



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

Renderizado transitorio en el dominio de  
frecuencias

Frequency-domain transient rendering

Autor

Darío Marcos Casalé

Director

Julio Marco Murria

ESCUELA DE INGENIERÍA Y ARQUITECTURA  
2023



## AGRADECIMIENTOS

Agradezco en primer lugar a Julio Marco por toda su ayuda y por guiarme durante el desarrollo del proyecto y a Diego Royo que, sin ser director del trabajo, me ayudó a comprender la arquitectura de Mitsuba 2. También me gustaría dar las gracias a mis amigos y familia por su interés y por escuchar mis “mini-charlas” improvisadas sobre renderizado transitorio en dominio de frecuencias. Por último, doy las gracias a mi madre por ayudarme y apoyarme incondicionalmente a lo largo del proyecto y mostrar su interés en él y se lo dedico especialmente mi padre que, como él decía con sus “matemáticas del cariño” me ha ayudado a seguir adelante a pesar de las adversidades hasta el final.



# Renderizado transitorio en el dominio de frecuencias

## RESUMEN

Los algoritmos de renderizado transitorio son una serie de técnicas que permiten simular el transporte de luz en una escena. A diferencia de las técnicas clásicas de renderizado, que asumen que la velocidad de la luz es infinita, las técnicas de renderizado transitorio eliminan esta restricción. Esto permite visualizar la luz a su paso por el espacio, así como la simulación de fenómenos físicos dependientes del tiempo de propagación de la luz (por ejemplo, la fluorescencia).

El renderizado transitorio es de gran importancia para el prototipado y análisis de técnicas de imagen computacional, en concreto de imagen transitoria. Estas técnicas permiten extraer información de las escenas, analizando la luz a escalas temporales comparables con su velocidad. Existen algoritmos de imagen transitoria que operan directamente en el dominio temporal, aunque en ocasiones es más conveniente trabajar en el dominio de frecuencias. En particular, merecen especial atención las técnicas basadas en campos de fasores, que son una representación compacta de una señal sinusoidal, y que se han utilizado en técnicas de imagen computacional, tales como la obtención de la profundidad de una escena o la reconstrucción de escenas que no se encuentran en la línea de visión del observador.

Los algoritmos de renderizado transitorio operan actualmente en el dominio temporal, pero las técnicas de imagen computacional basadas en frecuencias requieren aplicar una transformación de la información espacio-temporal a este dominio. La alta dimensionalidad de la información resuelta en tiempo supone unos costes de almacenamiento y procesado que llegan a ser inviables, y además, estos algoritmos trabajan generalmente con un subconjunto de frecuencias, por lo que gran parte de la información almacenada no se llega a usar.

El objetivo de este trabajo es desarrollar un algoritmo de renderizado transitorio que opere internamente en el dominio de frecuencias en vez del dominio temporal, de manera que la transformación al espacio de frecuencias se aplique durante el proceso de renderizado. Esto permite elegir el subconjunto de frecuencias a procesar a priori, de manera que se computen y almacenen únicamente las frecuencias relevantes para cada técnica de imagen computacional. Se ha estudiado el impacto en tiempo y memoria de la simulación

con el nuevo algoritmo, y se ha llevado a cabo una validación y comparativa de los resultados con las técnicas tradicionales. Finalmente, se ha analizado la potencial aplicación de los resultados obtenidos en dominio de frecuencias mediante el nuevo algoritmo en técnicas de imagen transitoria.

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Motivación y objetivos . . . . .	3
1.3. Metodología . . . . .	3
1.4. Organización del documento . . . . .	4
<b>2. Trabajo relacionado</b>	<b>7</b>
2.1. Renderizado transitorio . . . . .	7
2.2. Imagen transitoria . . . . .	8
<b>3. Fundamentos teóricos</b>	<b>9</b>
3.1. Renderizado y modelado del transporte de luz . . . . .	9
3.2. <i>Path integral</i> estacionaria . . . . .	11
3.3. <i>Path integral</i> transitoria . . . . .	13
3.3.1. Captura de la información espacio-temporal . . . . .	14
<b>4. Renderizado transitorio en el dominio de frecuencias</b>	<b>17</b>
4.1. Transformación a espacio de frecuencias . . . . .	17
4.2. <i>Path integral</i> en espacio de frecuencias . . . . .	19
<b>5. Estudio del algoritmo en técnicas de imagen computacional</b>	<b>23</b>
5.1. Captura del tiempo de vuelo . . . . .	23
5.2. Reconstrucción de profundidad . . . . .	24
<b>6. Resultados</b>	<b>27</b>
6.1. Renderizado transitorio en espacio de frecuencias . . . . .	27
6.1.1. Estudio cualitativo . . . . .	28
6.1.2. Impacto en memoria y tiempo de ejecución . . . . .	34
6.2. Aplicación del algoritmo para reconstrucción de profundidad . . . . .	35
<b>7. Conclusiones</b>	<b>39</b>

<b>8. Bibliografía</b>	<b>41</b>
<b>Lista de Figuras</b>	<b>45</b>
<b>Anexos</b>	<b>47</b>
<b>A. Planificación del trabajo</b>	<b>49</b>
<b>B. Resultados adicionales</b>	<b>51</b>

# Capítulo 1

## Introducción

### 1.1. Contexto

La informática gráfica es una disciplina cuyo objeto principal de estudio es obtener una representación visual de una escena sintética, así como obtener información de la misma. En particular, el renderizado basado en física estudia el transporte de la luz y define una serie de modelos para representar la geometría, fuentes de luz, materiales, modelos de cámara, etc. Esto permite simularlo de manera precisa de acuerdo a las leyes de la física, lo que posibilita la generación de imágenes realistas como si se tratara de una cámara real. Dado que la luz tiene una velocidad muy alta en comparación con la resolución temporal de las cámaras convencionales, en casi todas las técnicas de renderizado se suele asumir que la velocidad de la luz es infinita, lo que permite asumir que la propagación de la luz por la escena es instantánea.

Por otro lado, en ocasiones, el objeto de estudio es capturar imágenes a una resolución temporal ultrarrápida. Los primeros pasos en este ámbito fueron dados por el célebre fotógrafo y profesor del MIT Harold Edgerton, que consiguió capturar una fotografía de una bala atravesando una manzana a unos 850 m/s, con un tiempo de exposición de en torno a 5 millonésimas de segundo [1], en efecto “parando el tiempo”. Cincuenta años más tarde, Velten et. al (2013) proponen la técnica de **femto-fotografía** [2], que permite capturar escenas con tiempos de exposición comparables a la velocidad de la luz, permitiendo visualizar la propagación de la luz a su paso por la escena.

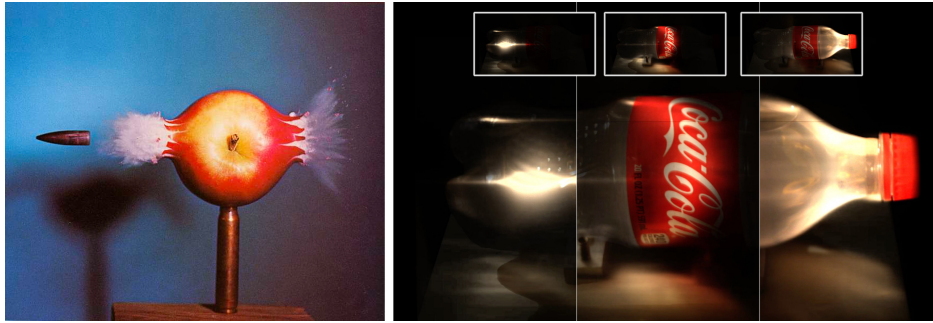


Figura 1.1: A la izquierda, “Bullet through Apple”, de Harold Edgerton (Fuente: MIT Museum). A la derecha, 3 fotogramas de una captura obtenida mediante femto-fotografía Fuente: Jarabo et al. (2014) [3]

Esta técnica abrió las puertas a un nuevo campo de estudio de la informática gráfica denominado **renderizado transitorio**, con el objetivo de simular y analizar el comportamiento de la propagación de la luz a través de una escena. La diferencia fundamental se encuentra en la eliminación de la restricción de que la propagación de la luz es instantánea, por lo que se puede simular el transporte de luz a una resolución temporal ultrarrápida. Esto permite estudiar la estructura de la información espacio-temporal debido a los retardos producidos por los caminos ópticos de la luz o las propiedades de los materiales como la fluorescencia, los materiales refractivos, etc. que habitualmente se pierden en una imagen estacionaria convencional.

Existen algoritmos que operan directamente en el dominio temporal, pero también son de interés aquellas técnicas que operan en el dominio de frecuencias, ya que en algunos casos es más conveniente operar en dicho espacio, y además existen dispositivos que capturan la luz en espacio de frecuencias. Estas técnicas se enfocan en la reconstrucción de ciertas características de la escena, como por ejemplo la profundidad de los objetos. Los sensores utilizados para la captura de esta información en el espacio de frecuencias son distintas a las cámaras convencionales, pero algunos de ellos están presentes en muchos dispositivos de uso cotidiano. Por ejemplo, se pueden encontrar cámaras de tiempo de vuelo (*Time of Flight*) en los *smartphones* que usamos en nuestro día a día, o en dispositivos de ocio, como la Kinect versión 2 de Microsoft[4], siendo ésta uno de los controladores de juego más vendidos de la historia. Por otro lado, existe otro conjunto de algoritmos que operan también en el espacio de frecuencias, cuyo objetivo es la reconstrucción de escenas que no se encuentran en la línea de visión del observador, las denominadas como *non-line-of-sight imaging* (NLOS). Las aplicaciones de este tipo de técnicas abarcan un amplio abanico de disciplinas como la visión por computador, la robótica, imagen médica y muchos otros, por lo que es un campo de estudio fundamental.

## 1.2. Motivación y objetivos

La motivación principal para desarrollar esta nueva técnica reside en los métodos actuales para obtener la información en el dominio de frecuencias, ya que requieren calcular y almacenar la información espacio-temporal por completo. La transformación a espacio de frecuencias se realiza a posteriori, y genera todo el espectro, a pesar de que los algoritmos que operan en este dominio no requieren utilizarlo por completo. Esto implica que mucha de la información no es útil y, en ocasiones, puede llegar a ser un problema, ya que el volumen de datos puede ser intratable. Por ejemplo, a la hora de estimar la profundidad de una escena sólo es necesaria una única frecuencia para reconstruir la profundidad, aunque con las técnicas clásicas es necesario almacenar toda la información espacio-temporal y aplicar la transformación a posteriori para seleccionar dicha frecuencia. También ocurre en la reconstrucción de escenas que no se encuentran en la línea directa de observación, donde la combinación de varias medidas realizadas en un plano bidimensional de una escena en 3 dimensiones puede generar un volumen 5-dimensional[5] de datos, que puede llegar a ser imposible de almacenar o tratar en la práctica. La principal ventaja del nuevo enfoque frente a la transformación a posteriori de la información espacio-temporal es que no requiere almacenar por completo el resultado, por lo que se puede ahorrar una gran cantidad de espacio en memoria. Además, la posibilidad de elegir el conjunto de frecuencias a devolver permite un mayor grado de flexibilidad a la hora de aplicar los resultados en algoritmos de imagen computacional, dado que cada uno de ellos opera sobre un conjunto de frecuencias distinto.

El objetivo de este trabajo es desarrollar un algoritmo de renderizado transitorio que opere directamente en el espacio de frecuencias, con el fin de estudiar las diferencias de precisión con respecto a los algoritmos de renderizado que operan en dominio temporal, así como reducir los costes de almacenamiento y cómputo. También se busca analizar el efecto del nuevo algoritmo en técnicas de imagen computacional, en particular a la hora de estimar la profundidad de la escena.

## 1.3. Metodología

La metodología seguida a lo largo del proyecto se divide en las siguientes fases:

1. En primer lugar, se ha hecho un estudio del estado del arte en técnicas de renderizado estacionario y transitorio, así como técnicas de imagen transitoria e imagen computacional.
2. Posteriormente, se ha integrado el algoritmo de renderizado transitorio en espacio de frecuencias en un renderizador transitorio. Para ello se han seguido los siguientes

pasos:

- Primero se llevó a cabo el estudio y comprensión de renderizadores existentes que implementan técnicas de renderizado transitorio.
  - Posteriormente, se ha formulado el transporte de luz en espacio de frecuencias y se han analizado las modificaciones necesarias para integrar el algoritmo en el renderizador.
  - Finalmente, se han desarrollado e implementado las modificaciones necesarias en la herramienta.
3. Después, se ha llevado a cabo una evaluación, validación y comparativa de la información generada por el nuevo algoritmo frente a la información espacio-temporal transformada a posteriori:
- En primer lugar, se han llevado a cabo estudios cualitativos para comprobar si el resultado obtenido se acerca al esperado.
  - Seguidamente, se ha analizado el impacto en el coste temporal y de almacenamiento en memoria con distintas configuraciones.
4. Finalmente, se ha analizado la utilización y los beneficios del algoritmo desarrollado en técnicas de imagen transitoria. En particular, se ha estudiado el resultado y las ventajas que ofrece a la hora de reconstruir la profundidad de una escena, usando técnicas basadas en modelos de cámaras de tiempo de vuelo.

El algoritmo se ha integrado dentro de una versión de Mitsuba 2 extendida para soportar renderizado transitorio [6], escrito en C++. Las escenas con las que se han llevado a cabo las pruebas también forman parte de los recursos de la herramienta Mitsuba 2. Finalmente, para la visualización, validación de los resultados y reconstrucción de la profundidad se han desarrollado una serie de scripts en Python.

## 1.4. Organización del documento

El contenido de este documento se ha dividido en los siguientes capítulos:

- En primer lugar, en el capítulo 2 se describe el estado del arte de los algoritmos de imagen transitoria.
- Después, en el capítulo 3 se introduce el marco teórico de los algoritmos de renderizado transitorio.
- Posteriormente, en el capítulo 4 se propone el nuevo algoritmo que opera en el espacio de frecuencias y el marco teórico sobre el que se sustenta.

- Seguidamente, en el capítulo 5 se estudia la aplicación del nuevo algoritmo en técnicas de imagen computacional. En particular, se analiza el algoritmo de estimación de profundidad de una escena.
- Finalmente, en el capítulo 6 se analizan los resultados obtenidos con el nuevo algoritmo y se lleva a cabo una comparativa con las técnicas ya existentes. También se estudian los resultados de utilizar el algoritmo a la hora de reconstruir la profundidad de una escena.



## Capítulo 2

# Trabajo relacionado

El nuevo algoritmo de renderizado transitorio en el dominio de frecuencias está estrechamente ligado con dos líneas de investigación de la informática gráfica: el renderizado transitorio y la imagen transitoria. En esta sección se estudia el estado del arte de estos dos ámbitos y la motivación de desarrollar el nuevo algoritmo.

### 2.1. Renderizado transitorio

El renderizado transitorio es una técnica que simula el proceso de adquisición de imágenes de los dispositivos de captura de imagen transitoria. Es útil para analizar y validar algoritmos de imagen computacional, como paso previo a su aplicación en dispositivos de imagen transitoria. Los primeros pasos en la implementación de un renderizador transitorio fueron dados por Smith et al. [7] y, posteriormente, Jarabo et al. [3] proponen un modelo formal para extender el trazado de rayos al dominio temporal. Gracias a este marco de trabajo, se han desarrollado extensiones de renderizadores transitorios ya existentes que permiten la simulación de algoritmos de imagen transitoria, tales como la estimación de profundidad simulando sensores de tiempo de vuelo [8] o la reconstrucción de escenas ocultas [6]. Por último, cabe destacar también la extensión de renderizadores diferenciables al dominio transitorio [9], que permiten resolver problemas de optimización (en este caso también en el dominio temporal) y son de gran utilidad en el campo de la imagen computacional. El nuevo algoritmo es también una extensión de las técnicas de renderizado transitorio ya existentes, con la diferencia de que opera directamente en el dominio de frecuencias.

Los renderizadores transitorios no sólo sirven para visualizar la luz en movimiento, sino que permiten generar información que sirve como referencia para desarrollar nuevos modelos de transporte de luz, así como validar el resultado de técnicas de imagen transitoria [10]. La generación de información de referencia puede incluso servir como datos de entrenamiento para aprendizaje profundo a la hora de mejorar la estimación de propiedades de la escena [11], por lo que el desarrollo de renderizadores transitorios es una herramienta

muy útil para el desarrollo de ciertas técnicas de imagen computacional.

## 2.2. Imagen transitoria

Las técnicas de imagen transitoria permiten capturar el transporte de luz en el dominio temporal y permiten extraer información de las escenas que sería más complicado extraer mediante una cámara convencional. Una de las características que se pueden reconstruir a partir de la información capturada en el dominio de frecuencias es la geometría de la escena, donde la recuperación de la profundidad de la escena es una de las aplicaciones más relevantes en el campo de la imagen transitoria. Existen técnicas que operan directamente en el dominio temporal, como las basadas en las técnicas de femto-fotografía propuestas por Velten et al. [2], pero su aplicación es muy limitada debido a su alto coste y la sensibilidad del método de captura, por lo que las técnicas que operan en el espacio de frecuencias son las más utilizadas. Existen modelos de captura que operan en este dominio, como los **sistemas de ondas continuas moduladas en amplitud**, o AMCW. Schwarte et al. [12] desarrollaron un método de captura AMCW utilizando unos dispositivos llamados *Photonic Mixer Devices*, que siguen siendo utilizados para capturar información en estado transitorio sin incurrir en costes demasiado elevados[13]. En general, estas técnicas asumen que la luz recibida sólo proviene de un punto de la escena, pero en la práctica no es así, por lo que se obtienen medidas erróneas. Este problema se denomina **interferencia multicamino**, o *multipath interference* (MPI) y es uno de los problemas más importantes que las técnicas de imagen transitoria buscan resolver. Existen enfoques que corrigen el efecto de la MPI mediante optimización [14] e incluso técnicas que corrigen este problema mediante aprendizaje profundo [11]. En particular, merece especial atención el modelo de **campos de fasores**, propuesto por Gupta et al. [15], que permite simplificar el transporte de luz en el dominio de frecuencias representando la luz en forma de fasores, que permiten modelar las señales utilizadas como un número complejo de una forma compacta. También existen técnicas que permiten reconstruir objetos que no se encuentran en la línea directa del observador debido a la presencia de elementos oclusores o medios turbios que dificultan la visibilidad [16], lo que se denomina *non-line-of-sight imaging*. Muchos algoritmos para reconstruir escenas ocultas operan en el dominio temporal [17, 18], aunque también existen técnicas que operan en el dominio de frecuencias [19, 20], basado en el marco de los campos de fasores propuesto por Gupta et al. [15]. Por ello, el desarrollo de un renderizador que opera en este dominio puede impulsar el desarrollo de nuevas técnicas de imagen transitoria y facilitar su validación.

## Capítulo 3

# Fundamentos teóricos

En este capítulo se introducirán las bases de la simulación del transporte de luz. En primer lugar, se definirán los conceptos necesarios para modelar el transporte de luz en una escena. Posteriormente, se planteará una descripción formal del transporte de luz en estado estacionario y seguidamente se extenderá su formulación para el dominio temporal. El objetivo es establecer una base formal que permita representar de manera precisa el modelo de renderizado en estado transitorio.

### 3.1. Renderizado y modelado del transporte de luz

El renderizado de imágenes consiste en obtener una representación en forma de imagen de una escena definida en tres o más dimensiones. La tarea principal es decidir qué color se debe asignar a cada píxel, y para ello existe un amplio abanico de técnicas. La más utilizada para obtener resultados fotorrealistas es el renderizado basado en física, cuyo objetivo es modelar el transporte de luz de acuerdo a una serie de leyes que modelan el comportamiento de la luz y su interacción con la escena. Es una tarea compleja, ya que intervienen una gran variedad de factores, como la geometría de la escena, las propiedades de los materiales, el medio a través del que se propaga la luz, las propiedades de las fuentes de luz, las características de la cámara, etc.

El objetivo principal es modelar la interacción de la luz en un punto de la escena y calcular la energía percibida por el sensor. Para ello, Kajija (1986) [21] propone la **ecuación de render**:

$$L(\vec{x}, \vec{\omega}_o) = L_e(\vec{x}, \vec{\omega}_o) + \int_{S^2} L_i(\vec{x}, \vec{\omega}_i) \rho(\vec{x}, \vec{\omega}_i, \vec{\omega}_o) |\cos\theta_i| d\vec{\omega}_i. \quad (3.1)$$

Esta ecuación representa la energía que llega desde un punto de la escena como la suma de un término de **emisión** y la suma de toda la energía que recibe dicho punto desde todas las direcciones de una esfera unitaria, como se puede observar en la figura 3.1. Cada término representa lo siguiente:

- $\bar{x}$  es un punto de la escena en el espacio tridimensional.
- $\vec{\omega}_o$  es la dirección en la que llega la luz desde el punto  $\bar{x}$  y  $\vec{\omega}_i$  es cualquier dirección desde la que llega luz al punto  $\bar{x}$ , en la esfera unitaria  $S^2$  centrada en dicho punto.
- $L(\bar{x}, \vec{\omega}_o)$  es la energía que llega desde un punto de la escena en la dirección  $\vec{\omega}_o$ .
- $L_e(\bar{x}, \vec{\omega}_o)$  es la cantidad de luz que emite el objeto observado, es decir, la **emisión** del material que lo conforma.
- $L_i(\bar{x}, \vec{\omega}_i)$  es la energía que llega a un punto de la escena desde una dirección entrante  $\vec{\omega}_i$ .
- La función  $\rho(\bar{x}, \vec{\omega}_i, \vec{\omega}_o)$ , denominada **función de dispersión bidireccional** (BSDF), que define la fracción de luz incidente en un punto  $\bar{x}$  en una dirección  $\vec{\omega}_i$  que es reflejada hacia una dirección  $\vec{\omega}_o$ , y depende de las propiedades del material del que está hecho el objeto.
- El término del coseno modela la atenuación causada por la geometría de la escena, que depende del ángulo  $\theta_i$  que forma el vector normal a la superficie y la dirección entrante de la luz,  $\omega_i$ .

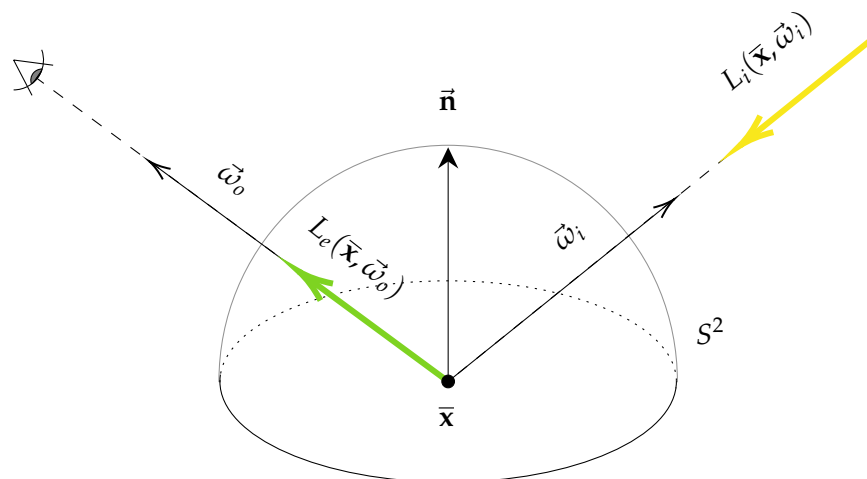


Figura 3.1: Representación gráfica de la interacción de la luz en un punto de la escena.

Existen varias técnicas que resuelven la ecuación de render, aunque el algoritmo más utilizado para simularlo es el **trazado de rayos**, inspirado en la idea de que la luz se propaga en línea recta desde la fuente de luz, interactuando con los elementos de la escena y rebotando hasta llegar al sensor. Cada trayectoria que sigue la luz desde la fuente hasta el sensor se denomina **camino** y el objetivo es simular todos los caminos que llegan a cada píxel para asignarle un color.

Dado que la velocidad de la luz es muy alta en comparación con el tiempo de exposición de las cámaras convencionales, en la práctica se puede asumir que la propagación de la luz en la escena es instantánea, de manera que se está llevando a cabo el renderizado en **estado estacionario**, donde la luz ha llegado a un equilibrio global en la escena. En este tipo de técnicas, se pierde la información temporal de los caminos trazados a cambio de simplificar el modelo de propagación de la luz, por lo que no es posible analizar la estructura temporal del transporte de luz. En cambio, si se levanta la restricción de que la propagación de la luz es instantánea, es posible analizar esta estructura, lo que da lugar a las técnicas de **renderizado transitorio**. Estos algoritmos no sólo permiten observar la luz en movimiento como si se tratase de una cámara ultrarrápida, sino que además permite simular ciertos efectos que sólo se pueden observar en estado transitorio, como los retardos de dispersión de la luz debido a las propiedades de los materiales, así como simular el comportamiento de las fuentes de luz o del sensor.

### 3.2. *Path integral estacionaria*

Una de las formulaciones más utilizadas para el modelado del transporte de luz estacionario es la **integral de caminos**, o *path integral* (Veach, 1998) [22], que reformula la ecuación de render para expresar la energía percibida en función de todos los caminos posibles que contribuyen al píxel en cualquier dirección  $\vec{\omega}_o$ . El algoritmo de trazado de rayos consiste en trazar una serie de caminos que provienen de una fuente de luz, rebotan a través de la escena y llegan al sensor. La intensidad  $I$  percibida en un píxel es el resultado de sumar la contribución de todos los caminos que traza la luz a lo largo de la escena hasta llegar a un píxel de la cámara:

$$I = \int_{\Omega} f(\bar{\mathbf{x}}) d\mu(\bar{\mathbf{x}}), \quad (3.2)$$

donde  $\bar{\mathbf{x}} = \mathbf{x}_0 \dots \mathbf{x}_k$  denota las coordenadas espaciales de los vértices de un camino y  $f$  representa la contribución de cada camino:

$$f(\bar{\mathbf{x}}) = L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1) \mathfrak{T}(\bar{\mathbf{x}}) W_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k), \quad (3.3)$$

donde  $L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1)$  denota la radiancia emitida por la fuente de luz en  $\mathbf{x}_0$  y  $W_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k)$  indica la importancia del sensor, que modela el comportamiento de la cámara para cada píxel, y depende de su posición y del ángulo con el que incide la luz en el sensor. En cierto modo, define la sensibilidad de un píxel a un camino concreto. Por último, se define el **rendimiento de un camino**, o *path throughput*  $\mathfrak{T}$  como la fracción de radiancia que proviene de la fuente de luz y llega a la cámara:

$$\mathfrak{T}(\bar{\mathbf{x}}) = \left[ \prod_{i=1}^{k-1} \rho(\mathbf{x}_i) \right] \left[ \prod_{i=0}^{k-1} G(\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}) V(\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}) \right], \quad (3.4)$$

donde  $G$  es un término que modela la atenuación de la energía del camino debido a la distancia y la orientación entre superficies,  $V$  es un término binario cuyo valor es 1 sólo si los vértices son mutuamente visibles y  $\rho$  expresa la BSDF de la ecuación de render en función del camino de luz.

Como se puede ver en la Figura 3.2, los caminos de luz que llegan al sensor pueden tener un número arbitrario de rebotes. Aquellos caminos que rebotan una única vez en la escena entre la fuente de luz y la cámara contribuyen a la **luz directa** de la escena, mientras que los que rebotan más de una vez son caminos de **luz indirecta**.

Cabe destacar que la *path integral* no tiene una solución analítica debido a la complejidad de la escena, por lo que se debe aproximar el resultado mediante métodos numéricos, donde la familia de métodos de Monte Carlo es la más utilizada para evaluar el resultado de la integral. Estos métodos consisten en muestrear aleatoriamente  $N$  valores del dominio de integración de acuerdo a una distribución de probabilidad  $p$  y calculando la siguiente estimación:

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^N \frac{f(\bar{\mathbf{x}}_i)}{p(\bar{\mathbf{x}}_i)}$$

En este caso, el dominio de integración es el espacio de todos los posibles caminos que puede trazar la luz desde una fuente de luz a la cámara, por lo que cada muestra es en efecto un camino de luz que contribuye a la energía percibida en un píxel.

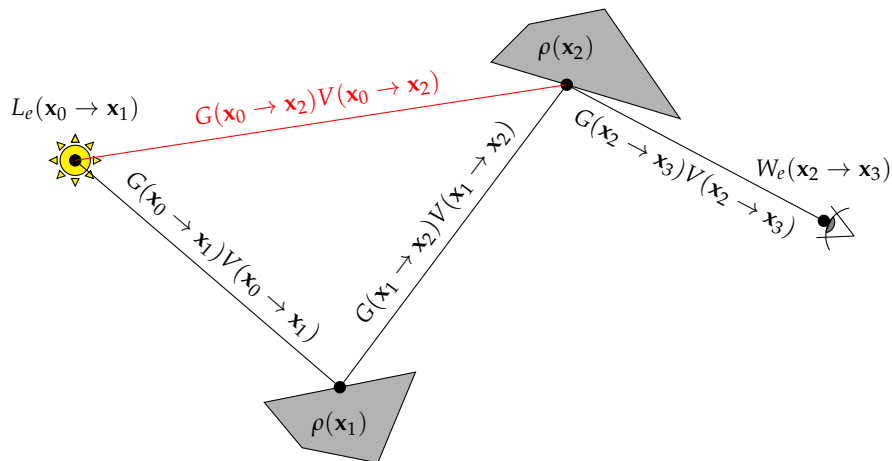


Figura 3.2: Esquema de la contribución de un camino de luz hacia la cámara.

### 3.3. *Path integral* transitoria

En las técnicas de render estacionario, se asume que la velocidad de la luz es infinita, por lo que no hay una dependencia temporal en el resultado. Esto impide modelar la estructura temporal del transporte de luz, por lo que si se elimina esta restricción, es posible modelar la propagación de la luz en el dominio transitorio, considerando el tiempo que tarda la luz en propagarse por la escena, así como los retardos de dispersión en cada punto de la misma debido a una serie de fenómenos físicos. Además, al añadir la dependencia temporal, se puede simular también el comportamiento de las fuentes de luz y el sensor en el tiempo. Por ejemplo, es posible modelar un pulso corto de luz o simular el método de captura de las cámaras convencionales, estableciendo un tiempo de exposición. La información temporal es muy útil, ya que no sólo permite visualizar la propagación de la luz en movimiento, sino que se puede analizar dicha información para obtener información de la escena que en general no se puede obtener mediante métodos estacionarios.

Partiendo de la base propuesta por Veach [22], Jarabo et al. [3] proponen un marco de trabajo para extender la formulación de la *path integral* al estado transitorio. La expresión que define la intensidad  $I$  percibida en un píxel de la imagen es una extensión de la ecuación 3.2, integrando a lo largo del espacio  $\Delta T$  de retardos temporales a lo largo de los vértices de cada camino  $\bar{\mathbf{x}} \in \Omega$ :

$$I = \int_{\Omega} \int_{\Delta T} f(\bar{\mathbf{x}}, \bar{\Delta \mathbf{t}}) d\mu(\bar{\Delta \mathbf{t}}) d\mu(\bar{\mathbf{x}}), \quad (3.5)$$

donde  $\bar{\Delta \mathbf{t}} = \Delta t_0 \dots \Delta t_k$  denota la secuencia de retrasos temporales asociados a cada vértice del camino. La función de contribución transitoria también es una extensión de la ecuación 3.3 al dominio temporal:

$$f(\bar{\mathbf{x}}, \bar{\Delta \mathbf{t}}) = L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1, \Delta t_0) \mathfrak{F}(\bar{\mathbf{x}}, \bar{\Delta \mathbf{t}}) W_e(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k, t_k), \quad (3.6)$$

donde  $t_k$  es el instante en el que llega el camino de luz al sensor, que es el resultado de acumular los tiempos de vuelo entre vértices y los retardos de dispersión en cada vértice. Cabe destacar que la emisión de la fuente de luz puede variar, lo que permite modelar el comportamiento del emisor (por ejemplo, un pulso corto de luz o una señal modulada a cierta frecuencia). También se incluye la dependencia temporal de la importancia del sensor  $W_e$ , lo que permite establecer un intervalo temporal de captura, similar al tiempo de exposición de las cámaras convencionales.

Por último, se extiende también la ecuación 3.4 del *path throughput* al dominio temporal, incluyendo el conjunto de retrasos temporales:

$$\mathfrak{T}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) = \left[ \prod_{i=1}^{k-1} \rho(\mathbf{x}_i, \Delta t_i) \right] \left[ \prod_{i=0}^{k-1} G(\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}) V(\mathbf{x}_k \rightarrow \mathbf{x}_{k+1}) \right]. \quad (3.7)$$

Es importante señalar que los términos de geometría y visibilidad no dependen del tiempo en la formulación transitoria del transporte de luz, debido a la suposición de que la escena es estacionaria (en comparación con la velocidad de la luz). Por otro lado, el término de dispersión de la luz sí que presenta una dependencia temporal, ya que se consideran posibles retardos en la emisión en cada vértice del camino debido a fenómenos físicos de dispersión. En la figura 3.3 se describe la propagación de un camino de luz incluyendo el dominio temporal, donde se puede observar el tiempo de propagación entre vértices consecutivos del camino y los retardos temporales causados por las propiedades físicas del material.

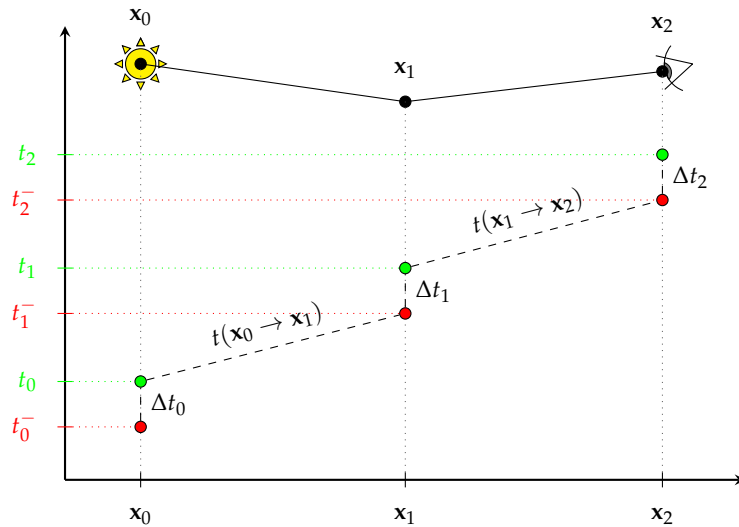


Figura 3.3: Esquema espacio-temporal de la propagación de un camino de luz. Adaptado de “A Framework for Transient Rendering”, de Jarabo et al. (2014) [3]

De nuevo, la *path integral* transitoria no tiene solución analítica, por lo que se obtiene una solución aproximada mediante métodos de integración numérica de Monte Carlo, mostrando también de manera aleatoria en el dominio temporal:

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^N \frac{f(\bar{\mathbf{x}}_i, \overline{\Delta \mathbf{t}}_i)}{p(\bar{\mathbf{x}}_i, \overline{\Delta \mathbf{t}}_i)} \quad (3.8)$$

### 3.3.1. Captura de la información espacio-temporal

La evolución de la energía percibida por el sensor es una función continua en el dominio temporal, pero no es posible guardar la información espacio-temporal a una resolución infinitamente alta, por lo que es necesario discretizar el dominio temporal y tomar una serie de imágenes consecutivas. Este procedimiento es similar al mecanismo que utilizan las

cámaras convencionales para grabar un vídeo, donde se toma una secuencia de fotografías consecutivas con un tiempo de exposición  $\tau$ . En renderizado transitorio, el procedimiento de captura es igual al de estas cámaras, pero a una resolución ultrarrápida, suficiente para apreciar el movimiento de la luz. Así, la información espacio-temporal se almacena como un bloque de imágenes  $\{I_j\}_{j \geq 0}$ , donde la franja temporal  $I_j$  representa la imagen obtenida en un periodo de captura  $[t_0 + j \cdot \tau, t_0 + (j + 1) \cdot \tau]$ , donde  $t_0$  es el tiempo a partir del que se empieza a capturar información. Esto implica que el comportamiento del sensor, que interviene en la contribución de un camino descrita en la ecuación 3.6, se modela para cada franja temporal  $j$  como una función rectangular  $W_e^{(j)}(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k, t_k)$ :

$$W_e^{(j)}(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k, t_k) = \begin{cases} 1 & t_0 + j \cdot \tau \leq t_k \leq t_0 + (j + 1) \cdot \tau \\ 0 & \text{si no} \end{cases} \quad (3.9)$$

A la hora de computar la secuencia de imágenes, cabe destacar que para cada franja temporal, el sensor descarta todas aquellas muestras que no hayan sido capturadas en el intervalo de tiempo asociado al *frame*, lo que es altamente ineficiente. Por ello, se utilizan técnicas de optimización que evitan descartar muestras, de las cuales la más utilizada es la de **reutilización de caminos**, o *path reuse* [3], donde se estiman simultáneamente todas las imágenes de la secuencia a capturar. Para ello, se determina el tiempo de vuelo total de un camino y se acumula la muestra obtenida en la franja temporal correspondiente, de manera que todos los caminos contribuyen a alguna de las franjas temporales.

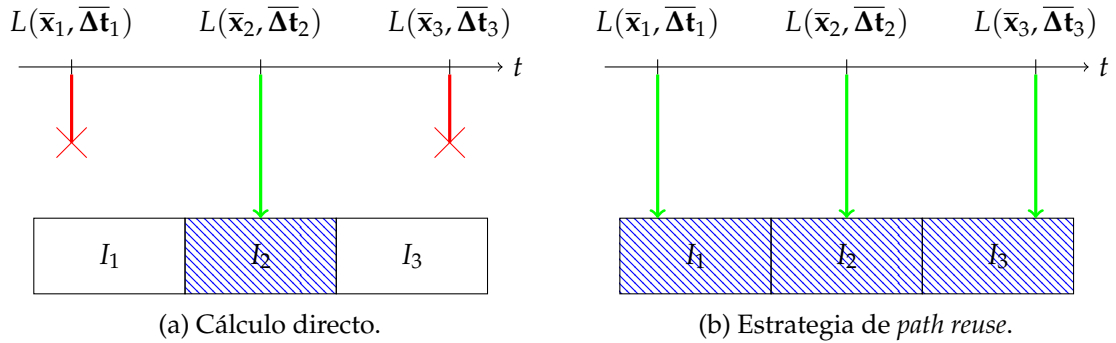


Figura 3.4: Diferencias entre un cálculo directo frente a la estrategia de *path reuse*.



## Capítulo 4

# Renderizado transitorio en el dominio de frecuencias

En esta sección, se introducirá el modelo de renderizado de transporte de luz transitorio en dominio de frecuencias. Se estudiarán las técnicas tradicionales que utilizan la información renderizada resuelta en tiempo para pasar al nuevo dominio y se analizarán las posibles desventajas que presentan. Seguidamente, se propondrá un modelo de renderizado que trabaja directamente en dominio de frecuencias y se estudiarán las ventajas frente a las técnicas tradicionales.

### 4.1. Transformación a espacio de frecuencias

Los algoritmos de imagen transitoria permiten analizar la estructura de la información espacio-temporal para extraer características de la escena. Muchas de las técnicas analizan directamente la información resuelta en tiempo, pero en ocasiones es más conveniente operar en el dominio de frecuencias. Existen muchas formas de modelar el transporte de luz en este espacio y una de las más relevantes es el modelo de **campos de fasores** [15]. Este modelo se basa en la observación de que si una escena se ilumina con una señal sinusoidal modulada a cierta frecuencia, la energía percibida en un píxel también será sinusoidal. Esto permite modelar la radiancia en cualquier punto y dirección mediante un número complejo o **fasor**, que representa dicha señal sinusoidal. Este modelo es muy útil, ya que permite simplificar el análisis de la estructura temporal del transporte de luz. Además existen técnicas de imagen computacional en espacio transitorio que utilizan esta representación de la radiancia para obtener información adicional de la escena, como las técnicas de tiempo de vuelo por correlación, o algoritmos de reconstrucción de escenas ocultas.

La representación del transporte de luz mediante campos de fasores está también estrechamente ligada con la **transformada de Fourier** de la información espacio-temporal. Esta transformación convierte una función en otra que toma valores complejos, y que representa la distribución de las frecuencias presentes en la función original. La

transformada de Fourier para una función que toma valores complejos se puede expresar mediante su forma integral:

$$\hat{f}(\xi) = \mathcal{F}\{f\}(\xi) = \int_{\mathbb{R}} f(t)e^{-2\pi i \xi t} dt, \quad (4.1)$$

donde  $\xi$  denota la frecuencia. La información temporal se almacena de forma discretizada en una serie de franjas temporales o *bins*, donde cada una de ellas se corresponde con un fotograma con un tiempo de exposición  $\tau$ , de manera que se debe calcular el análogo de la transformada de Fourier en un dominio discretizado. En lugar de transformar una función definida en un dominio continuo (en este caso, el tiempo), toma como argumento una sucesión discreta de valores (cada uno de ellos asociado a un *bin* temporal con un tiempo de referencia), y obtiene un espectro de frecuencias discreto. En este caso, los valores de la sucesión son la radiancia percibida por el píxel de cada bin temporal  $I_j$ , cuya expresión ha sido descrita en la ecuación 3.5, por lo que es posible calcular la transformada discreta de Fourier de dicha sucesión:

$$\hat{I}_k = \sum_{j=1}^{N-1} I_j \cdot e^{-2\pi i \frac{j}{N} k}, \quad k \in \{0, \dots, N-1\} \quad (4.2)$$

Es importante destacar que la transformada no dispone de información sobre el tiempo de vuelo exacto de cada muestra, sino que utiliza el tiempo asociado a cada bin temporal, lo cual es una potencial fuente de inexactitud dependiendo de la resolución temporal de los bins. En la ecuación anterior no aparece este valor, pero se obtiene implícitamente a partir de la posición del bin temporal. En otras palabras, supongamos que el tiempo de exposición de un render transitorio es  $\tau$ , por lo que el tiempo de captura asociado al  $i$ -ésimo bin temporal será  $t_0 + i \cdot \tau$ . El cálculo directo de esta operación es muy costoso, de orden cuadrático, por lo que para determinar la transformada discreta de la información espacio-temporal se utiliza el algoritmo de la transformada rápida de Fourier, o FFT (Cooley, Tukey, 1965) [23]. Su complejidad computacional es mucho menor, de orden  $O(N \log N)$ , ya que emplea una estrategia de “divide y vencerás” para calcular la transformada, generando  $N$  frecuencias uniformemente distribuidas en el intervalo  $\left[-\frac{f_s}{2}, \frac{f_s}{2}\right]^1$ , donde  $f_s$  es la **tasa de muestreo** de la señal, que en este caso equivale al inverso del tiempo de exposición.

Uno de los problemas que supone aplicar la transformada a posteriori es que se necesita generar toda la información espacio-temporal, lo que supone un coste muy elevado de almacenamiento en memoria. Además, reducir el número de bins temporales se traduce en una peor representación de la estructura temporal del transporte de luz, por lo que generalmente es necesario utilizar muchos bins temporales para obtener un buen resultado, a cambio de utilizar aún más espacio.

<sup>1</sup>Este intervalo contiene las frecuencias con las que es posible reconstruir la señal por completo a partir de sus componentes en espacio de frecuencias, de acuerdo al teorema de muestreo de Nyquist-Shannon.

## 4.2. Path integral en espacio de frecuencias

El nuevo algoritmo de renderizado transitorio en espacio de frecuencias permite elegir un conjunto  $\Xi = [\xi_{min}, \xi_{max}]$  de frecuencias, así como escoger el número de frecuencias muestreadas dentro del intervalo. De esta forma, se puede reducir significativamente el tamaño que ocupa la información en disco, y además no es necesario almacenar toda la información resuelta en tiempo ni el espectro de frecuencias completo.

La información espacio-temporal obtenida mediante los métodos existentes de renderizado transitorio se discretiza en el dominio temporal durante el proceso de captura, pero la transformada de Fourier opera con funciones continuas. De manera intuitiva, se puede observar que conforme se reduce el tiempo de exposición, la información capturada se asemeja cada vez más a una función continua, aunque se debe almacenar un mayor número de *frames*, como se puede observar en la figura 4.1, pero es imposible en la práctica. Si se pudiera almacenar una cantidad infinita de imágenes reduciendo el tiempo de exposición a un valor muy cercano a 0 (lo cual en el límite es imposible) se obtendría un resultado prácticamente continuo. Formalmente, si se considera el límite de la función de comportamiento del sensor definida en la ecuación 3.9 cuando el tiempo de exposición se acerca a 0, tiende a comportarse como una función delta:

$$\lim_{\tau \rightarrow 0^+} W_e^{(j)}(\mathbf{x}_{k-1} \rightarrow \mathbf{x}_k, t_k) = \delta(t_j - t_k) \quad (4.3)$$

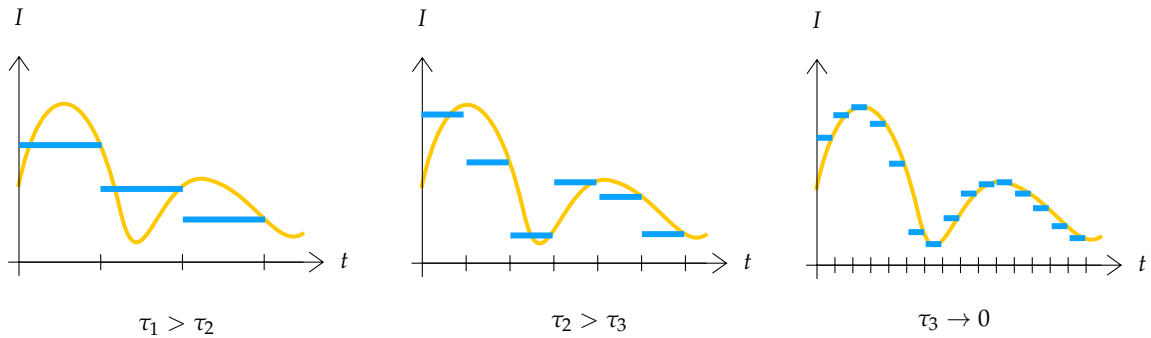


Figura 4.1: Efecto de la disminución del tiempo de exposición de cada *frame*.

Intuitivamente, la función delta toma valor 0 si su argumento es distinto de 0, y 1 en caso contrario<sup>2</sup>. De esta forma, la función de importancia del sensor sólo tomará valor 1 cuando el tiempo de vuelo del camino sea igual al instante en el que se captura la imagen, por tanto la energía percibida en un píxel en el instante  $t$  será la suma de las contribuciones de todos los caminos que llegan al sensor exactamente en el instante  $t$ . A partir de las Ecuaciones 3.5 y 3.6 podemos definir una forma continua de la path integral transitoria 3.6:

<sup>2</sup>La función delta en un sentido más riguroso es una distribución de probabilidad que se concentra en el valor 0. Se puede definir de forma implícita para cualquier función  $f$ , de manera que  $\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$

$$I(t) = \int_{\Omega} \int_{\Delta T} f(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) d\mu(\overline{\Delta \mathbf{t}}) d\mu(\bar{\mathbf{x}}) = \int_{\Omega} \int_{\Delta T} L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1, \Delta t_0) \mathfrak{T}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) \delta(t - t_k) d\mu(\overline{\Delta \mathbf{t}}) d\mu(\bar{\mathbf{x}}) \quad (4.4)$$

Finalmente, se puede aplicar la forma integral de la transformada de Fourier sobre la ecuación anterior para obtener la información resuelta en el dominio de frecuencias. Operando y aplicando el teorema de Fubini en la ecuación anterior, se puede reescribir el orden de integración para aplicar la transformada de Fourier únicamente a la función de comportamiento del sensor, de manera que se obtiene una expresión similar a la de la *path integral* transitoria, pero que opera directamente en el dominio de frecuencias:

$$\begin{aligned} \hat{I}(\xi) &= \int_{\mathbb{R}} \int_{\Omega} \int_{\Delta T} L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1, \Delta t_0) \mathfrak{T}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) \delta(t - t_k) e^{-2\pi i \xi t} d\mu(\overline{\Delta \mathbf{t}}) d\mu(\bar{\mathbf{x}}) dt \\ &= \int_{\Omega} \int_{\Delta T} L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1, \Delta t_0) \mathfrak{T}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) \left[ \int_{\mathbb{R}} \delta(t - t_k) e^{-2\pi i \xi t} dt \right] d\mu(\overline{\Delta \mathbf{t}}) d\mu(\bar{\mathbf{x}}) \\ &= \int_{\Omega} \int_{\Delta T} L_e(\mathbf{x}_0 \rightarrow \mathbf{x}_1, \Delta t_0) \mathfrak{T}(\bar{\mathbf{x}}, \overline{\Delta \mathbf{t}}) e^{-2\pi i \xi t_k} d\mu(\overline{\Delta \mathbf{t}}) d\mu(\bar{\mathbf{x}}) \end{aligned} \quad (4.5)$$

En la ecuación anterior, se puede observar que la función de importancia del sensor  $W_e$  descrita en la ecuación 3.6 es ahora una exponencial compleja. Esto implica que el sensor ahora tiene un comportamiento oscilatorio, que depende de la frecuencia medida, similar a los utilizados en las cámaras de tiempo de vuelo.

De nuevo, la aproximación de la integral se puede llevar a cabo mediante el método de integración numérica de Monte Carlo. Además, la discretización de la información en el espacio de frecuencias es trivial, ya que se puede asociar una franja a cada frecuencia  $\xi \in \Xi$ . Por último, la implementación del nuevo algoritmo permite reaprovechar la técnica de *path reuse*, ya que ahora cada muestra se debe acumular en todos los bins de frecuencias, multiplicada por el factor exponencial, por lo que todos los caminos son útiles. A continuación se incluye una descripción en alto nivel de la implementación del algoritmo de renderizado transitorio en dominio de frecuencias para un píxel:<sup>3</sup>:

---

<sup>3</sup>Para simplificar la notación, la contribución energética de la muestra  $i$  que ha llegado al sensor en el instante  $t$  se denotará  $L_i = \frac{f(\bar{\mathbf{x}}_i, \overline{\Delta \mathbf{t}}_i)}{p(\bar{\mathbf{x}}_i, \overline{\Delta \mathbf{t}}_i)}$

---

**Algorithm 1** Renderizado transitorio en espacio de frecuencias

---

```
1: procedure F-RENDER( $L = \{L_j\}_{j \geq 1}^N, \Xi = \{\xi_j\}_{j \geq 1}^{N_F}$ )
2:    $\hat{I} \leftarrow \bar{0}$  ▷ Se inicializa el valor de cada frecuencia a 0
3:   for  $j \in \{1, \dots, N\}$  do
4:     for  $k \in \{1, \dots, N_F\}$  do
5:        $t_j \leftarrow t(L_j)$  ▷  $t(L_j)$  es el tiempo de vuelo de la muestra  $j$ 
6:        $\hat{L}_j \leftarrow L_j \cdot e^{-2\pi i \xi_k t_j}$ 
7:        $\hat{I}[k] \leftarrow \hat{I}[k] + \hat{L}_j$  ▷ Se acumula el valor complejo en el bin
8:     end for
9:   end for
10:  return  $\hat{I}$ 
11: end procedure
```

---

El algoritmo toma como entrada la lista  $L$ , que contiene  $N$  muestras generadas por el algoritmo de trazado de rayos que han llegado al píxel y un conjunto  $\Xi$  de  $N_F$  frecuencias. El valor de salida son los valores de radiancia calculados en el dominio de frecuencias. La diferencia más importante respecto a la transformada de la información temporal discretizada es que se utiliza el tiempo de vuelo exacto de cada muestra, a diferencia de la FFT, que no dispone de dicha información, por lo que el nuevo algoritmo reduce los errores de cuantización que resultaría de aplicar cualquier transformada de Fourier a posteriori sobre el renderizado transitorio discretizado en el tiempo.



## Capítulo 5

# Estudio del algoritmo en técnicas de imagen computacional

En este capítulo, se analizará el funcionamiento de los algoritmos de estimación de profundidad basados en tiempo de vuelo, o *time of flight imaging*. Se estudiarán los métodos de captura y se describirá el algoritmo de reconstrucción de profundidad. Seguidamente, se estudiará el comportamiento del algoritmo de renderizado transitorio en espacio de frecuencias a la hora de estimar la profundidad de una escena y se analizarán las diferencias respecto a las técnicas clásicas de reconstrucción.

### 5.1. Captura del tiempo de vuelo

Los algoritmos de obtención del **tiempo de vuelo**, o *time of flight*, son técnicas de imagen computacional con un amplio rango de aplicaciones, como obtener la profundidad de la escena, por ejemplo mediante técnicas de LiDAR. El proceso de estimación de la profundidad consiste en emitir un pulso de luz **co-localizado** con la cámara (es decir, que se encuentra en la misma posición que la cámara) que rebota en la escena y es capturado por el sensor para computar el tiempo de vuelo. La obtención directa de este tiempo requiere un entorno de captura que permita visualizar el movimiento de la luz, por lo que los tiempos de exposición de cada fotograma deben ser sumamente pequeños. Esto también supone un problema, dado que si el tiempo de exposición disminuye, menor cantidad de energía será capturada por el sensor, por lo que la información obtenida será poco fiable y tendrá un nivel alto de ruido. Además, el pulso de luz que se debe emitir debe ser muy corto, lo que dificulta la captura de fotones por el sensor. Los sistemas basados en iluminación láser, como los femtoláseres [2] permiten emitir estos pulsos ultracortos, pero su utilización y el entorno de captura necesario para capturar la luz en estado transitorio son caros y muy sensibles. Por esta razón, puede ser más conveniente utilizar señales de luz sinusoidales y utilizar sensores que capturan el desfase de la señal recibida tras interactuar con la escena, que en general suelen ser mucho más baratos. Los métodos que utilizan la

fase para reconstruir la profundidad se denominan métodos de **tiempo de vuelo basado en correlación**, o *C-ToF imaging*. Su principio de funcionamiento es el mismo, pero para estimar el desfase se determina una **medida de correlación** entre la señal emitida y la recibida tras rebotar en la escena. Este principio de funcionamiento se puede observar en la figura 5.1:

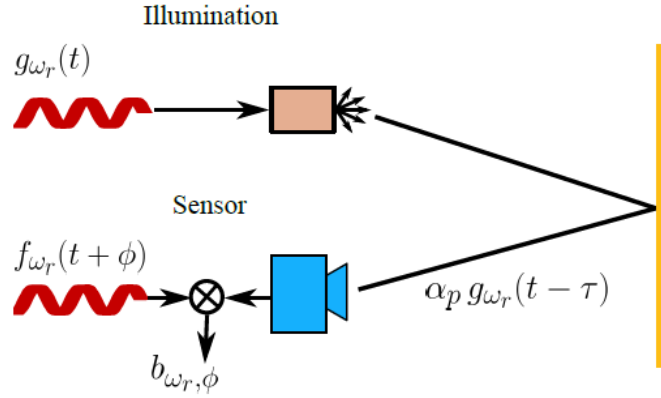


Figura 5.1: Esquema de captura con un sensor de tiempo de vuelo, utilizando una señal sinusoidal. La luz rebota en la escena, y el sensor calcula una medida de correlación para estimar el desfase de la señal recibida y calcular la profundidad. Fuente: Heide et al. (2014) [24]

## 5.2. Reconstrucción de profundidad

Consideremos en primer lugar la ecuación que modela la distancia del emisor a un punto según el retardo temporal  $\Delta t$  desde la emisión hasta la recepción de la señal, suponiendo que ésta se propaga a través del vacío a la velocidad de la luz y que la fuente de luz está colocalizada con la cámara. Esto es necesario dado que de esta forma el tiempo de vuelo de ida es igual al de vuelta y, por tanto, es posible estimar la profundidad:

$$d = \frac{c\Delta t}{2}. \quad (5.1)$$

Además, el desfase de la señal emitida a lo largo del tiempo se modela con la siguiente expresión, que se corresponde con el argumento de la función sinusoidal que describe la señal a lo largo del tiempo:

$$\phi = \phi_0 + 2\pi f\Delta t, \quad (5.2)$$

donde  $\phi_0$  es el desfase inicial de la onda. Despejando  $\Delta t$  en la ecuación 5.2, sustituyendo en la ecuación 5.1 y asumiendo por simplicidad que el desfase inicial es 0, se deduce la expresión que modela la distancia en función del desfase de la señal recibida:

$$d = \frac{\phi c}{4\pi f} = \frac{\phi \lambda}{4\pi}, \quad (5.3)$$

donde  $\lambda$  es la longitud de onda, que denota la distancia a partir de la que la señal sinusoidal se repite. Esto supone un problema, ya que el desfase medido por el sensor tomará valores entre 0 y  $2\pi$  radianes y la fase se repite cada vez que el pulso de luz recorre una distancia equivalente a una longitud de onda. Este problema se conoce como **envoltura de fase**, o *phase wrapping*, y se puede mitigar aumentando la longitud de onda, a cambio de reducir la precisión de la reconstrucción. Existen otras técnicas que resuelven el problema sin cambiar las características de la señal modulada, como la técnica de **desenvoltura de fase** [15], que permite deshacer la ambigüedad y estimar la fase a partir de una medición ambigua, aunque para ello es necesario emitir un conjunto de señales. Por ello, se puede deducir que la profundidad de reconstrucción máxima para una frecuencia  $f$  será  $\frac{2c}{\lambda}$ . A la hora de capturar la medida de fase, las cámaras de tiempo de vuelo utilizan 4 medidas  $c_1 \dots c_4$  con desfases iniciales distintos [11], que se combinan para estimar la fase de la siguiente forma:

$$\phi = \arctan \left( \frac{c_4 - c_2}{c_1 - c_3} \right) \quad (5.4)$$

En primer lugar, se puede asumir que cada medida proviene de un único punto de la escena, es decir, únicamente en condiciones de iluminación directa (es decir, sólo rebota una vez en la escena). Si hubiera más de un rebote, el tiempo de vuelo del camino depende de la geometría de la escena, por lo que la estimación de la profundidad de la escena será incorrecta. Así, el desfase de la señal en un píxel cualquiera se puede obtener directamente a partir de la información espacio-temporal recibida:

$$c_j = f(\bar{\mathbf{p}}_d) \cdot e^{2\pi i \Delta\phi(\bar{\mathbf{p}}_d) \frac{f}{c} + \phi_0^{(j)}}, \quad (5.5)$$

donde  $\bar{\mathbf{p}}_d$  es un camino de luz directa a un píxel del sensor,  $\phi_0^{(j)}$  es el desfase inicial de la señal y  $\Delta\phi(\bar{\mathbf{p}}_d)$  es el desfase percibido por el sensor. Sin embargo, si se tiene en cuenta la luz indirecta, es posible que la información recibida en un píxel provenga de varios puntos de la escena, por lo que se debe integrar sobre el espacio  $\Omega_i$  de todos los caminos de **luz indirecta**, es decir, todos aquellos que hayan rebotado más de una vez en la escena:

$$\hat{c}_j = c_j + \int_{\Omega_i} f(\bar{\mathbf{p}}_i) \cdot e^{2\pi i \Delta\phi(\bar{\mathbf{p}}_i) \frac{f}{c} + \phi_0^{(j)}} d\bar{\mathbf{p}}_i, \quad (5.6)$$

En la práctica, el efecto que provocan los múltiples rebotes en la escena se ignora, asumiendo que  $\phi = \hat{\phi}$ , por lo que se introduce un error de medición dado que un mismo píxel recibe información de varios puntos de la escena. Este efecto se conoce como el problema de la **interferencia multicamino**, o *multipath interference* (MPI). El efecto de la MPI a la hora de reconstruir la profundidad causa que las medidas obtenidas estén por encima

de la medida real, debido a que los caminos de luz indirecta van a tener un tiempo de vuelo mayor, por tanto, el desfase también será mayor.

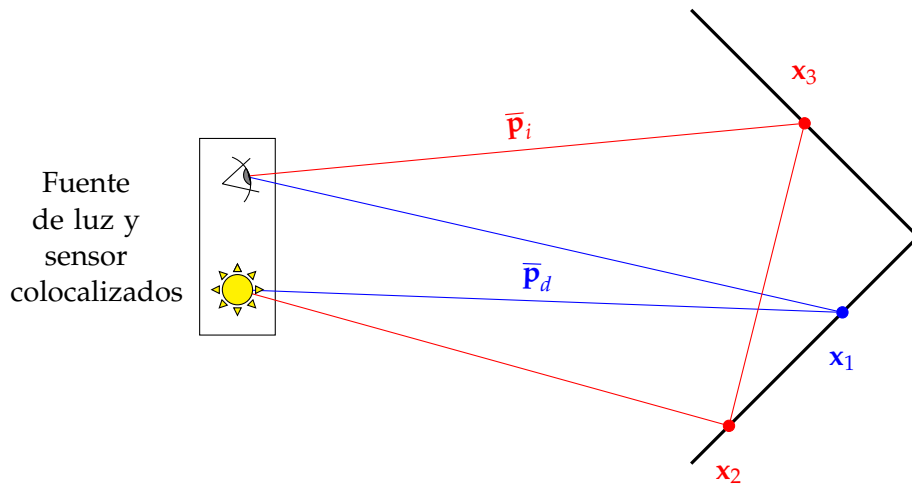


Figura 5.2: Esquema del problema de MPI.

En la figura 5.2 se puede observar este efecto, donde el camino de luz directa rebota únicamente en el punto  $x_1$  antes de llegar al sensor, por lo que el tiempo de vuelo de la fuente de luz a  $x_1$  es el mismo que el tiempo de vuelta de  $x_1$  al sensor. Sin embargo, el camino de luz indirecta tiene un tiempo de vuelo que depende de la geometría de la escena, ya que parte de la fuente a un punto  $x_2$ , pero llega a la cámara desde un punto  $x_3$  distinto, además de que hay un segmento adicional entre  $x_2$  y  $x_3$  que también contribuye al tiempo de vuelo.

La principal diferencia frente a la técnica anterior radica en que se puede obtener la fase directamente de la información en espacio de frecuencias, ya que la ecuación 5.6 es muy similar a la de la transformación descrita en la ecuación 4.5. Para poder reconstruir la profundidad, se ha asumido que la fase inicial de la señal emitida es 0, por lo que basta con obtener el argumento del valor complejo obtenido para cada píxel a una frecuencia dada. A pesar de ello, el problema de MPI sigue presente, dado que el nuevo algoritmo también considera la iluminación indirecta, por lo que es esperable que el error de reconstrucción aumente conforme al número de rebotes máximo establecido para los caminos de luz.

## Capítulo 6

# Resultados

En este capítulo se presentarán los resultados obtenidos al aplicar el nuevo algoritmo de renderizado transitorio en espacio de frecuencias. En primer lugar, se hará una comparativa con la información espacio-temporal transformada a posteriori, utilizando renders de varias escenas sintéticas, generados con Mitsuba 2. Después, se analizará el impacto en memoria y tiempo a la hora de generar la información y finalmente se analizará el resultado de utilizar la información generada para reconstruir la profundidad de una escena. A la hora de realizar las comparativas, se denotará el renderizado transitorio en el dominio temporal como TDTR (*Time Domain Transient Rendering*), el renderizado en el dominio de frecuencias como FDTR (*Frequency Domain Transient Rendering*), y la transformada de Fourier y su inversa como FT e IFT respectivamente.

### 6.1. Renderizado transitorio en espacio de frecuencias

La implementación del algoritmo de renderizado transitorio en dominio de frecuencias se ha llevado a cabo sobre una versión de Mitsuba 2 que soporta renderizado transitorio en el dominio temporal. Por razones de implementación y precisión numérica de la herramienta, para cada camino de luz se calcula la longitud del camino óptico en lugar del tiempo de vuelo, dado que los tiempos de vuelo son notablemente pequeños y el tipo de dato utilizado no alcanza la precisión necesaria para representar dichos valores. Esto no supone un problema para el algoritmo de renderizado en frecuencias, dado que en lugar de operar con frecuencias temporales se utiliza su análogo espacial, que es el inverso de la longitud de onda. Por esta razón, las magnitudes de referencia de las pruebas serán la frecuencia espacial y la longitud de onda. Por último, cabe destacar que la información generada con el nuevo algoritmo es monocroma, al igual que la que utilizan la mayor parte de los métodos de imagen transitoria.

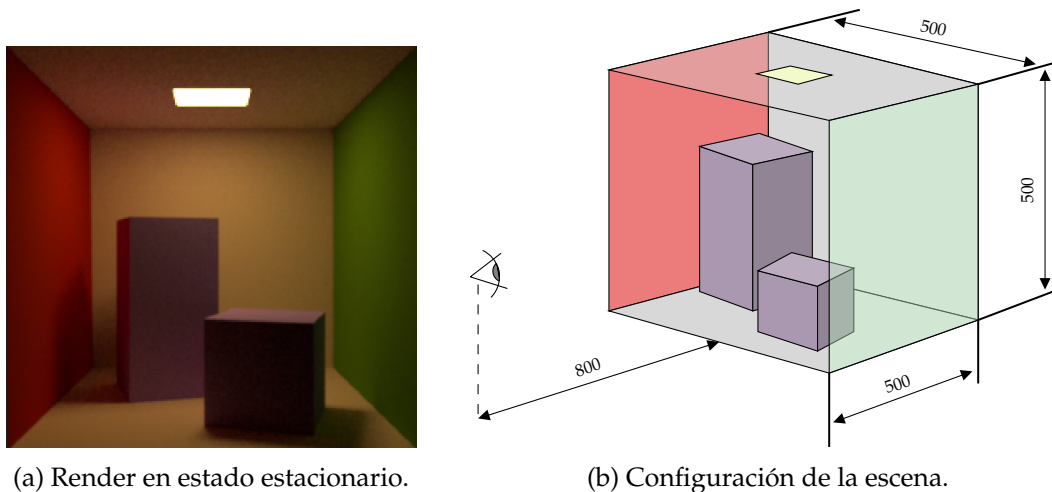


Figura 6.1: Cornell Box con luz de área.

### 6.1.1. Estudio cualitativo

Para la comparativa de los resultados, se ha renderizado cada escena aplicando la transformación durante el proceso de render o a posteriori. El resultado en ambos casos es un campo de fasores para cada frecuencia muestreada por la FFT. La metodología de validación consiste en generar el espectro de frecuencias mediante la FFT, para poder comparar los resultados de las siguientes formas:

- Se comparará la parte real y compleja de ambos renders, para observar si los patrones oscilatorios son similares. También se comprobará el valor de la fase de los fasores de ambos renders, que se obtendrá calculando el argumento del número complejo que representa el fasor.
- Se comparará la transformada inversa de Fourier de la información en frecuencias con la información espacio-temporal, dado que  $\mathcal{F}_t \{ \mathcal{F}_t^{-1} \{ I \} \} = I$ .

En primer lugar, se ha construido una *Cornell Box* con materiales difusos y se han realizado varios renders de la escena, tanto en estado estacionario como en estado transitorio, en el dominio temporal y en el de frecuencias. Además, se ha colocado una fuente de luz de área en el techo de la caja. La escena tiene un ancho y alto aproximado de 500 milímetros, y la cámara se encuentra a unos 800 milímetros de la *Cornell Box*. En la Figura 6.1 se incluye un render en estado estacionario de la escena, así como un esquema de la configuración y las dimensiones aproximadas de la misma.

### Comparativa de la parte real e imaginaria

Los primeros experimentos se han llevado a cabo lanzando 64 muestras por píxel, con una resolución temporal de 380 bins y un tiempo de exposición por bin equivalente a 8

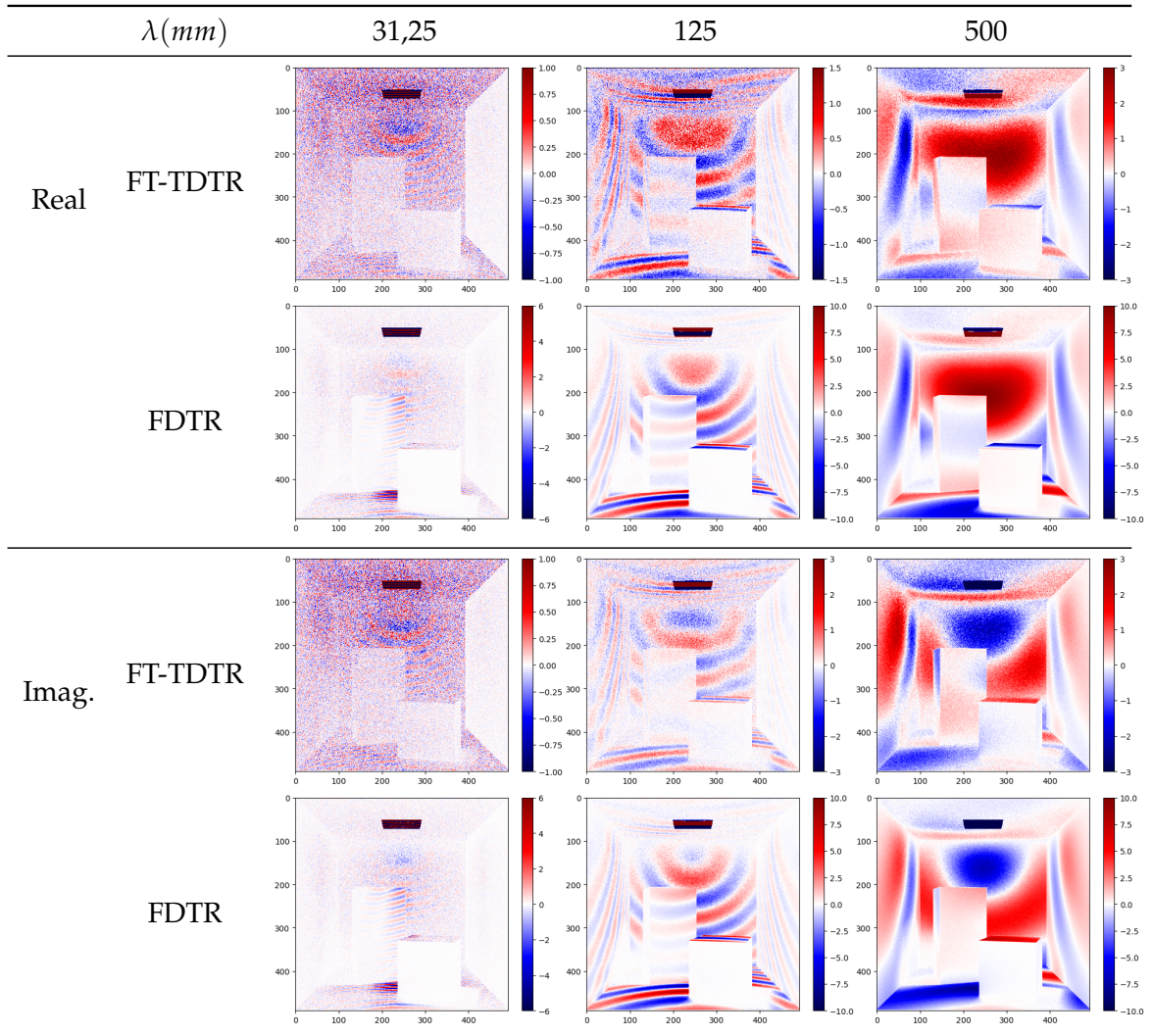
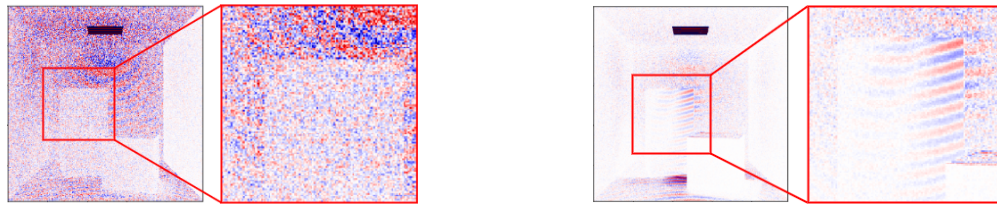


Figura 6.2: Comparativa de la información espacio-temporal transformada mediante FFT y la información directamente renderizada en espacio de frecuencias de la *Cornell Box*. Se incluye la parte real e imaginaria de 3 franjas moduladas con distintas longitudes de onda.

milímetros de camino óptico (en torno a  $8mm/c \approx 27$  picosegundos, siendo  $c$  la velocidad de la luz en el vacío). En la Figura 6.2 se puede observar que la información obtenida a partir de la FFT tiene un nivel de ruido mayor que la obtenida en el render en frecuencias. El cálculo de la FFT no considera los tiempos individuales de cada muestra sino que utiliza el índice de cada *bin* temporal, ya que es la única información de la que dispone, por lo que se introduce un error debido a la cuantización de los instantes temporales. Por otro lado, el render en frecuencias utiliza directamente el tiempo asociado a cada muestra, aplica la transformación y lo acumula después en los bins de frecuencias correspondientes, de manera que el ruido no se amplifica.

En ambos casos, se pueden apreciar los patrones oscilatorios típicos del resultado de una transformada de Fourier, que se alinean con la estructura del pulso de luz que se propaga a lo largo del tiempo. En este caso, se puede apreciar que el foco del que se dispersan los frentes



(a) Detalle del render transformado con la FFT.

(b) Detalle del render en frecuencias.

Figura 6.3: Detalle de una región en un render transitorio con 64 muestras, con una longitud de onda de 31.25 mm.

de onda parte de la fuente de luz de área que se encuentra en el techo de la *Cornell Box*. Cabe destacar también que se pierde esta estructura debido al ruido a medida que se incrementa la frecuencia. Sin embargo, al utilizar FT-TDTR, se pierde casi toda la estructura, mientras que usando FDTR se siguen apreciando ciertos patrones oscilatorios, como se puede observar en la Figura 6.3, así como la atenuación en los bordes de las paredes de la *Cornell Box*. Una primera hipótesis que se ha considerado es que este ruido es mucho mayor debido al error de cuantización que introduce la FFT, pero más adelante se señalarán otros posibles factores que puedan afectar a la calidad de la información en frecuencias. Es importante mencionar también el efecto del color de los materiales en la obtención de la información en espacio de frecuencias. En este caso, una de las paredes es de color verde, por lo que los valores del canal rojo en dicha región son más bajos, lo que explica que los tonos de la pared derecha sean más tenues. Finalmente, es importante destacar que el rango de valores de información espacio-temporal es distinto al de su equivalente en espacio de frecuencias, debido a que ahora cada muestra se acumula en todos los bins de frecuencias, por lo que ha sido necesario ajustar la escala de color de cada imagen para que se puedan comparar las imágenes entre sí.

Para la obtención de la fase, basta con computar el argumento del valor complejo asociado a cada píxel. El resultado obtenido se puede observar en la Figura 6.4, donde se muestra la fase para las mismas longitudes de onda del experimento anterior. En este caso, el valor del ángulo de la fase obtenido se encuentra entre  $-\pi$  y  $\pi$ . De nuevo, se pueden observar los patrones oscilatorios resultantes de calcular la transformada a espacio de frecuencias, así como el efecto de la variación de la longitud de onda escogida en la estructura de los frentes de ondas. Además, para facilitar la visualización de la envoltura de fase, se ha utilizado un mapa de color cíclico, por lo que los colores de los valores extremos son los mismos.

### Variación del número de muestras

Se han llevado a cabo también experimentos variando el número de muestras por píxel para observar si tiene algún efecto en la calidad de la reconstrucción. Se ha repetido la

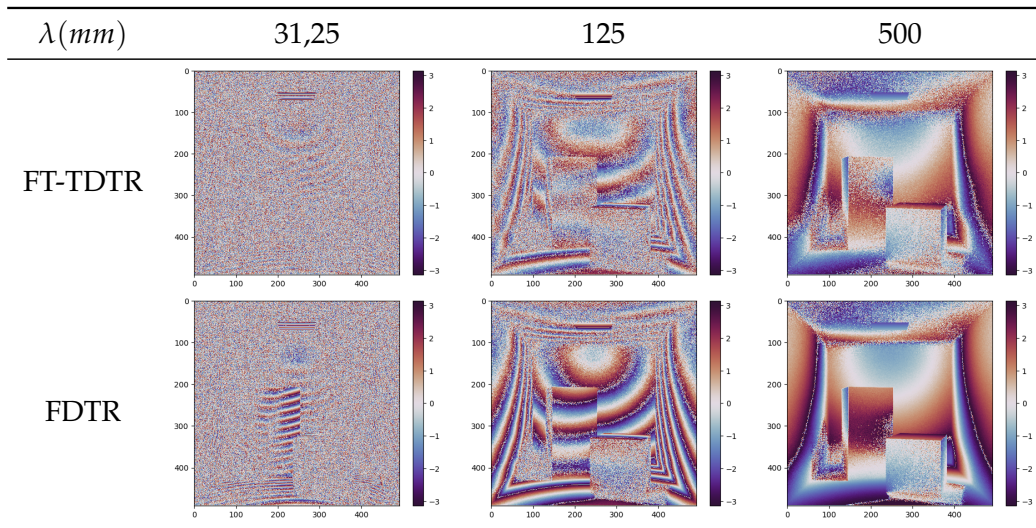


Figura 6.4: Comparativa de la fase obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias de la *Cornell Box*.

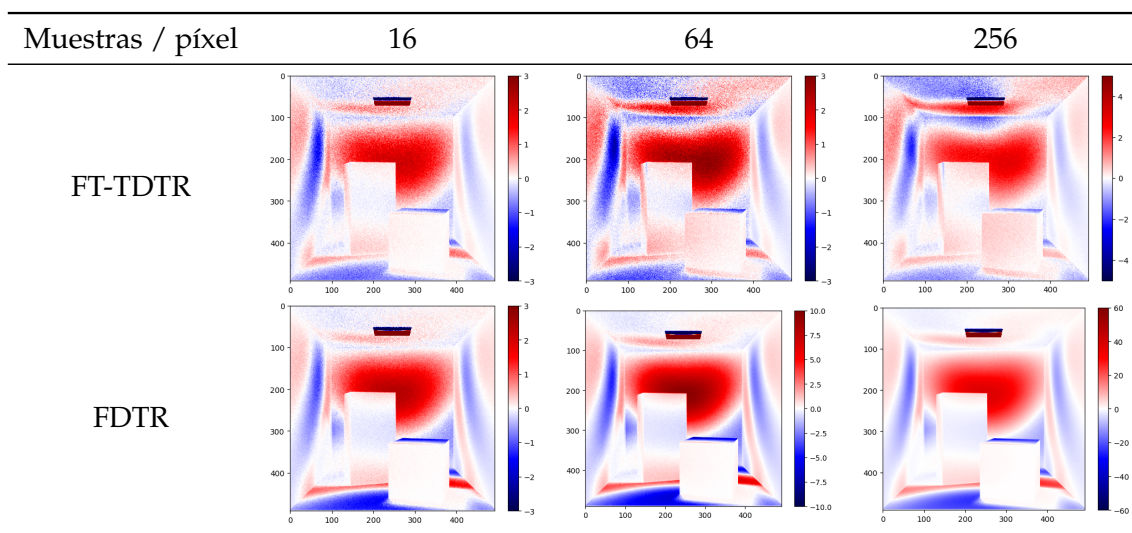


Figura 6.5: Comparativa de la parte real obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias para distinto número de muestras por píxel de la *Cornell Box*.

prueba para valores de 16, 64 y 256 muestras por píxel y se ha analizado el resultado para las longitudes de onda utilizadas en el experimento anterior. En la Figura 6.5 se incluye la parte real del resultado obtenido para la longitud de onda más larga, en torno a 5 metros. Se puede observar que en ambos casos el nivel de ruido disminuye a medida que se incrementa el número de muestras. Esto se debe a que cuantas más muestras por píxel se utilicen, mejor será la estimación realizada mediante el método de Monte Carlo. En el caso del render en espacio de frecuencias, se puede observar además que la amplitud de los valores también aumenta.

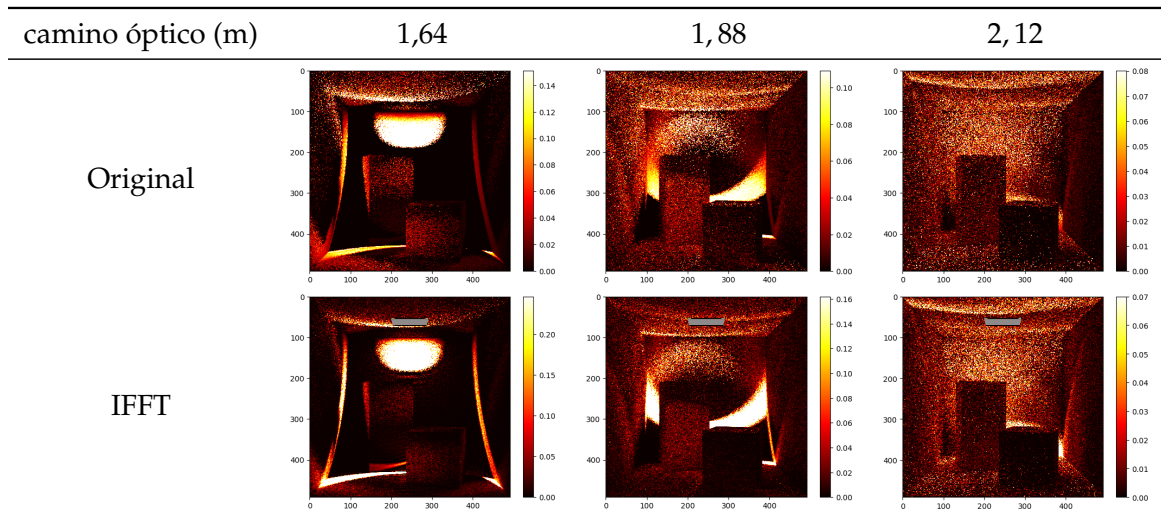


Figura 6.6: Comparativa de la información resuelta en tiempo y la transformada inversa del render en frecuencias de la *Cornell Box*.

### Transformada inversa de la información en frecuencias

En la Figura 6.6 se muestra la información temporal original usando TDTR y el resultado de la transformada inversa de Fourier de la información en frecuencias usando FDTR. Además, para visualizar mejor la estructura temporal se han utilizado imágenes de ráfaga, o *streak images*. Este tipo de imágenes representan una dimensión espacial y una temporal en lugar de representar el espacio de imagen completo, es decir, en un eje se representa una coordenada espacial (generalmente la  $X$ ), y en otro se incluye el dominio temporal. La utilización de imágenes de ráfagas es muy común en imagen transitoria, y es sumamente útil para analizar la información espacio-temporal obtenida. En primer lugar, se analizará el resultado en espacio de imagen:

En la Figura 6.6 se puede observar en ambos casos el frente del pulso de luz que se propaga por la escena, partiendo desde la fuente y dispersándose por la *Cornell Box*. La única diferencia se puede observar en la posición en la que se encuentra la fuente de luz, ya que en la información espacio-temporal no aparece, mientras que en el caso de la transformada inversa, aparece en la escena. Sin embargo, el fenómeno más peculiar se puede observar en la transformada inversa del render de frecuencias, que se puede apreciar especialmente bien en una imagen de ráfaga como la de la Figura 6.7. En este caso, se pueden apreciar que en ambas imágenes aparece el mismo frente del pulso de luz, que se corresponde con el primer rebote de la luz con la escena, pero en la Figura 6.7b es mucho más intenso, y aparecen una serie de artefactos ondulatorios.

Una posible hipótesis que podría justificar en cierto modo los artefactos obtenidos al determinar la IFFT se puede encontrar en el hecho de utilizar el tiempo de vuelo exacto para cada muestra en lugar de utilizar valores cuantizados. Esto implicaría que la transformación

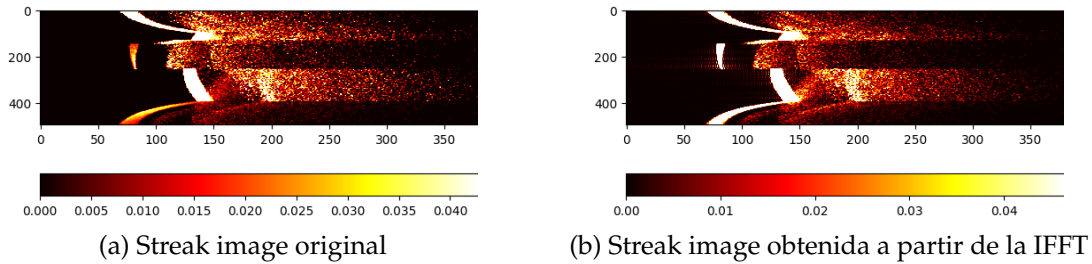


Figura 6.7: Imágenes de ráfaga de la Cornell box, a mitad de altura de la imagen. Se pueden observar ciertos artefactos ondulatorios en la IFFT de la información en frecuencias.

durante el proceso de renderizado no se podría invertir, salvo si se dispusiera de un espectro continuo de frecuencias, lo que es imposible en la práctica. Además, es posible que el efecto de la MPI se vea mitigado en cierto modo al hacer la estimación con la información de frecuencias. En el algoritmo de *Micro-ToF* propuesto por Gupta et al. (2015) [15] se menciona este efecto, en el que se observó que al utilizar fasores para modelar el transporte de luz, la componente de luz indirecta se desvanece a partir de cierta frecuencia de modulación, que es dependiente de la escena. No obstante, se estudiará este efecto en el futuro dentro del marco de renderizado en espacio de frecuencias para ver si esta es la causa por la que se obtiene una estimación de la profundidad mejor. Por ello, como trabajo futuro se podría investigar este fenómeno y analizar otras posibles causas que permitan justificar los artefactos, así como métodos que permitan mitigarlos.

### Variación del número de rebotes

Para analizar el comportamiento del algoritmo a la hora de aplicar técnicas de imagen computacional, se ha utilizado una escena con geometría más compleja. Esta escena, denominada *Sponza*, es una de las más utilizadas en el ámbito de la informática gráfica para validar el comportamiento de un renderizador. Las dimensiones aproximadas de la escena son de unos 30 metros de profundidad desde el punto de vista del observador y unos 8 metros de ancho. Además, se ha colocado una luz puntual co-localizada con la cámara que emite un pulso de luz para poder estimar la profundidad de la escena. En la Figura 6.8 se puede observar una imagen renderizada en estado estacionario de la escena.

En la Figura 6.9 se ha analizado el efecto de los rebotes al modular con una frecuencia alta. Se puede observar que si el límite de rebotes en la escena para cada camino es 1 (por tanto, sólo se utiliza la componente de luz directa), la información en espacio de frecuencias tiene la misma estructura. Además, en la información obtenida mediante la FFT, se introduce un error de cuantización debido a la discretización del dominio temporal que no aparece en la imagen renderizada en espacio de frecuencias. Este error se puede apreciar en las bandas de colores que aparecen en la información obtenida mediante FT-TDTR, lo que indica discontinuidades en los valores de fase causados por dicha cuantización. En cambio,

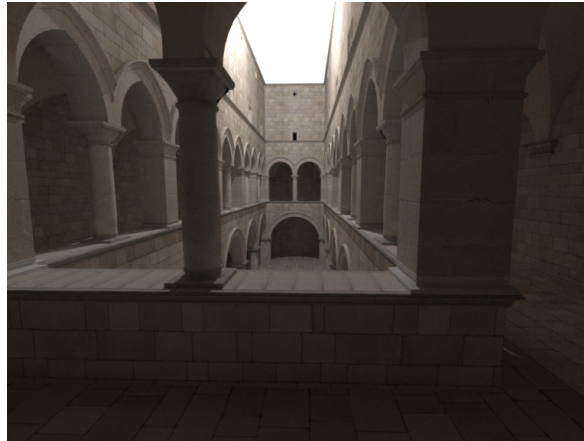


Figura 6.8: Imagen en estado estacionario de la catedral *Sponza*

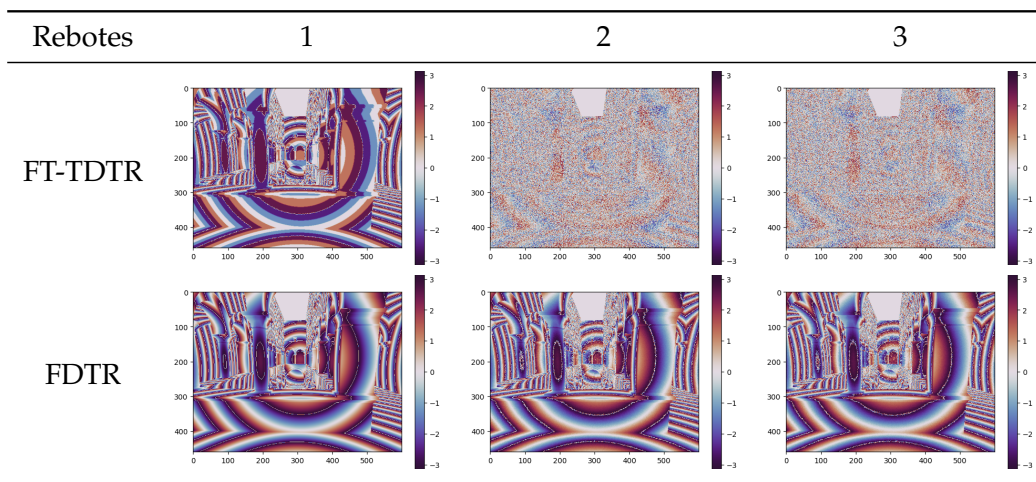


Figura 6.9: Comparativa de la fase obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias de la escena *Sponza*.

si se permite más de un rebote en la escena, la estructura de la imagen obtenida con la FFT es distinta a la obtenida con el nuevo algoritmo. También se puede observar que, al renderizar en el dominio de frecuencias, se preserva la misma estructura al aumentar el número de rebotes, a diferencia de la información espacio-temporal transformada con la FFT, que pierde la estructura debido al ruido.

### 6.1.2. Impacto en memoria y tiempo de ejecución

El tiempo de ejecución de un renderizador transitorio depende únicamente de forma lineal del número de muestras lanzadas y no del número de *bins* temporales, gracias a la estrategia de reutilización de caminos. En cambio, el renderizado transitorio en espacio de frecuencias depende también del número de frecuencias deseadas, dado que cada muestra se debe acumular en todos los *bins* de frecuencias. Para analizar el impacto del algoritmo en el coste computacional y de almacenamiento de los resultados, se han llevado a cabo una serie de pruebas en las que se analizan los costes según el número de frecuencias deseadas.

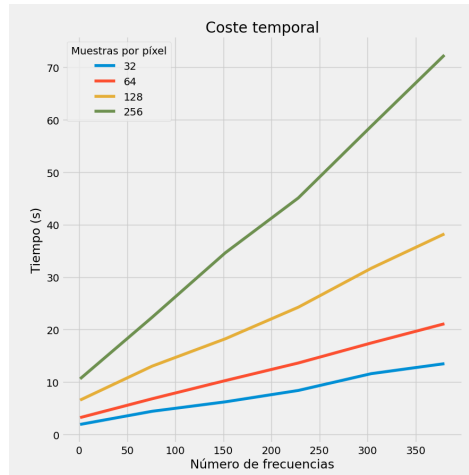


Figura 6.10: Evolución del coste temporal y de almacenamiento.

Se ha renderizado la escena de la *Cornell box* variando el número de frecuencias y el número de muestras, y se ha recogido el tiempo de renderizado de cada uno de los experimentos.

En la Figura 6.10 se puede observar que el coste temporal aumenta de forma lineal según el número de frecuencias devueltas y también depende linealmente del número de muestras por píxel para un número de frecuencias fijo. En cuanto al coste de almacenamiento, es evidente que el ahorro de espacio depende linealmente del número de frecuencias escogidas. Para determinar el posible ahorro de almacenamiento que se puede alcanzar con el nuevo algoritmo, supongamos que se quiere renderizar una imagen en estado transitorio de  $M$  píxeles de ancho,  $N$  píxeles de alto y  $T$  bins temporales. La información transformada a espacio de frecuencias mediante la FFT genera tantas frecuencias como instantes temporales haya en la imagen original. Asumiendo que se utiliza un tipo de datos que ocupa 8 bytes para representar los valores complejos obtenidos, el almacenamiento utilizado es de  $8 \cdot M \cdot N \cdot T$ . Sin embargo, si se generan  $F$  frecuencias con el nuevo algoritmo de renderizado, el bloque de información resultante ocupará  $8 \cdot M \cdot N \cdot F$  bytes, por lo que ocupará  $\frac{N}{F}$  veces menos espacio.

## 6.2. Aplicación del algoritmo para reconstrucción de profundidad

El objetivo es observar el efecto de la interferencia multicamino, que depende del número de rebotes de los caminos trazados. Si se limita el número de rebotes en la escena a 1, todos los caminos de luz que llegan al sensor serán de luz directa, por lo que no habrá error de estimación en la profundidad. En cambio, si se permite más de un rebote en la escena, el efecto inmediato es una sobreestimación de la profundidad de la escena. Para observar el incremento del error debido a este problema, se han realizado una serie de renders de la escena que varían dicho parámetro, utilizando la FFT y el render en espacio de frecuencias. Se ha aplicado la ecuación 5.1 para estimar la profundidad, donde la fase se ha normalizado

para que se encuentre en el intervalo  $[0, 2\pi]$ , y se ha utilizado una longitud de onda de aproximadamente 100 metros, dado que se debe superar el doble de la profundidad máxima de la escena para evitar el problema de *phase wrapping*.

En la Figura 6.11 se incluye una comparativa de la profundidad estimada a partir de la luz directa, así como permitiendo un máximo de 4 y 9 rebotes. En el techo no hay geometría, por lo que la estimación de profundidad devuelve un valor nulo. Se ha utilizado una longitud de onda de 100 metros para la reconstrucción, dado que la profundidad máxima de la escena es de en torno a 30 metros.

El efecto es mucho más evidente en la reconstrucción obtenida mediante la FFT, donde la estimación de profundidad aumenta considerablemente a medida que se incrementa el número de rebotes. Este problema se puede ver especialmente en el fondo del patio interior, donde la estimación de profundidad varía mucho con respecto a las superficies laterales esquinas. Sin embargo, en la estimación realizada a partir de la información renderizada en espacio de frecuencias, el error relativo es mucho menor, por lo que es difícil apreciar el error provocado por la interferencia multicamino en las figuras del mapa de profundidad. Para ello, en la Figura 6.12 se incluye un corte horizontal de la estimación de profundidad, aproximadamente a mitad de altura de la imagen. En él, se puede apreciar que el error de estimación debido a la interferencia multicamino es menor que el cometido al utilizar la FFT, pero sigue habiendo una sobre-estimación que crece con el número de rebotes. También se puede apreciar la silueta de las columnas laterales y el hueco del patio interior en el centro, donde también hay una sobreestimación de la profundidad. Además, en el caso de la FFT, se pierde casi todo el detalle debido al ruido provocado por la iluminación global, mientras que en el render en frecuencias se mantiene casi toda la estructura. En la Figura 6.12 también se puede observar el error de cuantización que comete la FFT al estimar la profundidad debido a la cuantización del tiempo de vuelo de las muestras en el perfil de la columna derecha, donde se puede observar un patrón de escalonado que manifiesta dicho error. El perfil de la reconstrucción realizada con el resultado de la FFT presenta unos patrones escalonados, mientras que en la estimación con la información obtenida con FDTR, el perfil es liso y continuo.

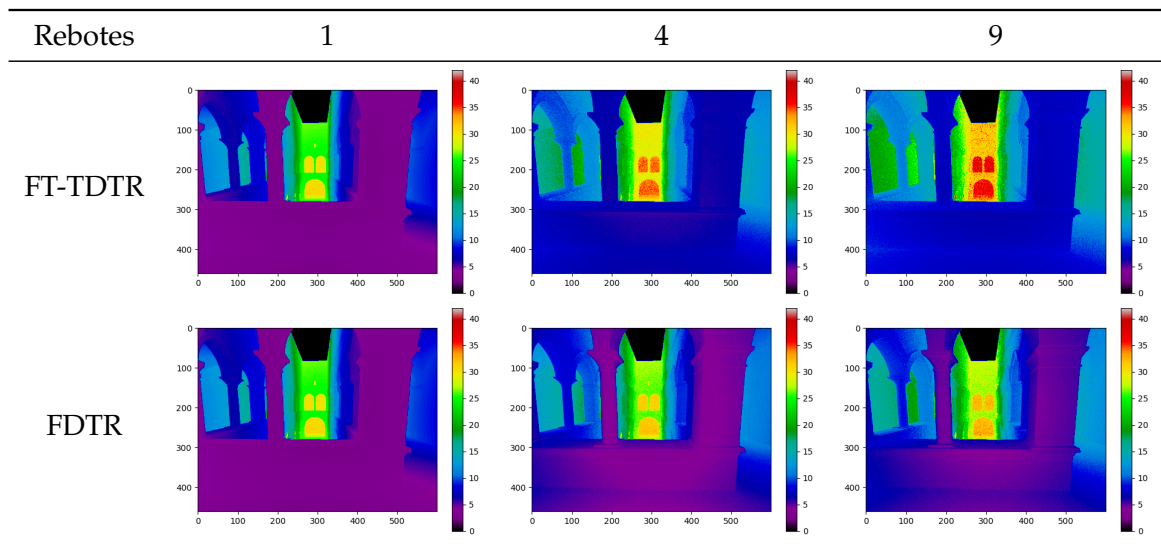


Figura 6.11: Comparativa de profundidad estimada a partir de la información espacio-temporal y el render en el dominio de frecuencias, para 1, 4 y 9 rebotes.

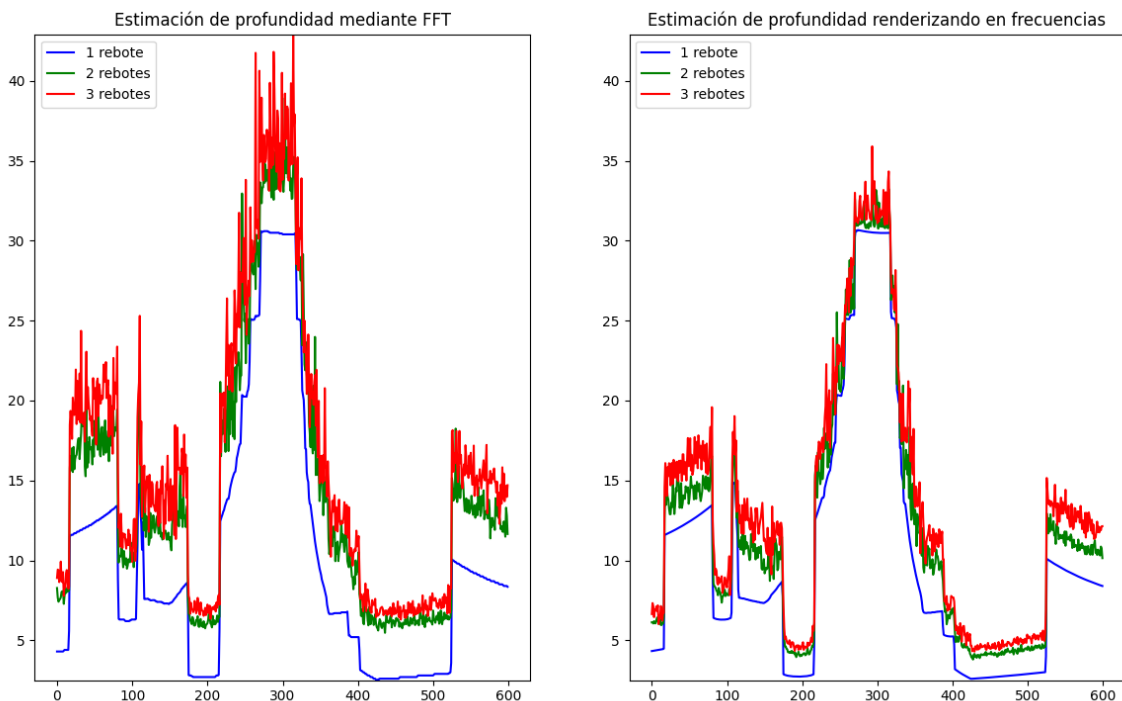


Figura 6.12: Perfil de la profundidad estimada con la FFT y con el render en frecuencias de la escena *Sponza*. En el eje horizontal se incluye la posición horizontal del píxel en la imagen, y en el eje Y se incluye la estimación de profundidad con cada algoritmo y número de rebotes.



## Capítulo 7

# Conclusiones

En este trabajo, se ha propuesto un algoritmo de renderizado transitorio que opera directamente en espacio de frecuencias. En primer lugar, se ha estudiado el estado del arte sobre las técnicas de imagen y renderizado transitorio en el dominio temporal y en espacio de frecuencias, así como sus posibles aplicaciones. Después, se ha formulado el transporte de luz en dominio de frecuencias a partir del marco de trabajo de Jarabo et al. [3], y se han analizado las potenciales ventajas de ahorro y precisión al utilizar un renderizador transitorio en dominio de frecuencias en lugar de en el dominio temporal. Después, se ha integrado el algoritmo en Mitsuba 2 y se ha llevado a cabo una comparativa de los resultados obtenidos con las técnicas clásicas de transformación a espacio de frecuencias de la información espacio-temporal, principalmente basadas en el algoritmo de la FFT. Finalmente, se ha estudiado el efecto de utilizar el nuevo algoritmo a la hora de aplicar técnicas de imagen computacional, en particular la estimación de profundidad de una escena y se ha realizado una comparativa de los resultados obtenidos.

El nuevo algoritmo de renderizado transitorio en espacio de frecuencias puede ser una herramienta de gran utilidad, ya que permite generar información de referencia que puede ser utilizada para validar y analizar nuevas técnicas de imagen transitoria que operen en este dominio. Además, la aplicación del algoritmo en una técnica de imagen computacional tan relevante y estudiada como la obtención de la profundidad ha permitido comprobar su validez y utilidad a la hora de simular técnicas de imagen transitoria ya existentes. Por último, se ha puesto de manifiesto el ahorro de almacenamiento gracias a la posibilidad de seleccionar el número de frecuencias, evitando guardar todo el volumen de datos de la información espacio-temporal. Dado el alcance esperado del trabajo, no se ha analizado de manera más detallada los efectos de renderizar directamente en espacio de frecuencias, aunque sí que se ha podido observar el efecto que tiene en la iluminación directa y global de la escena, así como los artefactos que aparecen al computar la transformada inversa, y se han planteado algunas hipótesis basadas en trabajos previos como el de Gupta et al. [15]. La implementación de este algoritmo puede servir como punto de partida para

estudiar este efecto en el futuro, así como su potencial aplicación en otras técnicas de imagen computacional basadas en campos de fasores como la reconstrucción de escenas ocultas.

## Capítulo 8

# Bibliografía

- [1] Adrian Jarabo, Belen Masia, Julio Marco, and Diego Gutierrez. Recent Advances in Transient Imaging: A Computer Graphics and Vision Perspective, November 2016. arXiv:1611.00939 [cs].
- [2] Andreas Velten, Di Wu, Adrián Jarabo, Belen Masia, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Mounqi Bawendi, Diego Gutiérrez, and Ramesh Raskar. Femto-photography: Capturing and visualizing the propagation of light. *ACM Transactions on Graphics (TOG)*, 32, 07 2013.
- [3] Adrián Jarabo, Julio Marco, Adolfo Muñoz, Raúl Buisan, Wojciech Jarosz, and Diego Gutiérrez. A framework for transient rendering. *ACM Transactions on Graphics*, 33(6):1–10, November 2014.
- [4] Roanna Lun and Wenbing Zhao. A survey of applications and human motion recognition with microsoft kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, 29:150330235202000, 03 2015.
- [5] Ahmed Kirmani, Tyler Hutchison, James Davis, and Ramesh Raskar. Looking around the corner using transient imaging. *2009 IEEE 12th International Conference on Computer Vision*, pages 159–166, 2009.
- [6] Diego Royo, Jorge García, Adolfo Muñoz, and Adrian Jarabo. Non-line-of-sight transient rendering. *Computers and Graphics*, 2022.
- [7] Adam Smith, James Skorupski, and James Davis. Transient Rendering.
- [8] Maik Keller and Andreas Kolb. Real-time simulation of time-of-flight sensors. *Simulation Modelling Practice and Theory*, 17(5):967–978, 2009.
- [9] Shinyoung Yi, Donggun Kim, Kiseok Choi, Adrian Jarabo, Diego Gutierrez, and Min H. Kim. Differentiable transient rendering. *ACM Trans. Graph.*, 40(6), dec 2021.

- [10] Rahul Nair, Stephan Meister, Martin Lambers, Michael Balda, Hannes Hofmann, Andreas Kolb, Daniel Kondermann, and Bernd Jähne. *Ground Truth for Evaluating Time of Flight Imaging*, pages 52–74. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [11] Julio Marco, Quercus Hernandez, Adolfo Muñoz, Yue Dong, Adrian Jarabo, Min Kim, Xin Tong, and Diego Gutierrez. Deeptof: Off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Transactions on Graphics (SIGGRAPH Asia 2017)*, 36(6), 2017.
- [12] Rudolf Schwarte, Zhanping Xu, Horst-Guenther Heinol, Joachim Olk, Ruediger Klein, Bernd Buxbaum, Helmut Fischer, and Juergen Schulte. New electro-optical mixing and correlating sensor: facilities and applications of the photonic mixer device (PMD). In Otmar Loffeld, editor, *Sensors, Sensor Systems, and Sensor Data Processing*, volume 3100, pages 245 – 253. International Society for Optics and Photonics, SPIE, 1997.
- [13] Felix Heide, Matthias B. Hullin, James Gregson, and Wolfgang Heidrich. Low-budget transient imaging using photonic mixer devices. *ACM Trans. Graph. (Proc. SIGGRAPH 2013)*, 32(4):45:1–45:10, 2013.
- [14] Andrew Dorrington, J.P. Godbaz, Michael Cree, Andrew Payne, and Lee Streeter. Separating true range measurements from multi-path and scattering interference in commercial range cameras. *Proceedings of SPIE - The International Society for Optical Engineering*, 7864, 01 2011.
- [15] Mohit Gupta, Shree K. Nayar, Matthias B. Hullin, and Jaime Martín. Phasor Imaging: A Generalization of Correlation-Based Time-of-Flight Imaging. *ACM Transactions on Graphics*, 34(5):1–18, November 2015.
- [16] Pablo Luesia, Miguel Crespo, Adrian Jarabo, and Albert Redo-Sanchez. Non-line-of-sight imaging in the presence of scattering media using phasor fields. *Optics Letters*, 47(15):3796, August 2022.
- [17] Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Mounqi Bawendi, and Ramesh Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications*, 3:745, 03 2012.
- [18] Sean I. Young, David B. Lindell, Bernd Girod, David Taubman, and Gordon Wetzstein. Non-line-of-sight surface reconstruction using the directional light-cone transform. In *Proc. CVPR*, 2020.
- [19] Xiaochun Liu, Ibón Guillén, Marco La Manna, Ji Hyun Nam, Syed Azer Reza, Toan Huu Le, Adrian Jarabo, Diego Gutierrez, and Andreas Velten. Non-line-of-sight

- imaging using phasor-field virtual wave optics. *Nature*, 572(7771):620–623, August 2019.
- [20] Xiaochun Liu, Sebastian Bauer, and Andreas Velten. Phasor field diffraction based reconstruction for fast non-line-of-sight imaging systems. *Nature Communications*, 11(1):1645, April 2020.
- [21] James T Kajiya. The rendering equation. 20(4), 1986.
- [22] Eric Veach. *Robust monte carlo methods for light transport simulation*. phd, Stanford University, Stanford, CA, USA, 1998. AAI9837162 ISBN-10: 0591907801.
- [23] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, 19:297–301, 1965.
- [24] Felix Heide, Lei Xiao, Andreas Kolb, Matthias B. Hullin, and Wolfgang Heidrich. Imaging in scattering media using correlation image sensors and sparse convolutional coding. *Opt. Express*, 22(21):26338–26350, Oct 2014.
- [25] D. Faccio, A. Velten, and G. Wetzstein. Non-line-of-sight Imaging. *Nature Reviews Physics*, 2(6):318–327, May 2020. arXiv:2005.08026 [physics].
- [26] Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), December 2019.
- [27] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory To Implementation*. 2018.



# Lista de Figuras

1.1. A la izquierda, “Bullet through Apple”, de Harold Edgerton (Fuente: MIT Museum). A la derecha, 3 fotogramas de una captura obtenida mediante femto-fotografía Fuente: Jarabo et al. (2014) [3] . . . . .	2
3.1. Representación gráfica de la interacción de la luz en un punto de la escena. . . . .	10
3.2. Esquema de la contribución de un camino de luz hacia la cámara. . . . .	12
3.3. Esquema espacio-temporal de la propagación de un camino de luz. Adaptado de “A Framework for Transient Rendering”, de Jarabo et al. (2014) [3] . . . . .	14
3.4. Diferencias entre un cálculo directo frente a la estrategia de <i>path reuse</i> . . . . .	15
4.1. Efecto de la disminución del tiempo de exposición de cada <i>frame</i> . . . . .	19
5.1. Esquema de captura con un sensor de tiempo de vuelo, utilizando una señal sinusoidal. La luz rebota en la escena, y el sensor calcula una medida de correlación para estimar el desfase de la señal recibida y calcular la profundidad. Fuente: Heide et al. (2014) [24] . . . . .	24
5.2. Esquema del problema de MPI. . . . .	26
6.1. Cornell Box con luz de área. . . . .	28
6.2. Comparativa de la información espacio-temporal transformada mediante FFT y la información directamente renderizada en espacio de frecuencias de la <i>Cornell Box</i> . Se incluye la parte real e imaginaria de 3 franjas moduladas con distintas longitudes de onda. . . . .	29
6.3. Detalle de una región en un render transitorio con 64 muestras, con una longitud de onda de 31.25 mm. . . . .	30
6.4. Comparativa de la fase obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias de la <i>Cornell Box</i> . . . . .	31
6.5. Comparativa de la parte real obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias para distinto número de muestras por píxel de la <i>Cornell Box</i> . . . . .	31

6.6. Comparativa de la información resuelta en tiempo y la transformada inversa del render en frecuencias de la <i>Cornell Box</i> . . . . .	32
6.7. Imágenes de ráfaga de la Cornell box, a mitad de altura de la imagen. Se pueden observar ciertos artefactos ondulatorios en la IFFT de la información en frecuencias. . . . .	33
6.8. Imagen en estado estacionario de la catedral <i>Sponza</i> . . . . .	34
6.9. Comparativa de la fase obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias de la escena <i>Sponza</i> . . . . .	34
6.10. Evolución del coste temporal y de almacenamiento. . . . .	35
6.11. Comparativa de profundidad estimada a partir de la información espacio-temporal y el render en el dominio de frecuencias, para 1, 4 y 9 rebotes. . . . .	37
6.12. Perfil de la profundidad estimada con la FFT y con el render en frecuencias de la escena <i>Sponza</i> . En el eje horizontal se incluye la posición horizontal del píxel en la imagen, y en el eje Y se incluye la estimación de profundidad con cada algoritmo y número de rebotes. . . . .	37
A.1. Diagrama de Gantt con la planificación seguida durante el proyecto. . . . .	49
B.1. Escena para analizar el efecto del número de rebotes . . . . .	51
B.2. Comparativa de la fase obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias de la escena <i>bunny</i> . . . . .	52
B.3. Comparativa de la información resuelta en tiempo y la transformada inversa del render en frecuencias de la escena <i>bunny</i> en espacio de imagen. . . . .	52
B.4. Imágenes de ráfaga de la escena del conejo, a mitad de altura del conejo. . . . .	53

# **Anexos**



# Anexos A

## Planificación del trabajo

El trabajo se ha realizado aproximadamente entre los meses de febrero y mayo, con una duración aproximada de 245 horas. La tarea que más tiempo ha ocupado ha sido la de validación, ya que se ha llevado a cabo un gran volumen de pruebas con distintas escenas y configuraciones.

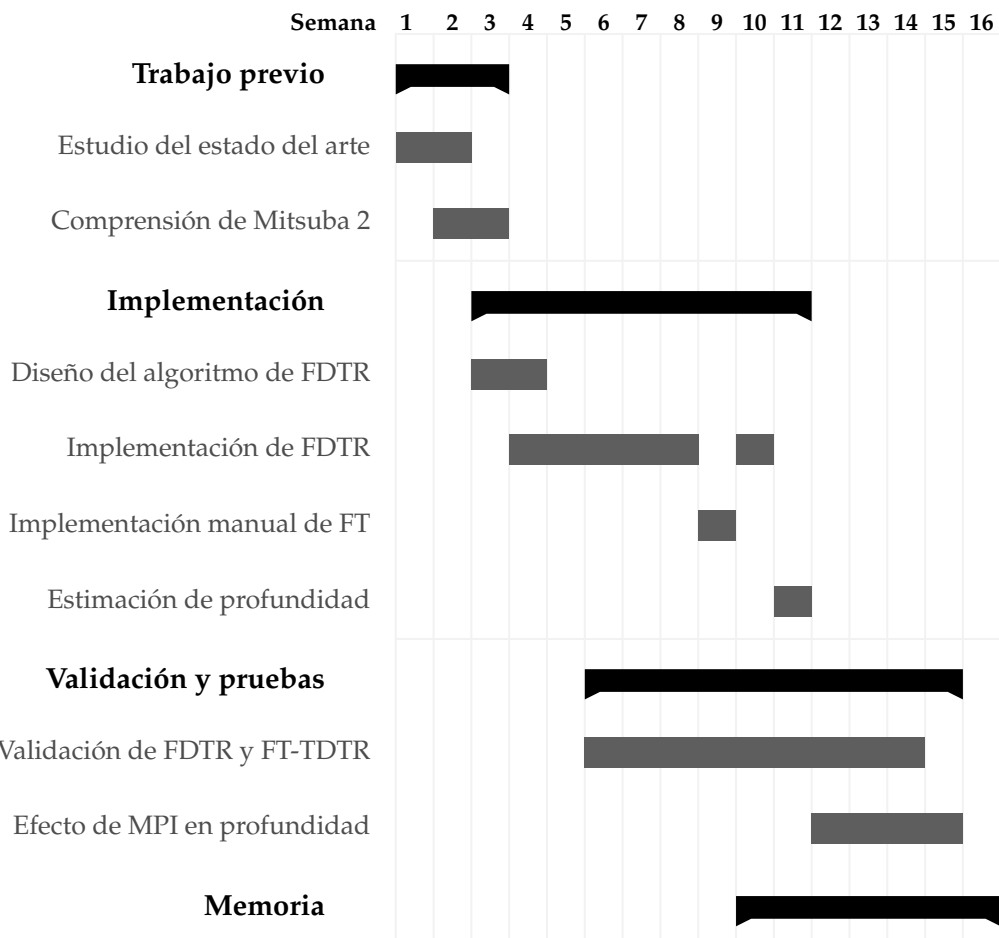
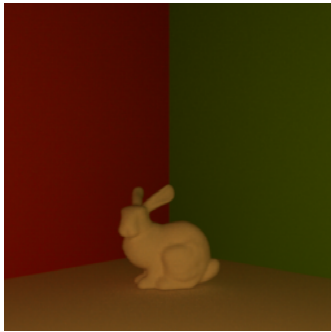


Figura A.1: Diagrama de Gantt con la planificación seguida durante el proyecto.

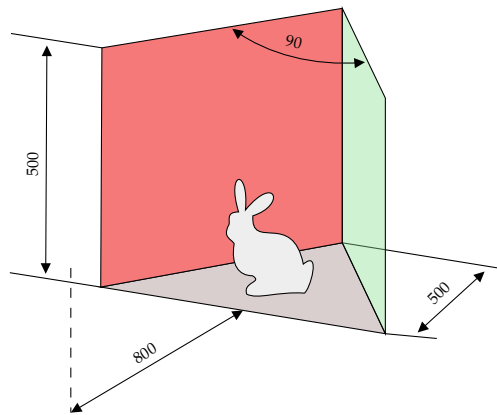


## Anexos B

# Resultados adicionales



(a) Render en estado estacionario.



(b) Configuración de la escena.

Figura B.1: Escena para analizar el efecto del número de rebotes

Se ha diseñado una escena adicional con un foco de luz puntual co-localizado con la cámara para probar la estimación de profundidad. Se han añadido un par de paredes formando un ángulo de  $90^\circ$ , para poder observar el efecto de la interferencia multicamino a medida que se incrementa el número de rebotes. También se ha incluido un objeto con características geométricas más complejas para observar la precisión de la reconstrucción en regiones con concavidades. En la figura B.1 se incluye un render en estado estacionario de la escena y la configuración de la misma. El tiempo de exposición e inicio de la captura en este caso se ha fijado al equivalente a 8 y 920 milímetros de camino óptico respectivamente. En la figura B.2 se incluye una comparativa de la fase variando el número de rebotes.

También se ha analizado el efecto de variar el número de rebotes en la transformada inversa del render en espacio de frecuencias, y los resultados se pueden visualizar en la figura B.3. En ambos casos se puede visualizar la propagación del pulso de luz por el espacio, aunque en esta ocasión, el artefacto que se ha visto en la *streak image* de la figura 6.7 se puede apreciar también en espacio de imagen, donde destacan los patrones ondulatorios, que son especialmente intensos cerca del pulso de luz. En el anexo B se incluye una comparativa de las imágenes de ráfaga de la escena del bunny, realizadas a mitad de

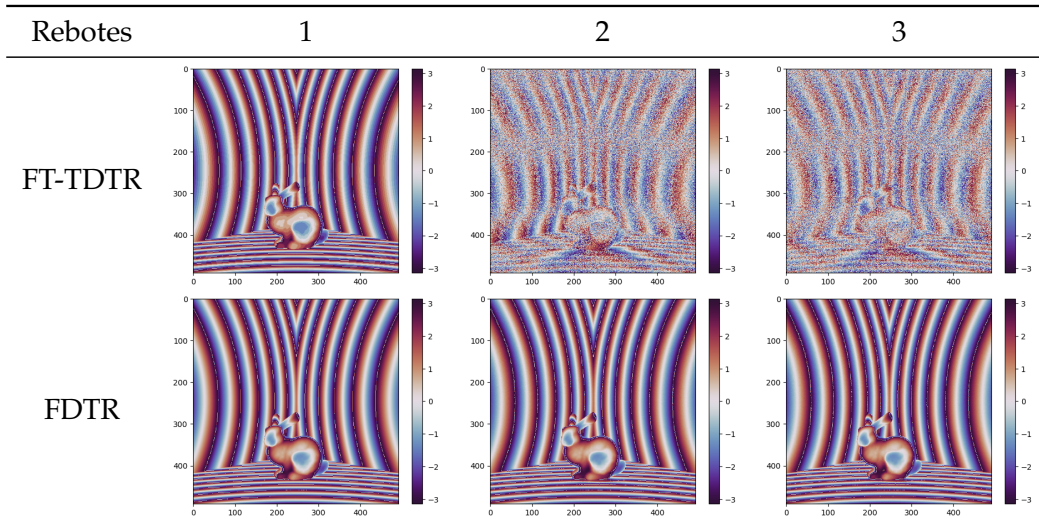


Figura B.2: Comparativa de la fase obtenida mediante FFT y la información directamente renderizada en espacio de frecuencias de la escena *bunny*.

altura del conejo. Además de estos artefactos presentes en la transformada inversa, se puede observar otro efecto que en este caso se debe a que la resolución de la imagen no es suficiente para mostrar todo el detalle de los patrones oscilatorios, por lo que aparecen unos patrones de interferencia, denominado efecto *Moiré*. Este problema es independiente del algoritmo de renderizado, y se puede solucionar fácilmente incrementando la resolución de la imagen.

Se han analizado también las imágenes de ráfaga de la escena del conejo, más o menos a la mitad de la altura del conejo en la imagen, y también se pueden observar los artefactos ondulatorios resueltos en tiempo en la figura B.4. El artefacto que aparece a la hora de aplicar la transformada inversa de Fourier se aprecia especialmente bien en el frente del pulso de luz cuando interacciona con el conejo, que se encuentra en el centro de las imágenes de

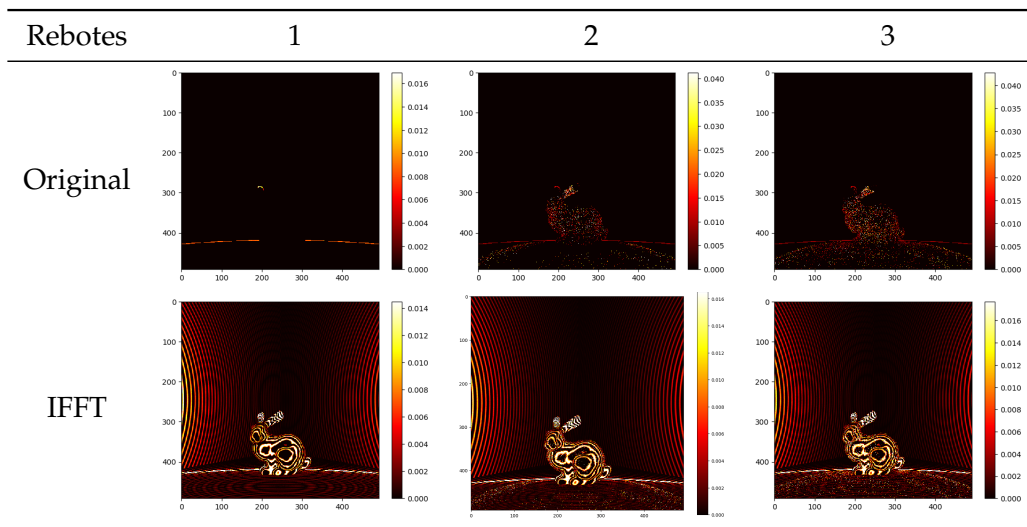


Figura B.3: Comparativa de la información resuelta en tiempo y la transformada inversa del render en frecuencias de la escena *bunny* en espacio de imagen.

ráfaga. En el caso de la información obtenida a partir del render en frecuencias, se vuelve a observar que el pulso de luz directa es mucho más intenso en comparación con los rebotes que lo siguen.

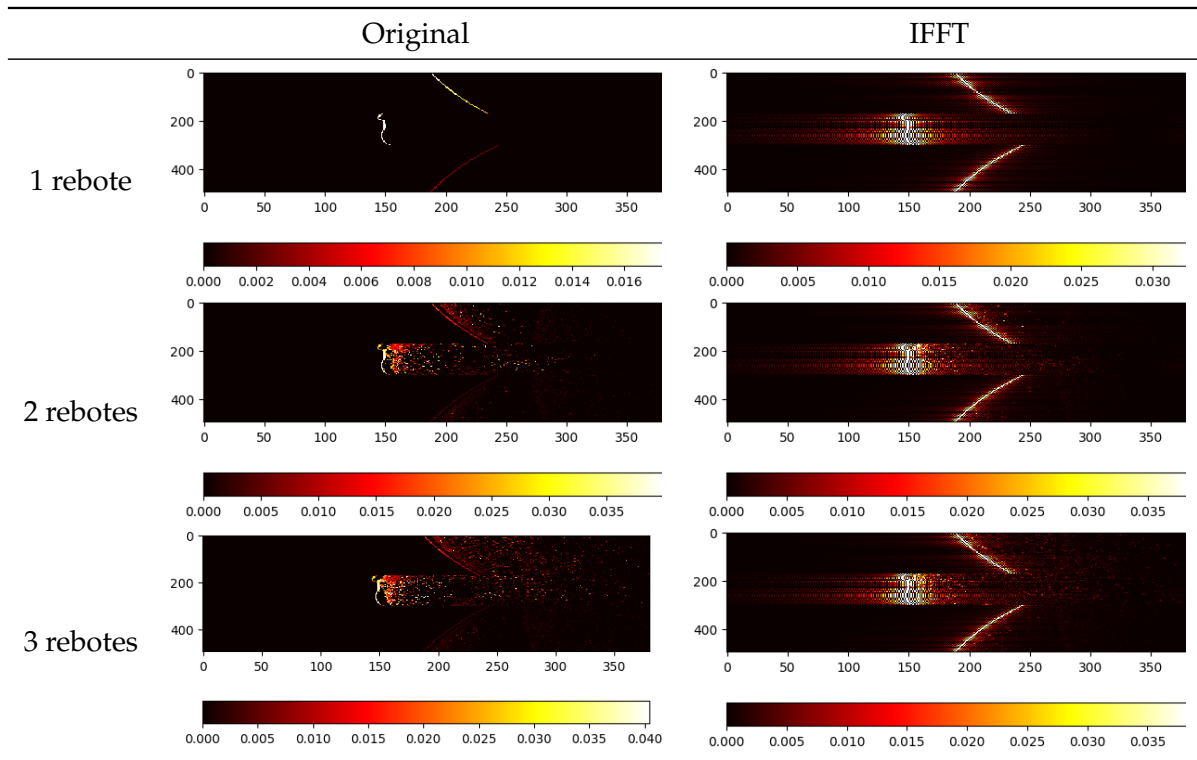


Figura B.4: Imágenes de ráfaga de la escena del conejo, a mitad de altura del conejo.