

UNIVERSIDAD POLITÉCNICA DE
VALENCIA

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS
Y COMPUTACIÓN

TESINA PARA EL TÍTULO “MASTER EN
INTELIGENCIA ARTIFICIAL, RECONOCIMIENTO DE
FORMAS E IMAGEN DIGITAL”

Análisis de las características de la visión periférica y la ceguera por falta de
atención para la aceleración del rendering en aplicaciones 3d en tiempo real

López Luro Francisco Guillermo
flopezluro@gmail.com

Director de la tesina: Dr Ramón Mollá Vaya

VALENCIA

ESPAÑA

Diciembre 2009

AGRADECIMIENTOS

Quiero agradecer a todos los que me acompañaron en este proceso y siempre me apoyaron en los momentos difíciles. A mis padres y hermanos, que tienen una fe y apoyo incondicional a todo lo que me toca vivir.

A mis compañeros del master, Armando, Gustavo, Vlad, Cesar, Ihab y David, a mi amigos en Argentina que siempre están presentes a la distancia y a mis nuevos amigos en España, de todos los países y colores que han enriquecido mi mente y mi espíritu de una manera inimaginable. A mi compañero Iñigo, con quien emprendí el estudio de este trabajo, por su fortaleza y tesón, que me permitieron avanzar a paso firme en varias materias del Master.

Finalmente a mi tutor, Don Ramón Mollá Vaya, por su guía y apoyo para emprender un camino nuevo, con aciertos y errores, de experimentación y charlas de café, y de momentos de pura creatividad que nos convierten a los estudiantes en los empleados mas afortunados en este mundo.

Índice general

1. Introducción	1
2. Sistema visual humano y percepción	4
2.1. Evolución y adaptación de la visión	5
2.2. Espectro visible	6
2.3. El ojo humano	7
2.3.1. Anatomía del ojo	7
2.4. Percepción visual	10
2.4.1. Función de sensibilidad al contraste (Contrast Sensitivity Function – CSF)	11
2.4.2. Sensibilidad temporal	15
2.4.3. Agudeza visual y visión periférica	16
2.4.4. Ceguera por falta de atención (Innattentional Blindness) .	18
3. Técnicas de aceleración	20
3.1. Nivel de detalle	21
3.1.1. Renderizado fuera de línea basado en la percepción	21
3.1.2. Nivel de detalle estático	21
3.1.3. <i>LoD</i> dinámico y dependiente de la vista	24
3.2. Particionado espacial	25
3.3. Impostores	25
3.4. Otras técnicas de <i>LoD</i>	27
3.4.1. <i>LoD</i> en función de la velocidad	27
3.4.2. <i>LoD</i> en función de la profundidad de campo	28
3.4.3. <i>LoD</i> aplicado a las animaciones de humanoides	30
3.5. Escalado en la periferia	32
4. Análisis y definición del problema	33
4.1. Técnicas de aceleración 3d	33
4.2. Percepción visual	34
4.3. Factores psicológicos	35

5. Estudios y resultados	37
5.1. La aplicación	37
5.1.1. Guión del juego	37
5.1.2. Punto de interés y arco visual	39
5.1.3. Detecciones de cambios en la periferia	39
5.1.4. Jugabilidad	41
5.2. Modelos 3d y técnica de transición para el LoD estático	42
5.2.1. Unpopping	42
5.3. Pruebas	44
5.3.1. Primera prueba	44
5.3.2. Segunda prueba	45
5.3.3. Tercera prueba	47
5.4. Resultados	48
6. Conclusiones y trabajo futuro	53
7. Apéndice	63

Índice de figuras

2.1. Espectro de ondas electromagnéticas	7
2.2. Anatomía del ojo	8
2.3. Distribución de conos y bastones en la retina	9
2.4. Patrón de ondas en función de la frecuencia y el contraste.	13
2.5. Ejemplos de patrones contraste de (a) baja y (b) alta frecuencia espacial. La curva debajo de cada patrón describe la naturaleza sinusoidal de la distribución de intensidad.	14
2.6. Función de sensibilidad al contraste	14
2.7. Descomposición del proceso de visión	15
2.8. (a) CSFs para velocidades de 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, y 128 grados/s (derecha a izquierda). (b) La frecuencia espacial detectable más alta posible en función de la velocidad.	16
2.9. Factor de magnificación cortical	18
3.1. Idea básica de la técnica <i>LoD</i> , (a) niveles de detalle para el conejo de Stanford, (b) uso de los niveles según su tamaño en pantalla [Lue03].	22
3.2. <i>True Impostors</i> , Júpiter con 150.000 rocas en su anillo	26
3.3. Simulación de emborronado en la demo de <i>Animusic Pipe Dream de ATI</i> . Deformación de los modelos en función de la velocidad	28
3.4. Área de fusión de Panum. El objeto A está enfocado, dentro del área de Panum [OYT96]	29
3.5. Efecto de profundidad de campo generado con un filtro de difusión anisotrópica y la información del <i>Z-Buffer [BFSC04]</i> . En la imagen, el punto de enfoque ha sido puesto en la segunda fila de coches.	30
3.6. Demo en tiempo real de <i>ATI</i> mostrando su último personaje, <i>Ruby</i> , creado por la compañía para promocionar sus productos.	31
3.7. Motion LoD para modelado de humanoides	32

5.1. Imagen de la aplicación desarrollada, manejando por la ciudad. El tanque verde es el punto donde debe colocarse el jugador, y desde allí disparar al <i>villano</i> que está atacando la estatua.	38
5.2. Modo disparo, atacando al <i>villano</i>	40
5.3. Pantalla inicial del juego, con las instrucciones	41
5.4. Modelos del repositorio de Stanford, el Conejo, el Armadillo y el Dragón	43
5.5. Unpopping	44
5.6. Función para mapear los LoDs en función de la distancia periférica con el <i>POI</i> . Además, se muestran las resoluciones para el Conejo que se usan en cada intervalo de distancia.	46
5.7. Movimiento del POI respecto del LoD en la segunda prueba. $I_0, I_1, ..$ son las trayectorias concéntricas que hace el POI alrededor del LoD.	47
5.8. Porcentaje de detecciones de cambios de LoD (de un promedio de 42 cambios por usuario)	48
5.9. Número total de detecciones hechas por 20 jugadores, de un total de 860 cambios de resolución	49
5.10. Número medio de detecciones al aumentar y disminuir la resolución, considerando el conejo y el dragón	50
5.11. Número medio de detecciones y no-detecciones de cambios de <i>LoD</i> en la periferia tomando solo el conejo.	51
5.12. Número medio de detecciones y no-detecciones de cambios de <i>LoD</i> en la periferia tomando solo el dragón.	51
5.13. Comparación de las distancias medias de detección de cambios de <i>LoD</i> entre saltos de resolución normales y saltos de resolución grandes. Los cambios grandes se realizan entre las siguientes resoluciones (3k, 750, 160, 80).	52

Índice de cuadros

2.1. Comparación entre conos y bastones desde la funcionalidad que proveen	10
5.1. <i>LoDs</i> construidos con MeshLab para cada modelo	42

Capítulo 1

Introducción

El sistema visual humano (SVH) tiene muchas limitaciones que pueden ser explotadas al momento de desarrollar aplicaciones 3d en tiempo real, como video juegos y mundos virtuales. Nuestra “agudeza visual” decrece de manera continua en la periferia del ojo, permitiendo que se pueda reducir el nivel de detalle de los modelos, el color de los mismos y hasta incluso el tamaño. Nuestro sistema visual presenta un alto grado de especialización funcional, en el cual sus canales de entrada procesan de manera independiente y diferente el estímulo visual que se percibe. Podemos distinguir dos etapas claras en el proceso de visión, una es la que detecta los cambios de menor resolución en una gran área visual, que es la visión periférica, y otra de mucho mayor resolución y sensible al color que nos permite identificar objetos y reconocerlos, la visión central [Tre86].

A lo largo de los años, la capacidad de percepción a detalles en el SVH ha sido modelada con el concepto de la función de sensibilidad al contraste, o *Contrast Sensitivity Function (CSF)*[MS74]. Esta función modela la sensibilidad que tienen las personas para detectar diferentes frecuencias espaciales en función del contraste con que se presentan los estímulos. Existen gran cantidad de versiones y modificaciones de este modelo, que contemplan variables que en sus inicios no se habían tenido en cuenta, como velocidad del movimiento del ojo, excentricidad del estímulo y profundidad de campo del estímulo entre otros [Lue03].

En el presente, para poder crear efectos más realistas, imágenes con mayor definición, y mantener altas tasas de cuadros por segundo, las aplicaciones gráficas 3d interactivas están demandando más y más poder de cómputo. Además, existen otras tareas no gráficas como simulación, IA y simulación de fenómenos físicos, que están migrando de la CPU a la GPU, sobrecargando la unidad de procesamiento gráfico. Por esto, siempre existe la necesidad de reducir el trabajo gráfico en la CPU y la GPU, para mejorar el rendimiento de las apli-

caciones, disminuir el consumo de energía (en dispositivos móviles) y mejorar otras características no gráficas como las mencionadas [Ols08, Mar08].

Una de las técnicas más utilizadas a la hora de reducir los costos de la síntesis de imagen, sin perder la calidad visual o controlando de manera precisa la degradación de la misma, es el uso de Niveles de Detalle (*LoDs*) para los modelos 3d que forman parte de la escena [Cla76]. Por otro lado, la calidad de una imagen generada por computadora depende inevitablemente de la percepción visual de la persona que está sentada frente al monitor. Las técnicas de LoD son usadas normalmente para disminuir el tamaño de las mayas poligonales 3d de los objetos en función de un criterio de distancia 3d entre la cámara y el objeto en cuestión. Como se mencionó, la visión sufre un decaimiento de la agudeza visual y percepción de pequeños cambios en la periferia.

Además, existe un factor psicológico adicional que puede afectar al proceso completo de visión. La *ceguera por falta de atención*, o “Innattentional Blindness” [MR98], es la incapacidad de notar objetos o cambios visibles (pero inesperados) porque el sujeto ha puesto su atención en otro objeto o en realizar una tarea específica.

En el presente trabajo se realizaron mediciones de la percepción visual en la periferia a los cambios de resolución de modelos 3d en aplicaciones interactivas como video juegos, agregando una variable que no ha sido considerada anteriormente en la literatura, que es la *ceguera por falta de atención*.

El uso de video juegos y aplicaciones interactivas 3d como herramienta de evaluación de factores psicológicos y fisiológicos está aumentando cada vez más, debido a que estos permiten generar ambientes más dinámicos y flexibles, proponiendo nuevos desafíos y sobre todo evaluando métricas en el mismo dispositivo donde luego se implementarán las soluciones. Por ejemplo, hasta el día de hoy, se creía que la *CSF* de las personas con visión normal se mantenía estable variando levemente entre personas de diferentes edades, sin embargo estudios recientes en la Universidad de Rochester [LPMB09] publicados en la revista *Nature* muestran que jugadores de video juegos de acción (*gamers*) tienen mejor percepción de pequeños detalles (altas frecuencias) que personas que no juegan frecuentemente video juegos, demostrando que el SVH puede adaptarse y mejorar, en función de los estímulos cotidianos.

Para provocar este estado de *ceguera por falta de atención* en los participantes, se desarrolló una aplicación interactiva 3d de tiempo real como un video juego, que mantiene al usuario ocupado y entretenido realizando movimientos de coordinación con sus dedos y vista para realizar una tarea específica, mientras se realiza el experimento y se miden sus percepciones a cambios de resolución de modelos 3d en la periferia.

Debido a que este factor psicológico no ha sido combinado previamente con

los modelos de las funciones *CSF* en la periferia, y con objetos en movimiento, los resultados del trabajo son aún experimentales y no resultan fáciles de comparar con otros experimentos previos.

Con una primera aproximación de una función para asignar intervalos de distancias a los diferentes *LoDs* de un modelo, nuestros resultados mostraron que los usuarios solo detectaron el 20% del total de los cambios que ocurrieron. Por otro lado, en base a las detecciones que observaron los usuarios, se determinaron distancias en grados de ángulo visual y en píxeles para realizar los cambios de resolución de los modelos utilizados sin que el usuario los detecte. Denominamos a estas zonas, zonas seguras, intentando dar una heurística simple e inmediata para los desarrolladores de aplicaciones interactivas que puedan generar condiciones de interacción similares a las de este experimento.

Por último, se analizaron las detecciones a los cambios de resolución variando la granularidad de estos, es decir, comparando las distancias a las que el usuario detecta un determinado cambio de resolución, cuando el modelo previo es muy parecido o difiere mucho en su geometría.

En el capítulo 2 se describen aspectos del sistema visual humano, desde su evolución inicial en los seres vivos hasta los modelos que se conocen hoy en día sobre su funcionamiento. Se describe en detalle la *CSF* y sus variantes, así como la *ceguera por falta de atención*.

El capítulo 3 presenta una breve descripción y análisis de algunas de las técnicas utilizadas hoy en día para acelerar el rendering en las aplicaciones 3d, poniendo énfasis en las técnicas relevantes para este trabajo. En el capítulo 4 se realiza un análisis del problema escogido, y se mencionan las consideraciones más importantes que se tuvieron para acotar la línea de trabajo.

El capítulo 5 describe la aplicación desarrollada para hacer el estudio, las pruebas realizadas y los resultados más relevantes del trabajo.

En el capítulo sexto se dan las conclusiones del trabajo y las líneas que surgen de los aciertos y los errores de este trabajo.

Capítulo 2

Sistema visual humano y percepción

El sistema visual humano (SVH) se ha desarrollado y perfeccionado después de millones de años de adaptación al medio que lo rodea, permitiendo al hombre interactuar de manera efectiva y precisa con su hábitat. El espectro visible, la visión central y periférica, las diferentes clases de células fotorreceptoras, y la visión binocular son algunas de estas adaptaciones. A lo largo de los años, los científicos han intentado definir modelos que expliquen lo mejor posible el funcionamiento del SVH. Un modelo muy conocido sobre la capacidad de percepción de detalles en la visión es la *Función de sensibilidad al Contraste* (*Contrast Sensitivity Function – CSF*), definida entre otros por Mannos y Sakrison [MS74], esta función intenta modelar la capacidad del sistema de visión de distinguir las diferentes frecuencias espaciales en función del contraste para un estímulo dado. Este modelo ha sido estudiado y modificado a lo largo de los años, para contemplar por ejemplo variables como velocidad y excentricidad, que permiten definir algunos criterios importantes al momento de analizar las técnicas de aceleración de aplicaciones 3d que se verán en el capítulo 3. El SVH tiene muchas limitaciones que pueden ser explotadas a la hora de hacer el rendering en nuestras aplicaciones. La "agudeza visual" decrece de manera continua en la periferia del ojo, permitiendo que se pueda reducir el nivel de detalle de los modelos, el color de los mismos y hasta incluso el tamaño.

Por otra parte, en la percepción de los estímulos también juegan un rol importante factores psicológicos, como la *ceguera por falta de atención* o *Inattentional Blindness* [SC99, MR98], que impide a una persona detectar con naturalidad cambios evidentes en un estímulo visual, debido a que se encuentra involucrado en una actividad que requiere su completa atención.

En la sección siguiente y la sección 2.2 se tratan algunos aspectos relativos a la evolución de los sistemas de visión en los seres vivos, particularmente los vertebrados como el hombre. Además se revisa la influencia de las ondas electromagnéticas emitidas por el Sol en la evolución del SVH. En la sección 2.3 se describe el órgano central en la visión del hombre, el ojo, y se explica como sus diferentes partes afectan en el proceso de la visión. Finalmente, la sección 2.4 profundiza en los aspectos cruciales de la percepción visual, la función *CSF*, la visión periférica, la capacidad del sistema de visión de percibir detalles en movimiento y algunos factores psicológicos que también pueden ser explotados a la hora de diseñar e implementar aplicaciones 3d interactivas.

2.1. Evolución y adaptación de la visión

En el comienzo, los sistemas de visión más elementales comenzaron a aparecer como simples fotorreceptores en organismos unicelulares, hace más de 600 millones de años. Estos operaban de forma refleja, donde el estímulo eletromagnético producía un impulso nervioso y permitía por ej. que el microorganismo flagelado cambiara su trayectoria en función de la dirección de una fuente de luz.

Estos sistemas sensoriales se han desarrollado para ser muy sensibles a variaciones de intensidad de luz y de movimiento. En los vertebrados, se pueden encontrar gran número de estos fotorreceptores en el órgano central del sistema, el ojo, más precisamente en la retina del mismo. Mediante un complejo mecanismo de lentes flexibles y filtros, el sistema de visión puede enfocar y captar escenas con gran precisión.

Luego de millones de años en la evolución de los animales vertebrados, su sistema de visión quedó formado por 2 ojos, y aunque han habido intentos de evolución a mayor número de ojos en algunos vertebrados [SHR*01], esta característica de visión frontal es muy importante para definir como funciona la visión en los humanos.

Entre las diferentes hipótesis sobre la posición frontal de los ojos en el hombre o los primates en general, y gran parte de los depredadores, se destaca la siguiente. Cuando los primates comenzaron a vivir sobre los árboles se encontraron con un nuevo problema, la visión lateral no ayudaba a encontrar superficies estables dentro del entramado de ramas, ni facilitaba el movimiento entre estas por carecer de agudeza visual y profundidad. Por esto, los moradores de los árboles desarrollaron eventualmente caras más planas con los ojos mirando al frente. Esto les permitió fundir las dos imágenes en una tridimensional, además de requerir un mayor desarrollo de sus cerebros, que se incrementaron acorde a ello. El perfeccionamiento de la visión espacial y la percepción de profundidad

provocó la reducción de la importancia del sentido del olfato y un incremento de la habilidad manual. El humano moderno es un ejemplo extremo de estas consecuencias.

Por otro lado, los depredadores en general también se vieron favorecidos por este cambio, debido a que su necesidad de supervivencia se basaba en calcular distancias con precisión y poder detectar a sus presas a grandes distancias, contrario a los predados, como el conejo, que necesitan tener una visión completa de lo que los rodea, para poder sobrevivir en un ambiente hostil. Estos últimos, desarrollaron la posición de los ojos a los lados de la cabeza, maximizando el campo de visión.

Como resultado de la posición frontal de los ojos, la visión periférica se encuentra muy disminuida, y solo es remediable mediante movimientos de cabeza en la dirección de interés.

2.2. Espectro visible

La capa de ozono actual filtra casi toda la radiación (ultravioleta) por debajo de los 280 nm (nanómetros), que por ser radiación ionizante, es capaz de causar gran daño biológico en los tejidos externos. Al absorber la mayoría de esta radiación emanada por el Sol, la capa de ozono se convierte en un componente crucial de nuestro ecosistema.

La mayor intensidad de la radiación del Sol que penetra en nuestra atmósfera está localizada en la parte visible del espectro electromagnético. La máxima intensidad de luz solar a la que estamos expuestos está en la región amarilla del espectro visible (560 a 590 nm.), lo que corresponde a la sensibilidad espectral de los conos fotorreceptores de la retina y responsables de la visión diurna.

La porción de rayos ultravioleta que llega hasta nosotros (de 280 a 400 nm) está en gran parte reducida por el ozono, pero viene en suficiente cantidad para causar reacciones biológicas familiares como un incremento en la producción de melanina y daños genéticos como cáncer de piel. Dado su potencial destructivo, es lógico en términos evolutivos que hayamos alejado nuestra sensibilidad de la luz ultravioleta.

El rango del espectro electromagnético visible para el humano está entre 400 y 750 nanómetros, pero no es casualidad que nuestro sistema visual pueda ver en ese rango de frecuencias, sino otra adaptación. El medio en el que nos movemos (el aire) es prácticamente transparente a esta porción del espectro. La radiación visible es capaz de transmitir información desde los objetos distantes hasta nuestros ojos. No necesitamos basarnos únicamente en el contacto físico o químico para saber qué más comparte nuestro medio, particularmente depredadores y presas. La figura 2.1 muestra el espectro visible al ojo humano, y su

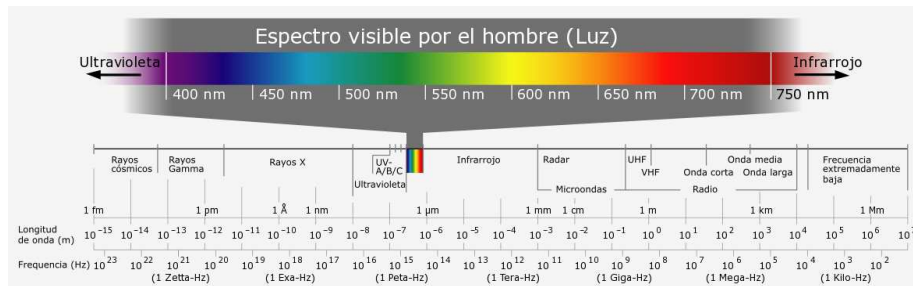


Figura 2.1: Espectro de ondas electromagnéticas

relación con el resto de las ondas electromagnéticas conocidas.

En resumen, nuestras capacidades visuales son el resultado directo de una larga evolución en un entorno relativamente estable bajo la atmósfera y el Sol.

2.3. El ojo humano

La información o estímulo visual viaja hasta nuestro sistema visual en forma de luz a través de un medio físico, comunmente aire. La luz es transformada (enfocada) por las lentes de cada ojo y proyectada en la retina en la parte trasera de cada ojo. La retina consta de fotorreceptores, denominados conos y bastones, que convierten la radiación electromagnética (luz) en actividad neuronal (química y eléctrica). Los nervios ópticos conectados a los ojos transmiten esta información hasta la corteza visual del cerebro, en ambos hemisferios.

El campo de visión (FOV) está limitado obviamente por el movimiento que puedan realizar nuestros ojos (con la cabeza fija), por nuestra nariz y las cejas. Además, existe un gran solapamiento entre las imágenes que genera cada ojo, que el cerebro debe luego ordenar y juntar para generar una imagen coherente.

2.3.1. Anatomía del ojo

La estructura del ojo es manejada por 3 pares de músculos, y su equilibrio y posicionamiento resultan en un constante e imperceptible movimiento, que hace que la información que recibe la retina cambie aunque sea muy poco, cada 100 ms. Existen muchas imperfecciones en el sistema visual, como las aberraciones cromáticas, distorsiones, y hasta fluidos en el ojo, que hacen que la luz no sea igualmente procesada, sin embargo el funcionamiento de la visión humana pareciera indicar que existen mecanismos temporales en el cerebro que permiten construir una imagen apropiada incluso ante la presencia de estos problemas. La figura 2.2 muestra un esquema general de la anatomía del ojo, y sus partes principales.

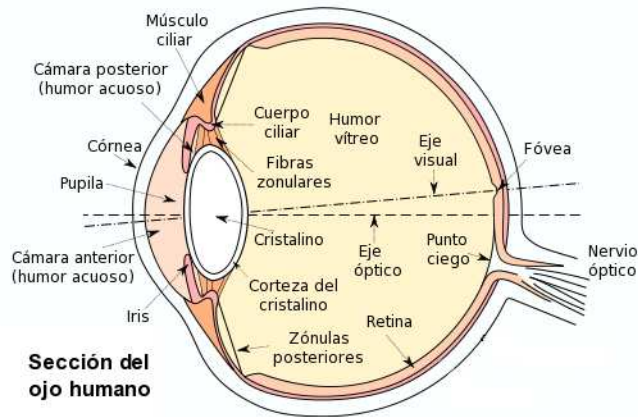


Figura 2.2: Anatomía del ojo

La retina tiene aproximadamente 125 millones de fotorreceptores, que convierten la energía electromagnética en energía eléctrica y química. Los fotorreceptores se clasifican en conos y bastones. Los bastones son muy sensibles a la luz y a los cambios de luminosidad, pero se saturan fácilmente ante altos niveles de luz. Los bastones son más efectivos en el proceso de visión en condiciones de baja iluminación, llamada visión escotópica, como en la noche. Existen unos 120 millones de bastones, estos son más sensibles a longitudes de onda cercanas a los 500 nm y son hasta 10 veces más sensibles que los conos. Los conos, aproximadamente 6 millones, son mucho más efectivos en condiciones de alta iluminación (durante el día), en la visión fotópica, y son sensibles al color. Los conos se dividen en tres tipos, cada uno de ellos más sensible a diferentes longitudes de onda, desde las cortas (420 nm), las medias (530 nm) hasta las largas (560 nm) [Gla95].

Los conos y bastones no están uniformemente distribuidos en la retina, los conos están ubicados en una zona en el centro de la mácula lutea, llamada fovea, de aproximadamente $1,5 \text{ mm}^2$. Los fotorreceptores de la fovea son los responsables de la agudeza visual en el proceso. Cuando se hace "foco" en un objeto, se centra la proyección de la luz que emite este justo en el centro de la fovea, la foveola, de aproximadamente 0.33 mm de diámetro. En la fovea, existe una convergencia 1:1 con las células ganglionares (del nervio óptico), por ello es la zona donde se obtiene mayor resolución o nitidez visual. El ser humano hace foco cambiando la forma de la lente. El "focal power" (en dioptrios) del humano es de 15 dioptrios.

Como se puede observar en la figura 2.3, en la periferia de la fovea la densidad de los fotorreceptores disminuye, además solo en el eje óptico hay una cantidad considerable de conos, y fuera de este solo participan los bastones en el proceso

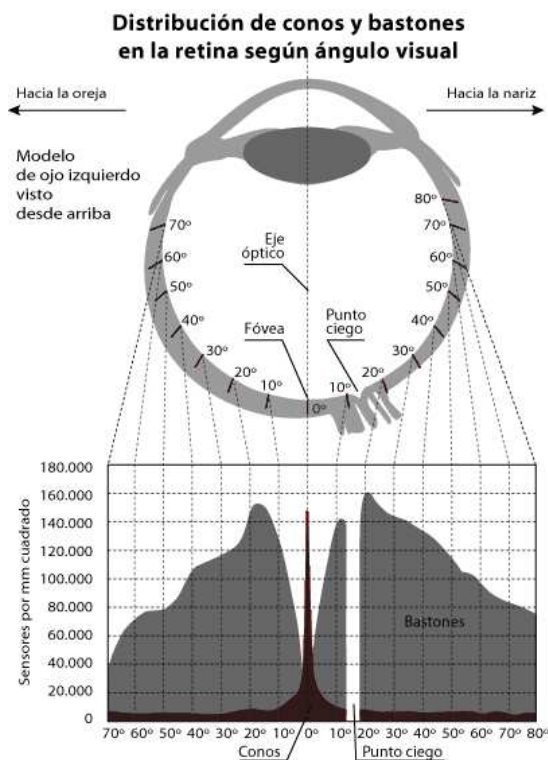


Figura 2.3: Distribución de conos y bastones en la retina

de la visión. En la figura también se puede apreciar el punto ciego, lugar donde se inserta el nervio óptico en el ojo. Obviamente no detectamos el punto ciego normalmente, puesto que el cerebro completa la información ausente con los estímulos del ojo opuesto. El cuadro 2.1 muestra las diferencias entre conos y bastones, vistos como dos partes de nuestro sistema de visión para diferentes propósitos, la visión diurna, y la visión nocturna y periférica.

Sumación temporal En cada fotoreceptor, ocurre un proceso químico que dura algunos milisegundos. Todos los fotones que recibe cada fotoreceptor durante este proceso contribuyen al resultado, dando lugar a una respuesta promedio en el tiempo que dura el proceso. El efecto de esta promediación se denomina "temporal smoothing", y se puede comprobar fácilmente al observar luces que parpadean a gran velocidad. Cuando una luz parpadea lentamente, se percibe cada parpadeo individual correctamente, pero al aumentar la velocidad, y superar la tasa denominada *CFF* (*critical flicker fusion*) el flash se fusiona en una imagen continua. Esta *CFF* está alrededor de los 60Hz. Bajo condiciones de iluminación bajas, el *CFF* disminuye levemente [Web64].

	Visión Nocturna (bastones)	Visión diurna (conos)
Sensibilidad relativa	1	1.000.000
Agudeza visual	Baja	Alta
Visión a color	NO	Si
Máxima sensibilidad	20 grados en la periferia	fóvea

Cuadro 2.1: Comparación entre conos y bastones desde la funcionalidad que proveen.

Sumación espacial Desde los más de 120 millones de fotorreceptores los estímulos deben transmitirse por aproximadamente un millón de células ganglionares. Estas últimas reciben los impulsos de múltiples fotorreceptores. La relación entre los fotorreceptores y estas células es muy diferente entre los conos y los bastones. Los conos, en la fóvea, tienen una relación de casi 1:1, requiriendo estímulos con suficiente energía para poder transmitir los impulsos, mientras que los bastones, en su mayor parte en la periferia de la fóvea, tienen una relación mucho mayor y muchos se conectan a la misma terminación de una célula ganglionar. La relación de muchos bastones a una sola célula ganglionar genera un efecto de amplificación del estímulo de luz, por eso es que en la noche solo los bastones participan en la visión, captando la poca luz disponible.

2.4. Percepción visual

El sistema visual presenta un alto grado de especialización funcional, en el cual sus canales de entrada procesan de manera independiente y diferente el estímulo visual que se percibe. Se pueden distinguir dos etapas claras en el proceso de visión, una es la que detecta los cambios de menor resolución en un gran área visual, que es la visión periférica, y otra de mucho mayor resolución y sensible al color que permite identificar objetos y reconocerlos con detalle, la visión central.

La teoría más aceptada hoy en día sobre como funciona la visión es el modelo multicanal. Desarrollado por Enroth-Cugell y Robson [ECR66] y Campbell [CR68], esta teoría propone esencialmente que el sistema visual procesa la imagen de la retina de manera simultánea a varias escalas espaciales diferentes. La mayoría de las escenas naturales que se pueden considerar tienen información visual a diferentes escalas. Por ejemplo, en el caso de un bosque, la silueta de todos los árboles puede dar un nivel de detalle bajo de la escena, aunque también se puede observar cada árbol en particular, aumentando la resolución de los detalles, llegando hasta las hojas, el máximo detalle en la escena. La teoría de

este modelo multicanal sugiere que el sistema visual extrae toda esta información “multiresolución” simultáneamente, y que después más adelante en alguna etapa del proceso de visión de más alto nivel se reúne y combina para formar la escena tal cual como la percibe el hombre.

Esta teoría concuerda con el conocimiento actual del diseño neuronal del sistema de visión humano. El tamaño del campo receptivo de una neurona define el tamaño del estímulo al cual esta es altamente sensible. En las diferentes etapas tempranas de la visión se encuentran células con un amplio rango de campos receptivos, capaces de detectar cada una diferentes niveles de detalle. Según este modelo, cada canal que procesa la información visual puede verse como una clase de neurona ajustada a un campo de recepción de un tamaño determinado. Aún más, estudios [Liv88] indican que el procesado del color, la agudeza visual, la velocidad de movimiento y el contraste también realizan en paralelo por diferentes partes del sistema visual.

2.4.1. Función de sensibilidad al contraste (Contrast Sensitivity Function – CSF)

Se pueden encontrar varias definiciones del contraste en la literatura, aquí se mencionan las tres (3) más empleadas.

Weber El contraste según Weber, directamente deducido de la ley general de Weber, intenta establecer una relación cuantitativa entre la magnitud de un estímulo físico y como éste es percibido. La ley general de Weber establece que *“el menor cambio discernible en la magnitud de un estímulo es proporcional a la magnitud del estímulo”*. En el caso del contraste, la función derivada de dicha ley es la siguiente:

$$C = \frac{I - I_b}{I_b} \quad (2.1)$$

donde I e I_b representan la luminancia de las características y del fondo respectivamente. Este modelo es usado normalmente cuando se encuentran pequeñas características sobre un fondo de luminancia constante (o asumido como tal).

Michelson El contraste de Michelson [Mic27] se define como

$$C = \frac{L_{max} - L_{min}}{L_{max} + L_{min}} \quad (2.2)$$

con L_{max} y L_{min} las luminosidades máximas y mínimas respectivamente. El contraste C está en el intervalo $\{0 \leq C \leq 1\}$. El contraste de Michelson se utiliza

en patrones en los cuales las características oscuras y claras son equivalentes y ocupan proporciones similares en la imagen.

RMS El contraste RMS no depende de la frecuencia espacial ni de la distribución de contraste de la imagen. Se define como el desvío estándar de las intensidades de los píxeles [Pel90].

$$C = \sqrt{\frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (I_{ij} - \bar{I})^2} \quad (2.3)$$

siendo I_{ij} la intensidad de cada píxel de una imagen de M por N , y \bar{I} como el promedio de intensidades de la imagen.

Con el sentido común, el contraste se puede interpretar con el siguiente ejemplo:

*Si un conductor de un automóvil se encuentra con una niebla muy densa en el camino, los objetos que normalmente vería sin problemas, como letreros con letras negras sobre fondo blanco, no se verán tan fácilmente puesto que todo ahora está más gris por la niebla. La diferencia entre lo blanco y lo negro (el contraste) ahora no será tan evidente, por ello algunos objetos ya no serán ni siquiera visibles, porque estarán debajo del **umbral de contraste**. Desde la perspectiva del conductor, es importante saber si todos los objetos con los que puede encontrarse se volverán visibles a la misma distancia (al mismo contraste) o los objetos de diferentes tamaños requerirán mayor o menor contraste para poder ser detectados.*

Se debe asumir que si dado un determinado contraste se pueden ver objetos o detalles de cierta resolución, bajo el mismo contraste se podrán ver objetos o detalles de menor resolución espacial.

La pregunta se puede reducir entonces a *¿Cuánto contraste se necesita para ver objetos de diferentes tamaños?*

Durante los años 60, los investigadores desarrollaron algunas ideas sobre la habilidad del hombre de percibir los detalles, e intentaron encontrar los límites del sistema de visión humano. El resultado de estos estudios es un modelo muy conocido denominado *Contrast Sensitivity Function (CSF)* presentado por Mannos y Sakrison[MS74] entre otros. En estos estudios se utiliza la definición del contraste de Michelson mencionada en la sección 2.4.1. Para realizar los estudios y poder medir las capacidades de percepción, utilizaron (y aún utilizan) unas gráficas o patrones (gratings en inglés) que muestran ondas cuadradas o

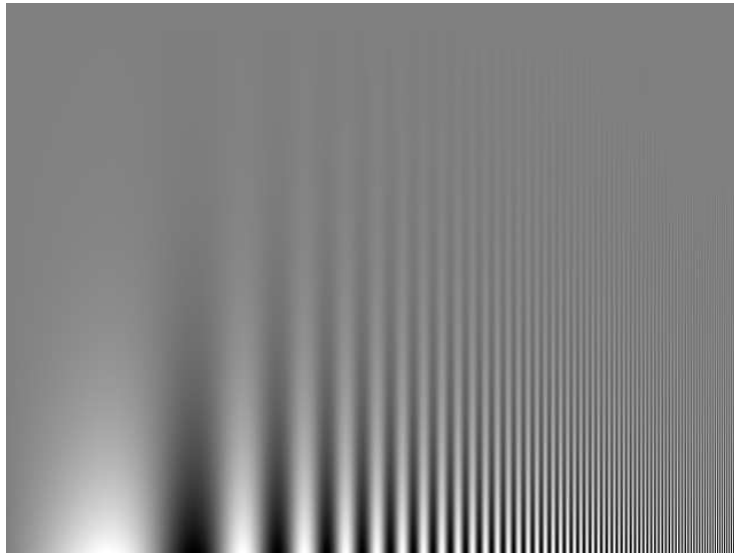


Figura 2.4: Patrón de ondas en función de la frecuencia y el contraste.

sinusoidales, en general verticales, de diferentes frecuencias y contrastes. La figura 2.4 muestra un patrón de ondas sinusoidales combinadas en frecuencia y contraste, creada por Campbell y Robson[CR68]. Los detalles imperceptibles en la gráfica, que pueden deberse a el bajo contraste o a la alta frecuencia, definen los límites de la percepción visual de la persona que lo observa. En estos patrones, un ciclo es un cambio de claro a oscuro o viceversa, y la frecuencia espacial de un patrón depende de a que distancia sea este presentado al individuo. La figura 2.5 muestra dos patrones, con diferentes frecuencias. Si el patrón (a) fuera posicionado para que ocupe un (1) grado de arco visual frente a una persona, su frecuencia espacial sería de 4 ciclos/grado.

Para tener una referencia práctica, dado que la fovea cubre entre 1 y 2 grados del ángulo visual, un dedo de la mano a distancia del brazo extendido se puede enfocar sin problemas.

La función CSF indica los límites de percepción a diferentes frecuencias y contrastes de un estímulo visual. Si la frecuencia del estímulo visual es muy alta (detalles imperceptibles) entonces no se podrá reconocer más el patrón del estímulo, y la percepción será solamente de un color uniforme. Estos límites de resolución espacial están directamente relacionados con el número y el tamaño de nuestros fotoreceptores como se mencionó anteriormente.

La función CSF a la que llegaron Mannos y Sakrison es la siguiente:

$$A(\alpha) = 2,6(0,0192 + 0,144\alpha)e^{-(0,144\alpha)^{1,1}} \quad (2.4)$$

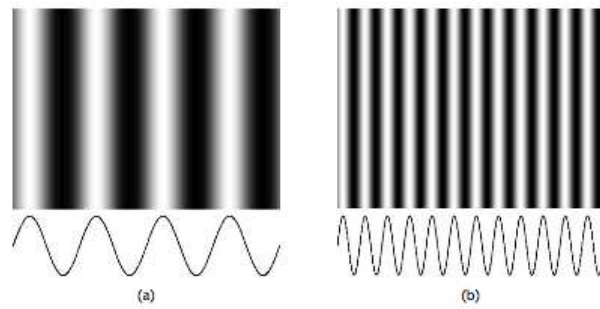


Figura 2.5: Ejemplos de patrones contraste de (a) baja y (b) alta frecuencia espacial. La curva debajo de cada patrón describe la naturaleza sinusoidal de la distribución de intensidad.

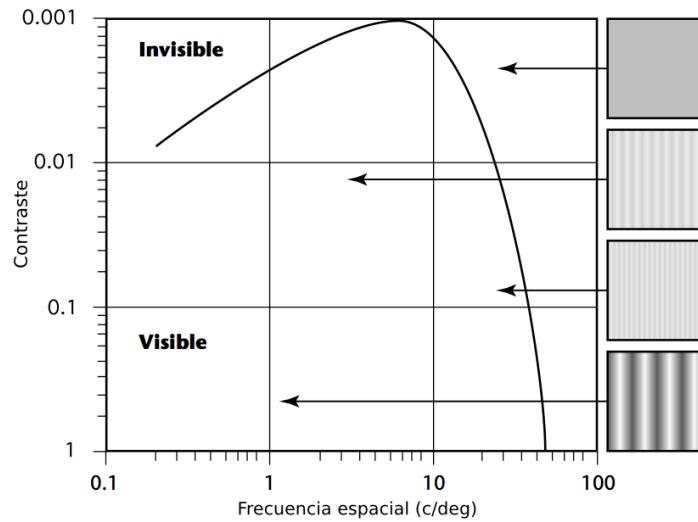


Figura 2.6: Función de sensibilidad al contraste

y en la figura 2.6 se puede observar una representación de tal función. La curva delimita lo que las personas, con visión normal, pueden ver y lo que no pueden ver, usando la visión central.

El modelo multicanal mencionado en la sección anterior se adapta perfectamente al modelo de la *CSF*, de hecho se cree que la *CSF* es una función que delimita cada una de las funciones de diferentes escalas de resolución espacial de cada canal. La figura 2.7 ilustra este concepto.

Hasta hace poco se creía que la *CSF* de una persona no podía mejorar, y que a medida que el sujeto avanza su edad su *CSF* indicaría una pérdida gradual de la agudeza visual. Estudios recientes en la Universidad de Rochester

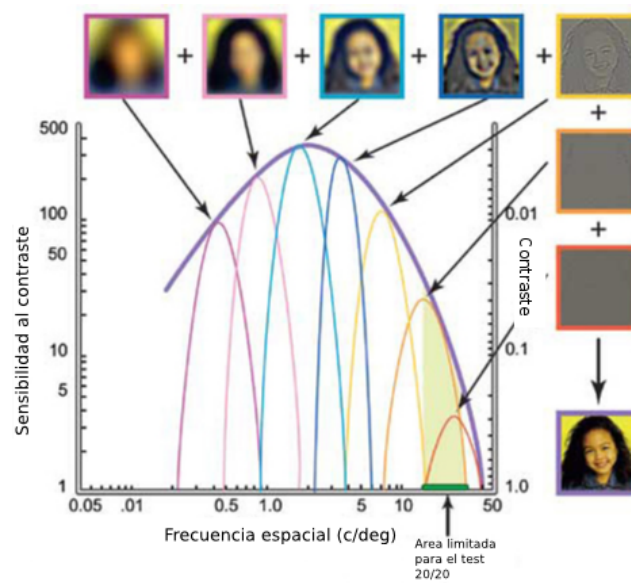


Figura 2.7: Descomposición del proceso de visión

[LPMB09] mostraron que jugadores de video juegos de acción (*core gamers*) tienen mejor percepción de pequeños detalles (altas frecuencias) que personas que no juegan frecuentemente video juegos, demostrando que el SVH puede adaptarse y mejorar, en función de los estímulos cotidianos.

2.4.2. Sensibilidad temporal

El SVH puede distinguir menos detalles cuando el objeto que se proyecta directamente en la fóvea está en movimiento. El resultado de esto es la sensación de emborronado familiar a medida que el objeto pasa frente a el punto de enfoque o cuando el sujeto mueve la cabeza para enfocar otro punto.

Murphy [Mur78] propuso que la incapacidad del ojo de seguir objetos en movimiento de manera precisa es la causa del efecto de emborronado que se percibe, sin embargo, estudios más recientes hechos por Tyler [Tyl85] y Nakayama [Nak90] sugieren que los fotorreceptores limitan la sensibilidad temporal a los detalles, concluyendo que el proceso de detectar movimiento en sí mismo implica la integración de energía en el tiempo durante el movimiento del objeto, y es justamente éste proceso de integración temporal y espacial el que puede destruir la información visual de determinadas características de alta frecuencia del objeto.

Daly [Dal98] extendió el modelo de la CSF para agregar la variable velocidad. Esta versión se puede aplicar bien a imágenes estáticas y enfocadas con la visión central (fóvea), es decir, cual es la percepción del estímulo cuando éste está en

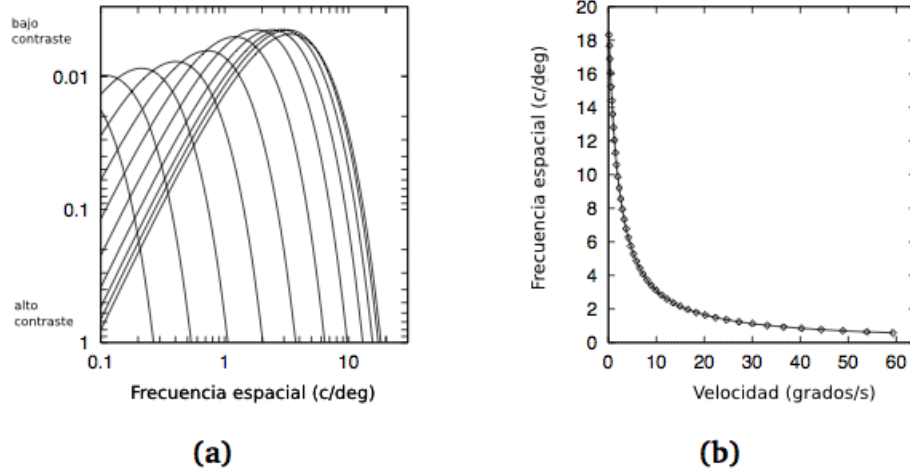


Figura 2.8: (a) CSFs para velocidades de 0.125, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, y 128 grados/s (derecha a izquierda). (b) La frecuencia espacial detectable más alta posible en función de la velocidad.

movimiento y el sujeto está siguiéndolo con su vista. La ecuación (empírica) es la siguiente:

$$CSF(\rho, v_R) = k \cdot c_0 \cdot c_2 \cdot v_R (c_1 2\pi\rho)^2 \exp\left(-\frac{c_1 4\pi\rho}{\rho_{max}}\right) \quad (2.5)$$

con:

$$k = 6,1 + 7,3|\log(c_2 v_R/3)|^3$$

$$\rho_{max} = 45,9/(c_2 v_R + 2)$$

$$c_0 = 1,14; c_1 = 0,67; c_2 = 1,7 \text{ (CRT a } 100\text{cd/m}^2\text{)}$$

En la figura 2.8 se pueden ver dos gráficas presentadas por Reddy [Red97] que muestran como la CSF se desplaza en el eje de la frecuencia espacial al aumentar la velocidad del estímulo. La mayoría de los autores coinciden que al incluir en el modelo el movimiento del estímulo, el resultado es que la CSF sufre desplazamientos y escalados.

2.4.3. Agudeza visual y visión periférica

La visión periférica es una parte del proceso de visión humano que ocurre fuera del centro de la fovea. Comparada con otros animales, la visión periférica de los seres humanos es pobre, especialmente a la hora de distinguir colores y formas. En el sistema de visión humano la densidad de fotorreceptores en la retina es mucho mayor en la parte central que en los bordes.

Los modelos anteriores de *CSF* no se aplican bien cuando se considera la agudeza visual en la visión periférica. Las principales funciones de la visión periférica son, reconocer estructuras y formas conocidas sin la necesidad de enfocarlas, identificar formas similares y movimientos (leyes de la psicología de Gestalt) y proporcionar sensaciones al sistema visual que conforman el trasfondo de la visión detallada.

El SVH no percibe las diferencias visuales como si fuera una función binaria, sino que lo hace como una función de probabilidad en función del ángulo visual en grados, que permite distinguir la característica. Un umbral sobre esa función de probabilidad define si el objeto o la característica es detectada por el sistema o no. Los diferentes modelos propuestos a lo largo de las últimas décadas son intentos de definir estas funciones que describan lo más preciso posible el comportamiento del sistema de visión.

Esta claro que la sensibilidad y agudeza visual decrecen en la periferia del ojo, aunque las hipótesis varían entre los autores. Por un lado, algunos autores [RVN78, VRLN82, KBBdMS78, Kel84] establecen que la sensibilidad al contraste decrece cuando aumenta la excentricidad, y además argumentan que la forma de la curva de tal *CSF* es consistente en todo el campo visual, sufriendo únicamente desplazamientos y escalados. Rovamo y Virsu muestran que la agudeza visual puede ser calculada con precisión para cualquier nivel de excentricidad utilizando un factor de escalado, denominado *Cortical Magnification Factor* (M) o *Radio de Magnificación Cortical*. La figura 2.9 muestra el factor M para las distintas regiones de la visión. El factor M no depende solo de la distancia radial desde la fovea, debido a que la agudeza visual en la periferia del ojo no es una función circular simétrica [RB83], las ecuaciones de la figura 2.9 provistas por Rovamo son las siguientes:

$$\text{Nasal} : M_N = M_0 / (1 + 0,33E + 0,00007E^3) \quad 0 \leq E \leq 60 \text{ deg} \quad (2.6)$$

$$\text{Superior} : M_S = M_0 / (1 + 0,42E + 0,00012E^3) \quad 0 \leq E \leq 45 \text{ deg} \quad (2.7)$$

$$\text{Temporal} : M_T = M_0 / (1 + 0,29E + 0,0000012E^3) \quad 0 \leq E \leq 80 \text{ deg} \quad (2.8)$$

$$\text{Inferior} : M_I = M_0 / (1 + 0,42E + 0,000055E^3) \quad 0 \leq E \leq 60 \text{ deg} \quad (2.9)$$

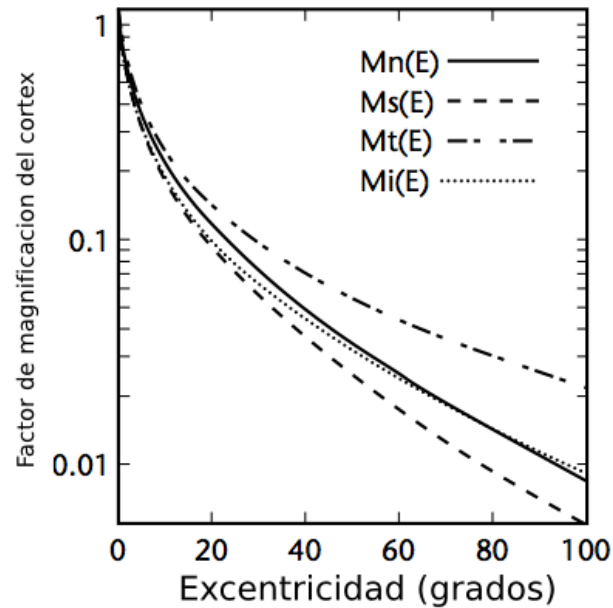


Figura 2.9: Factor de magnificación cortical

Por otra parte, Thibos et. al. [TSB96] entre otros, han analizado los modelos de la *CSF* en la visión periférica y han expuesto diferencias significativas con las *CSF* tradicionales, debido a como funciona el proceso visual en la periferia del ojo. Previo al trabajo de Thibos, se presumía que la resolución espacial en la periferia estaba limitada por el contraste, y por ello podía ser definida de manera similar a los modelos usados para la visión central. Thibos refutó esta hipótesis mediante varias pruebas, y estableció que, la resolución en la periferia está limitada por el muestreo que es capaz de hacer nuestro sistema en esta región del ojo, y es independiente del contraste en un rango de contrastes del 10% al 80%.

2.4.4. Ceguera por falta de atención (Innattentional Blindness)

Uno de los objetivos de los gráficos interactivos por computador es producir imágenes de alta calidad en el menor tiempo posible, sin embargo, muchas veces se dedica gran parte del tiempo a mejorar detalles que el usuario no percibirá. En general, cuando un usuario utiliza una aplicación 3d interactiva conoce de antemano cual es su objetivo o tarea asignada, y reconoce lo objetos relevantes para cumplir con su objetivo, restando importancia inconscientemente al resto de la imagen.

Se puede analizar el proceso de atención visual de dos maneras diferentes, denominadas *top-down* y *bottom-up*, que determinan como los humanos controlan su atención en un instante determinado. El proceso *bottom-up* responde a un estímulo involuntario y muy primitivo, inducido por una característica saliente e inconfundible, como una llama en la oscuridad, una fruta de color rojo en un árbol, etc. Por el contrario, el proceso *top-down* es totalmente voluntario, el sujeto enfoca su atención en objetos relevantes en una escena para poder alcanzar un objetivo de mayor nivel, como contar objetos, apuntar a un objeto, combinar objetos de distintas formas, etc.

Como consecuencia del proceso de visión *top-down*, ocurre un fenómeno denominado “Inattentional Blindness” o “Perceptual Blindness”, que ocurre cuando una persona es incapaz de ver objetos claramente visibles debido a que está concentrada en una tarea específica que causa distracciones mentales sobre el resto de los estímulos visuales.

Cater et al. [CCL02] realizaron experimentos con imágenes estáticas de alta calidad, intentando generar en el usuario un proceso *top-down* de visión, al pedirle que realice una tarea con las imágenes presentadas. Sus resultados muestran que los usuarios no fueron capaces, en la mayoría de los casos, de notar las diferencias entre las mismas imágenes a diferentes resoluciones, porque estaban enfocados en la tarea. Además, por completitud, utilizaron un sistema de seguimiento del ojo (*eye-tracker*) para analizar el patrón de movimiento del usuario y confirmar que la tarea propuesta definitivamente correspondía con el patrón observado del movimiento de los ojos. En [CCW03] Cater et al. modificó su trabajo previo, definiendo un framework de rendering progresivo basado en la percepción, para imágenes de alta calidad, tomando más de 10 segundos para generar una imagen.

En este capítulo se han revisado los fundamentos del sistema visual humano, la percepción visual y los demás factores que se pueden explotar para simplificar la complejidad visual en las aplicaciones interactivas 3d. En el próximo capítulo se introducen algunas de las técnicas más conocidas para acelerar el rendering en aplicaciones interactivas.

Capítulo 3

Técnicas de aceleración 3d en aplicaciones interactivas

Los algoritmos de rendering en tiempo real (*time critical rendering*) utilizan técnicas de LoD, impostores y recortes por visibilidad para alcanzar altas tasas de cuadros por segundo en las aplicaciones [ZPH03]. Además, gracias al gran poder de procesamiento de las tarjetas gráficas modernas, principalmente de *Nvidia y ATI*, están agregando efectos realistas a las imágenes manteniendo altas tasas de cuadros por segundo, como sombras, modelos de iluminación realistas, imágenes de alto rango dinámico de iluminación (*HDR*), y filtros de post-procesamiento en el espacio de la imagen para simular efectos como emborronado por movimiento y emborronado por profundidad de campo. En la sección 3.1 se describe *LoD* o nivel de detalle, una de las técnicas más importantes y tradicionales para reducir el coste de la generación de escenas con diversos modelos de mayas tridimensionales. La sección 3.2, por completitud, describe brevemente una técnica también muy utilizada, el particionado del espacio en las aplicaciones 3d, aunque no se extiende demasiado puesto que no es un tema aplicable en este trabajo.

En la sección 3.3 se definen los impostores y los billboards, técnicas muy utilizadas para acelerar el rendering en aplicaciones interactivas. La sección 3.4 describe las técnicas de emborronado por movimiento y emborronado por profundidad de campo. Aunque estas técnicas se han desarrollado con el principal interés de aumentar el realismo en la síntesis de imagen, sus aplicaciones como técnicas de aceleración 3d son consideradas. Esta sección también incluye una técnica denominada *Motion LoD*, utilizada para acelerar la simulación de multitudes de personajes en aplicaciones interactivas.

Finalmente, el uso de técnicas de escalado de los objetos en la periferia del

usuario es descrito en la sección 3.5.

3.1. Nivel de detalle

El incremento constante de poder del hardware gráfico permite generar cada vez más calidad y realismo en las aplicaciones interactivas 3d. Sin embargo, los modelos también incrementan su complejidad, sus animaciones y simulación de movimientos naturales, que hacen que el hardware disponible siempre tenga que ser utilizado con cuidado para poder alcanzar tasas de cuadros por segundo elevadas. Los algoritmos de simplificación de mayas poligonales y las técnicas de selección de modelos han estado bajo estudio durante los últimos 30 años, y continúan evolucionando día a día para aprovechar al máximo los recursos disponibles.

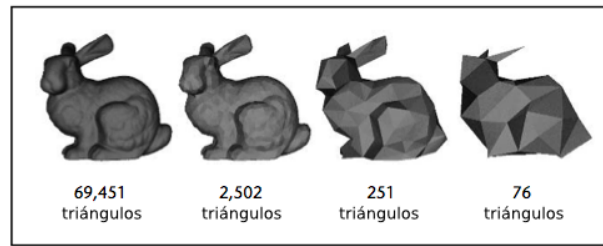
3.1.1. Renderizado fuera de línea basado en la percepción

Los algoritmos de simplificación basados en la percepción intentan aprovechar las limitaciones del sistema visual para evitar generar características o detalles que son imperceptibles para el usuario. Este tipo de algoritmos trabajan utilizando los límites teóricos de la percepción, y se aplican fuera de línea (*off-line*) para crear la imagen final. Obviamente este enfoque no es aplicable a gráficos interactivos con el poder de cómputo actual, aunque quizás en un futuro puedan tomar participación en aplicaciones 3d de tiempo real. Walter et. al. [WHS97], Bolin y Meyer [BM98] son algunos ejemplos de estas aplicaciones.

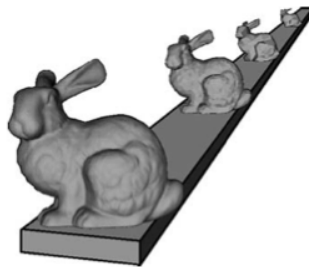
3.1.2. Nivel de detalle estático

El trabajo de Clark [Cla76] estableció las bases de esta técnica denominada *Nivel de Detalle (LoD)*, para regular la complejidad de la escena mediante la simplificación de objetos pequeños o distantes. Aunque han habido mejoras, la idea general se sigue manteniendo, y consiste en crear varias versiones de un modelo 3d de diferentes resoluciones llamadas *LoDs* y de todas las versiones disponibles de cada modelo se escoge una en particular para dibujar en la escena según criterios simples de distancia a la cámara en 3d y/o tamaño del modelo proyectado en pantalla. En la figura 3.1 se puede observar como se aplica este concepto basándose en el criterio del espacio que ocupa el objeto en la pantalla (o la distancia de este a la cámara).

Existen varias técnicas para implementar los cambios de nivel de detalle. Los cambios duros (*hard switching*), cambian el modelo directamente entre un



(a)



(b)

Figura 3.1: Idea básica de la técnica *LoD*, (a) niveles de detalle para el conejo de Stanford, (b) uso de los niveles según su tamaño en pantalla [Lue03].

cuadro y otro. Estos cambios pueden producir artefactos no deseados debido a las discontinuidades entre los modelos, como grandes cambios en la silueta. Este problema es denominado comúnmente *popping*. Una alternativa al cambio directo, es un cambio tardío (*late switching*), que intenta resolver el problema de *popping* realizando los cambios cuando el objeto se encuentra suficientemente lejos para que el usuario no lo detecte. Este último enfoque entra en conflicto directo con el objetivo en sí de la técnica de *LoD*.

Además del problema de *popping* asociado con los cambios directos entre modelos, si un objeto está sobre el umbral de distancia de un cambio, y se aleja y acerca constantemente, se producirán muchos cambios generando un efecto negativo en la experiencia del usuario. En el contexto de técnicas de *LoD*, histéresis es la introducción de un retardo en los cambios, cuando se pasa el umbral de cambio en cualquier dirección. Al utilizar la *histéresis* la frontera de cambio no queda definida por un valor, sino por un intervalo. Ajustando este intervalo correctamente se puede minimizar el efecto no deseado mencionado. Astheimer y Poche [AP94] trabajaron sobre este parámetro, y encontraron que un factor de histéresis del 10 % del tamaño del intervalo de los *LoDs* produce resultados aceptables. En caso que los rangos no sean uniformes, se puede tomar el 10 % de la media para cada par de intervalos.

Algunas representaciones poligonales permiten aplicar una técnica llamada *Geomorphing*, que transforma un modelo poligonal en otro haciendo una interpolación de las posiciones entre los vértices de los modelos a intercambiar. Para esto, cada vértice de uno de los modelos debe tener una correspondencia con algún vértice del otro modelo, lo cual es posible solo con algunas representaciones poligonales como *Progressive Meshes* de Hoppe H [Hop96, Hop97].

Una técnica menos costosa y aplicable a una variada gama de representaciones poligonales es la mezcla entre modelos usando transparencia (*alpha-blending*). Esta técnica debe ser cuidadosamente implementada para evitar hacer ambos modelos transparentes durante la transición. En [GW07] se soluciona este problema, realizando una transición suave entre los modelos y manteniendo siempre uno de los modelos opacos para evitar la transparencia. Esta técnica fue implementada en las experimentaciones de este trabajo para realizar los cambios entre diferentes modelos.

Entre los algoritmos más importantes para la simplificación de los modelos se pueden mencionar:

Vertex Clustering[RB93]: Este algoritmo trabaja individualmente con los triángulos y los vértices, asignando una importancia a cada vértice del modelo y simplificando los vértices menos importantes. No importa el tipo de topología que tenga el modelo. Es uno de los algoritmos de simplificación más rápidos, aunque no asegura que se preserve la topología y no tiene criterios de error para poder medir la calidad del modelo resultante con respecto al original. Uno de los puntos claves, es el criterio para la selección de la importancia de los vértices.

Decimation Algorithm[SZL*92]: Uno de los algoritmos más utilizados en la comunidad científica para simplificación de modelos en medicina, implementado en la librería *VTK*, este algoritmo se basa en la eliminación de vértices redundantes y tiene un orden lineal en el número de vértices del modelo. El modelo resultante después de una ejecución, es un subconjunto del modelo original. Esto presenta ventajas y desventajas. Como ventajas, se pueden reusar normales y coordenadas de texturas del modelo anterior. La principal desventaja es la flexibilidad, puesto que no se generan nuevas posiciones en la simplificación, siempre se reusan los vértices originales. Acepta mayas con topologías non-manifold, ignorando los vértices que cumplen con esta condición.

Image Driven Simplification (IDS)[LT00]: El algoritmo presentado por Lindstrom y Turk realiza la simplificación guiándose por las imágenes resultantes que generaría el modelo después de la operación de simplificación. Al

evaluar la calidad del modelo simplificado comparando los resultados en el espacio de imagen, y no por criterios geométricos, los resultados de este algoritmo son de gran calidad visual. El proceso de evaluación de la calidad en espacio de imagen también introduce un retardo considerable en el tiempo total comparado con cualquier otro algoritmo para generar *LoDs*.

Quadric Error Metric (QEM)[GH97]: Este algoritmo mide la distorsión que sufre la superficie del modelo al realizar una operación de simplificación. Guiando el proceso y limitando las simplificaciones por errores geométricos se logran resultados de gran calidad. Es uno de los algoritmos más utilizados para generar *LoD* estático, y es relativamente sencillo de implementar.

3.1.3. *LoD* dinámico y dependiente de la vista

Una de las críticas al enfoque del *LoD* estático es depender de unos pocos niveles de detalle para representar todas las posibles vistas de un objeto, limitando de alguna forma el grado en el cual se pueden aplicar métricas basadas en percepción. Esto ocurre en particular cuando los objetos a simplificar ocupan gran parte de la pantalla, y están en la visión central del usuario. El cambio de contraste en las siluetas de los objetos por ejemplo, es fácilmente detectable por el usuario cuando está observando directamente al modelo simplificado, puesto que este fue simplificado sin considerar una vista en particular, sino simplemente un criterio uniforme de número de triángulos objetivo.

Los métodos de simplificación dinámicos ofrecen una solución a este problema. En vez de calcular una serie de *LoDs* estáticos, los sistemas dependientes de la vista construyen una estructura de datos en base al modelo original desde la cual pueden extraer el nivel de detalle deseado en tiempo de ejecución. De esta manera, se puede ajustar la resolución en diferentes partes del modelo donde se cree que el usuario puede tener mayor sensibilidad a las altas frecuencias, como en las siluetas o simplemente en la zona donde esté enfocada el área central de la visión si se dispone de algún dispositivo de seguimiento del ojo. Existen muchos algoritmos de simplificación dinámicos, entre los más importantes se encuentran los trabajos de Hoppe, Luebke y Xia. [Hop97, LE97, XV96]. Estos algoritmos utilizan una jerarquía de operaciones de fusión de vértices que pueden ser aplicadas o removidas en tiempo de ejecución, generando modelos no uniformes en tiempo real.

3.2. Particionado espacial

Una técnica muy común en la implementación de mundos virtuales es el particionado del espacio en regiones que permitan determinar en todo momento con una serie de tests muy simples si una región está dentro de la vista de la cámara o no. De esta manera, la geometría asociada a las regiones no visibles no es enviada ni procesada hasta que sea visible por la cámara. Las técnicas más conocidas son *Octrees* y *Binary Space Partitioning trees*. Esta técnica no es aplicable en el presente trabajo, y no se profundizará más en sus detalles.

3.3. Impostores

Los video juegos y aplicaciones gráficas 3d presentan un entorno ideal para utilizar LoDs basados puramente en una imagen, también conocidos como *impostores (imposters)* [SS96]. La técnica en principio es simple, reemplazar un modelo 3d por una imagen o textura mapeada directamente en un polígono plano.

Previo a las aplicaciones 3d, lo más parecido a los *impostores* fueron los *sprites*, que eran *mapas de bits* dibujados en la pantalla del ordenador por hardware especializado. En el famoso juego *Mortal Kombat* se usaron *sprites* creados digitalmente en base a fotografías reales de personas practicando artes marciales.

Los *impostores* pueden ser calculados en tiempo de diseño (off-line) restringiendo la vista en la que estos se pueden presentar, o dinámicamente cuando la aplicación lo necesite. La generación dinámica tiene la ventaja de que el modelo 3d a reemplazar queda orientado perfectamente hacia la cámara, aunque requiere de alguna característica adicional del hardware como *render-to-texture* o poder acceder al *buffer* de color para recuperar la imagen generada del modelo 3d como si se hubiera proyectado normalmente, y guardarla en una textura para posteriores usos. La desventaja, es que al moverse la cámara o si el *impostor* representa un objeto animado, hay que actualizar la textura para que el usuario no note la diferencia. El mismo *impostor* es reutilizado, hasta que el error visual supera un umbral arbitrario, y el *impostor* es reemplazado. En general, se genera un número finito de *impostores* para un modelo, y se utiliza el que esté más cercano a la orientación real que tendría el modelo desde la vista actual.

Un tipo especial de *impostores* son los *billboards*, que mantienen su orientación siempre perpendicular hacia la cámara. Ejemplos de estos últimos se pueden observar en árboles, efectos volumétricos de partículas (humo, fuego, etc) y otras características en los terrenos.

Cuando los *impostores* se utilizan como reemplazo de objetos complejos es

aconsejable que la textura rectangular tenga definido un canal *alpha* o de transparencia, de al menos un (1) bit de profundidad para delimitar la silueta del objeto a representar, y dejar que el fondo de la escena se mezcle con el objeto apropiadamente.

Cuando se utilizan *impostores* para un modelo que puede variar su distancia a la cámara en grandes cantidades (en coordenadas del mundo 3d), se recomienda el uso de *mipmaps* (la misma textura a múltiples resoluciones) para no utilizar texturas de alta resolución con *impostores* muy pequeños y evitar así efectos no deseados de aliasing y ruido al escalar las texturas.

Con el Hardware moderno, están surgiendo ideas innovadoras que mejoran considerablemente las características de los *impostores*, como “*True Impostors*” presentado Risser [Ris07] en la publicación de *Nvidia GPU Gems 3*, que utilizan técnicas de ray-casting, o de ray-tracing (2 implementaciones diferentes) y mapas de elevación para generar impostores en función de la vista y en tiempo real. La figura 3.2 muestra una escena sintética generada con esta técnica, con 150.000 rocas 3d (con *true impostors*) en el anillo de Júpiter, a 30 cuadros por segundo.

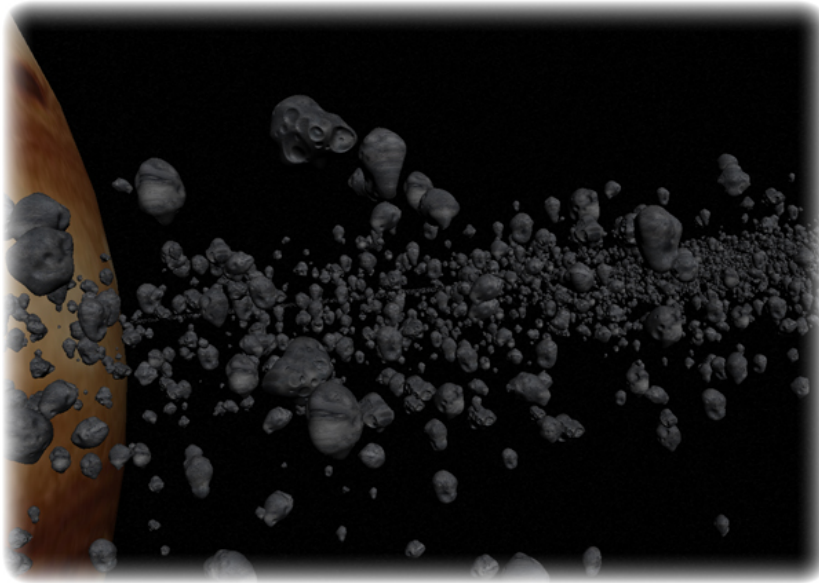


Figura 3.2: *True Impostors*, Júpiter con 150.000 rocas en su anillo

3.4. Otras técnicas de *LoD*

3.4.1. *LoD* en función de la velocidad

Al escoger el nivel de detalle en función de la velocidad, el criterio de selección del *LoD* a utilizar para un objeto es la velocidad relativa del objeto respecto al movimiento del ojo del usuario. Aún hoy en día, en las aplicaciones interactivas 3d de uso masivo, no se dispone de sistemas de seguimiento del ojo robustos y económicos para facilitar la implementación de técnicas como esta, que dependen en gran parte de conocer donde está mirando el usuario a cada instante.

Una correcta implementación de algoritmos de *LoD* que consideren movimiento debe tener en cuenta flujos de movimiento complejos, como por ejemplo simplificar el fondo de la escena cuando el usuario está siguiendo con su vista un objeto en movimiento, o en caso contrario, simplificar un objeto en movimiento cuando el usuario observa de manera estática algún punto del fondo de la escena [Lue03].

Como alternativa a la falta de disponibilidad de sistemas de seguimiento del ojo, la mayoría de los algoritmos realizan alguna simplificación y asumen el comportamiento del usuario. En general, para poder medir como se comporta el algoritmo, y qué percibe el usuario de las simplificaciones que se realizan, se asume que este último está mirando al centro de la pantalla, y la aplicación se organiza de tal forma que siempre se miden las velocidades y distancias respecto a este punto central. Funkhouser y Séquin [FS93] consideran la variable de velocidad en su algoritmo, reduciendo el nivel de detalle de los objetos en una cantidad proporcional al radio entre la velocidad aparente del objeto y el tamaño promedio de sus polígonos (un enfoque simple, pero efectivo). Hitchner y McGreevy [HM93] incorporan el término de velocidad para definir el *interés* relativo de un objeto, y en función de su *interés*, reducen o aumentan el nivel de detalle del objeto. El *interés* en función de la velocidad lo definen como el cociente entre una constante arbitraria y la velocidad del objeto. La velocidad absoluta del objeto la calculan en función de la posición del objeto entre dos imágenes consecutivas, asumiendo que tienen una tasa de dibujado constante (*fixed frame-rate*).

En implementaciones actuales de video juegos, se utiliza una técnica denominada emborronado por movimiento (*motion blur*) para agregar realismo a las escenas y dar al usuario la percepción del movimiento que está acostumbrado a experimentar en la vida real. Se puede considerar que estas técnicas de emborronado por movimiento están realizando simplificaciones del nivel de detalle por velocidad de manera implícita. Dichas simplificaciones pueden ser realizadas en el espacio del objeto o mundo 3d, o en espacio de imagen. Un grupo de investi-

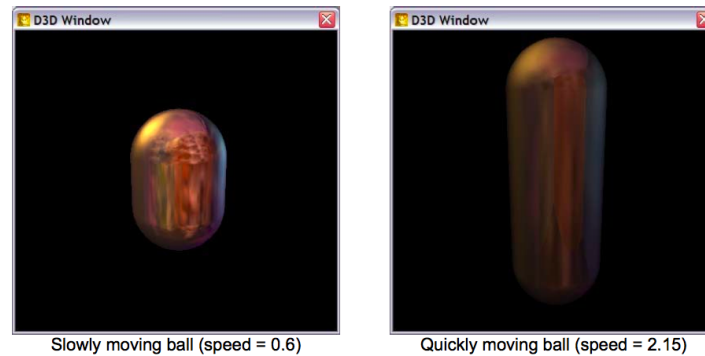


Figura 3.3: Simulación de emborronado en la demo de *Animusic Pipe Dream de ATI*. Deformación de los modelos en función de la velocidad

gación de *ATI* [TBI] realizó una implementación de emborronado en el espacio de los objetos, modificando los modelos poligonales con un *vertex shader* y modificando también la apariencia final con un *fragment shader* para que parezca que los objetos se mueven a gran velocidad. Esta simulación del movimiento se hace en un solo frame, sin necesidad de usar el *buffer de acumulación*.

Este cambio del modelo poligonal para generar un efecto más realístico para el usuario final sugiere que también es viable eliminar características de alta resolución de tal modelo, debido a que se está simulando un efecto físico de movimiento en el cual dichas características serían imperceptibles para el usuario. En esta demo [TBI], *ATI* generó una versión en tiempo real de la animación “*Animusic Pipe Dream*” publicada en SIGGRAPH 2001. La figura 3.3 muestra como se deforma la geometría de una bola a diferentes velocidades.

3.4.2. *LoD* en función de la profundidad de campo

El nivel de detalle basado en profundidad de campo selecciona una determinada representación de un objeto comparando, en el espacio del mundo 3d, su distancia con la distancia del objeto en el que han sido enfocados los ojos. En este caso, conocer exactamente donde está mirando el usuario es crucial para poder aplicar la técnica, puesto que el objeto que está desenfocado puede estar muy cerca del objeto enfocado en el espacio de proyección de la imagen. El área de enfoque de la visión, es conocida como área de fusión de Panum, y como muestra la figura 3.4, los objetos que están por detrás (más lejos) y por delante de este área se ven fuera de foco. Es importante recordar que, aunque la distancia de enfoque sea constante, objetos en la periferia también serán percibidos con menor detalle, por la pérdida de agudeza visual en la periferia mencionada anteriormente. Por esto, el área de aplicación de esta técnica está reducida solo a la vecindad del punto donde el usuario ha enfocado su mirada.

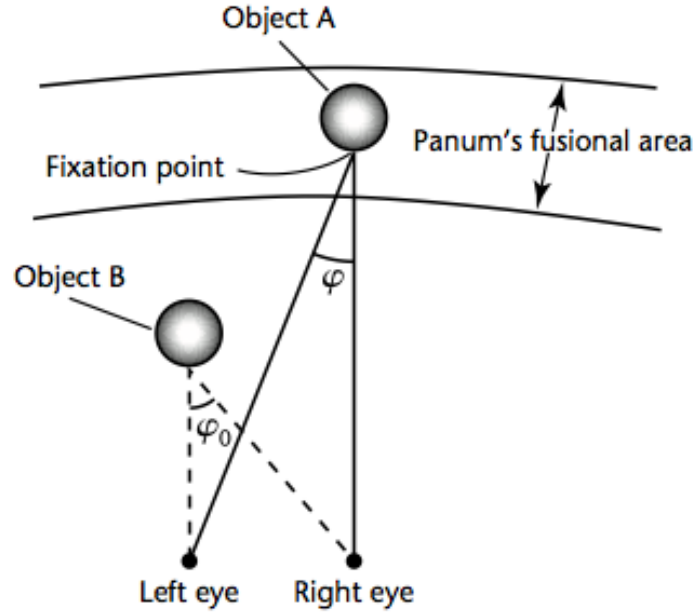


Figura 3.4: Área de fusión de Panum. El objeto A está enfocado, dentro del área de Panum [OYT96]

En [OYT96], Ohshima et al. sugieren la reducción del nivel de detalle fuera del área de enfoque, utilizando técnicas tradicionales de *LoD*. Su trabajo incluye criterios de visión periférica, velocidad y profundidad de campo para seleccionar el nivel de detalle adecuado para cada objeto. Para esta sección, solo el criterio de profundidad de campo es de interés. En su trabajo, utilizaron la siguiente ecuación:

$$h(\Delta\phi) = \begin{cases} 1 & \text{con } 0 \leq \Delta\phi \leq b \\ \exp(-\frac{\Delta\phi-b}{c_3}) & \text{con } b < \Delta\phi \end{cases} \quad (3.1)$$

siendo $\Delta\phi = |\phi - \phi_0|$, b un umbral del área de enfoque y c_3 una constante arbitraria de 0.63. Como se puede observar en la figura 3.4, para seleccionar el modelo del objeto de B se utiliza esta función $h(\Delta\phi)$, siendo 1 la máxima calidad (los objetos muy próximos entre sí) o un valor menor a 1 para el resto de las resoluciones disponibles. Si se utiliza *LoD* estático, como la función es continua, es necesario definir los intervalos de esta que corresponden a cada resolución disponible para el modelo. Esta distribución de los diferentes *LoDs* puede ser ajustada en cada sistema en particular.

Por otro lado, también se han considerado alternativas a este enfoque, trabajando sobre el problema en el espacio de la imagen generada. En [BFSC04], los



Figura 3.5: Efecto de profundidad de campo generado con un filtro de difusión anisotrópica y la información del *Z-Buffer* [BFSC04]. En la imagen, el punto de enfoque ha sido puesto en la segunda fila de coches.

autores proponen la simulación del efecto de profundidad de campo, mediante un filtro de difusión anisotrópica implementado en la *GPU* con un *shader de fragmentos*. Para ello, utilizan la información del *Z-buffer* para conocer la profundidad de los fragmentos resultantes en la imagen final. La figura 3.5 muestra como se obtiene el efecto de desenfoque deseado en una escena de automóviles a diferentes distancias. En dicho sistema, las pruebas se realizaron estableciendo el foco del usuario de manera arbitraria, aunque si se pudiera incluir información adicional sobre donde está mirando el usuario en un instante determinado, como en el trabajo de Ohshima mencionado, se podrían utilizar modelos de menor resolución en las zonas donde luego el filtro generará el efecto de desenfoque, sin que el usuario note la diferencia y disminuyendo drásticamente el tiempo de generación de la imagen.

Para tener una idea de la importancia que se le está dando a este tema en particular, la figura 3.6 muestra una captura de pantalla de una demo en tiempo real publicada por *ATI* para promocionar las tarjetas *Radeon X1800*. Se puede ver claramente como se ha generado el efecto de profundidad de campo para dar una sensación de realismo a la escena y para simplificar la síntesis en la zona desenfocada.

3.4.3. *LoD* aplicado a las animaciones de humanoides

La simulación de grandes multitudes de personas siempre ha sido un desafío en muchas áreas de la informática gráfica y la industria cinematográfica. Paseos virtuales, dibujos animados, guerras entre ejércitos, etc., todos hacen uso de alguna técnica de simulación de multitudes. Si se consideran las aplicaciones en tiempo real, surgen muchas limitaciones y problemas de escalabilidad para

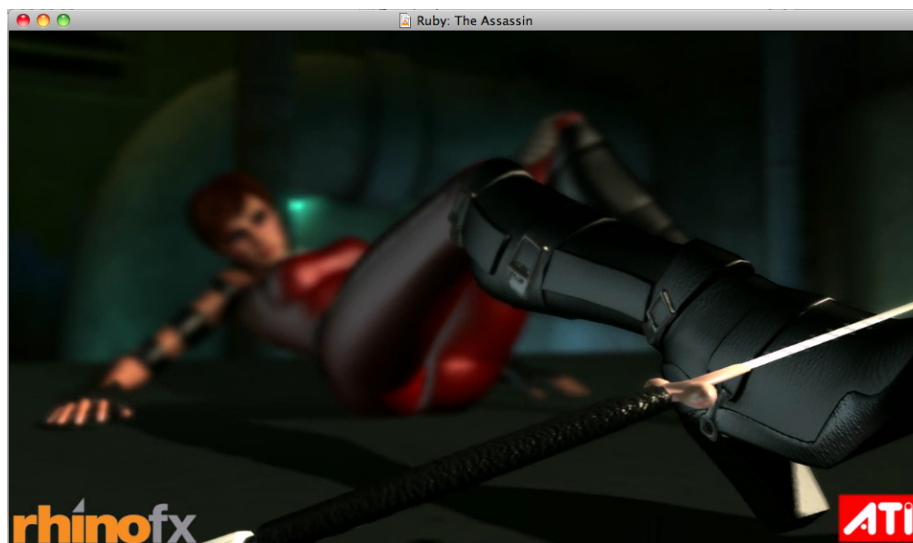


Figura 3.6: Demo en tiempo real de *ATI* mostrando su último personaje, *Ruby*, creado por la compañía para promocionar sus productos.

generar multitudes de personajes en tiempo real. El uso de impostores es un intento de reducción del alto costo computacional que implican tales simulaciones, aunque padece de problemas de precisión y detección de colisiones, manejo de memoria complejo, tratamiento de animaciones pequeñas como la respiración, e interactividad en el movimiento de los personajes [ABT00].

Ahn y Wahn [AW04] propusieron la simplificación del movimiento de las articulaciones, como solución escalable a la simulación masiva de personajes, denominando esta técnica *Motion Level of Detail*. En general, el movimiento es capturado de personas reales, usando variadas técnicas con marcadores infrarojos o de radio puestos sobre las articulaciones del cuerpo (*joints*) y luego trasladado a personajes virtuales compuestos de "skin" (maya poligonal y texturas) y esqueleto [MHK06]. La idea de "simplificación del movimiento" se define como la reducción del movimiento de n -articulaciones a un movimiento de m -articulaciones, con $m < n$. La técnica mejora la velocidad de las animaciones, conservando las características del movimiento original. Los detalles de la técnica escapan al alcance de este trabajo, pero su aplicación en el contexto de percepción visual en la periferia es de gran interés, puesto que los autores han utilizado criterios de tamaño, distancia y excentricidad para la selección del modelo a utilizar en cada personaje de forma individual. En la figura 3.7 se pueden observar 7 modelos para un personaje de un robot, indicando con colores diferentes cada parte del modelo que es articulada. El último modelo no tiene articulación alguna.

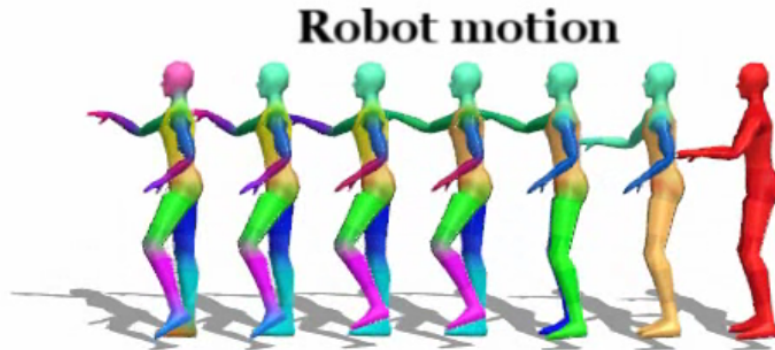


Figura 3.7: Motion LoD para modelado de humanoides

3.5. Escalado en la periferia

Las simplificaciones de modelos 3d por disminución de polígonos es una de las alternativas para simplificar la escena y disminuir los requerimientos de computación en aplicaciones de tiempo real, sobre todo cuando el número de polígonos es elevado y se requiere un gran ancho de banda para transmitir los vértices de la memoria principal a memoria de la tarjeta gráfica.

Existen otros límites que deben ser tenidos en cuenta al momento de realizar aplicaciones 3d de tiempo real, debido a que el proceso de la tarjeta gráfica es como una tubería de procesos, acelerar solo una parte no significa maximizar la performance global de la aplicación[CW03]. Durante la etapa de rasterización, en la tarjeta gráfica, el tiempo de relleno de polígonos que ocupan gran parte de la pantalla puede ser un cuello de botella importante para la performance global de la aplicación.

Escalado en la periferia es una técnica que reduce el tamaño de los objetos en la periferia, de manera continua, aprovechando la deficiencia del sistema de visión en dicha zona para disminuir el tiempo de dibujado de los modelos. En la sección 2.4.3 se definieron las funciones M del factor de magnificación del cortex para cada región del campo visual; estas funciones pueden ser utilizadas como un factor de escala continuo para los objetos en la periferia del ojo.

Como sugiere Luebke [Lue03], para realizar el escalado de los modelos en la periferia se puede considerar la función de la ecuación 2.8, (que es la más conservadora, puesto que escala menos que las otras para cualquier excentricidad posible), e incluso despreciar el término cúbico, introduciendo un error menor al 1 % cuando el modelo se encuentra a más de 100 grados de excentricidad.

Capítulo 4

Análisis y definición del problema

A continuación se discuten las motivaciones principales y consideraciones de este trabajo en los tres aspectos claves que abarca, técnicas de aceleración 3d, percepción visual y atención visual (*ceguera por falta de atención*).

4.1. Técnicas de aceleración 3d

Las técnicas de aceleración estudiadas en este trabajo son hoy en día muy populares en aplicaciones interactivas 3d como video juegos y mundos virtuales. Como se vio, algunas técnicas están evolucionando rápidamente en paralelo con los avances del hardware, como las implementaciones de impostores independientes de la vista [Ris07] y técnicas de emborronado por movimiento [BFSC04] y por profundidad de campo implementadas en la GPU. Sin embargo, la técnica más popular y aprovechada todavía sigue siendo *LoD*, que ha logrado una gran madurez en el área académica y de investigación y se ha diversificado y especializado para cada necesidad en particular. En aplicaciones 3d interactivas, donde la tasa de generación de cuadros es elevada, los efectos especiales cada vez más populares, y el tiempo de *CPU y GPU* dedicado a tareas no gráficas se incrementa, el uso de *LoD* estático sigue siendo el modo predominante para simplificar la complejidad de las escenas.

La técnica escogida para investigar es por lo tanto *LoD* estático, y durante el experimento estudiaremos la capacidad del usuario de detectar los cambios de *LoD*.

4.2. Percepción visual

El estudio de las características de la percepción visual humana y sus modelos se ha desarrollado mucho en los últimos 30 años, desde los trabajos fundacionales como los de Mannos y Sakrison [MS74] hasta los últimos descubrimientos sobre la forma de la *CSF* entre los jugadores de video juegos y los que no lo hacen [LPMB09]. Con los avances de investigación en visión por computador y los avances tecnológicos de los sistemas de seguimiento de la vista y la cabeza, la percepción visual es un factor que está tomando una posición predominante en muchas áreas de la ciencia. Por poner un ejemplo, la compañía *SR Link*, que produce sistemas de seguimiento del ojo registra 1134 publicaciones [SRL09] en revistas indexadas de estudios realizados con sus productos por investigadores de todo el mundo.

Se han realizado muchos trabajos respecto a la percepción visual en gráficos por computador. Sullivan [OHM*04] presenta una revisión del estado del arte muy completa. Para el fenómeno psicológico que se desea analizar, la ceguera por falta de atención, resulta de mayor interés hacerlo en aplicaciones interactivas y en la región de la periferia del sistema de visión, puesto que es el caso más común en el que ocurre este fenómeno y su aplicabilidad en video juegos y mundos virtuales tendrá más relevancia.

Otra cuestión que genera interés y motivación en el estudio de la región periférica de la visión es el incremento constante en el tamaño de los dispositivos de presentación como pantallas, monitores y proyectores, y su decremento de precios para el usuario hogareño.

Considerando entonces la periferia del sistema de visión como área de interés, a continuación se describen las opciones estudiadas para obtener la información de la visión periférica durante el experimento.

Existen al menos dos formas de explotar la falta de agudeza visual en la periferia en el momento de hacer el rendering. Una es combinar con el dispositivo de presentación (monitor, proyector, etc) el uso de otro dispositivo que permita hacer el seguimiento del ojo o la cabeza como un *eye-tracker* o *head-tracker*. Con la información provista por tal sistema, la aplicación o el video juego puede reducir la calidad de los modelos utilizando alguna función que asigne diferentes *LoDs* según la distancia respecto al punto del observador. Sistemas como este, fueron utilizados en [LMYM01], donde Loschky probó que puede generar imágenes filtradas (filtros de emborronado en el espacio de la imagen) que contienen alta resolución solo en un radio de 4.1 grados del punto donde el usuario está observando, y este último no es capaz de distinguir entre una imagen filtrada con baja resolución fuera del radio de 4.1 grados y una sin filtrar con solo alta resolución. Unos años mas tarde, Loschky [LW07] muestra que para que esta

técnica se pueda aplicar en tiempo real sin que el usuario detecte los cambios constantes de resolución en la imagen, esta se debe actualizar entre 5 a 80 milisegundos, por lo tanto el sistema de seguimiento también debe hacerlo con esa frecuencia. Desde un punto de vista práctico, en las aplicaciones de consumo masivo como video juegos, el uso de un sistema de seguimiento todavía no es viable, tecnológica ni económicamente. Varios jugadores mirando la misma pantalla, tiempo de calibración del sistema e integración transparente con los motores 3d del mercado, son algunas de las barreras que todavía tienen estos costosos sistemas que no han sido adoptados por las compañías de entretenimiento, pero sí para uso militar o comercial en aplicaciones muy específicas o estudios psicológicos.

El otro enfoque o alternativa a sistemas de seguimiento de la vista, es intentar usar información implícita definida por la tarea específica a realizar por el usuario. Por ejemplo, en un juego de FPS (*First-Person Shooter*) cuando el usuario se encuentra disparando a un objetivo en movimiento durante un par de segundos es altamente probable que este tenga su foco de atención en este punto de interés en la pantalla, porque de lo contrario no podrá alcanzar su objetivo debido a la dinámica y velocidad de la interacción en la aplicación.

Está claro que no es posible, sin un sistema de seguimiento del ojo, poder determinar en todo momento cuales son los puntos de interés del usuario en la escena. Por lo tanto, en este trabajo se asumirá que sí se conoce el punto de interés y se utilizará esta información para implementar el criterio de selección de *LoD*.

En el desarrollo del experimento, en el siguiente capítulo, se explica la metodología utilizada para determinar aproximadamente en que parte de la pantalla está mirando el usuario.

4.3. Factores psicológicos

La atención visual siempre ha sido un tema muy investigado en la psicología y otras ciencias cognitivas. En síntesis de imagen digital, las características de la percepción visual y la atención visual están tomando mayor importancia en la última década [CCL02, CCW03, SDL*05, GB03]. En [ENY06] se describe uno de los primeros intentos de analizar la atención visual del usuario en video juegos modernos. En dicho trabajo, los autores llevaron a cabo un experimento con 6 personas jugando a 2 video juegos actuales (*Legacy of Kain Blood Omen II* y *Halo II*). Además, registraron toda la actividad visual de los jugadores con un sistema de seguimiento de la vista. Se filmaron las partidas de los jugadores, y se analizaron posteriormente con cuidado durante 6 meses para poder determinar que características de cada video juego atrajeron la atención de la vista del

usuario. En este caso, los autores investigaron aspectos visuales del proceso de visión *bottom-up* como color y movimiento, y *top-down* como tareas de alto nivel u objetivos durante el juego. Las conclusiones fueron que, en la mayoría de los casos, los jugadores se vieron afectados inconscientemente por estímulos visuales del proceso de visión *bottom-up* como cambios de color con alto contraste en los objetos y el movimiento, pero también se observó un dominio del proceso *top-down* sobre el *bottom-up* durante las partidas. Los autores explican, en una de las secuencias analizadas, que cuando los jugadores tenían que salir de una habitación, lo primero que miraban eran las puertas y ventanas, y al no poder utilizarlas, notaban una región en la pared con mayor contraste (color), pero no tan intuitiva, que era la salida real. Este fenómeno muestra claramente un proceso *top-down* inducido por el tipo de juego (aventuras).

El análisis de la atención visual en video juegos, y su posible explotación para acelerar el tiempo de síntesis está en sus comienzos. Además, es claro que el incremento del uso de técnicas de visión por computador, para seguimiento de la vista, de la cabeza, y detección de gestos físicos del usuario (Wii Motion-Plus [Nin09], Project Natal [Mic09] y Sony PlayStation MotionController), está abriendo un sin fin de oportunidades para nuevos desarrollos e ideas en el mundo de los gráficos interactivos.

Por todo lo expuesto, la motivación de este trabajo es estudiar y medir la percepción visual en la periferia bajo una condición inducida de *ceguera por falta de atención* e intentar reducir el nivel de detalle de los modelos 3d sin que el usuario note los cambios.

Para generar el estado de *ceguera por falta de atención* se desarrolló un video juego interactivo, que requiere un alto grado de interacción por parte del usuario quien debe utilizar el teclado como dispositivo de control y debe alcanzar una meta de apuntar y disparar a un objetivo en movimiento por toda la pantalla.

En este capítulo se expusieron las motivaciones principales del trabajo y se analizaron las alternativas para llevarlo a cabo. En el siguiente capítulo se describe el estudio realizado y los resultados obtenidos.

Capítulo 5

Estudios y resultados

Para realizar el estudio, se desarrollo un video juego 3d interactivo utilizando las librerías OpenSceneGraph y OpenAL, junto con material gratuito o libre para uso académico disponible en la web.

5.1. La aplicación

El desarrollo de la aplicación tomó bastante tiempo de este trabajo, se hizo en C++ sobre una plataforma *Linux/i386*, usando la librería de código abierto OpenSceneGraph estudiada previamente en el curso *Programación Gráfica* dictado por el profesor Francisco Abad en el máster *IARFID*.

5.1.1. Guión del juego

En el juego, el usuario debe conducir un tanque por una ciudad muy grande, controlándolo con el teclado para doblar, acelerar y frenar. El usuario debe dirigirse a determinados puntos en la ciudad, que son indicados con una réplica idéntica del tanque que éste conduce, pero de color verde y transparente. Para encontrar dichos puntos el usuario debe hacer uso de la ayuda que le proporciona una brújula de orientación en la parte inferior de la pantalla (un *Head-Up Display o HUD*), como muestra la figura 5.1. El jugador tiene un tiempo para realizar la tarea, como se indica en la parte inferior derecha de la figura 5.1, así como un número limitado de disparos (*AMMO*) y un número de objetivos o estatuas a las que debe proteger. Cada vez que el jugador llega a uno de los objetivos, debe disparar al *villano* para proteger la estatua. Mientras el jugador dispara al *villano* es cuando se hacen las pruebas de cambios de *LoD* en la estatua y se toman todos los registros para su posterior análisis. Luego de una serie de disparos, el *villano* escapa hasta el próximo objetivo, el tiempo se



Figura 5.1: Imagen de la aplicación desarrollada, manejando por la ciudad. El tanque verde es el punto donde debe colocarse el jugador, y desde allí disparar al *villano* que está atacando la estatua.

vuelve a reiniciar para que el usuario vuelva al modo de *conducir* y busque el nuevo punto en la ciudad.

Además, en la figura se puede ver una barra verde en la esquina superior derecha, que representa la energía del *villano* durante el juego, y va decrementando después de cada disparo que acierta el usuario.

Una vez que el usuario llega al punto indicado, debe posicionar su tanque *justo encima* del tanque transparente, y presionar una tecla para cambiar del modo “*conducir*” al modo “*disparo*” en el juego. En el modo *disparo*, la cámara que estaba detrás del tanque mientras conducía por la ciudad, se coloca (mediante una animación) detrás del cañón, y ahora el usuario utiliza las mismas teclas de dirección que usaba para conducir el tanque, para mover el cañón y apuntar en la pantalla.

El cambio de modo le presenta al usuario una escena especial, mostrando los edificios, un modelo 3d en la parte superior de un edificio representando una estatua gigante y un personaje (*el villano*) volando cerca de la estatua e intentando destruirla disparando proyectiles. Para saber donde está apuntando el tanque con el cañón, se presenta una mirilla en el punto de intersección entre la línea del cañón y el objeto que esté delante. La mirilla va cambiando de tamaño en función de la distancia del objeto al que se está apuntando, que pueden ser

edificios, árboles y el *villano*. Este efecto de escalado de la mirilla le permite al usuario tener una sensación de profundidad y una noción de las distancias en el mundo virtual. Las figuras 5.2a y 5.2b muestran al tanque disparando al *villano*.

5.1.2. Punto de interés y arco visual

El objetivo del usuario en el juego, es proteger las estatuas que están encima de los edificios, atacando al *villano* en cada escena hasta destruirlo. Para ello, debe dispararle apuntando el cañón mientras este se mueve por toda la pantalla. El *villano* es nuestro *punto de interés (POI)*, es decir, el área de la pantalla donde asumiremos que está mirando el usuario en todo momento. Esta idea de usar un *punto de interés* y asumir que el usuario estará mirando siempre hacia él representa una gran simplificación frente al uso de un sistema de seguimiento de la vista.

Para conocer en que pixel se proyecta un determinado vértice que está en coordenadas del mundo 3d, se debe hacer un proceso similar al que hace la tarjeta gráfica. Se deben tomar las matrices que transforman el modelo, la matriz de la cámara y las matrices de proyección, y multiplicarlas por el vértice en cuestión, para obtener una coordenada en 2d en la pantalla.

Mediante la proyección de los centroides del *POI* y del modelo 3d, se pueden conocer sus coordenadas en el espacio de la pantalla, y medir la distancia en pixels entre ellos. Luego, si se conoce la distancia a la que está el ojo del usuario de la pantalla, y la distancia en unidades equivalentes entre los pixels del *POI* y del modelo 3d, aplicando pitágoras se puede conocer el arco visual desde el *POI* (que estará en la *visión central*) hasta el modelo 3d (en la *visión periférica*).

La figura 5.3 muestra la pantalla inicial del juego, donde se le explica al jugador el objetivo, y como debe proceder. Al usuario se le pide explícitamente que siempre debe mirar con su visión central al *villano o POI*, que es muy importante para el estudio.

Mientras el *villano* se mueve por toda la pantalla, el modelo 3d que representa la estatua sobre uno de los edificios, irá cambiando de resolución como se explicará en los siguientes apartados, para realizar diferentes experimentos.

5.1.3. Detecciones de cambios en la periferia

Consideramos que la dinámica de la actividad, y la dificultad de tener que apuntar constantemente el cañón al *villano* con el teclado, genera en el jugador el efecto de *ceguera por falta de atención* en la periferia, y en particular con respecto al modelo 3d de la estatua. Esto nos permite hacer el estudio sobre como



(a) Modelo 3d del conejo con alta resolución



(b) Modelo 3d del conejo con menor resolución

Figura 5.2: Modo disparo, atacando al *villano*

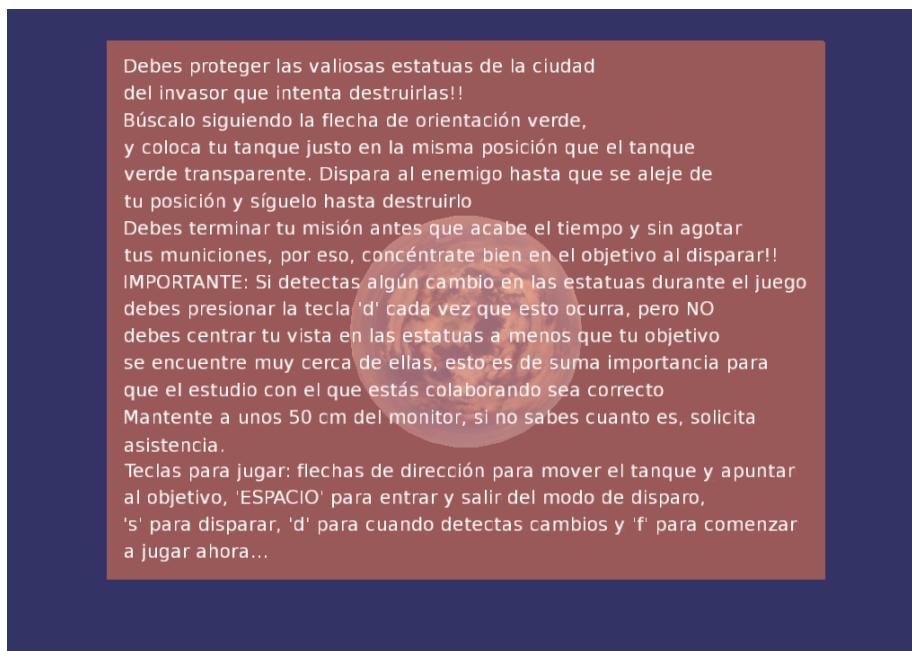


Figura 5.3: Pantalla inicial del juego, con las instrucciones

percibe el usuario los diferentes cambios de resolución en la estatua mientras intenta disparar al *villano*.

Por último, y muy importante, el usuario debe registrar las detecciones de los cambios de *LoD* en la periferia presionando una tecla cada vez que crea que esto ha ocurrido. Realizando esto *durante* el juego, y no posteriormente con una encuesta, se obtiene información detallada de cada instante en el que se detectó un cambio, que cambio fue exactamente, a que distancia del *POI* e incluso en que coordenadas de la pantalla. Con toda esta información generada durante el juego luego se analizan los resultados.

5.1.4. Jugabilidad

Debido a la diversidad de la población que analizamos, consideramos que el juego debía adaptarse en dificultad a las diferentes clases de jugadores, desde los que nunca juegan hasta los jugadores que tienen gran coordinación psicomotriz y como se mencionó anteriormente, tienen sus sentidos de percepción más entrenados [LPMB09]. Intentando aliviar estas diferencias, se tomó la siguiente consideración al respecto. El juego realiza la adaptación dinámica del tamaño del área de impacto del *POI*. De esta manera, si un usuario acierta rápidamente varios disparos al *POI* el área de impacto real se va decrementando tanto que eventualmente fallará, generando un estímulo y un desafío en el jugador, y evi-

	1	2	3	4	5	6
Armadillo	21000	10000	5000	2500	1000	675
Conejo	3000	1500	700	340	160	80
Dragón	7000	5000	3750	3000	2200	1500

Cuadro 5.1: *LoDs* construidos con MeshLab para cada modelo

tando que este se aburra y empiece a mirar otras zonas de la pantalla porque el juego le resulta muy fácil. Por el contrario, para tener en consideración a los jugadores no tan experimentados, ocurre un proceso inverso. Si el jugador no acierta, el área de impacto se agranda, hasta que será muy fácil acertarle y el usuario recibirá una recompensa por haber intentado repetidas veces pegarle al objetivo, y no se cansará rápido de jugar porque el juego le parezca muy difícil.

5.2. Modelos 3d y técnica de transición para el LoD estático

Para las pruebas de la percepción a cambios de resolución en el juego, se utilizaron 3 modelos del repositorio 3d de Stanford [LGCP] bastante conocidos, el Conejo, el Dragón y el Armadillo. La figura 5.4 muestra los 3 modelos utilizados. El esquema de *LoD* es estático, y se definieron 6 niveles de detalle para cada modelo. Las diferentes versiones de cada modelo fueron construidas usando el algoritmo *Quadric Edge Collapse Decimation*, un derivado del *Quadric Error Metric*, con la herramienta MeshLab [CCR08].

Los modelos, en su estado original, tienen una gran resolución, pero nuestro interés para este trabajo es evaluar el uso de modelos con un poligonado moderado o bajo, aplicable en video juegos 3d. El cuadro 5.1 presenta el número de triángulos de los diferentes *LoDs* construidos para las pruebas.

5.2.1. Unpopping

La técnica de transición entre *LoDs* implementada es la descrita en el trabajo [GW07], que sugiere utilizar *alpha-blending* entre los modelos sin exponer las caras ocultas de estos y estableciendo siempre uno de los modelos opacos durante la transición. Esta simple técnica mejora considerablemente la calidad de la transición, no permite que se vea el fondo detrás de los modelos durante el cambio y evita el efecto de *popping* mencionado anteriormente.

La figura 5.5a muestra la idea básica de esta técnica. Se define LoD_1 como el *LoD* actual, y LoD_2 como el *LoD* al que se está cambiando. El tiempo de transición utilizado en la implementación de este trabajo es de 300 ms. La figura



Figura 5.4: Modelos del repositorio de Stanford, el Conejo, el Armadillo y el Dragón

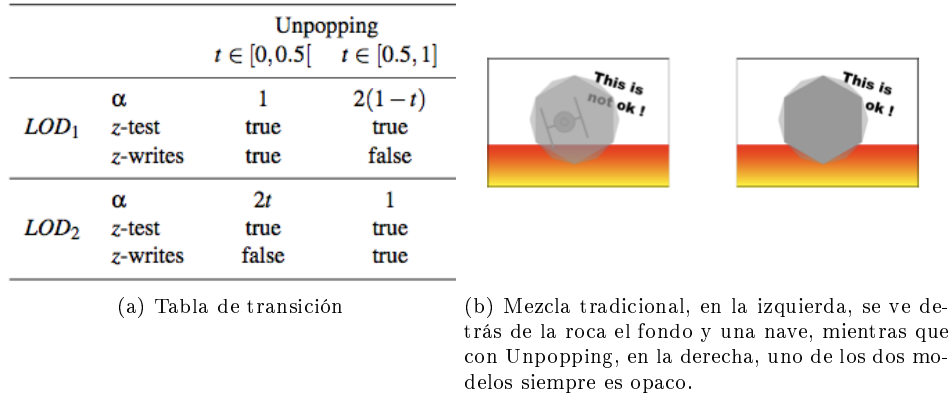


Figura 5.5: Unpopping

5.5a se puede interpretar de la siguiente manera, durante la mitad del tiempo de transición, 150 ms, el LoD_1 se mantiene opaco, y se dibuja normalmente, mientras que el LoD_2 se dibuja desde “totalmente transparente” ($alpha=0$ y $t=0$) hasta opaco ($alpha=1$ y $t=0.5$), luego se invierten los roles, y durante los 150 ms restantes se deja el LoD_2 opaco y se hace transparente de a poco el LoD_1 hasta que este desaparece. Siempre hay un LoD opaco durante la transición, y este LoD opaco siempre escribe en el $Z-Buffer$ de la tarjeta gráfica, evitando que se vea el interior (las caras ocultas) del LoD transparente.

En la figura 5.5b se puede ver la diferencia entre usar *alpha-blending* tradicional ($alpha$ y $1-alpha$) y usar la técnica implementada.

5.3. Pruebas

Las pruebas se realizaron en la *Universidad Nacional del Comahue*, en Argentina, con 20 estudiantes de la carrera *Licenciatura en Ciencias de la Computación* de la *Facultad de Economía y Administración*. Todos los estudiantes presentaban visión normal o corregida con lentes. Las pruebas se hicieron en un aula aislada, con iluminación artificial de tubos fluorescentes, en un ordenador con un monitor de 15.4 pulgadas, y resolución de 1024x768. Los participantes se sentaron a 50 centímetros del monitor y utilizaron una silla regulable para ajustar las diferencias de altura.

5.3.1. Primera prueba

La primera prueba realizada fue experimental y sirvió para definir otras ideas en el trabajo. Para poder probar el sistema desarrollado, se definió la función

de la ecuación 5.1, para realizar el mapeo entre diferentes *LoDs* y la distancia del *POI* a los modelos 3d.

Cuando se diseñó esta función, se consideró que el ojo tiene más agudeza visual en la parte central, y esta va decayendo hacia la periferia pero de una manera no lineal. Por eso se consideró una función que asigne un número mayor de modelos cerca de la parte central y menos en la periferia. La función tiene 3 parámetros para ajustar al problema en particular, y su forma general se puede ver en la figura 5.6.

Además, como muestra la figura 5.6 la función se puede ajustar con la variable *scale*, que aún es experimental en este estudio, para modelar lo mejor posible las distancias óptimas de los cambios de resolución en la periferia bajo las condiciones consideradas en este trabajo.

En la figura 5.6 se muestran las resoluciones del modelo del conejo que corresponden con el número de *LoD* de la función.

Para esta prueba, se utilizó el factor de escala $scale = 100$.

$$\frac{(1,0 - \log_{10}(1,0 + (scale * (numLods - x)/numLods))) * maxDistance}{\log_{10}(scale + 1)} \quad (5.1)$$

donde:

numLods: Número de *LoDs* utilizados en el modelo.

maxDistance: Máxima distancia considerada en pixels en la pantalla.

scale: factor de escala que da forma a la curva, permitiendo variar desde una forma lineal hasta una forma exponencial.

Durante esta primera prueba, cuando el usuario se enfrenta al *villano*, este último se mueve por toda la pantalla en dirección aleatoria, obligando al usuario a seguirlo con la mirilla del cañón e intentar destruirlo durante unos minutos. Al moverse por toda la pantalla, los cambios de *LoDs* son bastante frecuentes, y el usuario debe registrar en el sistema cada vez que detecta un cambio presionado una tecla designada para ello.

5.3.2. Segunda prueba

En esta prueba, se cambió el comportamiento de la aplicación drásticamente. Se modificó el algoritmo de cambio de *LoDs*, intentando encontrar una aproximación de una función que defina, para cada resolución, las zonas o distancias en las que el usuario definitivamente detecta los cambios, las zonas que las detecciones pueden alternar, y las zonas donde definitivamente no detecta ningún

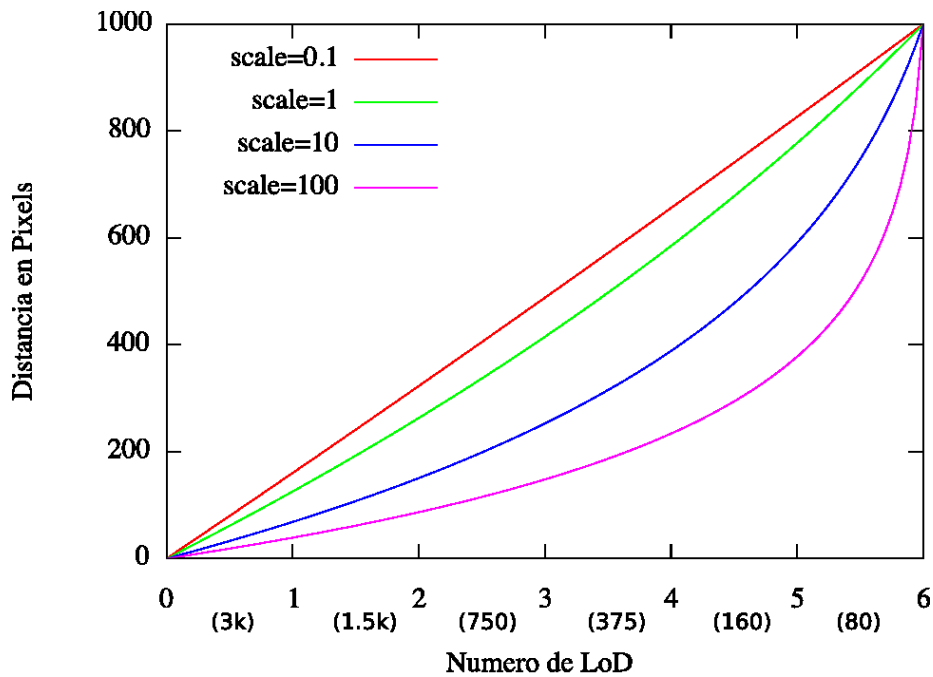


Figura 5.6: Función para mapear los LoDs en función de la distancia periférica con el *POI*. Además, se muestran las resoluciones para el Conejo que se usan en cada intervalo de distancia.

cambio. Para ello, se quitó la función fija utilizada en la prueba anterior, y se desarrolló el siguiente esquema de trabajo:

Cuando el usuario cambia del modo *conducir* al modo *disparo*, se encuentra con una escena similar a la prueba anterior, y el *villano (POI)* está “atacando” a la estatua (al modelo 3d) desde una dirección muy próxima. El usuario comienza a disparar al *POI*, y este empieza a moverse en forma concéntrica alrededor del *LoD*, en trayectorias aleatorias pero que siempre mantienen al *POI* dentro de un intervalo de distancias respecto del *LoD*. La figura 5.7 ejemplifica este comportamiento, inicialmente el *POI* se encuentra en el anillo más cercano al *LoD*, luego pasa al anillo inmediatamente más lejos, y así sucesivamente.

Cuando el *POI* se encuentra en el anillo más cercano al *LoD*, la resolución del modelo 3d es la máxima disponible, y cuando el *POI* comienza a moverse se ejecuta el algoritmo 5.1 para definir su nivel de detalle.

Al finalizar con la ejecución, el resultado es un conjunto de datos que indica para cada intervalo de distancias respecto del *POI* (*supuestamente la visión central del jugador*) el cambio de resolución que fué detectado, y los cambios que no lo fueron.

Algorithm 5.1 algoritmo de ajuste de distancias

-
1. Realizar una reducción de resolución de UN NIVEL
 2. Esperar 2 segundos
 3. Si el usuario detecta el cambio
 - a. Registrar la detección (LoD,distancia)
 - b. Aumentar la resolución UN NIVEL (al nivel anterior) ignorando cualquier detección de este cambio.
 - c. Luego de unos disparos, incrementar la distancia del POI al anillo siguiente (ver figura 5.7)
 - d. Ir al paso 1
 4. Si el usuario no detecta el cambio:
 - a. Registrar la NO detección (LoD,distancia)
 - b. Ir al paso 1
-

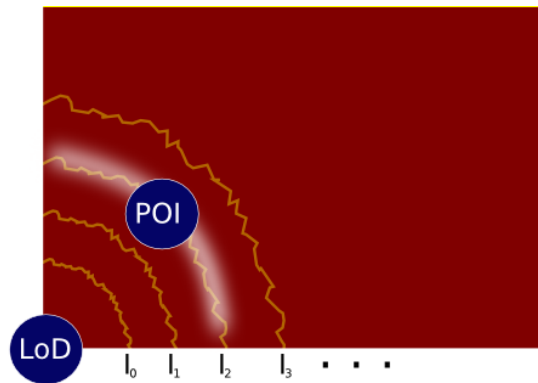


Figura 5.7: Movimiento del POI respecto del LoD en la segunda prueba. I_0, I_1, \dots son las trayectorias concéntricas que hace el POI alrededor del LoD.

5.3.3. Tercera prueba

En esta prueba se empleó la misma mecánica para realizar las mediciones que en la prueba anterior, pero en este caso se desea determinar si al realizar cambios de resolución más grandes entre los modelos el usuario detecta con mayor frecuencia dichos cambios, o si los detecta a distancias más cercanas. Desde un punto de vista teórico, al cambiar entre modelos que son “menos” parecidos geoméricamente y visualmente, el cambio de contraste será mayor para el ojo, incluso en la periferia, y deberíamos detectar más fácilmente estos cambios. Considerando nuestra suposición de que estamos generando un efecto psicológico de *ceguera por falta de atención* en el jugador, queremos ver si esto modifica el resultado que se espera en la percepción al aumentar el contraste en los cambios.

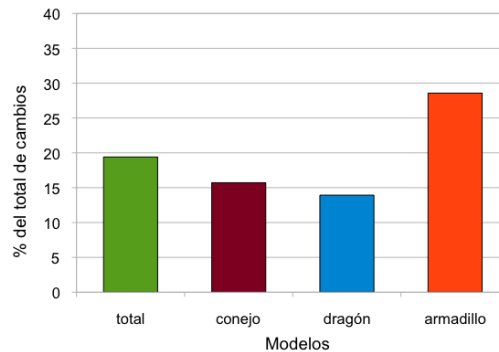


Figura 5.8: Porcentaje de detecciones de cambios de LoD (de un promedio de 42 cambios por usuario)

Sean las mayas poligonales A, B y C correspondientes a un modelo 3d y ordenadas de mayor a menor resolución. Queremos determinar si dado un cambio de *LoD* que termina en la maya C, iniciando en la maya A y pasando por la B, es detectado a distancias similares que si el cambio fuera directamente de la maya A a la C. Este análisis puede ser utilizado, por ejemplo, para determinar entre que resoluciones es necesario poner más mayas intermedias porque el usuario nota los cambios.

5.4. Resultados

En la primera prueba, se utilizó una función que mapea los diferentes *LoDs* en intervalos de distancias, concentrando la mayor cantidad de modelos cerca de la visión central y pocos en la periferia. La figura 5.8 muestra que, en promedio en cada partida de un jugador ocurren 42 cambios de resolución, y de estos 42 cambios fueron detectados menos del 20 %.

Luego de la simplificación de los modelos usando el MeshLab, una de las versiones del modelo del Armadillo quedó con una diferencia de luminosidad significativa respecto del resto de las versiones del mismo modelo. Esto se debe a que las normales no fueron calculadas correctamente por el programa. Decidimos dejar el modelo con mayor luminosidad para ver como afectaba los resultados esa pequeña diferencia. En las figuras 5.8 y 5.9 se puede notar claramente como aumentó el número de detecciones para este modelo simplemente por una pequeña diferencia en luminosidad. Esto indica claramente la importancia de los modelos de percepción basados en el contraste previamente vistos.

En la figura 5.9 se ve una descomposición de *todas* las detecciones en las distintas zonas de la visión periférica. Esta estadística considera todos los juga-

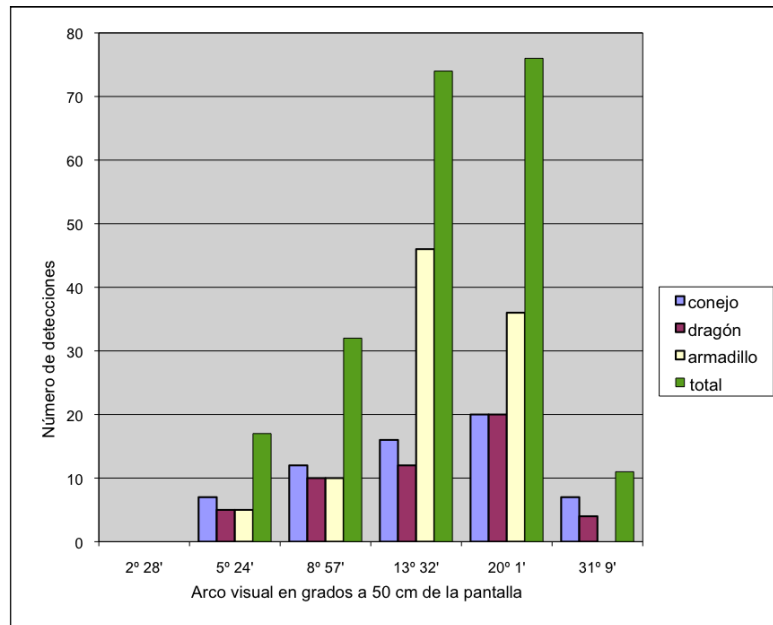


Figura 5.9: Número total de detecciones hechas por 20 jugadores, de un total de 860 cambios de resolución

dores y todas las detecciones. Los resultados son coherentes con el hecho de que nuestra función de mapeo dedica más modelos a la zona central y menos en la periferia, por eso se ve un incremento de detecciones a medida que aumenta el arco visual, porque hay cada vez menos modelos y los cambios son más grandes entre un modelo y otro, aumentando el contraste entre un cambio y otro.

Es importante notar esto, *hay más cambios en la zona central, pero menos detecciones, y hay menos cambios en la zona periférica pero muchas más detecciones*. Aunque esto parece contradictorio con la teoría de la percepción visual en la periferia, creemos que se debe a que los cambios de la periferia son mucho más evidentes, debido a que usamos menos modelos en esa zona y los saltos de resolución de un modelo son más notables para los usuarios. Por otro lado, también se puede ver que la técnica implementada para el intercambio entre modelos fue muy efectiva, por el bajo número de detecciones en la zona cercana a la visión central.

En los resultados, discriminamos las detecciones del usuario, diferenciando cuando éste detecta un incremento de resolución o una disminución de resolución. La figura 5.10 muestra los resultados. No podemos concluir en base a los resultados que haya una diferencia de percepción a los cambios si se trata de un aumento o de una disminución de resolución. Por otra parte, la ausencia de detecciones en la zona de 2 grados se debe a que la resolución es tan alta que el usuario no nota cuando ocurre un cambio en esa zona. Por último, en la zona

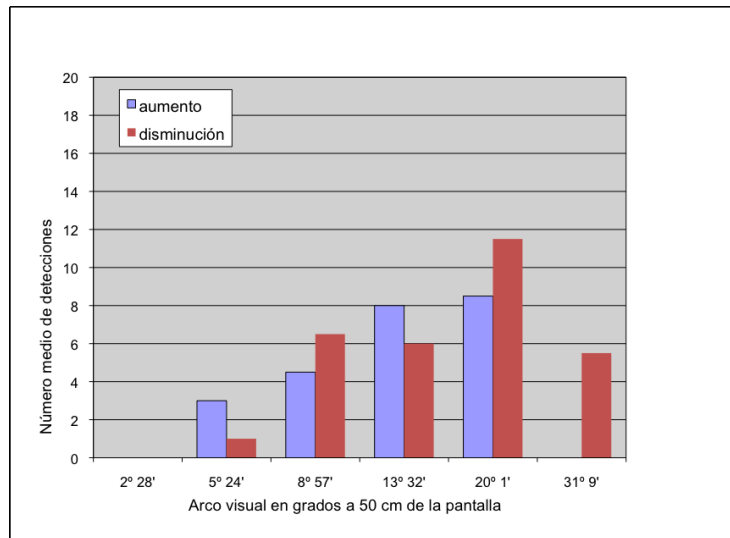


Figura 5.10: Número medio de detecciones al aumentar y disminuir la resolución, considerando el conejo y el dragón

de los 31 grados, no hay detecciones por aumento porque no hay aumentos de resolución en esa zona, solo un decremento a la menor calidad disponible.

Los resultados de la segunda prueba sirven para determinar a que distancias hubieron detecciones para cada resolución utilizada y en que distancias no hubieron detecciones. Las figuras 5.11 y 5.12 para el conejo y el dragón respectivamente muestran en grados de arco visual (escala izquierda) y en pixels (escala derecha) la distancia media y su desvío estándar de las detecciones para cada resolución, así como las distancias y desvíos para los cambios que no fueron detectados para cada resolución. La zona por debajo de la curva de cambios detectados (la curva inferior) es una zona en la que probablemente se detecten los cambios de resolución, por el contrario, si se realizan los cambios utilizando como función la curva superior de “no-detecciones”, es altamente probable que los cambios no sean detectados. Es interesante notar que más allá de un ángulo de arco visual de 13 grados (o unos 400 pixels en la configuración utilizada) los usuarios no son capaces de notar los cambios en el modelo del conejo de 80 triángulos y en el dragón con 1500 triángulos.

La última prueba realizada consistió en medir las detecciones cuando los cambios de resolución se hacían de a pasos más grandes. Para esto, lo que se hizo fue quitar algunos modelos intermedios y repetir el proceso de la segunda prueba. La figura 5.13 muestra los resultados, donde los saltos de resolución *normales* son los mismos que se usaron en la prueba 2, y los saltos de resolución *grandes* son cambios entre la secuencia de resoluciones (3k,750,160,80) triángulos. Los resultados confirman la suposición previa, que se esperaba que para una

determinada resolución las detecciones sean a menor distancia, puesto que los cambios de contraste ahora serían más grandes.

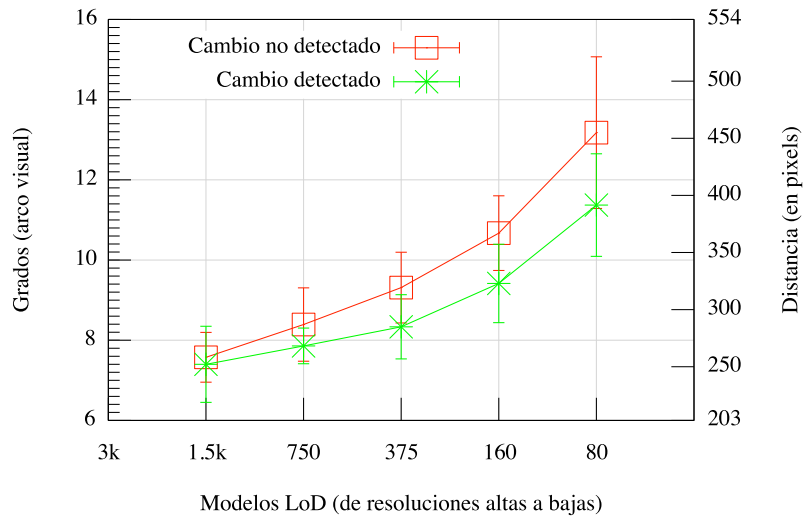


Figura 5.11: Número medio de detecciones y no-detecciones de cambios de *LoD* en la periferia tomando solo el conejo.

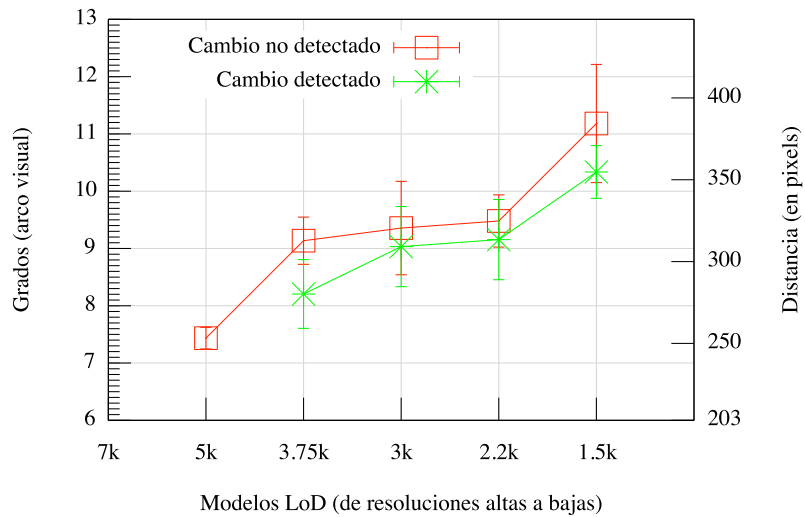


Figura 5.12: Número medio de detecciones y no-detecciones de cambios de *LoD* en la periferia tomando solo el dragón.

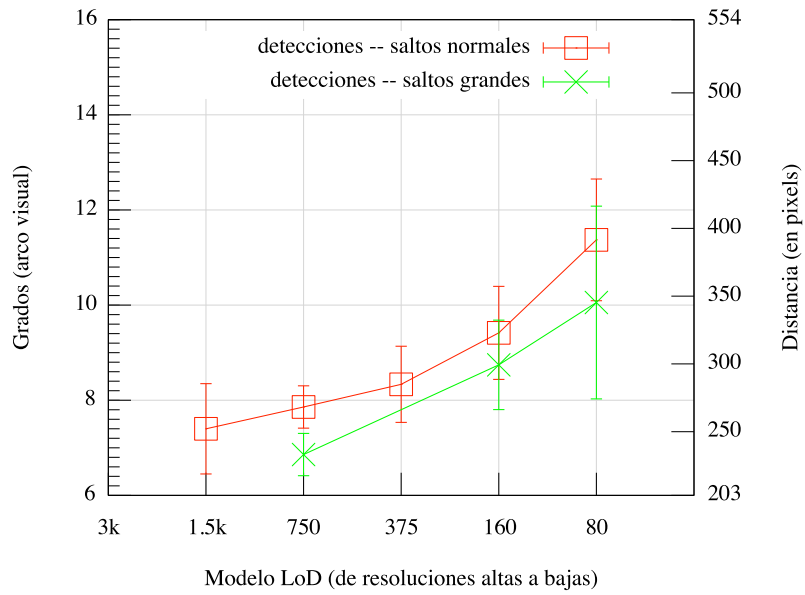


Figura 5.13: Comparación de las distancias medias de detección de cambios de *LoD* entre saltos de resolución normales y saltos de resolución grandes. Los cambios grandes se realizan entre las siguientes resoluciones (3k, 750, 160, 80).

Capítulo 6

Conclusiones y trabajo futuro

Conclusiones

El objetivo planteado en este trabajo era poder realizar mediciones de la percepción visual en la periferia mediante el uso de una aplicación interactiva 3d similar a un video juego, que presente al usuario un desafío adicional más allá del estudio en sí mismo. Una vez construida la herramienta para hacer las pruebas, y escogida la técnica de *LoD estático* como el medio a través del cual se medirían las detecciones de cambios en la periferia, la decisión más importante fue cómo implementar las pruebas descritas en el capítulo 5. La mayoría de los estudios sobre visión periférica revisados se realizaron en ambientes controlados, con sistemas de seguimiento de la vista, con imágenes estáticas y sin tareas adicionales, por lo tanto, en nuestro caso no se disponía de un método de referencia para llevar adelante el estudio. Debido a esta falta de experiencias previas en la bibliografía que se parecieran al experimento que queríamos hacer se tomaron algunas decisiones un tanto arbitrarias. Una de ellas fue la elección de las diferentes resoluciones para cada modelo. Es por ello que se determinaron saltos proporcionales (mitad de resolución) que pudieran abarcar una amplia gama de resoluciones para poder averiguar como se comportaba el SVH bajo determinadas condiciones.

La primera aproximación, de buscar un modelo matemático que permita definir con precisión los intervalos en los cuales usar cada *LoD* del modelo 3d, sin que los cambios entre estos sean detectados por el usuario, puede ser un avance en la dirección correcta. Sin embargo, el análisis de los resultados experimentales de la prueba 2 indica que no resulta sencillo ajustar el modelo propuesto a los resultados. Con el modelo planteado se detectaron solo el 20% de los cambios de resolución, por lo cual esta línea merece seguir siendo estudiada para intentar reducir más esa tasa de detecciones, redefiniendo el modelo con más datos

experimentales.

En la prueba 2, creemos que el algoritmo utilizado para encontrar las distancias óptimas de cambio de *LoD* fue muy efectivo. De todas formas, es necesario generalizar la técnica para contemplar diferentes modelos (no solo el Conejo, el Dragón y el Armadillo), diferentes resoluciones, más texturas, etc. Esto se propone más adelante en el trabajo futuro.

La tercera prueba fue de gran importancia, porque permitió entender como la granularidad de los cambios de resolución puede afectar la percepción. Además, estos resultados nos indican que es necesario evaluar otras técnicas de *LoD* como *Progressive Meshes* para aumentar la densidad de niveles para un modelo y evitar así cualquier percepción de cambios, o incluso evaluar la combinación de técnicas estáticas con técnicas dinámicas, según las zonas de la visión periférica.

Trabajo futuro

Los siguientes puntos de trabajo futuro pueden generar líneas de trabajo totalmente independientes, aunque creemos que todos pueden contribuir de alguna manera al estudio de la percepción visual en la periferia y la ceguera por falta de atención.

1. El emborronado por movimiento que se produce cada vez que el ojo se mueve puede causar un enmascaramiento del fenómeno que se intenta caracterizar, la *ceguera por falta de atención* en la periferia. Para aislar correctamente el problema habría que dejar el *POI* quieto durante las pruebas, aunque de una prueba a otra estuviera en diferentes posiciones, para seguir generando el efecto de *ceguera por falta de atención*, pero eliminando la variable movimiento.
2. El esquema de *LoD* estático utilizado parece resultar un poco limitado para medir con precisión la percepción visual en la periferia. Aunque en los video juegos esta sea una de las opciones predilectas para implementar *LoD*, por su simpleza y bajo costo computacional en tiempo de ejecución, deseamos realizar pruebas con *LoD* dinámico y también realizando mezcla entre transiciones de *LoD* estático para ciertas regiones y *LoD* dinámico para otras. Para ello, se estudiará la técnica de *Progressive Mesh* de Hoppe [Hop96] para introducirla en la aplicación desarrollada y volver a repetir los experimentos. Usando *LoD* dinámico esperamos reducir mucho las detecciones, aunque a un costo computacional que puede ser prohibitivo en un video juego moderno. De todas formas, el costo computacional debe ser evaluado y comparado con la técnica de *LoD* estático, para ver si su diferencia se hará despreciable con el constante aumento de las velocidades de las CPUs y las GPUs.

3. Para poder generalizar más los resultados obtenidos con la prueba número 2, es necesario utilizar diferentes modelos 3d y sus versiones de diferentes resoluciones deben poder ser comparables por algún criterio de similitud, geométrico o en el espacio de imagen. La distancia de Hausdorff es una técnica comúnmente utilizada para medir la diferencia entre dos conjuntos de puntos, en particular en nuestro caso entre dos mayas de polígonos en R^3 . Si trabajamos con diferentes resoluciones de los modelos 3d, que sean equivalentes en cuanto a la distancia de Hausdorff, estaremos generalizando mucho el proceso y los resultados. Esto implicaría que después un desarrollador podría usar esta métrica para construir sus *LoDs* y obtener resultados similares a los del estudio. Otra métrica que se puede utilizar en el espacio de imagen es medir la frecuencia más pequeña que cambia entre dos resoluciones, como plantea Reddy[Red97] en su trabajo, aunque esta métrica no es independiente de la vista y debe ser recalculada cada vez que la cámara se mueve.
4. Contrario a la propuesta de trabajo del punto 1, también creemos que los límites de percepción visual en un jugador están muy lejos de los modelos teóricos, que se desarrollaron bajo condiciones ideales y sin ningún tipo de distracción. Como se planteó en este trabajo, esto no es realista, y creemos que aun la aplicación desarrollada para este trabajo carece de mucho realismo en comparación con la cantidad de información visual que proporciona un juego moderno. El diseño e implementación de un video juego no es tarea sencilla, ni de una persona. Si deseamos realizar más estudios en un ambiente más complejo, creemos que es necesario buscar proyectos maduros que se adapten a nuestras necesidades, que sean Open Source para poder acceder al código fuente, y modificarlos en la medida que sea posible para hacer más estudios.
5. Por último, resulta de gran interés entender cuando se genera efectivamente un efecto de *ceguera por falta de atención* en el usuario, debido a que no será posible en todo momento asumir esto en un video juego. Desde la teoría de los video juegos, se plantea investigar que factores de los video juegos generan este fenómeno, desde aspectos visuales, la dificultad cognitiva de la tarea, habilidad psicomotriz necesaria, motivación real del usuario para lograr el objetivo, retribuciones por alcanzar los objetivos (económicas, de estatus, etc), etc. Si se pueden determinar con precisión estos factores, luego para el programador será más fácil escoger los momentos exactos en que puede hacer más simplificaciones de sus escenas esperando que las probabilidades de que el usuario note dicha reducción de calidad sean mínimas.

Bibliografía

- [ABT00] AUBEL A., BOULIC R., THALMANN D.: Real-time display of virtual humans: Levels of details and impostors. *IEEE Transactions on Circuits and Systems for Video Technology* 10, 2 (2000), 207–217.
- [AP94] ASTHEIMER P., POCHE L.: Level-of-detail generation and its application in virtual reality. In *Virtual reality software & technology: proceedings of the VRST'94 Conference, 23-26 August 1994, Singapore* (1994), World Scientific Pub Co Inc, p. 299.
- [AW04] AHN J., WOHN K.: Motion level-of-detail: A simplification method on crowd scene. *Proceedings Computer Animation and Social Agents (CASA)* (2004), 129–137.
- [BFSC04] BERTALMIO M., FORT P., SANCHEZ-CRESPO D.: Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on* (2004), pp. 767–773.
- [BM98] BOLIN M., MEYER G.: A perceptually based adaptive sampling algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998), ACM New York, NY, USA, pp. 299–309.
- [CCL02] CATER K., CHALMERS A., LEDDA P.: Selective quality rendering by exploiting human inattentive blindness: looking but not seeing. In *Proceedings of the ACM symposium on Virtual reality software and technology* (2002), ACM New York, NY, USA, pp. 17–24.
- [CCR08] CIGNONI P., CORSINI M., RANZUGLIA G.: Meshlab: an open-source 3d mesh processing system. *ERCIM News* 73 (2008), 45–46.

- [CCW03] CATER K., CHALMERS A., WARD G.: Detail to attention: Exploiting visual tasks for selective rendering. In *Proceedings of the 14th Eurographics workshop on Rendering* (2003), Eurographics Association, p. 280.
- [Cla76] CLARK J.: Hierarchical geometric models for visible surface algorithms. *ACM* (1976).
- [CR68] CAMPBELL F., ROBSON J.: Application of Fourier analysis to the visibility of gratings. *The Journal of Physiology* 197, 3 (1968), 551.
- [CW03] CEBENOYAN C., WLOKA M.: Optimizing the graphics pipeline. In *Game Developer's Conference* (2003).
- [Dal98] DALY S.: Engineering observations from spatiovelocity and spatiotemporal visual models. *Vision Models and Applications to Image and Video Processing* (1998), 179–200.
- [ECR66] ENROTH-CUGELL C., ROBSON J.: The contrast sensitivity of retinal ganglion cells of the cat. *The Journal of Physiology* 187, 3 (1966), 517.
- [ENY06] EL-NASR M., YAN S.: Visual attention in 3D video games. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology* (2006), ACM, p. 22.
- [FS93] FUNKHOUSER T., SÉQUIN C.: Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM New York, NY, USA, pp. 247–254.
- [GB03] GREEN C., BAVELIER D.: Action video game modifies visual selective attention. *Nature* 423, 6939 (2003), 534–537.
- [GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., p. 216.
- [Gla95] GLASSNER A.: *Principles of digital image synthesis*. Morgan Kaufmann Pub, 1995.

- [GW07] GIEGL M., WIMMER M.: Unpopping: Solving the image-space blend problem for smooth discrete lod transitions. In *Computer Graphics Forum (2007)*, vol. 26, Amsterdam: North Holland, 1982-, pp. 46–49.
- [HM93] HITCHNER L., MCGREEVY M.: Methods for user-based reduction of model complexity for virtual planetary exploration (Proceedings Paper).
- [Hop96] HOPPE H.: Progressive meshes, ACM Computer Graphics Proceedings. In *Annual Conference Series (SIGGRAPH'96) (1996)*, pp. 99–108.
- [Hop97] HOPPE H.: View-dependent refinement of progressive meshes. In *Proceedings of SIGGRAPH (1997)*, vol. 97, pp. 189–198.
- [KBBdMS78] KOENDERINK J., BOUMAN M., BUENO DE MESQUITA A., SLAPPENDEL S.: Perimetry of contrast detection thresholds of moving spatial sine wave patterns. II. the far peripheral visual field (eccentricity 0-50). *Journal of the Optical Society of America* 68, 6 (1978), 850.
- [Kel84] KELLY D.: Retinal inhomogeneity. I. Spatiotemporal contrast sensitivity. *Journal of the Optical Society of America A* 1, 1 (1984), 107–113.
- [LE97] LUEBKE D., ERIKSON C.: View-dependent simplification of arbitrary polygonal environments. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (1997)*, ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, pp. 199–208.
- [LGCP] LEVOY M., GERTH J., CURLESS B., PULL K.: The Stanford 3D scanning repository. URL <http://graphics.stanford.edu/data/3Dscanrep/>.
- [Liv88] LIVINGSTONE M.: Art, illusion and the visual system. *Scientific American* 258, 1 (1988), 78.
- [LMYM01] LOSCHKY L., MCCONKIE G., YANG J., MILLER M.: Perceptual effects of a gaze-contingent multi-resolution display based on a model of visual sensitivity. In *the ARL Federated Laboratory 5th Annual Symposium-ADID Consortium Proceedings (2001)*, pp. 53–58.

- [LPMB09] LI R., POLAT U., MAKOUS W., BAVELIER D.: Enhancing the contrast sensitivity function through action video game training. *Nature Neuroscience* 12, 5 (2009), 549–551.
- [LT00] LINDSTROM P., TURK G.: Image-driven simplification. *ACM Transactions on Graphics (ToG)* 19, 3 (2000), 204–241.
- [Lue03] LUEBKE D.: *Level of detail for 3D graphics*. Morgan Kaufmann, 2003.
- [LW07] LOSCHKY L., WOLVERTON G.: How late can you update gaze-contingent multiresolutional displays without detection? *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 3, 4 (2007), 7.
- [Mar08] MARK W.: Future graphics architectures. *ACM* (2008).
- [MHK06] MOESLUND T., HILTON A., KRÜGER V.: A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding* 104, 2-3 (2006), 90–126.
- [Mic27] MICHELSON A.: A 1927 Studies in Optics. *Scientific American* (1927).
- [Mic09] MICROSOFT: Microsoft Project Natal, 2009.
- [MR98] MACK A., ROCK I.: Inattention blindness: Perception without attention. *Visual attention* 8 (1998), 55–76.
- [MS74] MANNOS J., SAKRISON D.: The effects of a visual fidelity criterion of the encoding of images. *IEEE Transactions on Information Theory* 20, 4 (1974), 525–536.
- [Mur78] MURPHY B.: Pattern thresholds for moving and stationary gratings during smooth eye movement. *Vision Research* 18, 5 (1978), 521–530.
- [Nak90] NAKAYAMA K.: Properties of early motion processing: Implications for the sensing of egomotion. *The Perception and Control of Self Motion* (1990), 69–80.
- [Nin09] NINTENDO: Wii MotionPlus controller. In *www.wiimotionplus.co.uk* (2009).

- [OHM*04] O’SULLIVAN C., HOWLETT S., MORVAN Y., McDONNELL R., O’CONOR K.: Perceptually adaptive graphics. *Eurographics State of the Art Reports 4* (2004).
- [Ols08] OLSON T.: Hardware 3D graphics acceleration for mobile devices. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008* (2008), pp. 5344–5347.
- [OYT96] OHSHIMA T., YAMAMOTO H., TAMURA H.: Gaze-directed adaptive rendering for interacting with virtual space. In *Proceedings of VRAIS* (1996), vol. 96, pp. 103–110.
- [Pel90] PELI E.: Contrast in complex images. *Journal of the Optical Society of America A* 7, 10 (1990), 2032–2040.
- [RB83] REGAN D., BEVERLEY K.: Visual fields described by contrast sensitivity, by acuity, and by relative sensitivity to different orientations. *Investigative Ophthalmology & Visual Science* 24, 6 (1983), 754.
- [RB93] ROSSIGNAC J., BORREL P.: Multi-resolution 3D approximations for rendering complex scenes. *Modeling in Computer Graphics 455* (1993), 466.
- [Red97] REDDY M.: Perceptually modulated level of detail for virtual environments. *University of Edinburgh* (1997).
- [Ris07] RISSER E.: True Impostors. *GPU Gems 3* (2007).
- [RVN78] ROVAMO J., VIRSU V., NÄSÄNEN R.: Cortical magnification factor predicts the photopic contrast sensitivity of peripheral vision. *Nature* (1978).
- [SC99] SIMONS D., CHABRIS C.: Gorillas in our midst: Sustained inattention blindness for dynamic events. *PERCEPTION-LONDON- 28* (1999), 1059–1074.
- [SDL*05] SUNDSTEDT V., DEBATTISTA K., LONGHURST P., CHALMERS A., TROSCIANKO T.: Visual attention for efficient high-fidelity graphics. In *Spring Conference on Computer Graphics (SCCG 2005)* (2005), Citeseer, pp. 162–168.
- [SHR*01] SCHWAB I., HO V., ROTH A., BLANKENSHIP T., FITZGERALD P.: Evolutionary attempts at 4 eyes in vertebrates. *Transactions of the American Ophthalmological Society* 99 (2001), 145.

- [SRL09] SRLINK: Publicaciones que usan sistemas EyeLink segun la empresa SR Systems, 2009.
- [SS96] SCHAUFLEER G., STÜRZLINGER W.: A three dimensional image cache for virtual reality. In *Computer Graphics Forum* (1996), vol. 15, Citeseer, pp. 227–235.
- [SZL*92] SCHROEDER W., ZARGE J., LORENSEN W., ET AL.: Decimation of triangle meshes. *COMPUTER GRAPHICS-NEW YORK-ASSOCIATION FOR COMPUTING MACHINERY- 26* (1992), 65–65.
- [TBI] TATARCHUK N., BRENNAN C., ISIDORO J.: Motion blur using geometry and shading distortion. *ShaderX2, Shader Programming Tips and Tricks with DirectX 9*, 299–308.
- [Tre86] TREISMAN A.: Preattentive processing in vision. In *Papers from the second workshop Vol. 13 on Human and Machine Vision II* (1986), Academic Press Professional, Inc., p. 334.
- [TSB96] THIBOS L., STILL D., BRADLEY A.: Characterization of spatial aliasing and contrast sensitivity in peripheral vision. *Vision Research* 36, 2 (1996), 249–258.
- [Tyl85] TYLER C.: Analysis of visual modulation sensitivity. II. Peripheral retina and the role of photoreceptor dimensions. *Journal of the Optical Society of America A* 2, 3 (1985), 393–398.
- [VRLN82] VIRSU V., ROVAMO J., LAURINEN P., NÄSÄNEN R.: Temporal contrast sensitivity and cortical magnification. *Vision Research* 22, 9 (1982), 1211.
- [Web64] WEBB P.: *Bioastronautics data book*. Scientific and Technical Information Division, National Aeronautics and Space Administration;[for sale by the Superintendent of documents, US Govt. Print. Off.], 1964.
- [WHS97] WALTER B., HUBBARD P., SHIRLEY P., GREENBERG D.: Global illumination using local linear density estimation. *ACM Transactions on Graphics* 16, 3 (1997), 217–259.
- [XV96] XIA J., VARSHNEY A.: Dynamic view-dependent simplification for polygonal models. In *Proceedings of the 7th conference on Visualization'96* (1996), IEEE Computer Society Press Los Alamitos, CA, USA.

- [ZPH03] ZHANG M., PAN Z., HENG P.: Time-critical rendering algorithm with incorporation of LoD, visibility culling and object impostor. *The Journal of Visualization and Computer Animation* 14, 4 (2003), 211–223.

Capítulo 7

Apéndice

A continuación se anexan dos publicaciones en las que he participado durante el desarrollo de este trabajo.

La primera ha sido aprobada para el congreso GAME'ON en 2008, realizado en la Universidad Politécnica de Valencia.

La segunda publicación fue enviada al Workshop “Animate & Illuminate” de Eurographics 2009 en Irlanda. La publicación no ha sido aceptada como full paper, pero si fué presentada como Poster y se publicó un resumen detallado en los proceedings del congreso.