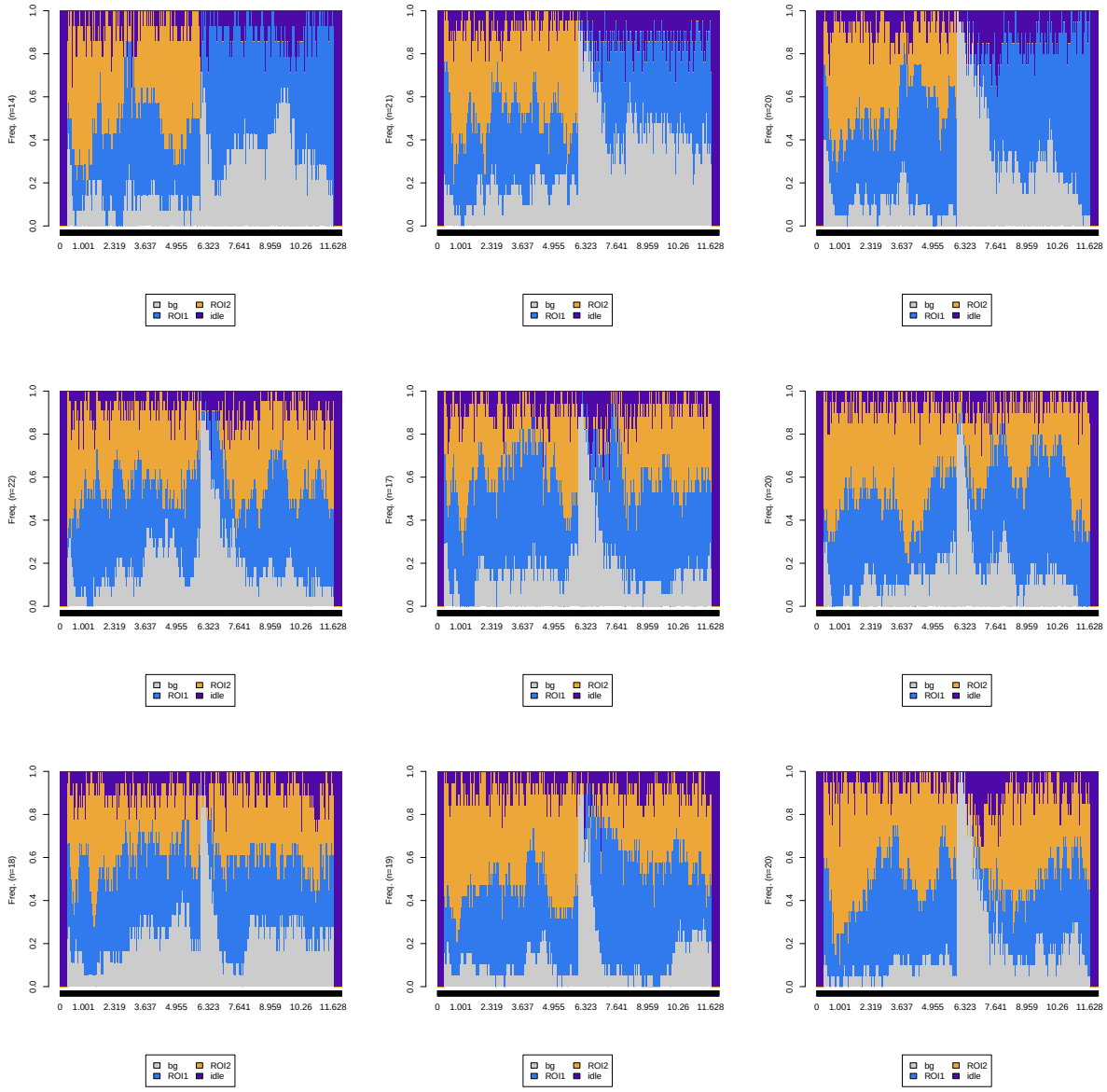


Figura E.4: Análisis de secuencias de estados para diferentes condiciones. Cada fila corresponde a una combinación diferente de R_b y R_a. De arriba a abajo: $(R_{b,1}R_{a,0})$, $(R_{b,1}R_{a,1})$, $(R_{b,1}R_{a,2})$. De izquierda a derecha: $(A_{40}E2)$, $(A_{80}E1)$, $(A_{80}E2)$.

E. Resultados de la distribución de secuencias de estados, congruencia entre observadores (IOC) y área bajo la curva (AUC)

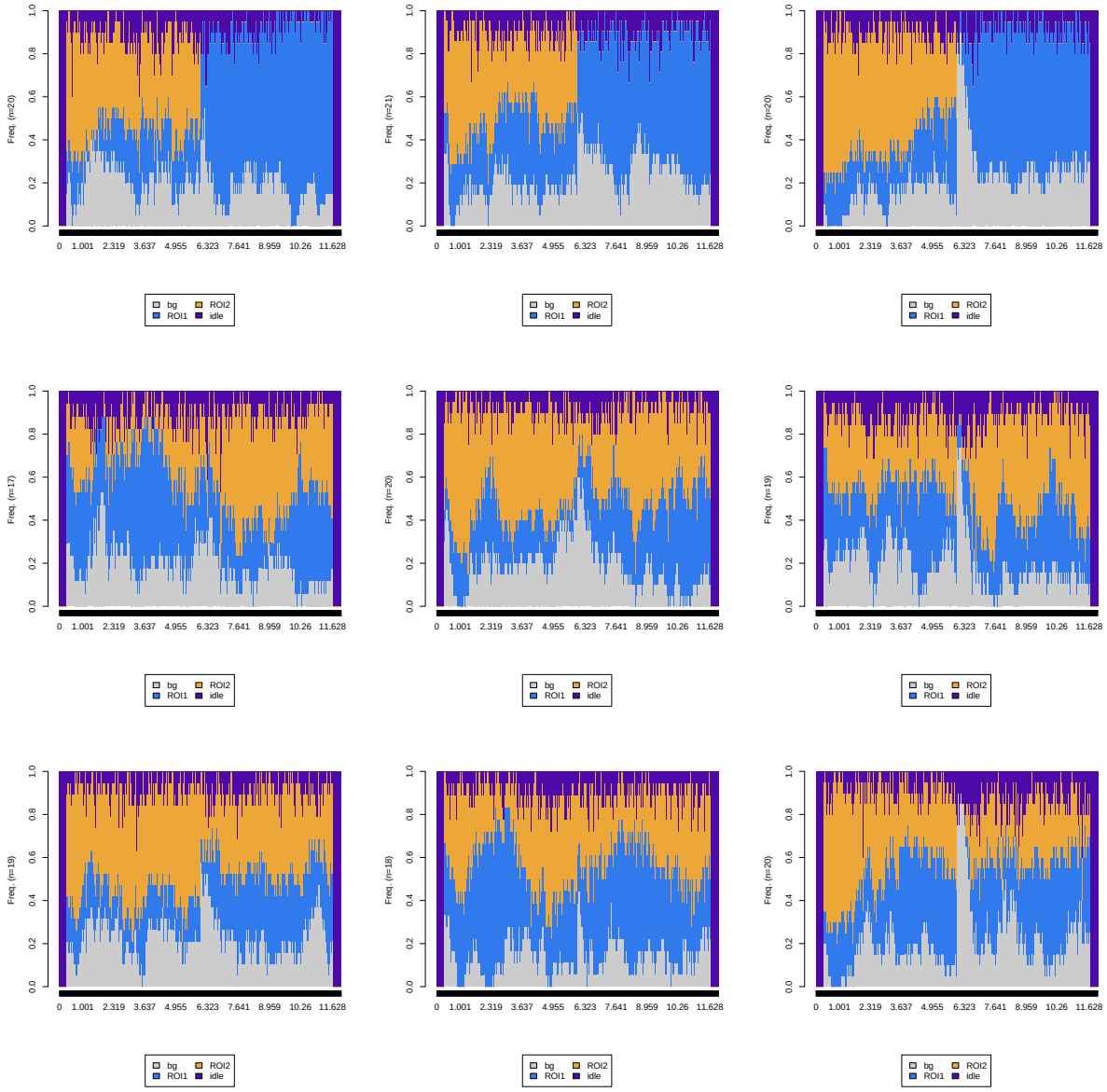


Figura E.5: Análisis de secuencias de estados para diferentes condiciones. Cada fila corresponde a una combinación diferente de R_b y R_a. De arriba a abajo: $(R_{b,2}R_{a,0})$, $(R_{b,2}R_{a,1})$, $(R_{b,2}R_{a,2})$. De izquierda a derecha: (A_0E1), (A_0E2), $(A_{40}E1)$.

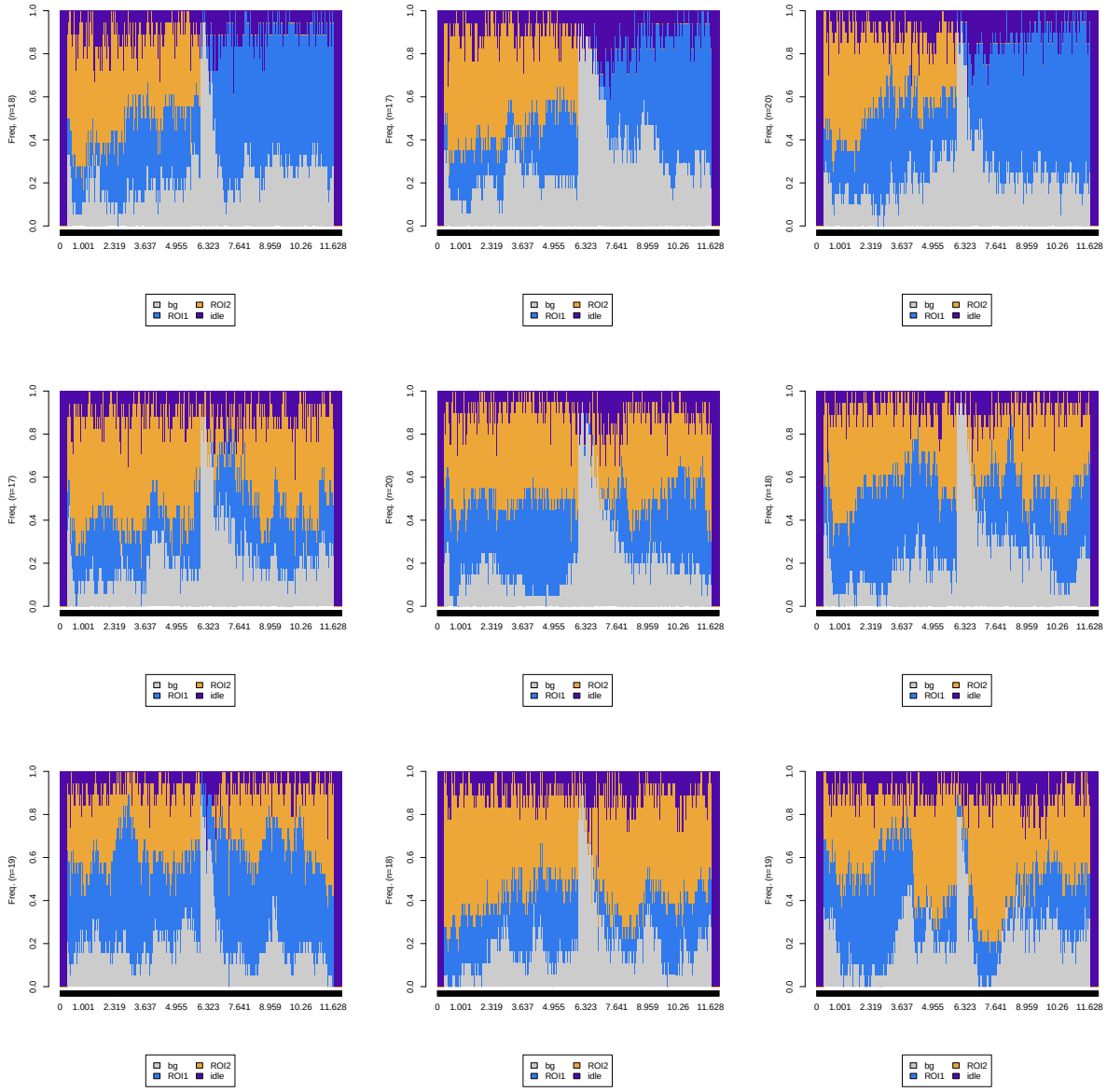


Figura E.6: Análisis de secuencias de estados para diferentes condiciones. Cada fila corresponde a una combinación diferente de R_b y R_a. De arriba a abajo: $(R_{b,2}R_{a,0})$, $(R_{b,2}R_{a,1})$, $(R_{b,2}R_{a,2})$. De izquierda a derecha: $(A_{40}E2)$, $(A_{80}E1)$, $(A_{80}E2)$.

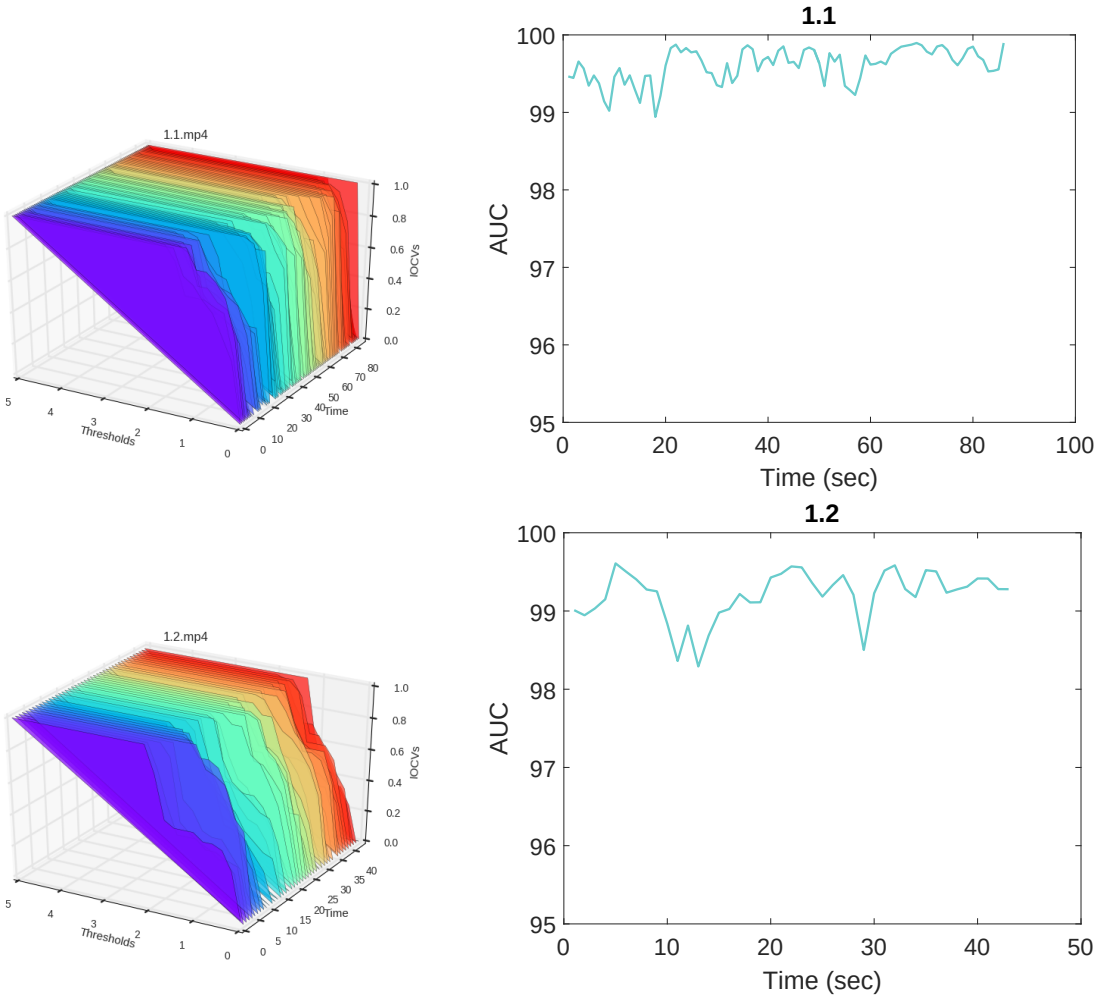


Figura E.7: Congruencia entre observadores (IOC) y área bajo la curva (AUC) para los 16 vídeos de referencia. Esta figura muestra los datos para los vídeos de referencia de la cocina (parte 1).

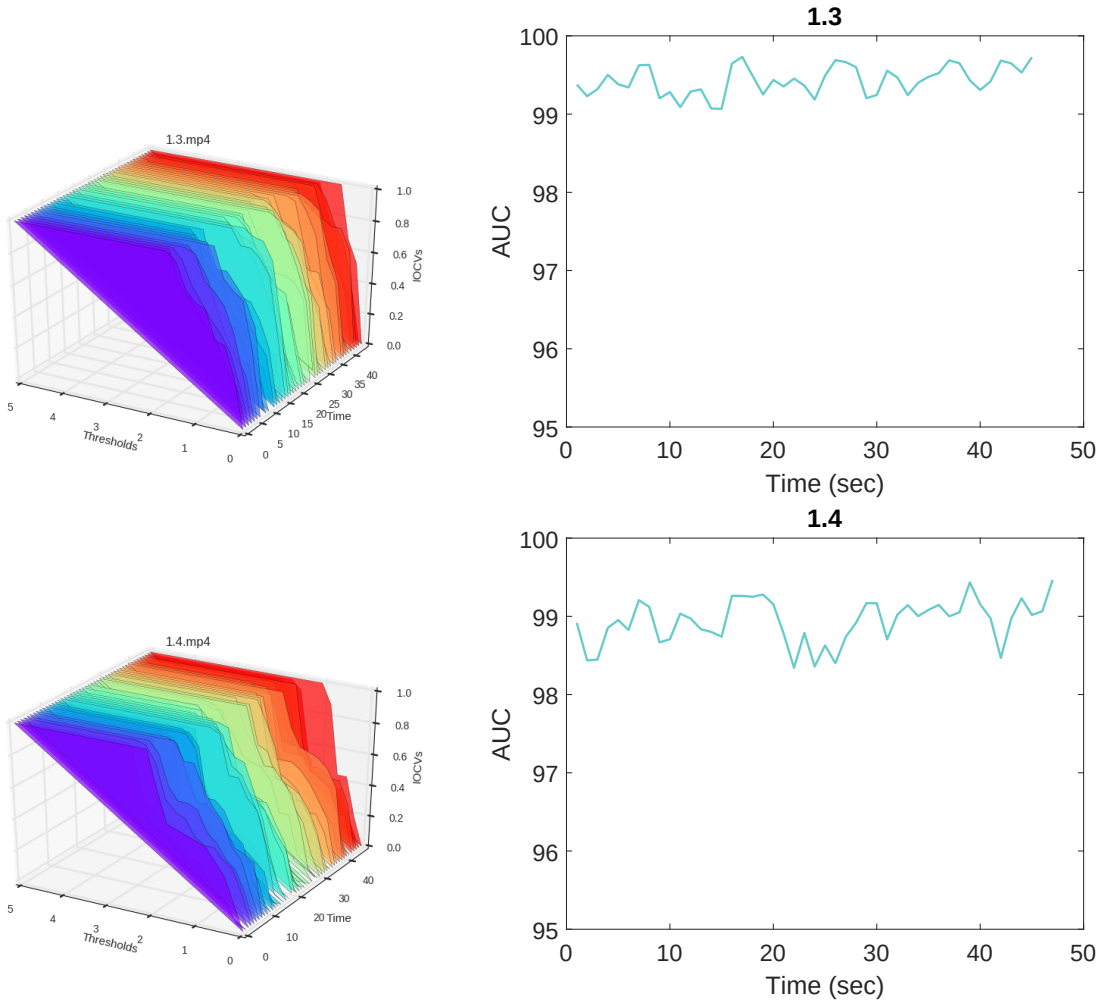


Figura E.8: Congruencia entre observadores (IOC) y área bajo la curva (AUC) para los 16 vídeos de referencia. Esta figura muestra los datos para los vídeos de referencia de la cocina (parte 2).

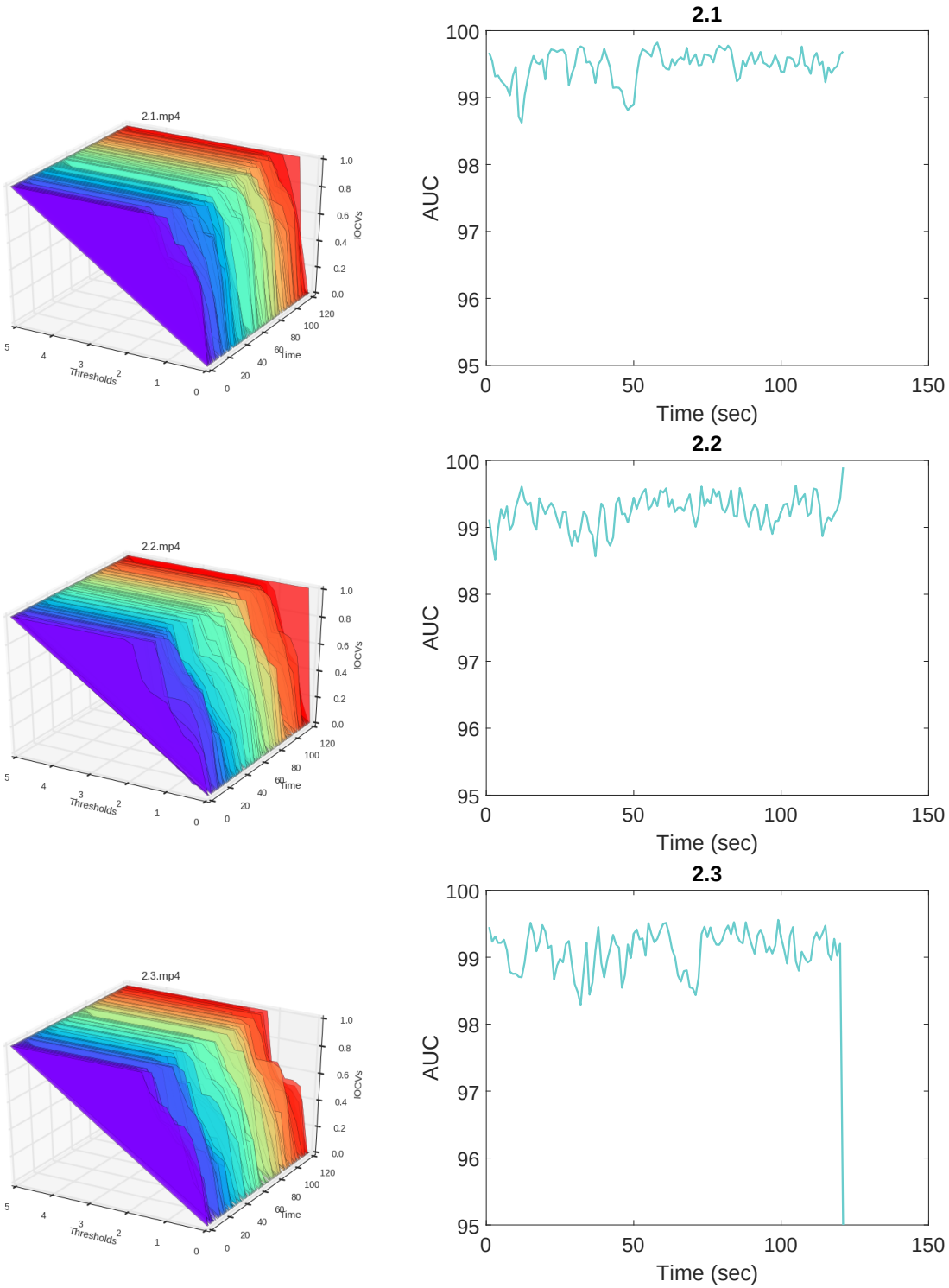


Figura E.9: Congruencia entre observadores (IOC) y área bajo la curva (AUC) para los 16 vídeos de referencia. Esta figura muestra los datos para los vídeos de referencia de la sala de estudio.

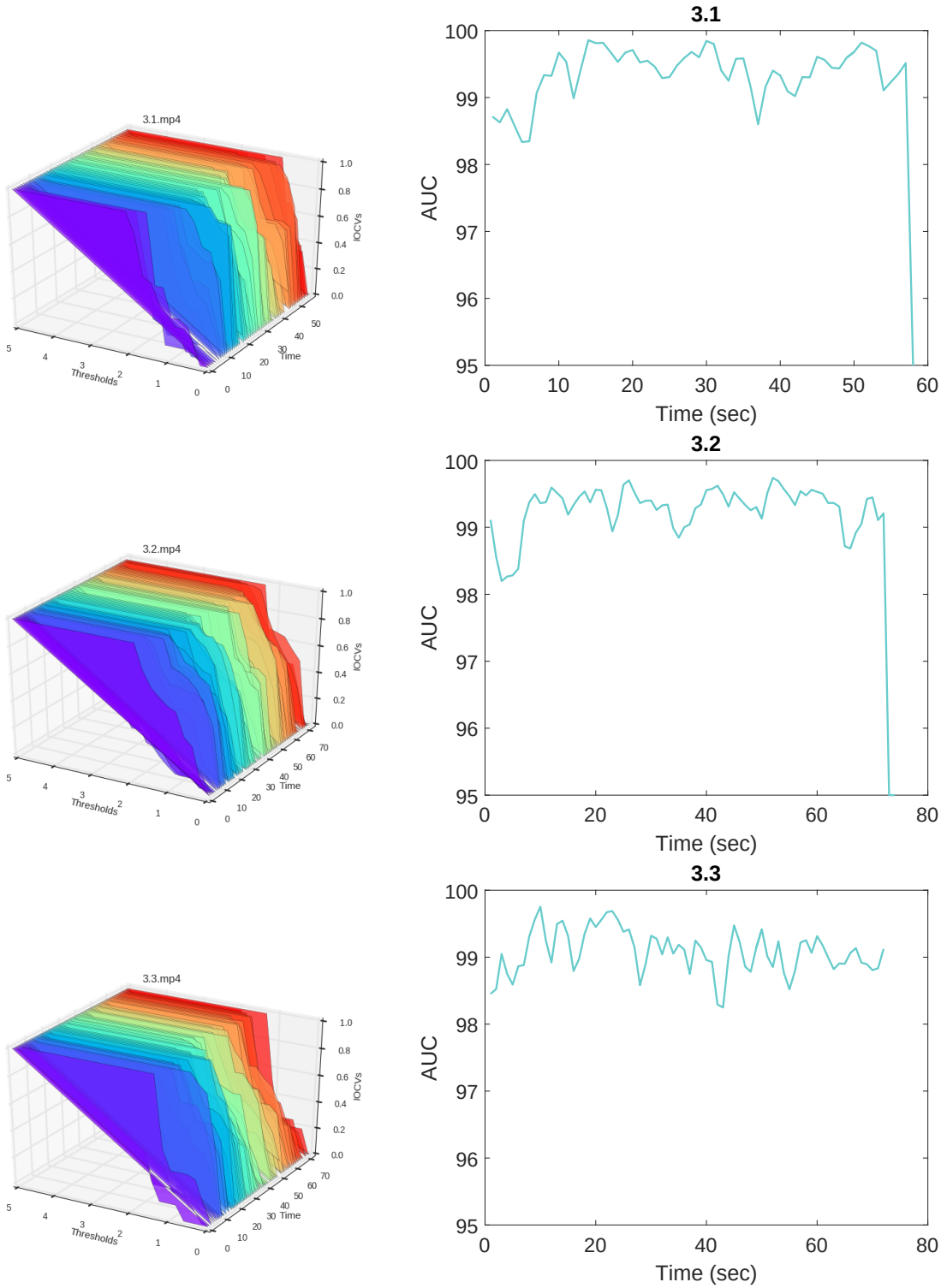


Figura E.10: Congruencia entre observadores (IOC) y área bajo la curva (AUC) para los 16 vídeos de referencia. Esta figura muestra los datos para los vídeos de referencia de la sala de estar.

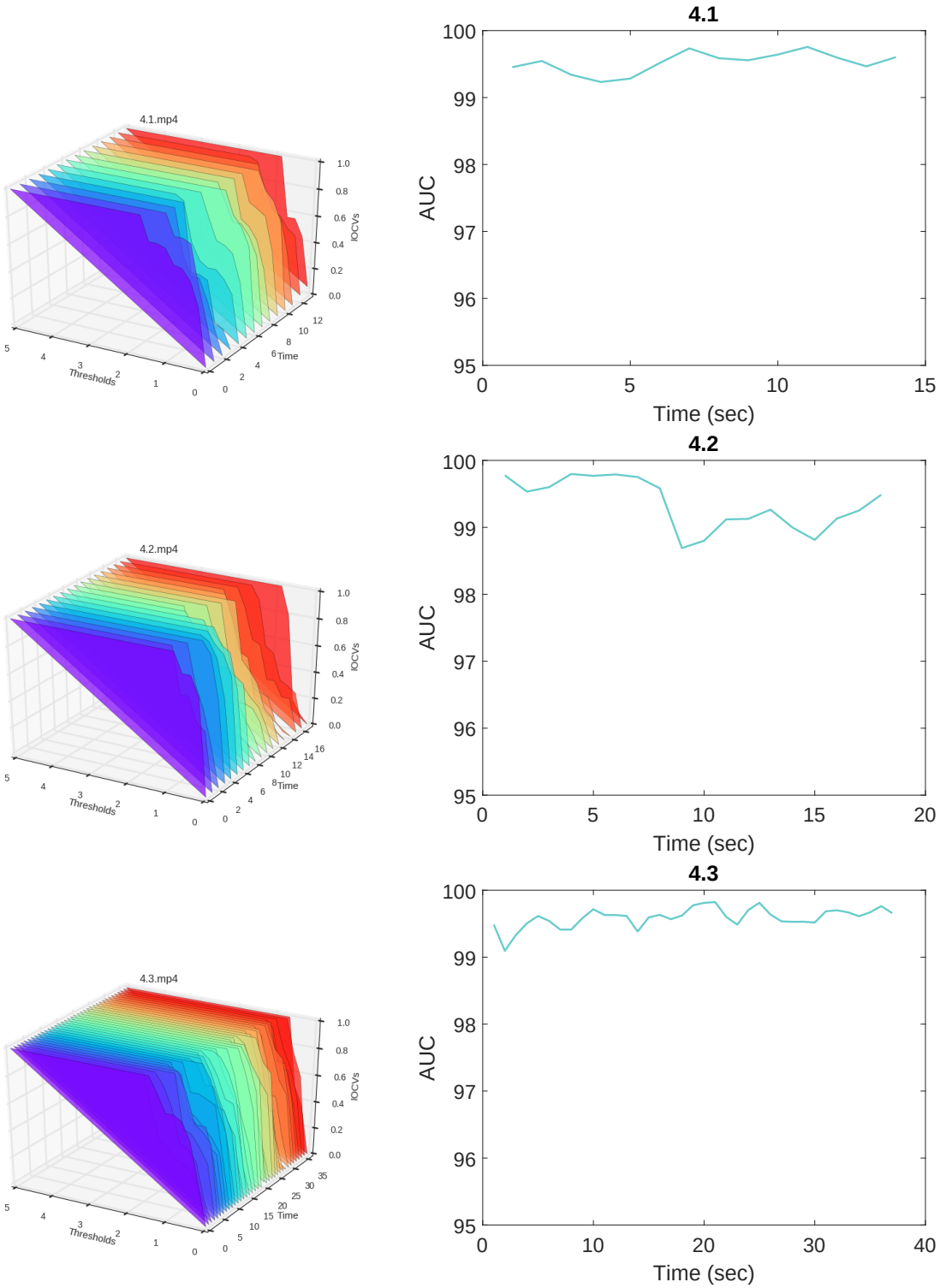


Figura E.11: Congruencia entre observadores (IOC) y área bajo la curva (AUC) para los 16 vídeos de referencia. Esta figura muestra los datos para los vídeos de referencia de las escaleras (parte 1).

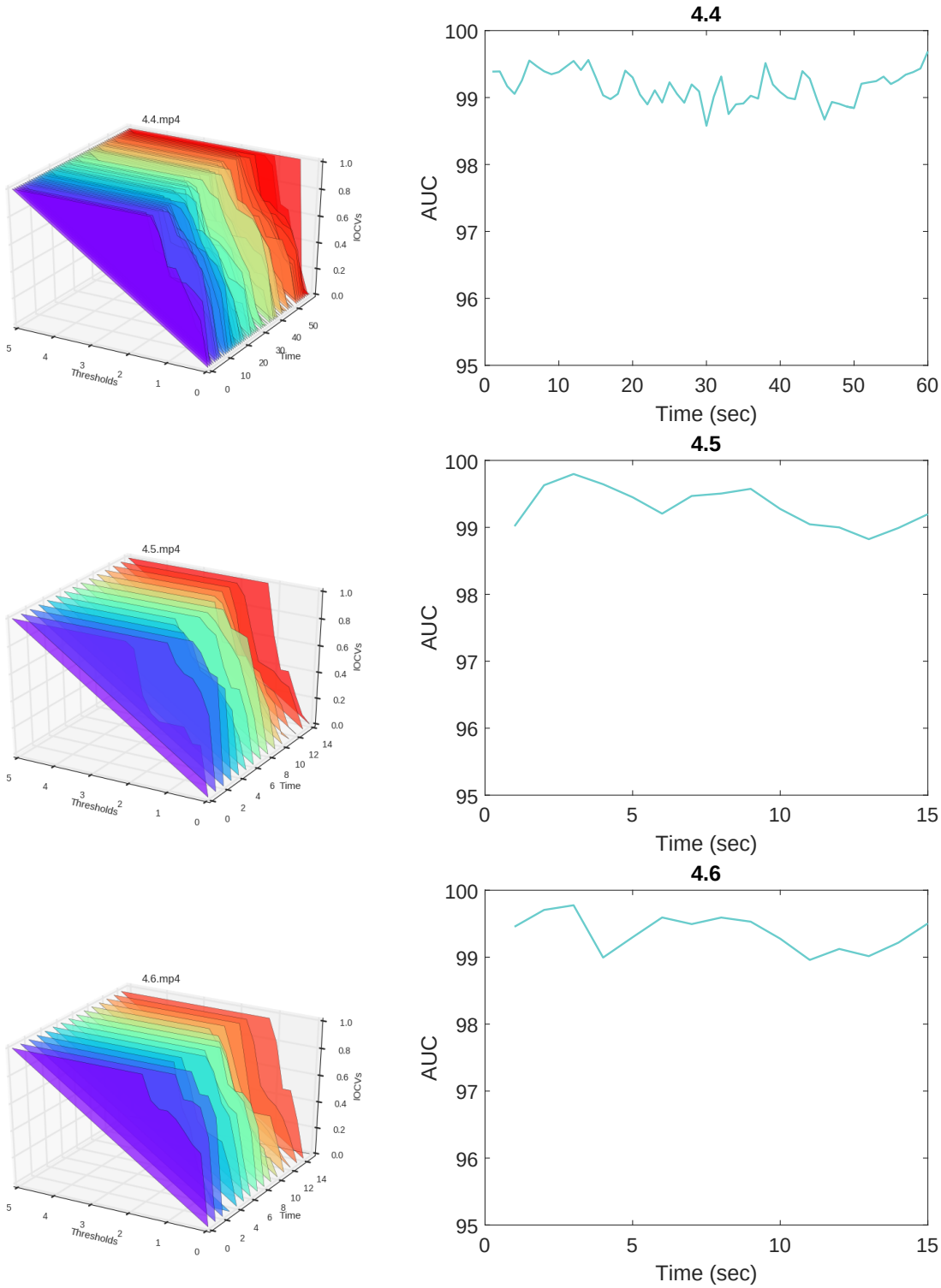


Figura E.12: Congruencia entre observadores (IOC) y área bajo la curva (AUC) para los 16 vídeos de referencia. Esta figura muestra los datos para los vídeos de referencia de las escaleras (parte 2).