

UNIVERSIDAD POLITÉCNICA DE VALENCIA
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN



Integración fotorrealista de objetos sintéticos
en escenas reales mediante visualización
basada en imagen

Tesis Doctoral
presentada por Francisco José Abad Cerdá
dirigida por los Doctores Roberto Vivó Hernando, y
Emilio Camahort Gurrea

Integración fotorrealista de objetos sintéticos
en escenas reales mediante visualización
basada en imagen

Francisco José Abad Cerdá

Trabajo realizado bajo la dirección de los Doctores
Roberto Vivó Hernando y Emilio Camahort Gurrea
y presentado en la Universidad Politécnica de Valencia
para optar al grado de Doctor en Informática

Valencia, 2004

Este trabajo ha sido desarrollado, principalmente, en el marco del proyecto TIC1999-0510-C02-01 subvencionado por la Comisión Interministerial de Ciencia y Tecnología y el Programa de incentivo a la investigación de la Universidad Politécnica de Valencia.

A mi familia.
A Sandra.

Resumen

Esta tesis trata sobre la integración de objetos sintéticos en imágenes de escenas reales. Se presentan dos posibles escenarios: imagen estática e imagen dinámica. Aunque ambos escenarios presentan coincidencias en objetivos, técnicas y tecnología a emplear, cada uno presenta una problemática distinta. Un objetivo troncal que ha guiado el desarrollo de la tesis ha sido la no utilización de hardware especializado, como cámaras robotizadas, lentes de gran angular, etc.

El objetivo principal de la primera parte de esta tesis es construir un sistema de visualización basada en imagen capaz de insertar objetos sintéticos en una escena real de una manera realista. Este objetivo conlleva el modelado de la escena real con el fin de capturar la luz incidente en un punto de la misma. Se propone generar el modelo a partir de una serie de fotografías del entorno. Este modelo debe ser lo bastante detallado como para poder insertar el objeto virtual en un punto dado de la escena, y que se vea afectado por la luz existente en dicho punto. De la misma forma, el objeto debe poder modificar la escena real, por ejemplo, lanzando sombras sobre ella.

En la segunda parte de la tesis el objetivo es insertar una serie de objetos sintéticos en un vídeo tomado de una escena real. El principal problema de esta aplicación es detectar la posición y parámetros de la cámara para cada fotograma, para visualizar los objetos sintéticos con la perspectiva adecuada. La técnica actual más flexible para poder calibrar cámaras en tiempo real, sin utilizar material especial, son los localizadores ópticos. En este trabajo presentamos un sistema de localización diseñado teniendo en cuenta el objetivo de alcanzar cierto realismo en tiempo real. Se ha diseñado un marcador poco invasivo y que se puede borrar fácilmente con un postproceso de los propios fotogramas donde se han de insertar los objetos sintéticos.

Resum

Aquesta tesi tracta sobre la integració d'objectes sintètics en imatges d'escenes reals. Es presenten dos possibles escenaris: imatge estàtica i imatge dinàmica. Encara que ambdós escenaris presenten coincidències en objectius, tècniques i tecnologia a emprar, cadascun presenta una problemàtica distinta. Un objectiu troncal que ha guiat el desenvolupament de la tesi ha estat la no utilització de maquinaria especialitzada, com càmeres robotitzades, lents de gran angular, etc.

L'objectiu principal de la primera part d'aquesta tesi és construir un sistema de visualització basada en imatge capaç d'afegir objectes sintètics en una escena real d'una manera realista. Aquest objectiu comporta el modelatge de l'escena real amb la finalitat de capturar la llum incident en un punt de la mateixa. Es proposa generar el model a partir d'una sèrie de fotografies de l'entorn. Aquest model deu ser suficientment detallat com per a poder afegir l'objecte virtual en un punt donat de l'escena, i que es veja afectat per la llum existent en aquest punt. De la mateixa forma, l'objecte ha de poder modificar l'escena real, per exemple, llançant ombres sobre ella.

En la segona part de la tesi l'objectiu és afegir una sèrie d'objectes sintètics en un vídeo pres d'una escena real. El principal problema d'aquesta aplicació és detectar la posició i paràmetres de la càmera per a cada fotograma, per a visualitzar els objectes sintètics amb la perspectiva adequada. La tècnica actual més flexible per a poder calibrar càmeres en temps real, sense utilitzar material especial, són els localitzadors òptics. En aquest treball presentem un sistema de localització dissenyat tenint en compte l'objectiu d'arribar a cert realisme en temps real. S'ha dissenyat un marcador poc invasiu i que es pot esborrar fàcilment amb un postprocés dels propis fotogrames on s'han d'afegir els objectes sintètics.

Summary

This thesis deals with the integration of synthetic objects into real scene images. We explore two areas: integrating into static images and into dynamic video. Although both areas share goals, techniques and technologies, each one has its specific problems. A central objective that has guided this work is the avoidance of the use of specialized hardware, like motion controlled cameras, fish-eye lenses, etc.

The first goal of this thesis is to build an image based visualization system to insert realistic synthetic objects into a real scene. This objective implies the need to capture the incident light at the points in the scene. We propose to build a model of this light from several images of the environment. We use that model to properly render the synthetic objects under the light conditions of the real scene. Furthermore, the synthetic object should be able to modify the real scene, e.g., by casting shadows into the scene.

The objective of the second part of the thesis is to insert several synthetic objects into a video stream of a real scene. The main problem of this application is the recovery of the camera position and settings in each frame, in order to render the synthetic object from the proper perspective. Currently, the most flexible technique to do camera calibration in real time without specialized hardware is the optical tracking. We present a tracking system that has been designed taking into consideration the goals of our work. Our marker is not quite invasive and it can be easily removed from the frames of the video on which the synthetic objects are going to be inserted.

Prólogo

El doctorado ha sido sin duda un camino extraño, sinuoso y a veces agradecido que, por suerte, ha encontrado la posada en la que recobrar fuerzas para seguir adelante. Después de varios años de trabajo, primero en Albacete y luego en Valencia, antes en Sistemas de información geográfica y ahora en Gráficos, por fin se congeló el título y tomó forma algo informe.

Después de acabar con algún que otro árbol, producir una cantidad indeterminada de residuos radioactivos, filtrar por medios naturales litros y litros de café para devolverlo al Mundo en forma de agua, descuidar más de una y más de dos amistades y molestar a los guardias jurados del Politécnico más de lo debido, estas páginas resumen los resultados obtenidos e intentan abrir caminos a lugares más lejanos.

Evidentemente, mi primer agradecimiento va a mi familia (Sandra incluida), que ha estado ahí para apoyarme en todo momento, aunque no supieran lo que hacía los fines de semana. También me gustaría agradecer a todos los compañeros con los que he compartido las penas (y las alegrías) del camino. Tanto los de Albacete (Elena, Jose Eduardo, Juan y Angelote), como los de Valencia (Javi, Inma, Maria José, Mon...).

Por supuesto, un recuerdo agradecido a mis directores de tesis, por sus consejos, su cercanía y sus ánimos, antes Jesús, y ahora Roberto y Emilio.

Gracias también a Pedro Jorquera y a Marcos Martí por echarme una mano con el OpenGL para la generación de los vídeos de la integración dinámica.

También mi recuerdo a todas las víctimas colaterales de esta tesis, léase: amigos, becarios, alumnos, vecinos, compañeros profesores, técnicos, etc.

Gracias a todos.

Índice general

Resúmenes	I
Prólogo	VII
Índice general	x
1. Introducción	1
2. Antecedentes y fundamentos	7
2.1. Estado del arte	7
2.1.1. Composición de objetos sintéticos en imagen real . . .	8
2.1.2. Composición de objetos sintéticos en vídeo	10
2.2. Aplicaciones	13
2.2.1. Efectos especiales en cine	13
2.2.2. Realidad aumentada	16
2.2.3. Aplicaciones en el campo de la televisión	22
2.3. Fundamentos	23
2.3.1. Funcionamiento básico de una cámara fotográfica . .	23
2.3.2. Modelo matemático de una cámara	29
2.3.3. Técnicas de calibración	31
2.3.4. Imágenes de alto rango dinámico	36
2.3.5. Mapas de entorno luminoso	40
2.3.6. RADIANCE	44
2.4. Resumen	45
3. Integración fotorrealista	47
3.1. Trabajos relacionados	47
3.2. Diseño del sistema	48
3.2.1. Sesión fotográfica	51
3.2.2. Calibración	66
3.2.3. Corregir imagen de fondo	69
3.2.4. Diseñar objetos sintéticos	70
3.2.5. Generación de geometría de escena local	70
3.2.6. Capturar iluminación	79
3.2.7. Generar máscara de los objetos	92

3.2.8. Visualización de la escena local sin objetos	94
3.2.9. Visualizar escena local con objetos	98
3.2.10. <i>Rendering</i> diferencial	100
3.3. Resumen	103
4. Integración dinámica	105
4.1. Trabajos relacionados	105
4.2. Diseño del marcador	108
4.3. Diseño del sistema de localización	109
4.3.1. Preproceso	109
4.3.2. Calibración	117
4.3.3. Postproceso	127
4.4. Experimentos	132
4.5. Sistema de calibración alternativo	133
4.5.1. Puntos circulares. Aplicaciones	134
4.6. Resumen	137
5. Conclusiones y trabajos futuros	139
5.1. Conclusiones	139
5.2. Trabajos futuros	141
A. Proyección del marcador	143
Bibliografía	149
Bibliografía <i>on-line</i>	155
Índice alfabético	157

Capítulo 1

Introducción

Uno de los objetivos más importantes perseguidos desde los inicios de la Informática Gráfica ha sido la generación de imágenes sintéticas fotorrealistas. Idealmente, se persigue conseguir imágenes de las que no se pueda decidir si provienen de una fotografía de una escena real o han sido generadas por computador. El problema principal de alcanzar dicho realismo es simular la complejidad de los objetos que forman parte del mundo real y, sobre todo sus relaciones [27]. Una forma de alcanzar el fotorrealismo que está captando gran atención por parte de la comunidad investigadora es la utilización de imágenes reales para generar otras imágenes, es decir, partir de fotografías tomadas de una escena real para generar nuevas vistas sintéticas. Genéricamente, se agrupan en la familia de algoritmos de visualización basada en imagen (*Image based rendering*), y hay varias técnicas en Informática Gráfica que siguen este principio como, por ejemplo, el uso de texturas, el morphing, el warping, los mapas de entorno luminoso [95] y los campos de luz. Aplicaciones derivadas de las técnicas anteriores intentan, por ejemplo, mostrar elementos arquitectónicos iluminados por luces de las que no se tienen fotografías [86], pasear por una escena real sin disponer de su geometría [17], construir un entorno virtual para la comunicación entre personas [41], etc.

Por otro lado, la popularización del hardware de captura de imágenes y la disponibilidad de una potencia de cálculo continuamente creciente para usuarios no profesionales, han favorecido la aparición de aplicaciones de visión por computador orientadas al entorno doméstico. Para llevar las aplicaciones de visión por computador desde el laboratorio al hogar, se deben desarrollar técnicas robustas, baratas y efectivas.

Esta tesis va a presentar dos técnicas para la integración de imagen sintética e imagen real. La primera técnica tiene como objetivo principal la integración fotorrealista de objetos sintéticos en escenas reales. Idealmente, se busca que el observador de la imagen no pueda distinguir entre los objetos de la imagen cuáles son reales y cuáles no. Para ello, se deben incluir en el

proceso de cálculo ciertas pistas visuales que permitan engañar al observador (sombras arrojadas, reflejos, etc.). En [75] se presentan los tres problemas principales que deben resolver este tipo de aplicaciones:

- **consistencia geométrica:** el objeto virtual debe ajustarse a la perspectiva de la escena real desde la que se capturó la imagen real.
- **consistencia de iluminación:** el objeto debe presentar una iluminación tal y como si estuviese realmente en la escena, y debe proyectar las sombras correspondientes sobre el entorno.
- **consistencia temporal:** el objeto sintético se debe mover coordinadamente con su entorno.

La primera técnica presentada en la tesis comparte la mayor parte de dichos problemas. Sin embargo, el campo de trabajo se ha restringido a escenas estáticas, por lo que la consistencia temporal no se tendrá en cuenta, y se hará hincapié en los otros dos aspectos. La mayor parte de la tecnología actual disponible comercialmente para crear estos efectos se utiliza en televisión [106, 104][70], donde lo más importante es que funcione en tiempo real. La calidad de la integración a nivel de luces, sombras e interreflexiones, por tanto, no ha recibido la atención necesaria para alcanzar resultados fotorrealistas.

La segunda parte de la tesis tiene como objetivo obtener un sistema de integración de imagen en tiempo real. En este caso también se busca alcanzar una integración de calidad, pero la imposición del tiempo real impide utilizar por el momento un sistema de visualización global que permita incluir efectos avanzados, al menos con hardware común. Ambas técnicas tienen en común su base en técnicas de visualización basada en imagen, pues en el proceso se utilizará principalmente información obtenida de la escena real mediante imágenes.

La localización (*tracking*) es la tarea más importante en un sistema de composición de imagen real y sintética en tiempo real. Es el subsistema que se encarga de recuperar los parámetros de la cámara real, para configurar la cámara sintética con la que se visualizarán los objetos virtuales. Si los parámetros calculados son precisos, los objetos sintéticos estarán alineados con el mundo real. En otro caso, el realismo de la integración de los objetos en la escena se verá comprometido.

Los sistemas de localización pueden estar basados en dispositivos mecánicos, electromagnéticos y ópticos. Los sistemas comerciales actuales emplean tecnologías híbridas, combinando sensores magnéticos o inerciales con alguna técnica de visión por computador. Los sistemas híbridos suelen producir resultados robustos y precisos, pero a costa de utilizar hardware especializado [31].

En esta tesis presentamos un sistema de localización óptico portable y autónomo que permite integrar fácilmente objetos generados por computador en entornos reales. Dicho sistema recupera la posición y orientación de la cámara y sus parámetros intrínsecos en tiempo real. Para ello, se introduce un marcador en la escena. Este marcador se ha diseñado teniendo en cuenta varias características deseables: calibración completa de la cámara, robustez y facilidad de construcción. Normalmente los marcadores o plantillas utilizadas para calibrar cámaras son difíciles de construir (un ejemplo de este tipo de plantilla son las de tres planos ortogonales entre sí), o son grandes, para obtener muchos puntos (un ejemplo típico de estas plantillas son las de tipo tablero de ajedrez). Nuestro marcador es simplemente un círculo hueco y una marca en el exterior. Además, el marcador también se puede borrar de las imágenes con facilidad.

Este trabajo tiene como objetivo aliviar algunos de los problemas con los que se encuentra un usuario de la tecnología actual para la integración de objetos sintéticos en entornos reales. El principal inconveniente que presentan los sistemas actuales es que dejan al usuario gran parte de la responsabilidad del éxito del efecto. Dichos métodos se basan en su experiencia y normalmente funcionan por la técnica de ensayo y error, lo que provoca que sean procesos lentos. Por ejemplo, es el usuario el que debe definir de alguna forma la posición de las fuentes de luz existentes en la escena real, tarea de crucial importancia en el resultado final. Para ello, normalmente lo que se hace es medir la posición de las fuentes de luz, así como su intensidad y color. Algunos técnicos de efectos especiales [44][107] han construido herramientas *ad hoc* para capturar con una mínima precisión la posición y características básicas de las fuentes de luz, y posteriormente modelarlas con el programa con el que se visualizarán los objetos sintéticos. Otro método que necesita gran pericia por parte del usuario es la simulación de una sombra arrojada por un objeto sintético sobre un fondo real. La solución aportada en [84] consiste en generar la sombra a partir del objeto sintético, proyectando la silueta de éste con un ángulo arbitrario, para luego restar un porcentaje del valor de los píxeles de la fotografía original donde cae dicha silueta. Un problema relacionado con el anterior, es la falta de ayuda en la construcción del entorno geométrico que modelará la escena en la que se van a visualizar los objetos sintéticos.

Otro objetivo de este trabajo es presentar un sistema que no necesite equipamiento especial o material difícil de obtener (objetivos especiales, como ojo de pez, escáners 3D...). Así, para obtener las imágenes, únicamente se utilizarán cámaras digitales ordinarias, bien de fotografía o bien de vídeo. En principio, no existe ningún requisito sobre dichas cámaras, aunque aquellas cámaras que tengan mejor objetivo y que den al usuario facilidades de control de su configuración producirán mejores resultados o facilitarán su trabajo.

Para alcanzar los objetivos de esta tesis se han utilizado diversas técnicas

de distintas disciplinas, que se presentan en el siguiente capítulo. Un ejemplo de estas actividades es la calibración de la cámara con la que se toman las fotografías del entorno donde se insertarán los objetos sintéticos. En aquellos casos en los que ha sido posible, se ha optado por la utilización de bibliotecas de libre disposición, documentadas y probadas para la implementación de técnicas conocidas.

Entre las aplicaciones de las técnicas desarrolladas en esta tesis están la generación de efectos visuales. El objetivo en dicha área es producir un vídeo donde se integren varios objetos sintéticos en la escena real de forma realista. Tradicionalmente, los usuarios profesionales han tenido que utilizar cámaras robotizadas, sistemas de múltiples cámaras sincronizadas o grabación contra fondo azul. En esta tesis mostraremos cómo alcanzar resultados realistas utilizando hardware no especializado. Aunque aún hoy se aplican técnicas tradicionales para la creación de personajes o decorados (animación tradicional tipo *Disney*, animación de personajes de plastilina, etc.), la corriente actual es la utilización de la computadora como herramienta principal en la creación de efectos especiales. Y aún en el caso en que se siga un proceso tradicional durante el rodaje de la película, se empieza a generalizar el uso de la composición digital en las últimas etapas de la producción. Lejos quedan los primeros tiempos de los efectos especiales, en los que criaturas míticas luchaban con los actores de carne y hueso a base de trucos ópticos y revelados en los laboratorios. Las técnicas de la época consistían en animar fotograma a fotograma a las criaturas (representadas por maquetas en miniatura o figuras a tamaño real), para luego mezclar dicha animación con los actores reales mediante un proceso óptico. Es interesante repasar la historia de los efectos especiales en el cine, ya que técnicas que se utilizan actualmente con asiduidad, como por ejemplo la composición digital con pantalla azul, nacen como evolución de técnicas desarrolladas para el cine en el primer tercio del siglo XX.

Una disciplina en auge en los últimos años en la que es crítica la necesidad de integrar objetos sintéticos en escenas reales es la realidad aumentada (*augmented reality*), utilizada principalmente para potenciar la percepción sensorial de un operador humano. En contraste con la realidad virtual, la realidad aumentada permite ver parte del mundo real al usuario. Las aplicaciones de realidad aumentada superponen a la imagen real datos, dibujos o imágenes que enriquecen la experiencia visual del usuario.

La inclusión de elementos sintéticos en escenas reales es un truco utilizado frecuentemente en la televisión. Por ejemplo, todos hemos sido testigos, quizá inconscientes, de la inclusión de publicidad en partidos de fútbol, en forma de carteles o pancartas que el público que se encuentra en el estadio no puede ver. Dicha publicidad ha sido incluida justo antes de emitir la señal de televisión por un sistema en tiempo real. Una ventaja indudable de este sistema es la posibilidad de incluir publicidad llamativa sin interferir en el desarrollo del juego.

Otros campos de aplicación de esta técnica son, por ejemplo, el diseño de interiores, en el que se puede decidir, con ayuda de imágenes sintéticas, dónde colocar nuevos elementos en un entorno. El sistema generaría imágenes que mostrarían los elementos sintéticos iluminados por la luz existente en la escena. La ayuda a la docencia, en la que se muestra un modelo esquemático superpuesto a objetos reales es otro campo que se puede aprovechar de este tipo de técnicas. Las interfaces de usuario avanzadas, creadas utilizando localizadores como el que presentamos más adelante en esta tesis, permiten al usuario interactuar con una aplicación utilizando un dispositivo más intuitivo que el ratón o el teclado.

El resto de la tesis se estructura de la siguiente forma: en el siguiente capítulo se hace un breve recorrido por las principales áreas de utilización de las técnicas que se presentarán posteriormente. Se presenta brevemente la historia de los efectos especiales en cine, la realidad aumentada y la composición en televisión. Además, en dicho capítulo, se introducen los fundamentos que se utilizarán en el resto de la tesis. En el Capítulo 3 se presenta una técnica para la mezcla fotorrealista de imagen sintética y real. Se hace una descripción detallada de cada uno de los pasos y se muestran varios resultados. El Capítulo 4 presenta un sistema de composición de imagen en tiempo real. Para ello, presenta un localizador diseñado teniendo en cuenta las especiales características de la aplicación a desarrollar. Se describe además un método de calibración de la cámara que, utilizando dicho marcador, permite recuperar los parámetros necesarios para calcular adecuadamente la perspectiva de los objetos sintéticos. El último capítulo presenta las conclusiones, las aportaciones, los resultados obtenidos durante el desarrollo de esta tesis y los trabajos futuros.

Capítulo 2

Antecedentes y fundamentos

En este capítulo se hará un estudio del estado del arte más relevante a cada una de las áreas de aplicación de esta tesis. Se comenzará presentando los trabajos entre cuyos objetivos está la composición de objetos sintéticos en imágenes estáticas. Típicamente estos trabajos han buscado alcanzar el máximo de realismo en sus resultados. Se verán trabajos que, aparte de integrar objetos sintéticos en imagen real, modelan la iluminación de la escena y permiten su modificación.

También se presentan las áreas de aplicación más importantes en las que actualmente se está utilizando la composición de imagen sintética y real. Se va a revisar brevemente la utilización de este efecto en los campos del cine, la realidad aumentada y la publicidad. De la misma forma, se presentan los conceptos y técnicas básicas en los que se apoyará el resto de la tesis. Entre dichos fundamentos se hará una breve descripción del modelo teórico de cámara que se ha utilizado, para luego describir el proceso de calibración automática de cámaras. También se presentará la fotografía de alto rango dinámico, los mapas de entorno luminoso y el paquete de simulación de iluminación RADIANCE.

2.1. Estado del arte

Pese a los avances obtenidos por la comunidad de gráficos por computador, la obtención de imágenes fotorrealistas es aún un problema abierto. La complejidad del mundo real hace muy difícil implementar una técnica que, por medio de una simulación matemática, obtenga imágenes realistas. Tanto el nivel de detalle geométrico necesario como el modelado de las características de la luz y su relación con los objetos, hace de esta una tarea ardua.

Una forma de reducir la complejidad para la obtención de imágenes realistas ha sido la introducción de imágenes tomadas de la realidad en el propio proceso de cálculo. Aquellas técnicas que buscan generar imágenes

del mundo real a partir de otras imágenes se agrupan bajo el nombre de *image-based rendering* [76, 14], mientras que aquellas técnicas que intentan capturar información del mundo real a partir de imágenes lo hacen bajo el nombre de modelado basado en imagen (*image-based modeling*) [88].

2.1.1. Composición de objetos sintéticos en imagen real

En este apartado se hará un recorrido por el estado del arte de las técnicas que implementan la composición fotorrealista de imagen real y sintética. Este tipo de aplicaciones típicamente trabajan con imagen estática, o bien con animación, pero situando la cámara en un punto fijo.

Uno de los trabajos pioneros en este área es el de Nakamae *et al.* [65]. En dicho trabajo se presenta una técnica para insertar objetos generados sintéticamente a una fotografía. El campo de aplicación de este trabajo es el estudio del impacto visual de una nueva construcción en un paisaje. El usuario debía aportar varios puntos en la imagen, junto a sus coordenadas 3D para calcular, mediante mínimos cuadrados, la posición de la cámara. Para el cálculo de las sombras generadas por el objeto sintético, se proponía calcular la posición del sol a partir de la situación de la cámara y de la hora en que se tomó la imagen. Para calcular la intensidad de la luz, se propone utilizar la relación entre la luminancia vista de una zona de la escena iluminada frente a otra zona no iluminada.

La iluminación es uno de los grandes problemas a resolver para obtener resultados fotorrealistas. Si se desea visualizar un objeto sintético integrado en una escena real, es necesario calcular la luz que recibiría un objeto real en la misma posición. Blinn y Newell presentaron en [12] uno de los primeros trabajos en los que se simulaba la iluminación de un objeto sintético mediante el uso de texturas. Dicho trabajo estaba orientado a la visualización de superficies curvas, y asignaba a cada punto en el objeto una región de la textura “de iluminación” por medio de la normal en dicho punto. Miller y Hoffman [60] hacen un repaso de las técnicas de la época (1984) para la utilización de mapas de reflejos para la iluminación de objetos sintéticos. Ya entonces apuntan algunos de los problemas y técnicas que aparecerán más tarde (por ejemplo, la necesidad del uso de imágenes de alto rango dinámico [22], o la utilización de esferas metálicas para obtener mapas de entorno). En [39] se puede encontrar una revisión de técnicas más recientes.

Otro trabajo que tiene como objetivo la composición de objetos sintéticos en una imagen real es [30]. Allí se indica que hay que resolver tres problemas: parámetros de visualización común, visibilidad común, e iluminación común, aunque se centra únicamente en éste último. El principal inconveniente de esta técnica es que el usuario tiene que modelar completamente la escena donde se insertarán los objetos sintéticos. Cada elemento de la escena real se tiene que modelar mediante un paralelepípedo. Después, se ha de tomar una fotografía ortográfica a cada una de las caras, y después indicar qué parte de

cada fotografía forma parte del objeto. Después que el usuario ha introducido toda esta información, el sistema aplica un cálculo de radiosidad a la escena modelada, suponiendo que los objetos son perfectamente difusos. Las luces también deben ser suministradas explícitamente al sistema.

En [24], Drettakis, Robert y Bougnoux presentan un sistema que intenta aliviar los problemas de la técnica anterior. Para ello, utilizan técnicas de visión por computador para modelar la escena con el nivel de detalle requerido. La técnica que presentan es semiautomática, pues, dadas varias imágenes de la escena, el usuario debe aportar puntos de control comunes a las imágenes para generar varios panoramas. El sistema utilizado para calcular los intercambios de luz entre las superficies de la escena vuelve a ser radiosidad. En este trabajo se limitan a escenas estáticas con la cámara estática, aunque una vez calculada la solución de radiosidad, la visualización de la escena final con los objetos sintéticos es rápida.

Un trabajo posterior, en la misma línea, es el de Loscos *et al.* [55, 54]. El objetivo que se plantean es poder controlar la iluminación de una escena real hasta el punto, por ejemplo, de “apagar” una luz real. Para ello, modelan la geometría de la escena tomando fotografías de la misma desde varios puntos de vista. A continuación, y para capturar la reflectancia de las superficies de la escena, fijan la cámara y toman varias fotografías de la escena, en las que se va variando la posición de una luz real. Para el proceso de cálculo de visualización, separan la iluminación directa de la indirecta, utilizando radiosidad para calcular esta última. Este trabajo también presenta una solución para “quitar” objetos reales de la escena por medio de una reconstrucción de textura.

El trabajo en el que nos hemos basado para el diseño de la primera parte de esta tesis es el presentado por Debevec en el SIGGRAPH de 1998 [20]. Utiliza mapas de entorno de alto rango dinámico para iluminar los objetos sintéticos y no necesita modelos geométricos muy detallados para obtener resultados fotorrealistas. Más adelante se presentará detalladamente esta técnica.

Sato *et al.* describen en [75] una técnica para capturar automáticamente el mapa de entorno de alto rango dinámico mediante dos imágenes tomadas con una lente gran angular, y aplicando técnicas de estereometría. De esta forma, reconstruyen aproximadamente la geometría de la escena mediante triangulación para, a continuación, mapear sobre ella el mapa de radiancias de alto rango dinámico que modela la luz. En este trabajo no se modelan las relaciones de los objetos sintéticos con la parte más cercana de la escena, por lo que éstos únicamente se ven afectados por la iluminación de la escena. Para visualizar los objetos sintéticos, se calcula la luz incidente en cada punto mediante un proceso de muestreo aleatorio sobre la semiesfera.

Gibson y Murta presentan en [34] un sistema para acelerar la integración de objetos sintéticos en imagen estática, y para ello dejan de utilizar sistemas de iluminación global. Para visualizar los objetos bajo la iluminación

real de la escena utilizan mapeado esférico (*sphere-mapping*), ya disponible en hardware gráfico estándar. Para reconstruir la iluminación real de la escena utilizan el mismo sistema de captura de Debevec [20], adaptado a las necesidades del hardware que se utilizará en la fase de visualización. Así, descomponen la iluminación de la escena real (almacenada en una imagen omnidireccional de alto rango dinámico) en un mapa de radiancia difuso, y una serie de mapas especulares, precalculados en función de distintos niveles de difusión (*roughness*) del material. También se calcula la posición de las fuentes direccionales automáticamente. La visualización consta de varios pasos: primero se generan las sombras arrojadas por los objetos sintéticos desde cada fuente, luego se generan los objetos sintéticos y luego se componen con la imagen de fondo en el *buffer* de acumulación de la tarjeta gráfica. En el sistema de Gibson y Murta no se modelan las interacciones entre los objetos sintéticos ni los reflejos de éstos sobre la parte de la escena real donde se insertarán.

Un trabajo aparecido recientemente [45] se centra en la inserción de objetos sintéticos en una imagen, para escenas de interior. Se centra principalmente en el problema de la consistencia geométrica, y para ello utilizan un sistema de autocalibración. Apoyándose en las características que suelen tener las escenas de interior (objetos rectangulares y rectas ortogonales entre sí), calibran la cámara mediante el cálculo los puntos de fuga de la imagen. No tienen en cuenta el problema de la iluminación común.

En general, para lograr el objetivo deseado de obtener una serie de objetos sintéticos perfectamente integrados en una escena real, algunos autores han aplicado un sistema de *inverse rendering* para recuperar las propiedades de reflectancia. Trabajos en este sentido son los de Yu *et al.* [86, 85], Gibson *et al.* [33], Ramamoorthi y Hanrahan [72], o Masselus *et al.* [58]. Con respecto a la iluminación de los objetos sintéticos, se ha avanzado en el sentido de utilizar la luz existente en la escena [21], capturada normalmente mediante imágenes de alto rango dinámico (explicada más adelante), y modelada mediante mapas de entorno [60, 39, 5].

2.1.2. Composición de objetos sintéticos en vídeo

En este apartado se estudiarán los trabajos que han aportado resultados a la integración de objetos sintéticos en vídeo de una escena real. Al revés del conjunto de trabajos anteriores, este área de aplicación normalmente busca obtener una eficiencia en la generación de las imágenes, a costa del fotorrealismo.

La principal fuente de literatura para la tarea de insertar objetos sintéticos en escenas reales capturadas en vídeo es la realidad aumentada. En los trabajos de recopilación de artículos de Azuma [8, 7] se puede obtener una buena aproximación de los desarrollos en este campo. Más adelante en este capítulo se hará una breve introducción a la realidad aumentada. En este

apartado nos centramos en presentar los trabajos más relacionados con la técnica que se desarrollará en el Capítulo 4.

El principal problema a resolver cuando se desea integrar objetos sintéticos en un flujo de vídeo es alinear correctamente los objetos sintéticos, o *localizar* la posición y calcular los parámetros de la cámara con la que se está capturando el vídeo. Existen múltiples formas de localizar la posición y orientación de cámaras. Las más precisas utilizan cámaras robotizadas controladas por ordenador. Otro tipo de localizadores (*trackers*) también utilizan dispositivos magnéticos, mecánicos, u ópticos. Los localizadores más flexibles son aquellos que utilizan el propio flujo de vídeo para extraer la información de la cámara. Normalmente los localizadores ópticos insertan un marcador en la escena especialmente diseñado para calcular los parámetros de la cámara automáticamente.

En el trabajo de Zhang *et al.* [87] se puede encontrar una comparativa de diferentes sistemas de marcadores visuales. Se comparan cuatro sistemas de localización óptica en términos de usabilidad, eficiencia, precisión y fiabilidad. La característica común de todos los marcadores estudiados es que son cuadrados, y utilizan la parte central del cuadrado para identificar el marcador. Aunque el caso de los marcadores cuadrados es el más común, en la literatura podemos encontrar ejemplos de sistemas de localización que utilizan marcadores circulares.

Uno de los trabajos pioneros sobre marcadores circulares, en el que se apoyan técnicas más modernas fue el de Forsyth *et al* [29]. Dicho trabajo se centraba principalmente en el desarrollo de un sistema de reconocimiento óptico de objetos mediante descriptores invariantes. Los invariantes algebraicos son relaciones calculadas sobre un objeto que permanecen constantes frente a transformaciones del objeto (poniendo un ejemplo simple en 2D, la longitud de un segmento como distancia euclídea es invariante frente a translaciones y rotaciones). Los autores presentan un sistema de invariantes aplicados a la detección de curvas planas y además ofrecen técnicas para recuperar, una vez detectada la curva, la posición en el espacio del objeto 3D. Más concretamente, dan el algoritmo para calcular la posición y orientación de la cámara, dados un círculo y su proyección correspondiente. En [74] siguen desarrollando el trabajo anterior, particularizando en pares de cónica.

Otro trabajo precursor en la utilización de cónicas como características básicas para algoritmos de visión fue el de Kanatani y Liu [43]. En dicho trabajo se relaciona el amplio conocimiento disponible de las cónicas derivado de su estudio por los matemáticos (geometría proyectiva) a su aplicación computacional. Como ejemplo, aportan un método para recuperar el plano de apoyo donde se encuentra un círculo, conocida la proyección de su centro. Posteriormente utilizaremos este resultado en el proceso de cálculo de localización del marcador desarrollado en esta tesis.

En el trabajo de Ahn y Rauh [6] se puede encontrar una colección de marcadores ópticos utilizados en sistemas de medida 3D. Justifican la utili-

zación de marcadores con características circulares debido a su robustez ante procesos de degradado (como desenfoque, o cambios en el ángulo y distancia de la cámara al marcador). La utilización de elementos circulares también permite el diseño de marcadores más compactos, de menor tamaño. El nuevo diseño propuesto en este trabajo, aparte de ser circular, contiene en su interior una serie de marcas circulares que permiten distinguir varios marcadores distintos en la misma escena. También ofrecen una serie de heurísticas para decidir si una cadena de píxeles extraída de la imagen es una elipse o no. En el capítulo correspondiente comentaremos dichos heurísticos. Para comprobar la robustez del marcador, presentan una serie de experimentos en los que se degrada la imagen añadiendo ruido aleatorio, y comprobando el buen comportamiento general de las características circulares.

Meng y Hu [59], por su parte, presentaron un nuevo método de calibración de cámaras mediante un marcador circular, cruzado por varias líneas rectas que pasan por el centro del círculo. Dadas tres imágenes del marcador desde distintas vistas, la calibración calcula linealmente los cinco parámetros intrínsecos de la cámara. Una de las ventajas del método mencionada por los autores es que se ahorran el difícil problema de la correspondencia (es decir, identificar los puntos comunes en una serie de imágenes). El método de calibración básicamente se basa en encontrar la proyección de los dos puntos circulares en cada una de las imágenes. Dados esos seis puntos (2 puntos, 3 imágenes), se puede calcular la cónica absoluta, y por lo tanto, los parámetros intrínsecos de la cámara. También presentan un método para recuperar parcialmente los parámetros extrínsecos. Un problema común de los marcadores circulares que también tiene este trabajo es que únicamente recuperan la normal al plano donde se encuentra el marcador (el centro del marcador es utilizado normalmente como origen del sistema de coordenadas del mundo), pero no puede recuperar la orientación X - Y del sistema de coordenadas global, porque el marcador es simétrico. En nuestro sistema hemos resuelto este problema.

Kim *et al.* [47, 46] presentan un marcador muy parecido al que proponemos en esta tesis. Dicho marcador está compuesto por una corona circular (una zona de color delimitada por dos círculos concéntricos). En ambos trabajos introducen sendas técnicas para recuperar el centro proyectado de los círculos. La primera técnica, presentada en [47], se basa en un nuevo invariante introducido en este trabajo. Dicho invariante establece que los centros de las elipses correspondientes a la proyección de una serie de círculos concéntricos están alineados con el centro proyectado de los círculos. Dado un marcador y el radio de los círculos que lo componen, es sencillo calcular la posición del centro proyectado de los círculos originales. En [46] presentan un método alternativo que no necesita conocer los radios de los círculos. Dicho trabajo presenta una función de coste que, al minimizarla, obtiene los parámetros intrínsecos y el plano de apoyo del marcador. Necesita al menos dos imágenes, y que el usuario proporcione el tamaño aproximado del sensor

CCD.

Un trabajo que se apoya en los resultados del anterior es el de Fremont y Chellali [32]. Utilizando el mismo diseño de marcador que los trabajos anteriores, este trabajo desarrolla un método para calcular directamente la distancia focal y la posición del plano de apoyo. Para reconstruir el sistema de coordenadas global, los autores proponen utilizar varios marcadores dispuestos en una estructura 3D. En el Capítulo 4 se explicará más detalladamente esta técnica y se presentará una modificación para obtener los parámetros extrínsecos por completo.

Se pueden mencionar otros trabajos que utilizan marcadores circulares, como el de Naimark y Foxlin [64, 31], que presentan una serie de marcadores circulares para calibrar un sistema de localización híbrido óptico-inercial. Dicha calibración utiliza muchos marcadores para calcular de forma precisa la alineación entre ambos sensores. López *et al.* [53] presentó un marcador circular para implementar un sistema de computación ubicua, en la que, por ejemplo, cada usuario lleva consigo un marcador único. Cada estación de trabajo de un laboratorio tiene asociada una *webcam* que identifica el marcador cuando se acerca (y por lo tanto, al usuario), y automáticamente abre una sesión, manteniendo siempre el mismo entorno aunque el usuario cambie de ordenador. Por último, en [28] se presenta un método de calibración de cámaras muy preciso (y muy lento) que utiliza características circulares asociadas a un objeto 3D.

2.2. Aplicaciones

2.2.1. Efectos especiales en cine

En este apartado se van a introducir brevemente algunas técnicas utilizadas para la creación de efectos especiales, muy relacionadas con el objetivo de esta tesis. En el estudio histórico del desarrollo de los efectos especiales [73, 67, 1][102, 98] aparecen los fundamentos de muchos de los trucos visuales que se utilizan hoy en día, tanto en el mundo del cine, como en el de la televisión y otros canales audiovisuales.

Desde los inicios del cine se han utilizado los efectos especiales para producir sensaciones en el espectador motivadas por escenas difíciles o imposibles de conseguir con actores o decorados reales. Entre las primeras técnicas para desarrollar efectos se encuentra el del *paro de imagen*, utilizado en la escena de la decapitación en “The Execution of Mary Queen of Scots”, 1895. El *matte shot* aparece en 1903, y consiste en componer en la misma cámara dos tomas distintas, cubriendo una parte del objetivo con una máscara opaca durante la primera toma, para evitar que la zona de la película correspondiente se exponga. Después, se rebobina la película, se cubre la zona ya expuesta, y se descubre la que estaba cubierta, volviendo a rodar sobre la misma película la segunda toma. Una técnica similar utilizada para integrar

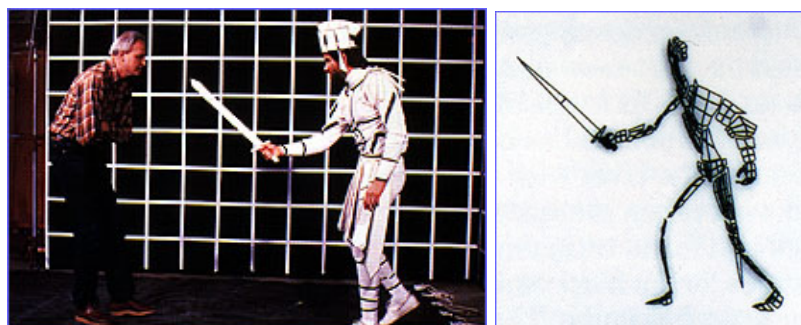


Figura 2.1: Toma de referencia y modelo alámbrico del caballero virtual en “Young Sherlock Holmes”

actores reales en un paisaje no real es el *glass shot*, que consiste en pintar el escenario de fondo sobre un cristal, colocándolo delante de la cámara para que parezca formar parte de la escena.

En 1916, Frank Williams mejora la técnica del *matte shot* al permitir movimiento en los personajes. Para conseguirlo, los fotografiaba contra un fondo monocromático para poder tratarlos luego como elementos individuales. La técnica se refinará posteriormente para convertirse en el procedimiento de la *pantalla azul* que se utiliza hoy en día.

Más recientemente, tras tres años de producción, se estrena “2001, Una Odisea Espacial”, 1968. En dicha película, se utiliza una forma temprana de control de movimiento (*motion control*), en la que un ordenador controla el movimiento y los parámetros de la cámara. De esta forma, se puede repetir el mismo movimiento de la cámara una y otra vez. Esto permite rodar diversos elementos de una escena compleja exactamente desde la misma perspectiva, obteniéndose vistas alineadas para la composición posterior. En el rodaje de la “La Guerra de las Galaxias”, 1977 también se utilizó el control de movimiento electrónico de las cámaras.

En la década de los 80, los gráficos por computadora empiezan a cobrar importancia en la producción de las películas. Por ejemplo, el Efecto Génesis de “Star Trek: La Ira de Khan”, 1982, que mostraba una tecnología capaz de dar vida instantáneamente a un planeta muerto, es la primera secuencia totalmente generada por computador que aparece en una película.

El primer personaje completamente generado por computador cobra vida desde una vidriera en la película “Young Sherlock Holmes”, 1985, con un coste aproximado de 6 meses de trabajo para una secuencia de 30 segundos (Figura 2.1).

El ciborg de metal líquido T-1000 debuta en “Terminator 2 - El día del Juicio”, 1991 con cinco minutos y medio de aparición en la película. Después de 60 años de impresión óptica, todos los efectos especiales de esta película



Figura 2.2: Fotografía y detalle del rodaje del efecto *bullet-time* en Matrix.

se componen digitalmente. A partir de entonces, han aparecido multitud de películas que, en mayor o menor grado, han utilizado personajes u otros elementos generados por computador, por ejemplo: “La máscara”, 1994, “Jumanji”, 1995, “Mars Attacks!”, 1996, “Men in Black”, 1997, “Godzilla”, 1998 (el monstruo es generado)...

Un ejemplo de película reciente que ha utilizado extensivamente técnicas relacionadas con las que se muestran en esta tesis, es “Dinosaurio”, 2000. En ella se utilizaron paisajes reales como fondo para mostrar una aventura protagonizada por dinosaurios completamente generados por computador [108]. Un ejemplo de utilización de una técnica similar, y con los mismos protagonistas, es la utilizada por la serie documental de la BBC “Caminando entre dinosaurios”, 1999 [91]. Películas recientes en las que los actores virtuales han desempeñado un papel importante han sido, por ejemplo, “Stuart Little”, 1999, “El señor de los anillos: Las dos torres”, 2002, “Hulk”, 2003, “Piratas del Caribe”, 2003...

El éxito de la saga Matrix (“The Matrix”, 1999, “The Matrix Reloaded”, 2003 y “The Matrix Revolutions”, 2003) se basó principalmente en el uso de unos efectos especiales nunca vistos en cine. En la primera parte, el efecto *bullet time* conseguía *congelar* el tiempo y cambiar el punto de vista de la cámara, mientras que el protagonista era tiroteado por el agente Smith (ver Figura 2.2). En la segunda parte, quizá el efecto más conseguido fue la pelea entre el protagonista y varias decenas de agentes Smith generados por computador. Para rodar esta escena se capturó tanto las expresiones de los protagonistas (mediante un proceso llamado *universal capture*, que utiliza cinco cámaras de alta resolución), como el aspecto de sus ropas (mediante un dispositivo de captura de la BRDF), como el escenario real (que fue escaneado por completo), como la luz presente en dicho escenario (con una técnica parecida a la que se verá posteriormente).

En este apartado se ha hecho un breve recorrido por las técnicas de creación de efectos especiales en el cine, ya que éste ha sido el principal motor para el desarrollo de métodos que permitan la convivencia entre personajes

reales y generados, la inclusión de personajes reales en entornos generados, o viceversa. Técnicas como el *croma key* (con la utilización de pantallas azules o verdes) han trascendido del mundo del cine y se utilizan habitualmente en otros entornos, como la televisión, como se verá en un apartado posterior. En el campo de los gráficos por computador, las máscaras se han utilizado desde la introducción de la tecnología raster para indicar qué píxel de una imagen contiene información y cuál no debe ser dibujado en pantalla. Una máscara de una imagen (dicha imagen puede ser una digitalización de una fotografía tomada con fondo azul), se basa en el mismo principio que el *croma*.

2.2.2. Realidad aumentada

La Realidad aumentada (RA) es una variante de los Entornos virtuales o Realidad virtual [8]. Mientras que en la Realidad virtual el usuario es inmerso en un entorno completamente generado por computador, sin posibilidad de ver el mundo físico que le rodea, la RA permite al usuario ver su entorno. La RA consiste básicamente en integrar objetos virtuales 3D en entornos reales 3D en tiempo real. Una disciplina relacionada es la telepresencia, en la que se busca presentar al usuario un entorno real. En este caso, normalmente el usuario y el entorno están alejados en el espacio.

Las tres características principales que definen la RA son: combinación de elementos reales y virtuales, interactividad en tiempo real y registro 3D.

La RA amplifica la percepción del usuario y facilita su interacción con el mundo real, por lo que es capaz de hacer perceptibles elementos antes ocultos a los sentidos del usuario. El objetivo de la RA es, normalmente, aportar información por medio de los objetos sintéticos para ayudar a realizar acciones en el mundo real.

A continuación se enumeran algunas de las áreas en las que se ha aplicado la RA:

- **Medicina.** Se utiliza principalmente para la realización de cirugía mínimamente invasiva. Para ello, se recogen datos del interior del paciente por medio de métodos no invasivos (resonancia magnética, tomografía axial, etc) para realizar un modelo tridimensional del mismo. Posteriormente, el médico puede visualizar sobre una imagen en vivo del paciente su estructura interior, y planificar la operación. La RA permite al médico “ver” dentro del paciente, sin la necesidad de realizar incisiones grandes (ver Figura 2.3). Otra aplicación dentro del quirófano sería la ayuda a la operación, presentando al cirujano los pasos a seguir sin necesidad de acudir a un manual, o la identificación de los órganos internos para evitar su daño durante la intervención.
- **Fabricación y reparación de maquinaria compleja.** En este campo, la RA permite conducir la actividad del operario directamente



Figura 2.3: Realidad aumentada en medicina: biopsia asistida

sobre una vista de la máquina a reparar mediante imágenes 3D en movimiento, en vez de fotografías y esquemas 2D en un manual. Se han construido aplicaciones de este tipo desde para reparar impresoras láser hasta para cablear aviones.

- **Anotación y visualización.** Estas aplicaciones permiten superponer información a una vista del mundo real, enriqueciéndolo de alguna forma. De la misma forma, permite potenciar la percepción del usuario en condiciones de visión difícil. Ejemplos son la delimitación de la pista de aterrizaje para aviones en condiciones meteorológicas adversas, la posibilidad de mostrar a un arquitecto dónde se encuentran las tuberías sobre la imagen del propio edificio, proporcionándole una especie de vista de rayos X, etc.
- **Cálculo de rutas en robots.** En este caso, la RA ayuda a estudiar la ruta que deberá seguir un robot, que normalmente se encuentra alejado del operador, existiendo incluso algún retraso en las comunicaciones. Para estos casos, se puede realizar una planificación sobre una vista del entorno real donde se encuentra el robot y posteriormente lanzar la orden de trabajo.
- **Entretenimiento.** Los decorados virtuales permiten incluir a actores reales en un entorno virtual, que de otra forma sería caro de construir. La aplicación de la RA a los videojuegos permite al jugador interactuar con el mundo real, otros jugadores y elementos virtuales a la vez, incrementando el realismo del juego. Con el desarrollo de la tecnología *wearable*, han aparecido pruebas de concepto de juegos en entornos abiertos reales [18, 71] (ver la Figura 2.4). Por otra parte, empiezan

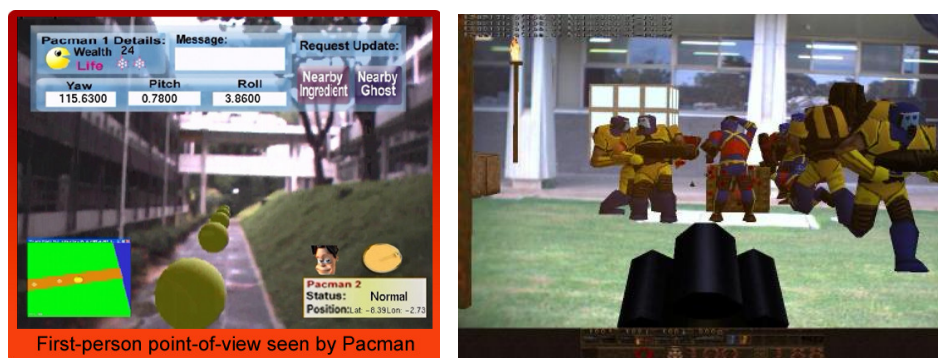


Figura 2.4: *Human Pacman* del *Mixed Reality Lab.* de la *National University of Singapore* y *ARQuake* del *Wearable Computer Lab* de la *Universidad de South Australia*



Figura 2.5: Dos fotogramas del juego *EyeToy: Play* para PlayStation 2

a aparecer en el mercado juegos cuyo interfaz es una cámara (*EyeToy* para PlayStation 2 de Sony), en el que el jugador se ve en la pantalla, interactuando con el juego con su propio cuerpo (Figura 2.5).

- **Aplicaciones militares.** En este caso son comunes las aplicaciones que permiten proyectar gráficos vectoriales sobre la vista de pilotos de aviones y helicópteros. Aparte de dar información básica de navegación y de vuelo, algunos sistemas permiten rastrear objetos reales en 3D, permitiendo seguir la trayectoria de objetos en movimiento en tiempo real.

Respecto a la tecnología que usa la RA puede ser principalmente de dos tipos: óptica y de vídeo. La tecnología óptica se basa en la utilización de visores montados en la cabeza que permiten la visión a través de ellos. Por el contrario, los visores cerrados no permiten ninguna visión del exterior, y por lo tanto necesitan capturar de alguna manera el entorno, normalmente por medio de cámaras de vídeo. Los visores que permiten ver el exterior

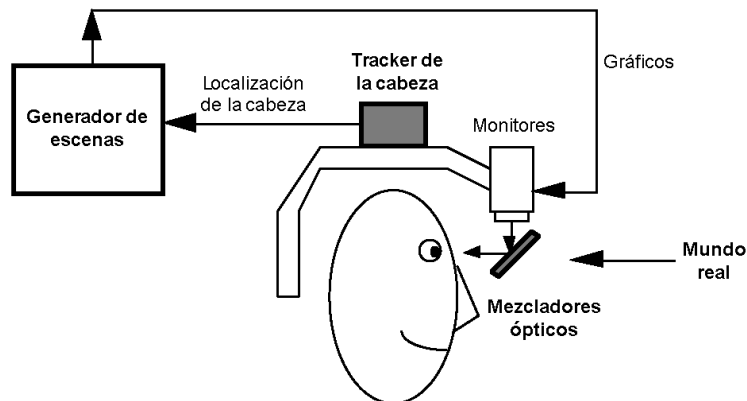


Figura 2.6: Esquema de la tecnología óptica en RA [8]

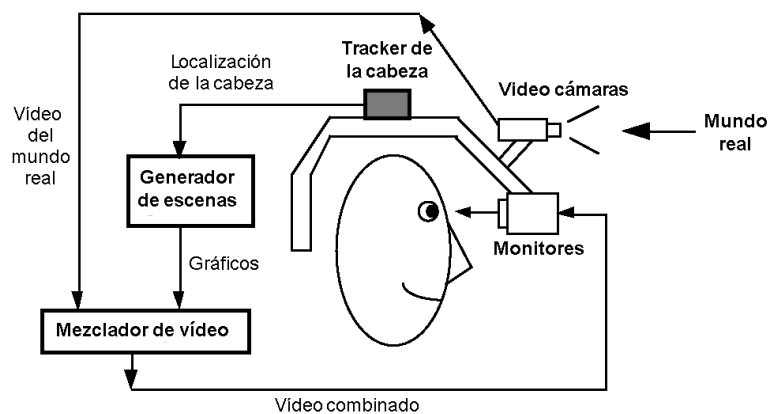


Figura 2.7: Esquema de la tecnología de vídeo en RA [8]

se basan en unos mezcladores ópticos que son parcialmente transparentes, por lo que el usuario puede ver a través de ellos, y reflexivos, de tal forma que reflejan la imagen que les llega del monitor (ver Figura 2.6). La problemática principal de estos sistemas es la elección de la cantidad de luz que deben dejar pasar del exterior. Normalmente los mezcladores ópticos reducen bastante la luz incidente, de tal forma que actúan como gafas de sol cuando no están funcionando. El problema de dejar pasar mucha luz es que los gráficos generados pueden resultar oscurecidos en un ambiente luminoso. El fenómeno inverso es igualmente problemático. Una opción para aliviar este problema es filtrar la luz entrante en función de su longitud de onda, de tal forma que si se va a utilizar un monitor monocromo, se deja pasar toda la luz del exterior, excepto la longitud de onda del color que emite el monitor.

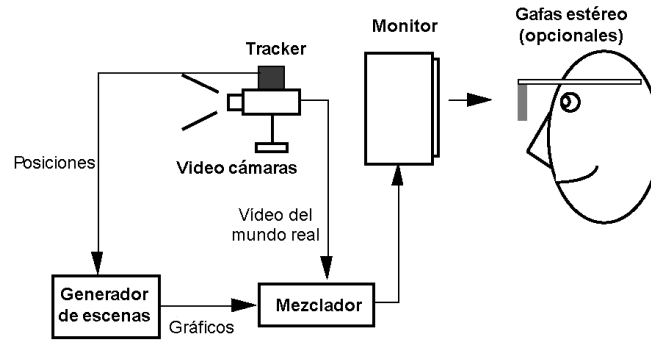


Figura 2.8: Realidad aumentada sobre un monitor [8]

En el caso de visores totalmente cerrados, el sistema contiene, además de los elementos anteriores, una o dos cámaras que se encargan de capturar la vista del mundo real que recibirá el usuario, como se puede ver en la Figura 2.7. De esta forma, el mezclador se encargará de integrar la imagen de vídeo recibida por las cámaras y los gráficos sintéticos. La forma más sencilla de realizar esta mezcla es mediante el *croma key* visto en el apartado anterior, por el que los gráficos sintéticos se generan con un fondo uniforme de un color conocido y prohibido en los elementos generados, que el mezclador de vídeo sustituye por la señal capturada por las cámaras. Si se tiene información de profundidad de la imagen que se está capturando por la cámara, se podrá realizar una composición más compleja para, por ejemplo, superponer elementos reales a los sintéticos, y viceversa.

Otra forma de implementar RA es mediante un sistema basado en un monitor. En este caso, el usuario no debe llevar ningún dispositivo en la cabeza, sino que el resultado se muestra directamente en un monitor (Figura 2.8). Una variante sería la representación en estéreo, mediante técnicas de mezclado de las dos imágenes correspondientes a cada ojo en el monitor, siendo necesario entonces que el usuario utilice gafas de visión estereoscópica.

Una variante del montaje con monitores es la utilización de mezcladores y monitores fijos, siendo el usuario el que se debe situar adecuadamente para su uso. Este es el ejemplo típico en aviones o helicópteros de combate, en el que el mezclador óptico está en la cabina, sobre los controles.

La tecnología óptica es más sencilla de implementar, ya que sólo se ha de preocupar de un flujo (el de los gráficos generados), mientras que si se utiliza vídeo existen dos flujos, cada uno de ellos con un retardo (imagen real capturada y gráficos generados). Además, en este último caso hay que tener en cuenta los problemas de resolución, seguridad en caso de que el sistema deje de funcionar, el desplazamiento que puede existir entre la vista de las cámaras y la vista “normal”, etc. Por el contrario, el sistema basado

en vídeo permite un mezclado más flexible (por ejemplo, se pueden eliminar elementos reales completamente, mientras que en el sistema óptico es difícil) resultando en una visualización más realista. Además, un sistema basado en vídeo puede corregir distorsiones producidas por las cámaras por medio de procesamiento automático, mientras que fabricar ópticas libres de distorsión es caro y complejo [48]. Otras ventajas de los sistemas basados en vídeo es que pueden utilizar métodos de registro 3D utilizando técnicas de Visión por computador, y es más sencillo hacer compatibles el enfoque y el contraste entre las imágenes sintéticas y las reales.

En contraste con los campos de Entornos Virtuales o Gráficos por computador, en la RA el fotorrealismo no es prioritario en la mayoría de las ocasiones, siendo a veces suficiente que el modelo sintético sea un modelo alámbrico. Como consecuencia de lo anterior, los requisitos del hardware de generación y visualización de gráficos son necesita también menores a los necesitados por la Realidad Virtual. Por el contrario, en la RA es más crítica la tarea del localizador (*tracker*), debido a que la correspondencia entre el mundo real y los objetos virtuales debe ser lo más precisa posible.

El problema del registro 3D consiste en asegurarse de que el mundo virtual se alinee lo máximo posible al mundo real. Este problema es vital en entornos médicos, donde el sistema indica al cirujano por dónde debe avanzar dentro del cuerpo del paciente. En el caso de la Realidad Virtual, los problemas de seguimiento del movimiento del usuario no son tan importantes porque no hay otra referencia visual aparte del mundo sintético. Así, en el usuario entrarán en conflicto los sentidos vista-kinestésico y vista-propioceptivo, es decir, la percepción de los músculos del cuerpo le dirá que ha realizado un movimiento, pero la vista no muestra el resultado esperado. Estos conflictos son menos evidentes que los conflictos vista-vista. Si el usuario levanta una mano ante sí y tiene un visor opaco, el sistema le mostrará una mano virtual, pero un desplazamiento de 5 milímetros respecto a la posición donde debería estar realmente la mano pasará desapercibido en la mayoría de las ocasiones. Si el visor le dejara ver su mano, entonces sí podría darse cuenta del error, debido a la alta resolución del sistema visual humano (la resolución espacial de la zona central del ojo es de medio minuto de arco). Además, el fenómeno de la captura visual ayuda a los sistemas de RA basados en vídeo. Este fenómeno consiste en que el cerebro da preferencia al sentido de la vista frente a otros (por ello funciona el ventrilocuismo), por lo que un operador con un casco opaco al que se le presente una mano virtual desplazada, asumirá que su mano está en dicha posición. En [8] se puede encontrar más información sobre los problemas aquí presentados, y la forma de combatirlos.

2.2.3. Aplicaciones en el campo de la televisión

El entorno más conocido de utilización del *croma key* dentro de la televisión son los programas de previsión meteorológica. Detrás del presentador suelen aparecer mapas o fotografías mostrando los fenómenos atmosféricos, dando la sensación de que realmente están allí, mientras que el presentador no los puede ver porque los ha insertado a posteriori un sistema mezclador.

También se utilizan en otros programas para la generación de decorados virtuales, en los que las personas del programa están en un estudio limpio monocromático, sin ningún elemento decorativo, mientras que el telespectador los está viendo inmersos en un entorno 3D generado por ordenador.

Aunque las aplicaciones anteriores están ya asentadas en el campo de la televisión, se está popularizando la utilización de efectos especiales avanzados, especialmente en los eventos deportivos. Un fin de dichos efectos es la inclusión de publicidad virtual [104]. En este caso, la señal tomada por las cámaras de televisión en el estadio pasan por un sistema informático en tiempo real que inserta un anuncio en una parte predeterminada del mismo. Idealmente, la integración con el entorno debe ser tal que el telespectador no se de cuenta de que la marca publicitaria que se ve en el centro del campo, o en un lateral, realmente no está ahí. De hecho, ése es el objetivo de todos los efectos especiales, tanto en cine como en otros medios: pasar desapercibidos.

La publicidad virtual tiene ventajas importantes frente a la publicidad tradicional, siendo una de ellas que no influye en el juego. A menudo han habido quejas por las vallas tradicionales (reales) en los campos de fútbol, sobre todo las de última generación, que contienen animaciones y que pueden entorpecer el desarrollo del juego. Otra ventaja del sistema de la publicidad virtual es la posibilidad de personalizar la publicidad: mientras que un grupo de telespectadores puede estar viendo un anuncio de una marca comercial, otro grupo puede estar viendo anunciada otra marca distinta. Una última ventaja es la posibilidad de insertar publicidad en lugares en los que sería muy difícil o imposible su colocación real, lo que incrementa la espectacularidad y el impacto en los espectadores de la misma (Figuras 2.9 y 2.10).



Figura 2.9: Ejemplos de publicidad virtual

Otra aplicación dentro de los eventos deportivos es la de mejorar la percepción de la acción, mediante la presentación de líneas o señales virtuales

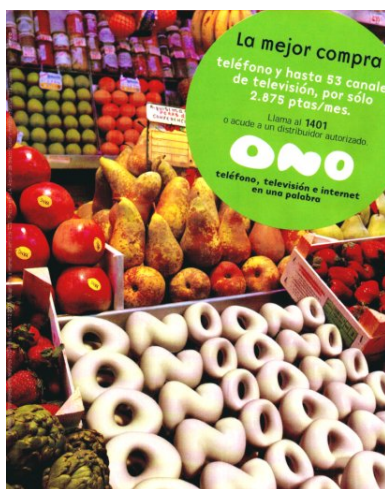


Figura 2.10: Ejemplo de mezcla de imagen sintética y real en publicidad

dibujadas sobre el campo. Este tipo de tecnología se ha aplicado, por ejemplo, en retransmisiones de partidos de fútbol americano, baseball, carreras de coches, golf, etc. (ver Figura 2.11). Así, se pueden presentar los resultados obtenidos u otra información integrados sobre el campo y pareciendo formar parte de él. Estos últimos sistemas sustituirían a los títulos sobreimpresos tradicionales [70][106, 92]. Mejorar la percepción de los telespectadores de un encuentro deportivo insertando información sobre las imágenes reales no es más que otro ejemplo de realidad aumentada.

Esta técnica y otras relacionadas que modifican las imágenes reales, aplicada en otros campos, como puede ser las emisiones de los servicios informativos tiene implicaciones éticas importantes, y algunas cadenas han prohibido explícitamente la alteración digital de las imágenes presentadas en sus informativos [100].

2.3. Fundamentos

A continuación, se explican las bases en las que se apoya la fusión de objetos sintéticos con escenas reales. Se hará una pequeña introducción al uso de cámaras fotográficas, la descripción del modelo teórico de la cámara, la calibración automática y la herramienta de iluminación global RADIANCE.

2.3.1. Funcionamiento básico de una cámara fotográfica

La mayoría de las bibliotecas gráficas y entornos de visualización tridimensional implementan un modelo de cámara, más o menos parametrizable, que permite definir el modo en el que la descripción matemática de una esce-



Figura 2.11: Ayudas virtuales en eventos deportivos

na se convertirá en una imagen 2D [27, 69, 49]. Normalmente esto se hace a partir de parámetros similares a los de una cámara real (posición, dirección de la vista, zoom, etc). Esta metáfora se conoce como cámara sintética, y sirve para facilitar al usuario la definición de la imagen que desea obtener.

Para visualizar los objetos sintéticos desde una perspectiva consistente con la escena real capturada por la cámara es necesario establecer adecuadamente los parámetros de la cámara sintética. Para obtener dichos parámetros, se pueden utilizar distintos sistemas de localización. Los sistemas de localización más flexibles, a la vez que asequibles, son aquellos en los que se extrae información de la escena real a partir de fotografías de la propia escena. Las técnicas de extracción de información a partir de imágenes obtenidas por medio de captadores se estudian en las disciplinas de Tratamiento digital de imagen [35, 15] y, a más alto nivel, Visión por computador [25, 79, 38][93].

Los componentes principales de cualquier cámara [48, 1] son los siguientes:

- **Objetivo:** sistema óptico formado por un conjunto de lentes dispuestas en el interior de una montura cilíndrica. Permite modificar el enfoque y la distancia focal (zoom).
- **Obturador:** es el mecanismo que abre y cierra el paso de luz hacia la cámara.
- **Diafragma:** es un conjunto de laminillas que se desplazan formando un orificio de superficie variable que permite regular con precisión la cantidad de luz que entra en la cámara.
- **Plano focal:** es el plano donde se forman las imágenes de la escena fotografiada. Se corresponde con la porción de película que se está exponiendo, en caso de cámaras convencionales, y con la matriz CCD en el caso de cámaras digitales.

El eje óptico es la normal a la lente que pasa por su centro, y el centro óptico es el punto central de la lente, en el eje óptico (Figura 2.12). Aquellos rayos de luz que pasan por él no sufren refracción, es decir, no se desvían.

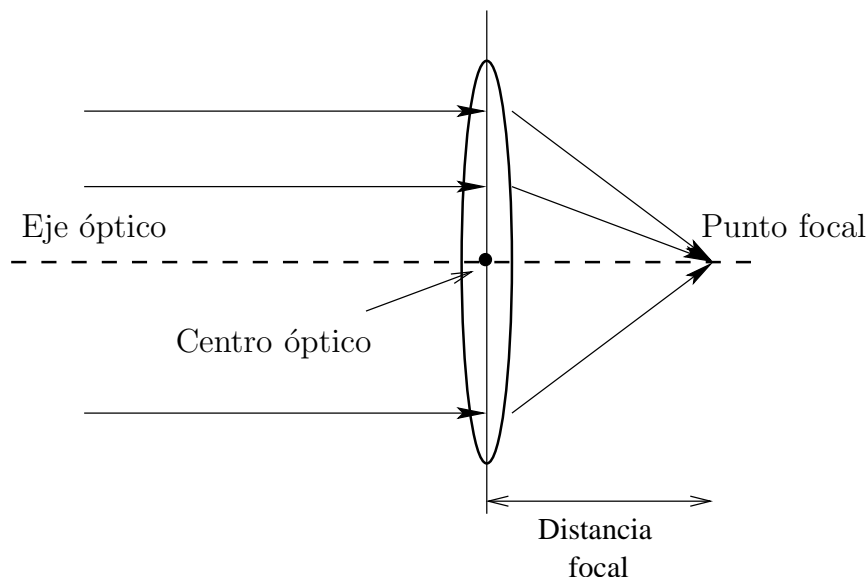


Figura 2.12: Algunos parámetros de una cámara

La distancia focal es la distancia entre el centro óptico y el plano focal, y es el parámetro de la cámara que define el tamaño de los objetos en la fotografía. Dicho tamaño es directamente proporcional a la distancia focal. Por ejemplo, la imagen de un objeto situado a la misma distancia, tomada con una cámara cuyo objetivo tiene 100 mm de longitud focal será el doble de grande que la imagen tomada con una longitud focal de 50 mm.

La imagen proyectada por el objetivo es circular y alcanza la calidad máxima en la zona central. La zona útil se denomina cobertura y ha de tener un diámetro superior al de la diagonal del formato de película o CCD para el que se diseña el objetivo. El ángulo de visión es el ángulo formado por las líneas imaginarias que unen los dos extremos de la diagonal del formato con el centro óptico. Por lo tanto, depende de la distancia focal y de la longitud de la diagonal del formato utilizado (ver Figura 2.13).

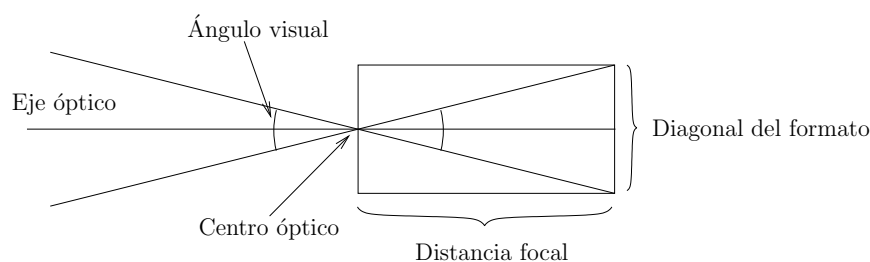


Figura 2.13: Ángulo de visión

Dependiendo de la relación entre la distancia focal y la diagonal del formato, se obtienen los siguientes resultados:

- **Si la distancia focal es igual a la diagonal del formato**, entonces el ángulo de visión es de 53° , que se corresponde con el ángulo visual del ojo humano, y por tanto considerado como normal. La perspectiva de los objetos aparece en la imagen tal y como se ve a simple vista.
- **Si la distancia focal es menor que la diagonal del formato**, entonces estamos ante una configuración de gran angular. Los objetos aparecen más pequeños y la perspectiva exagerada. Las líneas verticales y horizontales se van curvando en las zonas periféricas de la imagen.
- **Si la distancia focal es mayor que la diagonal del formato**, se obtiene el mismo resultado de ampliación de los objetos que se consigue con un telescopio. Únicamente el objeto enfocado se muestra nítidamente.

En las especificaciones del objetivo de una cámara convencional, suele aparecer grabada información como el nombre del fabricante, si el enfoque es manual o automático, la distancia focal en milímetros y el valor de su máxima luminosidad, es decir el máximo diafragma utilizable. En las cámaras digitales, las especificaciones pueden venir dadas en función de: número de aumentos, el equivalente a la distancia focal de una cámara de “paso universal” o el ángulo visual. Por ejemplo, en las especificaciones de la cámara

digital OLYMPUS Camedia 2000 Zoom, se puede leer: Objetivo de 6.5-19.5 mm, $f/2.0-2.8$ (equivalente a un objetivo de 35-105 mm en una cámara de 35 mm). Normalmente se utiliza la medida de la película convencional de 35 mm para expresar la equivalencia de un objetivo dado con una cámara “estándar”.

Para regular la cantidad de luz incidente se utilizan el diafragma y el obturador. El diafragma permite controlar la superficie de la apertura por la que entrará la luz. La cantidad de luz que llega depende tanto de la superficie abierta del diafragma como de la distancia focal del objetivo. La luminosidad del objetivo se define como la distancia focal dividida por la apertura máxima del diafragma:

$$\text{luminosidad} = \frac{\text{distancia focal}}{\text{apertura máxima}}$$

por ejemplo:

$$f3,5 = \frac{50mm}{14,28mm} \quad f7 = \frac{100mm}{14,28mm}$$

Normalmente la luminosidad aparece grabada en el objetivo en alguno de los siguientes formatos:

$$1:3.5 \quad f/3.5 \quad f3.5 \quad 1:3.5-4.5$$

La escala normalizada internacionalmente para expresar la apertura del diafragma es la de aperturas relativas, de “número f ” o $f-stops$. En aquellos objetivos en los que se puede regular la apertura, suele aparecer grabada una escala como la que sigue:

$$A \quad 22 \quad 16 \quad 11 \quad 8 \quad 5.6 \quad 3.5 \quad 2.8$$

donde:

- números menores se corresponden con mayor superficie (mayor apertura y, por lo tanto, mayor luminosidad),
- la cantidad de luz que pasa por un f es el doble de la que pasa por el siguiente f mayor. Aumentar un $f-stop$ implica reducir a la mitad la luz recibida.
- A: modo automático. Calcula la apertura a partir de una velocidad de obturación fija.

En las cámaras digitales es común que el ajuste del diafragma se realice automáticamente, en función de las condiciones ambientales en el momento de disparar la fotografía.

La apertura también afecta a otro parámetro de la fotografía que es la profundidad de campo (*depth of field*). La profundidad de campo indica qué parte de la escena saldrá enfocada en la fotografía. Si se reduce lo suficiente, únicamente aparecerá perfilado claramente el sujeto que se ha enfocado, mientras que el fondo aparecerá desenfocado. Cuanto mayor sea la profundidad de campo, mayor será el área de escena enfocada. Para controlar la profundidad de campo con la apertura de diafragma, hay que tener en cuenta que éste es mayor (mayor zona enfocada) cuanto menor es la apertura. Si se quiere obtener una imagen clara de toda la escena, tanto la cercana como la lejana, habrá que reducir el número de *f-stops*. El problema es que, al reducir la apertura, se reduce la cantidad de luz que llega a la película, por lo que para mantener el nivel de exposición adecuado, habrá que aumentar el tiempo en el que el obturador queda abierto.

El obturador es el mecanismo que controla el tiempo de exposición. Cuanto más tiempo está el obturador abierto (menos velocidad), más luz entra en la cámara, y por lo tanto, se obtiene más luminosidad en la fotografía. Normalmente la velocidad de obturación se puede ajustar en la cámara mediante un dial mecánico, o mediante una pantalla LCD. Usualmente se puede seleccionar la velocidad en una escala como la que sigue:

B 1 2 4 8 15 30 60 125 250 500 1000

La velocidad B indica que el obturador permanecerá abierto mientras que el usuario esté pulsando el botón de disparo. Los números representan fracciones de segundo, es decir, para calcular el tiempo que permanece abierto el obturador para un número dado, hay que dividir un segundo entre el número seleccionado en el dial. Por ejemplo, el 2 representa medio segundo. Usualmente, pasar de un número al siguiente mayor se describe como aumentar un *stop*, y si se pasa al número menor que se encuentra a dos números del actual, se ha reducido la exposición en dos *stops*. Aumentar un *stop* tiene como consecuencia reducir a la mitad la luz que recibirá la película, y reducir un *stop*, duplicar la luz recibida.

Las cámaras automáticas normalmente permiten al usuario seleccionar un parámetro para controlar la luz incidente, es decir, dar prioridad a la selección de la apertura o la velocidad de obturación, calculando el otro parámetro en función de la luminosidad de la escena.

Como se ha visto, el control del diafragma y la velocidad de obturación controlan la cantidad de luz que llega a la película. Por ello, diferentes combinaciones de ambos parámetros pueden resultar en la misma cantidad de luz recibida (exposiciones equivalentes). Por ejemplo, dada la cámara anterior, las combinaciones *f2.8* a 250, *f3.5* a 125 y *f5.6* a 60 resultarían en la misma exposición. Sin embargo, las tres combinaciones anteriores no resultan en la misma fotografía. Cada una de las configuraciones puede haberse seleccionado:

1. Para reducir los efectos de movimiento de la cámara. Una buena velocidad de obturación de uso general es de $1/125$ segundo. Una velocidad de obturación mayor a $1/250$ segundo puede producir incluso fotografías más nítidas. Con teleobjetivo puede ser necesario incluso mayores velocidades de obturación.
2. Para detener la acción. Una velocidad de obturación de $1/125$ segundo ayuda a detener la acción de alguien que va caminando. Sin embargo, puede haber ocasiones en que desee usar una mayor velocidad de obturación para detener una acción rápida, como una persona que va corriendo.
3. Para controlar la profundidad de campo. Como se explicó anteriormente, al usar una apertura pequeña o grande junto a la velocidad de obturación apropiada para mantener la exposición correcta se puede, respectivamente, aumentar o disminuir la profundidad de campo.

2.3.2. Modelo matemático de una cámara

En este apartado se introduce un modelo matemático de cámara fotográfica, lo que permitirá describir de forma precisa los fenómenos que se producen al captar una fotografía, según los parámetros vistos en el apartado anterior.

El comportamiento de las cámaras reales se suele aproximar por un modelo sintético sencillo, que respete las operaciones más importantes que realiza una cámara real. El modelo de cámara sintética más utilizado es el *pinhole* o cámara oscura estenopéica. La cámara *pinhole* consiste en un agujero infinitesimalmente pequeño a través del cual pasa la luz para formar una imagen invertida en la pared opuesta del agujero, dentro de la cámara. El plano donde se forma dicha imagen se llama plano focal.

Para trabajar con la imagen sin invertir, a menudo se sitúa el plano focal entre el centro de proyecciones (el objetivo o el agujero en el caso de la cámara *pinhole*) y la escena fotografiada. El proceso por el que se obtiene una imagen 2D a partir de una escena 3D se llama proyección perspectiva [27], y la Figura 2.14 muestra un esquema del mismo. La proyección de un punto 3D situado en la escena sobre la imagen 2D se consigue lanzando un rayo 3D desde el centro de proyecciones de la cámara hasta el punto en la escena, y calculando la intersección de dicho rayo con el plano focal.

En Visión por computador, el aparato matemático que se utiliza para describir numéricamente el proceso de captura de una fotografía es la geometría proyectiva. Aunque la geometría que estamos acostumbrados a utilizar es la Euclídea, la geometría proyectiva hace más elegante las expresiones de los procesos computacionales que hacen posible la visión (de hecho, la geometría Euclídea es un caso especial de la proyectiva) [25].

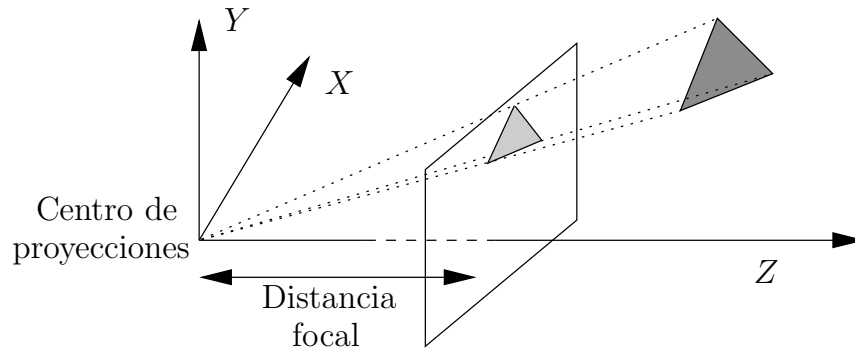


Figura 2.14: Esquema de la proyección perspectiva

Para definir de forma precisa desde dónde y cómo se tomó una fotografía, se deben definir una serie de parámetros que se dividen en dos grandes grupos: los que definen la posición y orientación de la cámara dentro del mundo, y los que definen la configuración de la cámara en el momento de tomar la fotografía (principalmente la distancia focal).

El modelo de la Figura 2.15 muestra los tres sistemas de coordenadas que se suelen utilizar a la hora de definir matemáticamente cada uno de los fenómenos envueltos en la toma de una fotografía. Así, se define un sistema de coordenadas asociado al plano focal que define la posición de los píxeles en la imagen final (sistema de coordenadas de la imagen), otro sistema de coordenadas, con origen en el centro de proyecciones de la cámara (sistema de coordenadas de la cámara) y un sistema de coordenadas global, en el que vendrán dadas las posiciones tanto de la cámara como de los objetos de la escena.

Los parámetros extrínsecos de la cámara definen su posición y orientación dentro de un sistema de coordenadas global (O_W en la Figura 2.15). Para una fotografía dada, dichos parámetros vienen definidos por una translación (el vector $\overrightarrow{O_C O_W}$) y una rotación, que definen la transformación necesaria para llevar el sistema de coordenadas del mundo al sistema de coordenadas de la cámara. La combinación de ambas transformaciones se encarga de llevar las coordenadas de los puntos definidos en el sistema de coordenadas global al sistema de coordenadas de la cámara.

Los parámetros intrínsecos definen las características propias de la cámara. Dichos parámetros son:

- **Punto principal** (O_R). Es la intersección del eje óptico con el plano focal (véase la Figura 2.15). Dicho punto a menudo no es el centro de la imagen, ya que el CCD no está perfectamente alineado con el eje óptico de la lente.

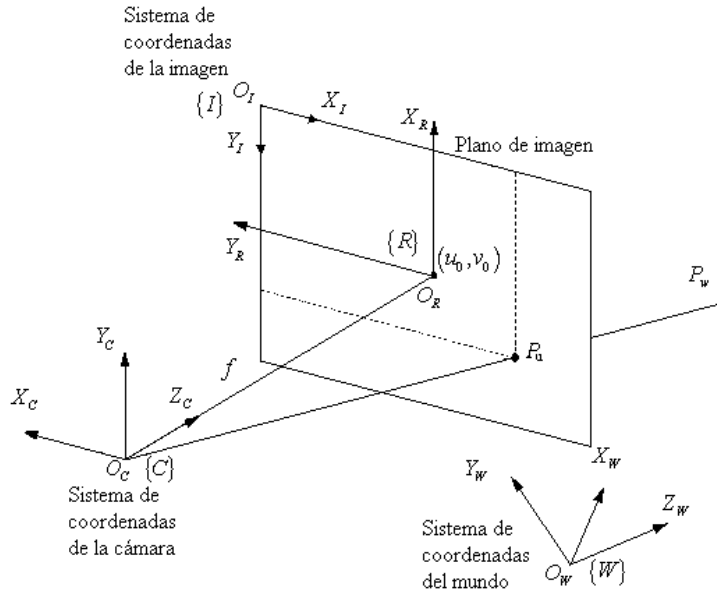


Figura 2.15: Modelo de los parámetros de una cámara

- **Distancia focal** (f). Es la longitud que separa el punto principal en la imagen del centro de proyecciones, y el que define el zoom de la cámara.
- **Distorsión**. Es una serie de parámetros que modelan las aberraciones de la lente, y que provocan deformaciones sobre la imagen. Dependen directamente de la calidad del objetivo.

En el apartado siguiente se hace uso de las definiciones vistas para describir una técnica para recuperar los parámetros intrínsecos y extrínsecos de una cámara a partir de fotografías de un objeto de geometría conocida.

2.3.3. Técnicas de calibración

Desde que las cámaras de vídeo y otros dispositivos de captura de imágenes han llegado al gran público (en forma de *webcams* o cámaras digitales asequibles), se hace necesario facilitar el uso de técnicas de Visión por computador a nivel de usuario, con aplicaciones lo suficientemente robustas para que no necesiten la intervención de éste. El presente apartado muestra una técnica que sigue esta filosofía.

La calibración de la cámara es un paso casi imprescindible en cualquier técnica de Visión por computador en la que se necesite extraer información métrica de las imágenes, y consiste en calcular los parámetros de la cámara

en el momento en el que se tomó la fotografía. Hay una gran cantidad de trabajos que tratan el tema (en [89] se puede consultar una lista de los principales). Los métodos tradicionales se pueden agrupar en dos grandes familias:

- **Calibración fotogramétrica.** En este caso la calibración se hace observando un objeto cuya geometría 3D se conoce con mucha precisión. Normalmente dichos objetos están formados por dos o tres planos ortogonales entre sí. Existen técnicas muy eficientes para calcular la calibración [25], pero por el contrario, se necesitan dispositivos de calibración caros y unos entornos y preparación costosos. Hoy día este tipo de calibración solo se realiza sobre cámaras utilizadas en fotogrametría, en las que es crítico conocer todos sus parámetros con una precisión extrema.
- **Autocalibración.** Las técnicas de esta categoría no utilizan ningún objeto de referencia. Para calibrar la cámara, simplemente hacen que ésta se mueva dentro de una escena estática. La rigidez de los objetos dentro la misma ofrece la información necesaria para calcular los parámetros internos de la cámara utilizando únicamente las imágenes. Teóricamente, si se toman las fotografías con una misma cámara sin variar los parámetros internos, las correspondencias entre tres imágenes bastan para calcular los parámetros internos y externos. Mientras que estos sistemas son muy flexibles, aún no son muy robustos, debido a que hay demasiados parámetros a estimar.

La técnica que se describe a continuación únicamente necesita una plantilla de calibración plana vista desde al menos dos posiciones distintas. La plantilla utilizada se puede encontrar junto a la documentación referente a las técnicas de calibración de la biblioteca *OpenCV* [99], de la que se hablará más tarde. Dicha plantilla se puede imprimir con una impresora láser y adjuntarla a una superficie razonablemente plana. Para tomar las fotografías, se pueden mover tanto la cámara como la plantilla de calibración, y no se necesita conocer el movimiento realizado. Esta técnica se puede encuadrar a medio camino entre la calibración fotogramétrica y la autocalibración, ya que necesita una plantilla de calibración, lo que le permite obtener resultados más fiables que con la autocalibración pura, pero la plantilla es 2D en vez de 3D.

La notación que se utilizará a continuación representa los puntos bidimensionales mediante $p = [u, v]^T$ y los tridimensionales con $P = [X, Y, Z]^T$. La notación \tilde{x} hace referencia al vector aumentado en una dimensión añadiendo un 1 como última componente, es decir $\tilde{p} = [u, v, 1]^T$ y $\tilde{P} = [X, Y, Z, 1]^T$. Según el modelo de cámara *pinhole* visto anteriormente, la relación entre un punto P en el espacio 3D y el punto correspondiente en la

imagen 2D generada por la cámara al proyectar dicho punto, p , viene dada por:

$$\lambda \tilde{p} = A \begin{bmatrix} R & T \end{bmatrix} \tilde{P},$$

donde:

- λ es un factor de escala arbitrario.
- A es la matriz intrínseca de la cámara, y tiene la forma:

$$A = \begin{bmatrix} f_x & \gamma & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

donde:

- f_x y f_y es la distancia focal medida en ancho y alto de píxel (si los píxeles son cuadrados, entonces $f_x = f_y$),
- γ es la medida de la oblicuidad entre los dos ejes (0, si son ortogonales) y
- (u_0, v_0) son las coordenadas del punto principal.
- R es la matriz de rotación y T el vector de translación que relacionan los sistemas de coordenadas del mundo y de la cámara (ver la Figura 2.15). Como se ha visto antes, este conjunto de valores forma la matriz extrínseca de la cámara:

$$M = \begin{bmatrix} R & T \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \end{bmatrix}$$

además:

- sea R_i la i -ésima columna de la matriz de rotación R . Entonces R_1 , R_2 y R_3 son X_w , Y_w y Z_w expresados en el sistema de coordenadas de la cámara, y
- T es la posición del origen del WCS con respecto a la cámara (ver la Figura 2.15).

Sin pérdida de generalidad, podemos asumir que el plano donde se encuentra la plantilla de calibración es el plano $Z = 0$ en el sistema de coordenadas global. Las plantillas de calibración sirven para definir puntos de control en el espacio 3D, en posiciones conocidas. Los puntos 3D que representan los puntos distinguidos de la plantilla 3D, según la suposición

anterior, tienen la forma $P = [X, Y, 0]^T$. Aplicando el modelo de la cámara *pinhole*:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} R_1 & R_2 & R_3 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} R_1 & R_2 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

Abusando de la notación, aún se utilizará P para denotar un punto en el plano de la plantilla de calibración, pero $P = [X, Y]^T$, dado que Z siempre es 0. De la misma forma, $\tilde{P} = [X, Y, 1]^T$. Por lo tanto, un punto en el modelo 3D y su punto proyectado en la imagen p están relacionados por una homografía H :

$$\lambda \tilde{p} = H \tilde{P} \quad (2.1)$$

donde

$$H = A \begin{bmatrix} R_1 & R_2 & T \end{bmatrix}.$$

H está definida hasta un factor de escala por lo que tiene 8 grados de libertad. Para obtener una estimación de la matriz H hacen falta, por lo menos, 4 pares de puntos (un punto 2D tiene 2 grados de libertad).

La Ecuación 2.1 se puede reescribir de la siguiente forma [89]:

$$\begin{bmatrix} \tilde{P}^T & \vec{0}^T & -u\tilde{P}^T \\ \vec{0}^T & \tilde{P}^T & -v\tilde{P}^T \end{bmatrix} \vec{h} = \vec{0}.$$

donde \vec{h} es un vector que contiene los elementos de la matriz H , fila a fila. Para cada par de correspondencias (X_i, Y_i, W_i) en el espacio, y $(u_i, v_i, 1)$ en pantalla se obtienen dos nuevas ecuaciones. Para n puntos, por tanto, se obtiene un sistema de $2n$ ecuaciones con 9 incógnitas. Como H está definida hasta un factor de escala, el sistema anterior se puede resolver mediante la descomposición en valores singulares. La solución será el vector singular derecho asociado con el menor valor singular [89].

Nótese que el sistema anterior está compuesto de coordenadas de píxeles en pantalla, coordenadas del espacio 3D y valores constantes a 1, por lo que el sistema puede estar numéricamente mal condicionado. Para obtener un resultado más estable, se debe aplicar una normalización que se puede resumir en los siguientes pasos [38]:

1. Normalizar P . Aplicar una translación y un escalado (T) a los puntos en el espacio, tal que el centroide de todos ellos sea el origen, y su distancia media al mismo sea $\sqrt{2}$.
2. Normalizar p . Calcular una transformación similar a la anterior (T'), aplicada a los puntos en la imagen.

3. Calcular H' sobre los puntos transformados con el algoritmo anterior.
4. Denormalizar. $H = T'^{-1}H'T$

Una vez obtenido H , se pueden calcular los parámetros intrínsecos y extrínsecos como sigue. Según se ha visto:

$$H = \begin{bmatrix} H_1 & H_2 & H_3 \end{bmatrix} = \lambda A \begin{bmatrix} R_1 & R_2 & T \end{bmatrix} .$$

Como R_1 y R_2 son ortonormales, se deben cumplir las siguientes expresiones:

$$H_1^T A^{-T} A^{-1} H_2 = 0 \quad (2.2)$$

$$H_1^T A^{-T} A^{-1} H_1 = H_2^T A^{-T} A^{-1} H_2 . \quad (2.3)$$

Sea

$$B = A^{-T} A^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \quad (2.4)$$

nótese que B es simétrica. Expresada como un vector:

$$\vec{b} = \begin{bmatrix} B_{11} & B_{12} & B_{22} & B_{13} & B_{23} & B_{33} \end{bmatrix} .$$

Se puede escribir:

$$H_i^T B H_j = v_{ij}^T \vec{b}$$

donde:

$$v_{ij} = \begin{bmatrix} H_{1i}H_{1j}, H_{1i}H_{2j} + H_{i2}H_{j1}, H_{2i}H_{2j}, \\ H_{3i}H_{1j} + H_{1i}H_{3j}, H_{3i}H_{2j} + H_{2i}H_{3j}, H_{3i}H_{3j} \end{bmatrix}^T .$$

Las restricciones 2.2 y 2.3 se pueden utilizar para establecer el siguiente sistema de ecuaciones:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} \vec{b} = \vec{0} .$$

Si hay más de una imagen disponible, se podrán añadir más ecuaciones como la anterior para calcular \vec{b} . Con una única imagen, sólo se pueden calcular dos valores de la matriz intrínseca (por ejemplo, f_x y f_y), asumiendo conocidos los demás. En [89] se puede encontrar cómo calcular A a partir de B . Para obtener los parámetros extrínsecos, se puede hacer:

$$\begin{aligned}
R_1 &= \lambda A^{-1} H_1 \\
R_2 &= \lambda A^{-1} H_2 \\
R_3 &= R_1 \times R_2 \\
T &= \lambda A^{-1} H_3
\end{aligned}$$

con $\lambda = 1/|A^{-1}H_1| = 1/|A^{-1}H_2|$. En [89] se puede encontrar, además del método aquí presentado, una extensión para calcular las distorsiones de la lente.

2.3.4. Imágenes de alto rango dinámico

Si se desea integrar objetos sintéticos en una escena real, es necesario capturar lo más fielmente posible dicha escena. Si, además, se quiere iluminar los objetos sintéticos con la luz real de la escena, es necesario buscar una forma de capturarla adecuadamente. Además, habrá que disponer de un sistema de cálculo de visualización que sea capaz de trabajar con magnitudes luminosas. En este apartado motivaremos la utilización de imágenes de alto rango dinámico como método apropiado de captura de la luz de una escena. Aunque existen equipos hardware especialmente diseñados para capturar fotografías de alto rango dinámico, en esta tesis nos hemos propuesto utilizar hardware común, por lo que presentamos un método de captura de este tipo de imágenes con cámaras fotográficas comunes.

El ojo humano tiene un comportamiento logarítmico frente a la luz, lo que le permite discriminar simultáneamente un rango grande de intensidades luminosas. La luz más brillante que puede distinguir es alrededor de nueve órdenes de magnitud superior a la más oscura, si se suman las intensidades del ojo adaptado a la oscuridad y a la luz. Si el ojo se encuentra adaptado, puede cubrir alrededor de cinco órdenes de magnitud [50]. Esto tiene problemas a la hora de representar las intensidades luminosas, tanto en un producto impreso como en un monitor.

El rango dinámico de una escena es el contraste existente entre las áreas más brillantes y las más oscuras. El efecto evidente del menor rango dinámico de la película fotográfica (o del CCD en el caso de fotografía digital) es la saturación. La saturación se produce cuando la cantidad de luz que llega a la película supera un cierto máximo. Por mucha luz que le siga llegando, el resultado final (intensidad de color en la fotografía) no variará.

En el caso del dispositivo informático de salida por excelencia, el monitor, el rango dinámico es menor que el del papel fotográfico, y únicamente puede mostrar 256 intensidades luminosas (niveles de gris). En [50] se muestra cómo maximizar la visibilidad de una imagen ajustando el rango dinámico de la misma al rango del monitor, adaptando el histograma de la imagen.

En dicho método se tienen en cuenta modelos perceptivos del sistema visual humano, tales como la sensibilidad al contraste, brillo, agudeza visual y sensibilidad al color.

La unidad física más importante para modelar el intercambio de energía es la radiancia (L), que es una unidad radiométrica (mide la radiación energética de los cuerpos). La radiancia describe la cantidad de energía luminosa que se envía desde un punto en una dirección específica por unidad de tiempo. Se mide en $\frac{W}{m^2 sr}$. La unidad fotométrica (que mide la percepción de la luz del ojo humano) equivalente a la radiancia es la luminancia, que se mide en candelas partido por metro cuadrado ($\frac{cd}{m^2}$) o *nits*. Por último, la irradiancia (E) mide el total de energía luminosa que llega a una superficie ($\frac{W}{m^2}$). Es importante en todo caso distinguir los conceptos de intensidad luminosa y brillo percibido, ya que debido a la naturaleza logarítmica de la visión humana, para duplicar el brillo observado desde una fuente, normalmente no basta con multiplicar por dos su intensidad luminosa.

En [66] se puede encontrar una revisión de técnicas de creación de imágenes de alto rango dinámico. Se presentan tanto técnicas utilizables únicamente en entornos estáticos como dispositivos especializados en capturar imágenes de alto rango en tiempo real. Además, en dicho trabajo se presenta una nueva propuesta, basada en calcular la imagen de alto rango dinámico a partir de un sensor donde los diferentes niveles de exposición varían espacialmente. Para ello, se inserta una máscara óptica delante del sensor, que provoca que cada elemento del mismo tenga una exposición diferente. Esta técnica conlleva una pérdida de resolución espacial.

Para ampliar el rango dinámico de radiancias captado por una fotografía, a continuación se muestra una técnica [22][94] que se basa en tomar una serie de fotografías de la misma escena, modificando la velocidad de obturación de la cámara.

El valor numérico de cada píxel de una imagen tomada por una cámara fotográfica de una escena real es el resultado de una función no lineal aplicada a la energía luminosa procedente de la escena e incidente en la posición correspondiente de la película o CCD de la cámara. Por ello, es difícil que el valor de los píxeles de una imagen sea una medida de la radiancia real en la escena. Por ejemplo, si un píxel en la imagen tiene un valor doble que otro píxel, es difícil que la radiancia en la zona correspondiente de la escena valga el doble que en la del píxel de menor valor. Por el contrario, las imágenes de alto rango dinámico tienen la propiedad de que el valor de sus píxeles es proporcional a la cantidad de luz existente en la escena real que se corresponde con dicho píxel.

Esa función no lineal, desconocida, está presente tanto en las cámaras digitales como en las convencionales de película. Dicha función es una composición de otras funciones no lineales mostradas en la Figura 2.16.

En la fotografía convencional, la película se expone a la luz para formar una imagen latente sobre ésta [48], y posteriormente en el proceso de reve-

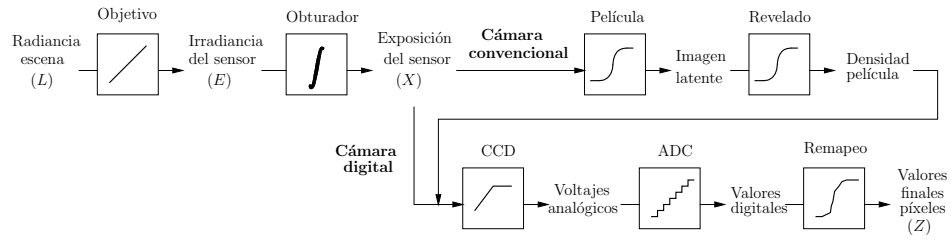


Figura 2.16: Proceso de conversión de energía en valor de píxel [22]

lado transformarla en variaciones de transparencia o densidad. El proceso de digitalización necesario para transformar el resultado de una fotografía convencional en valores numéricos se puede realizar directamente sobre la película o sobre el papel, una vez que el negativo se ha positivado sobre papel fotográfico. En este último caso, el proceso de impresión introducirá nuevas funciones desconocidas que dificultarán el cálculo de la radiancia.

En la primera etapa del proceso, la película responde a variaciones en la exposición $X = E\Delta t$, que es el producto de la irradiancia que recibe la película por el tiempo de exposición. Esta respuesta es una función no lineal descrita por la curva característica de la película. Dicha curva tiene en cuenta la pequeña respuesta que se produce cuando no hay exposición y la saturación producida a altas exposiciones (ver Figura 2.17).

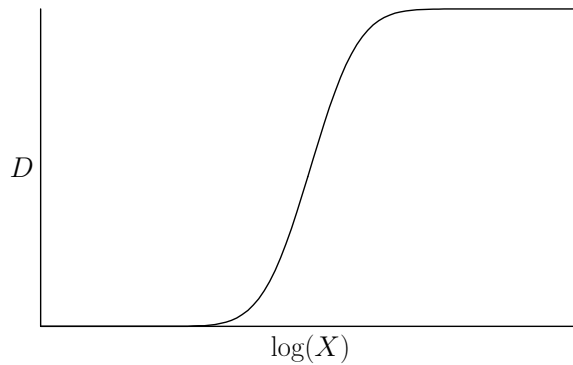


Figura 2.17: Curva característica de una película fotográfica. D es el flujo óptico

Las cámaras digitales que utilizan matrices CCD (*Charge Coupled Device*) tienen problemas similares. Aunque la energía recogida por los componentes de la matriz es proporcional a su irradiancia, la mayoría de las cámaras digitales aplican una función no lineal sobre la salida de la matriz antes de almacenar la imagen. Este proceso se realiza a menudo para repro-

ducir el comportamiento de una película fotográfica, anticiparse a respuestas no lineales en el dispositivo de visualización, o para reducir la profundidad de color del CCD (que normalmente es de 12 bits) para su almacenamiento (8 bits). Como en las cámaras convencionales, el agente no lineal más significativo es la saturación, que convierte todos los valores de irradiancia desde un punto al valor máximo de píxel.

Si se toma una serie de fotografías con distintos tiempos de exposición, se puede obtener un mayor rango de radiancias. En este apartado se describe brevemente cómo combinar dichas fotografías en una imagen de alto rango dinámico. A partir de los valores de los píxeles de las imágenes, se obtendrá un mapa de la radiancia de la escena, hasta un factor de escala. Para ello se utiliza una propiedad física de los sistemas de captura de imágenes: la reciprocidad. Dicha propiedad establece que la respuesta de una película (o una matriz CCD) es función de la energía que le llega. Dicha energía, definida por el término exposición, es el producto de la irradiancia que llega a la película multiplicada por el tiempo en que el diafragma está abierto. La reciprocidad dice que si la irradiancia se reduce a la mitad y el tiempo de exposición se duplica, la respuesta de la película será la misma. Esto se mantiene hasta llegar al punto de saturación, momento en el que si el tiempo es lo suficientemente grande, la curva pasa a ser una recta, como se puede ver en la Figura 2.17.

El proceso de conversión del valor de la exposición X en un valor de píxel Z se puede expresar mediante una función no lineal f , que es la composición de la curva característica de la película con el resto de las funciones no lineales introducidas posteriormente. El objetivo, por lo tanto, es encontrar la inversa a esa función con el fin de que, a partir de los valores de los píxeles, se puedan obtener las exposiciones: $X = f^{-1}(Z)$. Si, como parece lógico pensar, suponemos que f es monótona creciente, su inversa existirá y estará bien definida. Conociendo la exposición X y el tiempo de exposición, la irradiancia se puede recuperar mediante $E = \frac{X}{\Delta t}$.

Dado un conjunto de fotografías tomadas con tiempos de exposición conocidos Δt_j , denotamos los píxeles como Z_{ij} , donde i es el índice espacial sobre los píxeles de la imagen y j es el índice sobre los tiempos. La relación de reciprocidad se escribe como:

$$Z_{ij} = f(E_i \Delta t_j)$$

Invirtiendo la expresión:

$$f^{-1}(Z_{ij}) = E_i \Delta t_j$$

Tomando logaritmos:

$$\begin{aligned} \ln f^{-1}(Z_{ij}) &= \ln E_i + \ln \Delta t_j \\ \text{si } g &= \ln f^{-1} \Rightarrow g(Z_{ij}) = \ln E_i + \ln \Delta t_j \end{aligned}$$

donde Z_{ij} y t_j son conocidos, y se buscan los valores de E_i y la forma de la función g . Se puede aplicar optimización para calcular los mejores valores según mínimos cuadrados de las variables del problema. En [22] se puede ver el desarrollo. Una vez que se ha recuperado la curva, ésta se puede utilizar para convertir los valores de los píxeles en valores de radiancia relativa. La curva estimada se puede utilizar para calcular la radiancia de cualquier imagen captada con el mismo sistema de captura, y no sólo con las imágenes con las que se calculó dicha curva.

A continuación se muestra un ejemplo de esta técnica. En la Figura 2.18 se pueden ver 11 fotografías tomadas con distintas velocidades de obturación [22]. Se tomaron con una apertura de $f/8$ y con tiempos de exposición desde $1/30$ a 30 segundos, con diferencias de un *stop* (ver el Apartado 2.3.1). Posteriormente, se seleccionaron 45 posiciones de píxel y se aplicó el algoritmo anterior. En la Figura 2.19 se puede ver una imagen generada a partir del mapa de radiancias, optimizada para su visualización.

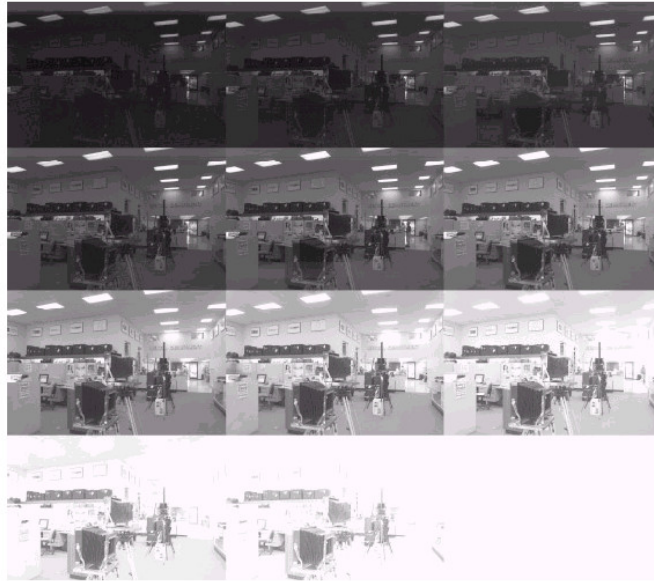


Figura 2.18: Fotografías tomadas con distintas velocidades de obturación [22]

En [94] se puede encontrar una aplicación que calcula el mapa de radiancias a partir de las imágenes de entrada suministradas por el usuario. En el capítulo siguiente se mostrará cómo utilizar dicha herramienta.

2.3.5. Mapas de entorno luminoso

La técnica del *reflection mapping* se utiliza desde principios de los 80. Se ideó como una forma de aumentar el realismo de los gráficos generados por



Figura 2.19: Imagen recuperada y optimizada para su visualización [22]

computador y consiste en aplicar a los elementos generados una textura de un entorno real capturado mediante una fotografía. Principalmente se utiliza para simular objetos brillantes, que reflejan la luz ambiente (ver Figura 2.20).

En 1984, Miller [60] ya advertía de los problemas de capturar el rango dinámico de una escena real (donde la diferencia entre las zonas emisoras y las zonas no emisoras puede ser de seis órdenes de magnitud) con una sola fotografía. En su trabajo ya propone dos técnicas, que implementará Debevec casi 15 años después. La primera es utilizar varias fotografías tomadas a distintas velocidades de obturación para ampliar el rango dinámico. La segunda propuesta mencionaba la utilización de una bola reflexiva, como se muestra en la Figura 2.20 para capturar fácilmente la luz de una escena. Otras referencias sobre este tema se pueden encontrar en [94] y [37]. Ejemplos más recientes del uso de esta técnica son las películas “Terminator II”, 1991 y “Star Wars: Episodio I. La amenaza fantasma”, 1999 [107] (ver Figura 2.21).

En [20] se desarrolla la técnica de capturar el mapa de entorno de una escena real fotografiando una bola reflectante, y aplicando la técnica vista en el Apartado 2.3.4. Para ello, se coloca la bola de espejo en la posición donde se desea insertar los objetos sintéticos, se fija el trípode, y se toma el mapa de entorno de alto rango dinámico siguiendo los pasos vistos en el apartado anterior. En el Capítulo 3 se detallará este proceso.

La técnica de capturar el mapa de entorno utilizando una sonda esférica tiene algunos problemas:

- hay mucha distorsión cerca de los bordes (la frecuencia de muestreo del entorno baja en las zonas exteriores de la esfera),
- hay una zona ciega, justo detrás de la bola, de la que no se tiene

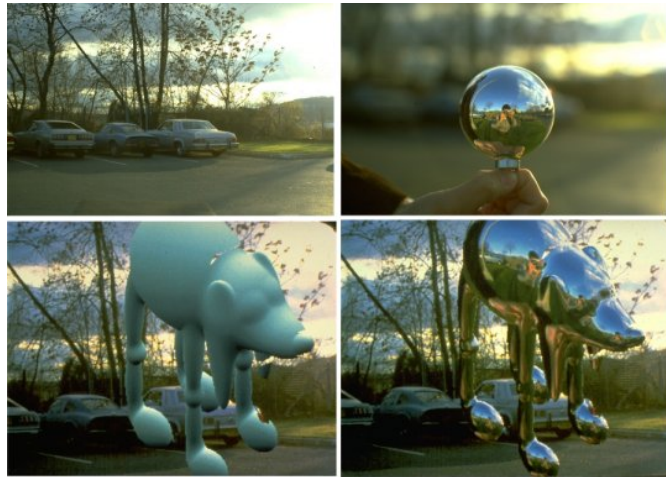


Figura 2.20: Ejemplo de *reflection mapping*



Figura 2.21: Fotograma de Terminator II y rodaje de La amenaza fantasma

información,

- en el mapa de entorno resultante aparece la cámara (y quizá el fotógrafo), y
- la sonda puede introducir artificios en la imagen, que luego aparecerá en el mapa de entorno, como por ejemplo sombras (ver Figura 2.22).

Para minimizar algunos de estos artificios, se pueden utilizar técnicas de retoque digital, o bien utilizar fotografías de la bola desde distintas posiciones, extrayendo de cada posición la zona de imagen de mayor calidad. Desde dos posiciones separadas se puede construir el mapa sin que aparezca la cámara y con una frecuencia de muestreo mejorada [94]. En el apartado posterior en el que se describe la forma de capturar el mapa de entorno se



Figura 2.22: La sombra de la sonda aparece en el mapa de entorno

muestra cómo realizar esta operación. Para evitar el problema de las sombras arrojadas por la sonda, hay que planificar cuidadosamente la sesión fotográfica y controlar la iluminación.

Una vez que se ha obtenido el mapa de radiancias, se ha de mapear sobre la geometría de la escena lejana. Para ello, hay que determinar con cierta precisión el mapeo entre las coordenadas de la esfera y la escena, siendo necesario calcular la posición de la bola con respecto a la cámara, el tamaño de la bola y los parámetros de la cámara (su posición en la escena y su distancia focal). A partir de esta información, es fácil trazar rayos desde el centro de la cámara hacia los píxeles, y reflejarlos sobre la esfera hasta la geometría de la escena lejana. Dicha geometría puede ser muy simple, por ejemplo una caja, como muestra la Figura 2.23. Una buena aproximación puede ser asumir que la bola es pequeña con respecto al entorno y que la vista de la cámara es ortográfica.

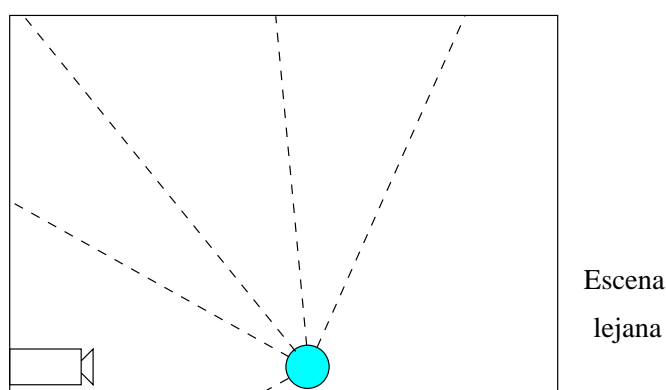


Figura 2.23: Captura de la escena lejana

Sato [75] presenta otra posibilidad para capturar la escena lejana, me-

diante la utilización de objetivos de ojo de pez y técnicas de reconstrucción de geometría a partir de imágenes tomadas en la misma escena desde puntos de vista ligeramente distintos (estereometría). En su trabajo, mientras que la generación del mapa de radiancias de alto rango dinámico utiliza la técnica desarrollada por Debevec vista anteriormente, se propone la captura automática de la geometría de la escena lejana a partir de las fotografías, en vez de que el usuario haga una estimación de la misma.

Debevec ofrece en su página web [94] una aplicación llamada HDR Shop que permite automatizar el proceso de creación de mapas de alto rango dinámico, así como la construcción de mapas de entorno luminoso. Más adelante se muestra cómo utilizar dicha herramienta.

2.3.6. RADIANCE

Como se comentó en la introducción, uno de los objetivos principales de este trabajo es la obtención de imágenes realistas. Para ello, es necesario que los objetos sintéticos se visualicen de la forma más realista posible. Por ello, para este trabajo se ha seleccionado una herramienta de cálculo de visualización global basada en propiedades físicas. Dicha herramienta se llama RADIANCE [82, 49][105], y está formada por un conjunto de aplicaciones *Open Source*. RADIANCE trabaja con valores de radiancia espectral, irradiancia e índices de brillo, en forma de imágenes, valores numéricos y curvas de nivel.

RADIANCE es un sistema de cálculo de soluciones de iluminación, y no un sistema de generación de gráficos fotorrealistas. La diferencia entre un tipo de sistemas y otro es importante: la mayoría de los sistemas de visualización actuales (3Dstudio, POV-Ray, etc) tienen como objetivo la generación de imágenes fotorrealistas, aunque para ello utilicen métodos de cálculo que no se basen en las propiedades físicas de los materiales. Dichos métodos, por ejemplo, utilizan fuentes de luz irreales que tienen una atenuación lineal con la distancia, en vez de la cuadrática que se produce en una fuente real, o la utilización de la luz ambiental (una cantidad de luz definida por el usuario, que es constante en toda la escena y no parte de ninguna fuente) como principal fuente de iluminación, etc. Por el contrario, RADIANCE intenta reproducir el comportamiento de la luz, y para ello utiliza una simulación numérica del intercambio de energía que se produce entre las superficies de la escena, y la distribución de luz generada por las fuentes.

RADIANCE no genera imágenes, sino mapas de la radiancia recibida desde un punto de vista dado. A través de los años, ha sido validado con escenas reales, comprobando que los resultados obtenidos son satisfactorios. El poder trabajar con valores de radiancia permite, utilizando los resultados obtenidos con la técnica vista en el Apartado 2.3.5, iluminar los objetos sintéticos con la luz que realmente se encuentra en la escena. El espíritu que inspiró RADIANCE es que, simulando la física de la interacción de la

luz entre las superficies tal y como se produce en la realidad, se obtendrán imágenes que se ajusten a la realidad.

2.4. Resumen

La integración de objetos sintéticos en entornos reales es una técnica de composición muy utilizada en una gran variedad de campos. En este capítulo se ha hecho un breve recorrido por algunas de las áreas de aplicación, por ejemplo, en el rodaje y postproducción de películas. Los efectos especiales en el cine han sido la punta de lanza en el desarrollo de esta clase de técnicas, que buscan “engañar” al espectador. Campos tan actuales como la realidad aumentada y la publicidad también son usuarios de este tipo de técnicas, lo que demuestra su interés.

Además, este capítulo ha presentado las bases para la implementación posterior de sistemas de integración de objetos sintéticos en entornos reales. Se ha mostrado un modelo teórico de cámara, un método automático de calibración, y cómo capturar y modelar la iluminación de una escena real para luego utilizar dicha luz para iluminar objetos sintéticos. Este último proceso se realizará con una herramienta de iluminación global.

Capítulo 3

Integración fotorrealista

En este capítulo se presenta una técnica para integrar uno o más objetos sintéticos en una imagen de una escena real, capturada mediante una fotografía. Esta técnica permite conseguir resultados fotorrealistas sin la necesidad de utilizar medios técnicos especializados. A partir de una serie de fotografías de una escena real, y utilizando un sistema de iluminación global se puede generar una nueva imagen de la escena, en la que los objetos sintéticos se confundan con los reales. Para lograr este objetivo, debe parecer que los objetos sintéticos interactúan con la escena real, es decir, debe parecer que están iluminados por la luz real de la escena, y a su vez que la escena recoge la sombra y reflejos de los mismos.

3.1. Trabajos relacionados

En la literatura se pueden encontrar varios trabajos en los que se trata la integración de objetos sintéticos en entornos reales, principalmente en el campo de la realidad aumentada [1]. Fournier [30] enumera los tres problemas que se deben resolver para obtener un resultado convincente:

- **parámetros de visualización comunes.** Los objetos sintéticos se deben mostrar bajo la misma perspectiva desde la que se tomó la fotografía. Con las técnicas de visión por computador disponibles, se puede extraer dicha información directamente de las fotografías originales, sin tener que hacer mediciones de la escena.
- **visibilidad común.** Este problema consiste en ordenar de forma adecuada los objetos sintéticos y los reales durante la composición, de tal forma que unos puedan ocultar a otros. La solución común a este problema es el uso de un Z-buffer [27].
- **iluminación común.** Consiste en calcular los cambios en la iluminación de las escenas reales producidos desde fuentes de luz sintéticas, y en la iluminación de los objetos sintéticos desde fuentes de luz reales.

Fournier se centra en el problema de la iluminación común y aplica radiosidad sobre una estimación de la luz presente en la escena, calculada a partir de las imágenes de entrada. La solución tiene en cuenta tanto la iluminación directa como la indirecta. El problema de esta aproximación es que el usuario debe modelar toda la escena a base de paralelepípedos, y además debe capturar una vista ortográfica de las caras de cada caja que haya definido. Además, se deben colocar marcas en los objetos de la escena para poder recomponer los parámetros de la vista, o para capturar el posible movimiento de los mismos, en el caso de que haya animación. El proceso es lento, porque a partir del valor de los píxeles de las imágenes se debe calcular el índice de reflexión de cada polígono, y modelar las luces presentes en la escena, si es posible.

Posteriormente, Drettakis [24] extendió el trabajo de Fournier introduciendo técnicas de visión por computador para la reconstrucción de la geometría 3D de la escena. Debido a que la información se obtiene directamente de las imágenes, este sistema necesita un campo de visión grande. Para obtener dicho campo de visión, introduce el uso de imágenes panorámicas construidas a partir de fotografías solapadas. Aún así, este método todavía necesita que el usuario introduzca una serie de vértices y defina la topología de todos los objetos de la escena.

Otros trabajos que han tratado de alguna manera la integración realista de elementos sintéticos y reales, teniendo en cuenta la iluminación son [77, 42, 85]. Trabajos actuales permiten modificar las luces de la escena casi interactivamente, llegando incluso a “apagar” luces reales o incluir luces sintéticas que afectan a toda la escena. Esto necesita la reconstrucción de la geometría de la escena real a un nivel de precisión aceptable (normalmente aplicando técnicas semiautomáticas) y radiosidad [55, 54, 85].

3.2. Diseño del sistema

El trabajo presentado en este capítulo se basa en la metodología desarrollada por Debevec en [22, 20]. Aunque dicha técnica tiene algunas restricciones, siendo la más importante de ellas que el impacto de la inserción de los objetos sintéticos en la escena real tiene un rango delimitado, el procedimiento descrito es el que ofrece mayor calidad con respecto a sus necesidades técnicas.

El procedimiento se basa en la definición de tres componentes:

- **Escena lejana.** Es la parte visible de la escena real que se encuentra lo bastante lejos de los objetos sintéticos como para que se vea afectada por ellos.
- **Escena local.** Es la parte de la escena que se verá afectada significativamente por la presencia de los objetos.

■ Objetos sintéticos.

La Figura 3.1 muestra un esquema de los intercambios de luz entre los componentes que se acaban de describir. Como se puede ver, la posible influencia de la escena local y los objetos sintéticos sobre la escena lejana se ignora, y por lo tanto, no hace falta capturar ni su geometría ni sus propiedades de respuesta ante la luz con detalle.

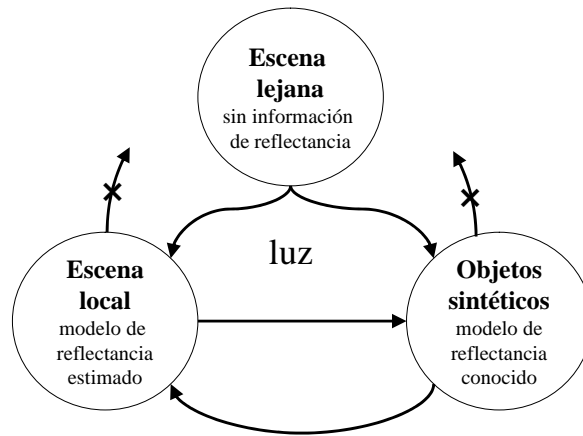


Figura 3.1: Componentes de la escena y flujo de intercambio de luz

Para iluminar los objetos sintéticos, se utilizará la luz que realmente se encuentra en la escena. En las técnicas que no capturan el mapa de entorno de la escena, se debe modelar la posición y las propiedades emisoras de las fuentes, bien por técnicas manuales, o bien por técnicas de visión por computador. Para capturar la luz presente en la escena utilizaremos un mapa de entorno luminoso de alto rango dinámico (véase el Apartado 2.3.5). Dicho mapa de luz se puede construir de varias formas, incluyendo más o menos información geométrica. Lo importante es que ofrezca una medida precisa de la luz que llega a la posición donde se desea insertar los objetos. Esa es la tarea de la escena lejana: almacenar la luz que había en la escena original [3, 2].

La escena local está formada por todas aquellas superficies de la escena real que se vayan a ver afectadas por la inclusión de los objetos sintéticos, incluyendo sombras, reflejos o cambios en su iluminación. Esta escena sí que podrá interactuar con los objetos sintéticos, y por lo tanto, modificar su aspecto. Para que esta escena participe en la solución, se deberá conocer tanto su geometría como sus características materiales, al menos aproximadamente. Normalmente, la escena local se reduce al plano donde se apoyarán los objetos sintéticos. Sin embargo, se puede querer incluir en el cálculo de la escena local una superficie altamente reflexiva en una pared lejana a los

objetos sintéticos, que debe recoger su reflejo. Esta es la forma por la que se pueden modificar porciones de la escena lejana: incluyéndolas en la escena local. Si la escena local se reduce al plano de apoyo, se puede calcular fácilmente su posición mediante los parámetros extrínsecos derivados de la calibración de la cámara, como se verá en un apartado posterior. Si la escena local incluye algo más aparte de dicho plano, existen métodos tanto activos (escáners 3D) como pasivos (directamente a partir de las imágenes, por estereometría u otros métodos de captura a partir de imagen) para recuperar su geometría.

Dependiendo del tipo de relaciones que se desee simular entre los objetos sintéticos y la escena real, los requisitos a nivel de modelado de la misma cambiarán. Así, si se desea que los objetos reales oculten a los sintéticos, modifiquen su apariencia proyectándoles sombras o reciban reflejos de los objetos sintéticos, habrá que obtener el modelo geométrico (y seguramente la reflectancia) de los objetos reales que entren en juego [42]. El principal problema que se presenta es capturar la geometría 3D con precisión suficiente para evitar posibles artificios en la etapa de composición.

Si en la composición de la imagen final se sustituyera la parte de la fotografía de fondo real por la escena local generada por el software de visualización 3D, la integración fallaría en muchos casos, debido a que normalmente se podrá apreciar la transición entre la parte real y la generada. Esto es debido a la dificultad de estimar de forma precisa las características físicas de los materiales. Por ello, lo que se calcula es el cambio que produce la inclusión de los objetos sintéticos en la iluminación de la escena local para luego, en un proceso de *rendering* diferencial, incluir dichos cambios en la fotografía original de la escena. La técnica de *rendering* diferencial se explicará en un apartado posterior. En la Figura 3.2 se puede ver el proceso a seguir para crear el efecto. Es un esquema detallado, en el que la mayoría de los procesos ocurren dentro de la computadora, y no necesitan intervención del usuario.

El esquema muestra los procesos principales a desarrollar. Cada proceso se encuentra en una caja del esquema. Las flechas que unen cada una de las cajas indican precedencia temporal, es decir, el proceso de una caja a la que llegan, por ejemplo, dos flechas, no se puede realizar hasta que no se hayan completado los dos procesos desde los que parten las flechas. Los procesos fuente, que no necesitan de otros procesos para su realización, son procesos externos que ha de realizar el usuario antes de iniciar el sistema.

La creación del efecto comienza con una sesión fotográfica que se debe planificar con extremo cuidado, ya que sus resultados incidirán directamente en la calidad del producto final. Además, dicha planificación es importante ya que puede ser necesario repetir todo el proceso por una sesión mal proyectada. En apartados posteriores se mostrarán los pasos a seguir.

El otro proceso externo (que debe realizar el usuario sin asistencia del sistema) es el diseño de los objetos sintéticos. Para ello puede utilizar cual-

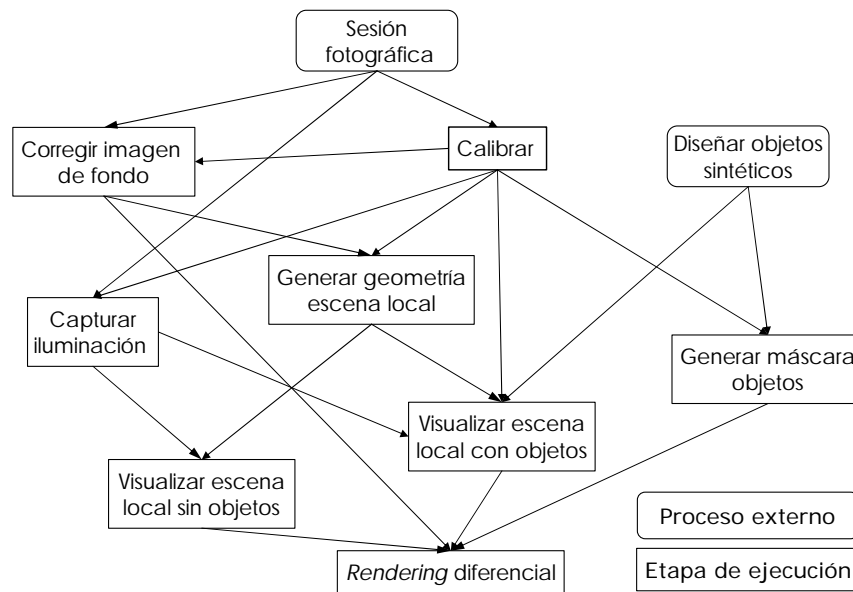


Figura 3.2: Esquema general del proceso de creación del efecto

quier aplicación que sea capaz de generar los formatos de escena 3D más conocidos, como *.3DS* de 3DStudio, *.DXF*, etcétera.

La mayor parte del procesamiento posterior prácticamente no necesita intervención del usuario. A continuación, se estudiarán uno a uno los procesos vistos en la Figura 3.2.

3.2.1. Sesión fotográfica

Este proceso es el que tiene mayor impacto en el resultado final, ya que los errores cometidos durante esta etapa tienen difícil corrección con posterioridad.

Durante la sesión fotográfica, el objetivo principal es el de obtener la imagen de la escena en la que se desea integrar los objetos sintéticos. Sin embargo, hay toda una serie de subtarefas que hay que cumplir durante la sesión fotográfica, enumeradas en la Figura 3.3.

El material que se necesita típicamente para una sesión fotográfica es el siguiente:

- Una cámara digital o, en su caso, una cámara normal y un escáner.
- Un trípode.
- Una plantilla de calibración de la cámara.
- Una esfera de material altamente reflexivo.

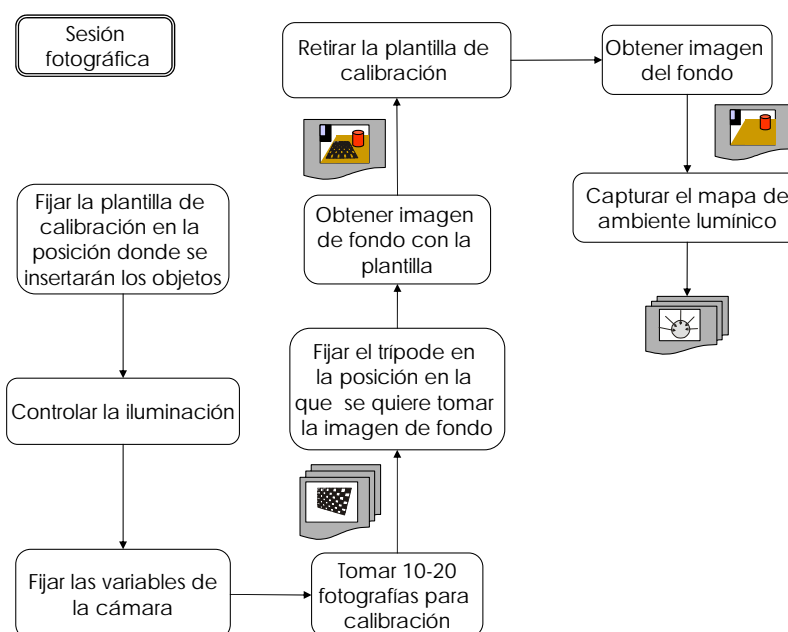


Figura 3.3: Pasos a seguir durante la sesión fotográfica

- Una cinta métrica.

A continuación se muestran con más detalle cada uno de los pasos a seguir durante la sesión fotográfica.

Fijar la plantilla de calibración donde se insertarán los objetos

Una vez que se ha preparado la escena donde se desea insertar los objetos sintéticos, el primer paso es fijar la plantilla de calibración. Aunque el proceso de calibración se realizará más tarde y se explicará en un apartado posterior, es necesario hacer una pequeña introducción para comprender qué se pretende conseguir en este paso.

La calibración de una cámara consiste en calcular los parámetros extrínsecos e intrínsecos utilizados en el momento de obtener la fotografía [40, 89]. Existen métodos automáticos para la obtención de dichos parámetros que tan sólo se apoyan en una serie de fotografías tomadas a un objeto de geometría conocida. El objeto que utiliza la implementación actual del sistema es el propuesto por la biblioteca *Open Source Computer Vision Library* (OpenCV en adelante) [99]. Es una biblioteca de código abierto, respaldada por *Intel* y utilizada en la actualidad por muchos investigadores independientes. Es una biblioteca genérica de funciones básicas para la implementación de aplicaciones de Visión por computador.

El marcador de calibración es una especie de tablero de ajedrez, impreso sobre un papel y adherido al plano donde se introducirán los objetos sintéticos. La Figura 3.4 muestra una miniatura de dicha plantilla.

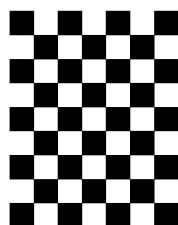


Figura 3.4: Ejemplo de plantilla de calibración

El tablero utilizado puede ser de cualquier dimensión, aunque el ancho y alto de los cuadros debe ser igual, y se aconseja utilizar un número distinto de cuadros horizontales y cuadros verticales para evitar un eje extra de simetría.

Una vez que se ha construido la plantilla, y se han comprobado las medidas de los cuadros impresos, ésta se debe fijar a la superficie donde se van a insertar los objetos sintéticos. Se debe prestar atención para que el papel quede lo más plano posible y asegurarse que la plantilla no se mueva durante la toma de fotografías (se puede utilizar cinta adhesiva para fijar la plantilla). También se debe tener en cuenta que la plantilla se pueda retirar posteriormente sin modificar la escena.

Controlar la iluminación

Antes de empezar a tomar fotografías, es importante asegurar que el nivel de iluminación va a permanecer constante durante la sesión, ya que durante esta actividad se van a tomar varias series de fotografías. Así, hay que capturar la iluminación de la escena (lo que requiere alrededor de 20 fotografías), tomar fotografías de calibración (unas 10) y la fotografía de fondo que se utilizará al final del proceso para insertar los objetos sintéticos. Durante el proceso de captura del mapa de entorno se debe asegurar que la iluminación sea lo más uniforme posible, especialmente si se utiliza luz natural.

Fijar las variables de la cámara

Como se ha visto anteriormente, la calibración se va a basar en una serie de fotografías tomadas desde distintas posiciones de un mismo objeto, y como resultado de dicha operación se determinará, entre otros parámetros, la

distancia focal de la cámara. Para determinar dicha distancia es importante que, mientras se capturan las imágenes para la calibración, y la posterior toma de la imagen de fondo, no se modifique la distancia focal. Por ello, durante este proceso no se debe manipular el zoom de la cámara.

La gran mayoría de las cámaras digitales tienen sistemas de ayuda al usuario que permiten modificar los parámetros de la cámara automáticamente, para obtener la mejor fotografía en las condiciones particulares de cada momento. Esto puede acarrear problemas durante la sesión fotográfica, especialmente cuando se capture la luz presente en la escena. En todo caso, si la cámara no permite controlar manualmente todos los parámetros, no habrá que modificar el zoom en ningún momento, y habrá que intentar que los parámetros de la cámara no varíen.

Tomar 10-20 fotografías para calibración

Para evitar que aparezcan problemas durante el cálculo de la calibración, es importante asegurarse que las fotografías se tomen con orientaciones bien diferenciadas, intentando cubrir la mayor parte de los posibles ángulos de visión de la plantilla de calibración.

Los parámetros que afectan a la calidad de la calibración, en orden de importancia decreciente son: el número y las orientaciones de las imágenes de entrada, la capacidad de encontrar automáticamente el tablero y el diseño de la plantilla de calibración [13].

Respecto al número de imágenes de entrada, la calidad del resultado es mayor cuantas más imágenes estén disponibles. De hecho, a partir de un número mínimo de imágenes, hay parámetros que no se pueden calcular. Por ejemplo, con una imagen, el punto principal no es observable, y se puede esperar estimar la longitud focal y la razón de aspecto únicamente si la orientación del plano es genérica. Si el plano de la plantilla de calibración es paralelo al plano focal, no se podrá estimar nada. Si la coordenada X o Y de la normal a la superficie es cero, entonces no se podrá estimar la razón de aspecto. Dos imágenes son la cantidad mínima para estimar, en ciertas condiciones, casi todos los parámetros, pero ahí se encuentra el límite de lo que se puede estimar. Con la cantidad de información mínima los errores de estimación pueden ser grandes (muy dependientes de la precisión en la localización de los puntos en la imagen), por lo que no se recomienda utilizar únicamente dos imágenes. Si las orientaciones de los planos son distintas, con tres imágenes se puede obtener el modelo completo, pero el sistema seguirá siendo muy sensible al ruido [89].

La cantidad de imágenes a utilizar dependerá de la aplicación. Si se quiere obtener una calibración de calidad, quizá para reconstruir la geometría de la escena a partir de las fotografías, se deberán tomar entre 15 y 20 imágenes. Si únicamente se requiere una aproximación, 5 pueden ser suficientes.

Para evitar problemas a posteriori, es preferible tomar más fotografías

de la cuenta, en previsión de que haya que descartar alguna. También es necesario asegurarse de que se toman las fotografías desde distintas posiciones y orientaciones.

El segundo aspecto en importancia que influye en la calidad del resultado de la calibración final es la facilidad de la extracción de los puntos de control. La calibración se basa en encontrar qué transformaciones hay que realizar en las coordenadas de los puntos de control que se encuentran en la plantilla de calibración (que son conocidos, ya que se sabe a priori la forma de dicha plantilla) para obtener una proyección perspectiva de los mismos que coincida con los puntos encontrados en la imagen de la fotografía. Por ello, en un sistema automatizado es vital la capacidad de extraer los puntos de control de forma precisa. En la Figura 3.5 se puede ver el resultado de una búsqueda automática de puntos de control. En el sistema de calibración utilizado, se toman como puntos de control a utilizar las esquinas de la plantilla que tienen cuatro cuadros a su alrededor (todas las esquinas de todos los cuadros, excepto las de los exteriores).

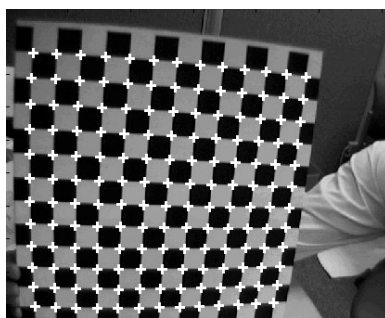


Figura 3.5: Esquinas encontradas de la plantilla de calibración [99]

Para facilitar la búsqueda automática de los puntos de control, se aconseja utilizar una plantilla de calibración donde los puntos sean simétricos (por ejemplo, en forma de tablero de ajedrez), y que tenga tanto contraste como sea posible (idealmente, blanco y negro). Se recomienda alcanzar una precisión de 0,1 píxeles, cosa que consiguen los detectores automáticos actuales.

Para conseguir unas fotografías claras, en las que se puedan detectar con precisión las esquinas del tablero, hay que tener cuidado en esta etapa y, o bien utilizar un trípode para fijar la cámara, o bien utilizar una velocidad de obturación lo suficientemente rápida (mínimo 1/125 de segundo).

Por último, el parámetro que afecta a la calidad en menor medida es, sorprendentemente, el diseño de la plantilla de calibración. En efecto, es más importante asegurarse que las imágenes utilizadas para la calibración estén limpias, que las orientaciones relativas de la plantilla de calibración sean tan genéricas como sea posible y que se encuentren los puntos de control con una

buena precisión que invertir recursos en una plantilla de calibración perfecta. Por ello, para construir la plantilla, basta utilizar una buena impresora láser, y asegurarse de que queda fijada al plano de la escena. La Figura 3.6 muestra una serie de fotografías tomadas durante esta etapa.

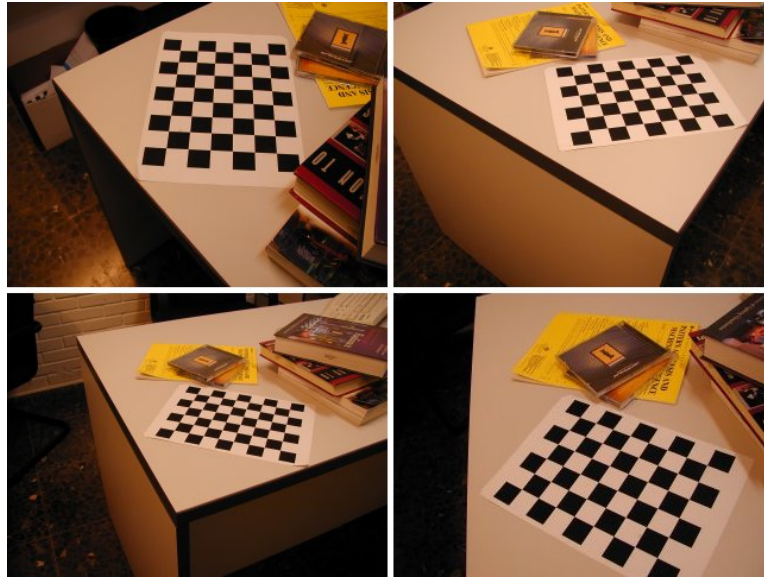


Figura 3.6: Ejemplo de imágenes tomadas para calibración

Fijar el trípode para tomar la imagen de fondo

A continuación, y sin modificar la configuración de la cámara con la que se han tomado las fotografías destinadas a la calibración, hay que fijar la cámara al trípode orientándola hacia el lugar donde se va a tomar la fotografía que servirá como fondo, y en la que se integrarán los objetos sintéticos. Es importante asegurar que el trípode no se moverá de su posición, ya que posteriormente se realizará otra fotografía desde el mismo lugar.

La plantilla de calibración debe permanecer fija en el mismo lugar.

Obtener imagen de fondo con la plantilla

Una vez fijado el trípode, tomar la fotografía de fondo. La plantilla de calibración permitirá calcular con posterioridad la posición de la cámara respecto a la escena (parámetros extrínsecos). La Figura 3.7 muestra la imagen que se utilizará como ejemplo.



Figura 3.7: Imagen de fondo tomada con la plantilla de calibración

Retirar la plantilla de calibración

En un paso posterior se va a necesitar conocer dónde se encontraba el centro de la plantilla respecto de la escena, por lo que en este paso puede ser útil marcar de alguna manera dicha posición. A continuación, retirar la plantilla de calibración.

Obtener imagen del fondo

En este momento, se procederá a tomar la fotografía que se compondrá digitalmente para integrar los objetos sintéticos (Figura 3.8). Nótese que no se ha movido el trípode, ni se han modificado los parámetros de la cámara, por lo que la fotografía tomada anteriormente con la plantilla de calibración define el conjunto de valores de los parámetros extrínsecos e intrínsecos de esta imagen.

Hay que tener en cuenta que, al modificar la escena haciendo desaparecer la plantilla de calibración, se ha modificado la luz presente en la misma, ya que la plantilla tiene colores saturados. Por ello, puede ser necesario ajustar la cámara para obtener una fotografía adecuada a la nueva configuración. En ningún caso se deberá modificar el zoom.

Capturar el mapa de entorno

En esta fase se va a tomar una serie de fotografías que posteriormente servirán para calcular el mapa de entorno luminoso de la escena, y que se



Figura 3.8: Imagen de fondo

utilizará para calcular la luz que recibirán los objetos sintéticos al situarse dentro de la escena real. Para ello, se sigue la técnica desarrollada por Paul Debevec [22, 20], consistente en fotografiar varias veces una sonda de luz (una esfera metálica), con distintos tiempos de exposición. Por ello es importante mantener constante la iluminación de la escena durante el proceso que se describe a continuación.

Como se introdujo en el Apartado 2.3.5, se debe colocar la sonda aproximadamente en la posición donde se encontraba el centro de la plantilla de calibración. En la Figura 3.9 se puede ver una posible configuración de referencia para la colocación de la cámara para capturar el mapa de entorno luminoso. Se puede ver la posición de la cámara desde la que se ha tomado la fotografía de fondo, la posición en la que se encontraba la plantilla de calibración, la sonda de luz y las posiciones para capturar el mapa de entorno luminoso.

Así, en primer lugar se coloca la sonda de luz en el centro de donde se encontraba la plantilla de calibración. A continuación se va a tomar una serie de fotografías desde dos posiciones separadas por 90° y equidistantes a la sonda, por lo que puede ser útil planificar de alguna manera dichas posiciones antes de empezar a tomar fotografías. Es importante notar que una de las dos posiciones debe tener una dirección cercana a la dirección desde donde se tomó la fotografía de fondo. Posteriormente hará falta una estimación de la geometría de la escena lejana, por lo que hay que anotar tanto la forma aproximada del entorno que rodea a la escena local, como la posición del

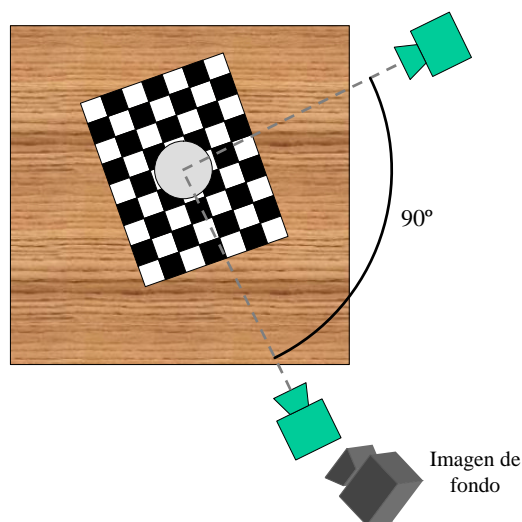


Figura 3.9: Configuración de las cámaras

tablero dentro de aquella. En escenas interiores, normalmente una caja del tamaño aproximado de la habitación bastará. Si es una escena exterior, una caja centrada en la sonda de luz y de tamaño grande permitirá replicar la estructura de luz presente en la escena.

Es conveniente que la cámara tenga una distancia focal grande para evitar distorsiones perspectivas. Para ello, hay que separar la cámara de la sonda, y utilizar el zoom para hacer que la imagen de la esfera ocupe buena parte de la fotografía.

Para capturar el mapa de entorno, hay que situar el trípode de tal forma que la cámara se encuentre a la misma altura del centro de la sonda y la vista paralela al suelo. De esta forma, para tomar el segundo conjunto de fotografías, habrá que mover el trípode hasta la nueva posición, pero no habrá que modificar la vista de la cámara (ambas posiciones deben ser equidistantes a la sonda). Cabe hacer hincapié en que una de las dos posiciones de referencia debe estar en la misma dirección que la imagen utilizada como fondo, aunque normalmente estará a una altura menor.

Una vez que se tiene situado el trípode, hay que configurar la cámara para la sesión. Se aconseja tomar entre 8 y 16 fotografías para capturar el mapa de entorno luminoso (el número mínimo depende del rango de radiancias presente en la escena), con un *stop* de diferencia entre ellas. Para ello, se aconseja estudiar en un primer momento las velocidades de obturación mínima y máxima, tal que en la velocidad mínima no se sature por completo la imagen, y se puedan observar los detalles de las zonas oscuras, y en la máxima se capturen los detalles de las zonas que emiten luz. Una vez que se

han estimado dichas velocidades, se pasará a tomar las fotografías, con las velocidades intermedias, dividiendo entre dos el tiempo de obturación entre una fotografía y la siguiente.

Como resumen, a continuación se muestran una serie de consejos a tener en cuenta durante esta fase [3, 2]:

- Las dos posiciones utilizadas para capturar el mapa de entorno deben ser equidistantes a la sonda para que la esfera sea del mismo tamaño en los dos conjuntos de fotografías.
- Una de las dos posiciones debe tener la misma dirección que se utilizó para capturar la imagen de fondo.
- Antes de empezar a tomar fotografías, hay que asegurarse que la sonda no va a introducir ruido en el mapa de entorno (sombras o reflejos visibles).
- Comprobar que el nivel máximo de luz en la escena es capturado convenientemente, con una velocidad adecuada de obturación.
- Utilizar un zoom grande.
- Asegurarse que la fotografía abarca toda la sonda.
- Desactivar el flash.
- Asegurarse de que la cámara no se mueve en ningún momento.

Para cada una de las fotografías que se tomen, hay que anotar la siguiente información para su utilización posterior: velocidad de disparo, apertura, la ganancia de la cámara y el valor del filtro de densidad neutral utilizado, si se ha utilizado alguno. Es útil preparar un formulario para ir rellenando los huecos durante la sesión.

Dependiendo del diseño de la cámara, y lo fácil que sea manipular sus parámetros de velocidad de obturación y apertura, puede ocurrir que al cambiar la velocidad de una fotografía a la siguiente, se mueva ligeramente la cámara. Este movimiento provocará que las imágenes no estén alineadas, y por lo tanto los bordes de los objetos vistos en el mapa de entorno resultante aparezcan difusos. El apartado siguiente incide en éste y otros problemas que pueden surgir en esta fase.

Para generar la imagen de radiancias de alto rango dinámico a partir de las fotografías anteriores, se utilizará la aplicación `mkhdr`, disponible en la página web de Paul Debevec [94]. Dicha aplicación trabaja en modo consola, y el formato de utilización es el siguiente:

```
mkhdr [opciones] entrada salida
```

donde las opciones pueden ser:

- **-w fichero:** guarda la curva de respuesta recuperada en el fichero.
- **-c fichero:** utiliza la curva almacenada en el fichero para calcular el mapa de radiancias.
- **entrada:** archivo de texto que indica las fotografías y parámetros a utilizar.
- **salida:** archivo donde se guardará el mapa de radiancias.

El archivo de entrada debe tener formato ASCII, y contiene una línea por cada fotografía utilizada para generar la imagen de alto rango dinámico. Cada línea contiene la siguiente información:

- nombre del archivo (formato .ppm),
- inversa del tiempo de exposición,
- apertura,
- ganancia y
- densidad neutral.

A continuación se muestra un ejemplo del contenido de este archivo:

```
img0076.ppm 1000      8.0  3  0
img0075.ppm 500       8.0  3  0
img0074.ppm 250       8.0  3  0
img0073.ppm 125       8.0  3  0
img0072.ppm 60        8.0  3  0
img0071.ppm 30        8.0  3  0
img0070.ppm 15        8.0  3  0
img0069.ppm 8         8.0  3  0
img0068.ppm 4         8.0  3  0
img0067.ppm 2         8.0  3  0
img0066.ppm 1         8.0  3  0
img0065.ppm 0.5       8.0  3  0
img0064.ppm 0.2666667 8.0  3  0
img0063.ppm 0.1333333 8.0  3  0
img0062.ppm 0.0666667 8.0  3  0
img0061.ppm 0.0333333 8.0  3  0
```

El primer campo de cada línea es el nombre del archivo que contiene la fotografía. El único formato gráfico aceptado por la aplicación es el **Portable Pixel Map** (extensión .ppm). El siguiente campo es la inversa del tiempo de obturación (si se ha utilizado un tiempo de obturación de 1/250, el valor será 250) y luego aparece la apertura. Los dos últimos campos indican la

ganancia que se ha utilizado al tomar la fotografía (se asume que está en decibelios, y un valor positivo indica que la señal se ha magnificado) y el valor del filtro de densidad neutral utilizado. Un filtro de densidad neutral es aquel filtro que reduce la luminancia de la luz que lo atraviesa de la misma forma en todas las longitudes de onda.

El valor de la densidad neutral está especificado en logaritmo base 2. Por ejemplo, un 2 indica que se utilizó un filtro de densidad neutral que absorbía un 75 % de la luz:

$$\log_2 \frac{1}{0,25} = 2$$

Las imágenes utilizadas en el ejemplo varían sólo en la velocidad de obturación. A pesar de que la aplicación tiene en cuenta la apertura, ganancia y filtrado, dichos valores son típicamente menos precisos que la velocidad de obturación, y es conveniente dejarlos constantes durante la toma de fotografías [22]. Por ejemplo, cambiar la apertura desde $f8.0$ a $f5.6$, normalmente deja pasar apenas el doble de luz, y de hecho produce a menudo una variación espacial en la modulación del brillo (*vignetting*), particularmente en objetivos gran angular. Sin embargo, no hay problema en recuperar una curva de respuesta bajo un conjunto constante de apertura, ganancia y filtrado de densidad neutral, y luego utilizar dicha curva para construir mapas de radiancia a partir de imágenes tomadas bajo un conjunto diferente de parámetros. Si las circunstancias lo requieren, o si se asume que la apertura, ganancia o filtrado son lo suficientemente precisos, puede no haber problema en variarlos mientras se captura un mapa de radiancia que se construya desde una curva de respuesta preexistente, pero en general no es buena idea variar dichos parámetros mientras que se toma una secuencia que se utilizará para recuperar la curva de respuesta de la película o CCD.

Como se ha comentado antes, el formato de las imágenes de entrada ha de ser *.ppm*. Además de convertir las imágenes desde el formato generado por la cámara, hay que recortar la zona que contiene información (es decir, el círculo en la imagen donde se ha proyectado la esfera). Para ello hay varias posibilidades. Lo ideal es utilizar alguna herramienta de tratamiento fotográfico con lenguaje de script, como GIMP o Adobe PhotoShop, para poder automatizar el proceso de recortado y conversión. También se pueden utilizar herramientas de retoque fotográfico que no tengan esta posibilidad, pero que permitan exportar la selección de una imagen a otra. De cualquier modo, hay que asegurarse de que las imágenes finales estén perfectamente alineadas.

En el caso de que la aplicación seleccionada para recortar las imágenes no permita almacenar la imagen en formato *.ppm*, se puede utilizar un paquete de conversión de formatos de archivo gráficos como ImageMagick [97].

Una vez terminado el proceso sobre las dos series de imágenes, se crearán los archivos de configuración necesarios para que **mkhdr** genere las dos imáge-

nes de alto rango dinámico, y se lanzará la aplicación tal y como se ha mostrado antes.

El resultado de la aplicación `mkhdr` es un archivo de imagen de alto rango dinámico que almacena, para cada píxel, la radiancia presente en el área correspondiente de la escena original. La extensión por defecto es `.pic`. Dicho archivo está diseñado para albergar imágenes de alto rango dinámico [49], y almacena los valores de radiancia en coma flotante. La idea básica de dicho formato es almacenar un byte de mantisa para cada color primario, y un byte común de exponente. La precisión de estos valores será del orden del 1 % sobre un rango dinámico desde 10^{-38} a 10^{38} . El formato exacto del archivo se puede encontrar en la documentación adjunta a la aplicación, pero normalmente no es necesario conocerlo, ya que RADIANCE proporciona las herramientas necesarias para visualizar dicho archivo (permitiendo ajustar la exposición, para ajustar el rango dinámico al rango del dispositivo de salida) y para generar imágenes en formatos gráficos estándar (por ejemplo, *TIFF*). Una vez que se convierte una imagen de alto rango dinámico a un formato gráfico estándar, se pierde la mayor parte de la información de radiancia que contenía el archivo original, ya que se ha debido escalar de alguna manera los valores de radiancia al rango dinámico del dispositivo de salida.

Si se aplica el procedimiento anterior a los dos conjuntos de fotografías tomadas desde dos ángulos distintos, se habrán obtenido dos imágenes de radiancia de alto rango dinámico, en las que aparecerá la cámara (y quizá el fotógrafo). En un apartado posterior se mostrará una técnica para eliminar la cámara y el fotógrafo de la imagen.

Problemas durante el proceso

Si alguna de las imágenes de alto rango dinámico obtenidas no es la esperada, es probable que sea debido a la aparición de algunos de los artificios que se muestran a continuación, junto a su posible causa:

- **la imagen tiene los bordes de los objetos difuminados.** Esto es debido a que se ha movido la cámara en el proceso de captura de las fotografías de una serie. Una solución pasa por alinear las imágenes para hacerlas coincidir.
- **aparecen sombras extrañas.** Puede ser debido a pequeños movimientos de los elementos de la escena de una fotografía a otra. Mediante la técnica de eliminación de la cámara que se verá más adelante se pueden eliminar dichas sombras. El recuadro de la Figura 3.10 muestra una sombra provocada por el fotógrafo, que aparece en algunas de las fotografías utilizadas para la compilación de la imagen de alto rango dinámico.

- **aparecen colores inesperados.** No se han seleccionado bien las velocidades de obturación en el proceso de captura de las imágenes. La única solución pasa por volver a repetir la sesión fotográfica. La Figura 3.10 muestra este fallo, en la zona de las ventanas a la izquierda de la imagen.



Figura 3.10: Errores en la generación de la imagen de alto rango dinámico

Volviendo al problema del movimiento de la cámara, antes que nada es importante incidir en la necesidad de obtener la máxima calidad en los datos de entrada, es decir, que las serie de fotografías de la sonda no presenten movimientos entre una y otra. Una ayuda para conseguirlo puede ser utilizar un mando a distancia para disparar la cámara. De esta forma se consigue eliminar el posible movimiento inducido al pulsar el botón de disparo.

Si la captura de fotografías ya se ha realizado, las imágenes de la sonda no están alineadas y no hay posibilidad de repetición de la sesión fotográfica, una posible solución pasa por registrar (o alinear) las fotografías. Lo ideal sería que dicho registro se realizase automáticamente, sin intervención del usuario. En la actualidad hay técnicas que permiten realizar este proceso automáticamente, principalmente en los campos de creación automática de mosaicos [68], georreferenciación de imágenes de teledetección [62] y estabilización de imagen [16, 68]. En particular, se probó el sistema presentado en [78]. Dicho sistema se encarga de buscar automáticamente puntos de control en una serie de imágenes solapadas entre sí, obteniéndose como resultado una lista de puntos coincidentes en todas las imágenes. Una vez que se han obtenido dichos puntos, se puede calcular la transformación que hay que aplicar a cada imagen para hacerla coincidir con la que se utiliza como base. El proceso de obtener la transformación a partir de los puntos de control consiste en calcular la mejor transformación geométrica que convierte los puntos de la imagen que se desea transformar en los puntos correspon-

dientes de la imagen base. En el proceso se obtiene la mayoría de las veces un sistema de ecuaciones lineales sobredeterminado, que hay que resolver por mínimos cuadrados.

El proceso de alineación de fotografías tomadas de la esfera reflectante, con el objetivo de eliminar los pequeños movimientos de la cámara producidos al cambiar los parámetros de la misma parece un campo propicio para la aplicación de técnicas automáticas. Esto es debido a que a priori se sabe que los cambios de una fotografía a otra van a ser pequeños, y que dichos cambios, en la mayoría de las ocasiones, se podrán modelar como una translación (si, como hemos asumido, se utiliza un zoom grande y las imágenes simulan una proyección paralela). Sin embargo, la aplicación automática de la técnica anterior para registrar las imágenes de la sonda de luz presenta los siguientes problemas:

- es difícil encontrar información en las fotografías que se toman con una velocidad de obturación alta. Para facilitar la búsqueda de puntos de control significativos en dichas imágenes, se ha aplicado una expansión de histograma [35] sobre la imagen en color, y posteriormente se ha convertido a grises, y
- el proceso de compresión de algunas cámaras digitales provoca artificios (bordes falsos). Dichos bordes se hacen evidentes al realizar la expansión de histograma.

Como ejemplo de estos problemas, la Figura 3.11 muestra un ejemplo de búsqueda de puntos de control entre dos fotografías. La imagen de la derecha se corresponde con una fotografía tomada a alta velocidad, por lo que la mayor parte de la fotografía original aparecía oscura, después de aplicarle una ecualización del histograma. La aplicación presentada en [78] encontró 11 puntos de control, mostrados en ambas imágenes como cuadros blancos. Las flechas indican dos puntos de control que se han asignado erróneamente.

Debido a que la aplicación de rastreo de puntos de control devuelve un número pequeño de ellos, la aparición de errores afecta al resultado final del registro. La imagen responsable del pequeño número de puntos de control es la imagen de la derecha, ya que en la imagen que se utiliza como base se encontraron 56 puntos susceptibles de convertirse en punto de control, en contraposición con los 11 puntos encontrados en la imagen oscura.

Por lo que se acaba de presentar, la problemática asociada al registro de imágenes utilizadas para capturar el mapa de entorno no se puede atacar con los métodos tradicionales de registro. A continuación se enumeran dos posibles soluciones que podrían paliar el problema, y que requieren de un esfuerzo de investigación adicional que se sale del objetivo del presente trabajo:

- Postproceso de los puntos de control encontrados. Dicho postproceso debería tomar partido de las condiciones especiales del problema.

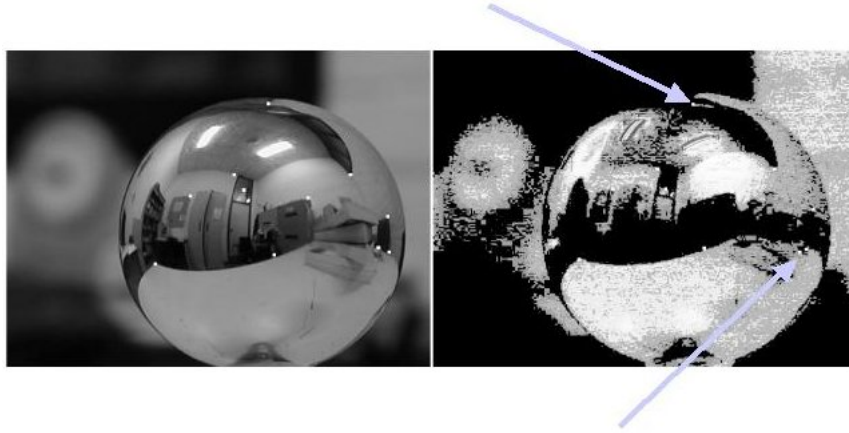


Figura 3.11: Fallos en la búsqueda de puntos de control

Es decir, se sabe que la transformación es pequeña, que tendrá una dirección predominante, y que será parecida en todos los puntos. De esta forma, se podrían rechazar aquellos emparejamientos de puntos de control que se aparten de la transformación predominante. Esta solución sería inaplicable en el caso de que se hubiera encontrado un número muy pequeño de puntos de control (6 o menos).

- Una aproximación completamente distinta a la anterior: búsqueda de la silueta de la sonda. Como se sabe la forma de la sonda (un círculo), su tamaño aproximado (un radio algo menor que la mitad de la dimensión menor de la imagen) y su posición (aproximadamente en el centro de la imagen), se podría aplicar una técnica de reconocimiento de formas para buscar dicha sonda. Ejemplos de técnicas aplicables serían la transformada de Hough [9], o, el detector de elipses de Fitzgibbon [26]. Una vez que se ha extraído la posición de la sonda en cada fotografía, se puede calcular la transformación correspondiente.

3.2.2. Calibración

Este apartado muestra los pasos a seguir para la calibración de una cámara a partir de las imágenes tomadas de la plantilla de calibración durante la sesión fotográfica. Este proceso se implementará a través de las funciones ofrecidas por la biblioteca OpenCV [99].

El primer paso consiste en encontrar los puntos de control de la plantilla de calibración en las imágenes de entrada. En el caso de la biblioteca OpenCV [99], la plantilla de calibración es, como se ha visto, una cuadrícula en forma de tablero de ajedrez, en la que los puntos de interés son las

esquinas interiores del tablero (ver Figura 3.4). La razón de utilizar dicha plantilla y dichos puntos de control es que permite utilizar unas funciones de búsqueda automática proporcionadas por la biblioteca, que tienen una precisión de subpíxel, lo que elimina por completo la necesidad de que el usuario indique la posición de los puntos de control en las fotografías.

Las funciones OpenCV implicadas en la búsqueda automática de puntos de control son:

- **cvFindChessBoardCornerGuesses**. Intenta determinar si la imagen de entrada sigue el patrón de un tablero de ajedrez, y busca las esquinas interiores del mismo. La función indica si se han encontrado todas las esquinas y si las ha podido ordenar (fila a fila, de izquierda a derecha). Las coordenadas que devuelve la función son aproximadas, y pueden tener un error de un par de píxeles respecto a la posición real en la imagen.
- **cvFindCornerSubPix**. Esta función itera la solución inicial para encontrar la localización de las esquinas con una precisión menor a un píxel.

Una vez que se han extraído las esquinas de la imagen, se puede pasar a calibrar la cámara. Para ello, se utilizará la función:

- **cvCalibrateCamera**. Calcula los parámetros de la cámara utilizando la posición 3D de los puntos sobre la plantilla de calibración respecto al sistema de coordenadas del mundo y las coordenadas 2D extraídas de las imágenes en el paso anterior. Esta función devuelve la matriz intrínseca de la cámara, que es común a todas las imágenes que se han utilizado para calibrar (durante la captura de las fotografías de calibración no se modificaron los parámetros de la cámara). Además, devuelve los parámetros extrínsecos (matriz de rotación y vector de translación) de cada imagen utilizada en la calibración.

Para definir la posición de las esquinas internas del tablero en el sistema de coordenadas del mundo, hay que definir la posición y orientación de dicho sistema de coordenadas. Si se siguen los pasos de calibración propuestos por OpenCV, el sistema de coordenadas del mundo está definido por la plantilla de calibración. Así, se asume que la plantilla de calibración está en el plano $Z = 0$. El origen está definido por la esquina interior superior izquierda de la plantilla. El eje de las X crece a lo largo de la misma fila definida por el origen, y el eje Y a lo largo de la misma columna del origen. OpenCV utiliza un sistema de coordenadas dextrógiro, por lo tanto, el eje Z crece perpendicularmente a la plantilla de calibración, hacia abajo. En la Figura 3.12 se puede ver un esquema de dicho sistema de coordenadas.

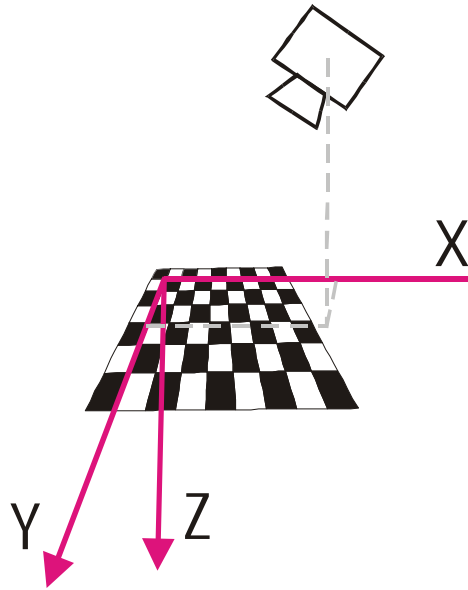


Figura 3.12: Sistema de coordenadas del mundo

En la Figura 3.13 se puede ver el sistema de ejes coordenados definido anteriormente sobre una imagen que se podría utilizar como fondo. La aplicación implementada muestra, además de los ejes con sus respectivos sentidos, una etiqueta que indica la longitud de los ejes en el sistema métrico que haya definido el usuario en la calibración (en el ejemplo se utilizaron metros). Dicha etiqueta facilita el posterior posicionamiento de los objetos sintéticos respecto al sistema de coordenadas.

Como resultado de este paso, en el que no se ha necesitado la intervención del usuario se ha obtenido:

- Matriz intrínseca de la cámara (distancia focal medida en unidades de píxel horizontal y vertical, punto principal),
- parámetros de la distorsión inducida por el objetivo,
- matrices extrínsecas de cada fotografía utilizada en la calibración (matriz de rotación y vector de translación), y
- marginalmente, se han obtenido las coordenadas de las esquinas de la plantilla de calibración para cada fotografía

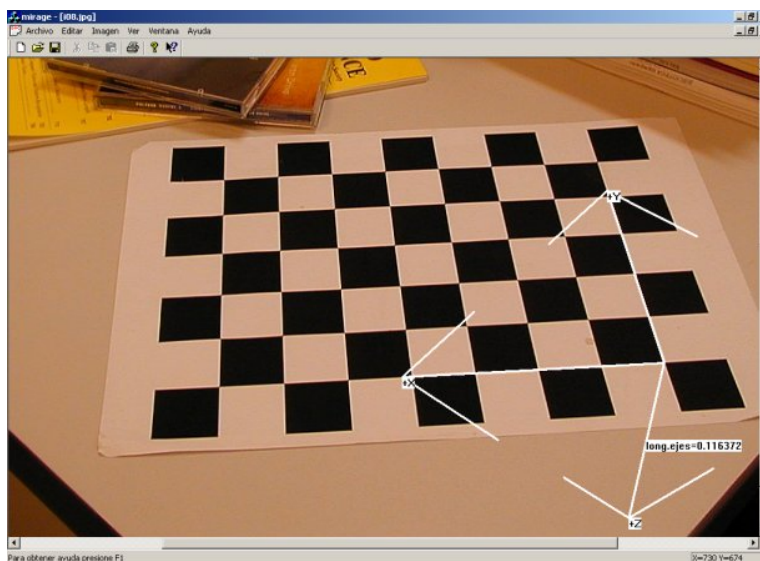


Figura 3.13: Ejes coordenados en una escena real

3.2.3. Corregir imagen de fondo

La biblioteca OpenCV permite, en un paso posterior a la calibración de la cámara, aplicar la transformación inversa a la producida por las aberraciones de su objetivo, para eliminar la distorsión de cualquier fotografía tomada por dicha cámara, siempre que no se hayan modificado los parámetros internos de la misma.

La función OpenCV que se encarga de corregir la distorsión de la cámara es:

- **cvUndistortOnce**. Corrige la distorsión de la cámara a partir de la matriz intrínseca y los coeficientes de distorsión calculados durante la calibración. Permite utilizar interpolación bilineal.

En este paso lo que se busca es corregir la fotografía que se utilizará como fondo para integrar los objetos. Dicha imagen no tiene plantilla de calibración y, por lo tanto, no se utilizó para calibrar la cámara, pero se le puede aplicar la función anterior, ya que al tomar las fotografías de calibración se mantuvieron los parámetros de la cámara. Este proceso, además de generar una imagen sin distorsiones que permite aliviar los problemas del uso de cámaras de baja calidad, crea una imagen en la que se podrán insertar con precisión los objetos 3D, ya que se conocen los parámetros precisos que definen la posición del plano donde se apoyarán los objetos.

3.2.4. Diseñar objetos sintéticos

La tarea de diseñar los objetos sintéticos es totalmente externa. Así, el usuario podrá utilizar cualquier entorno de diseño 3D al que esté habituado para construirlos. La aplicación que se va a utilizar para calcular la iluminación de los objetos es RADIANCE, la cual permite importar varios tipos de formatos de escena (entre ellos .3DS de 3DStudio). A la hora de diseñar los objetos sintéticos, se podrá utilizar cualquier característica que ofrezca el sistema de visualización.

3.2.5. Generación de geometría de escena local

En este proceso, el usuario debe crear la geometría de la escena local, que es la parte de la escena real que se modificará cuando se inserten los objetos sintéticos. Normalmente, dicha escena no será más que un plano. Para el usuario, la forma más sencilla para definir la posición y forma de dicho plano sería sobre la propia imagen que se utilizará como fondo. El usuario deberá marcar uno o más polígonos sobre la imagen, con lo que se obtiene una lista de coordenadas de píxeles (2D). A partir de dicha lista de píxeles, y asumiendo que el plano donde se van a apoyar los objetos es el plano que contenía la plantilla de calibración, se podrá reconstruir el polígono correspondiente en el sistema de coordenadas del mundo (3D). En la Figura 3.14 se muestra un ejemplo de definición de la escena local sobre la imagen de fondo.

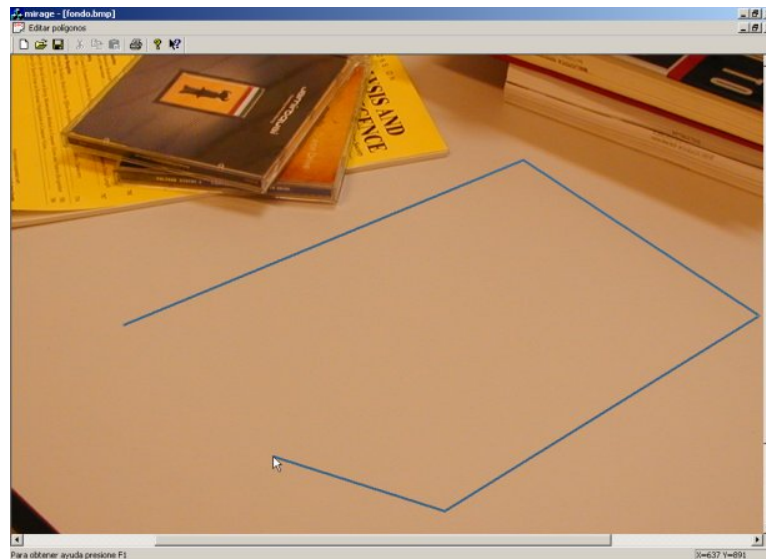


Figura 3.14: Definiendo la escena local sobre la imagen

El resultado de esta fase será un archivo en formato RADIANCE [49],

que podrá utilizarse directamente para generar una vista de la escena local bajo la iluminación capturada en la sesión fotográfica.

Durante el proceso de calibración se ha determinado que el sistema de coordenadas del mundo está definido a partir de la plantilla de calibración, y ésta se encuentra en el plano $Z = 0$. La Figura 3.15 muestra los sistemas de coordenadas implicados y la información necesaria para reconstruir la posición 3D de un punto sobre el plano $Z = 0$, a partir de un píxel en pantalla.

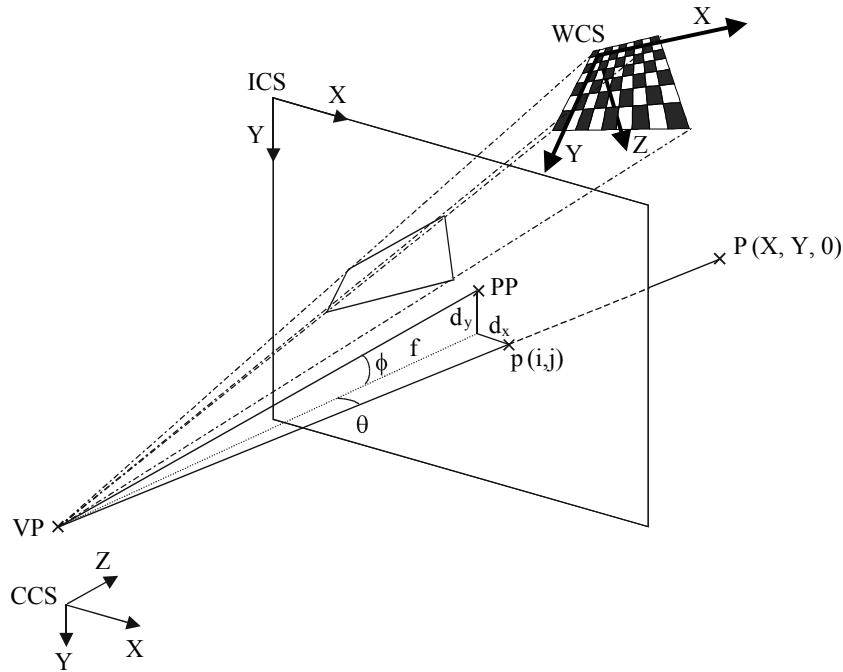


Figura 3.15: Reconstrucción de un punto 3D a partir de un píxel

En la Figura 3.15 se ha utilizado la siguiente notación:

- WCS: Sistema de coordenadas del mundo.
- CCS: Sistema de coordenadas de la cámara. El eje Z , define la dirección de la vista (vector VIEW), Y define el balanceo de la cámara y X , es ortogonal a los anteriores. Está definido en el WCS.
- VP: Posición del centro de proyecciones de la cámara respecto al WCS.
- ICS: Sistema de coordenadas bidimensional de la imagen.
- $p(i,j)$: Coordenadas del píxel seleccionado por el usuario.

- $P(X,Y,0)$: Punto sobre el plano $Z = 0$ que se desea calcular (respecto al WCS).
- PP: Punto principal de la imagen.
- f : Distancia focal.
- d_x, d_y : Distancia horizontal y vertical entre el píxel seleccionado y el punto principal.
- θ, ϕ : Distancia angular horizontal y vertical entre el píxel seleccionado y el punto principal.

Cálculo del punto 3D a partir del píxel

El problema consiste en calcular las coordenadas del punto P , que se encuentra en el espacio 3D sobre el plano $Z = 0$, a partir de las coordenadas sobre la imagen $p(i, j)$ proporcionadas por el usuario.

Los parámetros extrínsecos de la cámara, que definen la posición y la orientación dentro del mundo de ésta se han obtenido en el proceso de calibración en forma de matriz de rotación y vector de translación. Dichos parámetros se encargan de expresar las coordenadas de los objetos especificadas en el sistema de coordenadas del mundo en el de la cámara. Así, la matriz de rotación se encarga de alinear ambos sistemas de coordenadas, y el vector de translación indica cómo llegar de un origen al otro. Esto se puede especificar mediante la relación:

$$X_C = RX_W + T,$$

donde:

- X_C y X_W representan las coordenadas de un punto P en el sistema de referencia de la cámara y del mundo, respectivamente, y
- R es la matriz de rotación y T el vector de translación obtenidos en el proceso de calibración

Además, el vector T es el vector de coordenadas del origen del sistema de referencia del mundo con respecto al sistema de coordenadas de la cámara. Lo que se desea obtener, sin embargo, es la posición y orientación de la cámara respecto al sistema de coordenadas del mundo. Por ello, despejando X_W en la ecuación anterior se obtiene

$$X_W = R'X_C + T',$$

que se encarga de traducir las coordenadas que se encuentran definidas en el sistema de la cámara al sistema de coordenadas del mundo. La nueva matriz de rotación y el vector de translación vienen dados por:

$$\begin{aligned} R' &= R^{-1} = R^T \\ T' &= -R^T T \end{aligned}$$

Entonces, el origen del sistema de coordenadas de la cámara (el centro de proyecciones) se encuentra en las coordenadas definidas por T' respecto al sistema de coordenadas del mundo.

A partir de la matriz de rotación original R , se puede obtener la información necesaria para configurar la cámara sintética. Así, la última fila de la matriz es la dirección de la vista (eje Z , o vector VIEW), y la segunda el eje Y (ver Figura 3.16).

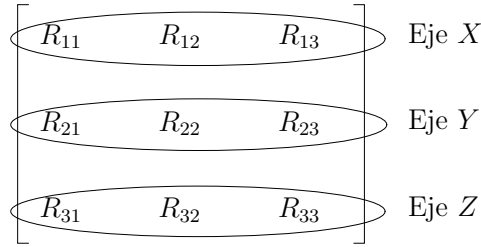


Figura 3.16: Interpretación de las filas de la matriz de rotación

Para calcular la intersección entre el rayo que pasa por el centro de proyecciones de la cámara y un píxel dado con el plano $Z = 0$, dada la calibración de la cámara, simplemente hay que invertir la homografía que relaciona dicho plano con la imagen.

Como se ha visto en el Apartado 2.3.3, la matriz H se puede calcular descartando la tercera columna de la matriz de proyección:

$$H = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix},$$

donde A es la matriz intrínseca de la cámara y r_1 , r_2 y t son la primera, segunda y cuarta columnas de la matriz extrínseca. La homografía H transforma los puntos del plano $Z = 0$ al píxel correspondiente, de la siguiente forma:

$$s \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = H \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

donde s es un factor de escala, (i, j) indica la coordenada en la imagen, y el punto a transformar tiene coordenadas $[X, Y, 0, 1]$.

Si lo que se desea es obtener las coordenadas de un punto en el plano $Z = 0$, expresado en el sistema de coordenadas global, a partir de su píxel correspondiente, se puede invertir la expresión anterior para obtener:

$$s \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$$

Dada la posición de la cámara en el sistema de coordenadas global y las coordenadas del punto 3D en el plano $Z = 0$ expresado en el mismo sistema de coordenadas, se puede calcular el rayo correspondiente.

Por supuesto, una vez que se ha reconstruido el rayo 3D que pasa por el centro de proyecciones y por el píxel indicado por el usuario, se podrá calcular su intersección con cualquier geometría que venga dada en el sistema de coordenadas del mundo, no sólo con el plano $Z = 0$.

Creación de la escena RADIANCE

A continuación se muestra un ejemplo de escena local (en este caso, únicamente un polígono sobre el plano $Z = 0$) obtenido por medio de la reconstrucción de la geometría a partir de las coordenadas de los píxeles dados por el usuario.

```
# Fichero generado por 'Mirage'. V1.0 (c) Francisco Abad. 2001.
# Grupo de Informática Gráfica. DSIC. U. Politécnica de Valencia

# CUIDADO! Este archivo se regenera automáticamente. No modificar

material_e_local polygon e_local_1
0
0
12
0.109706 -0.032862 0.000000
-0.056157 0.074919 0.000000
0.039075 0.211284 0.000000
0.254586 0.090432 0.000000
```

Las escenas RADIANCE se almacenan en un archivo ASCII en el que se definen un conjunto de primitivas, comentarios, invocaciones a programas externos o declaración de alias. A continuación se enumeran cada uno de los posibles componentes de una escena:

1. Comentarios. Son líneas que empiezan con el carácter #

```
# Descripción de la escena
```

2. Primitivas. Declaración de materiales, elementos geométricos, etc:

```

modificador tipo identificador
n S1 S2 S3...Sn
0
m R1 R2 R3...Rm

```

donde `modificador` puede ser la palabra reservada `void` o un identificador existente, y sirve para aplicar a la primitiva que se está definiendo una alteración, definida anteriormente con el identificador utilizado. `tipo` es la clase de elemento que se está definiendo, que puede ser un material (por ejemplo `plastic`, o `metal`), una forma geométrica (`sphere`, `polygon`...) u otro tipo (`pattern`, `texture`). Por último, el identificador sirve para nombrar esta primitiva en otro contexto. La segunda línea de cada primitiva es una serie de parámetros de tipo cadena de caracteres, separados por espacios. `n` es el número de dichos parámetros. La tercera línea correspondería con los parámetros de tipo entero, pero en la versión actual de RADIANCE no se utiliza. La última línea se corresponde con los parámetros de tipo real que necesita la primitiva, siendo `m` su número. El número de parámetros y su tipo depende del tipo de primitiva que se esté definiendo. En [49] y en la documentación adjunta a RADIANCE aparecen dichas posibilidades. A continuación se muestran dos ejemplos de declaración de primitivas. El primero es la definición de un material llamado `espejo`, que tiene un color prácticamente blanco (0.8, 0.8, 0.8), un alto índice de reflexión (0.9) y no presenta difusión de la luz. El segundo ejemplo es la definición de un polígono, que tendrá las propiedades reflexivas del tipo `madera` (que se ha debido definir antes). Para definir polígonos hay que proporcionar las coordenadas de sus vértices.

<code>void metal espejo</code>	<code>madera polygon pared</code>
<code>0</code>	<code>0</code>
<code>0</code>	<code>0</code>
<code>5 .8 .8 .8 .9 0</code>	<code>24 0 11 0</code>
	<code>6 11 0</code>
	<code>6 11 6.5</code>
	<code>11 11 6.5</code>
	<code>11 11 0</code>
	<code>11.75 11 0</code>
	<code>11.75 11 15</code>
	<code>0 11 15</code>

3. Llamadas a programas externos.

```
! genbox cemento base_casa 27.8 18.8 3
```

Estos programas externos permiten, por ejemplo, ayudar a definir geometrías complejas (en el ejemplo se está definiendo una caja de las dimensiones especificadas), aplicar transformaciones a los vértices de primitivas geométricas, etc. Los programas que se invoquen de esta forma deberán trabajar con el formato de escena RADIANCE (las respectivas distribuciones de RADIANCE están compuestas por varias decenas de programas que se invocan desde la línea de comandos).

4. Alias de identificadores.

```
modificador alias identificador_nuevo identificador_viejo
```

Permite crear sinónimos de tipos ya existentes.

Para simular la interacción de la luz entre la escena local y los objetos sintéticos, aparte de la geometría, hay que definir de alguna forma las propiedades físicas de la escena local, es decir, sus propiedades de reflectancia. La interacción de la luz con una superficie se puede expresar mediante una única función, llamada función de distribución bidireccional de reflectancia (*bidirectional reflectance distribution function*, o BRDF) [81]. Esta es una función de cuatro ángulos, dos incidentes y dos reflejados, y permite describir la radiancia emitida en función de la radiancia incidente:

$$L_r(\theta_r, \phi_r) = \int_0^{2\pi} \int_0^{\frac{\pi}{2}} L_i(\theta_i, \phi_i) \rho_{bd}(\theta_i, \phi_i; \theta_r, \phi_r) \cos \theta_i \sin \theta_i d\theta_i d\phi_i$$

donde:

- ϕ : ángulo azimutal, medido en el plano XY a partir del eje X.
- θ : ángulo polar, que mide la elevación sobre el plano XY.
- $L_r(\theta_r, \phi_r)$: radiancia reflejada en la dirección indicada.
- $L_i(\theta_i, \phi_i)$: radiancia incidente desde la dirección indicada
- $\rho_{bd}(\theta_i, \phi_i; \theta_r, \phi_r)$: BRDF (en sr^{-1})

La BRDF define la cantidad de energía reflejada en una cierta dirección, en función de la energía incidente. La función ρ_{bd} es bidireccional porque se pueden intercambiar las direcciones incidente y reflejada, y la función devolverá el mismo valor. Esto es debido a que la física de la luz es la misma hacia delante que hacia atrás.

Para obtener la BRDF se pueden utilizar varias técnicas:

- Goniorelectómetro. Es un aparato que sirve para medir la BRDF. Normalmente está compuesto por un fotómetro que se mueve alrededor de la superficie a medir, que a la vez se mueve con relación a una fuente de luz, bajo el control de un ordenador. Estos dispositivos son caros y el proceso de captura de la BRDF es lento.
- Otros dispositivos mecanizados que utilizan cámaras fotográficas en vez de fotómetros [81, 19]. Un dispositivo más simple se muestra en [57].
- Calcular la BRDF a partir de fotografías. En este caso, sólo se utilizan fotografías [85, 56].

La forma más sencilla (y barata) de obtener información sobre las características de reflectancia de las superficies sería a partir de las fotografías disponibles. Si la estimación fuera correcta, entonces la apariencia calculada al visualizar la escena local sería consistente con la apariencia medida. Por ello, se podría implementar un método iterativo de estimación de las propiedades de reflectancia de la escena local:

1. Asumir un modelo de reflectancia para la escena local (por ejemplo, difuso, difuso y especular, metálico...).
2. Elegir unos valores iniciales aproximados para el modelo de reflectancia.
3. Calcular la solución mediante iluminación global de la escena local con los parámetros actuales utilizando la iluminación capturada.
4. Comparar la apariencia de la escena local generada con su apariencia real en una o más vistas.
5. Si la escena local generada no se ajusta a la real, ajustar los parámetros en el modelo de reflectancia y volver al paso 3 .

El método anterior se podría automatizar, dejando que la computadora calculara los parámetros más adecuados para generar la escena local más ajustada. El paso 5 es el que debería, a partir de los resultados obtenidos a partir del último cálculo de iluminación y del resultado deseado, calcular la modificación de los parámetros. En [20] se presenta un método de realizar dichos ajustes, asumiendo que el modelo utilizado es únicamente difuso.

Hemos preferido dejar en manos del usuario la responsabilidad de seleccionar los parámetros oportunos para la definición del material de la escena local, tal y como muestra la Figura 3.17. Debido a la utilización del *rendering* diferencial, que se verá posteriormente, no es necesario obtener un modelo exacto de la escena local, sino que basta con una aproximación de



Figura 3.17: Diálogo para la selección de los parámetros de la escena local

las características de la misma. Por ello, la aplicación permite definir el material de la escena como **plastic** o **metal**, según los define RADIANCE (ver Figura 3.17).

La mayoría de los materiales se pueden definir como **plastic**. Dicho material tiene un color asociado con reflexión difusa, mientras que la componente especular no tiene color. Se utiliza este tipo para materiales como plástico, superficies pintadas, madera y rocas no metálicas. El tipo **metal** es exactamente igual a **plastic**, excepto que la componente especular viene modificada por el color del material. En el diálogo visto en la Figura 3.17 se pueden seleccionar todos los parámetros de ambos materiales: el color, la especularidad y el nivel de difusión de la luz sobre el material (*roughness*) [49].

Para la selección del color del material, el usuario tiene dos opciones: utilizar el cuadro de diálogo estándar de Windows para seleccionar un color, o bien calcular el color medio de la región de la imagen de fondo que ocupa la escena local. Por supuesto, el cálculo que genera este procedimiento es una aproximación burda al color real de la superficie correspondiente de la escena, ya que viene afectado por el ángulo de visión, la iluminación, los posibles brillos o texturas de la escena, etc. Se incide de nuevo en que no se necesita definir el modelo exacto de la escena local, sino que se busca una aproximación. Con respecto a los parámetros de especularidad y nivel de difusión, son competencia exclusiva del usuario. En la Tabla 3.1 se pueden ver las características que aporta al material cada uno de los dos parámetros.

	Especularidad	Nivel de difusión
Cercano a 0	Mate	Satinado
Cercano a 1	Pulido	Poco brillo

Tabla 3.1: Características del material en función del valor de los parámetros

El resultado de la definición del material de la escena local será otro fichero con formato de escena RADIANCE, como el que se muestra a continuación:

```
# Fichero generado por 'Mirage'. V1.0 (c) Francisco Abad. 2001.
# Grupo de Informática Gráfica. DSIC. U. Politécnica de Valencia

# CUIDADO! Este archivo se regenera automáticamente. No modificar

void plastic material_e_local
0
0
5 0.791613 0.611845 0.448798 0.200000 0.800000
```

3.2.6. Capturar iluminación

En esta fase del proceso se ha de construir la escena lejana a partir de las imágenes de alto rango dinámico obtenidas en el Apartado 3.2.1. El objetivo de esta fase es crear la estructura de datos adecuada para que RADIANCE pueda utilizar el mapa de entorno luminoso a la hora de calcular la visualización de los objetos sintéticos y la escena local.

Si se aplica el procedimiento explicado anteriormente a los dos conjuntos de fotografías tomadas desde ángulos distintos, se habrán obtenido dos mapas de entorno luminoso, en los que aparecerá la cámara (y quizá el fotógrafo). El objetivo de captar el mapa de entorno desde dos puntos de vista es tener la posibilidad de eliminar objetos extraños del mapa de entorno final. La técnica utilizada para ello es muy sencilla, y consiste en rotar una de las imágenes 90°, hasta hacerla coincidir con la otra. Después, únicamente queda cortar de una imagen el área que no interesa, y sustituirla por el área correspondiente en la otra imagen. De esta forma, no sólo se puede eliminar la cámara, sino una mancha en la bola, una persona que paseaba en el momento de tomar el mapa, etc.

Para realizar esta tarea, se utilizará el software ofrecido en la página de Paul Debevec [94] llamado HDR Shop, cuya ventana principal se puede ver en la Figura 3.18.

HDR Shop es una aplicación diseñada para la manipulación y el procesamiento de imágenes de radiancia de alto rango dinámico. Puede trabajar con formatos de archivos de alto rango dinámico, como por ejemplo, el formato .HDR de Radiance, el formato TIFF de píxeles de 16 *bits* o de tipo *float*, Portable Float Map (PFM) y archivos binarios sin formato. Además, puede importar y exportar formatos convencionales de imagen, incluyendo JPEG, Windows BMP y TIFF. Entre las muchas tareas que puede realizar sobre las imágenes de alto rango dinámico está la posibilidad de crear panoramas a partir de las imágenes de entrada. Dichos panoramas se generan a partir

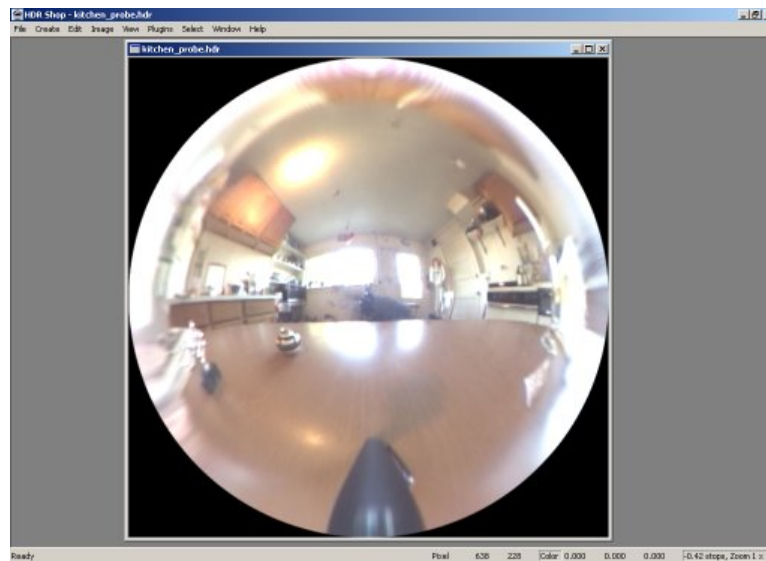


Figura 3.18: Ventana principal de HDR Shop con un mapa de radiancia

de las imágenes tomadas de una esfera reflectante, y pueden tener distintos formatos:

- primer plano de la bola: es la imagen de la bola original, reducida para ver únicamente 180° desde el centro,
- sonda de luz, o formato de mapa angular: es similar al formato de la bola original, excepto que la dimensión radial se hace corresponder linealmente con el ángulo. En la imagen original, los bordes de la esfera concentran más información que el resto, por lo tanto tienen peor calidad de muestreo,
- mapa cúbico: se construye haciendo corresponder los puntos de la esfera con las caras interiores de un cubo, para luego desplegarlo. La imagen final tiene forma de una cruz vertical, y
- mapa de latitud/longitud: hace corresponder las dimensiones horizontal y vertical con un plano. Es una proyección cilíndrica contra un plano.

A continuación se muestra paso a paso el proceso a seguir, una vez que se tienen las imágenes de alto rango dinámico separadas por 90° para construir la sonda. La herramienta que permite rotar imágenes de alto rango dinámico está accesible desde el elemento *Panoramic Transformations...*, dentro del submenú *Panorama* del menú *Image*. En la Figura 3.19 se puede ver el cuadro de diálogo correspondiente.

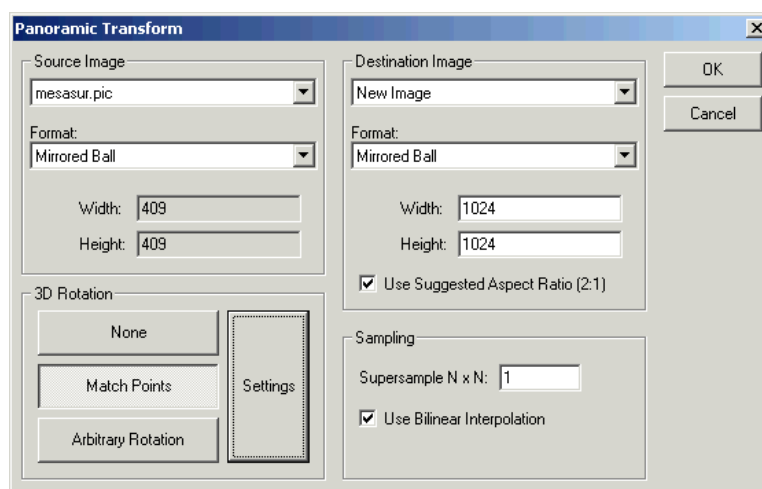


Figura 3.19: Cuadro de diálogo de las transformaciones panorámicas

El primer paso consiste en rotar una imagen hasta hacerla coincidir con la otra. Para ello, HDR Shop ofrece dos posibilidades: una rotación de un número de grados indicado por el usuario, o una rotación arbitraria definida por dos puntos de control. En el primer caso, es el usuario el que indica la magnitud de la rotación de la imagen respecto a los tres ejes. La segunda técnica se basa en los métodos de *warping* por puntos de control, en los que el usuario quiere hacer coincidir una imagen con otra, y para ello define pares de puntos equivalentes en ambas imágenes. Posteriormente, se calcula la mejor transformación que, aplicada a los puntos de la imagen origen, los lleva a los puntos de la imagen destino. La imagen resultante debe ser del mismo tamaño que la original.

En la Figura 3.20 se pueden ver dos mapas de alto rango dinámico de la misma escena, donde el punto de vista está separado por un ángulo de noventa grados. Para llevar el mapa de la izquierda a la posición definida por el mapa derecho, se utilizó la técnica de definición de dos puntos de control, mostrados en ambas imágenes. Nótese que se han buscado puntos cercanos a la zona en la que se desea mayor precisión, pero separados entre sí lo suficiente para evitar errores en la transformación.

Una vez que se transforma el mapa izquierdo mediante los puntos de control anteriores, se obtiene el mapa mostrado en la Figura 3.21.

Como se puede observar, el nuevo mapa está alineado con el segundo mapa de la Figura 3.20, pero el fotógrafo se encuentra en una posición distinta. Esto permite mezclar ambas imágenes para obtener un mapa sin el fotógrafo. Para ello, hay que definir una máscara de la zona que se desea remplazar de la imagen que se utilizará como “principal” (aquella de la que se utilizará mayor área). La imagen principal que se seleccionará es la que no

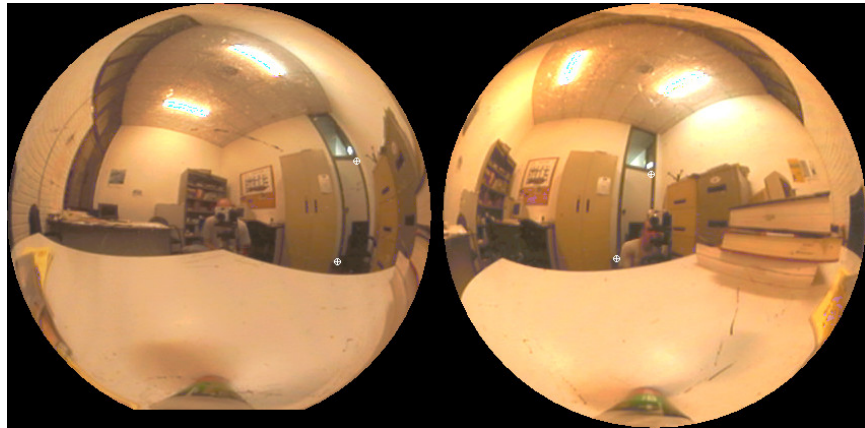


Figura 3.20: Mapas de alto rango dinámico de entrada



Figura 3.21: Mapa resultante de la transformación

se ha transformado, ya que tendrá su calidad original. Como se mencionó en un apartado anterior, la imagen principal debe ser aquella que esté más cerca del punto de vista de la imagen que se utilizará como fondo. Dentro de HDR Shop, se ha de duplicar la imagen principal (menú *Image, Duplicate*) y luego editarla en una aplicación de retoque fotográfico (menú *File, Edit in image editor*).

La edición de una imagen de alto rango dinámico con una aplicación externa permite modificar mapas de luz fácilmente. Para construir la máscara que definirá la zona a utilizar del segundo mapa, se modifica la imagen (que es una copia de la imagen principal) dejando la zona a sustituir en color blanco ($R = G = B = 255$), y el resto a negro ($R = G = B = 0$). En la Figura 3.22 se puede ver el proceso de definición de la máscara, una vez que se ha recortado el área ocupada por el fotógrafo, y a falta de establecer el

resto de la imagen al color negro.

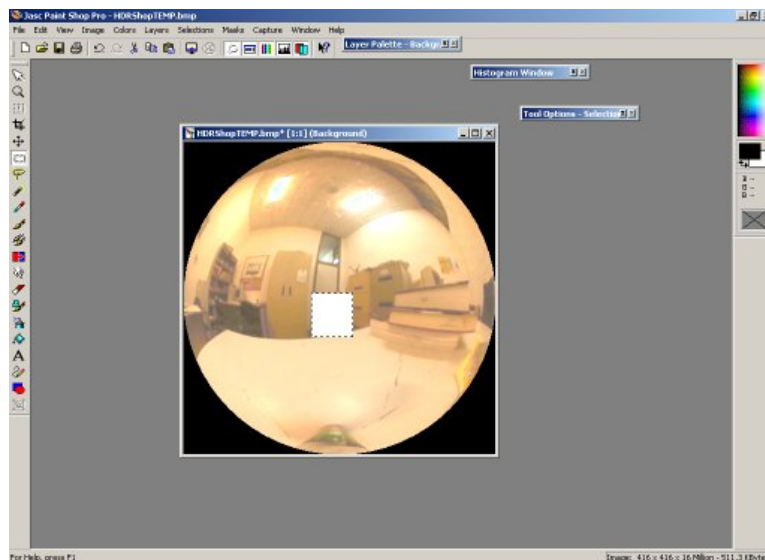


Figura 3.22: Recortando al fotógrafo

Una vez que se ha vuelto a HDR Shop, ya se puede calcular la mezcla de los dos mapas de luz mediante la herramienta *Image, Calculate...*. Dicha herramienta permite realizar cálculos aritméticos entre imágenes de alto rango dinámico, como muestra la Figura 3.23.

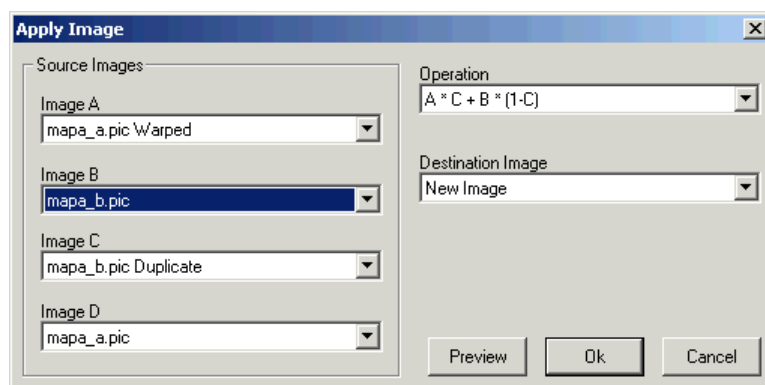


Figura 3.23: Operaciones aritméticas entre imágenes en HDR Shop

La operación a realizar es la que se muestra en la imagen: $A * C + B * (1 - C)$. Dicha operación se corresponde con la mezcla de dos imágenes (A y B), mediante un máscara (C). Si se ha definido la máscara con la zona a recortar en blanco y la zona a utilizar en negro, cada imagen se corresponde con:

- A: Imagen transformada.
- B: Imagen principal (la que no se ha transformado).
- C: Máscara.

El resultado de la operación anterior, aplicado a las imágenes de alto rango dinámico mostradas en la Figura 3.20 y en la Figura 3.21 se puede ver en la Figura 3.24.

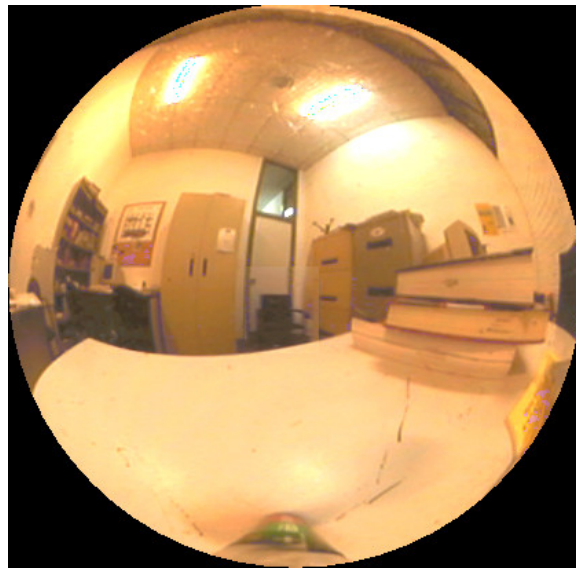


Figura 3.24: Imagen de alta radiancia sin el fotógrafo

El proceso para eliminar otros elementos de las imágenes originales (manchas, personas en mapas de luz exteriores, etc.) es análogo al visto, y sólo habría que cubrir dichas áreas con color blanco.

El último paso para obtener un mapa de entorno es generar lo que en HDR Shop se conoce como sonda de luz. Antes de realizar la conversión de formato, hay que asegurarse de que se utiliza únicamente el área circular correspondiente a la esfera, ya que si se utiliza información exterior a la esfera, aparecerán artificios en la sonda de luz. De la misma forma, si se deja de utilizar una parte de la esfera, la sonda final no tendrá toda la información disponible. Para realizar dicha selección, hay que activar la opción *Circle*, dentro del menú *Select*, submenú *Draw options*. A continuación hay que seleccionar un círculo con la porción de información relevante, y a continuación abrir el cuadro de diálogo de las transformaciones panorámicas, tal y como se ha visto anteriormente.

El conjunto de opciones que engloba el marco *Sampling* de la ventana mostrada en la Figura 3.19 permite ajustar las opciones de muestreo.

Aumentar el parámetro del cuadro de texto *Supersample* $N \times N$ ayuda a eliminar el *aliasing* del resultado final, especialmente cuando se está creando un panorama pequeño a partir de uno grande. El número de muestras es el cuadrado del número introducido (por ejemplo, $4 \times 4 = 16$ muestras por píxel). La interpolación bilineal, por otro lado, ayuda en el caso contrario, cuando se pasa de un panorama pequeño a otro grande, aunque no hay problema en dejarlo activado. En la Figura 3.25 se puede ver el resultado final.



Figura 3.25: Sonda de luz resultado

Una vez que se tiene el mapa de radiancias de alto rango dinámico, hay que prepararlo para su utilización desde RADIANCE como fuente de luz de la escena (de hecho, normalmente será la única fuente de la escena). En RADIANCE, toda fuente de luz debe estar asociada a una geometría. Por ello, hay que definir alguna forma de reprojectar los valores de radiancia recién calculados sobre la geometría aproximada de la escena lejana obtenida en la sesión fotográfica.

RADIANCE permite utilizar una primitiva llamada patrón para modificar el color de los materiales a los que se aplica. Normalmente, en gráficos por ordenador, este concepto se denomina textura, pero RADIANCE reserva esta denominación para una primitiva que permite modificar la normal de la geometría a la que se aplica. Por tanto, el patrón afecta a la reflectancia de un objeto. Hay varias formas de especificar un patrón. Por ejemplo, un patrón procedural viene dado por una fórmula que define el valor del patrón en términos de la intensidad del punto actual, la dirección del rayo,

la normal a la superficie, distancia, etc.

Los patrones también se pueden especificar a partir de imágenes. Así, hay una primitiva llamada `colorpict` que permite convertir una imagen bidimensional almacenada en un archivo en formato RADIANCE en un patrón. Para facilitar el cálculo de la correspondencia entre los píxeles de la imagen y la geometría a la que se ha de aplicar el patrón, además de para independizar el tamaño de la imagen de dicha correspondencia, se utilizan coordenadas normalizadas. De esta manera, las dimensiones de la imagen se normalizan de tal forma que la dimensión menor es siempre 1, siendo la otra dimensión el cociente entre la dimensión mayor y la menor. Por ejemplo, una imagen de 500×388 tendría unas coordenadas normalizadas definidas entre (0,0) y (1,29,1).

Para asociar la imagen a la geometría el usuario puede utilizar varios tipos de proyecciones incluidas en el sistema que mapean desde imágenes rectangulares a rectángulos, cilindros y esferas. Como la imagen que contiene la iluminación de la escena es un círculo, hay que hacer corresponder las coordenadas de la geometría de la escena lejana con un círculo. La siguiente función, definida mediante un archivo de texto tal y como se utiliza en RADIANCE, realiza el trabajo:

```
{
angmap.cal

Convierte direcciones del mundo en coordenadas de la esfera

-z hacia delante (borde exterior de la esfera)
+z hacia atrás (centro de la esfera)
+y hacia arriba (hacia la parte superior de la esfera)

}

sb_u = 0.5 + DDx * r;
sb_v = 0.5 + DDy * r;

r = 0.159154943*acos(DDz)/sqrt(DDx*DDx + DDy*DDy);

DDy = Py * norm;
DDx = Px * norm;
DDz = Pz * norm;

norm = 1/sqrt(Py*Py + Px*Px + Pz*Pz);
```

RADIANCE define una serie de variables que se pueden utilizar en los archivos de función. Dichas variables se pueden ver en el archivo `rayinit.cal`, que se encuentra en el directorio `[HOME_RADIANCE]\lib`, dentro de la instalación. A continuación se muestran las más importantes:

- Dx, Dy, Dz: dirección del rayo.
- Nx, Ny, Nz: normal a la superficie.
- Px, Py, Pz: punto de intersección.
- Rdot: coseno entre el rayo y la normal.
- `arg(0)`: número de argumentos.
- `arg(i)`: argumento i-ésimo de la llamada

El formato general de la primitiva `colorpict` es el siguiente:

```
modificador colorpict identificador
7+ rfunc gfunc bfunc <archpict> <archfunc> u v transformación
0
m A1 A2 ... Am
```

La primera línea tiene el formato común a todas las primitivas RADIANCE: el modificador (que puede ser `void`, el nombre de un material, etc.), la palabra clave que identifica a la primitiva (`colorpict`), y el identificador que designará dicha instancia de la primitiva.

A continuación aparece una serie de parámetros de tipo cadena de caracteres, precedidos por un número, que indica la cantidad de dichos parámetros. Los tres primeros parámetros de tipo cadena son tres funciones que permiten modificar el valor del color recuperado de la imagen (los valores `red`, `green`, `blue` dejan los colores sin modificar). A continuación viene el nombre del archivo que contiene la imagen (en formato RADIANCE) y luego el nombre del archivo que contiene la función que dará valor a los siguientes dos parámetros. El último parámetro es una o más transformaciones aplicados al patrón.

Las dos líneas siguientes tienen la misma estructura que la vista para los parámetros de tipo cadena. La tercera línea contiene los parámetros de tipo entero y la última contiene los de tipo real.

A continuación se muestra un ejemplo de escena lejana, cuya geometría se ha aproximado mediante una caja:

```
# Fichero generado por 'Mirage'. V1.0 (c) Francisco Abad. 2001.
# Grupo de Informática Gráfica. DSIC. U. Politécnica de Valencia

# CUIDADO! Este archivo se regenera automáticamente. No modificar

# Medidas de la escena lejana: 4.4x4.2x4

# Definición del mapeado (no está asociado a ninguna geometría)
void colorpict patron_e_lejana
11 red green blue result_light_probe_final.pic angmap.cal
sb_u sb_v -rx -90 -rz 165
```

```

0
0

# Creación de un material emisivo modificado por el mapa anterior
patron_e_lejana glow patron_e_l_glow
0
0
4 1 1 1 0

# Geometría que representa a la fuente
!genbox patron_e_l_glow caja_lejana 4.4 4.2 4 -i |
xform -t -3 -2.5 -0.8 -rz 63 -ry 180

```

Dicha escena se ha creado a partir de los datos introducidos por el usuario, y que fueron recogidos durante la sesión fotográfica. El cuadro de diálogo utilizado para la introducción de dichos datos se puede ver en la Figura 3.26.

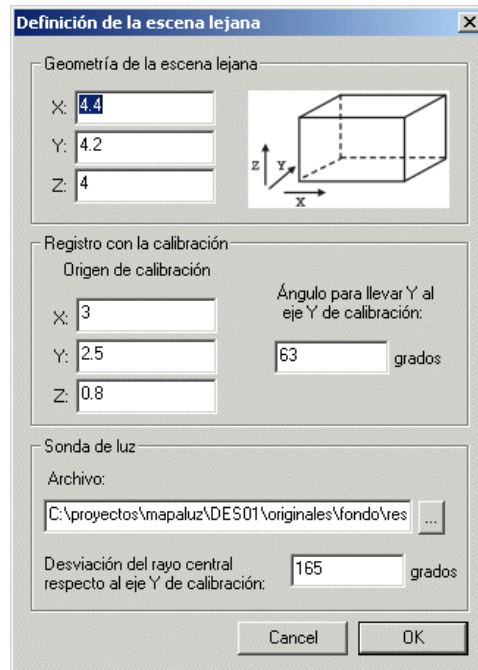


Figura 3.26: Definiendo la escena lejana

Los datos se corresponden con las medidas tomadas en la sesión fotográfica. La Figura 3.27 muestra el plano de la habitación en la que se capturaron los mapas de entorno de la Figura 3.20. En dicha figura, se debe observar la distinta orientación de los sistemas de referencia, O_R definido por RADIANCE, y O_M obtenido durante la calibración (véase la Figura 3.13). También se incluyen la posición y la dirección de la cámara que se utilizó para capturar la imagen de fondo, tal y como se extrajeron durante la calibración.

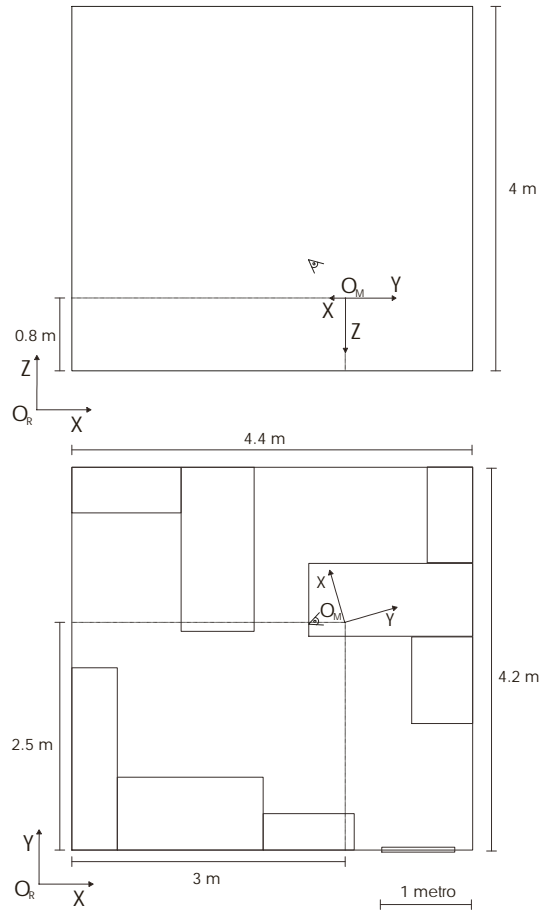


Figura 3.27: Plano de la escena utilizada en los ejemplos

A continuación se detalla cada uno de los datos que definen la escena lejana, y cómo se obtienen:

- Geometría de la escena lejana: define el tamaño de la caja que representará la escena. Se debe definir en las mismas unidades en las que se calibró la cámara (las unidades utilizadas cuando se definió el tamaño de los cuadros de la plantilla de calibración, véase el Apartado 3.2.2). En el ejemplo de la Figura 3.26, se han utilizado metros.
- Registro con la calibración. Estos datos permiten llevar el origen del mundo (O_R en la Figura 3.27) al origen que ha definido la calibración de la cámara, y que es el que se utilizará en el resto del proceso como sistema de coordenadas de referencia, O_M (véase la Figura 3.13). Dichos datos son: la posición de O_M respecto de O_R , y el ángulo ne-

cesario para alinear el eje Y de O_M con el eje Y de O_R , en el sistema de referencia O_R (los giros en el sentido contrario a las agujas del reloj son positivos). En la Figura 3.28 se puede ver dicho ángulo.

- Sonda de luz. Estos datos definen el archivo que contiene la sonda de luz y la forma de mapearla en la geometría de la escena lejana que se acaba de definir. En la Figura 3.28 se pueden distinguir los vectores V_1 y V_2 . Dichos vectores unen la posición original de la bola reflectante utilizada en la sesión fotográfica para capturar los mapas de entorno (Figura 3.9) con la posición del píxel central de dichos mapas dentro de la escena (Figura 3.20). El ángulo que hay que introducir en el cuadro de diálogo anterior es el necesario para llevar el eje Y del sistema de coordenadas O_M a la dirección del mapa utilizado (en el caso de la figura, V_1)

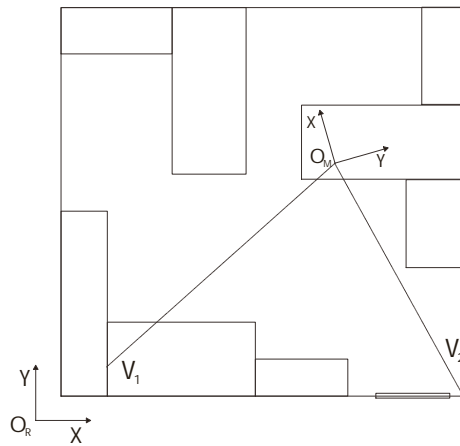


Figura 3.28: Registro de geometría y de mapa de entorno

A partir de la definición de la escena anterior, ya se podría generar una imagen de una esfera reflexiva, con la siguiente definición:

```
# Material reflexivo
void metal espejo
0
0
5 .9 .9 .9 1.0 0

# Esfera en su posición original durante la sesión fotográfica
espejo sphere bola
0
```

```
0
4 .126 .098 -0.05 .025
```

El programa RADIANCE encargado de generar imágenes a partir de descripciones de escenas es **rpict**. Dicho programa no acepta directamente los archivos ASCII con los diseños de escena, sino que primero hay que “compilarlos” en un *octree*. Para ello, se utiliza el programa **oconv**, del que a continuación se muestra un ejemplo:

```
oconv mesa.rad silla.rad silla2.rad > escena.oct
```

El ejemplo anterior crea un *octree*, agrupando las primitivas definidas en *mesa.rad*, *silla.rad* y *silla2.rad*. Dichas primitivas se organizarán en un archivo *octree* llamado *escena.oct*.

Sólo queda compilar la escena en un *octree*, y visualizarla:

```
oconv despacho.rad bola.rad > despacho.oct
```

Para seleccionar el punto de vista de la imagen final se puede utilizar la herramienta interactiva **rview**:

```
rview -vp 0 0 0 -vd 1.2 1 -.6 -vu 0 0 -1 despacho.oct
```

La Figura 3.29 muestra el proceso de selección de la vista con **rview**.

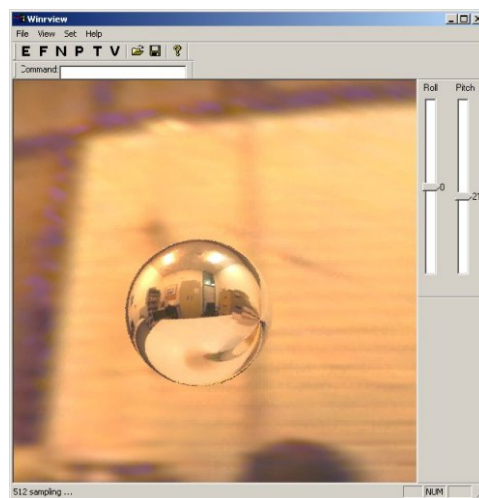


Figura 3.29: Selección del punto de vista con la herramienta **rview** (versión Windows)

Una vez seleccionado el punto de vista, se puede generar la imagen final con la herramienta **rpict**:

```
rpict -ab 2 -aa .15 -ps 8 -ar 80 -aw 40 -ad 800 -as 200 \
      -av 2 2 2 -vp 0 -1 0 -vd 0 1 0 despacho.oct > despacho.pic
```

Nótese que la orden va en una sola línea. En la Figura 3.30 se puede ver la imagen generada a partir de la escena anterior, comparada con una de las fotografías utilizadas para construir el mapa de luz que ha iluminado la esfera sintética (mostrado en la Figura 3.25).

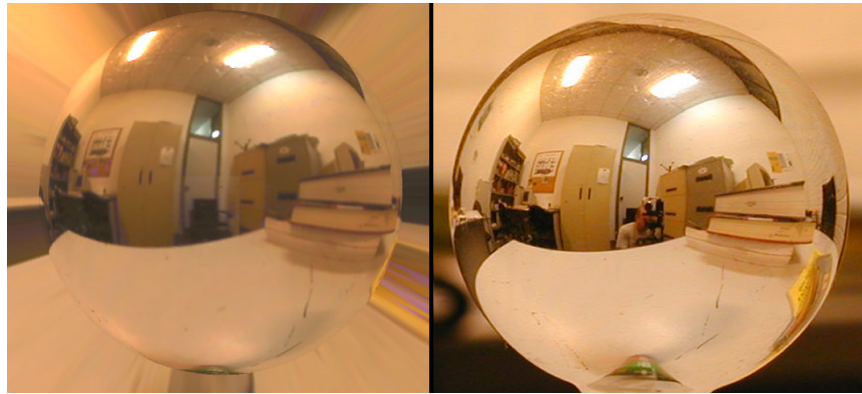


Figura 3.30: Imagen de una esfera generada con RADIANCE y fotografía desde la misma perspectiva

A continuación se muestra la importancia de seleccionar correctamente el punto de vista del mapa de entorno luminoso. Como se ha visto al principio de este apartado, el mecanismo para eliminar la cámara y el fotógrafo del mapa de entorno consiste en capturar dos mapas separados por 90° . La Figura 3.20 mostraba los dos mapas capturados, y la Figura 3.28 mostraba las dos direcciones V_1 y V_2 que definían cada uno. La Figura 3.31 muestra dos esferas virtuales, tomadas desde el mismo punto de vista. Para iluminar la esfera de la izquierda, se utilizó la sonda de luz basada en V_1 , mientras para la esfera derecha se utilizó la sonda basada en V_2 . En ambos casos de eliminaron la cámara y el fotógrafo. Como se mencionó anteriormente, hay que seleccionar siempre la dirección más cercana al punto de vista (V_1 en nuestro caso), para evitar que aparezca en la imagen final la zona ciega que existe justo a la espalda de toda sonda de luz capturada mediante una esfera reflexiva.

3.2.7. Generar máscara de los objetos

En esta tarea, hay que generar una máscara que indique en qué píxeles se encuentran los objetos sintéticos. Para ello, lo que se hace es generar una imagen con RADIANCE en la que aparezcan únicamente los objetos, con la opción de generación de un Z-buffer. El punto de vista y el tamaño de la



Figura 3.31: Esfera virtual iluminada con distintas sondas de luz

imagen final deben ser los mismos que se utilizarán para generar la imagen final. No es necesario incluir el mapa de entorno, ya que la imagen generada no se utilizará posteriormente.

Para obtener el Z-buffer de una escena a partir de su *octree* se puede utilizar la orden `rpict` del siguiente modo:

```
rpict -vf fvista.vf -x TAMX -y TAMY -z -ab 0 mascara.zbf
objetos.oct > NUL
```

donde:

- `fvista.vf`: archivo que contiene los parámetros de la vista.
- `TAMX`, `TAMY`: tamaño de la imagen que se debe generar.
- `mascara.zbf`: nombre del fichero que contendrá el Z-buffer.
- `objetos.oct`: nombre del fichero con el octree de los objetos.

De nuevo, la orden anterior se debe escribir en una sola línea. Como se puede ver, se ha redireccionado la salida al dispositivo nulo, ya que la imagen que se genera no tiene valor.

El archivo que contiene el Z-buffer no tiene formato alguno. Es una copia directa de los valores en coma flotante (`float`) en memoria de cada píxel, de izquierda a derecha y de arriba a abajo. No hay cabecera de información, y puede haber problemas si se utiliza dicho Z-buffer en otro tipo de máquina distinto al que la generó por el orden de almacenamiento en memoria de los tipos de datos de más de un byte (*little endian*: Intel, DEC Alpha RISC, *big endian*: Motorola, Sun SuperSPARC), por lo que habrá que tener este extremo en cuenta.

En el archivo Z-buffer, los píxeles que no tienen objeto se representan mediante un cero, por lo que valores mayores de cero indicarán cercanía a la cámara. Si se desea, a partir de este Z-buffer, se podrá generar una máscara de bytes. La Figura 3.32 muestra una máscara generada por el método anterior, y que se utilizará posteriormente para mostrar una escena ejemplo.

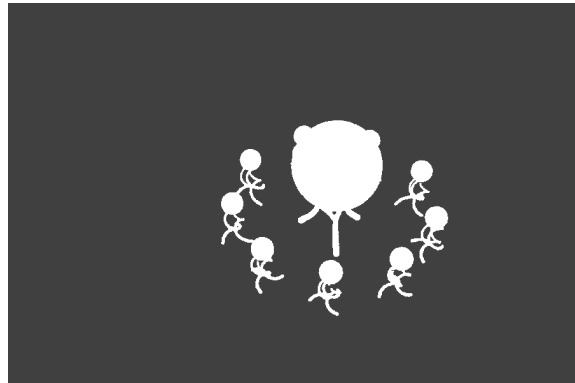


Figura 3.32: Máscara que muestra la posición de los objetos

3.2.8. Visualización de la escena local sin objetos

En el Apartado 3.2.5 se generó la descripción en formato RADIANCE de la escena local, a partir de la descripción de la geometría por parte del usuario desde la vista de la imagen de fondo. En este paso se debe invocar a RADIANCE para que visualice dicha escena, con el mapa de entorno luminoso que se ha generado anteriormente. Para ello, lo más sencillo es utilizar la herramienta `rad` de RADIANCE.

El programa `rad` se encarga de controlar todo el proceso de visualización de escenas, desde la creación del *octree* a partir de la descripción ASCII de la escena, al cálculo de la solución y posterior filtrado para la visualización. En cierta manera, es parecido al comando `make` que suele acompañar a los compiladores, para automatizar el proceso de compilación/enlace de módulos.

Cada uno de los programas que componen RADIANCE, y especialmente el comando de cálculo de la solución de una escena (`rpict`), admiten un alto número de parámetros con el fin de que el usuario pueda controlar la mayor parte del proceso. Sin embargo, esto tiene el inconveniente de la complejidad en el manejo la aplicación. Para aliviar este problema, la herramienta `rad` define una serie niveles cualitativos, que ajustan el resto de parámetros de acuerdo a unos valores preestablecidos. Por ejemplo, la variable `QUALITY` puede definirse con los valores `Low`, `Medium` o `High`, e indica la calidad de la

imagen final.

La entrada de **rad** es un archivo de configuración que contiene, entre otras cosas, las escenas RADIANCE a compilar, junto con sus dependencias, el área de la escena que se debe simular, los ajustes de la exposición, las vistas deseadas, la calidad de visualización, etc. La extensión por defecto de estos archivos de configuración es *.rif*. A continuación se muestra un ejemplo de archivo:

```
# Fichero generado por 'Mirage'. V1.0 (c) Francisco Abad. 2001.
# Grupo de Informática Gráfica. DSIC. U. Politécnica de Valencia

# CUIDADO! Este archivo se regenera automáticamente. No modificar

# ZONE. Por defecto, calcula caja de inclusión y vista exterior
# EXPOSURE. Deja a pfilt que calcule una exposición global
scene = escena_local.rad
scene= escena_lejana.rad
materials = mat_escena_local.rad
# illum = Archivos de geometría que se convertirán en fuentes
view = origen -vp 0.158111 -0.364813 -0.424822 -vd -0.050503
0.759709 0.648299 -vu -0.046920 0.646606 -0.761380 -vv 39.023998
-vh 50.591671
RESOLUTION = 1600 1200
QUALITY = Medium
PENUMBRAS = FALSE
INDIRECT = 2
PICTURE = fin_sin
# RAWFILE = para guardar la imagen sin filtrar generada por rpict
# ZFILE = archivo producido por la opción -z de rpict
# AMBFILE = para almacenar la caché de iluminación difusa
# OPTFILE = para almacenar las variables de visualización
DETAIL = Medium
VARIABILITY = Medium
render = -av 1 1 1 -ds 0.1
# oconv = opciones adicionales para el comando oconv
# mkillum = opciones adicionales para mkillum
# pfilt = opciones adicionales para pfilt
REPORT = 0.333333
```

Los archivos *.rif* no son más que una lista de variables con sus respectivos valores, que dirigirán la ejecución de **rad**. Las líneas que aparecen comentadas en el ejemplo anterior explican brevemente su función. De nuevo, para estudiar el funcionamiento exhaustivo del programa **rad**, se sugiere acudir al manual de RADIANCE. Hay dos formas principales de ejecución: una para generar la imagen de la escena, y otra para mostrar la imagen por pantalla mediante el programa **rview** ya visto. A continuación se muestran ambas órdenes:

```
rad escena.rif
rad -o win escena.rif # Muestra la imagen (versión Windows)
```

La Figura 3.33 muestra una escena local generada de la manera que se acaba de explicar. Dicha escena local es un polígono definido sobre el plano $Z = 0$, y cuyo material viene dado por la siguiente definición:

```
void plastic material_e_local
0
0
5 0.791693 0.612697 0.450312 0.141000 0.800000
```

Como se puede ver, se ha definido un material de tipo **plastic**, con una baja especularidad y una alta difusión de la luz. Sin embargo, el polígono de la Figura 3.33 muestra un texturado no homogéneo.

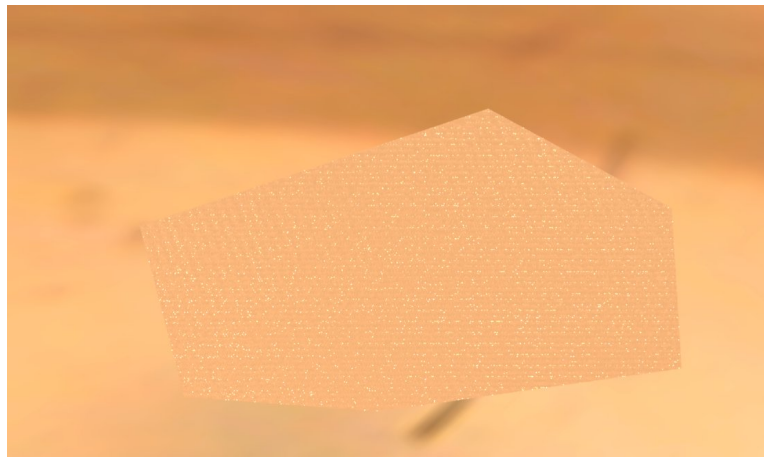


Figura 3.33: Escena local sin la definición de fuentes directas

La explicación de dicho texturado viene dada por la forma en que RADIANCE calcula la solución de iluminación de la escena. Simplificando el modo de operación de RADIANCE, para calcular la iluminación indirecta que llega a cada punto del polígono, lanza una serie de rayos con una dirección aleatoria, muestreando la escena a su alrededor. Dichos rayos se llaman rayos de entorno, y tratan de calcular la luz que llega al objeto desde sus alrededores. En el caso de la escena utilizada como ejemplo, dado que no se han definido fuentes luminosas explícitamente, la única iluminación es la que proviene de la escena lejana. Dentro de ella, la aportación principal de energía luminosa proviene del techo, de dos pares de focos fluorescentes. Por ello, si aleatoriamente se muestrea el espacio “visto” por cada punto del polígono mediante rayos de entorno, dichos rayos podrán encontrarse las

fuentes o dejarlas pasar. En función de dicha aleatoriedad pueden aparecer regiones del polígono que reciban más luz que otra cercana, por el mero hecho de que sus rayos de entorno hayan encontrado una fuente de luz intensa. Para solucionar este problema, y hacer que para calcular la iluminación de todos los puntos del polígono se tenga en cuenta las fuentes de luz del techo, se pueden utilizar las fuentes secundarias de RADIANCE.

El mecanismo esencial para describir las fuentes secundarias en RADIANCE es el tipo de material `illum`. Es un tipo de material similar a una fuente de luz normal y corriente, ya que contiene parámetros para la descripción de la radiancia emitida por una superficie, pero incluye también un nombre de material alternativo para el caso en que la vista haga visible su geometría (en RADIANCE, toda fuente debe estar asociada a una geometría). Normalmente se utiliza este mecanismo para capturar la luz promedio emitida por un objeto brillante e incluirla como luz directa, que siempre se tendrá en cuenta en el cálculo de la iluminación de los objetos [49]. Si algún rayo de entorno llegara a dicha superficie, se devolvería una energía de cero, ya que se habrá tenido en cuenta en el cálculo de la iluminación directa.

RADIANCE dispone de una herramienta llamada `mkillum`, que permite calcular la distribución de radiancia presente en una zona de la escena, y sustituirla por una geometría definida mediante el material `illum`. La entrada al programa es una escena que contiene la superficie o superficies que se desea convertir en `illum`, y la salida son las superficies convertidas y sus distribuciones de radiancia. Dichas distribuciones se calculan mandando rayos aleatorios desde los polígonos seleccionados para ser `illum`, con el fin de muestrear la radiancia incidente en dicho polígono.

La forma más sencilla de utilizar `mkillum` es mediante la herramienta `rad`, por medio de sus variables `illum` y `mkillum`. La primera indica el archivo que contiene los polígonos que se convertirán en `illum`, y la segunda las opciones que acepta el comando, y que se pueden encontrar en el manual de usuario de RADIANCE.

Para automatizar el proceso de definición de los polígonos que se convertirán en impostores (es decir, que sustituirán las porciones de la escena lejana que se han de tratar como fuentes directas), se pueden generar vistas de cada una de las paredes de la habitación, de forma que el usuario defina con el ratón los polígonos correspondientes. La forma adecuada de generar dichas vistas es mediante una perspectiva paralela, lo que facilitará luego la conversión de las coordenadas de los píxeles 2D introducidas por el usuario a polígonos 3D. En la Figura 3.34 se puede ver el proceso de definición de dichos impostores.

En la Figura 3.35 se puede ver la misma escena local iluminada con los impostores definidos. Se puede ver que la iluminación del polígono ahora es uniforme.



Figura 3.34: Definiendo impostores

3.2.9. Visualizar escena local con objetos

En este paso, la escena a visualizar estará compuesta por los objetos y por la escena local definida en un paso anterior, ambos iluminados por la luz almacenada en la escena lejana. Para que el usuario indique la posición de los objetos dentro de la escena, puede ser útil ofrecerle una herramienta parecida a la que utilizó para definir la escena local.

La herramienta `getbbox` de RADIANCE se encarga de leer la definición de la geometría de un objeto (o un conjunto de ellos) desde un archivo, y devuelve la caja de mínima inclusión que contiene a dicho objeto. A continuación se muestra un ejemplo de utilización de dicho programa:

```
C:\temp> getbbox silla.rad
      xmin      xmax      ymin      ymax      zmin      zmax
6.65917  69.4138  7.18526  70.8519 -82.4357      0
```

Dada la caja de mínima inclusión de los objetos sintéticos, se puede presentar al usuario una visualización inalámbrica de los mismos para que éste decida su posición. En última instancia, el usuario podrá acceder a los archivos ASCII que contienen la escena y modificar los parámetros de transformación de los objetos.



Figura 3.35: Escena local calculada con impostores

Es importante recordar que la cámara se encuentra en una coordenada Z negativa, por lo que la parte visible de los objetos deberá tener coordenadas Z negativas (ver Figura 3.12). La Figura 3.36 muestra el resultado de esta etapa. En dicha imagen se puede ver los objetos y la escena local definida anteriormente, además de las sombras que arrojan sobre la escena.

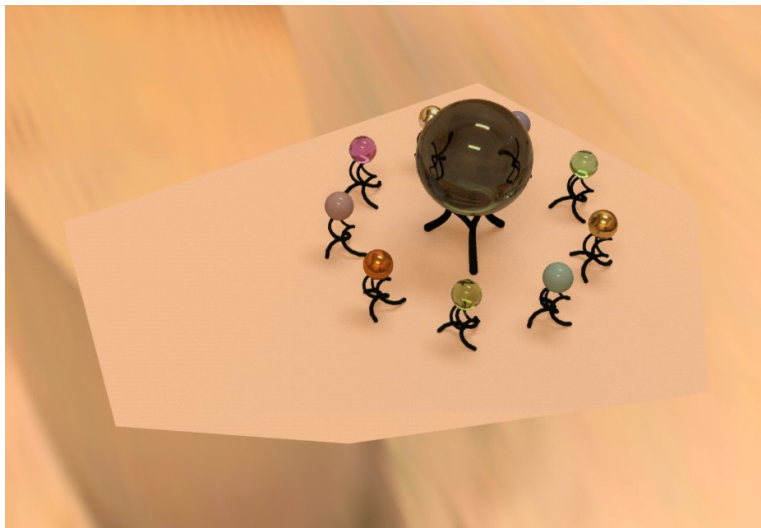


Figura 3.36: Escena local junto a los objetos sintéticos

3.2.10. *Rendering* diferencial

En este paso final, se debe generar la escena resultado a partir de los resultados obtenidos en etapas anteriores. Mediante una técnica muy semejante a la composición digital [44], se integrarán varias imágenes: la imagen de fondo rectificada, la imagen generada de los objetos junto a la escena local, la imagen generada de la escena local, y la máscara de los objetos.

Si la composición se realizara sustituyendo directamente en la fotografía de fondo la escena local y los objetos sintéticos, la transición entre la parte generada y la parte de la imagen de fondo contigua, sería evidente al espectador en la mayoría de los casos (véase la Figura 3.37). Esto es debido a que, para que este método funcionase, necesitaría que la escena local se modelase de forma muy precisa, tanto en su geometría como en las propiedades físicas (que varían espacialmente). Así, a la más mínima inconsistencia entre la parte real y la generada, se apreciaría un borde entre ellas. Para reducir este efecto, se define a continuación el proceso de *rendering* diferencial [20].



Figura 3.37: Composición directa de la imagen real y la sintética

Si se calculara la visualización de la escena local iluminada por el mapa de entorno sin incluir los objetos sintéticos, y teniendo un modelo geométrico y una BRDF perfectamente precisos, entonces se podría esperar que la apariencia de la imagen generada fuera consistente con la parte correspondiente de la imagen de fondo. Sean:

- LS_f la vista de la escena local desde la perspectiva deseada, es decir, la parte correspondiente a la escena local de la fotografía en la que se

integrarán los objetos sintéticos,

- LS_{noobj} la apariencia de la escena local sin los objetos sintéticos, tal y como ha calculado la solución de iluminación global (Figura 3.35), y
- LS_{obj} la apariencia de la escena local calculada por iluminación global con los objetos sintéticos situados en su posición final (Figura 3.36).

El error en la escena local generada (sin los objetos) es, por tanto: $Err_{el} = LS_{noobj} - LS_f$. Este error es la diferencia entre las características de reflectancia de la escena local real con la escena local generada. Se puede compensar si se calcula la imagen resultado LS_{final} como:

$$LS_{final} = LS_{obj} - Err_{el}$$

o, de forma equivalente:

$$LS_{final} = LS_f + (LS_{obj} - LS_{noobj})$$

Así, en el caso de que LS_{obj} y LS_{noobj} fueran iguales, es decir, que la inclusión de los objetos sintéticos no modificara la escena local, la imagen final sería igual a la imagen de fondo. Cuando LS_{obj} es más oscura que LS_{noobj} , entonces la luz es sustraída de la imagen de fondo formando sombras, y cuando LS_{obj} es más clara que LS_{noobj} , entonces se añade luz a la imagen de fondo para producir reflejos.

Mientras que la escena final se aprovecha de la gran calidad que ofrece un buen modelo de reflectancia de la escena local, no sufre de los errores producidos por la estimación de la función de reflectancia del material de la escena local.

Es importante resaltar que esta técnica puede producir resultados incorrectos dependiendo de la cantidad de error en la estimación de la BRDF de la escena local y de las imprecisiones en el modelo de luz de la escena lejana. De hecho, Err_{el} puede ser mayor que LS_{obj} , causando que aparezcan en la imagen final píxeles negativos. Una aproximación alternativa es compensar el error relativo en la apariencia de la escena local:

$$LS_{final} = LS_f \times \frac{LS_{obj}}{LS_{noobj}}$$

Las imprecisiones en la reflectancia de la escena local también se reflejarán en los objetos.

A continuación se muestra la secuencia de pseudocódigo que implementa el algoritmo de *rendering* diferencial.

```
resultado, fondo, con_objetos, sin_objetos : Tipo Imagen
mascara: Tipo matriz de byte
```

```

aux: Tipo entero

para y=0 hasta resultado.alto hacer
  para x=0 hasta resultado.ancho hacer
    para cada canal de color hacer
      si mascara[y][x]<>0 entonces
        resultado[y][x]=con_objetos[y][x];
      si no
        aux=fondo[y][x]+(con_objetos[y][x]-sin_objetos[y][x];
        si aux<0 entonces aux=0;
        si aux>255 entonces aux=255;
        resultado[y][x]=aux;
      fin si
    finpara
  finpara
finpara

```

Aplicando dicho algoritmo al ejemplo que se está desarrollando, se obtiene la imagen mostrada en la Figura 3.38. Las figuras 3.39 y 3.40 muestran otros ejemplos obtenidos con este mismo sistema.



Figura 3.38: Resultado final



Figura 3.39: Otro resultado

3.3. Resumen

En este capítulo se ha presentado una técnica que permite generar imágenes espectaculares con medios técnicos reducidos. Se ha presentado una implementación específica paso a paso, integrando técnicas procedentes de distintos campos.

Se ha demostrado la posibilidad de obtener un efecto complejo haciendo uso únicamente de una cámara y software disponible libremente. Técnicas como las que se han mostrado en este capítulo facilitan la generación de composiciones a usuarios no profesionales, a la vez que reducen el coste de implementación de sistemas. El principal contribuyente a la flexibilidad del sistema son las técnicas de visión, que permiten extraer la información necesaria directamente de la imagen, sin la necesidad de hardware especializado.



Figura 3.40: Otro resultado

Capítulo 4

Integración dinámica

El objetivo de este capítulo es aportar un nuevo sistema de integración dinámica de objetos sintéticos en escenas reales. Siguiendo con la filosofía de esta tesis de minimizar la utilización de hardware especializado, se utilizará únicamente una cámara de vídeo y un ordenador.

Al cambiar el escenario de la aplicación, de un entorno estático y *offline* a un entorno en tiempo real, los requisitos del sistema cambian, y cobra mayor importancia la eficiencia. Como se ha visto en el desarrollo de la tesis hasta este momento, el problema que debe resolver todo sistema de integración de imagen sintética y real es la alineación de ambos mundos. En este capítulo presentamos un sistema de localización visual que recupera posición, orientación y parámetros internos de la cámara en tiempo real. Para ello, se insertará un marcador artificial en la escena.

Aún así, se buscará realizar una integración lo más realista posible presentándose para ello varias técnicas que permiten reducir al mínimo las limitaciones inherentes a un sistema de localización visual por marcadores. Se intentará aliviar el inconveniente principal de utilizar marcadores, esto es, los propios marcadores. Para ello se ha diseñado un nuevo tipo de marcador que es poco invasivo, y que se puede borrar fácilmente de la imagen mediante un postproceso de la misma.

4.1. Trabajos relacionados

Las técnicas de calibración de cámaras más comunes se basan en capturar imágenes de un objeto cuya geometría es conocida. Las técnicas de calibración más utilizadas son las de Tsai [80, 25] y la más reciente de Zhang [89]. Dichas técnicas utilizan uno o más marcadores planos compuestos por varios cuadrados organizados de una forma determinada. Una vez tomada la fotografía, un algoritmo automático recupera las esquinas de los cuadrados mediante técnicas de procesamiento digital de imagen. Después se aplica un proceso de optimización para estimar los parámetros de la cámara. Hacen

falta un número moderado de esquinas y varias imágenes para obtener unos resultados precisos. A cambio, estas técnicas calculan un modelo de cámara muy rico (modelando las aberraciones de la lente, etc).

Las aplicaciones de realidad aumentada que se basan en localizadores ópticos han adaptado dichas técnicas, reduciendo el tamaño y la complejidad del marcador. La biblioteca de funciones para realidad aumentada Open Source más popular es ARToolKit [11][90]. Esta biblioteca ofrece una serie de funciones y un conjunto de marcadores para construir fácilmente aplicaciones completas de realidad aumentada.

ARToolKit es una biblioteca surgida en el seno del proyecto *Shared Space*. Dicho proyecto se desarrolló en el *Human Interface Technology Laboratory* de la Universidad de Washington [96] por Mark Billinghurst. El proyecto tenía como objetivo explorar varios aspectos de la realidad aumentada colaborativa. La visita de Hirokazu Kato al laboratorio en 1998 propició el desarrollo de la biblioteca, sentando las bases sobre la que se apoyarían las aplicaciones diseñadas en el resto del proyecto.

Algunas de las características presentadas por los autores de ARToolKit son el uso de una cámara para la localización de la posición y orientación de la cámara, marcadores cuadrados sencillos, capacidad de trabajar con varios marcadores mediante una técnica de reconocimiento de patrones, portabilidad y ser de código abierto. La principal desventaja que presenta es que necesita precalibrar la cámara.

En la Figura 4.1 se pueden ver varios marcadores utilizables por ARToolKit.



Figura 4.1: Algunos marcadores de ARToolKit.

Como se puede ver, cada marcador lleva una señal en su interior. ARToolKit mantiene una base de datos de señales conocidas, e implementa un sencillo algoritmo de reconocimiento de patrones que le permite identificar un marcador entre varios visibles en una imagen. Dicho algoritmo divide la zona interior del marcador en una rejilla de, por ejemplo, 16×16 posiciones. Una vez capturada la imagen de un marcador, la normaliza para obtener una vista frontal del mismo y luego la ajusta a la rejilla. Por último, compara dicha matriz con las firmas almacenadas en la biblioteca para identificar a qué símbolo pertenece el marcador encontrado.

ARToolKit no es el único sistema de localización por marcadores disponible. En [87] se puede encontrar una comparativa entre cuatro sistemas

de marcadores visuales. Dicho trabajo presenta un estudio comparativo de la precisión, velocidad, fiabilidad y usabilidad de cada uno de ellos. Los localizadores estudiados fueron los de: ARToolKit, *Institut Graphische Datenverarbeitung*, *Siemens Corporate Research* y *Hoffman marker system*. Es de reseñar que todos los sistemas estudiados utilizan marcadores cuadrados.

Aunque los marcadores de forma cuadrada son muy populares, históricamente también se han propuesto marcadores de formas circulares [59, 46, 64, 32, 53, 28]. En el trabajo de Ahn y Rauh [6] se puede encontrar, además, una muestra de varios tipos de marcadores circulares utilizados en sistemas de medida 3D. Los marcadores circulares tienen las siguientes ventajas frente a los cuadrados:

- las cónicas bajo proyección perspectiva se transforman en cónicas,
- no es necesario resolver el problema de la correspondencia entre puntos de distintas imágenes,
- presentan más robustez frente a la degradación de la imagen y a cambios en las condiciones de grabación,
- son compactas, permitiendo crear marcadores pequeños, y
- las cónicas son primitivas más ricas que los puntos, y permiten, con una sola imagen, recuperar el plano de apoyo del marcador [43].

En [29, 43, 74] se pueden encontrar las primeras referencias al uso de cónicas aplicado a visión por computador. Más tarde, la aparición de métodos automáticos para la detección y ajuste de elipses en imágenes ayudó a popularizar estos métodos. El ajuste automático de una elipse a un conjunto de píxeles en una imagen se basa principalmente en una minimización utilizando mínimos cuadrados [26] o en la transformada de Hough.

Kim y otros [47, 46] propusieron un método de calibración que utilizaba como marcador dos círculos concéntricos. Su algoritmo necesita una estimación a priori de los parámetros de la cámara para conseguir un valor inicial de la matriz intrínseca. Usan el tamaño del sensor CCD para hacer una estimación aproximada de la distancia focal. Después definen una función de coste sobre los parámetros de la calibración y la minimizan utilizando un algoritmo iterativo. Su algoritmo únicamente puede recuperar la normal al plano donde se encuentra el marcador (es decir, no recupera por completo la matriz de rotación de la cámara).

Otro método que recupera el plano de apoyo del marcador es el propuesto por Fremont y Chellali en [32]. Este método calcula la normal al plano de apoyo y un punto en el mismo expresado en el sistema de coordenadas de la cámara. También proponen un método para recuperar la matriz de rotación

completa utilizando tres planos perpendiculares entre sí. Asumen que el punto principal se encuentra en el centro de la imagen.

El algoritmo que se propone a continuación, en contraposición a los anteriores, no necesita información a priori sobre los parámetros de la cámara. Además, recupera completamente los parámetros extrínsecos de la cámara (es decir, la matriz de rotación y el vector de translación) utilizando un sencillo marcador, en tiempo real. Con una sola imagen del marcador y el radio de los círculos, nuestro sistema recupera los parámetros de la cámara que tomó la imagen. Si hay más de una imagen, entonces también permite el cálculo de la posición del punto principal.

4.2. Diseño del marcador

El marcador que utiliza nuestro sistema es una corona circular y una marca exterior impresas en una hoja de papel. Está compuesto por dos círculos concéntricos de radio r_1 y r_2 ($r_2 > r_1$), y por una marca exterior que intersecta con un círculo de radio r_3 (ver Figura 4.2).

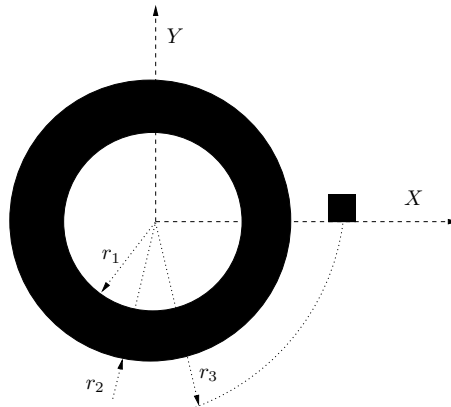


Figura 4.2: Diseño de nuestro marcador.

El marcador se ha diseñado teniendo en cuenta los siguientes requisitos:

- **completitud:** debería permitirnos recuperar los parámetros mínimos necesarios para calibrar la cámara y duplicar su configuración mediante una cámara sintética (es decir, obtener su posición, orientación y *zoom*),
- **simplicidad:** debería ser sencillo de construir y de procesar, y al mismo tiempo flexible, y
- **mínimamente invasivo:** debería ser sencillo eliminar el marcador de la escena mediante un postproceso del vídeo.

Como veremos más adelante, el algoritmo de calibración propuesto puede recuperar la posición y orientación de la cámara con una sola imagen del marcador. Este método se basa en las propiedades de los círculos bajo proyección perspectiva para recuperar la posición de la cámara con respecto al plano de apoyo del marcador.

El diseño de nuestro marcador es flexible ya que el usuario puede personalizar a su antojo el tamaño del marcador, definiendo el radio de los círculos que lo componen. Así, el localizador puede utilizarse tanto en escenas cercanas como en escenas alejadas de la cámara, ajustando el tamaño del marcador a cada situación. El usuario únicamente tiene que indicar los radios de los círculos concéntricos y el radio donde se puede encontrar la marca del eje X (véase la Figura 4.2).

La posibilidad de eliminar fácilmente el marcador a posteriori hace necesario que éste cubra un área tan pequeña dentro de la escena como sea posible. Por ello, nuestro marcador esta hueco. Históricamente, los marcadores utilizados en los sistemas de localización basados en visión son sólidos, y tapan una zona poligonal de la escena real. Nuestro marcador puede tener el tamaño de los marcadores tradicionales, pero tapar mucha menos área. Si se toman ciertas precauciones a la hora de colocar el marcador en la escena, entonces podrá ser eliminado con una simple interpolación de los píxeles de la escena real que rodean al marcador en cada uno de los fotogramas. En un apartado posterior revisaremos este tema.

4.3. Diseño del sistema de localización

En este apartado se presenta una vista completa de todo el proceso de integración dinámica de objetos sintéticos, desde el procesado inicial de la imagen, hasta la generación de la imagen final, pasando por la etapa de localización de la cámara. La Figura 4.3 muestra un diagrama de flujo con las diferentes tareas y elementos de información involucrados en el proceso. En los siguientes apartados se describen cada uno de los pasos.

Todo el proceso se ha dividido en tres etapas: preproceso, calibración y postproceso. A continuación se presentan detalladamente cada uno de los procesos en cada etapa.

4.3.1. Preproceso

El objetivo de este conjunto de pasos es obtener la posición del marcador dentro de la imagen de entrada. Aplicando métodos estándar de los campos de procesamiento digital de imagen y de visión por computador, se pueden encontrar las primitivas geométricas que componen el marcador.

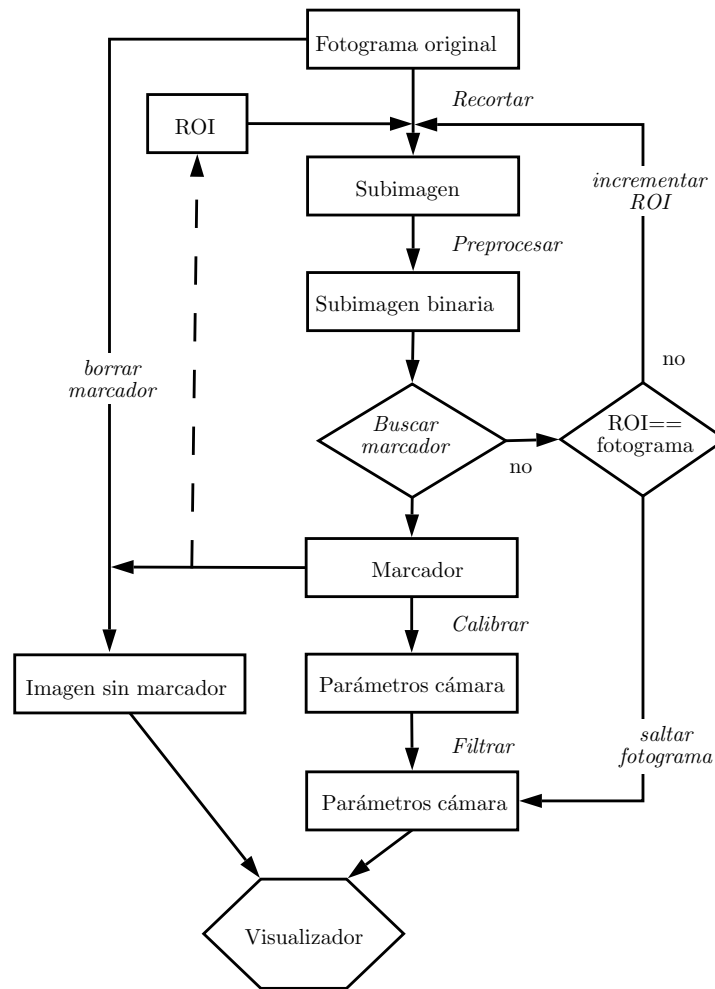


Figura 4.3: Diagrama de flujo del localizador. Los procesos se escriben en *cursiva*. Las flechas continuas representan el flujo de ejecución. Las flechas discontinuas representan información que será utilizada en el siguiente fotograma.

Recorte a una zona de interés

En aquellas aplicaciones que tratan una cantidad importante de información es crucial definir cuanto antes una zona de interés (ROI, en su denominación inglesa). Dicha zona limita el espacio de búsqueda, y permite acelerar el proceso al concentrarse en un subconjunto de los datos que, probablemente, permita resolver el problema. Al descartar información en los primeros pasos de un algoritmo se consigue disparar la eficiencia de la aplicación.

En nuestro localizador, una zona de interés es una región rectangular del fotograma original donde se va a buscar el marcador. La zona de interés inicial deberá ser el primer fotograma completo, pero en el resto de los fotogramas del vídeo el tamaño de la zona se puede reducir. Debido a la coherencia temporal, es probable que la posición del marcador en un fotograma sea cercana a la posición del marcador en el fotograma siguiente

Después de encontrar el marcador (por un proceso que será descrito más adelante), se calcula la zona de interés para el siguiente fotograma de la siguiente forma: se calcula el rectángulo de mínima inclusión del marcador y se incrementa en un porcentaje ajustable. Si el marcador no se puede encontrar dentro de la zona de interés calculada en el fotograma anterior, entonces se incrementa el tamaño de dicha zona. Este proceso se repite hasta que se encuentra el marcador, o hasta que la zona de interés coincide con toda la imagen. Si aún buscando en toda la imagen no se puede encontrar el marcador, entonces el frame se deberá descartar. Si se dispone de los resultados de la calibración en los fotogramas contiguos, entonces se pueden estimar los parámetros de calibración interpolando. Este proceso, que se realiza en el paso de filtrado, se discutirá en un apartado posterior.

Como se ha visto, una de las ventajas de la utilización de una zona de interés es la eficiencia computacional, pero tiene otra ventaja importante: simplifica la detección del marcador. Al reducir el tamaño de la zona de estudio a una zona cercana a donde se encontraba el marcador, se reduce la posibilidad de reconocer erróneamente otro objeto de la imagen como marcador.

Binarización

El objetivo principal de este paso es convertir la imagen en color original a una imagen en blanco y negro. En el campo del procesamiento de imágenes existen distintas técnicas para realizar esta tarea [15, 35], además de bibliotecas que implementan eficientemente dichos algoritmos. Nuestro sistema utiliza la biblioteca OpenCV [99] para realizar dicha tarea.

El primer paso consiste en convertir la imagen RGB obtenida de la cámara a una imagen en niveles de gris. Después, se aplica un filtro digital para eliminar ruido aleatorio. Los filtros gaussianos se comportan bien en varios

tipos de imágenes y existen diversas implementaciones disponibles. A continuación hay que binarizar la imagen en escala de grises. Existen varias posibilidades:

- aplicar un filtro de Canny [25, 15] y, posiblemente, un operador de dilatación, para obtener los contornos de los objetos en la imagen,
- utilizar un umbral fijo, o definido por el usuario, o
- utilizar un umbral adaptativo.

Cada uno de estos algoritmos tiene diversos parámetros que hay que ajustar. Dichos parámetros dependen del tipo de imagen de entrada, y de los requisitos temporales del sistema. Las aplicaciones de realidad aumentada normalmente implementan un umbral fijo que separa el marcador, típicamente negro, del resto de la imagen. Este método es el que utiliza AR-ToolKit [11]. Este tipo de binarización es muy sensible a los cambios en la iluminación de la escena. La umbralización adaptativa permite hacer frente a estos cambios en la iluminación. En [64], los autores proponen utilizar un algoritmo de mejora de contraste logarítmico para reducir el impacto de la iluminación, y para distinguir más fácilmente el marcador del fondo.

En nuestro localizador utilizamos una umbralización adaptativa, que se comporta razonablemente bien en varios tipos de escenas. Adaptando el umbral a cada situación, y teniendo en cuenta el pasado inmediato, se obtienen resultados muy buenos.

Detección del marcador

Después de la binarización debemos buscar el marcador en la imagen. La Figura 4.4 muestra un fotograma en el que se ha encontrado con éxito el marcador. En dicha figura se puede ver que los elementos geométricos que definen el marcador son dos elipses y un punto que define la dirección del eje X (mostrado en verde).

La tarea de encontrar el marcador se puede dividir en tres pasos: extracción de contornos, ajuste de elipses y búsqueda de la marca del eje X. La extracción de contornos, dada una imagen binaria, devuelve un conjunto de cadenas de píxeles contiguos. Esta operación suele estar implementada eficientemente en bibliotecas de procesamiento digital de imagen [99].

El siguiente paso consiste en decidir cuáles de las cadenas de píxeles encontradas se corresponden con el marcador. El marcador está compuesto, como se ha visto, por dos círculos concéntricos. Dichos círculos se transformarán en elipses bajo las leyes de la proyección perspectiva. Por lo tanto, el localizador debe buscar cadenas de píxeles que “parezcan” elipses. Esto es, se deben buscar cadenas cerradas que se puedan ajustar a una elipse. Antes de realizar el cálculo de dicho ajuste, se debe realizar alguna prueba de

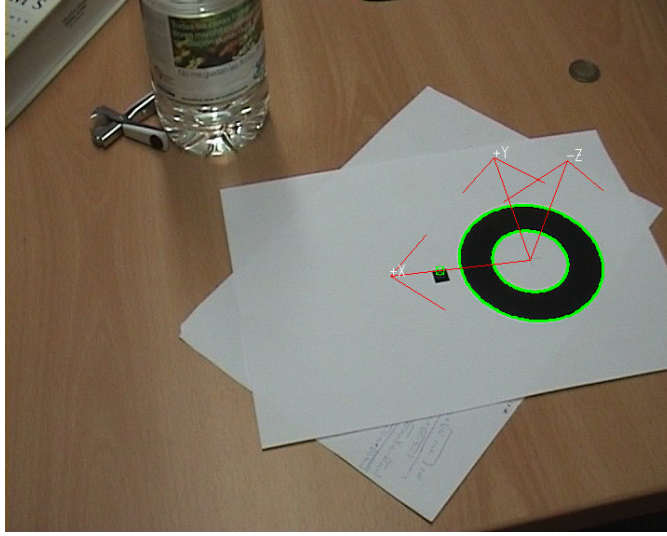


Figura 4.4: Ejemplo de detección del marcador y calibración.

rechazo para descartar aquellas cadenas que no puedan ser una elipse. Para ello, se aplican ciertos heurísticos que dan pistas acerca de si una cadena de píxeles puede ser una elipse o no, por ejemplo, si el perímetro de la cadena es muy pequeño, o si el factor de forma es mayor que 2:

$$F = \frac{p^2}{4\pi a}$$

donde F es el factor de forma, p es el perímetro, y a es el área de la cadena cerrada de píxeles. El factor de forma tiene un valor mínimo de 1 para círculos, siendo mayor que ese valor para el resto de formas geométricas. Según el estudio presentado en [6], aquellos círculos proyectado con ángulos menores a 77° tienen un factor de forma menor que 2. Aquellos círculos vistos desde un ángulo mayor y otros objetos cuyos factores de forma excedan dicho valor se descartan.

Después de dichos tests, se calcula la elipse que mejor aproxima a cada una de dichas cadenas, mediante el algoritmo de Fitzgibbon [26], obteniéndose así un conjunto de elipses. Cada elipse se define por cinco parámetros: la posición de su centro en la imagen, las longitudes de sus ejes mayor y menor, y un ángulo de rotación.

El resultado de este proceso es un conjunto de elipses detectadas en la imagen. Para decidir qué par de elipses forman parte del marcador es necesario realizar pruebas adicionales. Para ello, inicialmente las elipses se agrupan en subconjuntos que contengan aquellas de centros cercanos y ángulos parecidos (aunque esta última condición no se debería aplicar si las elipses son casi círculos).

Si se procesa una zona de interés en vez de toda la imagen, normalmente este proceso acabará generando unos cuantos grupos en los que típicamente no habrán más de dos elipses. Se debe utilizar un método de selección para decidir cuál de los grupos (o cuál de las elipses de un grupo) forman parte del marcador. Proponemos utilizar un método de selección basado en la siguiente invariante de los círculos bajo proyección perspectiva [47]: el centro proyectado de los círculos concéntricos en la imagen está en la línea definida por los centros de las elipses.

Como se ha visto, un círculo en el mundo real se proyectará normalmente a una elipse en la imagen. Sin embargo, el centro proyectado de un círculo no coincide con el centro de la elipse correspondiente en la imagen. La Figura 4.5 muestra dos círculos concéntricos que han sido proyectados a dos elipses por una cámara *pinhole*. En dicha figura se puede ver que el centro proyectado de los círculos (C'_0) está en la línea definida por los centros de las dos elipses (C_1 y C_2).

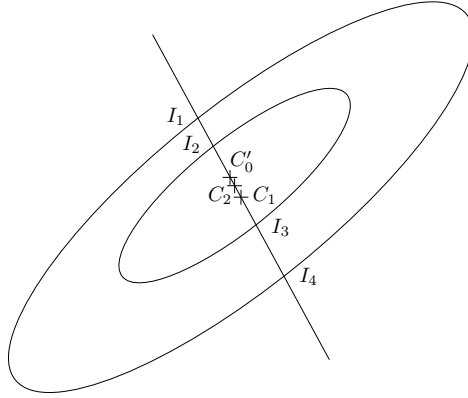


Figura 4.5: Invariante de dos círculos concéntricos bajo proyección perspectiva.

Para decidir qué par de elipses forman parte de la proyección de los dos círculos del marcador, podemos aplicar la invariante de la razón cruzada (*cross-ratio*) de cuatro puntos alineados [25], que establece que la expresión:

$$\text{Cr}_1 = \frac{d(I_1, I_3)d(I_2, I_4)}{d(I_1, I_4)d(I_2, I_3)}$$

es constante bajo cualquier proyección perspectiva, donde $d(A, B)$ es la distancia euclídea entre los puntos A y B . Si aplicamos el invariante al diseño original del marcador (es decir, dos círculos concéntricos de radios r_1 y r_2), y sustituimos las distancias por las medidas de los radios, podemos escribir:

$$\text{Cr}_1 = \frac{(r_1 + r_2)^2}{4r_1r_2} .$$

Esta invariante permite establecer un criterio de selección de elipses. El sistema asumirá que el par de elipses cuyo valor de la invariante esté más cercano a Cr_1 forma parte del marcador. Es conveniente establecer un umbral de aceptación, de tal forma que si el valor del invariante más cercano es muy diferente del teórico, se rechacen todas las elipses.

Una vez que se han encontrado las elipses que forman la parte principal del marcador, se debe buscar la posición de la marca del eje X (véase la Figura 4.4). Definimos arbitrariamente la parte positiva del eje X del sistema de coordenadas global como el vector que conecta el centro del marcador con el primer píxel de la marca del eje X del marcador, cuando se busca en el sentido contrario a las agujas del reloj.

La marca que fija el eje X se encontrará en la elipse resultante de proyectar un hipotético círculo de radio r_3 . Es decir, para encontrar la marca del eje X, se debe proyectar un círculo de radio r_3 utilizando los mismos parámetros (aún desconocidos) que proyectaron los otros dos círculos. A continuación se explicará cómo proyectar un círculo de radio arbitrario concéntrico a otros dos círculos cuyas proyecciones y radios son conocidos.

Las coordenadas de los puntos que se encuentran en un círculo C de radio r , centrado en el origen de coordenadas y en el plano $Z = 0$ deben cumplir la siguiente ecuación:

$$\tilde{X}^T C \tilde{X} = \begin{bmatrix} X & Y & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -r^2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} . \quad (4.1)$$

Como se ha visto, una matriz de proyección P (ver Apartado 2.3.2) proyecta un punto \tilde{X} del espacio 3D a un punto \tilde{x} en la imagen según:

$$\tilde{x} = P \tilde{X} . \quad (4.2)$$

Sustituyendo la ecuación 4.2 en 4.1, se obtiene que el círculo C se proyecta en una elipse Q según:

$$\lambda Q = P^{-T} C P^{-1} .$$

Si calculamos la diferencia entre la proyección de un círculo de radio $r + \alpha$ y la proyección de un círculo de radio r :

$$\lambda_1 Q_{r+\alpha} - \lambda_2 Q_r = P^{-T} (C_{r+\alpha} - C_r) P^{-1} = \alpha(\alpha + 2r) M \quad (4.3)$$

donde $M = \vec{q}^T \vec{q}$ y \vec{q} es la tercera fila de la matriz P^{-1} . Por lo tanto, podemos escribir la proyección de un círculo de radio r_3 en función de la proyección de dos círculos de radios r_1 y r_2 :

$$\begin{cases} Q_3 = Q_1 - \alpha_1(\alpha_1 + 2r_1)M \\ Q_3 = Q_2 - \alpha_2(\alpha_2 + 2r_2)M \end{cases}$$

donde $\alpha_1 = r_3 - r_1$ y $\alpha_2 = r_3 - r_2$. Utilizando estas dos ecuaciones podemos expresar Q_3 como:

$$Q_3 = kQ_2 - Q_1 \frac{r_3^2 - r_2^2}{r_3^2 - r_1^2} \quad (4.4)$$

donde k es un factor de corrección de escala de Q_1 y Q_2 . Dicho factor de escala se puede calcular aplicando la condición de rango 1 de las elipses. Como $C_{r+\alpha} - C_r$ en la Ecuación (4.3) tiene rango 1, $\lambda_1 Q_{r+\alpha} - \lambda_2 Q_r$ también debería tener rango 1 [46]. Buscar una k que haga que $Q_1 - kQ_2$ sea una matriz de rango 1 nos lleva al factor de corrección buscado. Por lo tanto, la Ecuación (4.4) nos permite proyectar un círculo de radio arbitrario dadas las proyecciones de otros dos círculos, siendo todos ellos concéntricos.

Una vez calculada la elipse que se corresponde con la proyección de un círculo de radio r_3 en la imagen, tan sólo resta recorrer dicha elipse en el sentido contrario de las agujas del reloj buscando una transición de blanco a negro (véase la Fig. 4.6). Este proceso se puede realizar, bien sobre la imagen binarizada, o bien sobre la imagen en niveles de gris. Si se opta por esta segunda posibilidad, se podría incluso obtener una precisión subpíxel, interpolando en qué posición se encuentra el salto entre dos píxeles adyacentes.

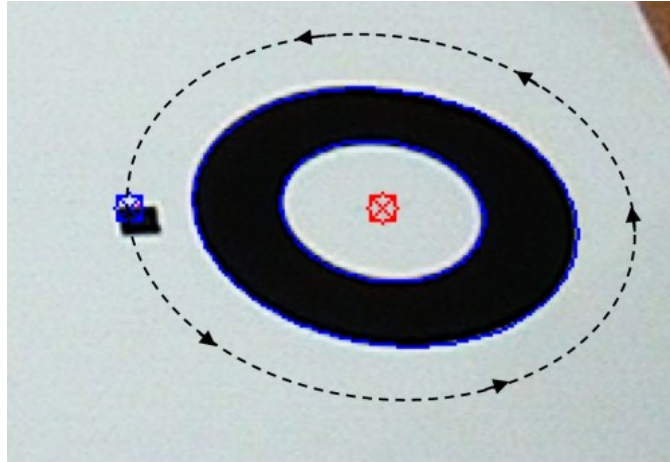


Figura 4.6: Búsqueda de la marca del eje X

En resumen, la etapa de búsqueda del marcador se puede describir en los pasos mostrados en la Figura 4.7. Primero se realiza una búsqueda de los bordes de la imagen. A continuación, se desechan aquellas cadenas de píxeles que se considera que no pueden formar parte del marcador (por no ser cerradas o no “parecerse” a una elipse). A continuación, se ajusta una elipse a cada una de las cadenas restantes y se compara todo par de elipses compatibles con la configuración teórica del marcador, por medio de un

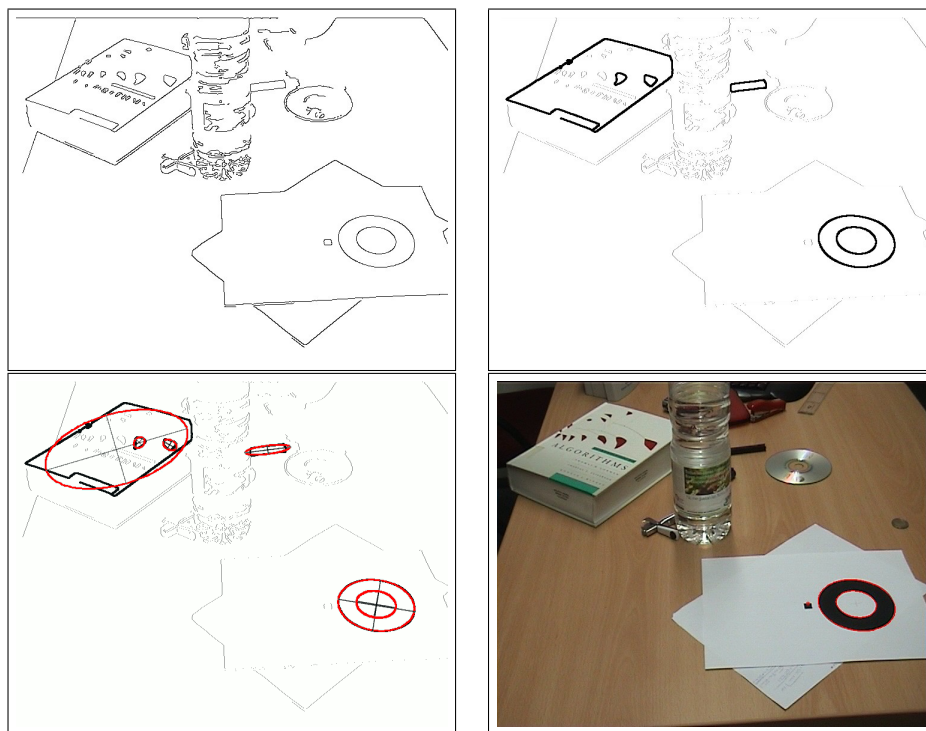


Figura 4.7: Etapas en la detección del marcador

cálculo que utiliza invariantes proyectivas. Después de encontrar las elipses del marcador, se busca la marca del eje X. El resultado final de esta etapa son las dos elipses y la marca del eje X (nótese que toda esta información viene dada en coordenadas de píxeles 2D).

4.3.2. Calibración

En los siguientes subapartados se detalla el proceso de calibración de la cámara, una vez que se ha detectado con éxito el marcador en el fotograma. El objetivo es calcular la posición, orientación y parámetros internos de la cámara real a partir del marcador. Presentaremos un algoritmo que, dada una imagen del marcador, recupera la posición y orientación de la cámara, y la distancia focal.

Es importante resaltar que el método de calibración aquí presentado no necesita información previa sobre la cámara, y que produce los resultados suficientes como para caracterizar una cámara sintética desde el primer fotograma. Dicho algoritmo de calibración únicamente necesita el marcador encontrado y el radio de los círculos que forman el marcador [4].

También conviene dejar claro que el método de calibración que se presenta no devuelve información de las posibles distorsiones del objetivo. Aunque el modelo de cámara sintética que se obtiene no es muy rico, contiene toda

Proyección de los círculos

La Figura 4.8 muestra la parametrización típica del modelo de cámara *pinhole*. Para el desarrollo del proceso de calibración de la cámara vamos a suponer el modelo de cámara más sencillo, con píxeles cuadrados, sin *skew* y eje óptico centrado en la imagen. Las dos primeras suposiciones no se alejan mucho de la realidad con las cámaras modernas de cierta calidad.

Más adelante en este apartado se mostrará una forma de calcular el punto principal. El proceso de proyección que convierte puntos 3D definidos en WCS a coordenadas de píxeles en la imagen se modela, en este caso, por las siguientes ecuaciones, ya vistas en el Apartado 2.3.3.

$$A = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

$$M = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \end{bmatrix}$$

En la Figura 4.8 se puede ver una descripción gráfica de cada uno de los parámetros de la cámara en una escena. La matriz de proyección (P) que calcula las coordenadas del píxel que corresponde a un punto 3D descrito en el WCS es:

$$P = AM = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (4.6)$$

Dado un punto \tilde{X} descrito en coordenadas homogéneas dentro de WCS, la ecuación $\lambda \tilde{x} = P \tilde{X}$ calcula sus coordenadas homogéneas en el sistema de coordenadas de la imagen. Los dos círculos del marcador se encuentran en el plano $Z_w = 0$, por lo tanto, podemos escribir:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11}X_w + p_{12}Y_w + p_{14} \\ p_{21}X_w + p_{22}Y_w + p_{24} \\ p_{31}X_w + p_{32}Y_w + p_{34} \end{bmatrix}.$$

Si suponemos que el sistema de coordenadas de la imagen está centrado en el punto principal de la imagen, entonces $u_0 = 0$ y $v_0 = 0$ en (4.5), y se puede escribir (véase [32] y el Apéndice A):

$$\begin{aligned} X_w &= \frac{\vec{x}^T (\vec{R}_2 \times \vec{T})}{\vec{x}^T \vec{R}_3} \\ Y_w &= \frac{\vec{x}^T (\vec{T} \times \vec{R}_1)}{\vec{x}^T \vec{R}_3} \end{aligned} \quad (4.7)$$

donde $\vec{x} = [u \ v \ f]^T$, y $M = [\vec{R}_1 \ \vec{R}_2 \ \vec{R}_3 \ \vec{T}]$.

En WCS, el círculo exterior de radio r_2 se describe según la siguiente fórmula:

$$C(X_w, Y_w) = X_w^2 + Y_w^2 - r_2^2 = 0 .$$

Sustituyendo (4.7) en (4.3.2), y factorizando, el círculo exterior se puede expresar en el sistema de coordenadas de la imagen según:

$$C'(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 , \quad (4.8)$$

que se corresponde con la ecuación utilizada habitualmente para describir una cónica.

Para recuperar el plano soporte del marcador a partir de la imagen de éste, es necesario calcular cuál es la posición del centro proyectado de los círculos. El subapartado siguiente presenta varios métodos para realizar dicho cálculo.

Cálculo del centro proyectado de dos círculos concéntricos

Toda cónica bajo proyección perspectiva se transforma en otra cónica [43]. Específicamente, los círculos se transforman en elipses (o en círculos) cuando se les fotografía con una cámara. Sin embargo, a no ser que el plano de proyecciones sea paralelo al círculo, el centro proyectado del círculo original no coincide con el centro de la elipse en la imagen.

El objetivo de este apartado es calcular el centro proyectado de dos círculos concéntricos de radio conocido a partir de sus proyecciones. Kim presentó en [47] un método basado en invariantes. Este método ya se ha presentado en la página 112 para calcular si un par de elipses cualquiera pueden ser el marcador. Utilizando el mismo invariante de la razón cruzada de cuatro puntos alineados, y sabiendo los radios de los círculos, es posible calcular las coordenadas del centro proyectado de los círculos.

El invariante establece que la siguiente ecuación mantiene su valor independientemente de la proyección a la que se sometan los círculos:

$$\text{Cr}_2 = \frac{d(I_1, C'_0)d(I_2, I_3)}{d(I_1, I_3)d(I_2, C'_0)} \quad (4.9)$$

donde $d(a, b)$ es la distancia euclídea desde el punto a al b , e I_1, I_2, I_3 e I_4 son los puntos de intersección entre la línea que pasa por los centros de las elipses y las elipses, como se puede ver en la Figura 4.5. C'_0 es el centro proyectado de los dos círculos concéntricos del marcador (con coordenadas x_c, y_c en la Figura 4.8). Como esta invariante se debe cumplir también en el marco del sistema global, se puede escribir:

$$\text{Cr}_2 = \frac{d(I_1, C'_0)d(I_2, I_3)}{d(I_1, I_3)d(I_2, C'_0)} = \frac{2r_2}{r_2 + r_1}$$

Una vez obtenidas las elipses y los radios de los círculos originales, es sencillo calcular C'_0 a partir de la expresión anterior.

El proceso presentado en la página 4.3.1 para proyectar círculos de radio arbitrario a partir de la proyección de dos círculos concéntricos y sus proyecciones ofrece un método alternativo para calcular el centro proyectado de los círculos concéntricos. Como el centro proyectado de un círculo estará siempre dentro de la elipse correspondiente, si proyectamos círculos de menor y menor radio, estaremos reduciendo el espacio donde se puede encontrar el centro proyectado. En el límite, un círculo de radio cero debería proyectarse en el centro proyectado de los círculos. Aplicando la Ecuación (4.4) a un círculo de radio $r_3 = 0$ se obtiene una elipse (de radio cero) cuyo centro está en el centro proyectado de los círculos concéntricos:

$$Q_3 = kQ_2 - Q_1 \frac{r_2^2}{r_1^2}$$

Las coordenadas del centro (x_c, y_c) de una elipse en forma matricial viene dada por [47]:

$$\begin{aligned} x_c &= \frac{Q_{(2,2)}Q_{(1,3)} - Q_{(1,2)}Q_{(2,3)}}{(Q_{(1,2)})^2 - Q_{(1,1)}Q_{(2,2)}} \\ y_c &= \frac{Q_{(2,3)}Q_{(1,1)} - Q_{(1,2)}Q_{(1,3)}}{(Q_{(1,2)})^2 - Q_{(1,1)}Q_{(2,2)}}. \end{aligned}$$

Por último, mencionar que en [46] se presenta otro método para calcular el centro proyectado de dos círculos concéntricos que no necesita conocer el radio de los círculos originales. Según experimentos realizados, dicho método presenta rendimientos menores a los vistos en este apartado.

Cálculo de los parámetros extrínsecos

La dirección del eje \vec{Z}_w en el sistema de coordenadas de la cámara (o, \vec{R}_3 en (4.5)) se puede calcular como sigue [29, 43]:

$$\begin{bmatrix} R_{13} \\ R_{23} \\ R_{33} \end{bmatrix} = \pm N \left\{ Q \begin{bmatrix} x_c \\ y_c \\ f \end{bmatrix} \right\} \quad (4.10)$$

donde (x_c, y_c) son las coordenadas del centro de los círculos proyectado en el sistema de coordenadas de la imagen (véase la Figura 4.8), N representa la normalización a un vector unitario, y Q es la matriz que describe la elipse, como se definió en [43]:

$$Q = \begin{bmatrix} A & B/2 & D/2f \\ B/2 & C & E/2f \\ D/2f & E/2f & F/f^2 \end{bmatrix}. \quad (4.11)$$

Los parámetros desde A hasta F son los mismos que en (4.8) y f es la distancia focal. Despejando R_{13} , R_{23} y R_{33} de la Ecuación 4.10 se obtiene [32]:

$$R_{13} = \frac{\alpha_1}{n}$$

$$R_{23} = \frac{\alpha_2}{n}$$

$$R_{33} = \frac{\alpha_3}{fn}$$

donde:

$$n = \sqrt{\alpha_1^2 + \alpha_2^2 + \frac{\alpha_3^2}{f^2}}$$

Cada parámetro de la elipse en (4.8) se puede expresar en función de f^2 , f^2/T_z^2 y un término constante sustituyendo (4.11) en (4.10) [32]. En el Apéndice A se puede ver el desarrollo, en el que se utilizan las propiedades de las matrices de rotación y las siguientes relaciones derivadas del modelo de cámara de la Figura 4.8:

$$T_x = \frac{T_z x_c}{f} \quad T_y = \frac{T_z y_c}{f} . \quad (4.12)$$

Los resultados se pueden ver en la Ecuación (4.13), donde:

$$\begin{aligned} \alpha_1 &= \frac{2Ax_c + By_c + D}{2} \\ \alpha_2 &= \frac{Bx_c + 2Cy_c + E}{2} \\ \alpha_3 &= \frac{Dx_c + Ey_c + 2F}{2} \end{aligned}$$

$$Q' = \begin{bmatrix} \alpha_1^2 & -\alpha_1^2 r_2^2 & (\alpha_1^2 + \alpha_2^2)y_c^2 + 2\alpha_2\alpha_3 y_c + \alpha_3^2 \\ 2\alpha_1\alpha_2 & -2\alpha_1\alpha_2 r_2^2 & -2(\alpha_2\alpha_3 x_c + (\alpha_1^2 + \alpha_2^2)x_c y_c + \alpha_1\alpha_3 y_c) \\ \alpha_2^2 & -\alpha_2^2 r_2^2 & (\alpha_1^2 + \alpha_2^2)x_c^2 + 2\alpha_1\alpha_3 x_c + \alpha_3^2 \\ -2\alpha_1(\alpha_1 x_c + \alpha_2 y_c) & -2\alpha_1\alpha_3 r_2^2 & -2\alpha_3(\alpha_3 x_c + \alpha_2 x_c y_c - \alpha_1 y_c^2) \\ -2\alpha_2(\alpha_1 x_c + \alpha_2 y_c) & -2\alpha_2\alpha_3 r_2^2 & -2\alpha_3(-\alpha_2 x_c^2 + \alpha_1 x_c y_c + \alpha_3 y_c) \\ (\alpha_1 x_c + \alpha_2 y_c)^2 & -\alpha_3^2 r_2^2 & \alpha_3^2(x_c^2 + y_c^2) \end{bmatrix} \quad (4.13)$$

Por lo tanto, (4.8) se puede expresar como:

$$C'(x, y) = \left(Q' \begin{bmatrix} f^2 \\ f^2/T_z^2 \\ 1 \end{bmatrix} \right)^T \begin{bmatrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{bmatrix} = 0 \quad (4.14)$$

Las incógnitas a calcular son f^2 y f^2/T_z^2 , así que reordenando (4.14) para dejar los términos constantes en el lado derecho de la expresión, queda:

$$\left(\begin{bmatrix} q_{11} & q_{21} & q_{31} & q_{41} & q_{51} & q_{61} \\ q_{12} & q_{22} & q_{32} & q_{42} & q_{52} & q_{62} \end{bmatrix} G \right)^T \begin{bmatrix} f^2 \\ \frac{f^2}{T_z^2} \end{bmatrix} = - \begin{bmatrix} q_{13} \\ q_{23} \\ q_{33} \\ q_{43} \\ q_{53} \\ q_{63} \end{bmatrix}^T G \quad (4.15)$$

donde $G = \begin{bmatrix} x^2 & xy & y^2 & x & y & 1 \end{bmatrix}^T$ y q_{ij} es el elemento (i, j) de la matriz Q' en (4.13). Dados N puntos en la elipse de la imagen, podemos construir un sistema de ecuaciones sobredeterminado $WX = B$:

$$\begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ \vdots & \vdots \\ W_{N1} & W_{N2} \end{bmatrix} \begin{bmatrix} f^2 \\ \frac{f^2}{T_z^2} \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} \quad (4.16)$$

donde W_{i1} , W_{i2} y B_i se calculan utilizando la ecuación (4.15) reemplazando (x, y) por cada una de las coordenadas (x_i, y_i) del i -ésimo punto en la elipse.

Este sistema de ecuaciones se puede resolver por mínimos cuadrados mediante la técnica de la pseudo-inversa:

$$\begin{bmatrix} f^2 \\ \frac{f^2}{T_z^2} \end{bmatrix} = (W^T W)^{-1} W^T B .$$

Resolviendo el sistema obtenemos f y T_z . Las componentes de \vec{R}_3 se pueden calcular substituyendo f en (4.10). T_x y T_y se pueden calcular desde (4.12).

Siguiendo los pasos descritos en este apartado calculamos la normal al plano que contiene los círculos (\vec{R}_3) y la posición del origen del sistema de coordenadas global (WCS) respecto al sistema de coordenadas de la cámara (\vec{T}) (véase la Figura 4.8). Fremont propuso en [32] un patrón de calibración compuesto por tres planos ortogonales para calcular los otros dos ejes (\vec{X}_w y \vec{Y}_w , o \vec{R}_1 y \vec{R}_2 , respectivamente). En vez de los tres planos, nosotros proponemos una modificación más sencilla del marcador que también permite

recuperar completamente la matriz de rotación. Proponemos el uso de una simple marca en el exterior de los círculos que defina la dirección del vector \vec{X}_w , una idea utilizada antes en el campo de detección de marcadores [53].

En el proceso de detección del marcador en la imagen, se han calculado las coordenadas del píxel que define la dirección del eje X_w (el punto $(r_3, 0, 0)$ en WCS, y (x_m, y_m) en el sistema de coordenadas de la imagen de la Figura 4.8). Dadas las coordenadas del píxel de la marca del eje \vec{X}_w , la podemos reproyectar en el plano de soporte de los círculos concéntricos. Dicho plano está definido completamente, ya que conocemos tanto su normal (\vec{R}_3) como un punto en el mismo (\vec{T}):

$$R_{13}x + R_{23}y + R_{33}z + D = 0 ,$$

donde $D = -\vec{R}_3^T \vec{T}$. Sea la posición de la marca del eje X ($r_3, 0, 0$) en WCS, (x_m, y_m) en el sistema de coordenadas de la imagen, y \vec{X}_m en el sistema de coordenadas de la cámara (véase la Figura 4.8). Entonces,

$$\vec{X}_m = \mu \begin{bmatrix} x_m \\ y_m \\ f \end{bmatrix} ,$$

donde:

$$\mu = \frac{-D}{R_{13}x_m + R_{23}y_m + R_{33}f} .$$

Dadas las coordenadas 3D de la marca del eje \vec{X}_w en el sistema de coordenadas de la cámara y las coordenadas 3D del origen del WCS, dadas también en el sistema de coordenadas de la cámara, el eje \vec{X}_w (o \vec{R}_1) está definido por:

$$X_w = N \left\{ \begin{bmatrix} X_m \\ Y_m \\ Z_m \end{bmatrix} - \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \right\}$$

donde N es un operador de normalización. En un sistema de coordenadas dextrógiro, $\vec{Y}_w = \vec{Z}_w \times \vec{X}_w$, o $\vec{R}_2 = \vec{R}_3 \times \vec{R}_1$.

Siguiendo estos pasos se puede calcular la posición, orientación y la distancia focal de la cámara. Dichos parámetros permiten definir la cámara virtual de un motor de visualización gráfica para generar objetos que estén alineados con la escena real, con respecto al marcador. Es importante recordar que como cada fotograma se calibra independientemente de los demás, nuestro sistema permite el cambio del *zoom* de la cámara mientras que se captura la película. En un apartado posterior se presentará un experimento en el que se demuestra este punto.

Cálculo del punto principal

Hasta ahora se ha supuesto que el eje óptico de la cámara estaba centrado perfectamente en la imagen (es decir, el punto principal coincidía con el centro de la imagen). En esta sección eliminamos esta suposición y calculamos el punto principal utilizando resultados de apartados anteriores.

Debido al error en la estimación del punto principal, si se reprojeta el círculo original utilizando los parámetros calculados en los apartados anteriores, en general no se obtendrá la elipse extraída de la imagen. Este desajuste es proporcional al error incurrido en la estimación del punto principal. Minimizando dicho error se puede calcular la posición del punto principal.

Una vez obtenidos los parámetros que definen la proyección (es decir, la matriz P), podemos reprojectar el círculo de radio r_2 en la imagen según la ecuación:

$$\lambda Q = P^{-T} C P^{-1} ,$$

donde:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -r_2^2 \end{bmatrix} .$$

Minimizando la diferencia entre la elipse extraída de la imagen y la elipse calculada por el método anterior podemos calcular el punto principal. Una forma de medir el error de reproyección es calculando la distancia entre el centro de ambas elipses. Un método alternativo es calcular la diferencia entre los ángulos de los ejes principales de ambas elipses. El algoritmo se podría resumir en los siguientes pasos:

1. Inicializar a cierto valor el punto principal calculado (por ejemplo, al centro de la imagen).
2. Calibrar la cámara.
3. Reprojectar el círculo original (de radio r_2) usando los parámetros obtenidos en el paso anterior.
4. Calcular el error de reproyección y actualizar en consecuencia el punto principal.

Se pueden aplicar métodos de optimización como el de Levenberg-Marquardt [61] (implementado en el paquete Minpack [101]) para calcular eficientemente la posición 2D del punto principal que minimiza el error de reproyección. Dichos métodos minimizan una función proporcionada por el usuario, que en nuestro caso es una función que toma como entrada la posición del punto principal y devuelve como salida el error de reproyección.

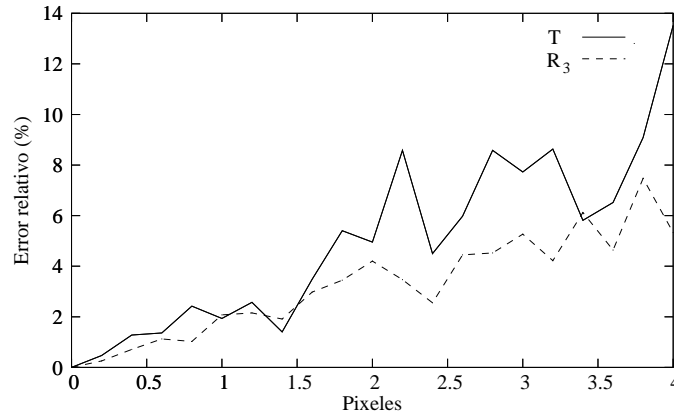


Figura 4.9: Errores relativos en la estimación de \vec{T} y \vec{R}_3 (\vec{Z}_w).

En el caso de estar procesando un flujo de vídeo, el punto principal se puede calcular una vez con una serie de fotogramas, y mantenerse fijo para el resto. Si la estimación del punto principal es correcta, éste valor será válido mientras que la configuración del zoom de la cámara no cambie.

Validación del método

En este apartado presentamos algunos resultados que comprueban la robustez de la calibración. Hemos utilizado tanto datos sintéticos como reales para comprobar el comportamiento del algoritmo frente al ruido.

Experimentos con datos sintéticos

Hemos realizado varias pruebas para medir la robustez del proceso de calibración en presencia de errores en los datos de entrada. Un buen algoritmo de calibración debería ser estable, de tal forma que pequeños errores en los datos de entrada deberían producir pequeños errores en los datos de salida.

Para comprobar la robustez de nuestro algoritmo, hemos proyectado dos círculos concéntricos utilizando una configuración de cámara sintética conocida. Después, se han generado dos conjuntos de puntos sobre las elipses correspondientes y se han perturbado dichos puntos añadiendo ruido aleatorio a sus componentes. Después, se han ajustado sendas elipses a cada uno de dichos conjuntos utilizando el algoritmo de Fitzgibbon [26]. Por último, se ha aplicado el método de calibración a ambas elipses. La Figura 4.9 muestra los errores que el ruido añadido induce sobre la normal al plano soporte (\vec{R}_3) y sobre el vector de translación (\vec{T}). Nótese que el error producido es relativamente pequeño.

Las matrices utilizadas para realizar el experimento han sido:

$$A = \begin{bmatrix} 859,13 & 0 & 360 \\ 0 & 859,13 & 288 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} -0,970 & -0,237 & 0,054 & 6,862 \\ 0,198 & -0,643 & 0,74 & 10,327 \\ -0,140 & 0,729 & 0,67 & 59,551 \end{bmatrix}.$$

Hemos encontrado que el algoritmo es muy robusto frente a errores sistemáticos, es decir, cuando ambas elipses se ven afectadas por el mismo error (por ejemplo, por un eje óptico no alineado). Por otro lado, si el nivel de ruido modifica las elipses más allá de un cierto límite, la precisión de los resultados decrece dramáticamente.

Por lo tanto es muy importante ajustar las elipses lo mejor posible, y asegurarse de haberlas encontrado correctamente antes de empezar el proceso de calibración. Para ajustar las elipses a los píxeles de entrada usamos el algoritmo de Fitzgibbon, que alcanza una precisión de subpíxel.

Experimentos con datos reales

Para validar la calibración realizada sobre imágenes reales, hemos aplicado el algoritmo a varias imágenes tomadas con una cámara de vídeo digital SONY DCR-PC3E.

En la Figura 4.10 se pueden ver varias imágenes procesadas. Después de extraer automáticamente las elipses de la imagen, y siguiendo los pasos descritos en los apartados precedentes, se obtuvieron los parámetros de la cámara. Utilizando dichos parámetros, se han dibujado los ejes del sistema de coordenadas global sobre la imagen. Además, también se han reproyectado los dos círculos del marcador utilizando los mismos parámetros. El marcador de la imagen tiene las siguientes características: $r_1 = 2,6$ cm, $r_2 = 5$ cm y $r_3 = 6,5$ cm.

4.3.3. Postproceso

Por postproceso entendemos aquellas operaciones realizadas una vez que se ha calibrado la cámara, y se han obtenido (*o no*) los parámetros que definen su relación con respecto al sistema de coordenadas global definido por el marcador. Como se verá en los siguientes apartados, dicho postproceso está orientado a la mejora o reconstrucción de la calibración, al borrado del marcador, y a la inserción de los objetos sintéticos.

Filtrado

Debido al ruido en las imágenes de entrada, cambios de iluminación, sombras o movimientos rápidos de la cámara pueden aparecer errores en la

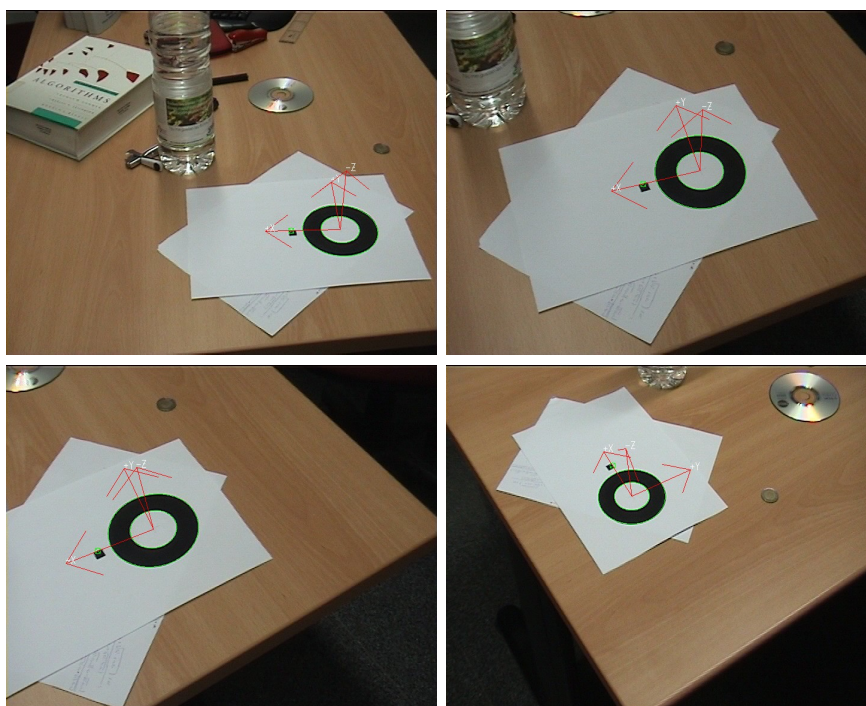


Figura 4.10: Reproyección del marcador y del sistema de coordenadas global

detección y ajuste de las elipses del marcador. Cuando se integra un flujo de vídeo real y una animación generada por computador, dichos errores inducen un temblor en los objetos sintéticos. Dichos movimientos erráticos reducen el realismo en el resultado de la integración. Para reducir este temblequeo se debe aplicar un filtro a los resultados de la calibración.

En principio, hay dos posibles escenarios de aplicación de nuestro sistema: localización en tiempo real, y localización *offline*. En éste último caso, el proceso de filtrado se puede realizar en dos pasos. Se puede obtener una mejor calidad de filtrado ya que existe información pasada y futura sobre la calibración para cada fotograma.

Por otro lado, las aplicaciones de localización en tiempo real sólo disponen de la información pasada para filtrar los resultados del fotograma actual. En esta clase de sistemas se aplica comúnmente el filtro de Kalman [83, 25]. El filtro de Kalman es un estimador estocástico que puede predecir el estado de un sistema basándose en mediciones previas de la salida del sistema. Es muy utilizado en las aplicaciones de localización, donde, a partir de las posiciones previas de un objetivo se debe predecir su posición futura. El filtro de Kalman puede integrar a su vez información proveniente de diversas fuentes (por ejemplo, un localizador óptico y otro inercial [64]).

El parámetro que más contribuye al temblequeo de los objetos generados

por computador es el vector R_3 (es decir, la normal al marcador, véase la Figura 4.8). Dicho parámetro se debe filtrar cuidadosamente para obtener buenos resultados.

El filtro *offline* implementado en nuestro sistema es un filtro de media ponderado, que trabaja como se explica a continuación. En primer lugar, se eliminan aquellas muestras que más se apartan de las demás para favorecer la concentración de las mismas. A continuación, se calcula el valor medio de las muestras ponderando cada una según un parámetro. Dicho parámetro viene calculado desde la calibración, e indica cuánta confianza se tiene de la calibración realizada. En el apartado dedicado a la detección del marcador de la página 112 ya se definió dicha medida, en función del invariante de la razón cruzada de cuatro puntos alineados.

Se podría obtener otra posible medida de la calidad del proceso de calibración comparando la posición de un punto en la imagen con su posición teórica obtenida tras la calibración. Por ejemplo, se podría calcular la proyección del punto $(r_3, 0, 0)$ utilizando la cámara sintética. La posición en la imagen de ese punto se recuperó directamente de la imagen en el proceso de detección del marcador. Por lo tanto, otra medida de calidad de la calibración vendría dada por la distancia euclídea entre esos dos puntos.

Otra ventaja de la utilización de un filtro es que se pueden interpolar los datos de la calibración desde los datos pasados (y futuros, si están disponibles). Esto permite al sistema ser más robusto frente a condiciones de ruido intenso u oclusiones temporales del marcador.

Borrado del marcador

A menudo es deseable poder eliminar de las imágenes el marcador utilizado en la calibración para integrar objetos sintéticos en escenas reales. Si los objetos sintéticos no cubren completamente el marcador, es necesario borrarlo de la escena. Existen varios métodos para eliminar el marcador del flujo de vídeo. Cada método tiene sus ventajas e inconvenientes, y por tanto es el usuario el que debe determinar qué método es el que más le conviene según sus necesidades.

Debido al diseño del marcador, es relativamente sencillo borrarlo digitalmente. Este borrado consiste en “rellenar” el hueco dejado por el marcador en la imagen. En gráficos por computador, este problema es conocido como *inpainting* [10]. El *inpainting* está muy relacionado con el problema de la restauración de imágenes en el campo de visión por computador.

Para alcanzar una calidad aceptable en el proceso de borrado del marcador, se debe planificar con cuidado su posición dentro de la escena real. Si el marcador se coloca en una superficie cuyo color sea uniforme, el marcador se podrá borrar simplemente interpolando linealmente entre los píxeles contiguos y suavizar el resultado. Si, por el contrario, el marcador está sobre una superficie texturada, entonces habrá que utilizar un proceso más refinado (y

más costoso computacionalmente), como por ejemplo, *inpainting*.

La Figura 4.11 muestra varios ejemplos de fotogramas en los que se ha borrado el marcador. Se ha utilizado interpolación bilineal entre los píxeles que rodean al marcador. Este proceso es tan eficiente que apenas añade sobrecarga a la aplicación. En dicha figura se muestra la importancia de seleccionar adecuadamente la posición del marcador: las superficies que no contienen texturas producen muy buenos resultados, mientras que en las superficies texturadas el borrado del marcador se hace evidente.

Otro modo de borrar digitalmente el marcador de la escena, si este se apoya en una superficie plana es tomando una fotografía perpendicularmente a dicho plano, sin el marcador. Después, durante el proceso de visualización, se puede reemplazar la posición ocupada por el marcador por un polígono texturado con la fotografía tomada a priori. Posteriormente se deberá aplicar un filtro de suavizado para obtener una integración de mayor calidad. Algunos problemas de esta aproximación son el posible cambio local de la iluminación debido a reflejos cuando la cámara se mueve, cambios en la posición o intensidad de las luces, etc. En estos casos, se podría aplicar una solución de texturado dependiente de la vista [23].

Integración de los objetos sintéticos

En esta etapa se realiza la integración propiamente dicha de los objetos sintéticos en el flujo de vídeo real. Este proceso es dependiente del motor de *rendering* utilizado. En nuestra implementación hemos utilizado OpenGL [69][103] para la visualización de los objetos sintéticos.

El primer paso para integrar los objetos sintéticos es cargar el fotograma (del que posiblemente se ha eliminado el marcador) en el *framebuffer* del motor. La función OpenGL que se encarga de ello es `glDrawPixels`. Dicha función recibe los bytes que conforman la imagen y el formato utilizado para su almacenamiento.

El siguiente paso consistiría en especificar las características de la cámara sintética. Dichas características se derivan de la calibración. La calibración, como hemos visto, devuelve una matriz intrínseca y otra extrínseca. La matriz intrínseca establece las características propias de la cámara, y la extrínseca su situación y orientación dentro del mundo.

La función OpenGL que usamos para especificar los parámetros de la cámara es `gluPerspective`, que necesita la medida del ángulo de visión vertical, la razón de aspecto, y las distancias de los planos de recorte. Los dos primeros parámetros se pueden extraer de la matriz intrínseca obtenida durante la calibración. La situación de los planos de recorte es dependiente de la aplicación. Dada una matriz como la siguiente:

$$A = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

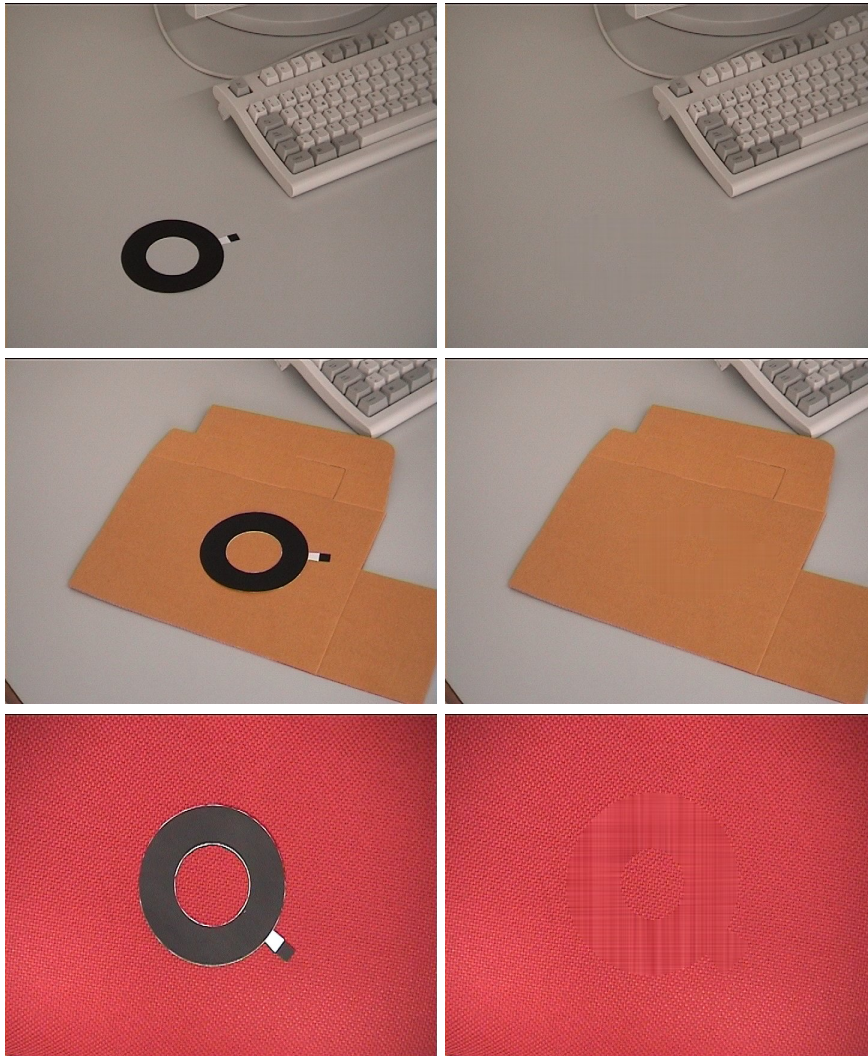


Figura 4.11: Borrando el marcador: la columna izquierda muestra los fotogramas originales, y la derecha los mismos fotogramas, después de borrar el marcador.

donde f_x y f_y es la distancia focal, medida en número de píxeles, teniendo en cuenta la anchura y la altura del píxel, respectivamente (si los píxeles son cuadrados, entonces $f_x = f_y$).

Como se puede derivar a partir de la Figura 4.12:

$$\alpha = \arctan \frac{\text{ancho}/2}{f_x}$$

$$\beta = \arctan \frac{\text{alto}/2}{f_y},$$

estando especificados el ancho y el alto de la imagen en píxeles. β es la mitad del ángulo que indica la apertura del campo de visión vertical.

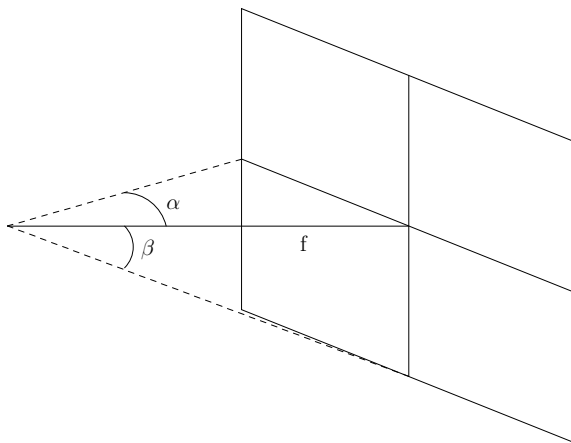


Figura 4.12: Parámetros intrínsecos de la cámara.

A partir de la matriz extrínseca se puede calcular la posición y orientación de la cámara, como ya se vio en el apartado correspondiente de la página 72. La función OpenGL que permite situar la cámara dentro del sistema de referencia es `gluLookAt`, y necesita la posición de la cámara, la dirección de la vista y el vector *up*.

4.4. Experimentos

En este apartado presentamos algunos experimentos realizados con nuestro sistema sobre vídeo real. Se han capturado varios segmentos de vídeo con una cámara de vídeo digital no profesional.

La Figura 4.13 muestra un fotograma del vídeo original y otros tres en los que se ha insertado un objeto sintético. Como se puede ver, el *zoom* de la cámara varía y el tamaño de los objetos sintéticos se adapta siguiendo sus evoluciones.

La Figura 4.14 muestra varios fotogramas de un vídeo que ilustra un sistema de exploración de objetos sintéticos. El usuario puede “mover” el objeto sintético por medio del marcador. Este es un ejemplo de aplicación de nuestro localizador a la implementación de un interfaz de usuario avanzado.

La Figura 4.15 muestra varios fotogramas de un vídeo donde se ha integrado una escena sintética animada. Mientras que la cámara real se mueve alrededor de la escena, los objetos sintéticos se mantienen alineados a esta, manteniendo el realismo de la integración. Nótese que se ha eliminado el marcador de la escena digitalmente obteniendo un resultado satisfactorio. También se ha simulado una sombra del objeto sintético sobre la escena real, mediante un polígono (no hay cálculo de sombras, ya que se ha utilizado OpenGL).

4.5. Sistema de calibración alternativo

Aparte del algoritmo de calibración que se ha descrito en el apartado 4.3.2, se ha probado otra opción, propuesta por un revisor anónimo. El algoritmo de calibración alternativo utiliza el mismo marcador, pero utiliza otra aproximación. Se basa en el cálculo de la homografía planar mediante una transformación lineal directa (*direct linear transformation*). Dicho

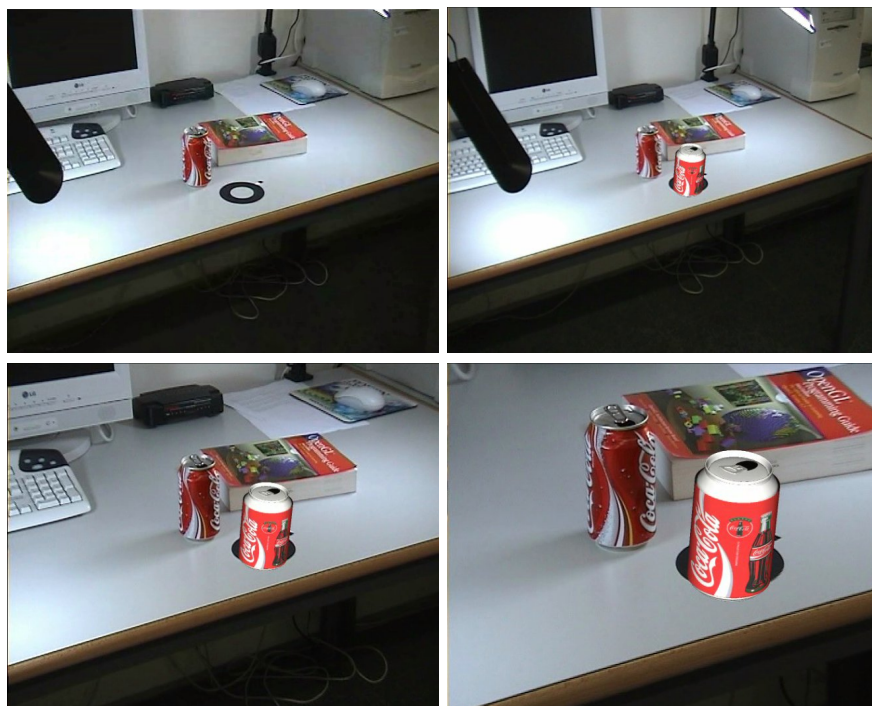


Figura 4.13: Variación del zoom durante la grabación.

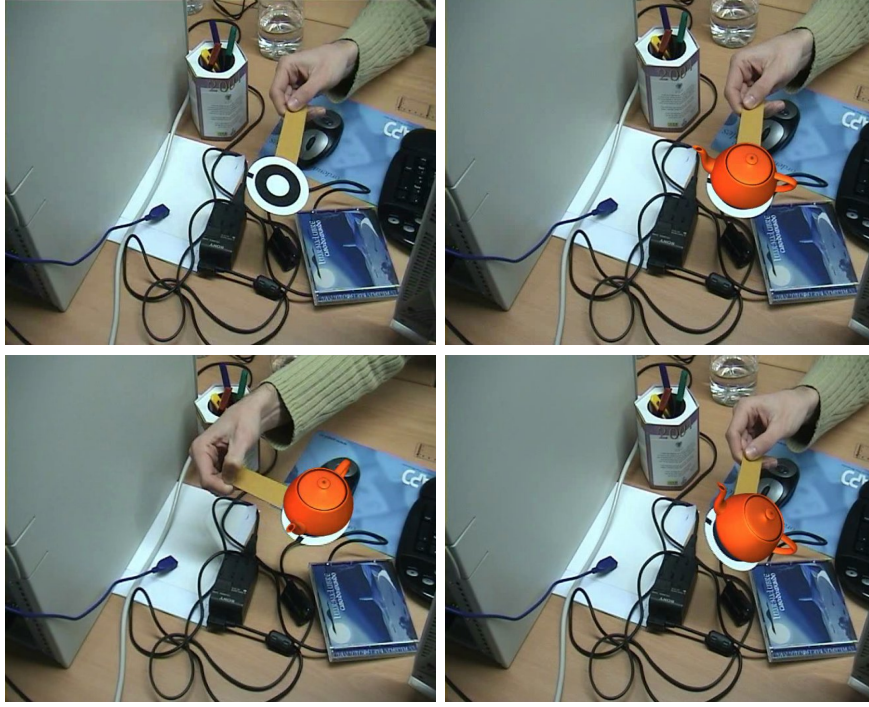


Figura 4.14: Explorando objetos virtuales mediante un marcador

cálculo se basa en establecer la correspondencia entre cuatro puntos en la imagen y cuatro puntos definidos en el sistema de coordenadas global.

Como se ha visto en el Apartado 2.3.3, la cámara proyecta puntos del plano $Z = 0$ al plano de la imagen según la siguiente relación:

$$s\tilde{\mathbf{m}} = \mathbf{H}\tilde{\mathbf{M}} \quad \text{con} \quad \mathbf{H} = A \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}. \quad (4.17)$$

H está definida hasta un factor de escala, y por lo tanto tiene ocho grados de libertad [38]. Como cada punto 2D define dos grados de libertad, hacen falta establecer cuatro correspondencias para definir completamente H .

El revisor propuso obtener dichos cuatro puntos del marcador:

- el centro de los círculos (que es, a su vez, el origen del sistema de coordenadas global),
- la marca del eje X,
- y los puntos circulares.

4.5.1. Puntos circulares. Aplicaciones

Proyectivamente, dos círculos concéntricos únicamente intersectan en los puntos circulares, es decir, en $I = \begin{bmatrix} 1 & i & 0 & 0 \end{bmatrix}^T$ y $J = \begin{bmatrix} 1 & -i & 0 & 0 \end{bmatrix}^T$



Figura 4.15: Remplazando el marcador con una escena sintética animada

(todos los círculos intersectan en los puntos circulares [63]). Las proyecciones de dos círculos concéntricos intersectarán en la proyección de los puntos circulares (véase la Figura 4.16). Además dichos puntos están en la línea del horizonte del plano l_∞ . Esta línea es la polar al centro proyectado de los círculos concéntricos (C' en la Figura 4.16) [59, 52].

Dadas las dos elipses E_1 y E_2 en la imagen, se puede calcular su intersección. Todo par de cónicas intersecta en cuatro puntos, según la casuística mostrada en la Figura 4.17 [74]. De izquierda a derecha:

- cuatro intersecciones reales,
- dos reales y una real doble,
- dos reales y un par complejo conjugado, y
- un par complejo conjugado doble.

El caso que nos interesa es el último. Dos círculos concéntricos únicamente intersectan en los puntos circulares, por lo que aparecerá un par complejo conjugado, de multiplicidad dos. Esto en teoría, ya que en la práctica, el error en la estimación de las elipses producirá dos pares de intersecciones complejas conjugadas.

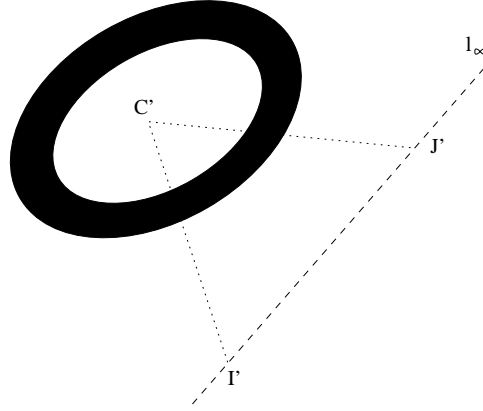


Figura 4.16: Dos círculos concéntricos intersectan en los puntos circulares, que están en la línea del horizonte.

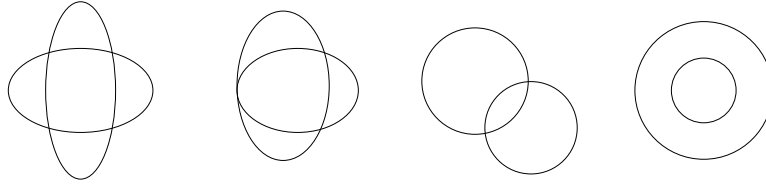


Figura 4.17: Intersecciones entre cónicas

Una de las razones para descartar este método es que hemos encontrado que el cálculo de la intersección de las elipses es muy sensible al ruido.

El cálculo de los puntos circulares nos ofrece otro método para calcular el centro proyectado sobre la imagen de dos círculos concéntricos. Conocidas las coordenadas de la proyección de los puntos circulares en la imagen, I' y J' , se puede calcular la línea de horizonte del plano donde se encuentran los círculos [25, 38]. La línea que pasa por dos puntos es sencillamente su producto vectorial:

$$l_{\infty} = I' \times J'$$

Como l_{∞} está en configuración polo-polar con el centro proyectado de las elipses, podemos calcular:

$$C' = E_1^{-1} l_{\infty}$$

donde E_1 es la matriz de la elipse correspondiente.

Una vez calculados los cuatro puntos, y teniendo su posición en el sistema de coordenadas global, se puede calcular la homografía planar. En la Tabla 4.1 se puede encontrar a modo de resumen los puntos que se utilizarán.

<i>Punto en la imagen</i>	<i>Punto 3D</i>	<i>Obtenido a partir de...</i>
Puntos circulares: I', J'	$\begin{bmatrix} 1 & i & 0 & 0 \\ 1 & -i & 0 & 0 \end{bmatrix}^T$	la intersección de las dos elipses
Centro proyectado de los círculos: C'	$\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$	alguno de los métodos vistos en la página 120, o el mencionado en el apartado anterior
Marca del eje X: M'	$\begin{bmatrix} r_3 & 0 & 0 & 1 \end{bmatrix}^T$	la detección del marcador (página 112)

Tabla 4.1: Puntos utilizados para obtener la homografía

A continuación sólo queda seguir los pasos para calcular los elementos de H [89, 38] y, a partir de ellos, calcular las matrices de calibración de la cámara. En el Apartado 2.3.3 se describió un método de calibración de cámaras utilizando homografías.

4.6. Resumen

Este capítulo ha mostrado el diseño de un sistema para la integración dinámica de objetos sintéticos en un vídeo de una escena real en tiempo real. Se ha diseñado un nuevo sistema de localización óptica teniendo en cuenta algunas características deseables para el objetivo buscado de obtener una buena calidad en la integración. Así, se ha buscado un diseño que, a la vez que permita obtener los parámetros necesarios para establecer la cámara sintética, sea lo menos intrusivo posible en la escena.

Se ha estudiado la robustez del sistema de localización mediante experimentos sintéticos. Además, se ha mostrado la utilidad del mismo sobre vídeos de escenas reales. Se han aportado distintos ejemplos de aplicaciones del sistema a:

- realidad aumentada,
- efectos digitales, y
- interfaces de usuario.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

En una sociedad como la actual, en la que el ocio es tan importante, tecnologías aplicables a campos como los videojuegos, cine, televisión, etc. tienen gran interés. El constante aumento en las capacidades de cálculo de las computadoras y la progresiva bajada de precios en la tecnología de captura de imágenes y visualización, permite pensar en la aparición de entornos de realidad aumentada en entornos domésticos y en otros campos aún no explorados.

El estudio del estado del arte en este área pone de manifiesto que se está produciendo una confluencia de distintas disciplinas en aras de alcanzar herramientas más potentes. Disciplinas tan relacionadas, y sin embargo tan lejanas entre sí como gráficos por computador y visión por computador se implican cada vez más para aprovechar lo mejor de ambos mundos. Es de esperar que de esta conjunción se obtengan resultados cada vez más espectaculares.

En esta tesis hemos demostrado la posibilidad de realizar efectos especiales avanzados con hardware convencional. En este sentido, se han presentado dos aplicaciones cuyos resultados permiten acercar al gran público técnicas que normalmente están reservadas a profesionales.

En la primera parte del trabajo se ha presentado una técnica para integrar de una forma fotorrealista objetos sintéticos en la fotografía de una escena real. Dicho fotorrealismo se ha alcanzado gracias al cálculo de la interacción entre los objetos sintéticos y el mundo real, permitiendo que uno modifique al otro. Se ha demostrado que la inclusión de sombras y reflexiones, junto con una alineación precisa de los objetos con la escena permiten generar imágenes muy realistas. La utilización de técnicas como la calibración de cámaras y las imágenes de alto rango dinámico han permitido

mejorar sensiblemente la composición de imágenes.

Además, este proceso se ha realizado con una cámara fotográfica digital y otro material convencional, aparte de software disponible libremente. En contraposición con los sistemas profesionales, proponemos una técnica fácilmente exportable al entorno doméstico.

En la segunda parte del trabajo, por otra parte, se ha buscado una integración de objetos sintéticos en tiempo real. En este caso, el requisito que ha primado sobre los demás ha sido el del registro temporal, es decir, la creación en tiempo real del efecto. Este requisito lleva aparejado, por el momento, la necesidad de utilizar hardware y motores gráficos convencionales, con la consiguiente merma en el fotorrealismo de los objetos sintéticos. Sin embargo, la alineación de un objeto sintético en una escena real, y la posibilidad de implementar algún “truco”, como el borrado del marcador de localización o la generación de polígonos de sombra, permite obtener resultados muy vistosos, en los que el tiempo real suple la calidad gráfica.

Hemos presentado una solución completa para un sistema de localización basado en visión. Nuestro sistema es sencillo de construir, barato, fácil de utilizar y es poco intrusivo. Tiene características deseables tanto para las aplicaciones de realidad aumentada como para las de composición digital. Nuestro sistema no hace suposiciones sobre el dispositivo de captura utilizado (es decir, no se tiene que precalibrar) y permite la modificación de los parámetros de la cámara durante la captura. Nuestro localizador es sencillo de utilizar por un usuario no experto porque la única información que necesita es la longitud de los tres radios que componen el marcador. Mientras que el marcador esté visible, nuestro localizador calcula los parámetros necesarios para configurar adecuadamente el modelo de cámara sintética que se utilice para visualizar los objetos sintéticos. Dichos parámetros están definidos sobre un sistema de coordenadas global compartido por la escena real y por la sintética, centrado en el marcador. Se han presentado varios resultados que demuestran al localizador en diversas aplicaciones, como interfaces de usuario avanzadas, realidad aumentada o efectos digitales.

Entre las aportaciones realizadas por este trabajo, podemos enumerar:

- La demostración de cómo componer objetos sintéticos en imagen real de una forma sencilla y sin hardware especializado.
- El desarrollo de un flujo de trabajo claro para el desarrollo del efecto.
- La integración de técnicas de distintos campos para la mejora del sistema.
- El diseño de un nuevo sistema de localización sencillo de construir y de operar.
- La utilización de invariantes proyectivos para la detección y reconstrucción del marcador.

- El desarrollo de un algoritmo de calibración de la cámara a partir del marcador.
- El diseño de un método de obtención de vídeos calibrados por medio de marcadores sin marcadores, y con una sola pasada.
- La demostración de la viabilidad del método mediante la implementación de distintas aplicaciones.

A continuación se enumeran las publicaciones científicas generadas en el transcurso de la tesis:

- Francisco Abad, Emilio Camahort y Roberto Vivó. Camera Calibration Using Two Concentric Circles. *Lecture Notes in Computer Science*. **Aceptado**.
- Francisco Abad, Emilio Camahort y Roberto Vivó. Integrating synthetic objects into real scenes. *Computers & Graphics. Elsevier Science*, 27:5–17, 2003.
- Francisco Abad, Emilio Camahort y Roberto Vivó. On the Integration of Synthetic Objects with Real-World Scenes *Eurographics. Short Presentations Proceedings*, 237–246, 2002.
- Francisco Abad, Emilio Camahort y Roberto Vivó. Antecedentes y fundamentos de la integración de objetos sintéticos en escenas reales. *Informe técnico DSIC-II/05/02*, 2002.
- Francisco Abad, Emilio Camahort y Roberto Vivó. Una técnica para la integración de objetos sintéticos en escenas reales. *Informe técnico DSIC-II/06/02*, 2002.

5.2. Trabajos futuros

A lo largo de este trabajo se han mostrado distintos campos en los que se han aplicado técnicas parecidas a las propuestas, tales como el cine, la televisión, la realidad aumentada o la asistencia a la docencia. En sus dos vertientes, tanto la composición fotorrealista en imagen estática, como la animación y el tiempo real presentan campos de trabajo atractivos.

Un problema que se ha puesto de manifiesto durante el trabajo y que requiere un esfuerzo adicional es del registro automático de la secuencia de fotografías tomadas para generar una imagen de alto rango dinámico. Aunque lo ideal sería que la cámara permaneciera totalmente inmóvil durante la toma de fotografías, la manipulación de los parámetros de apertura y velocidad de obturación puede provocar pequeños movimientos. Dichos movimientos hacen que la imagen pueda aparecer con errores y con bordes difusos. Para solucionar dichos movimientos de cámara se pueden aplicar

técnicas de estabilización de imagen [16], que automáticamente alineen todas las fotografías.

Otro campo que no se ha abordado en este trabajo, y que tiene una importancia capital para que este tipo de técnicas sea accesible a usuarios no expertos, es la estimación automática de las propiedades reflectantes de las superficies reales, es decir, estimar la BRDF de las superficies de la escena.

El siguiente paso para extender este trabajo sería la introducción de técnicas de reconstrucción de geometría basadas en visión por computador. Dichas técnicas permitirían, bien capturar la geometría de la escena, o bien obtener alguna clase de estimación de la profundidad de los objetos de la misma para poder ordenar e intercalar objetos reales y sintéticos. Para ello, son aplicables técnicas como la captura de la forma a partir del movimiento, a partir de las sombras, estéreo, etc.

Respecto al sistema de localización, se debería intentar mejorar el proceso de calibración del marcador para hacerlo más robusto al ruido en la imagen y a oclusiones parciales del marcador. El siguiente paso sería permitir el uso de distintos marcadores para incrementar el tamaño de la escena real que se puede visitar, y para mejorar, en aquellos casos que fuera necesario, la precisión del proceso de calibración.

Una posible solución para la visualización de objetos sintéticos en entornos reales con iluminación realista en los límites del tiempo real, puede ser la aplicación de la técnica de los campos de luz [36, 51]. Los campos de luz son estructuras altamente optimizadas para el almacenamiento de información luminosa. Mediante los algoritmos de consulta adecuados pueden generar nuevas vistas de una escena en tiempo real, interpolando la información de mapas de radiancia existente. El estudio de dichas estructuras podría proporcionar un medio adecuado para almacenar los mapas de entorno con los que iluminar los objetos sintéticos. Entroncado con este objetivo estaría el estudio de la generación eficiente de mapas de entorno.

Apéndice A

Derivación de la proyección del marcador

A continuación se muestra la derivación de la fórmula 4.13 en el Capítulo 4.
Matriz de parámetros intrínsecos:

$$A = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matriz de parámetros extrínsecos:

$$M = \begin{pmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \end{pmatrix} = (\vec{R}_1 \quad \vec{R}_2 \quad \vec{R}_3 \quad \vec{T})$$

Matriz de proyección:

$$P = AM = \begin{pmatrix} fR_{11} + u_0R_{31} & fR_{12} + u_0R_{32} & fR_{13} + u_0R_{33} & fT_x + u_0T_z \\ fR_{21} + v_0R_{31} & fR_{22} + v_0R_{32} & fR_{23} + v_0R_{33} & fT_y + v_0T_z \\ R_{31} & R_{32} & R_{33} & T_z \end{pmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \quad (\text{A.1})$$

El marcador está situado en el plano $Z = 0$, por lo tanto:

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{pmatrix} \begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11}X_w + p_{12}Y_w + p_{14} \\ p_{21}X_w + p_{22}Y_w + p_{24} \\ p_{31}X_w + p_{32}Y_w + p_{34} \end{bmatrix}$$

Calculando u y v :

$$u = \frac{p_{11}X_w + p_{12}Y_w + p_{14}}{p_{31}X_w + p_{32}Y_w + p_{34}}$$

$$v = \frac{p_{21}X_w + p_{22}Y_w + p_{24}}{p_{31}X_w + p_{32}Y_w + p_{34}}$$

Despejando X_w e Y_w de la ecuación anterior:

$$X_w = \frac{u(p_{22}p_{34} - p_{24}p_{32}) + v(p_{14}p_{32} - p_{12}p_{34}) + p_{12}p_{24} - p_{14}p_{22}}{u(p_{21}p_{32} - p_{22}p_{31}) + v(p_{12}p_{31} - p_{11}p_{32}) + p_{11}p_{22} - p_{12}p_{21}} \quad (\text{A.2})$$

$$Y_w = \frac{u(p_{24}p_{31} - p_{21}p_{34}) + v(p_{11}p_{34} - p_{14}p_{31}) + p_{14}p_{21} - p_{11}p_{24}}{u(p_{21}p_{32} - p_{22}p_{31}) + v(p_{12}p_{31} - p_{11}p_{32}) + p_{11}p_{22} - p_{12}p_{21}} \quad (\text{A.3})$$

Substituyendo cada p_{ij} por su valor definido en la ecuación A.1, paso a paso:

$$\begin{aligned} u(p_{21}p_{32} - p_{22}p_{31}) &= u(R_{32}(fR_{21} + v_0R_{31}) - R_{31}(fR_{22} + v_0R_{32})) = \\ &= u(fR_{32}R_{21} + v_0R_{31}R_{32} - fR_{31}R_{22} - v_0R_{31}R_{32}) = \\ &= uf(R_{32}R_{21} - R_{31}R_{22}) \end{aligned}$$

$$\begin{aligned} v(p_{12}p_{31} - p_{11}p_{32}) &= v(R_{31}(fR_{12} + u_0R_{32}) - R_{32}(fR_{11} + u_0R_{31})) = \\ &= v(fR_{12}R_{31} + u_0R_{31}R_{32} - fR_{11}R_{32} - u_0R_{31}R_{32}) = \\ &= vf(R_{12}R_{31} - R_{11}R_{32}) \end{aligned}$$

$$\begin{aligned} p_{11}p_{22} - p_{12}p_{21} &= (fR_{11} + u_0R_{31})(fR_{22} + v_0R_{32}) - \\ &= (fR_{12} + u_0R_{32})(fR_{21} + v_0R_{31}) = \\ &= f^2R_{11}R_{22} + v_0fR_{11}R_{32} + u_0fR_{31}R_{22} + u_0v_0R_{31}R_{32} - \\ &= f^2R_{12}R_{21} - v_0fR_{12}R_{31} - u_0fR_{21}R_{32} - u_0v_0R_{31}R_{32} = \\ &= u_0f(R_{31}R_{22} - R_{21}R_{32}) + v_0f(R_{11}R_{32} - R_{12}R_{31}) + \\ &= f^2(R_{11}R_{22} - R_{12}R_{21}) \end{aligned}$$

Por lo tanto, el denominador de las ecuaciones A.2 y A.3 es:

$$\begin{aligned} &u(p_{21}p_{32} - p_{22}p_{31}) + v(p_{12}p_{31} - p_{11}p_{32}) + p_{11}p_{22} - p_{12}p_{21} = \\ &= uf(R_{32}R_{21} - R_{31}R_{22}) + vf(R_{12}R_{31} - R_{11}R_{32}) + \\ &= u_0f(R_{31}R_{22} - R_{21}R_{32}) + v_0f(R_{11}R_{32} - R_{12}R_{31}) + \\ &= f^2(R_{11}R_{22} - R_{12}R_{21}) = \\ &= f[(u - u_0)(R_{21}R_{32} - R_{22}R_{31}) + (v - v_0)(R_{12}R_{31} - R_{11}R_{32}) + \\ &= f(R_{11}R_{22} - R_{12}R_{21})] = \\ &= f \left[\begin{pmatrix} u - u_0 & v - v_0 & f \end{pmatrix} \begin{pmatrix} R_{21}R_{32} - R_{22}R_{31} \\ R_{12}R_{31} - R_{11}R_{32} \\ R_{11}R_{22} - R_{12}R_{21} \end{pmatrix} \right] = \\ &= f \left[\begin{pmatrix} u - u_0 & v - v_0 & f \end{pmatrix} \begin{pmatrix} \vec{R}_1 \times \vec{R}_2 \end{pmatrix} \right] = \\ &= f \begin{pmatrix} u - u_0 & v - v_0 & f \end{pmatrix} \vec{R}_3 = f(\vec{X} \vec{R}_3), \end{aligned}$$

donde $\vec{X} = \begin{bmatrix} u - u_0 & v - v_0 & f \end{bmatrix}$. Sustituyendo y desarrollando, la Ecuación A.2 queda:

$$X_w = \frac{(u - u_0)(T_z R_{22} - T_y R_{32}) + (v - v_0)(T_x R_{32} - T_z R_{12}) + f(T_y R_{12} - T_x R_{22})}{\vec{X} \vec{R}_3}$$

y, en forma vectorial:

$$X_w = \frac{\begin{pmatrix} u - u_0 & v - v_0 & f \end{pmatrix} \begin{pmatrix} T_z R_{22} - T_y R_{32} \\ T_x R_{32} - T_z R_{12} \\ T_y R_{12} - T_x R_{22} \end{pmatrix}}{\vec{X} \vec{R}_3} = \frac{\vec{X} (\vec{R}_2 \times \vec{T})}{\vec{X} \vec{R}_3}. \quad (\text{A.4})$$

De la misma forma:

$$Y_w = \frac{\begin{pmatrix} u - u_0 & v - v_0 & f \end{pmatrix} \begin{pmatrix} T_y R_{31} - T_z R_{21} \\ T_z R_{11} - T_x R_{31} \\ T_x R_{21} - T_y R_{11} \end{pmatrix}}{\vec{X} \vec{R}_3} = \frac{\vec{X} (\vec{T} \times \vec{R}_1)}{\vec{X} \vec{R}_3} \quad (\text{A.5})$$

En el sistema de coordenadas global, las coordenadas de los puntos del círculo exterior de radio r_2 deben cumplir la siguiente relación:

$$C(X_w, Y_w) = X_w^2 + Y_w^2 - r_2^2 = 0 \quad (\text{A.6})$$

Sustituyendo las ecuaciones A.4 y A.5 en A.6, queda:

$$C'(u, v) = \left(\frac{\vec{X} (\vec{R}_2 \times \vec{T})}{\vec{X} \vec{R}_3} \right)^2 + \left(\frac{\vec{X} (\vec{T} \times \vec{R}_1)}{\vec{X} \vec{R}_3} \right)^2 - r_2^2 = 0$$

Desarrollando:

$$C'(u, v) = \left(\vec{X} (\vec{R}_2 \times \vec{T}) \right)^2 + \left(\vec{X} (\vec{T} \times \vec{R}_1) \right)^2 - r_2^2 (\vec{X} \vec{R}_3)^2 = 0.$$

Suponiendo conocido el punto principal, X se puede suponer descrito en el sistema de coordenadas xy de la Figura 4.8 de la página 118 y, por lo tanto, $u_0 = v_0 = 0$. Desarrollando la expresión anterior, se obtiene una (gran) expresión, que se puede simplificar aplicando las propiedades de las matrices de rotación. Una matriz de rotación es una matriz ortonormal, es decir, tanto sus filas como sus columnas tienen módulo 1, y además son ortogonales entre ellas. Por ello dada una matriz de rotación R , podríamos enumerar las siguientes propiedades:

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix}$$

- $R_{11}^2 + R_{12}^2 + R_{13}^2 = 1$
- $R_{21}^2 + R_{22}^2 + R_{23}^2 = 1$

$$\blacksquare R_{31}^2 + R_{32}^2 + R_{33}^2 = 1$$

y, de la misma forma:

$$\blacksquare R_{11}^2 + R_{21}^2 + R_{31}^2 = 1$$

$$\blacksquare R_{12}^2 + R_{22}^2 + R_{32}^2 = 1$$

$$\blacksquare R_{13}^2 + R_{23}^2 + R_{33}^2 = 1$$

Como consecuencia de ello, se puede escribir, por ejemplo:

$$\blacksquare R_{11}^2 + R_{12}^2 = 1 - R_{13}^2 = R_{23}^2 + R_{33}^2$$

$$\blacksquare R_{21}^2 + R_{22}^2 = 1 - R_{23}^2 = R_{13}^2 + R_{33}^2$$

$$\blacksquare R_{31}^2 + R_{32}^2 = 1 - R_{33}^2 = R_{13}^2 + R_{23}^2$$

Con respecto a la ortogonalidad de las filas (y columnas) de la matriz, podemos escribir:

$$\blacksquare R_{11}R_{21} + R_{12}R_{22} + R_{13}R_{23} = 0$$

$$\blacksquare R_{11}R_{31} + R_{12}R_{32} + R_{13}R_{33} = 0$$

$$\blacksquare R_{21}R_{31} + R_{22}R_{32} + R_{23}R_{33} = 0$$

y sus equivalentes por columnas.

Aplicando las relaciones anteriores para expresar C' en términos de la tercera columna de la matriz de rotación, quedaría:

$$\begin{aligned} C'(x, y) = & x^2(T_y^2(R_{13}^2 + R_{23}^2) + 2T_yT_zR_{23}R_{33} + T_z^2(R_{13}^2 + R_{33}^2) - r_2^2R_{13}^2) \\ & - 2xy(r_2^2R_{13}R_{23} + T_x(T_y(R_{13}^2 + R_{23}^2) + T_zR_{23}R_{33}) + T_z(T_yR_{13}R_{33} - T_zR_{13}R_{23})) \\ & + y^2(T_x^2(R_{13}^2 + R_{23}^2) + 2T_xT_zR_{13}R_{33} + T_z^2(R_{23}^2 + R_{33}^2) - r_2^2R_{23}^2) \\ & - 2xf(r_2^2R_{13}R_{33} + T_x(T_z(R_{13}^2 + R_{33}^2) + T_yR_{23}R_{33}) + T_y(T_zR_{13}R_{23} - T_yR_{13}R_{33})) \\ & - 2yf(r_2^2R_{23}R_{33} - T_x^2R_{23}R_{33} + T_x(T_yR_{13}R_{33} + T_zR_{13}R_{23}) + T_yT_z(R_{23}^2 + R_{33}^2)) \\ & + f^2(T_x^2(R_{13}^2 + R_{33}^2) + T_y(2T_xR_{13}R_{23} + T_y(R_{23}^2 + R_{33}^2)) - r_2^2R_{33}^2) = 0 \end{aligned}$$

Realizando en la ecuación anterior las siguientes sustituciones, ya explicadas en el Apartado 4.3.2 de la página 121:

$$T_x = \frac{T_z x_c}{f} \quad T_y = \frac{T_z y_c}{f}$$

$$R_{13} = \frac{\alpha_1}{n} \quad R_{23} = \frac{\alpha_2}{n} \quad R_{33} = \frac{\alpha_3}{fn}$$

resulta, después de desarrollar y multiplicar por nf^2 para simplificar:

$$\begin{aligned}
C'(x, y) = & x^2(y_c^2 T_z^2(\alpha_1^2 + \alpha_2^2) + 2\alpha_2\alpha_3 y_c T_z^2 + T_z^2(\alpha_1^2 f^2 + \alpha_3^2) - \alpha_1^2 r_2^2 f^2) \\
& - 2xy(\alpha_1\alpha_2 r_2^2 f^2 + x_c T_z^2((\alpha_1^2 + \alpha_2^2)y_c + \alpha_2\alpha_3) + T_z^2(\alpha_1\alpha_3 y_c - \alpha_1\alpha_2 f^2)) \\
& + y^2(x_c^2 T_z^2(\alpha_1^2 + \alpha_2^2) + 2\alpha_1\alpha_3 x_c T_z^2 + T_z^2(\alpha_2^2 f^2 + \alpha_3^2) - \alpha_2^2 r_2^2 f^2) \\
& - 2x(\alpha_1\alpha_3 f^2 r_2^2 + x_c T_z^2(\alpha_1^2 f^2 + \alpha_3^2 + \alpha_2\alpha_3 y_c) + y_c T_z^2(\alpha_1\alpha_2 f^2 - \alpha_1\alpha_3 y_c)) \\
& - 2y(\alpha_2\alpha_3 f^2 r_2^2 - x_c^2 T_z^2\alpha_2\alpha_3 + x_c T_z^2(\alpha_1\alpha_3 y_c + \alpha_1\alpha_2 f^2) + y_c T_z^2(\alpha_2^2 f^2 + \alpha_3^2)) \\
& + x_c^2 T_z^2(\alpha_1^2 f^2 + \alpha_3^2) + y_c T_z^2(2\alpha_1\alpha_2 x_c f^2 + y_c(\alpha_2^2 f^2 + \alpha_3^2)) - \alpha_3^2 r_2^2 f^2 = 0
\end{aligned}$$

Agrupando términos en función de f^2 y $\frac{f^2}{T_z^2}$ y separando cada uno de los términos de la elipse en x^2 , xy , y^2 , x , y y 1 queda, finalmente:

$$\begin{aligned}
C'(x, y) = & x^2 \left(\alpha_1^2 f^2 - \alpha_1^2 r_2^2 \frac{f^2}{T_z^2} + (\alpha_1^2 + \alpha_2^2)y_c^2 + 2\alpha_2\alpha_3 y_c + \alpha_3^2 \right) \\
& + xy \left(2\alpha_1\alpha_2 f^2 - 2\alpha_1\alpha_2 r_2^2 \frac{f^2}{T_z^2} - 2(\alpha_2\alpha_3 x_c + (\alpha_1^2 + \alpha_2^2)x_c y_c + \alpha_1\alpha_3 y_c) \right) \\
& + y^2 \left(\alpha_2^2 f^2 - \alpha_2^2 r_2^2 \frac{f^2}{T_z^2} + (\alpha_1^2 + \alpha_2^2)x_c^2 + 2\alpha_1\alpha_3 x_c + \alpha_3^2 \right) \\
& + x \left(-2\alpha_1(\alpha_1 x_c + \alpha_2 y_c)f^2 - 2\alpha_1\alpha_3 r_2^2 \frac{f^2}{T_z^2} - 2\alpha_3(\alpha_3 x_c + \alpha_2 x_c y_c - \alpha_1 y_c^2) \right) \\
& + y \left(-2\alpha_2(\alpha_1 x_c + \alpha_2 y_c)f^2 - 2\alpha_2\alpha_3 r_2^2 \frac{f^2}{T_z^2} - 2\alpha_3(\alpha_1 x_c y_c - \alpha_2 x_c^2 + \alpha_3 y_c) \right) \\
& + \alpha_1^2 x_c^2 + \alpha_2 y_c(2\alpha_1 x_c + \alpha_2 y_c)f^2 - \alpha_3^2 r_2^2 \frac{f^2}{T_z^2} + \alpha_3^2(x_c^2 + y_c^2) = 0
\end{aligned}$$

que se puede escribir matricialmente del siguiente modo:

$$C'(x, y) = \begin{bmatrix} x^2 & xy & y^2 & x & y & 1 \end{bmatrix} Q' \begin{bmatrix} f^2 \\ \frac{f^2}{T_z^2} \\ 1 \end{bmatrix},$$

donde Q' es la matriz 4.13 de la página 122.

Bibliografía

- [1] Francisco Abad, Emilio Camahort, y Roberto Vivó. Antecedentes y fundamentos de la integración de objetos sintéticos en escenas reales. Informe Técnico DSIC-II/05/02, Depto. Sistemas Informáticos y Computación. Univ. Politécnica de Valencia, 2002.
- [2] Francisco Abad, Emilio Camahort, y Roberto Vivó. On the integration of synthetic objects with real-world scenes. En *Short Presentations Proceedings*, páginas 237–246. Eurographics, 2002.
- [3] Francisco Abad, Emilio Camahort, y Roberto Vivó. Integrating synthetic objects into real scenes. *Computers & Graphics*, 27(1):5–17, 2003.
- [4] Francisco Abad, Emilio Camahort, y Roberto Vivó. Camera calibration using two concentric circles. *Lecture Notes in Computer Science*, 2004. Aceptado.
- [5] Sameer Agarwal, Ravi Ramamoorthi, Serge Belongie, y Henrik Wann Jensen. Structured importance sampling of environment maps. *ACM Transactions on Graphics*, 22(3):605–612, julio 2003.
- [6] Sung Joon Ahn, Wolfgang Rauh, y Sung Il Kim. Circular coded target for automation of optical 3D-measurement and camera calibration. *Int. Journal Pattern Recognition and Artificial Intelligence*, 15(6):905–919, 2001.
- [7] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, y Blair MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, noviembre/diciembre 2001.
- [8] Ronald T. Azuma. A survey of augmented reality. En *Presence: Teleoperators and Virtual Environments*, tomo 4(6), páginas 355–385. 1997.
- [9] Dana H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.
- [10] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, y Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Processing*, 12(8):882–889, aug 2003.
- [11] Mark Billinghurst y Hirokazu Kato. Collaborative mixed reality. En *International Symposium on Mixed Reality*, páginas 261–284. 1999.
- [12] James F. Blinn y Martin E. Newell. Texture and reflection in computer generated images. *Commun. ACM*, 19(10):542–547, 1976.
- [13] Jean-Yves Bouguet. Nota en la distribución del *toolbox* de calibración en OpenCV, 2001.

- [14] Darius Burschka, Dana Cobzas, Zach Dodds, Greg D. Hager, Martin Jagersand, y Keith Yerec. Recent methods for image-based modeling and rendering. En *IEEE Virtual Reality Tutorial*. 2003. <http://www.cs.ualberta.ca/~vis/VR2003tut>.
- [15] Kenneth R. Castleman. *Digital Image Processing*. Prentice-Hall, 1996.
- [16] Alberto Censi, Andrea Fusiello, y Vito Roberto. Image stabilization by features tracking. En *Proceedings of the IEEE International Conference on Image Analysis and Processing (ICIAP '99)*, páginas 665–667. IEEE Computer Society Press, 1999.
- [17] Shenchang Eric Chen. QuickTime VR - An Image-Based Approach to Virtual Environment Navigation. En *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '95)*, páginas 29–38. 1995.
- [18] Adrian David Cheok, Siew Wan Fong, Kok Hwee Goh, Xubo Yang, Wei Liu, y Farzam Farbiz. Human pacman: A mobile entertainment system with ubiquitous computing and tangible interaction over a wide outdoor area. En *Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services*. 2003.
- [19] Kristin J. Dana, Bram van Ginneken, Shree K. Nayar, y Jan J. Koenderink. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, enero 1999.
- [20] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. En *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '98)*, tomo 30, páginas 189–198. 1998.
- [21] Paul Debevec. Image-Based Lighting. *IEEE Computer Graphics and Applications*, 22(2):26–34, 2002.
- [22] Paul E. Debevec y Jitendra Malik. Recovering high dynamic range radiance maps from photographs. En Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, páginas 369–378. ACM SIGGRAPH, Addison Wesley, agosto 1997.
- [23] Paul E. Debevec, Camillo J. Taylor, y Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Computer Graphics*, 30(Annual Conference Series):11–20, 1996.
- [24] George Drettakis, Luc Robert, y Sylvain Bugnoux. Interactive common illumination for computer augmented reality. En *8th Eurographics workshop on Rendering*. Saint Etienne, France, Jun 1997.
- [25] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, MA, 1993.
- [26] Andrew W. Fitzgibbon, Maurizio Pilu, y Robert B. Fisher. Direct least squares fitting of ellipses. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.
- [27] James Foley, Andries van Dam, Steven Feiner, John Hughes, y Richard Phillips. *Introduction to Computer Graphics*. Addison-Wesley Publishing Company, Cambridge, MA, 1993.

- [28] Keith Forbes, Anthon Voigt, y Ndimi Bodika. An inexpensive, automatic and accurate camera calibration method. En *Thirteenth Annual Symposium of the Pattern Recognition Association of South Africa*. 2002.
- [29] David Forsyth, Joseph L. Mundy, Andrew Zisserman, Chris Coelho, Aaron Heller, y Charles Rothwell. Invariant descriptors for 3-d object recognition and pose. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(10):971–991, October 1991.
- [30] Alain Fournier, Atjeng Gunawan, y Chris Romanzin. Common illumination between real and computer generated scenes. En *Proc. Graphics Interface '93*, páginas 254–262. 1993.
- [31] Eric Foxlin y Leonid Naimark. Miniaturization, calibration & accuracy evaluation of a hybrid self-tracker. En *International Symposium on Mixed and Augmented Reality*, páginas 151–160. 2003.
- [32] Vincent Fremont y Ryad Chellali. Direct camera calibration using two concentric circles from a single view. En *12th International Conference on Artificial Reality and Telexistence*. 2002.
- [33] Simon Gibson, Toby Howard, y Roger Hubbard. Flexible image-based photometric reconstruction using virtual light sources. *Computer Graphics Forum*, 20(3), 2001. ISSN 1067-7055.
- [34] Simon Gibson y Alan Murta. Interactive rendering with real world illumination. En B. Peroche y H. Rushmeier, editores, *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)*. Springer Wien, New York, NY, 2000. 365–376.
- [35] Rafael C. Gonzalez y Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [36] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, y Michael F. Cohen. The lumigraph. En *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '96)*, páginas 43–54. 1996.
- [37] Paul Haeberli y Mark Segal. Texture mapping as a fundamental drawing primitive. En Michael F. Cohen, Claude Puech, y Francois Sillion, editores, *Fourth Eurographics Workshop on Rendering*, páginas 259–266. Eurographics, junio 1993. Held in Paris, France, 14–16 June 1993.
- [38] Richard Hartley y Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2002.
- [39] Wolfgang Heidrich. Environment Maps and Their Applications. En *SIGGRAPH Course notes - Real-Time Shading*. 2000.
- [40] Janne Heikkilä y Olli Silvén. A four-step camera calibration procedure with implicit image correction. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1996.
- [41] Tadashi Ichikawa, Kunio Yamada, Takahumi Kanamaru, Takeshi Naemura, Kiyoharu Aizawa, Shigeo Morishima, y Takahiro Saito. Multimedia ambience communication. *IEEE Signal Processing Magazine*, 18(3):43–50, 2001.

- [42] Pierre Jancène, Fabrice Neyret, Xavier Provot, Jean-Philippe Tarel, Jean-Marc Vézien, Christophe Meilhac, y Anne Vérroust. RES: computing the interactions between real and virtual objects in video sequences. En *Second IEEE Workshop on Networked Realities*, páginas 27–40. Boston, Massachusetts (USA), octubre 1995.
- [43] Kenichi Kanatani y Wu Liu. 3D interpretation of conics and orthogonality. *CVGIP: Image Understanding*, 58(3):286–301, November 1993.
- [44] Doug Kelly. *Digital compositing in depth*. The Coriolis Group, 2000.
- [45] Sara Keren, Ilan Shimshoni, y Ayellet Tal. Placing three-dimensional models in an uncalibrated single image of an architectural scene. En *Proceedings of the ACM symposium on Virtual reality software and technology*, páginas 186–193. ACM Press, 2002.
- [46] Jun-Sik Kim, Ho-Won Kim, y In So Kweon. A camera calibration method using concentric circles for vision applications. En *5th Asian Conference on Computer Vision*. 2002.
- [47] Jun-Sik Kim y In-So Kweon. A new camera calibration method for robotic applications. En *IEEE Int. Conference on Intelligent Robots and Systems*, páginas 778–783. 2001.
- [48] Michael Langford. *Tratado de fotografía*. Ediciones Omega, 1994.
- [49] Greg Ward Larson y Rob Shakespeare. *Rendering with Radiance: The Art and Science of Lighting Visualization*. Morgan Kaufmann, San Francisco, CA, 1998.
- [50] Gregory Ward Larson, Holly Rushmeier, y Christine Piatko. A Visibility Matching Tone Reproduction Operator for High Dynamic Range Scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997.
- [51] Marc Levoy y Pat Hanrahan. Light field rendering. *Computer Graphics*, 30(Annual Conference Series):31–42, 1996.
- [52] David Liebowitz y Andrew Zisserman. Metric rectification for perspective images of planes. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 482–488. 1998.
- [53] Diego López de Ipiña, Paulo R. S. Mendonça, y Andy Hopper. Trip: A low-cost vision-based location system for ubiquitous computing. *Personal Ubiquitous Comput.*, 6(3):206–219, 2002.
- [54] Céline Loscos, George Drettakis, y Luc Robert. Interactive virtual relighting of real scenes. *IEEE Transactions on Visualization and Computer Graphics*, 6(4):289–305, 2000.
- [55] Céline Loscos, Marie-Claude Frasson, George Drettakis, Bruce Walter, Xavier Granier, y Pierre Poulin. Interactive virtual relighting and remodeling of real scenes. En *Rendering techniques '99 (Proceedings of the 10th Eurographics Workshop on Rendering)*, tomo 10, páginas 235–246. Jun 1999.
- [56] Sebastian Magda, David Kriegman, Todd Zickler, y Peter Belhumeur. Beyond lambert: Reconstructing surfaces with arbitrary BRDFs. En *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01)*, páginas 391–399. IEEE Computer Society, Los Alamitos, CA, julio 9–12 2001.

- [57] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, y Kenneth Torrance. Image-based bidirectional reflectance distribution function measurement. *Applied Optics-OT*, 39(16):2592–2600, junio 2000.
- [58] Vincent Masselus, Philip Dutre, y Frederik Anrys. The free form light stage. En *13th Eurographics Workshop on Rendering*. Eurographics Association, 2002.
- [59] Xiaoqiao Meng y Zhanyi Hu. A new easy camera calibration technique based on circular points. *Pattern Recognition*, 5(36):1155–1164, 2003.
- [60] Gene S. Miller y C. Robert Hoffman. Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments. En *SIGGRAPH Course notes - Advanced Computer Graphics Animation*. 1984.
- [61] Jorge J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. *Lecture Notes in Mathematics*, 630:105–116, 1978.
- [62] David M. Mount, Nathan S. Netanyahu, y Jacqueline Le Moigne. Efficient algorithms for robust feature matching. *Pattern Recognition*, 32:17–28, 1999.
- [63] Joseph Mundy y Andrew Zisserman, editores. *Geometric Invariance in Computer Vision*. MIT Press, 1992.
- [64] Leonid Naimark y Erik Foxlin. Circular data matrix fiducial and robust image processing for a wearable vision-inertial self-tracker. En *International Symposium on Mixed and Augmented Reality*, páginas 27–36. 2002.
- [65] Eihachiro Nakamae, Koichi Harada, Takao Ishizaki, y Tomoyuki Nishita. A montage method: the overlaying of the computer generated images onto a background photograph. En *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, páginas 207–214. ACM Press, 1986.
- [66] Shree Nayar y Tomoo Mitsunaga. High dynamic range imaging: Spatially varying pixel exposures. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, páginas 472–479. IEEE, Los Alamitos, junio 13–15 2000.
- [67] Patricia Netzley. *Encyclopedia of Movie Special Effects*. Greenwood Publishing Group, 1999.
- [68] Francesca Odone y Andrea Fusiello. Applications of 2D image registration. Informe Técnico Research Memorandum RM/99/15, Department of Computing and Electrical Engineering, Heriot-Watt University, Edinburgh, UK, 1999. <http://profs.sci.univr.it/~fusiello/resume/pap-itl.html>.
- [69] OpenGL Architecture Review Board. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.4*. Addison-Wesley, 2003.
- [70] Tekla S. Perry. All in the Game. *IEEE Spectrum*, 40(11), Nov 2003.
- [71] Wayne Piekarski y Bruce Thomas. ARQuake: The Outdoor Augmented Reality Gaming System. *Communications of the ACM*, 45(1):36–38, 2002.
- [72] Ravi Ramamoorthi y Pat Hanrahan. A signal-processing framework for inverse rendering. En *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, páginas 117–128. ACM Press, 2001.

- [73] Richard Rickitt. *Special effects: the history and technique*. Billboard, 2000.
- [74] Charles Rothwell, Andrew Zisserman, Constantinos Marinos, David Forsyth, y Joseph L. Mundy. Relative motion and pose from arbitrary plane curves. *Image and Vision Computing*, 10(4):250–262, May 1992.
- [75] Imari Sato, Yoichi Sato, y Katsushi Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):1–12, enero/marzo 1999.
- [76] Heung-Yeung Shum y Sing Bing Kang. A Review of Image-based Rendering Techniques. En *IEEE/SPIE Visual Communications and Image Processing (VCIP)*, páginas 2–13. 2000.
- [77] Andrei State, Gentaro Hirota, David T. Chen, William F. Garrett, y Mark A. Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Computer Graphics*, 30(Annual Conference Series (ACM SIGGRAPH '96 Proceedings)):429–438, 1996.
- [78] T. Tommasini, A. Fusiello, E. Trucco, y V. Roberto. Making good features to track better. En *Proceedings IEEE Computer Society Conference on Computer Vision Pattern Recognition*, páginas 145–149. 1998.
- [79] Emanuele Trucco y Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.
- [80] Roger Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Trans. Robotics and Automation*, 3(4):323–344, 1987.
- [81] Gregory J. Ward. Measuring and modeling anisotropic reflection. En *Computer Graphics (SIGGRAPH '92 Proceedings)*, tomo 26(2), páginas 265–272. 1992.
- [82] Gregory J. Ward. The RADIANCE lighting simulation and rendering system. *Computer Graphics*, 28(Annual Conference Series):459–472, julio 1994.
- [83] Greg Welch y Gary Bishop. An introduction to the Kalman filter. SIGGRAPH Tutorial, 2001.
- [84] Tim Wittenburg. *Photo-based 3D graphics in C++: compositing, warping, morphing and other digital effects*. John Wiley and Sons, 1995.
- [85] Yizhou Yu, Paul Debevec, Jitendra Malik, y Tim Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. En *Proceedings of SIGGRAPH 99*, Computer Graphics Proceedings, Annual Conference Series, páginas 215–224. 1999.
- [86] Yizhou Yu y Jitendra Malik. Recovering photometric properties of architectural scenes from photographs. En *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, páginas 207–218. 1998.
- [87] Xiang Zhang, Stephan Frönz, y Nassir Navab. Visual marker detection and decoding in AR systems: A comparative study. En *International Symposium on Mixed and Augmented Reality*, páginas 97–106. 2002.
- [88] Zhengyou Zhang. Image-Based Geometrically-Correct Photorealistic Scene/Object Model (IBPhM): A Review. En *Asian Conference on Computer Vision (ACCV)*. 1998.

- [89] Zhengyou Zhang. A flexible new technique for camera calibration. Informe Técnico MSR-TR98-71, Microsoft Corp., 2000. <http://research.microsoft.com/~zhang>.

Las referencias on-line se encuentran a continuación

Bibliografía *on-line*

- [90] ARToolKit. <http://www.hitl.washington.edu/artoolkit>.
- [91] BBC. Walking with dinosaurs. <http://www.bbc.co.uk/dinosaurs/index.shtml>.
- [92] Shel Brannan. How the First-Down Line Works. <http://www.howstuffworks.com/first-down-line.htm>.
- [93] The Computer Vision Homepage. <http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/vision.html>.
- [94] Paul Debevec. <http://www.debevec.org>.
- [95] Paul Debevec. The story of reflection mapping. <http://www.debevec.org/ReflectionMapping>.
- [96] Human Interface Technology Laboratory. University of Washington. <http://www.hitl.washington.edu>.
- [97] ImageMagick. <http://www.imagemagick.org>.
- [98] The Internet Movie Database. <http://spanish.imdb.com>.
- [99] Intel. Open Source Computer Vision Library. <http://www.intel.com/research/mrl/research/opencv>.
- [100] Rushworth M. Kidder. CBS News: Fiddling with Visual Facts? <http://www.globalethics.org/members/commentary1-17-00.html>.
- [101] Minpack. <http://www.netlib.org/minpack>.
- [102] Nova Online. Special Effects. Titanic and Beyond, 2001. <http://www.pbs.org/wgbh/nova/specialfx2>.
- [103] OpenGL. <http://www.opengl.org>.
- [104] Princeton Video Image. <http://www.pvimage.com>.
- [105] RADIANCE, <http://radsite.lbl.gov/radiance>.
- [106] SporTVision. <http://www.sportvision.com>.
- [107] Spherical Solutions. Star Wars. <http://www.starwars.com/episode-i/feature/19991025/index.html>.
- [108] Walt Disney Company. Página web de la película Dinosaur. <http://disney.go.com/disneyvideos/dinosaur>.

Índice alfabético

- ángulo de visión, 26, 130, 132
- ARToolKit, 106–107
- BRDF, 15, 76–77, 100–101
- calibración, 3, 4, 10, 12–13, 31–35, 50, 52, 66–68, 105, 117–127, 133–137, 139
 - autocalibración, 32
 - captura de imágenes, 54–56
 - fotogramétrica, 32
- cámara
 - componentes, 24–29
 - fotográfica, 23–29
 - modelo matemático, 29–32, 118–119
 - sintética, 2, 23, 124, 130, 140
- campo de luz, 1, 142
- centro óptico, 25–26
- cine, 4, 13–16, 139
- cónicas, 11–12, 107, 112–113, 135
 - centro proyectado, 120–121, 136
 - proyección, 115–116
- consistencia
 - de iluminación, 2, 8, 47
 - geométrica, 2, 8, 10, 47
 - temporal, 2
- correspondencia, 12, 107, 134
- cross ratio*, véase razón cruzada
- curva característica, 38, 39
- decorados virtuales, 22
- distancia focal, 25, 26, 30, 33, 68, 132
- distorsión, 31, 68, 117
 - corrección de, 69
- efectos visuales, 4, 13–16, 133, 139
 - bullet time*, 15
 - control de movimiento, 14
 - glass shot*, 13
 - matte shot*, 13
 - pantalla azul, 4, 14, 16, 21
 - paro de imagen, 13
- eje óptico, 25, 30, 118, 125
- escena
 - lejana, 48, 79, 85–90, 96
 - local, 48–50, 70–74
- ética, 23
- factor de forma, 113
- geometría proyectiva, 11, 29
- HDR Shop, 43, 79–85
- homografía, 34, 73, 133, 136
- iluminación, 8, 9, 10, 92
 - directa, 9, 97
 - indirecta, 9, 96
- Image based rendering*, véase visualización basada en imagen
- imagen de alto rango dinámico, 8–10, 36–40, 60, 63, 79
- inpainting*, 129
- interfaces de usuario, 5, 133, 140
- invariante, 11, 12, 113–115, 120–121, 129, 140
- inverse rendering*, 10
- irradiancia, 36–39
- localización, 2, 5, 11, 24, 105, 140
 - diseño, 109
 - sistemas de, 2, 11, 13
- luminancia, 8, 36
- mapa de entorno, 1, 8–10, 40–43, 49, 92, 94
 - captura, 57–66
 - construcción, 79–92
- marcador, 3, 11, 106–107, 140
 - borrado del, 109, 129–130, 140
 - circular, 11–13, 107–109
 - de calibración, 53, 66–67

- mkhdr, 60, 62
- modelado basado en imagen, 8
- Open Source, 44, 106
- OpenCV, 32, 52, 66–69, 111
 - cvCalibrateCamera, 67
 - cvFindChessBoardCornerGuesses, 67
 - cvFindCornerSubPix, 67
 - cvUndistortOnce, 69
- OpenGL, 130
 - glDrawPixels, 130
 - gluLookAt, 132
 - gluPerspective, 130
- panorama, 79
- parámetros
 - extrínsecos, 30, 33, 50, 67, 72–74, 121–124, 130–132
 - intrínsecos, 30, 33, 67, 130–132
- plano focal, 29, 54
- postproceso, 132
 - borrado del marcador, 129–130
 - filtrado, 127–129
 - integración, 130–133
- preproceso
 - binarización, 111–112
 - detección del marcador, 112–117
 - recortado, 111
- profundidad de campo, 27–29
- publicidad, 4, 21–23
- punto principal, 30, 33, 68, 118, 119, 125–126
- puntos circulares, 12, 134–137
- RADIANCE, 44, 63, 70, 85–99
 - definición de escenas, 74–76
 - definición de materiales, 77–79
 - getbbox, 98
 - mkillum, 97
 - oconv, 91
 - rad, 94–96, 97
 - rpict, 91, 93
 - rview, 91, 95
- radiancia, 36–40, 63, 76, 97
- radiosidad, 9, 48
- rango dinámico, 36
- razón cruzada, 114, 120–121, 129
- realidad aumentada, 4, 10, 16–21, 23, 106–107, 112, 137, 139
- realidad virtual, 4
- reciprocidad, 39
- reflection mapping*, 40
- registro automático, 64
- rendering* diferencial, 50, 77, 99–102
- robustez, 12, 32, 107, 126–127
- saturación, 36–39
- sesión fotográfica, 50, 51–66, 88
- sistema de coordenadas, 67–68, 71, 89, 115, 118, 119–124, 127
- televisión, 4, 13, 16, 21–23, 139
- tiempo real, 2, 4, 22, 37, 105, 140
- tracking*, véase localización
- transformación lineal directa, 133
- visión por computador, 1, 2, 9, 24, 29, 31, 47, 107, 109, 129, 139
- visualización basada en imagen, 1, 2, 8
- zoom, 30, 54, 57, 59, 124, 132