



UNC

Universidad
Nacional
de Córdoba

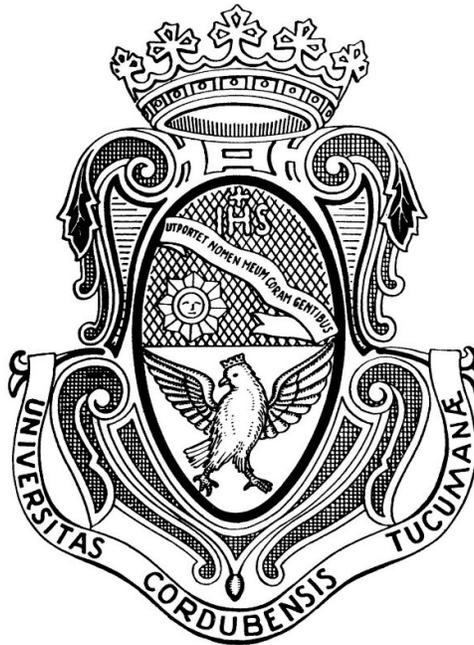
FAMAF

Facultad
de Matemática,
Astronomía, Física
y Computación

FCEyN

Facultad de
Ciencias Exactas
Físicas y Naturales

MAESTRÍA EN ANÁLISIS Y PROCESAMIENTO DE IMÁGENES



Tesis:

“IMPLEMENTACIÓN Y EVALUACIÓN DE ALGORITMOS PARA LA VISUALIZACIÓN DE IMÁGENES DE CAMPOS DE LUZ”

Bioing. Gustavo H. Bizai

Marzo de 2018

Directores: M. en I. A. Bioing. Adrián Salvatelli – FI-UNER

Lic. Jesús Humberto Calderón – FaMAF-UNC



Implementación y evaluación de algoritmos para la visualización de imágenes de campos de luz. Por Gustavo H. Bizai. Se distribuye bajo una [Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Agradecimientos

A Dios: nada es mérito mío, todo es su regalo.

A mi amada familia, mi esposa Roxana y mis hijos Jeremías, Justina y Donato, por su comprensión, paciencia y acompañamiento en esta etapa de mi vida.

A mi querida madre y mis hermanos y en especial a Ariana, por recibirme en su hogar de Córdoba; siempre me sentí como en casa.

A mi colega Adrián Salvatelli, con quien llevo recorridos más de 20 años en esta maravillosa vocación docente y de quien he aprendido muchas cosas pero sobre todo, el valor de la amistad.

Al Licenciado “Beto” Calderón, quien fuera profesor en mi carrera de grado y con quien tuve la dicha de recontrarme como profesor de maestría. Gracias por sus enseñanzas y consejos.

A Ana Laffitte, Bartolomé Drozdowicz y Facundo “Jerry” Peiretti, por su aliento y apoyo para culminar este trabajo.

A los evaluadores de esta tesis, por su gentileza, esmero y dedicación.

Al Doctor Oscar Bustos, siempre bien dispuesto y atento a mis necesidades, muchas gracias.

Abstract

In this thesis we determine the possibilities of using the light field model to generate new representations of a 3D scene by analyzing the space-angular information that contains the plenoptic function and its coding in a 4D matrix, selecting parametrization of two parallel planes. We implemented this coding, mutiperspective visualization and refocusing in the spatial and frequency domain, based on numerous authors and an experimental device. We evaluate the algorithms based on process times, preservation of the photometric attributes of the scene and refocus ranges. We conclude that the plenopic photography is a powerful tool for 3D visualization.

Resumen

En la presente Tesis determinamos las posibilidades de la utilización del modelo de campos de luz para generar nuevas representaciones de una escena 3D analizando la información espacio-angular que contiene la función plenóptica y su codificación en una matriz 4D, seleccionando parametrización de dos planos paralelos. Implementamos dicha codificación, visualización mutiperspectiva y reenfoque en el dominio espacial y frecuencial, basados en numerosos autores y un dispositivo experimental. Evaluamos los algoritmos en base a tiempos de proceso, preservación de los atributos fotométricos de la escena y rangos de reenfoque. Concluimos que la fotografía plenóptica es una potente herramienta para visualización 3D.

Listado de Figuras, Gráficos y Tablas

Figura 1.1. Imágenes reenocadas y vistas desde diferentes perspectivas	8
Figura 1.2. Manuscrito de Leonardo Da Vinci, sobre visión binocular	11
Figura 1.3. Dispositivos históricos para visualización estéreo	12
Figura 1.4. Barreras de paralaje y lenticulas	13
Figura 1.5. Dibujo original de Lippmann de su sistema de lenticulas	15
Figura 1.6. Esquema de funcionamiento del sistema autoestéreo de Lippmann	16
Figura 1.7. Dibujo original de Sokolov, sistema de imagen integral basado en estenopes	17
Figura 2.1. Parametrización de los rayos Posición-Ángulo y Dos Planos Paralelos	21
Figura 2.2. Parametrización de los rayos simplificado	21
Figura 2.3. Espacio de las fases q-p para Posición-Ángulo	22
Figura 2.4. Espacio de las fases q-p para Dos Planos Paralelos	22
Figura 2.5. Esquema de una cámara de agujero	24
Figura 2.6. Espacio de los rayos correspondiente a una cámara de agujero	24
Figura 2.7. Información de profundidad en una cámara de agujero	25
Figura 2.8. Esquema de formación de imagen para cámara de lentes convencional	26
Figura 2.9. Recuperación de la imagen a partir de la integración de las proyecciones sobre el eje x del espacio de los rayos	26
Figura 2.10. Esquema de formación de imagen mediante una cámara de lentes convencional, para una escena con puntos en diferentes profundidades	26
Figura 2.11. Esquema de una cámara plenóptica cuyas lenticulas refractan los conos de luz correspondientes a puntos pertenecientes al plano focal	27
Figura 2.12. Diferentes geometrías y organización de microlentes	28
Figura 2.13. Dependencia entre la resolución espacial de la imagen y el pitch de las microlentes de una cámara plenóptica	29
Figura 2.14. Espacio de los rayos de una cámara plenóptica para dos puntos de la escena pertenecientes al plano focal	29
Figura 2.15. Espacio de los rayos de una cámara plenóptica para puntos de la escena pertenecientes a diferentes profundidades	30
Figura 2.16. Integrales de proyección para reenfoques en diferentes profundidades de la escena	31
Figura 2.17. Coordenadas del espacio de los rayos para imagen focalizada a distancia F'	32
Figura 2.18. Cámara plenóptica Plenoptic 2.0	34
Figura 2.19. Imagen Renderizada Plenoptic 2.0	34
Figura 2.20. Prototipo Primera Generación de Adobe LightField Camera	35
Figura 2.21. Izq.: Segunda Generación de Adobe LF Camera	36
Figura 2.22. Izq.: Cámara plenóptica de Georgiev, 3ª Generación	36
Figura 2.23. Diferentes prototipos de cámaras plenópticas desarrolladas por el Grupo CAFADIS de la Universidad de La Laguna	37
Figura 2.24. Matriz de microcámaras "PiCam", integrada en un teléfono móvil	38
Figura 2.25. Esquema de patente "Super Light - Field Lens"	38
Figura 2.26. Izq.: Microcámara plenóptica de 8Mp desarrollada por Toshiba	39
Figura 2.27. Anteojos de Campos de Luz de NVIDIA	40

Figura 2.28. Módulo MEMS de DigitalOptics “MEMS cam”	40
Figura 2.29. Modelos de cámaras Lytro	41
Figura 2.30. Modelos de cámaras Raytrix	42
Figura 2.31. Gafas de Realidad Aumentada de AVEGANT	42
Figura 3.1. Detalle de las microimágenes capturadas con cámara plenóptica enfocada	45
Figura 3.2. Detalle de imagen cruda de cámara Plenoptic 1.0	46
Figura 3.3. Influencia de la apertura de las microlentes en la profundidad de campo	48
Figura 3.4 Incremento del ruido a medida que disminuye la apertura de las microlentes	49
Figura 3.5. Generación de imágenes en diferentes perspectivas para Plenoptic 1.0	49
Figura 3.6. Ejemplo de imágenes recuperadas en diferentes perspectivas	50
Figura 3.7. Esquema de renderización para arquitectura Plenoptic 2.0	51
Figura 3.8. Renderización de una imagen capturada con una cámara Plenoptic 2.0	52
Figura 3.9. Reenfocos con algoritmo desplazamiento suma	54
Figura 3.10. Esquema de la Transformación de Sesgado del espacio de los rayos	55
Figura 3.11. Refocalización por transformación de la radiancia	55
Figura 3.12. Esquema comparativo Cámaras Plenoptic 1.0 y Plenoptic 2.0	56
Figura 3.13. Espacio de las fases para geometría Plenoptic 1.0 y Plenoptic 2.0	57
Figura 3.14. Esquema de formación de imagen en una cámara plenóptica enfocada	58
Figura 3.15. Renderización por bloques para Plenoptic 2.0	58
Figura 3.16. Relaciones entre el tamaño del bloque y de la microlente	59
Figura 3.17. Imágenes resultado del algoritmo CorteBloques	60
Figura 3.18. Espacio de fase q-p de una arquitectura Plenoptic 2.0	61
Figura 3.19. Diagrama simplificado del algoritmo de Georgiev Basic Patch-Based Rendering .	61
Figura 3.20. Imágenes renderizadas con diferente tamaño de bloques	64
Figura 3.21. Espacio de fase q-p de una arquitectura Plenoptic 2.0 para lograr el efecto de suavizado de las zonas fuera de foco	65
Figura 3.22. Imágenes con Renderización por Bloques convencional y Renderización por Bloques con suavizado	65
Figura 3.23. Pila Focal de 4 imágenes obtenida con el algoritmo Transformada Rápida Discreta del Focal Stack	66
Figura 3.24. Transformación de un rayo según se desplace o refracte, parametrización posición – ángulo	67
Figura 3.24. Refocalizaciones por corte del hiperespacio de Fourier de la radiancia	70
Figura 3.25. Esquema del Teorema del Corte de Fourier	71
Figura 3.26. Relaciones entre la posición del foco, sus proyecciones en el dominio espacial y las Transformadas de Forier de dichas proyecciones	72
Figura 3.27. Proyecciones en el espacio de fase y sus espectros según rango de profundidades mapeado	73
Figura 3.28. Imágenes reenfocadas con Filtro Planar Frecuencial	74
Figura 3.29. Imágenes reenfocadas con Filtro frecuencial Hyperfan para foco volumétrico	74
Figura 4.1. Representación de EPI y relación de la inclinación de las rectas con las profundidades en la escena.....	79

Figura 4.2. Flujo de trabajo del algoritmo combinación de análisis de desenfoque y correspondencia, de Tao y colaboradores	79
Figura 4.3. Resultados intermedios y final del mapa de profundidad, combinación de desenfoque y correspondencia	80
Figura 4.4. Imágenes de mapas de profundidades obtenidas con algoritmo de Tao	80
Figura 4.5. Imagen original, su mapa de profundidad generado por <i>Lytro Desktop</i> y Mapa de Profundidad combinado Desenfoque-Correspondencia de Tao	81
Figura 4.6. Imagen original de foco extendido (enfocada en todas las profundidades) y su mapa, generado por <i>Lytro Desktop</i>	82
Figura 4.7. Imagen original de foco extendido (enfocada en todas las profundidades) y su mapa, generado por algoritmo de Tao	82
Figura 4.8. Mapa de profundidades generado por <i>Lytro Desktop</i> y por algoritmo de Tao	83
Figura 4.9. Imagen original de un gancho y su comparación con otros elementos	84
Figura 4.10. Mapa de desenfoque, correspondencia y combinado obtenidos con el algoritmo de Tao. Mapa de profundidades generado por <i>Lytro Desktop</i> . Mapa de profundidades combinado, en formato de malla 3D	84
Figura 5.1. Algunas fotografías de la escena preparada para obtener el set de datos	87
Figura 5.2. Montaje de la escena con fantoma escalera	87
Figura 5.3. Visualización de 2 perspectivas diferentes, cámara Lytro Illum	89
Gráfico A. Tiempos de cómputo Desplazamiento-Suma, diferentes alfa	91
Gráfico B. Tiempos de cómputo Desplazamiento-Suma, distintos pixeles / lentícula	91
Gráfico C. Tiempos de cómputo Desplazamiento Suma, diferente número microlentes	91
Gráfico D. Relación del tiempo de cómputo con el N° lenticulas, Desplazamiento Suma	91
Figura 5.4. Perfiles de imagen para observar regiones en foco	92
Figura 5.5. Imagen reenfocada y su par filtrado Laplaciano, para diferentes alfa	94
Figura 5.6. Pares Filtrado Laplaciano y Sustracción Enfocada, para mismo valor alfa	95
Figura 5.7. Imagen reenfocada y complemento de la diferencia reenfocada – infocus	95
Figura 5.8. Pares Imagen reenfocada - Laplaciano para diferentes valores de alfa	96
Figura 5.9. Pares Imagen renfocada - LoG para diferentes valores de alfa	98
Figura 5.10. Imágenes fusionadas LoG para parámetros alfa contiguos	99
Gráfico E. Tiempos de cómputo Sesgado de la Radiancia, diferentes parámetros theta	100
Gráfico F. Tiempos de cómputo Sesgado de la Radiancia, distintos pixeles / lentícula	100
Gráfico G. Tiempos de cómputo Sesgado de la Radiancia, distinto número microlentes	100
Gráfico H. Relación del tiempo de cómputo con N° lenticulas, Sesgado de la Radiancia	100
Figura 5.11. Pares Imagen renfocada - LoG para diferentes valores de theta	101
Figura 5.12. Imágenes fusionadas LoG para parámetros theta contiguos	102
Figura 5.13. Preservación de los atributos de brillo, contraste y balance de color	102
Figura 5.14. Perfiles de imagen por canal de color e histogramas	103
Gráfico I. Tiempos de cómputo Corte del Espectro, diferentes parámetros beta	104
Gráfico J. Tiempos de cómputo Corte del Espectro, distintos pixeles / lentícula	104
Gráfico K. Tiempos de cómputo Corte del Espectro, distinto número microlentes	104
Gráfico L. Relación del tiempo de cómputo con N° lenticulas, Corte del Espectro	104
Figura 5.15. Atributos fotométricos según reenfoque con diferentes algoritmos	105-106
Figura 5.16. Pares Imagen renfocada - LoG para diferentes valores de beta	107

Figura 5.17. Imágenes fusionadas LoG para parámetros beta contiguos	108
Gráfico M. Tiempos de cómputo Foco Volumétrico, diferentes parámetros s_1, s_2	109
Gráfico N. Tiempos de cómputo Foco Volumétrico, distintos pixeles / lentícula	109
Gráfico O. Tiempos de cómputo Foco Volumétrico, distinto número microlentes	109
Gráfico P. Relación del tiempo de cómputo con N° lentículas, Foco Volumétrico	109
Figura 5.18. Foco Volumétrico para diferentes rangos slope1, slope2	110
Figura 5.19. Foco Volumétrico para un rango total 1, -1 de profundidades	110
Figura 5.20. Pares Imagen renfocada - LoG para diferentes valores de delta	111
Figura 5.21. Refocalizaciones con BasicFullRender para diferentes tamaños bloque	113
Gráfico Q. Tiempos de cómputo para algoritmo de renderizado para Plenoptic 2.0, con diferentes tamaños de bloque	114
Gráfico R. Tiempos de cómputo para algoritmo de renderizado para Plenoptic 2.0, con diferentes tamaños de imagen reenfocada	114
TABLA I. Comparación de los Algoritmos	115

Contenido

Introducción	7
1.1. La información tridimensional	9
1.2. El enfoque, la paralaje y la sensación de profundidad.....	10
1.3. Estrategias para captura y visualización de imágenes en 3 dimensiones	10
1.3.1. Visualización 3D a partir de múltiples imágenes.....	11
1.3.2. Visualización 3D a partir de una única imagen	13
1.4. Origen y Estado del Arte de la Imagen Plenóptica	14
1.5. Objetivos de la Tesis.....	17
1.5.1. Objetivo General	17
1.5.2. Objetivos Específicos.....	17
La Imagen Integral	19
2.1. Parametrización de los rayos	20
2.1.1. Parametrización de dos planos paralelos y posición-ángulo	20
2.1.2. Campo de luz y función plenóptica	22
2.2. Estrategias de captura de los rayos.....	23
2.2.1. La cámara de agujero: sus ventajas y limitaciones	23
2.2.2. La cámara de lente convencional.....	25
2.2.3. La cámara plenóptica	27
2.3. Equipamiento disponible en la actualidad	35
2.3.1. Prototipos de Adobe	35
2.3.2. Cámara de Campos de Luz CAFADIS.....	36
2.3.3. Cámara Matricial Pelican.....	37
2.3.4. Pixar Super LightField Lens.....	38
2.3.5. Toshiba LightField Camera Module.....	39
2.3.6. Nvidia Near-Eye Light Field Display	39
2.3.7. MEMScam de DigitalOptics Corporation.....	40
2.3.8. Lytro	40
2.3.9. Raytrix.....	41
2.3.10. Avegant: nueva plataforma de Realidad Mixta.....	42
2.3.11. Detectores de grafeno.....	42
Procesamiento de las Imágenes Plenópticas	43
3.1. Función Plenóptica aplanada y 4D	44
3.1.1. Captura del LF.....	44
3.1.1. Codificación del LF.....	44

3.2.	Renderización de captura original y subaperturas	47
3.2.1.	Imagen en perspectiva para Plenoptic 1.0	49
3.2.2.	Imagen en perspectiva para Plenoptic 2.0	50
3.3.	Algoritmos de Refocalización	52
3.3.1.	Refocalización en el dominio espacial	52
3.4.	Teorema del Corte de Fourier y refocalización	67
	Visualización tridimensional	75
4.1.	La pila focal	76
4.2.	Mapa de profundidades y malla 3D	76
	Evaluación de Resultados	85
5.1.	Diseño del experimento	86
5.2.	Criterios de selección de los algoritmos	88
5.3.	Resultados experimentales	90
5.3.1.	Ensayos Algoritmo Desplazamiento Suma (desplaz_suma)	90
5.3.2.	Ensayos Algoritmo Sesgado de la Radiancia (RefocusRadiancia)	99
5.3.3.	Ensayos Algoritmo Corte del Espectro (RefocusEspectro)	104
5.3.4.	Ensayos Algoritmo Foco Volumétrico (LFBuild4DFreqHyperfan)	108
5.3.5.	Ensayos Algoritmo Renderizado Plenoptic 2.0 por Bloques (BasicFullRender)	112
5.4.	Comparación de los Resultados	114
	Conclusiones	116

Capítulo 1

Introducción

A modo de resumen, en los siguientes párrafos pretendemos capturar la atención del lector sobre el potencial de la fotografía plenóptica. Mediante una cámara fotográfica convencional que incorpora un cambio en la óptica, es posible obtener imágenes que nos permiten aproximarnos a una descripción tridimensional de una escena. Dicho cambio consiste en anteponer al sensor digital una matriz de microlentes que posibilitan a la cámara muestrear la distribución geométrica de la luz que ingresa, en una única exposición. Una vez hecha la captura, este tipo especial de fotografía requiere reordenar y combinar los valores registrados en cada fotosito, para lo cual se necesitan herramientas de cómputo.

El presente trabajo no se basa en el diseño de la cámara sino en el estudio, implementación y evaluación de diferentes algoritmos para el tratamiento de las imágenes capturadas, como un aporte al emergente campo de un tipo de fotografía computacional denominada *fotografía plenóptica*. El término *plenóptica* hace referencia a la distribución geométrica completa de los rayos de luz en el espacio. El desafío es enriquecer a la imagen sumándole dos atributos: enfoque a demanda en diferentes profundidades de la escena y visualización de la misma desde diferentes ángulos, para aprovechar el efecto de la paralaje. Ambos atributos le confieren a la imagen importante información acerca de las propiedades tridimensionales de la escena.

La manipulación de la información provista por la cámara plenóptica, a partir de una única captura, permite la obtención de información de la topografía de la escena. Cabe destacar que, como veremos más adelante, hay un par de estrategias para capturar la escena y existe una relación de compromiso entre la resolución de las imágenes recuperadas y los tiempos de proceso para lograrlo, a tal punto que uno de los dispositivos comerciales incorpora procesamiento paralelo en GPU.

Como ejemplo y para observar las posibilidades que brinda la fotografía plenóptica, en la Figura 1.1 presentamos algunos resultados de los algoritmos implementados. Ubicamos muñecos Lego® sobre una hoja cuadrículada para poder controlar las distancias entre ellos y fotografiamos con una cámara plenóptica comercial denominada Lytro® F01. Cabe destacar que estas imágenes reenfocadas en diferentes profundidades y observadas desde diferentes ángulos se obtienen del procesamiento de una única captura.

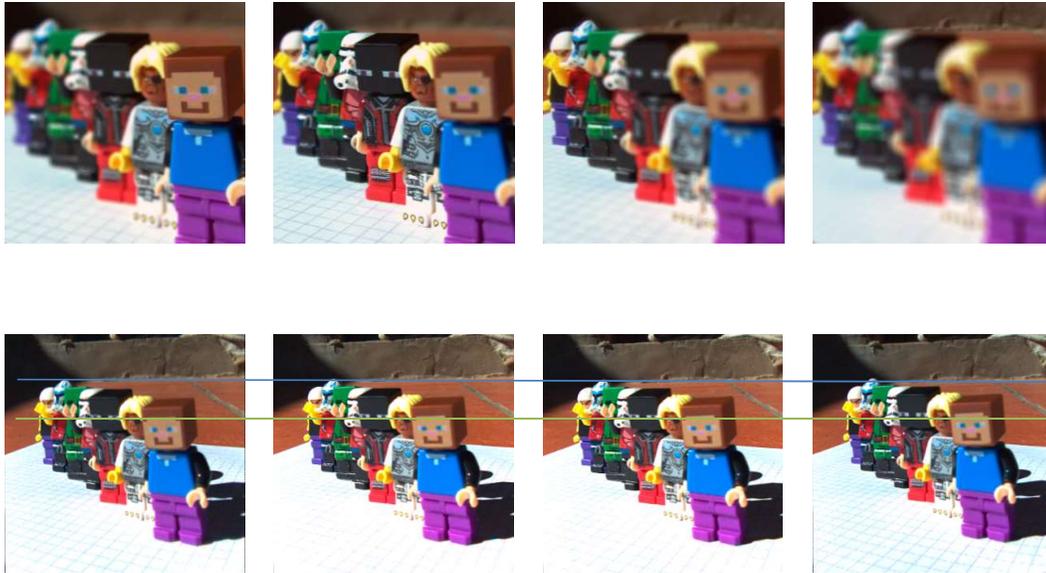


Figura 1.1. Fila superior: reenfoques en diferentes profundidades de la escena. Fila inferior: visualización de la misma escena desde diferentes perspectivas de desplazamiento vertical. La línea azul subtendida sobre todas las imágenes muestra que, a esa profundidad en la escena, no se observan cambios de perspectiva. En cambio, la línea verde delata los desplazamientos del muñeco que se encuentra más cercano a la cámara.

En el presente Capítulo destacaremos la importancia que tiene la información tridimensional en la descripción de una escena, y cómo se fueron ideando desde la invención de la fotografía, diferentes dispositivos para percibir la sensación de profundidad. Justificaremos la elección de la cámara plenóptica como medio para obtener información tridimensional y presentaremos el origen y estado del arte de las imágenes plenópticas, para finalmente definir los objetivos de la presente Tesis.

1.1. La información tridimensional

Nuestra vida se desarrolla en un espacio tridimensional donde las cosas se posicionan, se miden y se explican según descriptores dimensionales espaciales. La visualización de una escena en dicho espacio involucra en el ser humano no sólo a sus ojos como sistemas de lentes y detectores de la luz, sino además mecanismos de percepción visual a nivel neurológico. Los ojos funcionan como par de sensores separados una determinada distancia que, por un método de triangulación que puede explicarse desde la óptica geométrica, perciben profundidad. Este mecanismo denominado estereopsis o visión estereoscópica, fue explicado por primera vez por Charles Wheatstone (1802-1875) [1]. Cada ojo captura en su retina una representación 2D del espacio, a la que denominamos imagen. David Marr [2], uno de los precursores del campo de la neurociencia computacional, define a la imagen mental creada a partir de dos imágenes retinianas como “representación 2.5D”. Pero la escena brinda más información: los objetos tienen diferentes texturas, proyectan sombras y al cambiar la posición de observación percibimos una variación aparente de la posición de los objetos, fenómeno denominado paralaje. Todo esto, junto con el aprendizaje que tenemos del mundo, extienden esta imagen 2.5D a la imagen mental plenamente 3D que nos informa acerca de la distribución y los descriptores arriba mencionados del espacio tridimensional.

La fotografía convencional reduce la información tridimensional a dos dimensiones, superponiendo la información de la dimensión “profundidad” en un plano (el plano focal de la cámara), con la consiguiente pérdida de dicha información 3D. La cámara convencional no registra todas las direcciones de la luz reflejada por la escena, sino la suma total de los rayos que alcanzan cada fotosito del sensor CMOS o CCD. Surge entonces la pregunta: ¿podemos capturar imágenes del mundo sin perder la información de profundidad?

Motiva la realización del presente trabajo la posibilidad de preservar y recuperar información espacial a partir de una toma fotográfica única. El valor agregado de profundidad en la imagen encuentra numerosas aplicaciones, por ejemplo, los citados en los párrafos siguientes. Al comienzo fue un mero objetivo comercial el dotar a los televisores de la capacidad 3D para poder reproducir las películas de la industria cinematográfica, lujo que se limitaba a las salas de cine equipadas con proyectores especiales. Nadie niega la “satisfacción visual” que provocan estos filmes, aunque hay que reconocer que este agregado de información no es imprescindible.

Otras aplicaciones sí convierten a la información 3D en un aliado insustituible. Tal es el caso de los vehículos autónomos que deben monitorear en tiempo real la distancia a los obstáculos. Otro campo de aplicación es la medicina, donde algunas evaluaciones son subjetivas y se basan en la experiencia del médico. Por ejemplo, en la visualización del fondo ocular el especialista va adquiriendo, con práctica, la capacidad de detectar patrones asociados con anomalías topográficas de la retina a partir de las imágenes bidimensionales que le brinda el biomicroscopio. Esta es una valoración subjetiva. Dotar a estas imágenes de un mapa de profundidades sería una valoración objetiva que constituye un valor agregado de alto interés en el diagnóstico del Edema Macular Diabético [3]. Especialidades como la dermatología y la otorrinolaringología pueden también incorporar información tridimensional como soporte para la toma de decisiones [4] [5]. Otro ejemplo médico es la utilización de información de profundidad en robótica para constituir robots quirúrgicos [6].

1.2. El enfoque, la paralaje y la sensación de profundidad

La fotografía es uno de los inventos que ha permitido al hombre dar un salto cualitativo en todos los ámbitos de su vida: histórico, social, artístico, científico e industrial. El registro de una escena, su almacenamiento y hasta su procesamiento, constituyen hoy en día una realidad al alcance de la mayoría de las personas y resulta difícil imaginar al mundo sin la fotografía. Han pasado 178 años desde su descubrimiento por Louis Daguerre (1787-1851) y 175 años desde que el inventor inglés William Fox Talbot (1800-1877) obtuvo el primer registro fotográfico en papel. De aquellos tiempos hasta nuestros días, las mejoras en la fotografía han sido sustanciales y los avances de la electrónica le han dado una impronta tal que una cámara fotográfica o de video puede anexarse a casi cualquier dispositivo: teléfonos celulares, laptops, telescopios, entre tantos otros.

El hombre puede explorar una escena del mundo real tridimensional centrando su atención en diferentes puntos de interés y modificando la posición de observación para describir, por paralaje, la ubicación relativa de los objetos en el espacio. La Real Academia Española define a la paralaje como la “variación aparente de la posición de un objeto, especialmente un astro, al cambiar la posición del observador”. Entonces mediante movimientos de la cabeza el observador puede hacerse una idea cabal de las profundidades en la escena. Los objetos lejanos parecen moverse más lentamente que los objetos cercanos, y según vaya desplazándose el observador van descubriéndose aquellos que en otra posición quedaban ocluidos por los más cercanos. Cuando ve la misma escena en una fotografía ya sea en papel o en una pantalla, estas posibilidades están ausentes y el observador debe recurrir a otras pistas para inferir distancias. Estas pistas son, entre otras, perspectiva lineal, oclusiones, sombras, variaciones de luminancia y textura [7] y la interpretación que el cerebro hace de las mismas.

Otra de las pistas que utiliza el sistema visual humano para interpretar las profundidades en una escena se basa en los enfoques en la fotografía. Haciendo foco en diferentes puntos de la escena, el resto de nuestro sistema visual puede inferir la posición relativa de las cosas, por delante y por detrás del plano focal. Si se hace un “barrido” del foco en distintas profundidades de la escena, van apareciendo los detalles en los distintos planos focales mientras se mantiene difuso el resto de la escena y la interpretación de las profundidades mejora. Lo cierto es que focalizar exactamente en la profundidad deseada no es una tarea trivial y una vez hecha la captura, si el dispositivo es una cámara fotográfica convencional, no hay chances de cambiar a posteriori dicho plano focal.

La Figura 1.1 nos presenta las pistas antes mencionadas: reenfoque en diferentes profundidades y vistas en perspectiva de la escena.

1.3. Estrategias para captura y visualización de imágenes en 3 dimensiones

En las cámaras fotográficas digitales convencionales los elementos sensores del CCD o CMOS integran la luz que les llega desde una pequeña zona definida por el campo visual de cada fotosito, de acuerdo a la óptica que se esté usando, como veremos en detalle en el Capítulo 2.

Entonces cada fotosito acumula la cantidad de luz reflejada desde estas pequeñas zonas, lo que resulta en una captura 2D de una escena tridimensional, lo que se define como fotografía simple.

La extracción de información tridimensional de una imagen bidimensional consiste en la recuperación de los descriptores dimensionales espaciales perdidos por el mapeo de la escena 3D al plano de la imagen. Para obtener una representación tridimensional existen numerosos métodos basados en diferentes técnicas, que podrían clasificarse como reconstrucción o visualización 3D a partir de múltiples imágenes o a partir de una única imagen [8].

1.3.1. Visualización 3D a partir de múltiples imágenes

La estereoscopia es la técnica más difundida y estudiada para la visualización tridimensional y cálculo de profundidades. Se atribuyen al matemático griego Euclides que vivió entre los años 330 y 275 a.C. los primeros escritos donde se reconoce que la percepción de profundidad se debe a la visión binocular, por triangulación de la posición de sendos ojos. Leonardo Da Vinci (1452-1519) dejó manuscritos sobre sus estudios del fenómeno de la visión binocular [9] (Figura 1.2). Los desarrollos en visualización estéreo se iniciaron con el inglés Sir Charles Wheatstone (1802-1875) que describió el “fenómeno de la visión binocular” [1] y creó el estereoscopio (Figura 1.3.A), un dispositivo que mediante espejos ubica en la retina de cada ojo una imagen ligeramente desplazada. Esta doble imagen es procesada por el sistema visual como una sola imagen tridimensional. Sir David Brewster (1781-1868), contemporáneo de Wheatstone, propuso en 1845 una mejora (Figura 1.3.B) que tuvo su competencia en EEUU con el estereoscopio de Oliver Holmes (1809-1894) en 1859 (Figura 1.3.C). Este aparato fue evolucionando en el tiempo, pero bajo el mismo principio de funcionamiento, dando lugar a varios dispositivos (Figura 1.3.D). Una revisión histórica muy interesante puede consultarse en [10].

Cabe destacar que este sistema basado en un par de imágenes desplazadas depende de los mecanismos centrales de la visión para generar la sensación tridimensional.

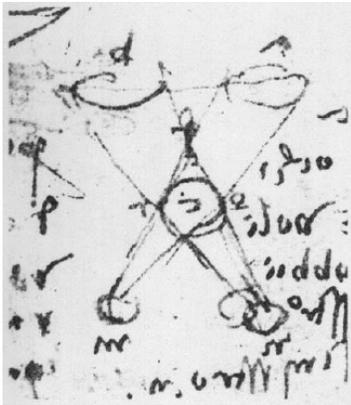


Figura 1.2. Manuscrito de Leonardo Da Vinci, sobre visión binocular.
Fuente: [9].

Una alternativa es imitar la estereoscopia del sistema visual utilizando un par de cámaras separadas una distancia (así como están separados los ojos) y recurrir al cálculo de profundidades y la reproducción de las propiedades tridimensionales de la escena por triangulación. Para ello se requiere la captura de la escena desde dos posiciones diferentes, por lo que supone una cámara que toma dos imágenes consecutivas desde dos posiciones diferentes, o la presencia de dos

cámaras montadas en un sistema calibrado, cuyas dimensiones conocidas permiten los cálculos. Estas cámaras capturan el par de imágenes simultáneamente y se requiere identificar puntos de correspondencia entre las dos imágenes.

Los métodos de visión estéreo como disciplina de la Visión por Computadora vienen estudiándose hace más de 50 años. Uno de los primeros investigadores en investigar el problema de las ambigüedades y oclusiones en la correspondencia fue el húngaro Bela Julesz (1928-2003) [11] y hasta el día de hoy continúan perfeccionándose los métodos, traccionados por los desafíos que impone la navegación autónoma de robots y vehículos.

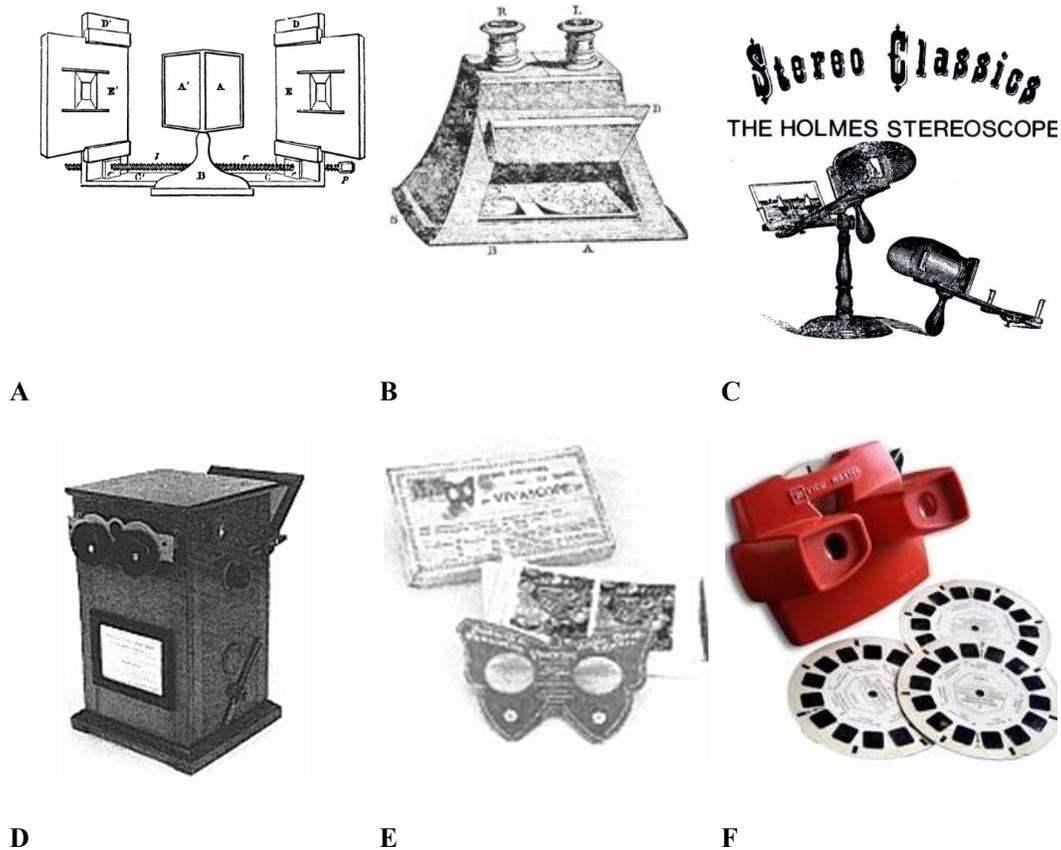


Figura 1.3. A: Estereoscopio de espejos reflectantes de Sir Charles Wheatstone (1833). B: Estereoscopio mejorado de Sir David Brewster (1845). C: Estereoscopio de Holmes (1859). D: Plastoscopio de Krügener (1899). E: Vivascopio de Harry Cox (1920). F: View_Master, introducido en 1939. Fuente: [10].

Una alternativa al uso de dos cámaras en un sistema calibrado es la cámara estereoscópica, que utiliza en un mismo cuerpo dos objetivos y captura en simultáneo un par de imágenes desplazadas. Apareció unos 10 años después de inventarse la fotografía en 1839 y fue utilizada hasta la década del 50 del siglo XIX.

Con una cámara convencional se presentan dos opciones para reconstruir 3D, siempre realizando capturas consecutivas en el tiempo. La primera se basa en múltiples capturas de la misma escena con distancia focal fija, con diferentes distancias plano objeto – plano imagen. Esto posibilita la

reconstrucción 3D por el método denominado seccionamiento óptico, muy utilizado en microscopía [12], [13]. La segunda opción requiere múltiples capturas de la misma escena desde ángulos diferentes, en tomas sucesivas. Esta técnica se denomina “Structure from Motion” (SfM) y requiere como mínimo, tres imágenes para hacer la reconstrucción 3D. Existen varias líneas de investigación, principalmente en la Universidad de Stanford [14] y el Instituto Tecnológico de Massachusetts [15].

La única manera de obtener múltiples imágenes desde diferentes vistas con un único disparo es utilizando una matriz de cámaras. Tal es el caso de la matriz de 128 cámaras de video del Laboratorio de Computación Gráfica de Stanford, dispositivo que generó una profusa producción [16], [17].

1.3.2. Visualización 3D a partir de una única imagen

Basados en una única fotografía convencional y haciendo uso de un conocimiento previo de las características de la escena, es posible reconstruir dicha escena 3D a partir de su imagen 2D si se conocen texturas, formas y tamaños, oclusiones, puntos de fuga, etc., mediante los cuales pueden inferirse profundidades [8]. Hay varias propuestas, pero todas se limitan a escenas del mundo donde hay pistas tales como líneas de horizonte, cielo, puntos de fuga, y las reconstrucciones logradas guardan poco detalle de la escena.

Otras técnicas aprovechan nuestra visión binocular y su capacidad estereoscópica. Los TV 3D actuales hacen uso de la estereoscopia, presentando al observador en una única pantalla dos imágenes ligeramente desplazadas, que pueden estar organizadas de diferentes maneras: contiguas o intercaladas. Para mostrar estas imágenes de manera independiente a cada ojo, se requieren anteojos especiales denominados “pasivos”, que seleccionan o bloquean cada imagen usando colores complementarios o polarizadores, o sistemas de anteojos “activos” de obturación electrónica. Si el sistema no requiere gafas especiales se denomina autoestereoscópico, enviando una imagen diferente, ligeramente desplazada, a cada ojo. Esto puede lograrse, por ejemplo, anteponiendo a la imagen un dispositivo similar a una grilla, denominado barreras de paralaje, que ocluye para un ojo algunas regiones mientras deja pasar otras al otro ojo. También pueden utilizarse lenticulas, como podemos observar en la Figura 1.4. Fue Frederic Ives (1856-1937) [18] quien investigó estas estrategias, a principios del siglo pasado. El problema con estos sistemas es el cansancio visual que provocan, dado que someten a los ojos a un conflicto entre acomodación de cristalinicos y convergencia de los globos oculares.

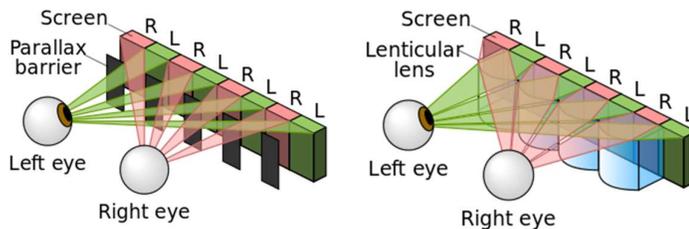


Figura 1.4. Barreras de paralaje y lenticulas, como medios para obtener visión autoestereoscópica de una imagen.

Fuente: Cmglee - Trabajo propio, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=14886721>

Otra alternativa es la holografía, una técnica fotográfica que, mediante iluminación por láser, permite obtener imágenes tridimensionales en color. Este método proporciona la mejor

experiencia 3D, pero requiere instrumentación específica y ambientes controlados por lo que su uso no es simple. El estudio en detalle de esta técnica escapa a los objetivos del presente trabajo.

En 1908 el Profesor Gabriel Lippmann (1845-1921) propuso un método basado en una matriz de lentes [19], [20], [21] y obtuvo imágenes que debían reproducirse luego anteponiendo nuevamente dicha matriz. Denominó a su técnica Fotografía Integral puesto que el resultado es la integración de todas las imágenes producidas por las pequeñas lentes en una imagen completa tridimensional. Este desarrollo presentaba inconvenientes técnicos para su implementación y quedó latente hasta el desarrollo de las computadoras y los sensores CCD. Edward Adelson y John Wang redescubren en 1992 esta técnica, a la que bautizan Fotografía Plenóptica [22]. Este nuevo tipo de fotografía permite, a partir de una única captura, reconstruir imágenes de la escena como si fuera observada desde diferentes puntos de vista y con foco en distintas profundidades. La cámara plenóptica captura el “campo de luz”, definido como la radiancia de los rayos de la escena en el espacio. La función que describe la intensidad de cada rayo de luz como una función de su posición y dirección se denomina “función plenóptica” y se tratará en detalle en el Capítulo 2.

La fotografía plenóptica requiere una matriz de microlentes antepuesta al sensor de imagen y un sistema de cómputo para realizar el postproceso. Se verá más adelante que existen cámaras plenópticas dedicadas, pero es factible obtener imágenes de campos de luz a partir de una cámara fotográfica convencional y un acople óptico con un arreglo de microlentes.

Ya sea bajo la modalidad de cámara plenóptica dedicada o cámara convencional con acople óptico de microlentes, la posibilidad de obtener con una única cámara y una única captura información acerca de la tridimensionalidad de la escena es una buena alternativa. En comparación con las opciones anteriores de múltiples capturas, dispositivos tales como estereoscopios o la necesidad de gafas especiales, la fotografía plenóptica se sitúa en un lugar de ventaja competitiva para la visualización 3D. Asimismo, la aparición constante de sensores de cada vez mayor resolución y la posibilidad de proceso en tiempo cuasi real en GPUs en reemplazo de computadoras, hacen de la fotografía plenóptica una técnica de imagen tridimensional muy promisoría.

1.4. Origen y Estado del Arte de la Imagen Plenóptica

Fue el Profesor Gabriel M. Lippmann quien propuso el 2 de marzo de 1908 frente a científicos de la Academia Francesa de las Ciencias, las bases teóricas para capturar una escena y reproducirla para brindar una sensación tridimensional [20], técnica que bautizó como “fotografía integral”. Asimismo, Lippmann inventó la reproducción de fotografías en color en 1886 y por este motivo recibió en 1908 el Premio Nobel, coincidentemente con su propuesta de Fotografía Integral.

Los avances tecnológicos de la época sólo permitían trabajar con emulsiones fotográficas sobre sustratos de acetato o vidrio. Lippmann conformó un sistema de lentes al termoformar dos láminas de acetato con un relieve esférico, en una de las cuales depositó emulsión fotográfica. Los radios de curvatura de una y otra lámina eran diferentes (Figura 1.5) de modo que cada protuberancia esférica se comportaba como una lente convergente que proyectaban “células” de imagen sobre las protuberancias opuestas recubiertas con la emulsión. Lippmann comparó esta estructura con los ditiscidos, una familia de coleópteros cuyo sistema visual está constituido por un gran número de córneas transparentes y mini retinas.

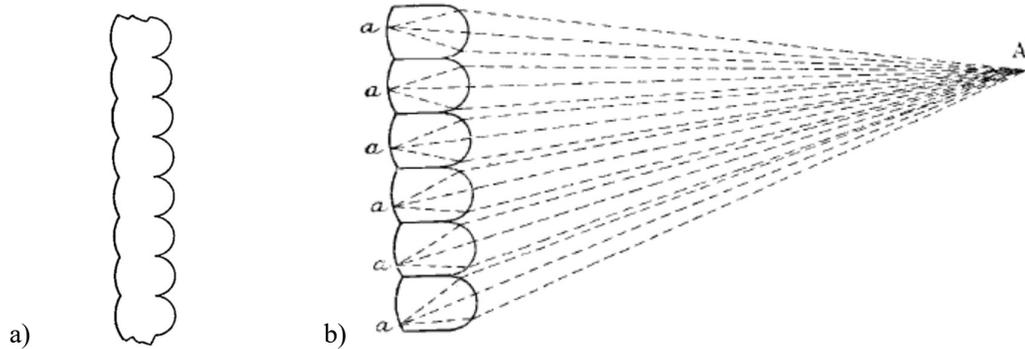
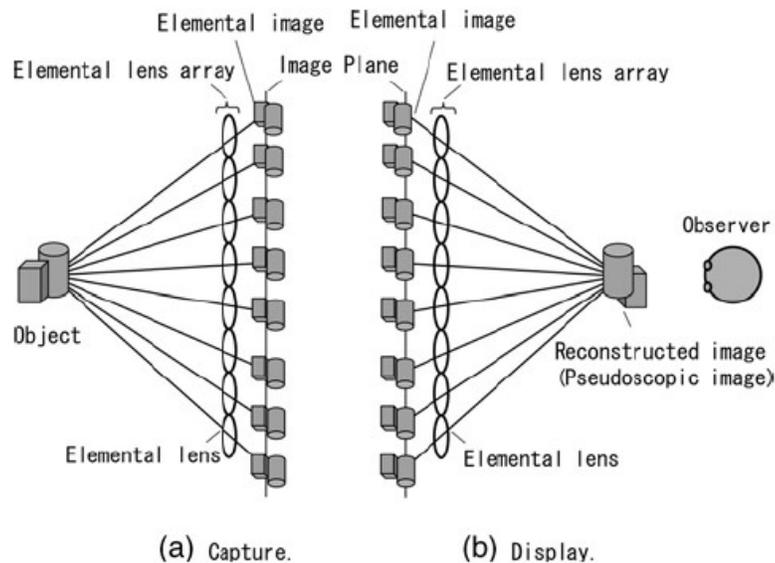


Figura 1.5. Dibujo original de Lippmann. **a):** Película de celuloide o colodión moldeada térmicamente con muescas esféricas en ambos lados, con emulsión fotosensible en la muesca de menor radio. **b):** Esquema original que muestra cómo un punto A se distribuye en las múltiples células. Fuente: [20].

Para la reproducción de la imagen capturada puede utilizarse el mismo sistema gracias a la reversibilidad del camino de los rayos luminosos. La fotografía integral constituye un verdadero método autoestéreo, es decir, no requiere colocarse gafas especiales sino anteponer a la película revelada una matriz de lentes. Si se retroilumina la placa fotográfica especial (Figura 1.6.B) y se observa desde el lado de las muescas de menor radio, es posible visualizar la imagen con sensación de profundidad cuando se cambia levemente la perspectiva de visión moviendo la cabeza de lado a lado. Uno de los inconvenientes es que la película revelada es negativa y que la imagen se ve volteada de izquierda a derecha y de arriba a abajo, por lo que Lippmann debió recurrir a una doble rectificación, fotográfica y geométrica. El mismo año describió una comprobación experimental de su propuesta con unos bastones de vidrio en lugar del colodión termoformado, obteniendo mejores resultados [19]. Pero no fue hasta 1911 que Lippmann pudo verificar de manera teórica su propuesta, cuando utilizó 12 bastones de vidrio rectangulares de 6 mm por 2.5 mm, con un relieve esférico de un lado y plano del otro [21]. Después de la toma fotográfica, el negativo se procesa como de costumbre y se invierte, de manera que se obtiene un positivo transparente. Las doce imágenes formadas son diferentes, pues cada lente forma su imagen desde un punto de vista diferente. El positivo transparente se coloca donde estuvo el film fotosensible original y se ilumina a través de un difusor por detrás, luz que atraviesa el conjunto positivo-lentes. El observador mira a través de las lentes con los dos ojos y observa una sola visión en relieve del objeto fotografiado. Al mover la cabeza de lado a lado o de arriba a abajo, el mismo efecto se observa como si fuera en circunstancias similares cuando se mira el objeto real, es decir, objetos que se cubren unos a otros cuando se miran desde un punto, son vistos separados cuando se observan desde otra posición. Incluso con este simple aparato, el efecto es muy agradable. Lippman expresó: “al iluminar la placa, uno ya no ve imágenes microscópicas individuales; ellas son reemplazadas por una única imagen (integral), que se ve desde el mismo ángulo que el sujeto original” (“*in illuminating the plate one no longer sees individual microscopic images; they are replaced by a single (integral) image, which is seen under the same angle as the original subject*”) [21]. El 19 de agosto de ese mismo año, la revista Scientific American publicó su descubrimiento y fotografías de su cámara [23]. La Figura 1.6 es un esquema del dispositivo.

Figura 1.6. Esquema de funcionamiento del sistema autoestéreo de Lippmann. Fuente: modificado de [21].



Dados los inconvenientes técnicos para fabricar la matriz de lentes, P. P. Sokolov [24] utilizó una matriz de estenopes, y realizó una detallada comprobación matemática y experimental del método de Lippmann. Los estenopes (Figura 1.7) son pequeños orificios de perfil cónico de un diámetro de décimas de milímetro, sencillos de maquinar, pero presentan el problema de la escasa luz que dejan pasar. Sokolov capturó entonces la imagen del filamento incandescente de una lámpara, que pudo observarse luego como “flotando” en el espacio.

Tanto la propuesta de Lippmann como la de Sokolov tenían otro problema: las imágenes proyectadas son pseudoscópicas, esto es, están invertidas en profundidad observándose como más cercanos los objetos que en realidad están más lejanos en la escena. En 1930 Herbert Ives (1882-1953, hijo de Frederic Ives) estudió este problema y además de proponer soluciones, su aporte más significativo fue usar una lente de gran diámetro para formar la imagen sobre los estenopes [25], [26].

A partir de H. Ives varios investigadores hicieron sus propuestas. Clarence Kanolt patentó en 1934 un sistema basado en lente objetivo y una matriz de estenopes [27]. Más tarde en 1935, Coffey combina los conceptos de Lippmann e Ives y gracias a la gran lente, la matriz de lentes puede ser plana del lado opuesto a la escena. El problema de la superposición de las imágenes elementales (“células” de Lippmann) se resuelve controlando la apertura numérica de ambas ópticas [28]. El inventor Douglas Winnek patentó en 1936 una cámara integral con refinamientos y mejoras al dispositivo patentado por Kanolt en 1933. Los franceses Granont y Planovern propusieron una matriz de espejos planos en lugar de microlentes o múltiples cámaras oscuras [29]. Fueron muchas las investigaciones, patentes y prototipos de cámaras integrales basados en la propuesta original de Lippmann; una interesante revisión puede consultarse en [30] y en [31]. Pero recién en los años 80 comenzaron a producirse matrices de pequeñas lentes en serie, denominadas matrices de microlentes (MML), lo que disparó el interés por las imágenes plenópticas. Las MML actuales pueden contar con lenticulas de diámetros tan pequeños como 10 micrones, como veremos en el Capítulo 2.

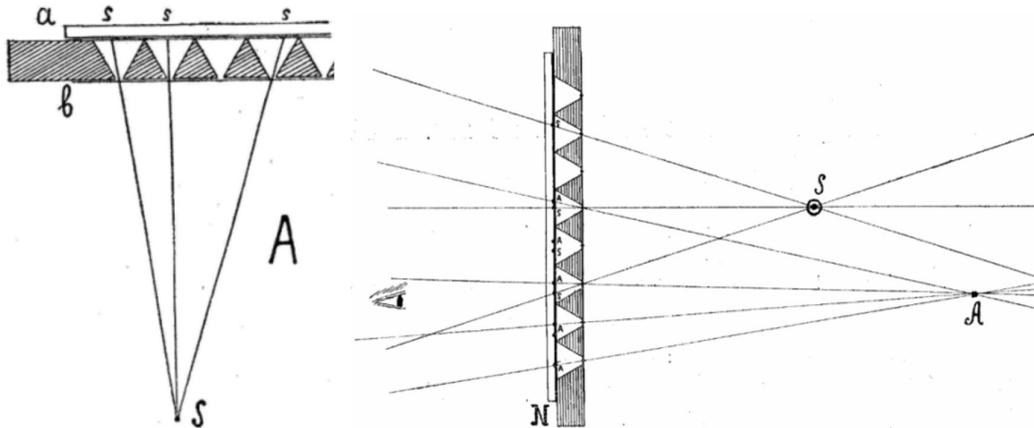


Figura 1.7. Dibujo original de Sokolov. Izq.: proceso de grabación de un punto luminoso a través de una matriz de estenopes con estructura cónica sobre una película fotográfica. Der.: Reconstrucción de la imagen tridimensional a través de una matriz de estenopes. Fuente: [24].

El trabajo que puso nuevamente sobre el tapete la fotografía integral fue el de Edward Adelson y John Wang [22], quienes presentaron la “cámara plenóptica” basada en la antigua propuesta de Lippmann y sustentada en las posibilidades que brindaban una MML, un sensor de imagen CCD y las capacidades de cómputo disponibles. Un año antes, el mismo Adelson junto con James Bergen habían formalizado la “función plenóptica” [32]. Dicha función será estudiada más adelante, pero es oportuno comentar que describe el conjunto de todos los rayos de luz que ocupa todo el espacio durante todo el tiempo. La insistencia de la “totalidad” de esta función es la razón de su nombre: óptica plena, completa, llena. Con esta función, Adelson y Bergen buscan representar una escena no como una superposición de superficies planas correspondientes a diferentes profundidades, sino más bien como un volumen desde el que fluyen rayos en todas las direcciones del espacio.

Desde Adelson hasta nuestros días, los avances de las cámaras plenópticas fueron de la mano de los avances de la microóptica y las capacidades de los sensores y cómputo de los procesadores. En el Capítulo 2 describiremos brevemente los desarrollos recientes más importantes.

1.5. Objetivos de la Tesis

1.5.1. Objetivo General

Motivados por lo presentado en los párrafos precedentes, nos proponemos como objetivo general de esta tesis generar algoritmos para la reconstrucción de imágenes a partir de su campo de luz y evaluar su performance.

1.5.2. Objetivos Específicos

Para alcanzar el objetivo general antedicho es necesario cumplimentar un conjunto de metas, que planteamos como objetivos particulares:

1. Estudiar las técnicas disponibles para la reconstrucción de imágenes a partir de capturas con cámaras plenópticas.
2. Generar las matrices imagen correspondientes a campos de luz de acceso público, para su manipulación en un entorno de programación.
3. Desarrollar algoritmos para representar imágenes correspondientes a campos de luz.
4. Evaluar los resultados obtenidos.

Capítulo 2

La Imagen Integral

En el presente Capítulo estudiaremos las estrategias para la captura del campo de luz de la escena a partir de una única toma fotográfica.

Comenzaremos observando cómo a partir de las cámaras de agujero y de lente simple se forma la imagen en el sensor, con las ventajas y desventajas de cada caso. Estudiaremos alternativas de modelización de los rayos en el espacio que permitan representar la posición y dirección de los mismos. Veremos diferentes estrategias de captura de la información espacio-angular de una escena tridimensional y describiremos dos modelos de cámara plenóptica, con sus ventajas y desventajas. Cada una provee una codificación diferente del campo de luz. Pretendemos en esta parte de la Tesis comprender cómo se codifica y representa un campo de luz.

Finalmente, presentaremos algunas de las aproximaciones tecnológicas para la captura y representación de imágenes, unas disponibles comercialmente y otras como prototipos en desarrollo.

2.1. Parametrización de los rayos

En primer lugar, es conveniente repasar algunos conceptos de la óptica a los efectos de acotar el marco teórico.

La *radiometría* describe la transferencia de energía desde una fuente a un receptor, para todas las longitudes de onda del espectro electromagnético. Si dicha transferencia se normaliza a la respuesta espectral del ojo humano, se denomina *fotometría*. Es importante tomar en consideración esta diferencia cuando se trabaja con imágenes para diagnóstico médico, por ejemplo, dado que algunas modalidades operan en regiones no visibles del espectro y deben realizarse conversiones porque, en última instancia, importa la interpretación que el sistema visual humano hace de la escena.

La magnitud básica de la radiometría es la *radiancia*, propiedad de una superficie emisora de radiación electromagnética, sea que emerja de ella o se refleje. Se define como el flujo radiante emitido (en vatios) por unidad de longitud de onda (en micrones), por unidad de ángulo sólido (estereorradianes) y por unidad de área (metros cuadrados) proyectada sobre un plano normal a la dirección del rayo. La radiancia es una magnitud vectorial (depende del punto de observación) y se conserva con la propagación libre de oclusiones. Su par en fotometría es la *luminancia*, propiedad de las superficies emisoras de luz que se mide como intensidad luminosa (en candelas) por unidad de superficie en una dirección dada (en metros cuadrados). El equivalente desde el punto de vista de la percepción visual de la luminancia es el brillo.

La *irradiancia* es el flujo radiante que incide sobre una superficie y se mide en vatios por metro cuadrado (W/m²). La irradiancia es la magnitud radiométrica que “leen” los sensores. Su par fotométrico es la *iluminancia*, que se refiere a la recepción de la luz y se mide en cada punto de una superficie, en lúmenes por metro cuadrado o lux. En un CCD o CMOS de una cámara digital, la iluminancia se convierte en corriente eléctrica (mA) proporcional a los lux recibidos por cada elemento del sensor denominado fotosito y se somete a un procesamiento electrónico. Los valores analógicos originales se convierten en digitales y se construye la matriz imagen, que es el conjunto de los elementos de imagen denominados píxeles, cuyos valores numéricos representan la iluminancia original.

Para estudiar las transformaciones que sustentarán las estrategias para la codificación de los campos de luz y extracción de imágenes, se tendrá en cuenta el modelo de rayos de la luz (no el de onda ni el de partícula). En este marco, la luz se propaga en línea recta (rayo de luz) y satisface las propiedades geométricas de la reflexión y refracción. A los efectos de modelizar matemáticamente los sistemas ópticos, es necesario definir los parámetros que describen unívocamente a un rayo.

2.1.1. Parametrización de dos planos paralelos y posición-ángulo

Existen varios modelos de parametrización de los rayos. Los más difundidos son el de posición-ángulo (PA) utilizado por la mayoría de los textos de óptica geométrica y el de dos planos paralelos (2PP) de Levoy y Hanrahan [33]. En la Figura 2.1 podemos comparar cómo puede definirse un rayo según cada parametrización. Es importante remarcar que los planos en todos los casos son normales al eje óptico. En la parametrización PA un rayo atraviesa a un plano por un punto y su proyección en el plano forma un ángulo ϕ respecto al eje y del sistema coordenado de referencia. Otro ángulo θ del rayo con respecto al plano termina de definir totalmente a dicho

rayo, que puede representarse por el punto (x,y) y la dupla de ángulos (ϕ, θ) . En la parametrización 2PP el rayo queda definido por los puntos de intersección con 2 planos normales al eje óptico, separados una determinada distancia.

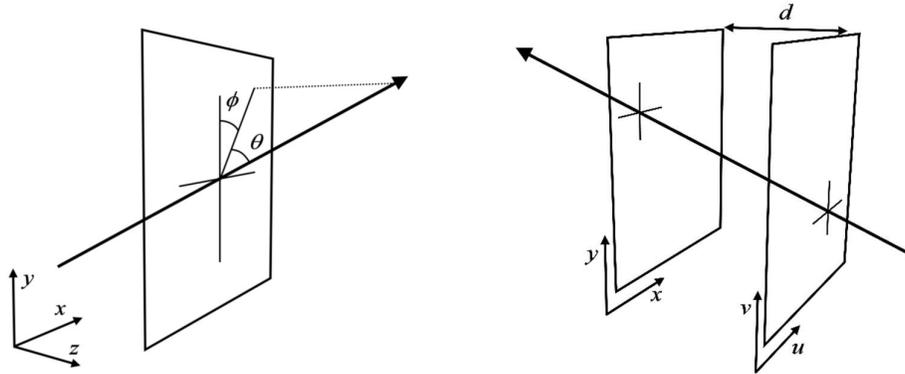


Figura 2.1. Parametrización de los rayos. Izq.: Posición-Ángulo (PA). Der.: Dos Planos Paralelos (2PP).

Una simplificación del esquema de la Figura 2.1 nos permite representar a los rayos en un diagrama 2D. Para el caso de la parametrización PA, existe un sistema coordenado de referencia xyz donde el eje óptico coincide con el eje z y la ordenada es el eje y (también puede ser el eje x) del plano $x-y$. Para el caso de la parametrización 2PP, nuevamente el eje óptico se asigna a la abscisa y los planos paralelos se muestran como los ejes x y u de cada plano (también pueden ser los ejes y y v), que están separados una distancia d . Denominamos con la letra q a los puntos posición y con la letra p a la representación del ángulo como pendiente. En la Figura 2.2 observamos los esquemas.

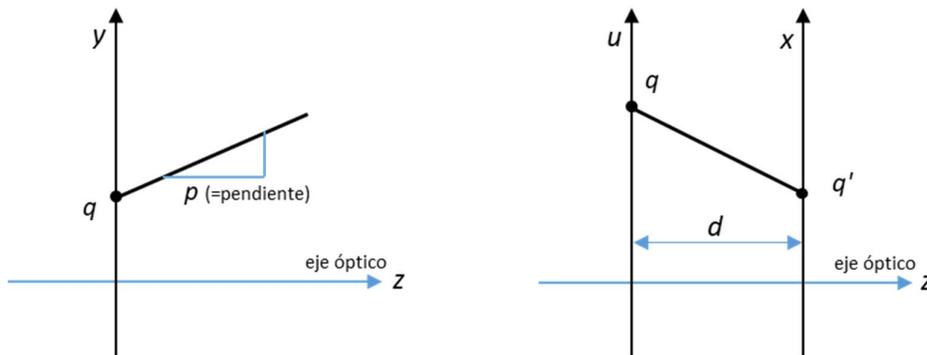
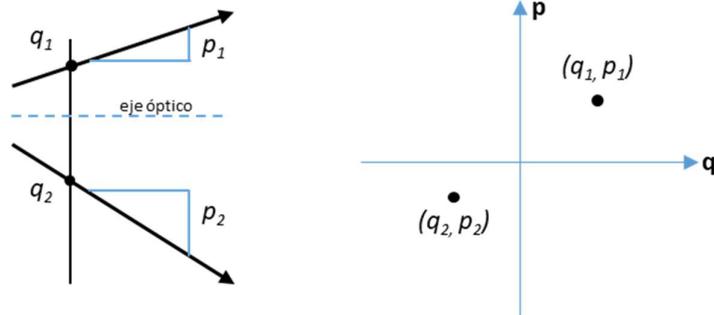


Figura 2.2. Parametrización de los rayos. Izq.: el rayo atraviesa al plano en la posición q con una pendiente p en el diagrama simplificado 2D, para la parametrización posición – ángulo (PA). Der.: el rayo atraviesa dos planos paralelos (2PP) por las posiciones q y q' en el diagrama simplificado 2D.

Podemos ir un paso más allá. El denominado “espacio de las fases” es la representación cartesiana de la parametrización PA y en este espacio de las fases un punto, cuyas coordenadas son posición q y ángulo p , representa un rayo. En la Figura 2.3 observamos dos rayos en diferentes posiciones q y con diferentes pendientes p .

Figura 2.3. Espacio de las fases q - p (derecha), correspondiente a dos rayos presentados en el diagrama simplificado 2D de la parametrización PA (izquierda).



De manera similar, podemos representar la parametrización 2PP en el “espacio de los rayos”. Por ejemplo, si dos rayos atraviesan los 2PP en diferentes puntos, estos rayos se observan como un par de puntos como se esquematiza en la Figura 2.4.

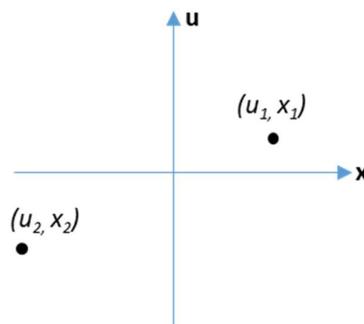
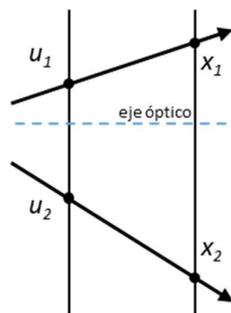


Figura 2.4. Espacio de los rayos x - u (derecha), correspondiente a dos rayos presentados en el diagrama simplificado 2D de la parametrización 2PP (izquierda).

¿Por qué nos interesa conocer a los rayos por sus atributos de posición y dirección? La respuesta está en el concepto de “campo de luz” que nombramos fugazmente en el capítulo anterior y que de aquí en adelante nombraremos con la sigla LF (del inglés “*light field*”).

2.1.2. Campo de luz y función plenóptica

Definido originalmente por Gershun [34], el LF es la cantidad de luz como función de la posición y dirección en el espacio, libre de oclusores. Representa el conjunto de todos los rayos de luz que ocupa el espacio.

La función que describe al LF es la función plenóptica, definida en 1991 por Adelson y Bergen [32]. En 1992 Adelson y Wang proponen la “cámara plenóptica” [22] basados en los trabajos de Ives y Lippman a los inicios del 1900 e impulsados por el acceso a sensores digitales de mayor

número de píxeles y matrices de microlentes más pequeñas, que estudiaremos más adelante en el presente Capítulo.

La función plenóptica es intermediaria entre la escena y el sistema óptico. Representa los parámetros de la energía radiante del entorno y comunica los objetos físicos con los sensores de imagen. Es una función que describe el conjunto de todos los rayos de luz que ocupa el espacio durante todo el tiempo t , en otras palabras, contiene la información visual disponible para un observador en cualquier punto en el espacio. La insistencia de la “totalidad” de esta función es la razón de su nombre: óptica plena, completa, llena. La función plenóptica describe la luz en 7 dimensiones: posición en el espacio (3D), dirección (2D), frecuencia (1D) y tiempo (1D). La representación posición-ángulo esquematizada en la Figura 2.1 indica que un plano normal al eje óptico, coincidente con el eje z de un sistema coordenado de referencia xyz , es intersecado por un rayo en un punto (x,y,z) y forma dos ángulos con él (ϕ, θ). Con esta función, Adelson y Bergen buscan representar una escena como un volumen desde el que fluyen rayos en todas las direcciones del espacio. Cada punto en este espacio 7 dimensional corresponde a un único rayo de luz. Ahora bien, si consideramos imágenes estáticas podemos prescindir del parámetro temporal t y si trabajamos con imágenes de intensidad, tampoco es necesario considerar la longitud de onda λ que describe el color del rayo. En todo caso, utilizando un modelo de color RGB podemos procesar cada canal por separado para después concatenarlos y recuperar la cromaticidad. Con estas consideraciones la función plenóptica puede reducir su multidimensionalidad a 5D, que representan la posición espacial del rayo (x,y,z) y los ángulos (ϕ, θ) que definen su dirección en el espacio. A posteriori y con la parametrización 2PP, Levoy y Hanrahan [33] redujeron a 4D la función plenóptica para referirse al campo de luz.

Nos interesa conocer, luego de una captura, la composición del LF de una escena que, como veremos de aquí en adelante, nos revelará información de los atributos tridimensionales de dicha escena.

2.2. Estrategias de captura de los rayos

Comprendidas las parametrizaciones de los rayos de luz y la necesidad de conocer sus posiciones y direcciones, vamos a estudiar cómo los rayos forman la imagen según las diferentes arquitecturas de las cámaras: de agujero, convencional de lentes y plenóptica, destacando sus ventajas y desventajas.

2.2.1. La cámara de agujero: sus ventajas y limitaciones

La cámara de agujero llamada también cámara estenopeica, consiste en una caja estanca a la luz, en una de cuyas caras hay sólo un pequeño orificio circular y en la cara opuesta, un sensor de imagen. Cuanto más pequeño es el agujero, más nítida es la imagen, con el recaudo de no llegar a dimensiones tales que se ponga en juego el límite de difracción. Para tener una idea aproximada, una apertura del orden de 0,5mm de diámetro es adecuada para producir una imagen de buena calidad luego de una exposición de no menos de 5 segundos. En la Figura 2.5 presentamos un esquema de cómo un rayo proveniente de un punto de la escena impacta en un punto del sensor luego de atravesar el agujero.

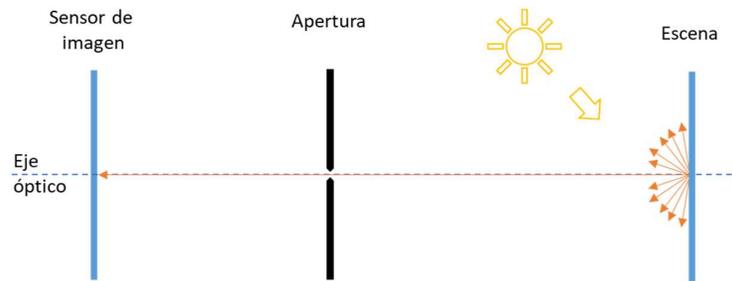


Figura 2.5. Esquema de una cámara de agujero.

Si asignamos al sensor de imagen un plano $x-y$ y a la cara de la cámara donde está la apertura, un plano $u-v$, paralelo al anterior y ambos normales al eje óptico, podemos representar a los rayos mediante la parametrización 2PP. Entonces el rayo que atraviesa al plano en la apertura por el punto (u,v) e impacta en el punto (x,y) del plano del sensor, corresponde al punto (x,u) en el espacio de los rayos. En la Figura 2.6 presentamos el esquema de la cámara de agujero y el espacio de los rayos correspondiente a dicho esquema. Cada rayo de luz se convierte en un punto en la imagen. Dado que el plano del sensor es donde se forma la imagen de la escena, habitualmente se lo denomina “plano imagen”.

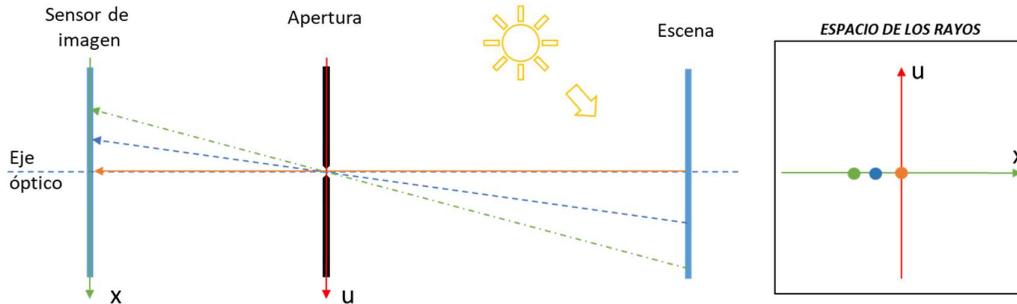


Figura 2.6. Espacio de los rayos correspondiente a una cámara de agujero, cuando provienen de un mismo plano a igual distancia escena-apertura.

Si hubiera cambios de profundidad en la escena, la posición (x,y) de los puntos en el plano imagen puede hallarse por una simple relación de triángulos semejantes, conocida la distancia entre el plano imagen y el plano de la apertura, denominada distancia focal. En la Figura 2.7 representamos el caso de un punto P en el espacio XYZ , que refleja un rayo que impacta en el punto (x,y) del plano imagen. Entonces $x=fX/Z$ e $y=fY/Z$ lo que significa que es sencillo calcular las distancias a las que se encuentra cada punto en la escena. Pero más allá de la profundidad Z a la que se encuentre, el punto en la escena siempre genera un punto en la imagen. Esto significa que la cámara de agujero genera imágenes en foco en todas las profundidades, “*all in focus*” en inglés, y queda claro que la cámara realiza un mapeo de 3D a 2D.

El espacio de los rayos revela que no importa la distancia de la escena, la información en el eje u siempre será cero dada la naturaleza de la apertura de la cámara estenopeica.

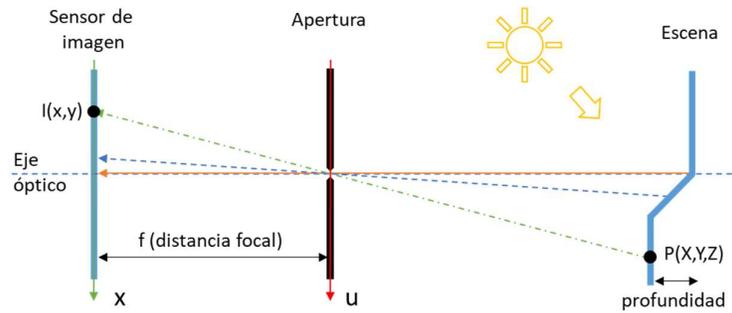


Figura 2.7. Información de profundidad en una cámara de agujero.

2.2.2. La cámara de lente convencional

Pareciera que la cámara de agujero resolvería nuestro problema de calcular las profundidades en la escena para luego representarlas por alguna superficie o similar, si no fuera por la escasísima luz que penetra por el estenope. La única manera de superar este impedimento es exponiendo la escena durante un tiempo prolongado, que hace impracticable el método para escenas dinámicas y cámaras sin apoyo firme.

La alternativa es ampliar la apertura para que ingrese más luz. Pero esto conduce a que un punto en la escena ya no se mapee en un punto en la imagen, sino en un círculo. Y dichos círculos van superponiéndose parcialmente para puntos contiguos en la escena, reproduciéndose imágenes borrosas. Entonces la solución es, manteniendo la apertura ampliada para el ingreso de luz, colocar lentes que nuevamente hagan converger un punto de la escena en un punto en la imagen. La Figura 2.8 representa un esquema para dos puntos ubicados a una distancia tal que los conos de luz que reflejan convergen en los correspondientes puntos en el plano imagen. Esto significa que la escena está en el plano focal y la imagen formada tiene la mayor nitidez. El sensor de imagen integra los rayos de los conos de luz correspondientes a cada punto y la gran diferencia con la cámara de agujero es que los puntos en el plano imagen reciben más luz y por lo tanto requieren menor tiempo de exposición.

Por otra parte, desde el punto de vista del espacio de los rayos, cada rayo sigue siendo representado por un punto, pero ahora la apertura es atravesada por un cono de rayos que se integran en una única posición en el plano imagen. Entonces el diagrama $x-u$ se ve como en la Figura 2.8.

Cabe en este momento destacar que, dado que el plano imagen es el plano $x-y$ y que los rayos de los conos que atraviesan la apertura $u-v$ se integran en las posiciones que alcanzan dicho plano, esto es equivalente a calcular integrales de proyección en el espacio de los rayos. Esto es, cada valor en la posición x es la suma de los rayos que pasan para todo u . En el esquema de la Figura 2.9 mostramos esta idea.

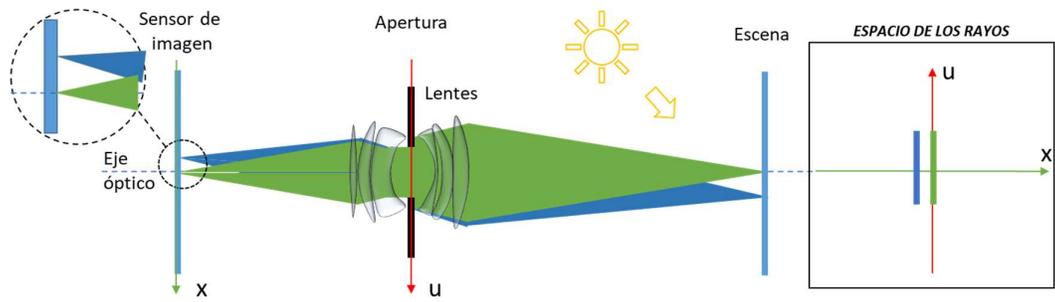


Figura 2.8. Esquema de formación de imagen mediante una cámara de lentes convencional, para una escena en el plano focal.

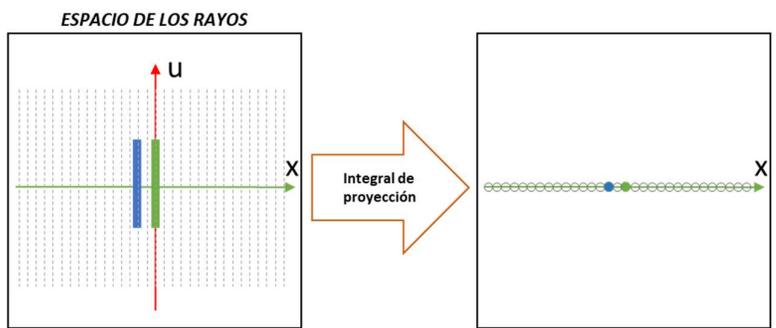


Figura 2.9. Recuperación de la imagen a partir de la integración de las proyecciones sobre el eje x del espacio de los rayos.

Si la escena presenta diferentes profundidades, los correspondientes puntos de imagen se formarán en diferentes planos si las lentes tienen foco fijo. En la Figura 2.10 observamos que la recta en el espacio de los rayos correspondiente a un punto que no pertenece al plano focal y que está más cerca del objetivo, se inclina en sentido horario. Si el punto estuviera por detrás del plano focal, la inclinación seguiría el sentido antihorario en la medida en que se incrementa esta distancia. Como hemos visto en la Figura 2.8, los puntos pertenecientes al plano focal se mapean como rectas verticales en el espacio de los rayos.

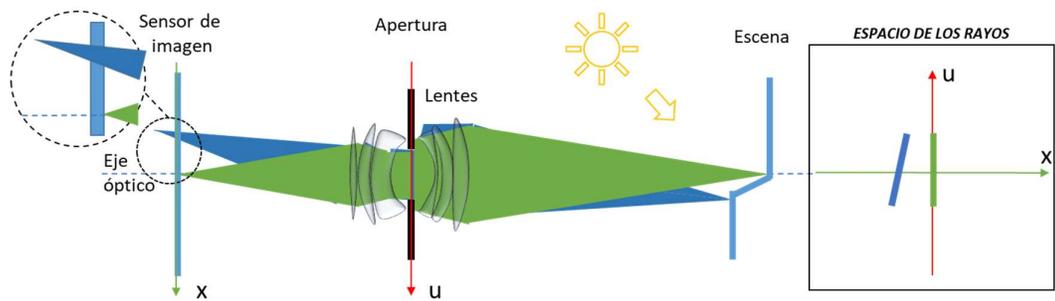


Figura 2.10. Esquema de formación de imagen mediante una cámara de lentes convencional, para una escena con puntos en diferentes profundidades.

El punto de la escena que no pertenece al plano focal produce una mancha en el plano imagen, no un punto. Esto se asocia al borronco observado en las estructuras de la escena que no están en foco. Para poner en foco dichas estructuras tenemos dos alternativas: desplazar el plano imagen a la posición donde convergen estos puntos o cambiar el foco de las lentes. Lo que hacen las cámaras convencionales es cambiar el foco de las lentes al ajustar las distancias entre las lentes que conforman el objetivo, sea de manera manual o automática, esto último denominado autofocus.

2.2.3. La cámara plenóptica

Hay una alternativa al hecho de que hasta ahora, si hacemos foco en una posición, no podemos enfocar en otra.

La cámara plenóptica cuenta con los mismos elementos que tiene una cámara tradicional, pero se diferencia por insertar antes del sensor de imagen, una matriz de microlentes (MML). La idea original de Lippmann [20] retomada por Adelson y Wang en 1992 [22], consiste en multiplexar la radiancia 4D sobre un sensor 2D, anteponiendo a dicho sensor una MML. Dichas microlentes refractan los conos de luz correspondientes a los puntos de la escena, de modo que los rayos se reparten en los fotositos del sensor digital ubicado una distancia f detrás de la matriz. Esta distancia f es la distancia focal de las microlentes de la MML. En el esquema de la Figura 2.11 mostramos que los pixeles ubicados detrás de cada microlente capturan por separado los rayos que provienen desde diferentes ángulos en el cono de luz, de manera que pueden guardar información angular que antes, en el caso de la cámara convencional, se integraba en el sensor. Llamaremos u' al eje donde se ubica el sensor digital y du' al tamaño del fotosito.

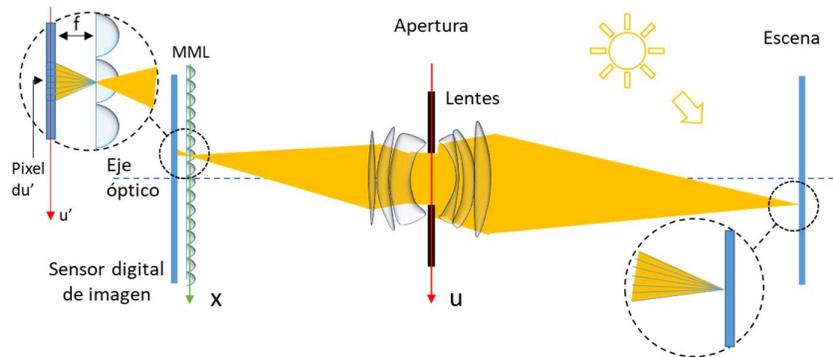


Figura 2.11. Esquema de una cámara plenóptica cuyas lenticulas refractan los conos de luz correspondientes a puntos pertenecientes al plano focal.

Dado que la MML es el diferencial de la cámara plenóptica, consideramos conveniente presentar una breve descripción de las diferentes opciones en microlentes que existen en la actualidad.

Las microlentes se denominan así dadas sus dimensiones, del orden de los micrones, aunque pueden extenderse en un rango de $10\ \mu\text{m}$ hasta $2\ \text{mm}$ de diámetro o lado. También se las conoce como lenticulas, e históricamente (antes de la fotografía plenóptica) son utilizadas para el acople de la luz a fibras ópticas. En este caso, es determinante la alta precisión de las dimensiones laterales de la matriz de microlentes (mejor que $\pm 0.25\ \mu\text{m}$), además de perfiles de lentes que minimicen las aberraciones esféricas. Otra utilización es como sensores Shack Hartmann, para

medir tanto la distribución de intensidad como la distorsión de fase, en tiempo real, de frentes de onda. En esta aplicación, las microlentes deben tener apertura pequeña a los efectos de brindar la mayor resolución espacial en el sensado del frente. Pequeña apertura y longitud focal grande conducen a un número de Fresnel pequeño, con comportamientos más dominados por efectos de difracción que por refracción. Finalmente, otra de las aplicaciones de las matrices de microlentes es la homogenización de haces láser, que provee a la salida diferentes patrones de iluminación (superficie plana, patrón de puntos y patrón de líneas).

Las microlentes son refractivas y como tales, desvían la luz incidente de acuerdo a la Ley de Snell de manera similar a las grandes lentes. Pero cuando sus dimensiones son muy pequeñas, pueden aparecer fenómenos relacionados con el límite de difracción. Las geometrías pueden ser circulares (dispuestas en grillas cuadráticas o hexagonales), cuadradas, cilíndricas o hexagonales. Las cuadradas y hexagonales son las de mayor factor de llenado, dado que hacen contacto *full* entre vecinas optimizando la eficiencia de captación y transmisión de la luz (Figura 2.12).

Los materiales utilizados en la confección de las microlentes son el cuarzo (*fused silica*), silicio, borofloat (*pyrex*), epoxy y diferentes polímeros. Sus índices de refracción diferentes conducen a diferentes distancias focales y paso de diferentes bandas de luz.

La fabricación de microlentes involucra tecnologías de fabricación de semiconductores tales como fotolitografía y RIE (*reactive-ion etching*). Estas tecnologías permiten un muy exacto moldeado de los perfiles de las microlentes así como también su preciso posicionamiento dentro de la matriz.

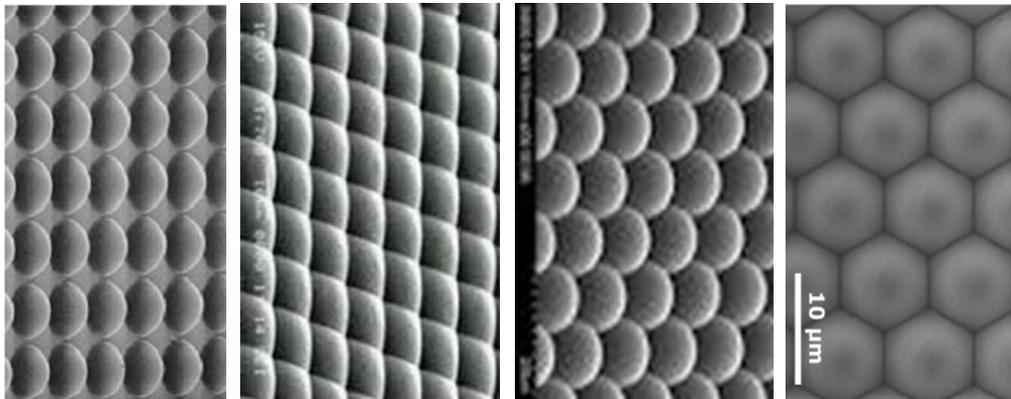


Figura 2.12. Diferentes geometrías y organización de microlentes. De izquierda a derecha: *Circular Lenses Quadratic Grid*; *Square Lenses*; *Circular Lenses Hexagonal Grid*; *Hexagonal Lenses*.

Volviendo al caso del esquema de la Figura 2.11, vemos que son seis los pixeles en los que se reparten los rayos. En la medida en que aumenta la cantidad de estos pixeles aumenta la resolución angular, es decir, la capacidad de la cámara de discriminar entre ángulos pequeños es mayor. Volveremos sobre este tema en el Capítulo 3.

Por otra parte, si consideramos los conos pertenecientes a dos puntos cercanos en la escena y tomamos en cuenta el camino que siguen haces paralelos de dichos conos, observamos que la distancia que puede resolver el plano imagen depende de la distancia entre las microlentes,

denominada *pitch*. Cuanto más pequeño es el *pitch* de las microlentes, mayor es la resolución espacial que puede alcanzarse. En el diagrama de la Figura 2.13 esquematizamos esta propiedad.

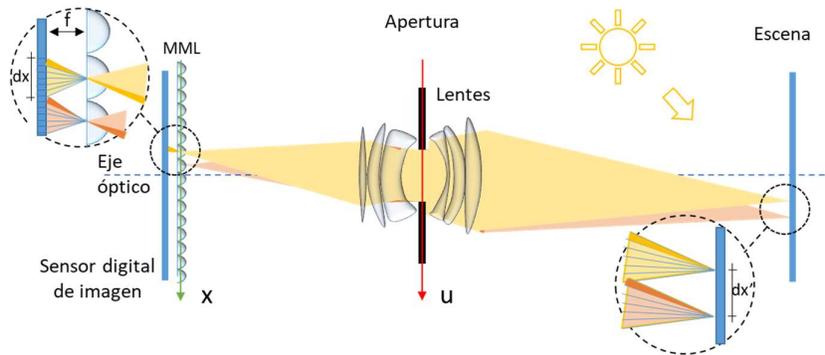


Figura 2.13. Dependencia entre la resolución espacial de la imagen y el *pitch* de las microlentes de una cámara plenóptica.

El espacio de los rayos correspondientes a la cámara plenóptica es como el mostrado en la Figura 2.14. Ahora los boxes representan a cada fotosito. Como en este ejemplo hay 6 fotositos debajo de cada microlente, en la dirección u del espacio de los rayos hay 6 boxes. En la dirección x hay tantos boxes como microlentes. Entonces cada pila vertical de boxes en una posición x es el conjunto de los píxeles bajo la microlente ubicada en dicha posición x . Basados en los esquemas de las Figuras 2.11 y 2.13 y apoyados en la Figura 2.14, podemos decir que la posición es muestreada por las microlentes mientras que la dirección es muestreada por los píxeles del sensor.

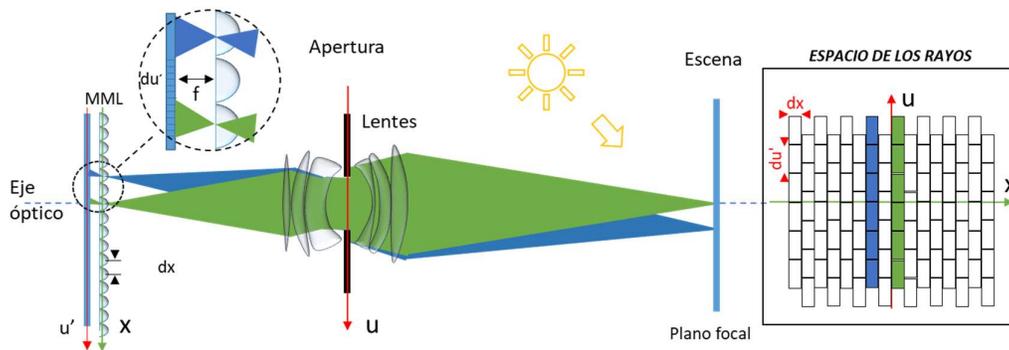


Figura 2.14. Espacio de los rayos de una cámara plenóptica para dos puntos de la escena pertenecientes al plano focal.

Nuevamente, de manera similar a lo que sucede con el espacio de los rayos de una cámara convencional, los puntos enfocados se observan como rectas verticales.

En la Figura 2.15 presentamos el caso en que la cámara captura puntos de la escena ubicados en diferentes profundidades.

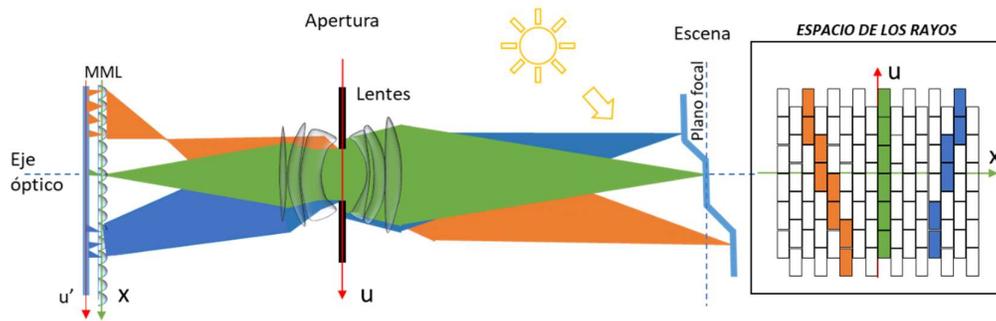


Figura 2.15. Espacio de los rayos de una cámara plenóptica para puntos de la escena pertenecientes a diferentes profundidades.

La imagen se obtiene integrando en todas las direcciones, es decir, proyectando sobre el eje de las posiciones x los ángulos u . Para el caso de la Figura 2.15, la única proyección que da como resultado un punto, suma de los rayos que conforman el cono, es la correspondiente al punto ubicado en el plano focal. Por otra parte, las proyecciones en el resto de las posiciones dan como resultado manchas en el eje de las posiciones x , es decir, puntos fuera de foco, como era de esperar para puntos en la escena que no pertenecen al plano focal. Queda claro que la imagen correspondiente al “foco óptico”, es decir, al que hizo la cámara de acuerdo a la refracción de las lentes del objetivo en la captura, se obtiene al sumar los píxeles correspondientes a cada una de las microlentes. Entonces se asigna este valor, que en realidad es el promedio de las intensidades para preservar el brillo, a cada posición (x,y) que ocupa cada microlente. A esta recuperación de la imagen la denominaremos “renderización”.

Al inicio del presente apartado afirmamos que hay una alternativa al hecho de que, para una cámara convencional, si hacemos foco en una posición no podemos enfocar en otra. Observemos el espacio de los rayos de la Figura 2.15. Si en lugar de calcular las integrales de proyección para sólo proyecciones verticales, lo hacemos variando la inclinación de dichas proyecciones, podemos obtener resultados muy diferentes. Veamos la Figura 2.16, donde repetimos el espacio de los rayos correspondiente a la Figura 2.15 pero presentamos dos alternativas diferentes a la correspondiente al “foco óptico”.

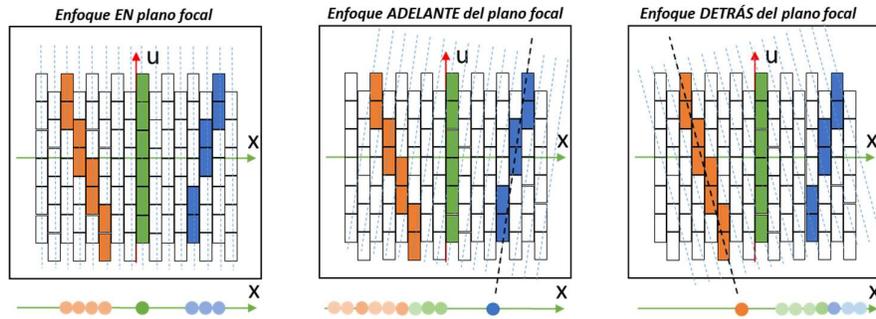


Figura 2.16. Integrales de proyección para reenfoques en diferentes profundidades de la escena de la Figura 2.15.

Según la inclinación en la que calculamos las integrales de proyección, obtenemos imágenes reenfocadas en diferentes profundidades.

En el apartado 2.1.2 comentamos que con la parametrización 2PP, Levoy y Hanrahan [33] redujeron a 4D la función plenóptica para referirse al LF. Los dos planos paralelos de la cámara plenóptica son el (x,y) y el (u,v) . Entonces cada rayo queda definido por los puntos (x,y,u,v) y el conjunto de todos ellos ocupando el espacio es el $LF(x,y,u,v)$. Veremos a continuación la relación que existe entre el LF y la distancia entre los planos paralelos, a través del Operador de Formación Fotográfica.

El Operador de Formación Fotográfica.

La irradiancia desde la apertura en la lente sobre un punto en el sensor, es la integral de la radiancia que entra por la lente. La irradiancia sobre el sensor en la posición (x,y) es [35]:

$$E_F(x,y) = \frac{1}{F^2} \iint LF(x,y,u,v) \cos^4 \theta \, du \, dv \quad (1)$$

El coseno a la cuarta potencia se refiere al viñeteo óptico que se da cuando se incrementa el ángulo con el que ingresan los rayos de la escena, en relación con el eje óptico. Cuanto más oblicuo es dicho ángulo, mayor la probabilidad de que las lentes lo desvíen fuera del área activa de los fotositos, disminuyendo el nivel de iluminación. Como pretendemos representar la dirección de los rayos en función de las distancias entre los planos de parametrización, no interesa este ángulo y por lo tanto, puede obviarse. Entonces asumimos que $LF(x,y,u,v) := LF(x,y,u,v) \cos^4 \theta$ [36]. Otra asunción es que los planos xy y uv son infinitos y que la luz es simplemente cero fuera de los límites físicos de la apertura y el sensor.

Haciendo referencia a la ecuación (1), interesa conocer qué sucede con las ecuaciones cuando las escenas de interés no hacen foco en el plano (x,y) del sensor, es decir, cuando la distancia F cambia a otra F' . En el esquema de la Figura 2.17 se observa esta situación.

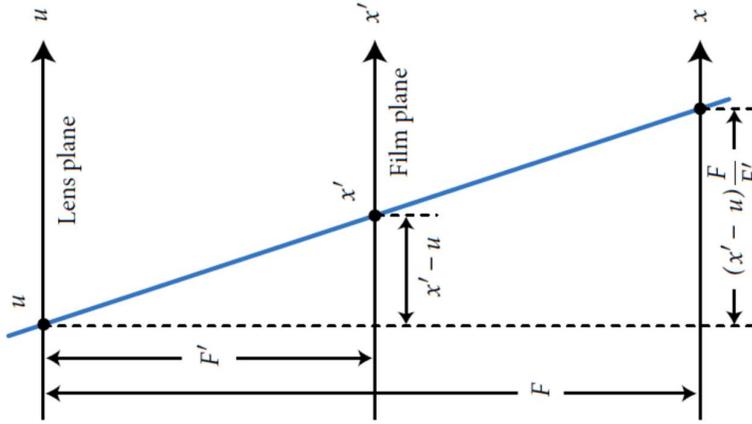


Figura 2.17. Coordenadas del espacio de los rayos para una imagen focalizada a una distancia F' . Fuente: [37].

Ahora la imagen está focalizada a una nueva distancia F' y necesitamos convertir (1) expresando el nuevo $LF'(x',y',u,v)$ en términos de $LF(x,y,u,v)$. Cabe recordar que, a los efectos de facilitar la comprensión, se muestra el diagrama en 2D, para u y x . Con las dimensiones v y y las relaciones son las mismas. Si definimos *alfa* como la relación $\alpha = F'/F$, se demuestra por triángulos semejantes que:

$$LF'(x',y',u,v) = LF\left(u + \frac{x'-u}{\alpha}, v + \frac{y'-v}{\alpha}, u, v\right) = LF\left(u\left(1 - \frac{1}{\alpha}\right) + \frac{x'}{\alpha}, v\left(1 - \frac{1}{\alpha}\right) + \frac{y'}{\alpha}, u, v\right) \quad (2)$$

Entonces (1) se expresa en términos del nuevo plano de refocalización (2) ubicado en la posición αF como:

$$E_{\alpha F}(x',y') = \frac{1}{\alpha^2 F^2} \iint LF\left(u\left(1 - \frac{1}{\alpha}\right) + \frac{x'}{\alpha}, v\left(1 - \frac{1}{\alpha}\right) + \frac{y'}{\alpha}, u, v\right) du dv \quad (3)$$

La ecuación (3) muestra al conjunto de rayos que converge a la posición (x',y') . $E_{\alpha F}$ es la fotografía formada sobre el sensor virtual a una profundidad αF y se denomina Operador de Formación Fotográfica. Esto significa que si LF es el campo de luz parametrizado por un plano x - y a una distancia F del plano u - v de las lentes objetivo, podemos poner en foco (“*refocalizar*”) la escena a diferentes distancias F' . Es necesario evaluar esta integral para cada punto (x',y') a fin de obtener una fotografía enfocada en F' .

Una forma de evaluar esta integral es mediante técnicas de cuadratura numérica, muestreando el integrando para diferentes valores de u y v y sumándolos. Otro método se basa en darnos cuenta de que, como hemos mostrado en las Figuras 2.15 y 2.16 para una sola dimensión, el LF se deforma en función de las distancias a las que se hace foco y la imagen enfocada surge de integrar en u y v este LF deformado, es decir, obtener una proyección hacia un plano 2D. Los algoritmos para recuperar las imágenes a partir del LF se tratarán en el Capítulo 3.

La cámara “Plenoptic 1.0”

La cámara “Plenoptic 1.0” o tradicional es la propuesta por Adelson y Wang [22], basada en el modelo de Lippmann. La matriz de microlentes está ubicada a una distancia f del sensor CMOS, donde f es la distancia focal de las lenticulas. Este es el modelo con el que hemos venido

explicando el muestreo del LF. La gran limitante de la Plenoptic 1.0 es que la resolución de las imágenes reenfocadas coincide con el número de lenticulas de la MML. Aunque esto está dejando de ser un problema dado que la industria de la microóptica está produciendo MML con pitch cada vez más pequeños, se contraponen con la necesidad de contar con el mayor número posible de píxeles por microlente para mejorar la resolución angular. Fotositos muy pequeños disminuyen la SNR dada la escasa luz que pueden capturar.

Una alternativa es trabajar con MML de pitch mayores que permitan ubicar mayor cantidad de fotositos debajo de cada microlente, introduciendo algunos cambios en el posicionamiento de las partes en la cámara, como lo propone la arquitectura “Plenoptic 2.0” de Lumsdaine y Georgiev [38].

La cámara “Plenoptic 2.0”

Plenoptic 2.0 posiciona la matriz a una distancia levemente mayor a la focal de las microlentes. La consecuencia de este cambio es que las lenticulas hacen foco en un plano imagen interior al cuerpo de la cámara: el plano imagen de las lentes objetivo (Figura 2.18). Por este motivo, a la cámara Plenoptic 2.0 también se la denomina “plenóptica enfocada”. Las lenticulas de la MML se comportan como pequeñas cámaras que generan en el sensor subimágenes que son porciones solapadas del plano imagen formado por el objetivo, de una manera ordenada y de gran precisión geométrica (Figura 2.19).

En la Plenoptic 2.0 el muestreo del LF es diferente y ahora existe una relación de compromiso entre las resoluciones espaciales y angulares de la imagen renderizada. Este modelo de cámara plenóptica no requiere tantas microlentes pero sí muchos píxeles por microlente. La resolución espacial dependerá de los parámetros ópticos de la cámara y de la distancia a la que la imagen renderizada hace foco [39]. A mayor distancia para el reenfoque, menos píxeles en la imagen renderizada y viceversa, lo que veremos en detalle en Capítulo 3.

Hemos presentado diferentes parametrizaciones del LF capturado por diferentes configuraciones de cámaras basadas en microlentes. A partir de las mencionadas alternativas, estudiaremos en el siguiente Capítulo algunas estrategias para renderizar imágenes reenfocadas en diferentes profundidades de la escena y vistas desde diferentes perspectivas.

En el siguiente apartado, hacemos una revisión del estado del arte de las cámaras plenópticas. Su desarrollo está experimentando una expansión tal que puede suceder que cuando usted esté leyendo este trabajo, hayan aparecido otras alternativas que estaban en proceso de diseño y desarrollo en el transcurso de la escritura de esta tesis.

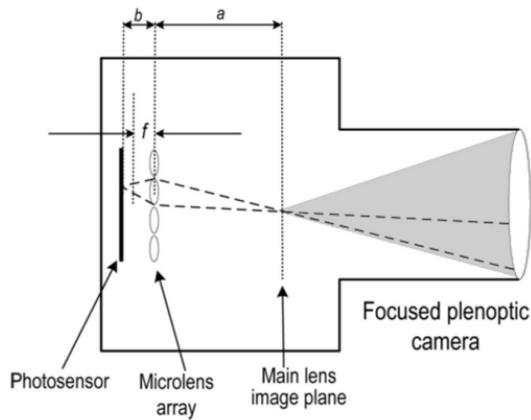


Figura 2.18. Cámara plenóptica “Plenoptic 2.0”. Las lentes del objetivo hacen foco en un plano situado una distancia “a” enfrente de la matriz de microlentes, mientras que su distancia al sensor CMOS es $b > f$, donde f es la distancia focal de las lenticulas. Fuente: [39].

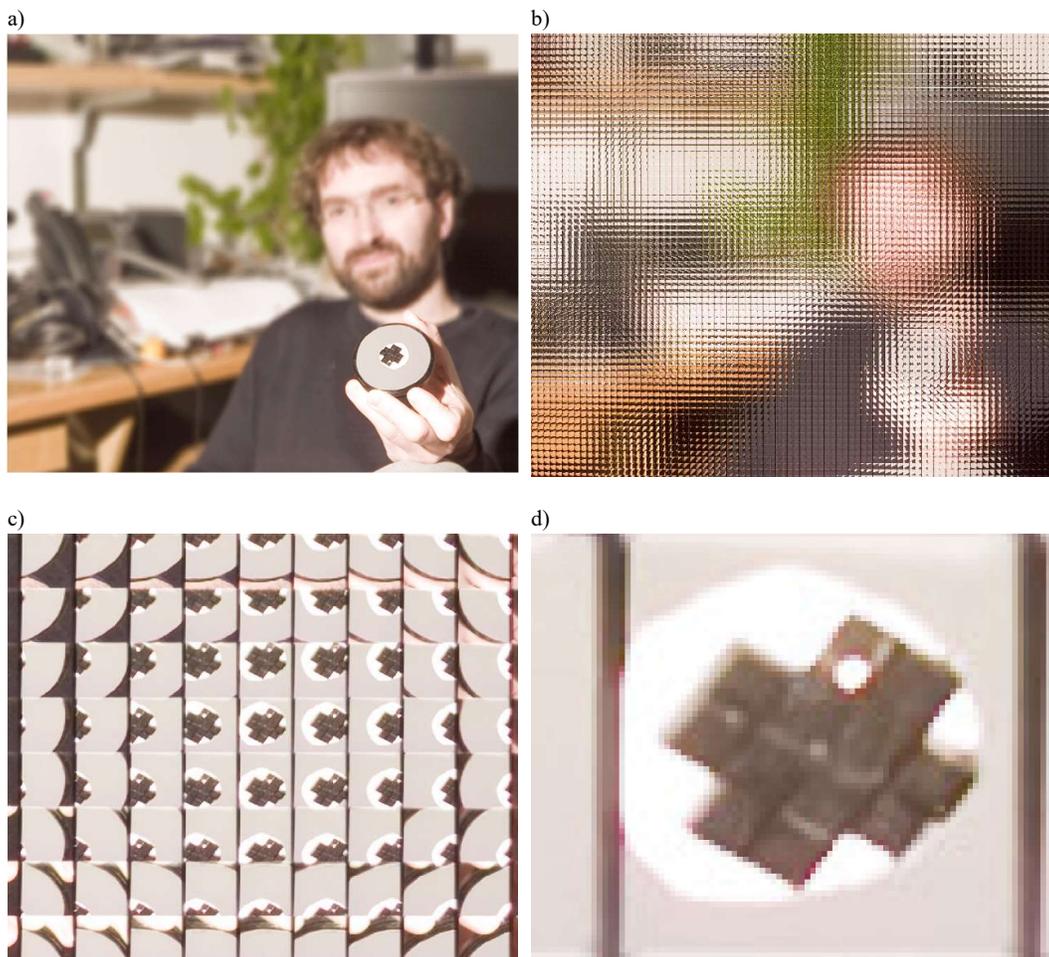


Figura 2.19. a) Imagen Renderizada. b) Imagen cruda 5433 x 7240 px, de aprox. 74x98 lenticulas. c) detalle del objeto en la mano del sujeto. d) detalle de los pixeles bajo una lenticula, en este caso, 74x74 pixeles. Fuente: [40].

2.3. Equipamiento disponible en la actualidad

La fabricación tanto de los sensores CMOS como de las microlentes hacen uso de técnicas empleadas en la producción de microchips, por lo que el desarrollo de los procesadores y memorias que todos los días nos sorprende con novedades ha impulsado el desarrollo de la fotografía plenóptica. Aun así, los dispositivos comerciales todavía no tienen llegada masiva a los consumidores, quizás porque sus precios son aún elevados para los de buena resolución en megapíxeles, mientras que los más accesibles todavía no alcanzan las resoluciones de una cámara fotográfica convencional económica.

La industria del cine y la televisión están a la espera de nuevos desarrollos y hay muchos prototipos que han generado expectativa pero que requieren grandes inversiones para su escalado y comercialización. Mientras tanto son dos las empresas que efectivamente comercializan cámaras plenópticas: Lytro y Raytrix. A continuación, describiremos brevemente los desarrollos, prototipos y productos comerciales disponibles.

2.3.1. Prototipos de Adobe

Adobe Systems ha sido líder en manipulación de imágenes. Uno de sus investigadores, el Dr. T. Georgiev, comenzó en 2004 a trabajar en un prototipo consistente en una matriz hexagonal de 19 lentes que capturan la escena desde 19 perspectivas levemente diferentes (Figura 2.20), dando por resultado una pila focal (*focal stack* en inglés, FS) de 19 imágenes. Cada subimagen tuvo una resolución de 5.2Mp y se llamó a este prototipo la Primera Generación [41].

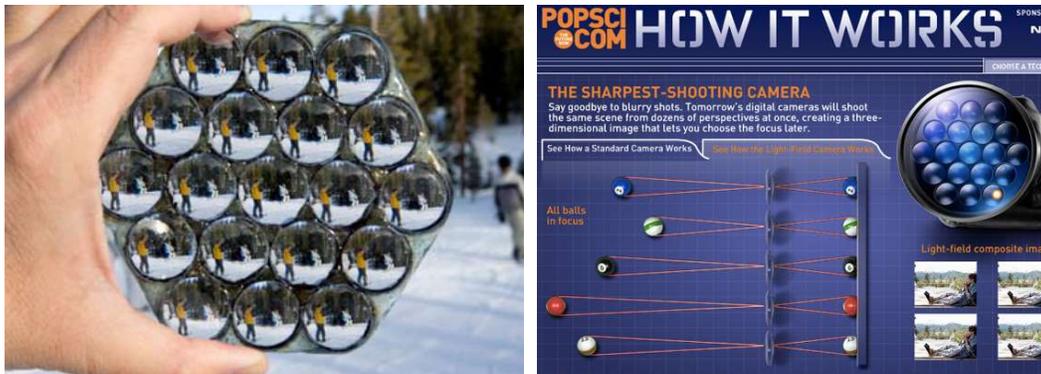


Figura 2.20. Prototipo Primera Generación de Adobe LightField Camera. Izq.: Matriz de lentes bautizado “Magic Lens”. Der.: Publicación en popsci.com de la cámara. Fuente: [41], [42].

La Segunda Generación de Adobe utilizó una matriz rectangular de 4x5 lentes colocada en frente del objetivo de una cámara convencional. Lo distintivo fue el software para el procesamiento y la posibilidad de elegir dónde hacer foco en la imagen (Figura 2.21).



Figura 2.21. Izq.: Segunda Generación de Adobe LF Camera. Der.: Lente con matriz rectangular de sub-lentes. Fuente: [42].

La Tercera Generación se presentó en 2010 y colocó una matriz de aproximadamente 7.000 microlentes dentro del cuerpo de la cámara, entre el sensor de imagen y las lentes objetivo (Figura 2.22). Ninguna de estas cámaras adquirió un perfil comercial. El aporte de Georgiev es una mejora en la resolución de la imagen posprocesada, como se verá en detalle más adelante cuando tratemos el tema del procesamiento de los LF.

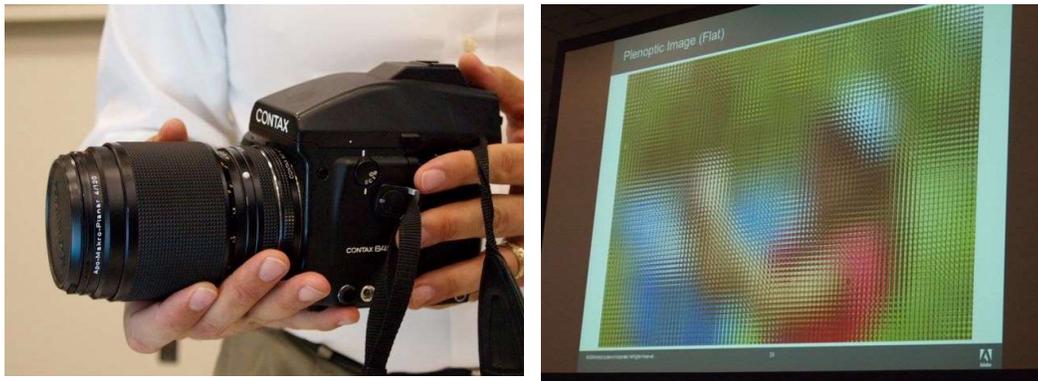


Figura 2.22. Izq.: Cámara plenótica de Georgiev, 3ª Generación. Der.: Imagen que captura la cámara. Fuente: [42].

2.3.2. Cámara de Campos de Luz CAFADIS

En la Universidad de la Laguna (España) se desarrolló un prototipo denominado CAFADIS (CAMara de FAsa – DISTancia) que se proyectó para estimación de distancias y frentes de onda [43]. El grupo está liderado por el Dr. Rodríguez Ramos y cuenta con un grupo de investigación que ha generado numerosas publicaciones y aplicaciones de la fotografía plenótica a campos tan diferentes como la endoscopia y la astronomía. También son autores de dos patentes internacionales.

El primer prototipo consistió en una cámara réflex Olympus E-P1 con una matriz de 510.000 microlentes circulares de 100 micrones de diámetro. A posteriori se trabajó con una cámara de

video RED One y se obtuvieron imágenes multifoco de 1920 x 1080 píxeles. Este prototipo obtuvo un stack de 24 planos focales y 200 perspectivas. Luego se sumaron otros desarrollos tales como una cámara de video 3D Full HD y una cámara subacuática (Figura 2.23).

Es muy interesante cómo este grupo ha encontrado que la fotografía plenóptica puede resolver problemas en diferentes aplicaciones. Por ejemplo, obtuvieron imágenes de campos de luz bajo el agua e investigaron el problema de la refracción de la luz entre los medios líquido y gaseoso. Lo mismo en astronomía: CAFADIS puede corregir las aberraciones causadas por turbulencia atmosférica creando mapas 3D de frentes de onda y mapas de profundidad. En 2013 generaron imágenes de endoscopia 3D en tiempo real, Sección Demostraciones en [43].



Figura 2.23. Diferentes prototipos de cámaras plenópticas desarrolladas por el Grupo CAFADIS de la Universidad de La Laguna. Fuente: [44].

2.3.3. Cámara Matricial Pelican

Pelican Imaging es una compañía de California (USA) que desde 2008 investiga la visualización de imágenes multifoco aplicada a los teléfonos móviles. Proponen la utilización de una matriz de microcámaras en lugar de anteponer al sensor CMOS una matriz de lenticulas. Esta arquitectura cambia el paradigma de la fotografía plenóptica actual basada en lenticulas, pero el resultado es similar. Lo interesante es la miniaturización que han alcanzado: el módulo al que denominaron PiCam, contiene 16 microcámaras de 1.000 x 750 píxeles cada una [45] que devuelve imágenes y mapas de profundidad de 8Mp (Figura 2.24). Además de la ventaja de poder integrar todas estas microcámaras en un único sustrato monolítico, el poder de la PiCam radica en su procesador que genera una imagen de alta resolución (HR) a partir de la combinación de imágenes de baja resolución (LR). Las imágenes individuales LR se normalizan a los efectos de homogeneizar las diferencias geométricas y fotométricas entre las cámaras individuales. Luego se procede a la detección de paralaje, que determina las correspondencias de pixel entre las imágenes respecto de una imagen de referencia (usualmente alguna cercana al centro del módulo). La cantidad de profundidades detectadas a partir de la paralaje es lo suficientemente grande como para habilitar la detección a niveles sub-píxel de exactitud. Por otra parte, la PiCam tiene sobre otras configuraciones de cámaras estéreo la ventaja de la cantidad de cámaras individuales que hace más robusta la detección de profundidades aún en condiciones de luz escasa. A posteriori comienza el proceso de superresolución con la fusión de las imágenes LR sobre una grilla HR. Por supuesto que este proceso involucra algoritmos de restauración Bayesiana tales como Maximum-a-posteriori y posprocesos de interpolación y estimación de incertezas. El PiCam fue

presentado en febrero de 2013. Su espesor es de sólo 3mm y fue Nokia la empresa que licenció su uso para los teléfonos de la serie Lumia, aunque si bien estaba anunciado que se lanzarían en 2014, hasta el momento no se han comercializado [45].



Figura 2.24. Matriz de microcámaras “PiCam”, integrada en un teléfono móvil. Fuente: [45].

2.3.4. Pixar Super LightField Lens

El estudio de Animaciones Pixar ha presentado en 2010 numerosas patentes, siendo interesante como aplicación la idea de combinar cámaras convencionales con sistemas de lentes plenópticas para la captura de los campos de luz de la escena. La riqueza que provee el procesamiento de estos campos de luz permite a la animación por computadora la manipulación de las imágenes como volúmenes en el espacio, haciendo más sencillo y realista la producción de películas.

Los dispositivos incluyen una matriz de microlentes, pero su posicionamiento se realiza próximo al plano focal de las lentes objetivo principales, disponiendo a continuación del camino óptico un conjunto de lentes de relevo (*relay lens*) que proyectan la radiancia sobre el sensor de imagen de cualquier cámara (Figura 2.25). Quien lidera la generación de estas ideas y figura como el inventor es el Dr. Di Francesco [46] que continúa investigando alternativas para la instrumentación para la captura de campos de luz para su utilización en cine y video 3D. Queda claro que su aporte es estrictamente óptico, generando acoples para las cámaras convencionales.

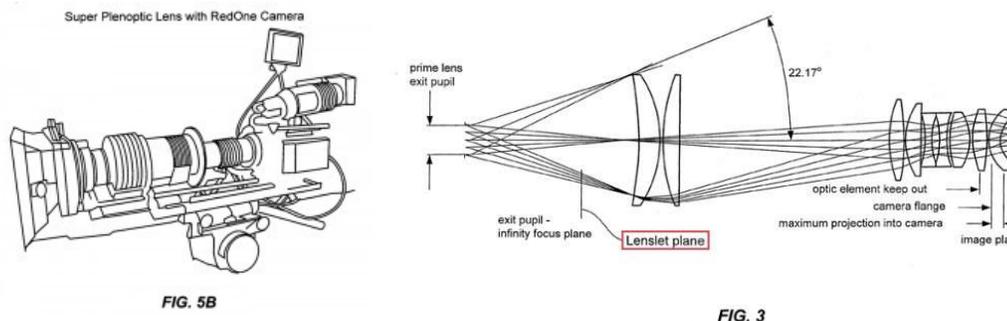


Figura 2.25. Patent application “Super Light-Field Lens” (March 4 2010, application no. 20110169994). Fuente: [47].

2.3.5. Toshiba LightField Camera Module

De modo similar a lo sucedido con la PiCam, Toshiba desarrolló a finales de 2012 una microcámara plenóptica similar a un cubo de dimensiones aproximadas a 8 mm de lado. El sensor CMOS de 7 mm x 5 mm tiene una resolución de 8Mp y se le antepone una matriz de 500.000 lenticulas circulares de unos 30 micrones de diámetro (unos 20 pixeles por microlente). Para tener una idea comparativa, la matriz de microlentes de la Lytro F01 es 5 veces más grande. La resolución efectiva es de 2Mp y tiene capacidad de procesamiento para reproducir video a 30 c/s

Esta miniaturización tiene por objeto integrar la cámara a los teléfonos móviles. Si bien Toshiba no ha revelado demasiados detalles, ha mostrado algunas aplicaciones tales como software para refocalización y extracción de fondo (basada en mapas de profundidad) para fotografía artística.

En abril de 2014 Toshiba lanzó el módulo TCM9518MD, que consiste en dos cámaras de 5Mp cada una, que capturan simultáneamente imagen y dato de profundidad para generar imágenes multifoco (Figura 2.26). Cuenta con un chip dedicado que procesa las imágenes capturadas por las cámaras y produce una imagen de 13Mp con información de profundidad [48].



Figura 2.26. Izq.: Microcámara plenóptica de 8Mp desarrollada por Toshiba. Der.: Módulo TCM9518MD de Toshiba. Fuente: [48].

2.3.6. Nvidia Near-Eye Light Field Display

En Julio de 2013 NVIDIA presentó en la Conferencia SIGGRAPH (grupo de interés en computación gráfica de la ACM, *Association for Computing Machinery*) un prototipo de anteojos que presenta a cada ojo un panel OLED (*organic light-emitting diode*, diodo orgánico de emisión de luz) cubierto por una matriz de microlentes (Figura 2.27). El dispositivo incluye la electrónica que codifica y muestra el campo de luz directamente enfrente de los ojos del usuario, quien puede hacer foco en varias profundidades de la escena [49].



Figura 2.27. Anteojos de Campos de Luz de NVIDIA. Fuente: [49].

2.3.7. MEMScam de DigitalOptics Corporation

La empresa californiana DigitalOptics lanzó a principios de 2013 un módulo para su incorporación en teléfonos móviles. Se trata de un sistema microelectromecánico (MEMS) que provee precisión de nivel micrométrico para hacer foco (Figura 2.28). Lo distintivo de este dispositivo en comparación con las alternativas antes presentadas, es que posee un actuador que posiciona las lentes de la microcámara en un lapso de 10ms y captura alrededor de 50 imágenes consecutivas con foco en diferentes profundidades de la escena. Entonces crea una imagen resultado de 8Mp que puede mostrar 6 zonas en diferente foco.



Figura 2.28. Módulo MEMS de DigitalOptics “MEMS|cam”. Fuente: [50].

2.3.8. Lytro

En 2003 Ren Ng comenzó sus investigaciones en el campo de la fotografía computacional y en 2006 presentó su tesis doctoral [37] y consiguió financiamiento para lanzar una startup denominada Lytro, que en la actualidad comercializa tres equipos: la cámara F01, la cámara Illum y el sistema Immerge de reciente lanzamiento (Figura 2.29). La F01 devuelve una imagen cruda de 3280 x 3280 píxeles, que luego de procesadas se convierten en 1080 x 1080. En rigor de verdad, estas dimensiones en HD surgen de la interpolación de la imagen reconstruida que en realidad tiene 360X360 píxeles aproximadamente, a partir de una matriz de 130.000 microlentes y sensor CMOS de 11Mp. La F01 puede conseguirse en el mercado a unos US\$100. En 2014 aparece la Illum, que representa una gran evolución de la F01 al brindar una resolución de 4Mp y un sensor CMOS de 40Mp, junto a mejoras en la óptica y el posprocesamiento. Su precio ronda los US\$1.400. Finalmente, el Immerge es una plataforma que captura el campo de luz situado 360° alrededor del equipo, lo que brinda la experiencia de visualizar todo el espacio. Se presenta como otra opción en el mundo de la Realidad Virtual (VR) utilizando campos de luz [51].



Figura 2.29. De izquierda a derecha: Lytro F01 (First Generation), Lytro Illum y Lytro Immerge. Fuente: [51].

2.3.9. Raytrix

Por otra parte, la empresa alemana Raytrix [52] fundada en 2009, vende cámaras plenópticas desde 2010 (Figura 2.30). Se diferencia de Lytro porque apunta a aplicaciones científicas e industriales y no a fotografía de ocio. Es muy interesante el abanico de aplicaciones propuesto por esta empresa:

- Sistemas de Inspección Óptica Automática (*AOI Systems*) 3D: permite con un solo disparo la obtención de imágenes 2D convencionales y el perfil de superficie de los objetos en una línea industrial;
- Análisis tridimensional de plantas: disponible con luz visible e IR, permite el estudio del crecimiento de las plantas inclusive en el exterior donde puede haber movimientos (debidos al viento, por ejemplo) que no son un problema debido al disparo único necesario.
- Medición de velocidad de partículas en fluidos (*Particle Tracking Velocimetry*, PTV): lo que antes se hacía con un sistema de multicámaras, ahora es posible con la captura de campos de luz. Los requerimientos son alta resolución espacial, buena profundidad de campo y calibración métrica.
- Robótica: posibilidad de navegación autónoma mediante medición en tiempo real de las distancias en el espacio.
- Microscopía 3D: posibilidad de efectuar mediciones de volumen en muestras microscópicas.

Raytrix ofrece 10 cámaras de muy diferentes características, según las aplicaciones antes mencionadas. Desde monocromo pasando por IR hasta RGB, con resoluciones efectivas (de la imagen posprocesada final, no la resolución del sensor) entre 1Mp hasta 42 Mp, frecuencias de video entre 7 y 180 fps, algunas con sensor CMOS y otras con CCD, cubren un amplio espectro de necesidades diferentes. Sus precios no son comparables con Lytro: el modelo R1 cuesta €1.490 mientras que la R5 (1Mp de resolución efectiva) parte de los €3.000 hasta llegar a los €10.000 incluidos algunos accesorios y el software para refocalización. No se dispone por el momento del precio de los modelos más sofisticados, por ejemplo, el C42i con 42Mp de resolución.



Figura 2.30. Raytrix R5 (modelo más económico). Raytrix C42 (módulo de captura y conjunto con óptica). Fuente: [52].

2.3.10. Avegant: nueva plataforma de Realidad Mixta

La *startup* Avegant [53] ha presentado en marzo de 2017 un nuevo desarrollo basado en campos de luz, que permite visualizar objetos en múltiples planos focales, interactuando los objetos sintetizados con la escena del mundo real, lo que provee una experiencia interactiva más realista. Requiere la utilización de un visor de Realidad Aumentada [54] y por el momento no se dispone de información acerca de cómo trabaja exactamente dicho visor. Puede consultarse que proyecta en la retina de ambos ojos una imagen. En la Figura 2.31 observamos las gafas de Realidad Aumentada.



Figura 2.31. Gafas de Realidad Aumentada de AVEGANT [54]. Cuadro de video que muestra la interacción del usuario con las imágenes sintéticas, mezcladas con el mundo real [53].

2.3.11. Detectores de grafeno

Un grupo de la Universidad de Michigan, liderado por Zhaohui Zhong [55], está trabajando en detectores de luz basados en grafeno [56]. El grafeno es un material laminar obtenido del grafito, consistente en una única capa de átomos de carbono obtenido, que es transparente y posee propiedades como conductor eléctrico. Dado lo novedoso del método, no encontramos información en detalle, sólo algunos bosquejos en los que está trabajando el grupo [57].

Capítulo 3

Procesamiento de las Imágenes Plenópticas

En este capítulo presentamos los algoritmos propuestos para la codificación del LF y su manipulación para visualizar la escena desde diferentes perspectivas y reenfocada en diferentes profundidades.

Hemos estudiado que el LF es la distribución espacial y angular de toda la luz en el espacio libre de ocluidores. Pero la estructura de datos 4D del LF no permite comprender de manera intuitiva la radiancia de la escena, a lo sumo somos capaces de interpretar un mundo 3D. Entonces es necesario reducir la dimensionalidad del LF a imágenes que puedan representarse en un papel o un monitor. Se habla de “cortes 2D” del LF 4D, o proyecciones bidimensionales del LF. Existen varias formas de obtener dichos cortes del LF o los de su espectro, como detallaremos en este Capítulo. Veremos que existen diferentes maneras de representar el LF. A partir de ellas, se proponen algunos algoritmos para recuperar imágenes enfocadas en diferentes profundidades y observar la escena desde diferentes puntos de vista o perspectivas. Esto puede hacerse en el dominio espacial, sea con la LF cruda, tal como fue capturada por el sensor o codificada en una matriz 4D, y en el dominio frecuencial, aplicando la Transformada de Fourier al LF.

MATLAB (abreviatura de *MATrix LABORatory*, "laboratorio de matrices") es un entorno de desarrollo integrado con un lenguaje de programación propio, que permite implementar algoritmos de manera relativamente sencilla. A las diferentes estrategias mencionadas en el párrafo anterior las convertimos en algoritmos en el entorno de programación MATLAB y presentamos los resultados. Asimismo, describimos brevemente y presentamos resultados de algunos algoritmos cuya implementación está disponible en la web, a los efectos de mostrar el abanico de posibilidades que existe, cada una con sus ventajas y desventajas.

3.1. Función Plenóptica aplanada y 4D

En el Capítulo 2 estudiamos que es posible discriminar la dirección de donde provienen los rayos reflejados por una escena. La MML funciona como un multiplexor espacial que construye, debajo de cada lenticula, pequeñas porciones de la imagen que denominamos microimágenes. Esta denominación se debe a las dimensiones micrométricas de las microlentes de las matrices utilizadas en fotografía plenóptica. La observación en detalle de microimágenes contiguas revela que las porciones de la escena se repiten pero levemente desplazadas, mostrando que provienen desde ángulos diferentes para cada lenticula.

3.1.1. Captura del LF

Aunque el LF es 4D la cámara captura necesariamente, por la geometría plana del sensor, una imagen 2D. La MML, como hemos explicado, genera tantas microimágenes como microlentes en la matriz. Entonces la imagen capturada por el sensor es en realidad una matriz 2D de microimágenes 2D y guarda la distribución posicional y angular del LF. En la bibliografía se denomina a esta imagen la “función plenóptica aplanada” (*flattened* en inglés) [58]. La recuperación de una imagen vista desde diferentes perspectivas y reenfocada en diferentes profundidades a partir del LF capturado por el sensor se denomina “renderización”, castellanización del término inglés *rendering* que significa generar una imagen a partir de un modelo, usando una aplicación de computadora.

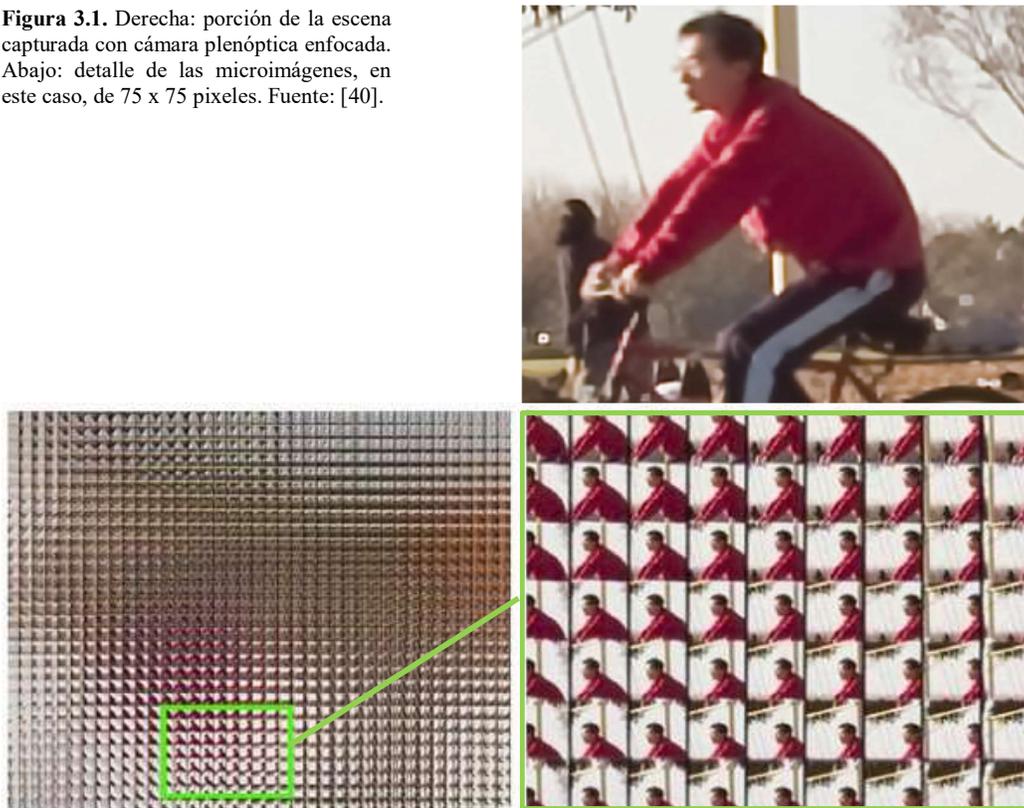
Para el proceso de renderización podemos trabajar directamente sobre la función plenóptica aplanada o convertirla en una matriz 4D, como fue propuesto por Levoy y Hanrahan [33]. En este momento cabe hacer un paréntesis y destacar que las características constructivas de los sensores plenópticos, tales como la alineación de las microlentes con el sensor de imagen, las deformaciones geométricas de las lentes, el patrón de Bayer para codificar color, aportan degradaciones a la imagen cruda. Si los bordes de las microlentes no están alineados con los bordes de los elementos de imagen del sensor, o no hay un número entero de fotositos debajo de cada microlente, existen rotaciones, etc., se requiere un proceso previo de alineación e interpolaciones que puede resultar computacionalmente caro, pero es necesario antes de comenzar la renderización. En la presente tesis no estudiaremos los procesos de interpolación cromática (*demosaicing*), calibración y rectificación, dado que exceden los alcances del trabajo. Remitimos al lector a [59] y [60] donde se realiza un pormenorizado abordaje del preproceso de la imagen cruda en su preparación para la codificación del LF.

Si la MML está compuesta por xy lenticulas y debajo de cada microlente se genera una microimagen de uv pixeles, a partir de la función plenóptica aplanada podemos representar el LF como una “matriz de matrices”: posición (x,y) de cada lenticula, posición (u,v) de cada pixel de la microimagen. Dicho de otro modo, el LF puede guardarse como una matriz de microimágenes (Figura 3.1), es decir, una matriz 4D $LF(x,y,u,v)$.

3.1.1. Codificación del LF

La síntesis de la función LF 4 dimensional requiere recorrer la matriz imagen cruda y asignar cada pixel a su correspondiente lenticula en su correspondiente posición debajo de dicha lenticula. Esto puede hacerse con bucles anidados que recorran la matriz de microlentes y en cada lenticula, los correspondientes pixeles, o recurriendo a una vectorización que involucre saltos de acuerdo a la cantidad de pixeles por lenticula. Implementamos ambos algoritmos y evaluamos su performance.

Figura 3.1. Derecha: porción de la escena capturada con cámara plenóptica enfocada. Abajo: detalle de las microimágenes, en este caso, de 75 x 75 píxeles. Fuente: [40].



Implementación por bucles anidados

La función `read_LFfor` simplemente recorre la imagen cruda con cuatro `for` anidados, y crea la matriz cincodimensional $LF([N_x, N_y, N_u, N_v, 3])$ donde $N_x * N_y$ es el número de lenticulas de la MML y $N_u * N_v$ es la cantidad de píxeles por lenticula. Lo hace para los tres planos del modelo de color RGB. A continuación, presentamos el pseudocódigo del algoritmo.

Pseudocódigo 1.- Read_LFfor

Input: matriz imagen cruda `im` de $N_x * N_y$ microlentes, cada una con $N_u * N_v$ píxeles. Modelo color RGB.

Output: matriz cincodimensional $I_f(N_x, N_y, N_u, N_v, \text{CanalColor})$.

For cada CanalColor

For microlente en renglón N_x

For microlente en columna N_y

For pixel en renglón N_u

For pixel en columna N_v

$I_f(\text{renglón}N_x, \text{columna}N_y, \text{renglón}N_u, \text{columna}N_v) =$

$= \text{im}(\text{renglón}N_x * N_u + \text{renglón}N_u, \text{columna}N_y * N_v + \text{columna}N_v, \text{CanalColor})$

End

End

End

End

End

Implementación por vectorización de bucles

La función `read_LFvect` convierte bucles `for` en operaciones equivalentes de vectores y/o matrices, estrategia que se denomina *vectorización de bucles* y optimiza la ejecución de los algoritmos. La función utiliza menos bucles y construye el LF considerando el recorrido por saltos de la matriz imagen, lo que la hace más eficiente. El pseudocódigo es el siguiente:

Pseudocódigo 2.- Read_LFvector

Input: matriz imagen cruda `im` de $N_x \times N_y$ microlentes, cada una con $N_u \times N_v$ pixeles. Modelo color RGB.

Output: matriz cincodimensional `lf(Nx,Ny,Nu,Nv,CanalColor)`.

Calcular: $\text{mitadNu} = \text{renglones Nu} / 2$; $\text{mitadNv} = \text{renglones Nv} / 2$;

For `pixelNu` entre $-\text{mitadNu}$ y $+\text{mitadNu}$

For `pixelNv` entre $-\text{mitadNv}$ y $+\text{mitadNv}$

`lf(renglónNx,columnaNy,pixelNu+mitadNu,pixelNv+mitadNv,CanalColor)=`

`=im(pixelNu+mitadNu+Nu entre 0 y $(N_x \times Nu)$, pixelNv+mitadNv+Nv entre 0 y $(N_y \times Nv)$,CanalColor)`

End

End

En sintaxis de MATLAB, la expresión en pseudocódigo `pixelNu` entre $-\text{mitadNu}$ y $+\text{mitadNu}$ es equivalente a la expresión `pixelNu = -mitadNu: mitadNu` y este tipo de expresiones reemplaza bucles, optimizando el código.

A modo de ejemplo, codificamos la imagen `'golgihighres1.jpg'` de $4624 \times 4624 \times 3$ uint8, 4.414MB, $N_x = N_y = 289$ y $N_u = N_v = 16$, publicada para su uso en [61], con ambos algoritmos: `read_LFfor` requirió 169.389411 segundos mientras que `read_LFvect` lo hizo en 0.157109 segundos, con un Intel® i7 6700HQ @ 2.60GHz, 16.0GB RAM. La diferencia es notable y el resultado es exactamente el mismo. La Figura 3.2 muestra una ampliación de un sector de la imagen cruda. Se observa que las microlentes son del tipo *Circular Lenses Quadratic Grid*.

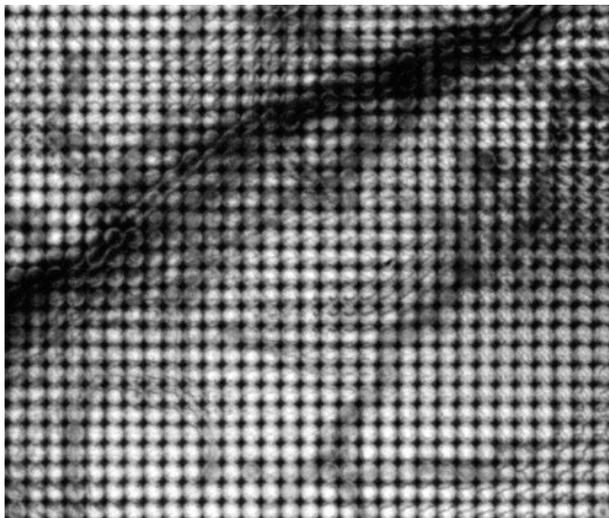
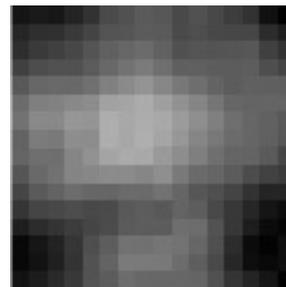


Figura 3.2. Izquierda: Porción de la imagen cruda `'golgihighres1.jpg'`. La imagen es del tipo RGB pero todos los canales presentan los mismos valores, motivo por el cual se visualiza como si fuera del tipo intensidad (o escala de grises).

Abajo: una de las lenticulas de la imagen. Fuente: [61].



3.2. Renderización de captura original y subaperturas

Una vez codificado el LF, podemos manipular la matriz 5D utilizando la potencia de MATLAB para trabajar con matrices. Por ejemplo, podríamos seleccionar una única microimagen de la imagen cruda especificando las coordenadas de la lenticula y solicitando que se muestren todos los pixeles pertenecientes a dicha microlente, para todos los canales de color. Para ello se deben fijar algunas coordenadas de la hipermatriz y recorrer todos los elementos de interés. Entonces, si nos interesa observar una de las microimágenes de la matriz de microlentes, fijamos la posición de la lenticula y recorreremos los pixeles posicionados debajo de ella. En MATLAB la función `squeeze` elimina las dimensiones de una matriz que son igual a 1 de manera que reduce su dimensionalidad y permite manipular hipermatrices. En el caso que quisiéramos observar la microimagen de la lenticula de posición $N_x=N_y=60$ (mostrada en la Figura 3.2, abajo), en MATLAB debe escribirse:

```
microimagen=squeeze(LF(60,60, :, :, :));  
imshow(microimagen);
```

En el Capítulo 2 estudiamos el espacio de los rayos de las diferentes configuraciones de cámaras: de agujero, convencional y plenóptica. Observamos que, así como para formar la imagen de una cámara convencional era necesario integrar los rayos correspondientes a cada cono de luz (Figuras 2.8 y 2.9), para una cámara plenóptica se requería calcular las integrales de proyección para cada posición (Figuras 2.14 y 2.15). Cuando las proyecciones son verticales, la integración es equivalente a promediar los valores capturados por los fotositos pertenecientes a cada microlente y obtenemos la imagen tal como fue capturada en el plano óptico con la cámara plenóptica (Figura 2.16, enfoque en plano focal). La función `renderizar(LF)` promedia los valores de los pixeles correspondientes a cada microimagen y devuelve una imagen de tamaño igual al de la cantidad de microlentes, $N_{xy}=N_x*N_y$.

Pseudocódigo 3.- Renderizar

Input: matriz cincodimensional $lf(N_x, N_y, N_u, N_v, \text{CanalColor})$.

Output: matriz imagen im de N_x*N_y elementos, modelo de color RGB.

Calcular: $k=N_u*N_v$.

For cada microimagen (N_x, N_y)

Promedio=(sumatoria de los k pixeles N_u, N_v) / k

$im(N_x, N_y) := \text{Promedio}$

End

Sabemos que $N_{uv}=N_u*N_v$ es la cantidad de pixeles que componen cada microimagen correspondiente a cada lenticula de la MML. Para renderizar la imagen originalmente enfocada, promediamos la totalidad de dichos pixeles de cada microimagen. Podríamos preguntarnos qué sucede si excluimos de dicho promedio la hilera (o “anillo”) de los pixeles más exteriores de cada microimagen. Esto es equivalente a disminuir la apertura de las lenticulas de la MML. En una

cámara convencional, disminuir la apertura implica incrementar la profundidad de campo. La función `dismapertura(LF, beta)` disminuye la apertura un anillo de *beta* píxeles de ancho para cada microlente, integrando solamente los que quedan dentro de dicho anillo. Por ejemplo, si la imagen tiene $N_{uv}=11$ píxeles, un valor $\beta=3$ significa que quedará fuera del cálculo un anillo exterior de 3 píxeles de ancho y por lo tanto se integrarán los 5x5 píxeles centrales de cada microlente. La Figura 3.3 muestra resultados de la función `dismapertura` con diferentes valores de β . La salida para $\beta=0$ es la misma que la obtenida si aplicáramos la función `renderizar`. Observamos que a medida que disminuye la apertura, la imagen se hace más nítida para todas las profundidades, es decir, se incrementa la profundidad de campo.

Pseudocódigo 4.- Dismapertura

Input: a) matriz cincodimensional $I_f(N_x, N_y, N_u, N_v, \text{CanalColor})$. b) $BETA=N^\circ$ de píxeles de ancho del anillo exterior que disminuye apertura.

Output: matriz imagen im de $N_x \times N_y$ elementos, modelo de color RGB.

Calcular: $k=(N_u-(2*BETA))^2$.

For cada microimagen (N_x, N_y)

Promedio=(sumatoria de los k píxeles centrales) / k

$im(N_x, N_y):=Promedio$

End

Por otra parte y como es de esperar, a medida que disminuimos la apertura la imagen renderizada disminuye su brillo y aumenta el ruido, dada la participación de menor cantidad de rayos en la reconstrucción de la imagen. En la Figura 3.4 observamos el detalle de las áreas encerradas en rectángulos de la Figura 3.3.



Figura 3.3. Influencia de la apertura de las microlentes. En la figura de la izquierda no hay modificación, por lo tanto la imagen presenta el foco como fue capturado originalmente por la cámara plenóptica. A medida que disminuye la apertura, se incrementa la profundidad de campo.



Figura 3.4. Ampliación de los sectores de la imagen de la Figura 3.3, encerrados en recuadros. Se observa cómo se incrementa el ruido a medida que disminuye la apertura de las microlentes.

3.2.1. Imagen en perspectiva para Plenoptic 1.0

Observamos que, de manera similar a como obtenemos una microimagen de la imagen cruda, es posible seleccionar un pixel en la misma posición (u,v) para cada lenticula (x,y) y observar la escena desde diferentes puntos de vista o perspectiva. Esta acción es equivalente a restringir la entrada de luz desde un ángulo particular, por lo que a estas imágenes de perspectiva también se las denomina *subaperturas*. Recordemos que los pixeles debajo de cada microlente recogen los rayos que provienen desde diferentes direcciones. Si elegimos uno de estos pixeles, estamos eligiendo los rayos provenientes todos desde la misma dirección y por lo tanto, reconstruyendo la escena desde aquella perspectiva, como lo esquematiza la Figura 3.5. La cantidad de imágenes en perspectiva que puede obtenerse con Plenoptic 1.0 es igual a la cantidad de pixeles por microlente, $N_{uv}=N_u * N_v$.

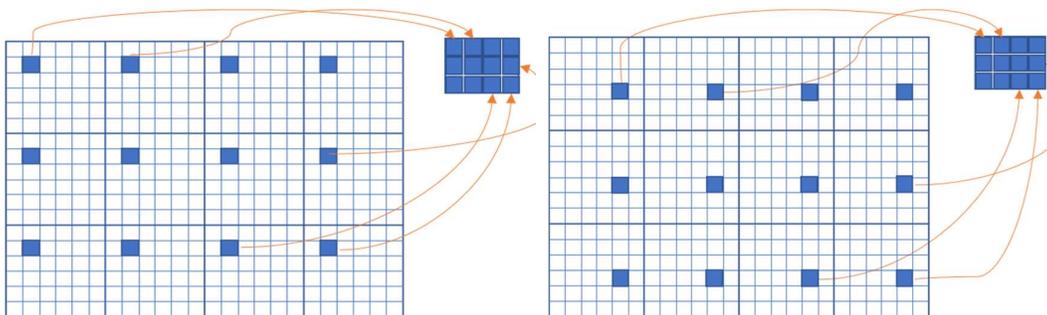


Figura 3.5. Conjunto de 3x4 microlentes, cada una de 6x6 pixeles. Izquierda: la imagen en perspectiva se forma seleccionando el pixel (2,2) de cada microimagen. Derecha: la misma imagen en otra perspectiva se forma seleccionando el pixel (4,5) de cada microimagen.

La imagen de perspectiva contendrá (x,y) pixeles y en sintaxis de MATLAB, se construye utilizando nuevamente la función `squeeze`. Por ejemplo,

```
perspectiva=squeeze(LF(:,:,4,11,:));
```

En este caso estamos seleccionando para todas las l nt culas, el pixel ubicado en la posici n $(u,v)=(4,11)$. En la Figura 3.6 observamos dos im genes de la misma escena desde diferentes perspectivas. La imagen es una captura con una c mara Lytro F01, que nos permite obtener un campo de luz $LF(378,379,11,11,3)$. La  ltima dimensi n guarda los tres canales de color RGB. Observamos un desplazamiento de perspectiva de izquierda a derecha, hecho m s evidente en la posici n de la punta de la bombilla sobre el mate o c mo se ocluye el libro naranja (posiciones extremas en la profundidad de campo).



Figura 3.6. A la izquierda se han seleccionado los pixeles de posici n $(u,v)=(5,3)$. A la derecha se observan los pixeles de posici n $(u,v)=(5,10)$. Ambas im genes tienen una resoluci n de 378 x 379 pixeles.

3.2.2. Imagen en perspectiva para Plenoptic 2.0

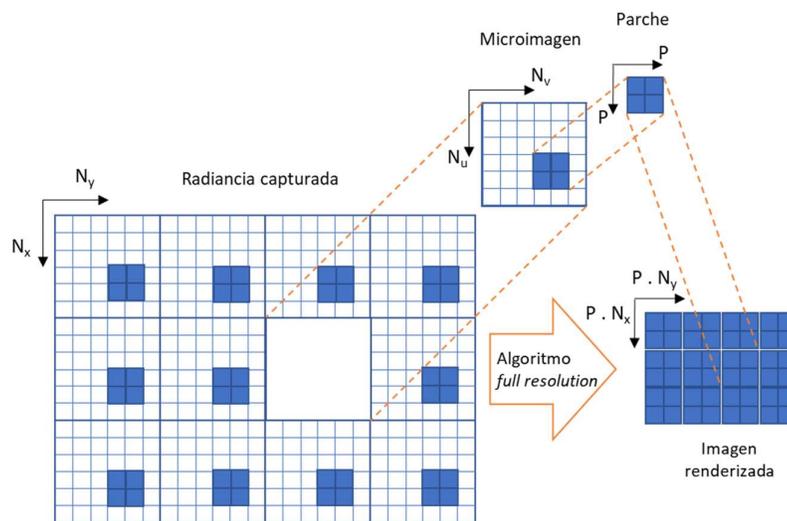
Los algoritmos hasta ahora explicados proveen una imagen reenfocada $im(N_x, N_y)$, es decir, restringen su resoluci n digital a la cantidad de microlentes de la c mara, dado que elegimos un pixel por l nt cula. Por otra parte, cuanto mayor sea el n mero de pixeles debajo de cada microlente, mayor ser  la cantidad de  ngulos diferentes para obtener reconstrucciones en perspectiva. Esta propiedad se define como resoluci n angular: m s pixeles por l nt cula, mayor resoluci n angular. Para lograr recuperar una imagen de buena resoluci n espacial y angular necesitamos matrices de muchas microlentes y sensores de muchos pixeles. En alg n punto se alcanza una relaci n de compromiso entre ambas resoluciones, si consideramos esta estrategia para reconstruir la imagen de perspectiva. Como se ha comentado en el Cap tulo 2, una alternativa es la configuraci n Plenoptic 2.0 que permite la obtenci n de mejores resoluciones espaciales pero cuyas resoluciones angulares dependen de la profundidad de la escena donde se hace foco, como analizaremos a continuaci n.

Las microlentes en las c maras del tipo Plenoptic 2.0 presentan un tama o o paso (denominado *pitch*) mayor que el utilizado en Plenoptic 1.0. Esto significa que, para un mismo sensor CMOS, la cantidad de l nt culas de la MML ser  menor pero la cantidad de pixeles correspondientes a cada una de ellas, mayor. Seg n lo que hemos analizado hasta el momento, esto significar a que empeora la resoluci n espacial y mejora la angular, aunque resulta al rev s, con algunos matices.

La arquitectura de la cámara Plenoptic 2.0 separa la matriz de microlentes y el sensor de imagen una distancia mayor al foco de las microlentes, y captura una imagen real invertida en el cuerpo de la cámara, formada por la lente objetivo. El resultado es una matriz de microimágenes en foco, invertidas, de una buena cantidad de pixeles cada una y que van desplazándose unos pocos pixeles entre vecina y vecina.

Estudiaremos en el siguiente apartado el método para obtener imágenes en foco en diferentes profundidades de la escena, para Plenoptic 2.0. Entonces comprenderemos mejor cómo se construye una imagen en perspectiva para este formato de cámara, pero por el momento aceptemos que dicha imagen se obtiene por seleccionar un mismo subconjunto de pixeles o recorte de cada microimagen y pegarlos uno a continuación del otro, como se esquematiza en la Figura 3.7.

Figura 3.7. Imagen renderizada para arquitectura Plenoptic 2.0. Se seleccionan bloques (o *parches*) de tamaño $P \times P$ para la misma posición para cada lenticula. Al concatenar dichos subconjuntos obtenemos la imagen reenfocada. Fuente: modificado de [39].



En el caso de la Figura 3.7 hemos seleccionado un subconjunto de 2×2 píxeles de cada microimagen. Si elegimos un parche del mismo tamaño pero ubicado en otra posición dentro de las microimágenes obtendremos otra perspectiva de la imagen, enfocada en la misma profundidad anterior.

Observemos que para Plenoptic 2.0 es necesario fijar dos parámetros para renderizar imágenes en diferentes perspectivas: uno, el tamaño del parche, (que determina posición del foco, como estudiaremos en el apartado 3.3.1.3) y otro, la posición del parche dentro de la microimagen (que determina la perspectiva). Entonces la resolución angular será dependiente de la posición del foco: cuanto más grande sea el parche (foco más cercano a la cámara, como veremos más adelante), menor cantidad de perspectivas. La Figura 3.8 muestra un ejemplo de una imagen enfocada en la misma profundidad, en dos perspectivas diferentes.



Figura 3.8. Renderización de una imagen capturada con una cámara Plenoptic 2.0 [40]. Los bloques son de 12x12 píxeles (para microlentes de 75x75 píxeles) y se eligieron posiciones (1,1) para la izquierda y (1,30) para la derecha. La niña está en foco mientras que el fondo se ve desenfocado y presenta artefactos, que se explicarán más adelante. Se muestran flechas para indicar donde es más evidente la diferencia por perspectiva.

3.3. Algoritmos de Refocalización

Denominamos *refocalización* al proceso computacional que permite la obtención de una nueva imagen, enfocada en una determinada posición dentro del rango de profundidades de la escena capturada con una cámara plenóptica. Cabe recordar que la refocalización es una acción a posteriori de la toma fotográfica y por lo tanto puede hacerse en cualquier momento y variando las posiciones del foco, siempre que se cuente con la imagen cruda del LF capturado por la cámara.

En el Capítulo 2 hemos presentado los fundamentos de la captura del LF, su parametrización y muestreo según se utilice la estrategia Plenoptic 1.0 ó 2.0. Sus diferencias plantean la utilización de enfoques sutilmente distintos, cada uno con sus ventajas y desventajas. Asimismo, dada la naturaleza matemática de la “señal imagen”, podemos aplicar al LF la Transformada de Fourier, por lo que presentaremos propuestas de refocalización en ambos dominios: espacial y frecuencial.

3.3.1. Refocalización en el dominio espacial

3.3.1.1. Desplazamiento-Suma

El Operador de Formación Fotográfica (3) presentado en el Capítulo 2 permite la obtención de una imagen enfocada en la posición $F' = \alpha F$, donde F es la distancia que separa el sensor de la apertura [37]. Esta integral doble puede evaluarse por técnicas de cuadratura numérica, pero una alternativa mucho más sencilla se basa en darnos cuenta que las posiciones (x,y) integradas en $u-v$ del nuevo LF presentan dos términos:

- Un desplazamiento $u \left(1 - \frac{1}{\alpha}\right)$ (para el caso de la posición x)
- Una amplificación (o reducción, dependiendo del valor de alfa) x'/α

El término de desplazamiento significa, para cada imagen de subapertura, correr su posición (x,y) el equivalente a uk donde $k = \left(1 - \frac{1}{\alpha}\right)$. La imagen enfocada se obtiene al promediar todas las subaperturas desplazadas su correspondiente uk .

La amplificación x'/α afecta a todas las subaperturas por igual y en la imagen final enfocada no cambia el resultado de la refocalización.

La implementación es sencilla y los resultados son satisfactorios, habida cuenta del bajo consumo de recursos de memoria del algoritmo, desde el punto de vista computacional. En sintaxis MATLAB, con la función `squeeze` obtenemos las imágenes de subapertura (tal como vimos en el apartado 3.2.1) y con la función `imtranslate` desplazamos a cada una, para luego sumarlas. Esta última función simplemente suma a las coordenadas de cada pixel un determinado valor para obtener una nueva imagen trasladada aquel valor de posiciones.

Pseudocódigo 5.- Shift_sum

Input: a) matriz cincodimensional `lf(Nx,Ny,Nu,Nv,CanalColor)`. b) **ALFA**=parámetro que relaciona distancias de foco.

Output: matriz imagen `im` de $Nx*Ny$ elementos, modelo de color RGB.

Calcular: $k=1 - (1/ALFA)$;

Vistadesplazada = `zeros(Nx,Ny)`;

For cada CanalColor

For cada subapertura

 Transitorio=subapertura trasladada **k** posiciones

 Vistadesplazada =Vistadesplazada + Transitorio

End

End

`im(Nx,Ny):= Vistadesplazada / (Nu*Nv)`

La Figura 3.9 muestra resultados de refocalización por desplazamiento-suma, que requirieron un tiempo de proceso de apenas 1.52 segundos. El LF posee 381x383 microimágenes de 9x9 pixeles cada una, como lo indica el comando

```
>> size(lf)
>> ans =    381    383     9     9     3
```



Figura 3.9. Izq.: imagen enfocada al fondo con algoritmo desplazamiento suma, $\alpha=0.6$. Der.: imagen enfocada al frente con algoritmo desplazamiento suma, $\alpha=1.8$. Fuente: [62].

Observamos que un valor $\alpha=1$ significa que no hay desplazamiento y el algoritmo simplemente promedia todas las imágenes de subapertura. Esto es equivalente a asignar a la posición correspondiente a cada lentícula el promedio de todos los píxeles debajo de ella, es decir, a la integración de los rayos en todas las direcciones en cada posición. La imagen resultado con $\alpha=1$ es exactamente la enfocada en el plano de foco óptico en la captura original, como lo resuelve la función `renderizar(LF)`.

3.3.1.2. Transformación de la Radiancia

Otra alternativa de refocalización en el dominio espacial puede plantearse desde el espacio de los rayos, presentado en el capítulo anterior.

La imagen cruda capturada por la cámara plenóptica, enfoca los rayos sobre el plano de las microlentes. Dicha captura puede hacerse eligiendo que el enfoque óptico de la cámara se haga al frente, en algún punto intermedio o al fondo de la escena. En el Capítulo 2 estudiamos cómo se representan los puntos en el espacio de los rayos según pertenezcan o no al plano focal. En la Figura 2.16 mostramos que, si las integrales de proyección se calculan con diferentes inclinaciones, obtenemos reenfoques en diferentes profundidades. Otra manera de arribar a los mismos resultados es “deformando” el espacio de los rayos mediante la aplicación de una transformación T de sesgado, como lo esquematiza la Figura 3.10. Vemos que esta transformación modifica la inclinación de las rectas originales correspondientes a puntos enfocados en diferentes profundidades. Ahora el LF parametrizado originalmente se convierte en un nuevo LF' . Si ahora aplicamos la función `renderizar(LF')` obtendremos enfoques en posiciones diferentes, de acuerdo a las integrales de proyección calculadas para las diferentes inclinaciones con que se deforme el espacio de los rayos.

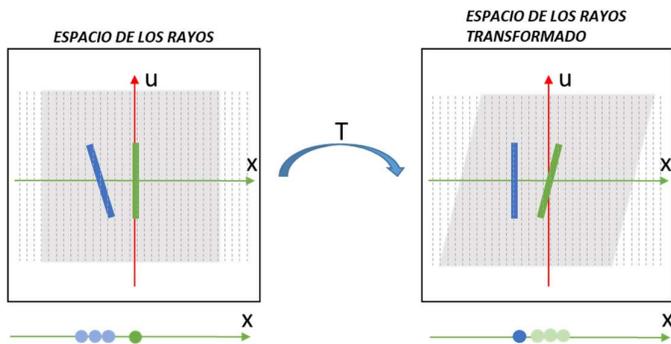


Figura 3.10. Transformación de Sesgado del espacio de los rayos. A veces se hace referencia a esta operación como “corte” de la radiancia. Debajo de los espacios de los rayos mostramos el resultado de las integrales de proyección, que devuelven un punto para los puntos pertenecientes al plano focal y un conjunto de puntos (borroneo), para los puntos fuera de dicho plano.

Nosotros transformamos la radiancia con nuestra función `RefocusRadiancia(LF, theta)` que encuentra las coordenadas del *sesgado* de la radiancia, según el valor de un parámetro `theta` que afecta la inclinación, e interpola el LF original a estas nuevas coordenadas para crear un nuevo LF'. Finalmente, proyectamos los valores de la nueva radiancia sobre el eje `x` con el algoritmo `renderizar(LF')` para obtener la imagen refocalizada. El proceso consume unos 30 segundos promedio, tiempo elevado comparado con los menos de 2 segundos que requiere el algoritmo de desplazamiento-suma presentado antes.

La Figura 3.11 muestra algunos resultados de la transformación de la radiancia.



Figura 3.11. Refocalización por transformación de la radiancia. `Theta=0` significa que no hay transformación y la imagen es la del enfoque óptico de la cámara. `Theta=-3` enfoca a una distancia más cercana a la cámara, mientras que `Theta=8` refocaliza a una profundidad mayor.

El pseudocódigo del algoritmo de la transformación de la radiancia es el siguiente:

Pseudocódigo 6.- RefocusRadiancia

Input: a) matriz cincodimensional `If(Nx,Ny,Nu,Nv,CanalColor)`. b) `THETA`=parámetro desplazamiento de la radiancia.

Output: matriz imagen `im` de `Nx*Ny` elementos, modelo de color RGB.

Calcular `[XX,YY,vv,ww]` = grilla 4D donde `XX=Nx`, `YY=Ny`, `uu=Nu`, `vv=Nv`;

`corte_x = XX + THETA*uu/Nu`;

`corte_y = YY + THETA*vv/Nv`;

For cada `CanalColor`

`Transitorio`= interpolación de los valores de `If` a sus nuevas posiciones `corte_x`, `corte_y`

End

Renderizar `Transitorio`

3.3.1.3. Renderizado por Bloques

La tercera alternativa de refocalización en el dominio espacial es exclusivo de la arquitectura Plenoptic 2.0 [58]. La imagen renderizada $I(k,l)$ puede obtenerse de un sistema plenótico enfocado (Plenoptic 2.0) por comprender el proceso de renderizado como la extracción de sub-bloques de las subimágenes (píxeles debajo de cada microlente) y su concatenación, para obtener la imagen enfocada I a una determinada profundidad. A diferencia de la cámara plenótica tradicional (Plenoptic 1.0), las muestras angulares para un punto espacial dado son muestreadas por diferentes microlentes. Entonces el renderizado para Plenoptic 2.0 involucra la integración “a través de” las microlentes en vez de la sumatoria de los píxeles dentro de un mismo microlente, como sucede con Plenoptic 1.0.

Tal como se ha venido parametrizando la radiancia, el espacio de las fases q-p que representa el muestreo de la radiancia es diferente según la geometría del conjunto sensor-microlentes-objetivo de la cámara plenótica. Como se observa en la Figura 3.13, es muy distinta la disposición de los elementos de imagen pertenecientes a las microlentes según la distancia sensor-microlentes sea igual al foco de las microlentes (Figura 3.13, Izq.) o levemente mayor a dicho foco (Figura 3.13, Der.). Las dimensiones indicadas en la Figura 3.13 pertenecen a las correspondientes en la Figura 3.12.

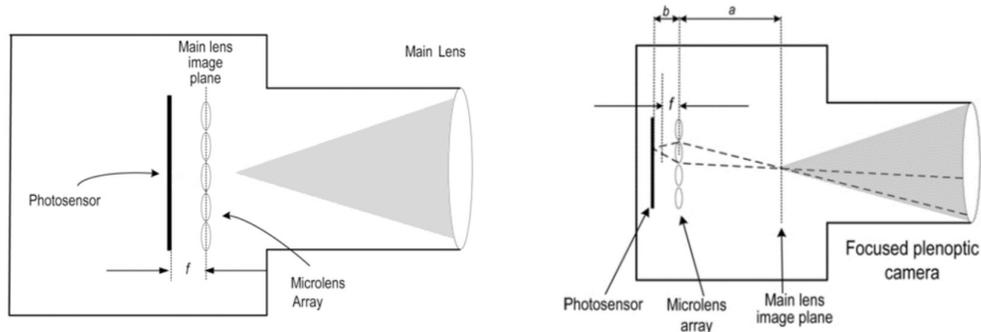


Figura 3.12. Izq.: Cámara plenótica convencional, o “Plenoptic 1.0”. Las lentes del objetivo hacen foco enfrente de la MML, que se posiciona a una distancia f del sensor de imagen, siendo f la focal de las microlentes. Der.: Cámara plenótica “Plenoptic 2.0”. Las lentes del objetivo hacen foco en un plano situado una distancia “ a ” enfrente de la matriz de microlentes, mientras que su distancia al sensor CMOS es $b > f$, donde f es la distancia focal de las lenticulas. Fuente: [39]

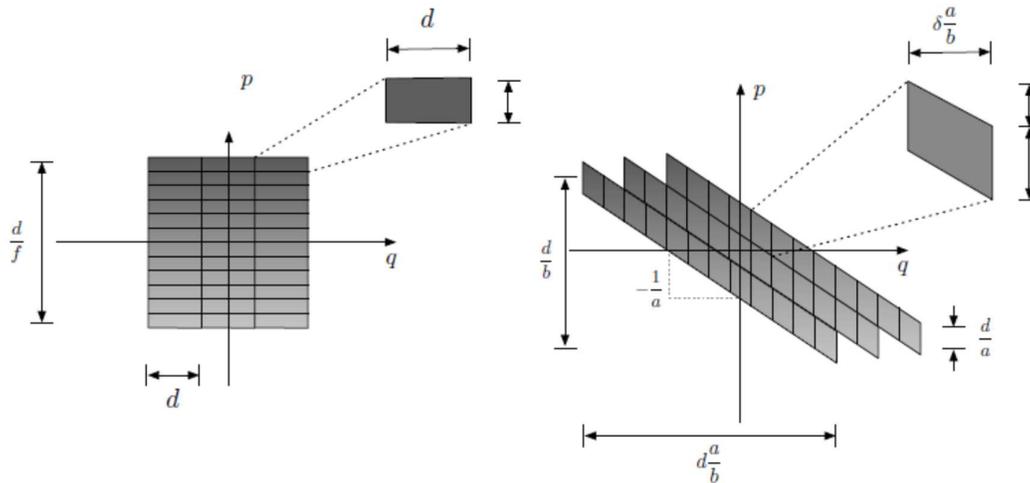


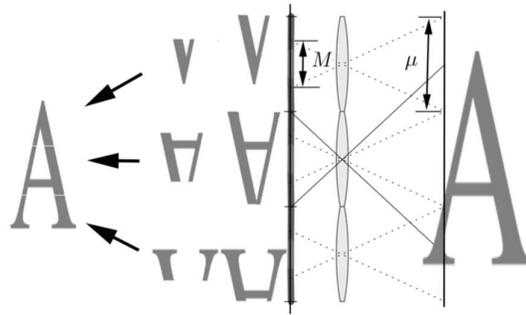
Figura 3.13. Izq.: espacio de las fases para geometría Plenoptic 1.0. Cada columna representa a los elementos de una misma microimagen y la renderización consiste en integrar dichos elementos, es decir, sumar para cada posición q el conjunto de pixeles que capturan la escena desde diferentes ángulos p . Der.: espacio de las fases para geometría Plenoptic 2.0. Cada columna correspondiente a diferentes posiciones q contiene elementos de diferentes microimágenes. La dimensión d es el paso o *pitch* de las microlentes y δ es el tamaño del fotosito del sensor. Fuente: [58].

En la Figura 3.13 de la derecha, el cálculo de la radiancia correspondiente a cada punto espacial involucra a pixeles pertenecientes a diferentes microlentes y una consecuencia importante es que la resolución de las renderizaciones con Plenoptic 2.0 no dependen del número de lenticulas y puede alcanzar la misma cantidad de pixeles que la imagen de radiancia original, lo que Lumsdaine y Georgiev llaman “*full resolution*” [38], [63].

Lo importante de este proceso es comprender la importancia del tamaño del sub-bloque. La Figura 3.14 muestra un ejemplo, que esquematiza el proceso. La imagen de ejemplo (letra \mathcal{A}) está en el plano imagen del objetivo, a una distancia “ a ” del plano de las microlentes y éstas, a su vez, a una distancia “ b ” del plano del sensor (cabe recordar que “ b ” es mayor que la distancia focal “ f ” de las microlentes).

El paso de las microlentes (conocido como *pitch* en inglés) es μ y el tamaño del sub-bloque seleccionado es $M \times M$. Observamos que cada microlente obtiene una subimagen invertida y parcial de la letra \mathcal{A} . A su vez, el sub-bloque $M \times M$ recorta dicha subimagen a otra de menor tamaño. Si cada sub-bloque se invierte y se concatena, puede obtenerse la imagen de la letra \mathcal{A} original. Vemos que en este caso, el tamaño $M \times M$ seleccionado provee un encaje perfecto de las partes tal que la imagen obtenida está libre de artefactos. El tamaño M con el que se renderiza la imagen se denomina *Pitch de Renderizado* (*rendering pitch*).

Figura 3.14. Esquema de formación de imagen en una cámara plenóptica enfocada de paso μ . Sólo cuando se ajusta el tamaño de los bloques a un tamaño M , puede renderizarse la imagen de la letra A correctamente. Fuente: [38].



Si el M seleccionado es tal que satisface:

$$\mu = M \cdot \frac{a}{b} \quad (1)$$

significa que el plano imagen situado a una distancia “a” de las microlentes está en foco y la imagen renderizada estará libre de artefactos (encaje perfecto de los sub-bloques). En otras palabras, la imagen renderizada está focalizada sólo en el plano según la elección de M . Eligiendo M , los sub-bloques encajarán perfectamente sólo para un plano imagen, registrándose artefactos para el resto de las profundidades en la escena. Cuando se satisface (1), se alcanza la máxima resolución correspondiente con el plano imagen en foco, denominándose renderizado “*full resolution*” este método.

Diferentes tamaños M de sub-bloques corresponden a diferentes profundidades en la escena. Esto es, diferentes profundidades de una escena requieren diferentes tamaños M de sub-bloques para ser renderizados apropiadamente. Entonces, para un valor fijo de M diferentes profundidades en una escena presentarán artefactos de no-coincidencia de los sub-bloques mientras que aquella que satisfaga (1), se verá perfectamente y a la máxima resolución, equivalente a b/a veces la resolución del sensor.

La Figura 3.15 esquematiza un ejemplo [38]. A la izquierda, el caso donde el pitch de renderizado M es demasiado pequeño para la escena, y el caso contrario para la imagen de la derecha.

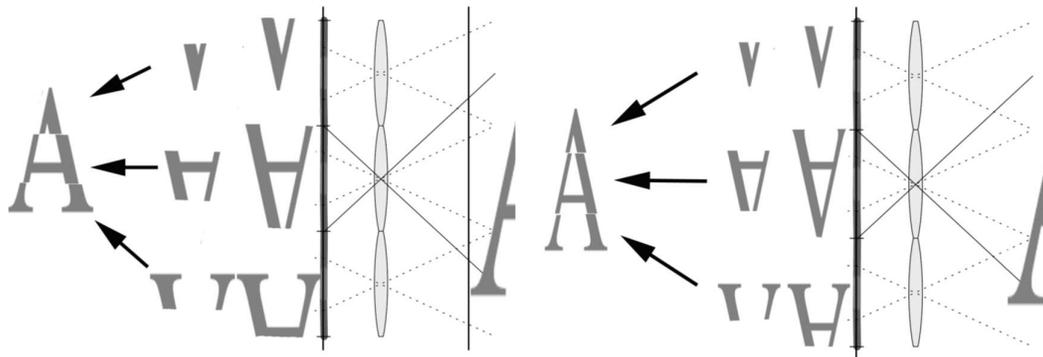


Figura 3.15. Renderización por bloques para Plenoptic 2.0. Izq.: el bloque seleccionado es pequeño y el resultado es una falta de registro de la imagen reenfocada. Der.: caso contrario al anterior. Fuente: [38].

El lado de la microlente si fuera de geometría cuadrada, o diámetro si fuera circular, denominado μ , hace el mejor aprovechamiento del sensor cuando cubre bajo toda su área los correspondientes fotositos (Figura 3.16).

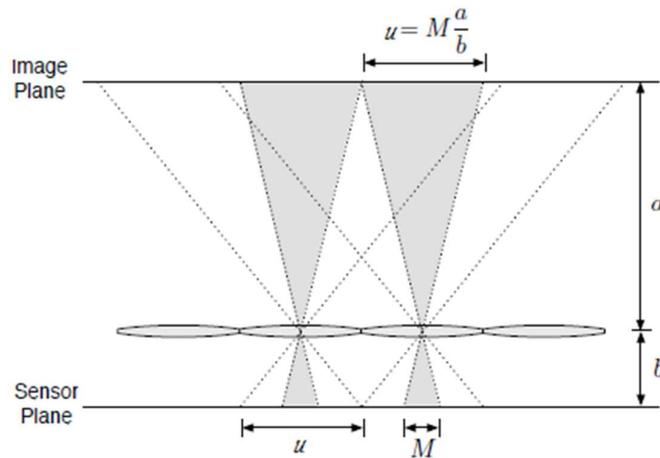


Figura 3.16. Relaciones entre el tamaño del bloque (M) y el tamaño de la microlente μ . Fuente: [58].

El renderizado básico consiste en tomar, de cada microlente de μ^2 elementos, un parche de M^2 elementos y conformar la imagen renderizada uniendo dichos parches. Puede observarse en las Figuras 3.14 y 3.15 que, según el tamaño de los parches, hay coincidencia (o continuidad) en la imagen renderizada para una determinada distancia a , que representa la ubicación del plano imagen virtual donde la escena está en foco. Para otras posiciones de dicho plano imagen, habrá superposición o espacios entre los parches. Entonces el tamaño de M determina qué parte (profundidad) de la escena se ve en foco, en la imagen renderizada.

Basados en los conceptos presentados en los párrafos precedentes, implementamos dos algoritmos para refocalizar imágenes capturadas con cámara de arquitectura Plenoptic 2.0.

La solución más sencilla es la implementación del algoritmo `CorteBloques` en sintaxis MATLAB, que consiste en utilizar las funciones `imcrop` y `blockproc` para recortar y unir los bloques, respectivamente. `imcrop` permite el recorte de una imagen ingresando las coordenadas de renglón-columna de las esquinas del recorte. `blockproc` permite el procesamiento por bloques de una imagen, definiendo el tamaño de dichos bloques y especificando en una función `fun` el proceso específico.

Pseudocódigo 8.- CorteBloques

Input: a) matriz imagen `im` de $N_x \times N_u \times N_y \times N_v$ elementos, modelo de color RGB. b) `mu`=pixeles de lado de una microlente= $N_u = N_v$. c) `M`: pixeles de lado de un bloque por microlente, $M < \mu$.

Output: matriz imagen **renderizada** de $N_x \times N_u \times N_y \times N_v$ elementos, modelo de color RGB.

For cada CanalColor

For cada microimagen

 Block = recorte central de $M \times M$ pixeles

 Block = Block rotado 180 grados

End

End

renderizada = unión de todos los Block

renderizada = **renderizada** reescalada a tamaño de `im`

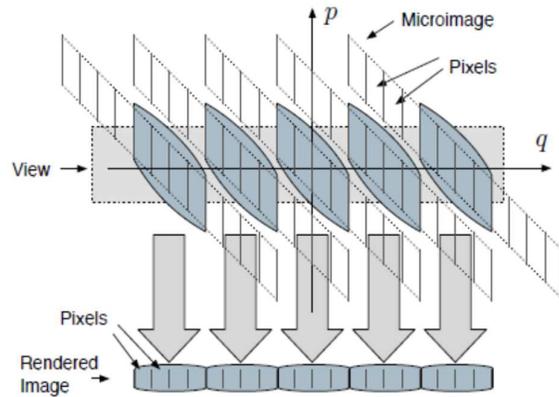
Previo a estas operaciones es necesario rotar 180° cada microimagen y finalmente, hacer un reescalado al tamaño original con la función `imresize`, que devuelve una imagen que es x veces el tamaño de la imagen original y permite diferentes tipos de interpolación (`nearest`, `bilinear`, `bicubic`, etc.). En la Figura 3.17 mostramos algunos resultados.



Figura 3.17. Resultados del algoritmo `CorteBloques`. Imagen original [40] de $\mu=75$. Izq.: Bloques de 5x5 píxeles, enfoca lejos (observar detalles en ventana y araña). Medio: Bloques de 7x7 píxeles, enfoca en sujeto que sostiene casco. Der.: Bloques de 7x7 píxeles, enfoca en sujeto de espaldas ubicado más cerca de la cámara (observar detalles del reloj pulsera).

En términos del espacio de las fases $q-p$, para el renderizado básico (también denominado renderizado basado en parches, *Basic Patch-Based Rendering*, por lo anteriormente dicho) vemos que los píxeles de la imagen renderizada se seleccionan uno a uno de cada microimagen, en la cantidad determinada por el tamaño M , Figura 3.18. En dicha Figura se ha seleccionado el subconjunto del centro de los elementos de cada microimagen. Si se seleccionaran subconjuntos del mismo tamaño pero en otras posiciones dentro de las microimágenes, se observaría la imagen renderizada enfocada en la misma profundidad de la escena, pero desde perspectivas diferentes.

Figura 3.18. Espacio de fase q-p de una arquitectura Plenoptic 2.0. Caso de una vista de $M=5$ centrada en la microimagen de 13×13 pixeles. Fuente: [58].



Si consideramos realizar el renderizado directamente desde la radiancia aplanada, la Figura 3.19 ayuda a comprender el proceso. Se observan algunas microimágenes de tamaño d pixeles, de las cuales se han seleccionado parches centrados de tamaño M^2 pixeles para reconstruir una imagen por el método básico. La idea es, a partir de una posición x en la imagen renderizada determinar la posición correspondiente en el sensor de imagen, cuya intensidad deberá indexar en x .

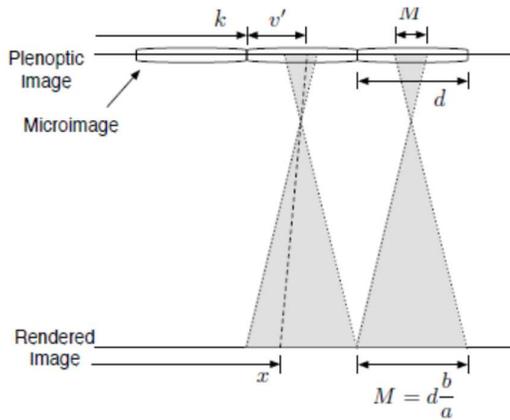


Figura 3.19. Diagrama simplificado del algoritmo de Georgiev Basic Patch-Based Rendering. Fuente: [58].

Basados en la Figura 3.19, ilustramos el desarrollo del algoritmo con un ejemplo numérico. Supóngase una radiancia aplanada de sólo dos microimágenes de 7×7 pixeles y que se desea renderizar con parches centrados de tamaño 3×3 (en negritas dentro de borde a rayas), como se muestra a continuación:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
15	16	17	18	19	20	21	22	23	24	25	26	27	28
29	30	31	32	33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96	97	98

En primer lugar, obsérvese de la Figura 3.19 que la imagen renderizada no cambia de tamaño: el parche abarca el tamaño de la microimagen, según un factor de amplificación igual a a/b . Para el ejemplo, debería esperarse que los parches originales de 3×3 se expandan con sus valores de intensidad al tamaño de las microimágenes de 7×7 .

El algoritmo propuesto por Georgiev et al. [38] considera, dada una posición x de la imagen de salida (renderizada), hallar la correspondiente posición en la imagen de entrada (radiancia aplanada). Para el caso de ejemplo, se podría querer determinar qué valor asumiría la imagen renderizada en la posición [renglón , columna] = (3,9) (posición donde la imagen original tiene una intensidad de 37).

Lo primero es determinar de cuál de las microlentes se renderizará la posición x . Para ello se calcula la parte entera k del cociente x/μ , que marcará la microlente de pertenencia:

$$k = \left\lfloor \frac{x}{\mu} \right\rfloor$$

En sintaxis de MATLAB, la función `floor` redondea el cociente al entero más cercano hacia menos infinito. En este caso, para los renglones: `floor(3/7)=0` y columnas: `floor(9/7)=1`. La posición del pixel de inicio de aquella microlente es resultado del producto $k\mu$, es decir, en renglón: 0 y en columna: 7. A continuación se calcula el corrimiento correspondiente a x dentro del parche de tamaño M . Para ello, se computa la diferencia entre x y el pixel de inicio de la microlente $k\mu$:

$$q = \left(x - \left\lfloor \frac{x}{\mu} \right\rfloor \mu \right) \frac{M}{\mu} = \left(\frac{x}{\mu} - k \right) M$$

Para el ejemplo este valor es (3, 2), pero se requiere conocer dicho corrimiento en la imagen de entrada, para lo cual es necesario multiplicar por el factor de amplificación M/μ lo que da (1.2857, 0.8571). Pero es necesario otro paso más: centrar el parche $M \times M$ de la imagen de entrada a la correspondiente región en la imagen de salida. Para ello debe sumarse un corrimiento de $(\mu-M)/2$:

$$qp = q + \frac{\mu - M}{2} = \left(\frac{x}{\mu} - k \right) M + \frac{\mu - M}{2}$$

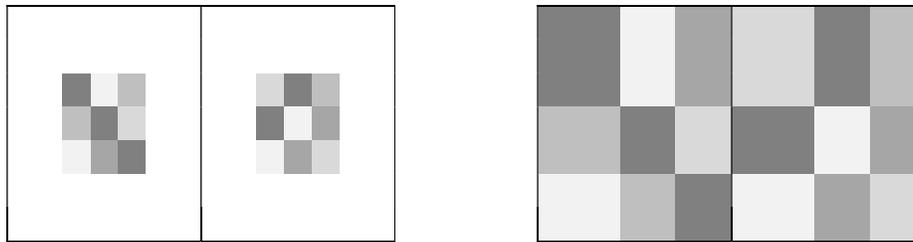
Para el ejemplo, esta posición es (3.2857, 2.8571). Finalmente, este corrimiento qp calculado debe sumarse al número de microlente k calculado antes, siendo la posición correspondiente en la imagen de entrada para un punto x dado de la imagen de salida:

$$fx = k\mu + qp$$

Entonces la posición (3,9) en la imagen renderizada asume el valor de la radiancia aplanada en la posición (3.2857, 9.8571). Dado que los índices deben ser números enteros, se redondea el resultado fx a (3,10). Entonces la posición (3,9) de la imagen renderizada tendrá un valor de intensidad correspondiente a la posición (3,10) de la radiancia aplanada, es decir, de 38. La imagen renderizada resultante es la siguiente:

31	31	31	32	32	33	33	38	38	38	39	39	40	40
31	31	31	32	32	33	33	38	38	38	39	39	40	40
31	31	31	32	32	33	33	38	38	38	39	39	40	40
45	45	45	46	46	47	47	52	52	52	53	53	54	54
45	45	45	46	46	47	47	52	52	52	53	53	54	54
59	59	59	60	60	61	61	66	66	66	67	67	68	68
59	59	59	60	60	61	61	66	66	66	67	67	68	68

Obsérvese en el siguiente esquema que los valores de intensidad correspondientes a los parches de 3x3 de la radiancia aplanada, se distribuyen en la imagen renderizada ocupando la dimensión original 7x7 de cada microlente. La gran ventaja del algoritmo de Georgiev es justamente que la resolución de la imagen no depende del número de microlentes.



Nuestra implementación en MATLAB requiere, en primer lugar, rotar 180° cada una de las microlentes, dado que la configuración Plenoptic 2.0 enfoca en las microlentes una imagen real invertida que forma el objetivo dentro del cuerpo de la cámara. Luego sí, calculamos los índices y renderizamos la imagen enfocada según el tamaño del bloque M elegido. Incluimos además un valor de $offsetX, offsetY$ que indica el corrimiento del bloque en cantidad de pixeles en X e Y respectivamente, para brindar una salida renderizada desde diferentes perspectivas. La función se denomina `BasicFullRender(im, mu, M, offsetX, offsetY)` acepta como parámetros la imagen de radiancia de Plenoptic 2.0, el valor en pixeles de lado μ de la microimagen (igual a 75 para las imágenes de dominio público cedidas por Georgiev), el valor en pixeles de lado M del bloque y los corrimientos mencionados para obtener diferentes perspectivas.

La Figura 3.20 muestra dos resultados para diferentes tamaños de bloque: M pequeño implica enfoque lejano a la cámara, y viceversa para M grande. Obviamente, debe respetarse que el valor de M no supere al valor de μ . Lo mismo sucede con los offset, que no deben superar un valor tal que el corrimiento saque fuera de la microimagen al bloque elegido. Entonces el algoritmo de Georgiev no encuentra la limitante de resolución dada por el número de microlentes como sucede con Plenoptic 1.0. Sin embargo, existe una relación de compromiso entre la cantidad de vistas en perspectiva y la profundidad donde hace foco la renderización. Mencionamos que M grande enfoca cerca y M pequeña, lejos. Si enfocamos cerca, la cantidad de desplazamientos del bloque dentro de la microimagen es menor (dado su tamaño grande) que si enfocamos lejos. Por lo tanto, un enfoque cercano nos permite menor cantidad de perspectivas que un enfoque lejano.



Figura 3.20. Izq. Bloque $M=6$. Der.: Bloque $M=12$. Ambas con $\text{Offset}=0$. Imagen *Bike.jpg* de dominio público, cedida por Georgiev [40].

Obsérvese los artefactos que se ven como bloques desplazados, en la imagen de la izquierda en la parte anterior de la cerca en sus punteras curvas y en la de la derecha, en el ciclista y fondo. Esto se debe al propio algoritmo que, según el tamaño de M elegido, hace que algunos bloques registren bien unos con otros (donde hace foco la renderización) y otros, no coincidan sus bordes. Más adelante propondremos una mejora a este artefacto.

Si comparamos los scripts `CorteBloques` con `BasicFullRender` obtenemos exactamente los mismos resultados, pero las funciones `imcrop`, `blockproc` e `imresize` de `CorteBloques` son menos eficientes para hacer el trabajo que la propuesta de indexación original de Georgiev, mucho más directa sobre cada pixel de la radiancia original. Los tiempos que insume cada algoritmo para la imagen *Bike.jpg* y $M=6$ son:

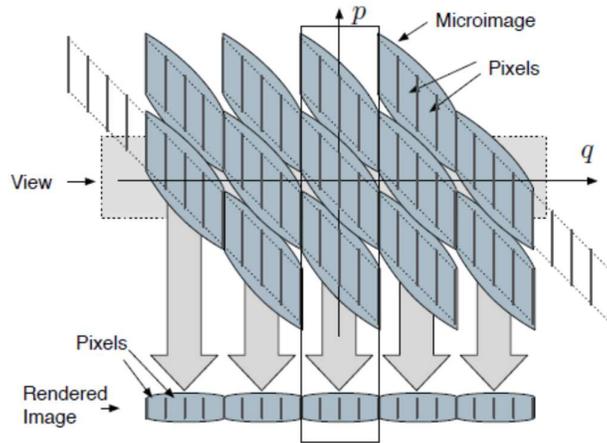
```
BasicFullRender: 1.162136 seconds
CorteBloques: 2.523296 seconds.
```

Un análisis con la herramienta `Profile Summary` de MATLAB indica que el mayor costo computacional en `BasicFullRender` corresponde al proceso de rotar 180° cada microimagen de la radiancia original. Para el caso de `CorteBloques`, el mayor peso del script corresponde a las funciones `imcrop` e `imresize`.

3.3.1.4. Renderizado por bloques con suavizado

Dados los artefactos que surgen de las no coincidencias de los bordes de los bloques en las zonas fuera de foco, buscamos una alternativa de suavizado de este fenómeno. En la Figura 3.18 mostramos el caso de renderizado por bloques de una vista centrada en la microimagen. Si el bloque de igual tamaño M se colocara en otra posición dentro de la microimagen, obtendríamos otra perspectiva de la imagen renderizada, a la misma profundidad. En términos del diagrama de las fases q - p , siempre integramos en q (posición) para diferentes ángulos (p). Ahora bien, si en la integración involucramos mayor amplitud angular, esto es equivalente a promediar los mismos puntos espaciales que se ubican en diferentes microlentes (vecinas). La Figura 3.21 muestra la interpretación del renderizado con suavizado (*blending*) desde el diagrama de las fases q - p .

Figura 3.21. Espacio de fase q-p de una arquitectura Plenoptic 2.0, que integra bloques de microimágenes vecinas para lograr el efecto de suavizado de las zonas fuera de foco. Fuente: [55].



Podemos observar que se integran los píxeles de igual posición, pero ángulos diferentes. Esto es, podemos lograr el suavizado promediando renderizaciones de igual M y diferentes perspectivas. El algoritmo `renderblock_blend` simplemente promedia un determinado número de vistas para un mismo M , obtenidas con el algoritmo `BasicFullRender`. En la Figura 3.22 comparamos la renderización de la misma imagen con ambos métodos.



Figura 3.22. Izq.: Renderización por Bloques convencional. Der.: Renderización por Bloques con suavizado, que involucra sólo 8 vecinos (promediado de 9 vistas en perspectiva) para $M=10$. Imagen Jeff.jpg de dominio público, cedida por Georgiev [40].

Los artefactos observados en el ángulo inferior derecho en ambas imágenes de la Figura 3.22 se deben a que la imagen de radiancia cruda comienza en la primera posición (ángulo superior izquierdo) desplazada algunos píxeles, y el proceso de bloques va acumulando el corrimiento de píxeles haciéndose más acentuado en las últimas posiciones. Aun así, mejora la percepción del fondo, quedando en foco el frente (dado por M alto) y desenfoándose con borronco el fondo.

3.3.1.5. Transformada Rápida Discreta del Focal Stack

Pérez Nava y colaboradores [64] [65] proponen otra forma de reenfoque digital. Ellos extraen una pila focal de una imagen cruda de radiancia a partir de una discretización diferente del Operador

Fotográfico propuesto por Ng [66]. Basan su trabajo en la interpolación del LF por medio de polinomios trigonométricos 4D y una aproximación al proceso de integración. Su ganancia es una reducción de la complejidad computacional de la solución del Operador Fotográfico.

Los autores han hecho de dominio público los scripts, de modo que nosotros hemos hecho algunos ajustes y mostramos los resultados. Cabe destacar que, si bien los autores elogian su bajo costo computacional, el algoritmo ha insumido tanto tiempo que fue necesario recortar el LF original y optamos por hacer un stack focal de sólo cinco imágenes, a partir de la radiancia capturada con la Lytro F01. En la Figura 3.23 presentamos 4 imágenes de 256x256 píxeles, enfocadas a diferentes profundidades. Observamos que la calidad de las renderizaciones es muy buena, pero es necesario reformular el código para hacerlo más eficiente en cuanto al tiempo de cómputo. Obtener esta pila de 5 imágenes requirió casi 30 minutos.

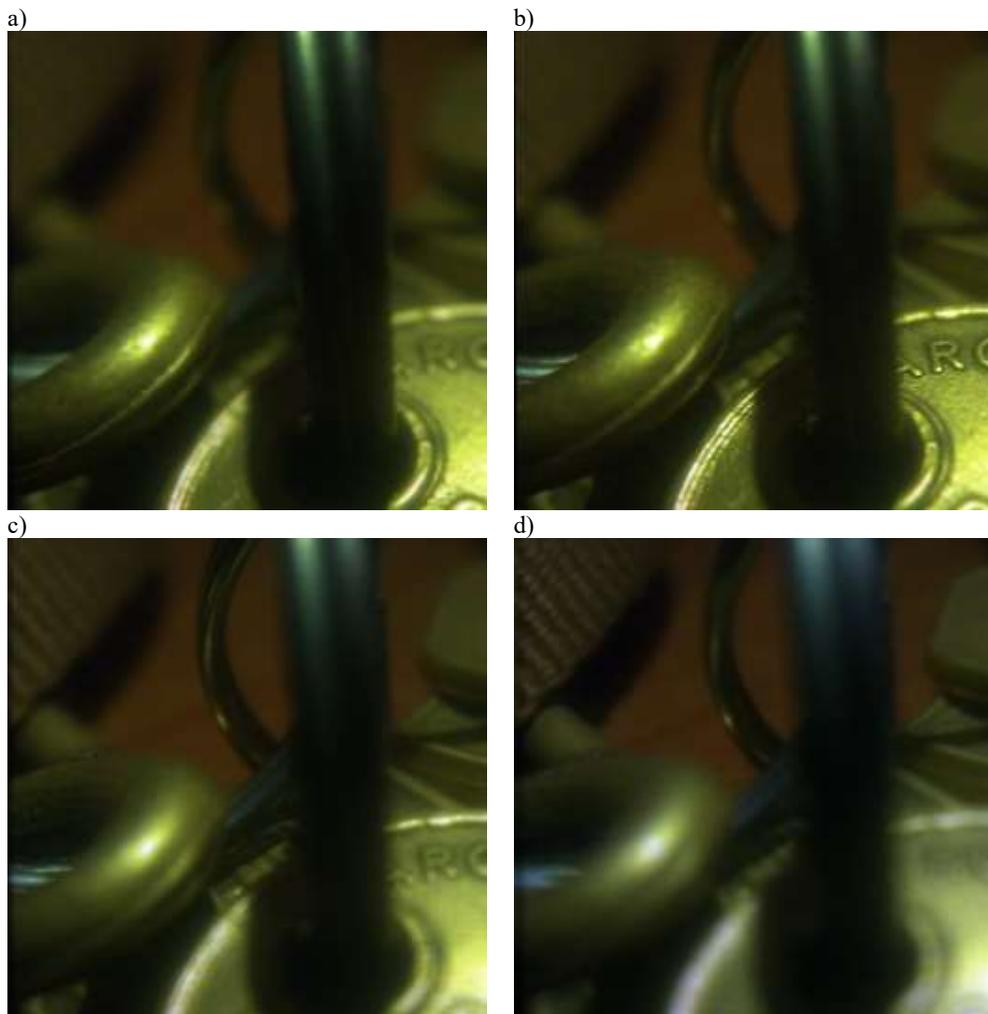


Figura 3.23. Pila Focal de 4 imágenes obtenida con el algoritmo Transformada Rápida Discreta del Focal Stack. Imagen capturada con Lytro F01 y recortada a $N_x=N_y=256$. $N_u=N_v=11$. Desde a) hasta d) el enfoque va pasando desde los planos más cercanos a la cámara (detalle en ángulo inferior derecho de a)) hasta los más alejados (detalle en ángulo superior izquierdo de b)).

3.4. Teorema del Corte de Fourier y refocalización

El LF 4D es una representación matricial de la radiancia de la escena, como hemos visto. Nos interesa recuperar una imagen enfocada a una determinada profundidad, para lo cual estudiamos en los párrafos anteriores cuatro propuestas de refocalización en el dominio espacial del LF.

La Transformada de Fourier de una señal brinda un enfoque diferente y cuantificable de su contenido frecuencial y como sabemos, también las imágenes pueden ser estudiadas desde sus espectros. De la misma manera, podemos estudiar la radiancia desde el punto de vista espectral. Es interesante comparar y evaluar la refocalización de la escena a diferentes profundidades a partir de la radiancia por un lado y de su Transformada de Fourier (TF), por otro. Verificar si los resultados son consistentes con las relaciones matemáticas que guardan el LF y su hiperespacio transformado y su eficiencia en términos de tiempo de cómputo.

Sea $r(x)$ la radiancia de un rayo x parametrizada como posición-ángulo $r(x) = r(q, p)$, podemos proponernos representarla en su dominio frecuencial a partir de la TF.

En el dominio frecuencial representaremos la radiancia $r(q, p)$ como $R(\omega) = R(\omega_q, \omega_p)$ a partir de la TF:

$$R(\omega) = \int r(x) e^{i\omega x} dx$$

donde $\omega = \begin{bmatrix} \omega_q \\ \omega_p \end{bmatrix}$, $r(x) = r(q, p)$ y $\omega x = \omega_q q + \omega_p p$.

Por otra parte, la óptica geométrica estudia las transformaciones que sufren los rayos, según se propaguen una distancia cualquiera y/o se refracten por su paso a través de una lente. En la Figura 3.24 esquematizamos estas situaciones. Vemos que cuando el rayo se propaga una distancia t , la posición q cambia a q' y la pendiente p se conserva mientras el rayo se propague libre de ocluidores o refractores.

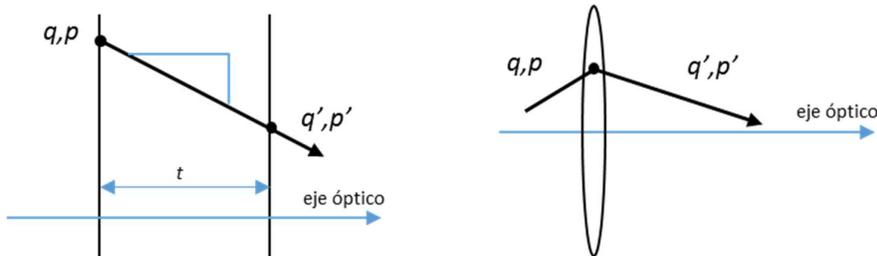


Figura 3.24. Transformación de un rayo según se desplace o refracte, según la parametrización posición – ángulo. Izq.: desplazamiento una distancia t . Der.: refracción en una lente fina (a mayor distancia del eje óptico, mayor refracción).

Dado que la pendiente es $p = (q - q')/t$, entonces:

$$p' = p \quad (\text{la pendiente se conserva})$$

$$q' = q - tp \quad (\text{la posición cambia})$$

y en notación matricial,

$$\begin{bmatrix} q' \\ p' \end{bmatrix} = \begin{bmatrix} 1 & -t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q \\ p \end{bmatrix} = T \begin{bmatrix} q \\ p \end{bmatrix}$$

Si un sistema óptico transforma el rayo x a x' , entonces en términos de una matriz de transformación A ,

$$x' = Ax \quad (2)$$

Los elementos ópticos también transforman la radiancia. Pero la radiancia se conserva. Entonces la radiancia r' del rayo transformado x' es la misma que la radiancia r del rayo x , por el principio de conservación de la radiancia en un espacio libre de absorbentes y ocluidores, es decir, $r'(x') = r(x)$.

Multiplicando ambos miembros de (2) por A^{-1} (la inversa de A):

$$A^{-1}x' = x$$

Entonces la radiancia r' después de la transformación óptica A se relaciona con la radiancia original r según la siguiente expresión:

$$r'(x') = r(x) = r(A^{-1}x') \quad (3)$$

Dado que el rayo x' es arbitrario, podemos reemplazarlo en (3) por cualquier rayo x :

$$r'(x) = r(A^{-1}x) \quad (4)$$

Si la transformación A es una propagación, entonces

$$A^{-1} = T^{-1} = \begin{bmatrix} 1 & -t \\ 0 & 1 \end{bmatrix} \quad \text{y} \quad r'(x) = r'(q, p) = r(q - tp, p)$$

Dada una transformación óptica $x = Ax_0$ y la radiancia se transforma a $r'(x)$, entonces:

$$\begin{aligned} R'(\omega) &= \int r'(x) e^{i\omega \cdot x} dx \\ &= \int r(A^{-1}x) e^{i\omega \cdot AA^{-1}x} dx \\ &= \int r(x_0) e^{i\omega \cdot Ax_0} dx_0 \\ &= \int r(x_0) e^{iA^T \omega \cdot x_0} dx_0 \\ &= R(A^T \omega) \end{aligned} \quad (5)$$

La expresión (5) indica que en el espectro de Fourier la radiancia se transforma por la traspuesta de la matriz de transformación A .

Veamos ahora a qué es equivalente la TF de una imagen refocalizada en alguna posición. Recordemos que, en términos del espacio de las fases, una imagen se forma al integrar los rayos desde todas las direcciones p en cada posición q , es decir,

$$I(q) = \int_p r(q, p) dp$$

Su TF es:

$$\begin{aligned} I(\omega_q) &= \int I(q) e^{i\omega_q q} dq \\ &= \iint r(q, p) e^{i\omega_q q} dp dq = R(\omega_q, 0) \end{aligned} \quad (6)$$

Dado el valor cero correspondiente a ω_p , podemos decir que la TF de la imagen renderizada es la componente DC direccional del espectro $R(\omega)$ de su LF. Si la transformación de la radiancia es una traslación (es decir, estamos queriendo calcular la radiancia en otro plano, esto es, refocalizar la imagen), recordemos que:

$$r'(q, p) = r(q - tp, p)$$

En términos del espectro, $R'(\omega_q, \omega_p) = R(\omega_q, \omega_p - t\omega_q)$ por (5) y $I(\omega_q) = R(\omega_q, 0)$ por (6). Entonces:

$$I'(\omega_q) = R'(\omega_q, 0) = R(\omega_q, -t\omega_q) \quad (7)$$

La expresión 7 nos está indicando que el espectro 2D de una imagen refocalizada se obtiene del espectro 4D del LF original, seleccionando para cada posición ω_q un valor entre las direcciones ω_p equivalente a su producto por una pendiente t .

Nuestro algoritmo reproduce estos resultados teóricos. La función `[Refocalizada] = RefocusEspectro(LF, theta)` reenfoca la escena en diferentes profundidades según el parámetro `theta`. Obtiene un plano transformado 2D a partir de tomar de la FFTLF (4D), un pixel en la nueva posición `(uu, vv)` para cada lenticula `(XX, YY)` según la siguiente relación:

$$\begin{aligned} uu &= theta * (XX - m/2) + p/2; \\ vv &= theta * (YY - n/2) + q/2; \text{ donde } [m, n, p, q] = \text{size}(LF) \end{aligned}$$

Aplicando la Transformada Inversa de Fourier 2D, recuperamos la imagen `Refocalizada`. Trabajamos con grillas `ndgrid` y la función `interp` para interpolar los valores de FFTLF en las posiciones `YY, XX, vv, uu`.

La Figura 3.24 muestra algunos resultados. $\theta = 0$ no reenfoca, da como resultado la imagen cruda cuyo foco es el capturado según la óptica. $\theta < 0$ enfoca al fondo (valores no menores a -0.05) y $\theta > 0$ enfoca al frente (valores no mayores a +0.05).



Figura 3.24. Refocalizaciones por corte del hiperespacio de Fourier de la radiancia. $\text{Alfa} < 0$ hace foco en el fondo de la escena (alejado de la cámara). $\text{Alfa} > 0$ hace foco en el frente de la escena (cercano a la cámara).

Ren Ng propuso otro enfoque para trabajar en el dominio frecuencial [66] [37]. En el Capítulo 2 estudiamos el espacio de los rayos en una cámara plenóptica y observamos que un punto en foco se representa como una línea vertical. Cualquier otro punto fuera del plano focal se representará como una línea con una determinada pendiente, correspondiente con las distancias.

Bracewell en 1956 [67] estudiaba la reconstrucción de imágenes en un radar de apertura sintética y descubrió una relación que, dos décadas después, contribuiría a la obtención de cortes con la Tomografía Computada (TC). Una proyección paralela en TC es el vector (1D) cuyos valores son la integración de los valores de atenuación de un corte de tejido en una determinada dirección o ángulo Θ . Dicho ángulo corresponde a la rotación del sistema Tubo-Detector en un tomógrafo de geometría paralela (primera generación) respecto de un sistema coordenado de referencia. En la Figura 3.25, las rectas azules representan los rayos paralelos que atraviesan el tejido y la señal $g_{\theta}(x')$ es la proyección, siendo cada uno de sus puntos el valor de la integral de línea a lo largo de cada rayo paralelo. El denominado Teorema del Corte de Fourier establece que la Transformada de Fourier de dicha proyección tomada a un ángulo Θ , corresponde a los valores

del Plano Transformado de Fourier (2D) ubicados sobre un recta subtendida el mismo ángulo Θ . Entonces, si se adquirieran proyecciones en todos los ángulos sería posible construir el espectro 2D y recuperar, por transformación inversa, la imagen del corte. Dado que es imposible obtener infinitas proyecciones, es necesario interpolar valores del espectro para completar la información faltante. Esta es una ardua tarea desde el punto de vista matemático dado que se trata de valores complejos (información de módulo y fase), por lo que en la actualidad las reconstrucciones utilizan otras estrategias matemáticas como la convolución-retroproyección filtrada.

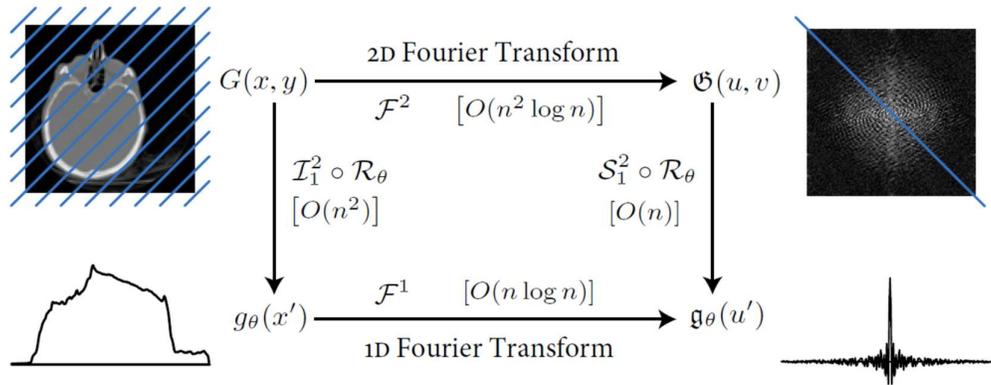


Figura 3.25. Esquema del Teorema del Corte de Fourier. Fuente: [37].

Si observamos la Figura 3.26, el espacio de los rayos se observa como proyecciones cuyos ángulos Θ se corresponden con diferentes planos de foco en la escena. Ren Ng supo interpretar esta similitud con el Teorema del Corte de Fourier y propuso utilizar el espacio transformado del LF para recuperar imágenes enfocadas en diferentes profundidades de la escena [66].

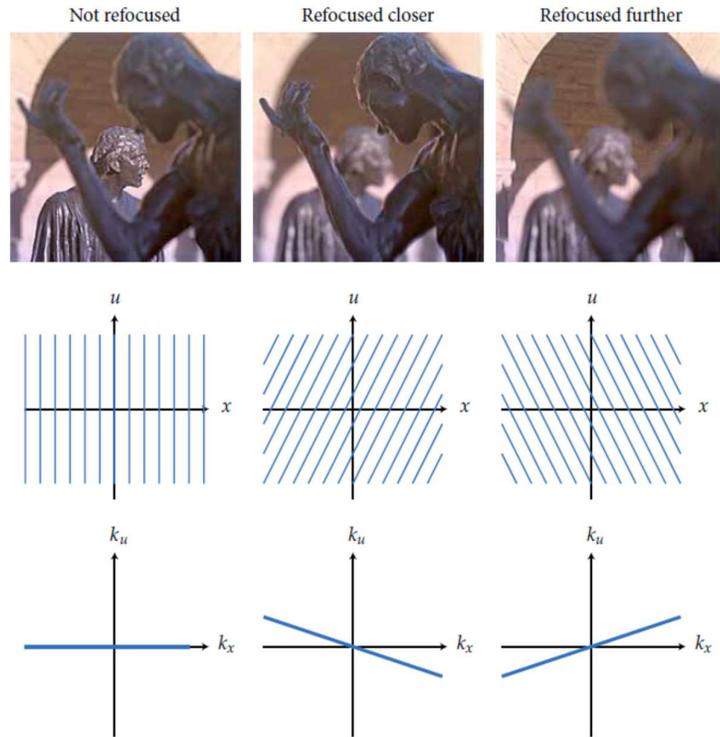
Con una cámara plenóptica codificamos la radiancia de la escena como un LF 4D. Como hemos visto, no es más que una “matriz de matrices” siendo (x, y) las posiciones de las lenticulas y (u, v) las posiciones de los pixeles debajo de cada lenticula, pero codificadas como una matriz cuatrodimensional $\text{LF}(x, y, u, v)$. Así como el “corte” del plano transformado de Fourier es una recta que pasa por el origen, de acuerdo a lo que consigna el Teorema, la pregunta es: ¿a qué es equivalente el “corte” del espacio 4D de Fourier del LF? En este momento cabe dedicar un momento a conceptos de geometría 4D.

Dado que el espacio que nos rodea tiene geometría 3D, nos resulta difícil interpretar espacios de mayores dimensiones (denominados “hiperespacios”). Pero es interesante notar que la geometría de nuestro espacio 3D puede extenderse a n dimensiones sin perder generalidad. Esto es, un plano 3D puede expresarse como:

$$d_x x + d_y y + d_z z = k$$

y su orientación se describe por un vector normal 3D. Un punto perteneciente a dicho plano tiene 2 grados de libertad y la intersección de dos planos en el espacio 3D es una recta.

Figura 3.26. Relaciones entre la posición del foco (primera fila), sus proyecciones en el dominio espacial (espacio de los rayos en fila del centro) y las Transformadas de Fourier de dichas proyecciones (fila inferior). Fuente: [37].



De la misma manera, un hiperplano en 4D se describe completamente por la ecuación:

$$d_s s + d_t t + d_u u + d_v v = k$$

Y un punto perteneciente a este hiperplano tiene 3 grados de libertad. La intersección de dos hiperplanos describe un plano y un punto perteneciente a dicho plano en 4D, tiene dos grados de libertad.

Teniendo en mente estos conceptos, cuando nos referimos a un “corte” en un plano 2D estamos hablando de una recta, mientras que un corte de un hiperespacio 4D es la intersección de dos hiperplanos cuyo resultado es un plano en 3D. Dansereau generaliza estas observaciones hechas en 2D al LF 4D y propone el diseño de filtros frecuenciales que son el resultado de la intersección de hiperplanos que definen hiperabanicos (hyperfans) e hiperconos (hypercones), basado en el trabajo de Bruton [68]. Para estudiar en detalle estos filtros remitimos al lector a la bibliografía [69] [70] [71].

En el dominio frecuencial, por Teorema del Corte de Fourier el conjunto de rectas paralelas del espacio de los rayos $u-x$ encuentra su par transformado en una recta ortogonal que pasa por el origen del plano transformado $\Omega_s-\Omega_u$. Como hemos mencionado oportunamente, la representación $u-x$ y $\Omega_s-\Omega_u$ es una simplificación 2D del caso general 4D. Así como estudiamos que una proyección paralela corresponde a puntos ubicados en un mismo plano focal, cuando estos puntos pertenecen a un rango de profundidades describen en el espacio de los rayos rectas no paralelas y su contraparte en el dominio frecuencial es un “doble abanico”. La Figura 3.27 muestra estas relaciones.

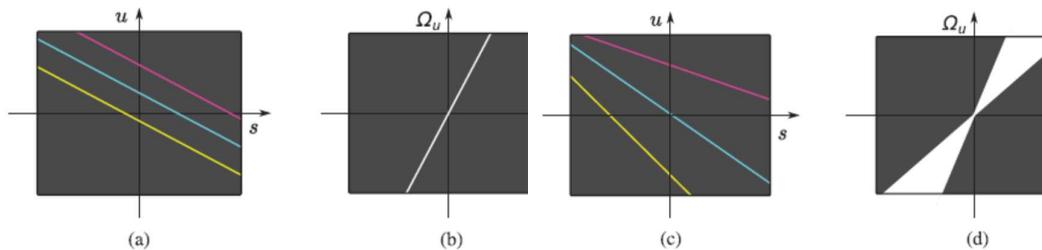


Figura 3.27. a) Puntos ubicados a una misma profundidad se mapean en el espacio de fase como rectas paralelas, cuya pendiente se relaciona con dicha profundidad. b) la proyección en a) se mapea en el dominio frecuencial como una recta que pasa por el origen, de igual pendiente que la proyección. c) puntos en diferentes profundidades corresponden a rectas de diferentes pendientes en el espacio de fase. d) el espectro de c) se convierte en un doble abanico cuya apertura depende del rango de profundidades mapeado. Fuente: [72].

Dansereau a partir de su tesis doctoral [72] generó un Toolbox para MATLAB, de dominio público, denominado “LFToolbox” cuya última versión v0.4 corresponde al año 2015. Nosotros utilizamos una función del mencionado Toolbox denominada `LFUtilDecodeLytroFolder` que nos permite decodificar el dato crudo de la captura con una cámara Lytro F01. Esta función devuelve un LF 5D cuya estructura es $LF(u, v, x, y, 4)$. Vemos que están invertidas las posiciones de los píxeles de cada lenticula y las posiciones de las mismas, en relación a nuestra convención $LF(x, y, u, v, 3)$. Por otra parte, observamos que además de los canales de color del modelo RGB, existe un cuarto canal denominado canal de peso (*weight channel*) que Dansereau comenta que será utilizado en versiones posteriores del Toolbox para operaciones de filtrado. Nosotros descartamos este canal y nos quedamos con los RGB. Nuestra función `dans2nos` permuta los parámetros del LF de manera que los acomoda a nuestra convención, comentada antes.

El LFToolbox presenta varias opciones de refocalización. Dansereau denomina “foco planar” al resultado de una refocalización en un plano a una determinada profundidad $z = z_0$ de la escena. Por otra parte este investigador define el “foco volumétrico” como aquella refocalización que pone en foco los objetos ubicados en un rango de profundidades $z = [z_a, z_b]$ [73]. Los filtros mencionados han sido implementados en el LFToolbox v0.4 y presentamos a continuación algunos resultados de su aplicación a imágenes capturadas con Lytro F01.

2D Line Frequency Filter

`LFBuild2DFreqLine` construye un filtro como cascada de dos filtros de línea en los planos de fase (s, u) y (t, v) , siendo el resultado idéntico a un filtro planar 4D. El tiempo aproximado para hacer el cálculo es de unos 10 segundos para una imagen capturada con Lytro F01 ($N_{xy}=378$, $N_{uv}=11$). Dada su similitud con el filtro planar 4D e idénticos resultados, mostraremos algunas pruebas de refocalización en el siguiente punto.

4D Planar Frequency Filter

`LFBuild4DFreqPlane` construye el filtro y `LFfilt4DFFT` lo aplica. Este filtro genera un plano de corte al espacio transformado del LF 4D. Es equivalente a nuestra implementación `RefocusEspectro(LF, theta)` y reenfoca la escena en un plano a una profundidad z . En la

Figura 3.28 mostramos algunos resultados a diferentes profundidades, según el `slope` elegido. Cada operación de filtrado insume aproximadamente 10 segundos.



Figura 3.28. Filtro Planar Frecuencial. Izquierda: refocalización anterior correspondiente a un plano de corte de pendiente $\text{slope}=1/3$. Centro: refocalización intermedia, pendiente $\text{slope}=3/9$. Derecha: refocalización en el fondo de la escena, pendiente $\text{slope}=8/9$.

4D Hyperfan filter

`LFBuild4DFreqHyperfan` construye el filtro y `LFfilt4DFFT` lo aplica. Este filtro requiere como entrada dos pendientes diferentes `slope1` y `slope2` que determinan el rango de profundidades que se renderizan en foco entre dichas pendientes. En la Figura 3.29 mostramos los resultados para dos rangos, que brindan un foco volumétrico anterior (cercano a la cámara) y otro posterior (alejado de la cámara). Comparamos con la imagen de entrada, de foco óptico original, obtenida con `render (LF)`. Cada operación de filtrado lleva unos 13 segundos.

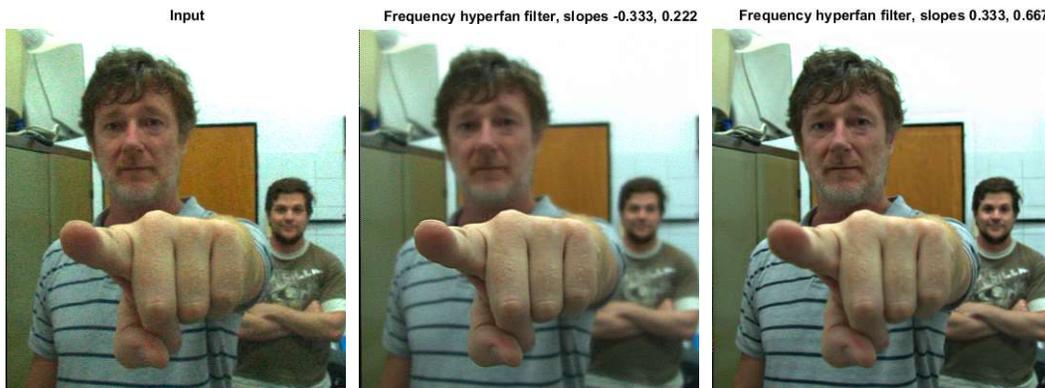


Figura 3.29. Filtro frecuencial Hyperfan para foco volumétrico. Al centro: foco volumétrico anterior obtenido con pendientes $\text{Slope1} = -3/9$; $\text{Slope2} = 2/9$. Derecha: foco volumétrico posterior obtenido con pendientes $\text{Slope1} = 3/9$; $\text{Slope2} = 6/9$.

Capítulo 4

Visualización tridimensional

Las texturas, sombras y oclusiones en una imagen permiten inferir su contenido en tres dimensiones, pero requieren un grado de conocimiento previo basado en experiencias anteriores (aprendizaje) para obtener una idea cabal de las profundidades en la escena. La posibilidad de sumar información con gráficos tridimensionales puede agregar alto valor a las imágenes. Hemos visto en el Capítulo 1 que es posible obtener información de profundidad a partir de múltiples capturas o sólo de una. Remitimos al lector a consultar la abundante bibliografía propuesta. En los Capítulos 2 y 3 estudiamos diferentes estrategias para, a partir de un LF, renderizar imágenes reenfocadas en diferentes profundidades de la escena. En este Capítulo haremos una breve introducción a los fundamentos para la confección de mapas de profundidad a partir de imágenes plenópticas, al sólo efecto de ver un ejemplo de aplicación y agregado de valor que puede aportar este tipo de fotografía computacional.

4.1. La pila focal

Denominamos “pila focal” (*focal stack* en inglés) al conjunto de las imágenes enfocadas en diferentes profundidades de la escena. Dichas imágenes pueden obtenerse, como hemos visto, a partir de múltiples cámaras (par estéreo, matriz de cámaras) o con una sola. En este último caso, existen varias alternativas para obtener imágenes enfocadas en diferentes profundidades:

- Hacer capturas de la misma escena con la cámara siempre en la misma posición, pero cambiando levemente el plano de enfoque.
- Hacer capturas con la cámara siempre en la misma posición, pero variando por pasos la distancia de la escena (cuando es posible, por ejemplo, montar un objeto sobre una platina con control de movimiento), con los parámetros de la cámara fijos.
- Hacer una captura con una cámara plenóptica y obtener imágenes reenfocadas por paso, cambiando los parámetros correspondientes al algoritmo de reenfoque digital utilizado.

Como hemos visto, el desenfoque está asociado al borrono y lo enfocado, a mayor nivel de detalle. Un recurso habitual en fotografía convencional es utilizar aperturas grandes (número f pequeño) para hacer foco en una profundidad específica bien acotada (donde se desea atraer la atención) mientras el resto de las profundidades queda fuera de foco, observadas como borrosas. Decimos en este caso que la profundidad de campo es pequeña, es decir, el rango de profundidades donde se hace foco es pequeño. La ventaja de un número f pequeño es la gran cantidad de luz que entra al cuerpo de la cámara, puesto que su diafragma está abierto al máximo. Basados en estas propiedades, una técnica ampliamente utilizada para obtener imágenes de gran profundidad de foco y buena iluminación (por lo tanto, buena relación señal-ruido) es la denominada Apilamiento de Enfoque (*focus stacking*) que requiere la captura de varias fotos con f pequeño, enfocadas en diferentes profundidades. Luego se asigna a las regiones de cada una de las imágenes un peso, según el grado de enfoque y se las compone en una imagen final, donde prevalecen los mayores pesos correspondientes a las regiones en foco de cada una de las imágenes originales. El resultado es una imagen de gran profundidad de foco, es decir, enfocada en toda la profundidad de la escena.

La técnica *focus stacking* requiere de estrategias para detectar las regiones en foco y diferenciarlas de las desenfocadas. En principio, cualquier filtro que resalte bordes puede aportar información de las regiones de alto detalle o enfocadas. Pero las regiones de la imagen con textura o rugosidad pueden complicar el proceso dado que esta información de alta frecuencia es detectada como bordes y los algoritmos fallan [74].

El *focus stacking* tiene como propósito obtener imágenes de foco extendido. Otra tarea diferente al propósito de la técnica anterior es la extracción de formas de una escena, es decir, a partir de una o varias imágenes, construir un mapa de profundidades (*depth map*).

4.2. Mapa de profundidades y malla 3D

El mapa de profundidades (MP) es una imagen que contiene información de las distancias relativas a la cámara que tienen los objetos de la escena. Dicha información puede representarse como una imagen en escala de grises o cualquier otro mapa de color, donde cada nivel representa las distancias de los objetos al objetivo de la cámara. Habitualmente se asigna el blanco a las regiones más alejadas a la cámara y el negro a las más cercanas, aunque puede utilizarse la paleta

de colores que permita la mejor interpretación de las distancias relativas. Otra opción de visualización es en forma superficie 3D (*surf*) o malla 3D (*mesh*), que además de contar con información de distancias según la paleta de colores antes mencionada, representa dichas distancias como valores del eje z en un sistema coordenado xyz . Esta representación es la más rica en información tridimensional, pues habitualmente las interfaces gráficas de los softwares permiten su exploración desde diferentes ángulos cambiando los valores de elevación y azimuts. Pero hay que ser cuidadosos con aquellos valores en el MP original que por algún motivo se van de rango o fallan. Estos casos se visualizan en el *mesh* como agujeros o espigas que pueden ser muy molestos o, si son numerosos, confundir al usuario. Estos inconvenientes no se ven tan graves cuando se visualiza el MP como imagen.

Los MP son las representaciones de salida de los algoritmos de estimación de formas 3D, los cuales pueden generarse como hemos visto, a partir de las combinaciones de una única cámara, una sola imagen, varias cámaras y/o varias imágenes. Estas estrategias dan lugar a numerosos algoritmos, tales como estereografía, estructuras a partir de movimiento (*Structure from Motion*), iluminación estructurada (*photometric stereo base don controlled light*), formas a partir de las sombras (*Shape from shading*), formas a partir de las texturas (*Shape from texture*), formas a partir de siluetas (*Shape from silhouettes*), formas a partir del enfoque (*shape from focus, SfF*), formas a partir del desenfoque (*Shape from defocus, SfD*) y borrono por movimiento (*Motion blur*) [75].

De las técnicas anteriores, nos interesa aquellas que puedan aplicarse a fotografía plenóptica. Tal es el caso de SfF y SfD, que requieren pilas focales. Si bien existen algunas diferencias (por ejemplo, en el caso de SfD pueden cambiarse los parámetros de la cámara durante la conformación de la pila, mientras que para SfF esto no está permitido [76]) ambas técnicas sacan provecho del mismo efecto: cuando se coloca un objeto de la escena dentro de la profundidad de foco, los objetos que están fuera de ella aparecen borrosos y la cantidad de borrono se incrementa con la distancia relativa.

Para el caso SfF, la pila focal es obtenida variando la distancia entre la cámara y los objetos de la escena. Si la cámara fuera ideal (estenopeica), los puntos de la escena se proyectarían perfectamente como puntos en la imagen. Pero para cámaras reales y debido a la necesidad de incrementar la cantidad de luz que llega al sensor de imagen, estos sistemas usan lentes de apertura finita lo que resulta en imágenes de limitada profundidad de foco. En este caso, la dependencia entre el desenfoque y la profundidad requiere otros tratamientos además de la óptica geométrica [77].

Nayar [78], [79] obtiene una pila focal dejando la cámara fija y desplazando, por pasos, el objeto cuyo mapa de profundidad interesa obtener. Debido a que el desenfoque es un proceso de filtrado pasa-bajo, el operador de medición de enfoque debe ser un filtro pasa-altos. Nayar propone utilizar el Laplaciano, un filtro convolutivo de derivada segunda. Pero cuando la imagen presenta textura, este filtro puede comportarse de manera inestable dado lo aleatorio de las cancelaciones de signos opuestos según la derivada segunda para las direcciones en x e y . Entonces define el Laplaciano Modificado (ML), que sencillamente suma los módulos de las derivadas segunda en ambos sentidos, superando este inconveniente. Finalmente, la medición del foco en cada punto de la imagen es la suma de los ML en una pequeña ventana (típicamente de 3×3 o 5×5) alrededor de dicho punto [79]. Esta operación se hace para cada imagen de la pila focal y se obtienen vectores $F(d)$, la medida del foco por SML en función de la distancia d equivalente a la profundidad a la que se obtuvo cada imagen de la pila. La estimación de profundidad de cada

punto de superficie utiliza una distribución gaussiana para modelizar la vecindad al pico de Foco (coincidente con la distancia focal del plano imagen) e interpola elementos del vector $F(d)$ seleccionando el mayor. La distancia que le corresponde es la distancia que se asignará al mapa de profundidades en esa posición de la imagen. Remitimos al lector a una explicación detallada en las referencias a S. K. Nayar.

Otra propuesta para construir un mapa de profundidades es la de Favaro [80], [81], [82] que se basa en la técnica SfD. La pila se construye tomando fotografías de la escena mientras se cambia la posición relativa de las lentes del objetivo con respecto al sensor de imagen. Como hemos visto, hay una relación entre las distancias y el desenfoque y se puede estimar la geometría de una escena midiendo la cantidad de desenfoque en una imagen. Pero otra vez, se necesita más de una imagen desenfocada para resolver ambigüedades. La tarea es extraer la información de las formas haciendo el *deblurring* (la acción opuesta a difuminar o borronear) de las imágenes desenfocadas. Favaro propone varias alternativas que pueden consultarse en las referencias, en general sus trabajos apuntan a resolver la deconvolución ciega mediante aproximaciones geométricas.

Un enfoque diferente a los anteriores es el propuesto por Tao y colaboradores [83], que proponen aprovechar las dos cualidades que posee la fotografía plenóptica a partir de una única captura: reenfoque y perspectiva. Hacen uso combinado del desenfoque y la correspondencia para estimar la profundidad, aprovechando lo mejor de cada característica. Las técnicas de SfF trabajan bien en presencia de texturas y ruido; la técnica de correspondencia es robusta en presencia de puntos brillantes. Entonces Tao y colaboradores trabajan con SfF por un lado y correspondencia por el otro, y combinan los resultados con campos aleatorios de Markov (MRF). Otra propuesta que los diferencia del resto es su trabajo con la representación epipolar (EPI) de la radiancia. EPI es quizás la visualización más abstracta del LF, pero contiene información muy interesante sobre las distancias de los objetos a la cámara. Recordemos el diagrama de espacio de los rayos $x-u$ estudiado en el Capítulo 2, donde los puntos de la escena ubicados en el plano focal se mapean como rectas verticales, mientras que los puntos por delante (cerca de la cámara) y por detrás (lejos de la cámara) se mapean como rectas inclinadas. El diagrama EPI reordena los píxeles de la radiancia aplanada de forma que para un $LF(x,y,u,v)$, se representa una matriz donde x y u se mantienen fijos mientras que y y v varían (o viceversa). En las “tiras $y-v$ ” se observa un patrón de líneas en diferentes ángulos. La profundidad de los objetos en la escena puede estimarse desde la pendiente de dichas líneas del diagrama EPI. En la Figura 4.1 se observa un ejemplo de EPIs que corresponden a las posiciones en la imagen por donde pasan las rectas amarilla y roja.

Tao calcula por un lado el DfF integrando el EPI a lo largo de la dimensión angular u y aplicando a este resultado, un operador Laplaciano. Por otro lado, al EPI le realiza el análisis de correspondencia considerando la varianza angular para cada pixel en x . En ambos análisis trabaja con entornos alrededor de cada pixel (9x9) para dar más robustez al proceso. A continuación, debe analizar las respuestas con un estimador de confianza y finalmente combina los resultados con MRF.

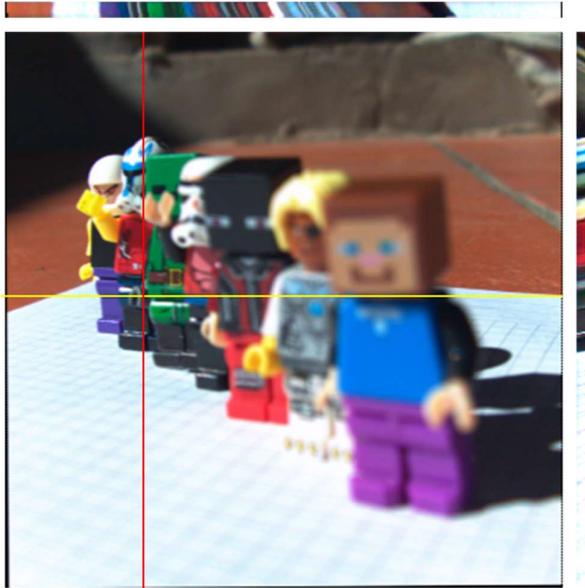


Figura 4.1. Representación de EPI para x y u fijos (en este ejemplo, $x=180$, $u=5$), observada como la tira horizontal de arriba. La tira vertical de la derecha representa las variaciones en x y u para $y=95$ y $v=5$ para este caso. Vemos en ambas tiras un patrón de rectas cuya inclinación está relacionada con las profundidades en la escena.

En la Figura 4.2 observamos un esquema del flujo de trabajo del algoritmo de Tao y colaboradores. Estos investigadores han hecho de dominio público el código de su algoritmo, disponible en [84].

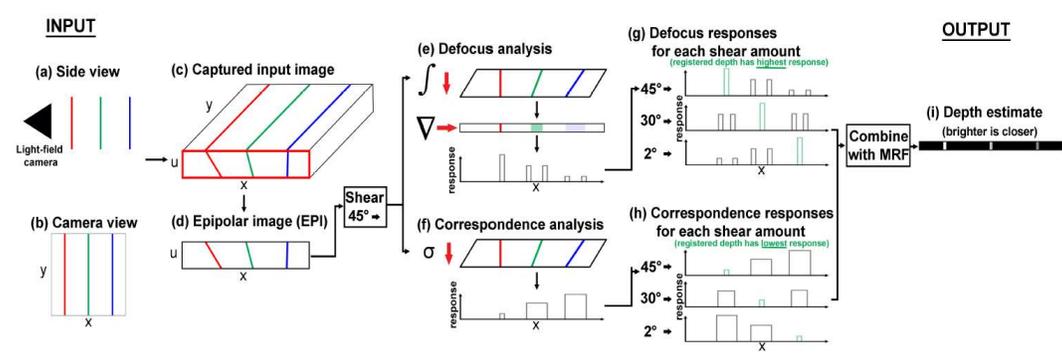


Figura 4.2. Flujo de trabajo del algoritmo combinación de análisis de desenfoque y correspondencia, de Tao y colaboradores [84].

En la Figura 4.3 observamos algunos de los resultados obtenidos en [83]. Como hemos comentado, en la imagen del MP a partir del desenfoque se observan parches de tonos planos (claros para objetos cercanos a la cámara, oscuros para los más lejanos). Donde en la imagen original hay zonas muy oscuras o muy brillantes, el MP por desenfoque produce ambigüedades. Por otra parte, la imagen del MP por correspondencia se observa con mayor nivel de detalle, pero la técnica también produce inconsistencias en zonas muy ruidosas o texturadas.

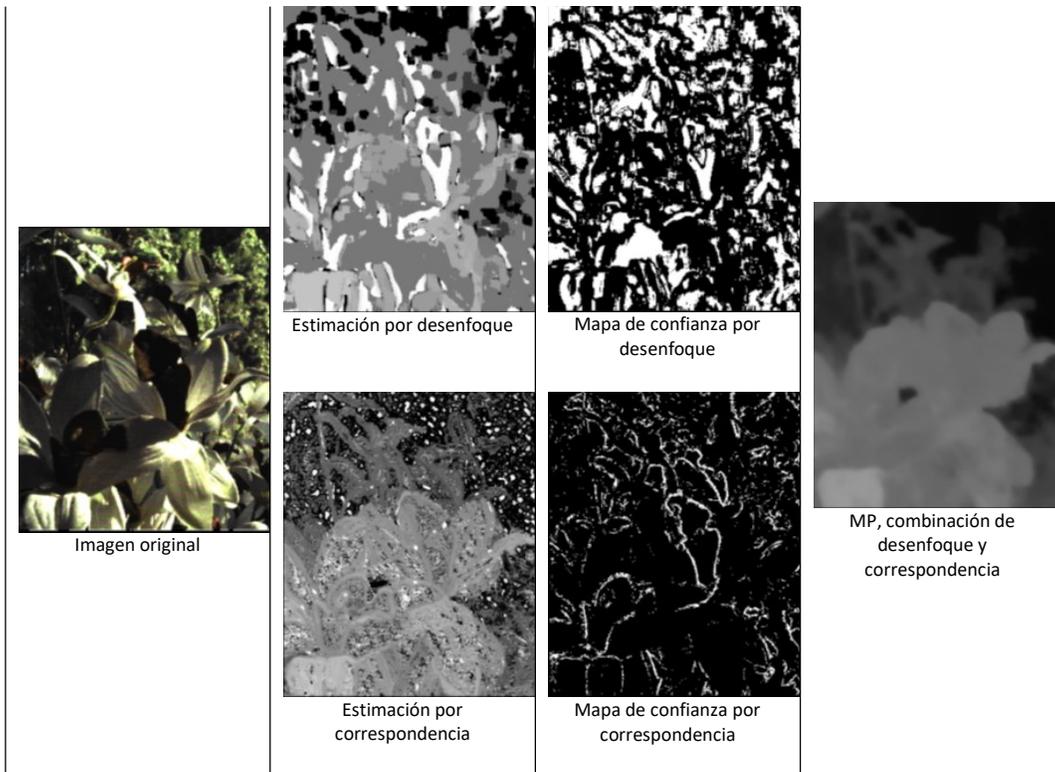


Figura 4.3. Resultados intermedios y final del mapa de profundidad, combinación de desenfocado y correspondencia. Adaptado de [83].

Luego de realizar algunas adaptaciones al código de Tao, ensayamos nuestras imágenes (adquiridas con la Lytro F01), para evaluar su performance. La Figura 4.4 muestra los resultados.

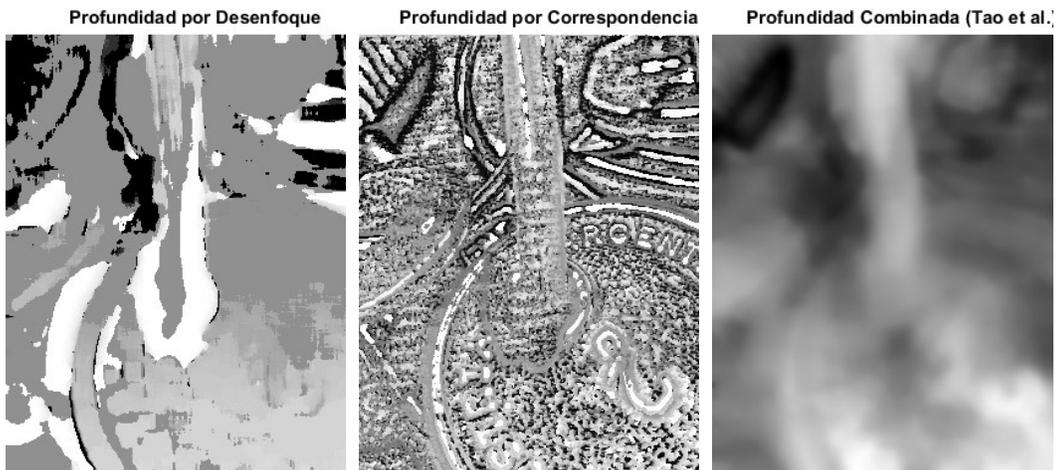


Figura 4.4. Imágenes de mapas de profundidades obtenidas con algoritmo de Tao. El resultado final es la combinación de profundidad por desenfocado y correspondencia. Captura hecha con Lytro F01.

En la Figura 4.5 presentamos la imagen original y el mapa de profundidades generado por el software *Lytro Deskop*, el gestor de imágenes provisto con la cámara Lytro. Dicho programa

procesa la imagen que se descarga de la cámara y permite el reenfoque digital y visualización en perspectiva. Asimismo, provee los mapas de profundidad de las capturas. No hemos encontrado documentación sobre la estrategia que utiliza *Lytro Desktop* para crear los mapas de profundidad. Estimamos que, dada la similitud con los resultados obtenidos en [83], se basa en Desenfoco. En la Figura 4.5 observamos que, más allá del efecto de suavizado que introduce el método de Tao et al., hay coincidencia con las profundidades calculadas por Lytro. Cabe notar que el mapa de Lytro confunde, para este ejemplo, las profundidades donde hay brillo, como observamos en la llave del frente, cuyo mapa marca en blanco las zonas de brillo atribuyéndolas a posiciones en el fondo (blanco, más profundo; negro, más cercano).

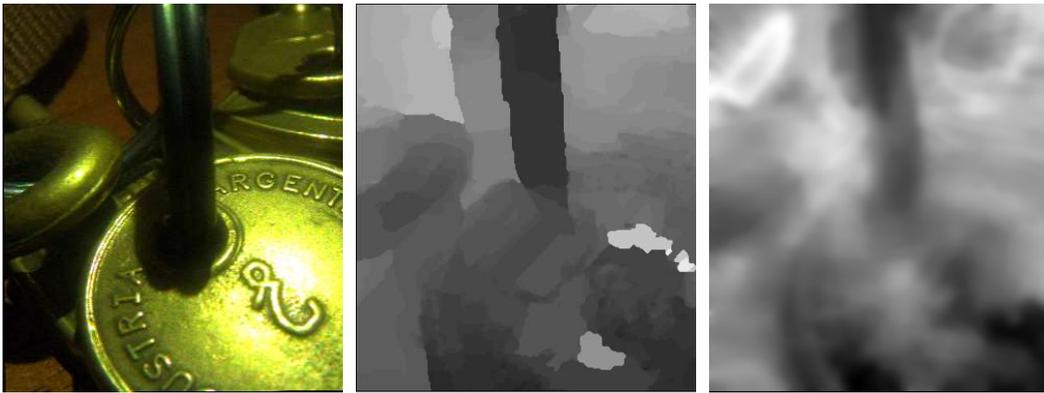


Figura 4.5. Imagen original. Su mapa de profundidad generado por *Lytro Desktop*. Mapa de Profundidad combinado Desenfoco-Correspondencia de Tao et al.

Contamos con un fantoma de escalera, que ensayaremos en el Capítulo 5. Obtuvimos una captura con Lytro F01 y obtenemos su mapa de profundidades con ambos métodos, el de *Lytro Desktop* y el combinado Desenfoco-Correspondencia propuesto Tao. En las Figuras 4.6 y 4.7 observamos los resultados.

En la Figura 4.8 presentamos, a los efectos de hacer más sencilla la comparación, una vista de malla 3D para cada estrategia. El mapa de *Lytro Desktop* presenta mejor resolución en profundidad, pero introduce ruido que se visualiza como espigas en la malla. Este software elige la imagen refocalizada a una determinada profundidad haciendo click en alguna posición del cuadro. Dicha posición corresponde a una determinada profundidad en el mapa de profundidades y extrae de la pila focal, la imagen refocalizada en aquella profundidad. Si justo se elige una posición coincidente con alguna de las espigas del mapa, esto conduce a un error. Por ejemplo, si hacemos click sobre alguna de las dos grandes espigas que aparecen en la Figura 4.8 (arriba), la imagen que seleccionará *Lytro Desktop* corresponderá a una refocalización en lo profundo de la escena, todo lo contrario a lo esperado.

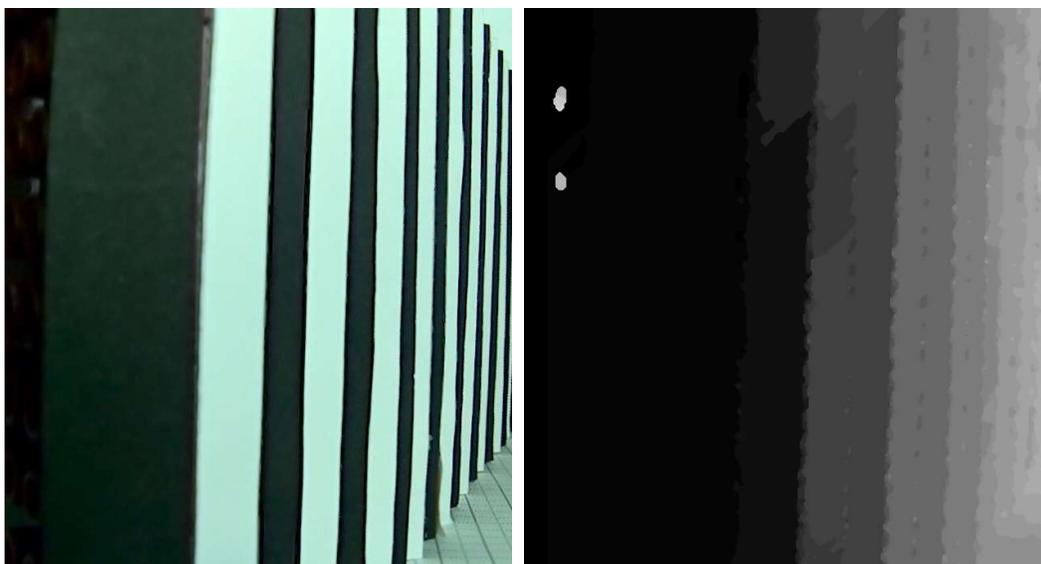


Figura 4.6. Imagen original de foco extendido (enfocada en todas las profundidades) y su mapa, generado por *Lytro Desktop*.

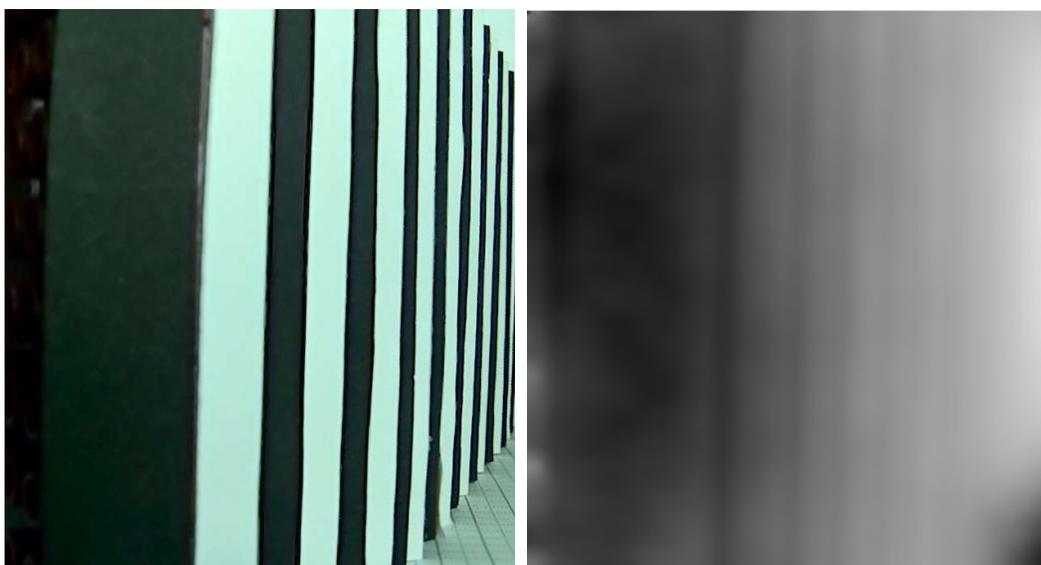


Figura 4.7. Imagen original de foco extendido (enfocada en todas las profundidades) y su mapa, generado por algoritmo de Tao et al.

El mapa generado por el algoritmo de Tao et al tiene un aspecto suavizado, lo que hace más difuso el salto de un escalón a otro. Esto desde el punto de vista de medición de profundidad con precisión es negativo (peor resolución en profundidad). Pero la ausencia de espigas en el mapa disminuye los errores mencionados en el párrafo anterior, siendo por lo tanto una mejor opción si se lo utiliza como lo hace *Lytro Desktop*. Sin embargo, cabe acotar que el mapa de Tao presenta errores en las regiones correspondientes al piso (papel cuadriculado en la imagen original) y en los extremos de las profundidades de la escena (al frente y al fondo). *Lytro Desktop* reproduce acertadamente la profundidad de dichas regiones.

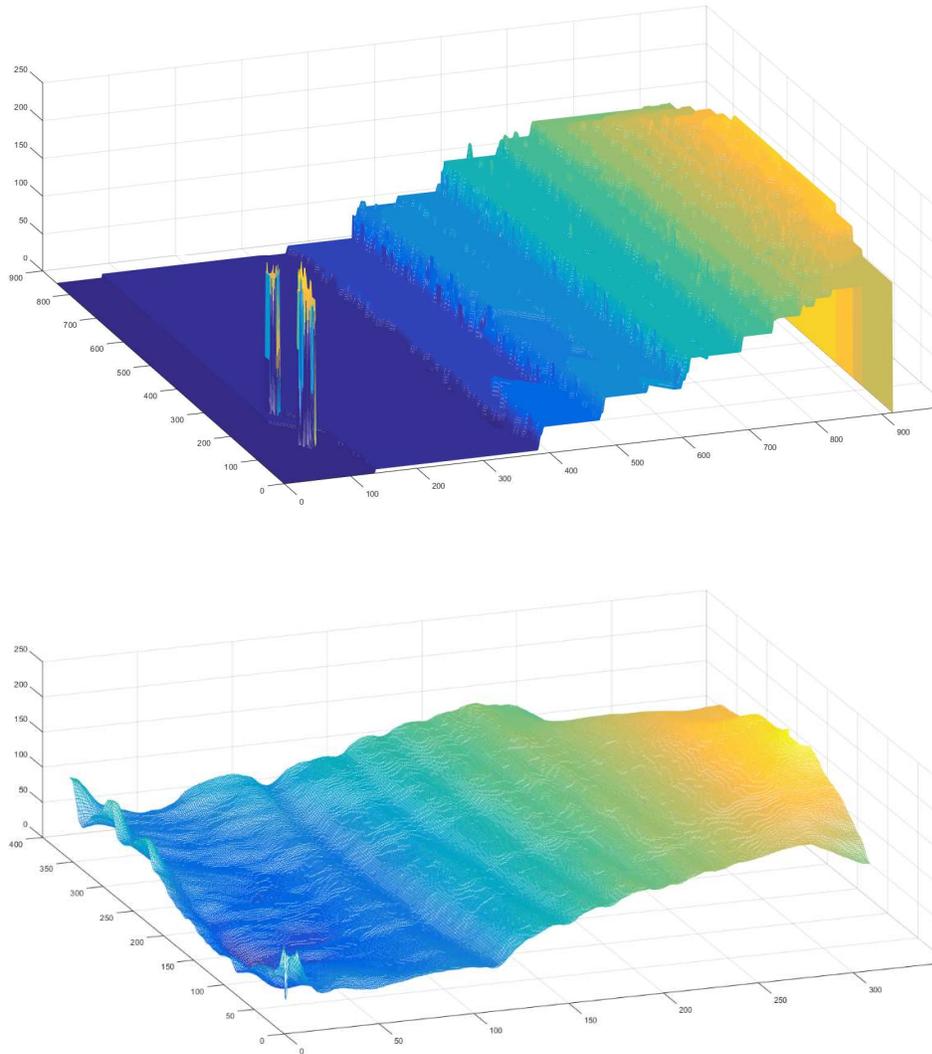


Figura 4.8. Arriba: mapa de profundidades generado por *Lytro Desktop*. Abajo: mapa de profundidades generado por algoritmo de Tao et al.

Finalmente y a los efectos de mostrar el potencial de la fotografía plenóptica para la representación de objetos pequeños, presentamos un resultado obtenido con la Lytro F01. Recordemos que esta cámara es la más básica de las disponibles en el mercado.

Disponemos de la óptica correspondiente al objetivo de otra Lytro F01 que hemos desmantelado a los efectos de esta prueba. Colocamos dicha lente objetivo en posición “nose to nose” (“nariz con nariz”) con el de la cámara plenóptica y obtuvimos imágenes de un pequeño gancho que mostramos en la Figura 4.9. El diámetro del alambre de dicho gancho es de 0.8 mm. Este resultado observado en la Figura 4.10 nos permite adentrarnos en la posibilidad de obtener imágenes médicas como lo comentamos en el primer capítulo. Imágenes de la topografía de lunares o de la membrana timpánica son aplicaciones concretas de interés, de acuerdo con consultas a profesionales médicos.



Figura 4.9. Imagen original de un gancho, colocado sobre una púa de guitarra y al lado de una moneda, a los efectos de su comparación en tamaño.

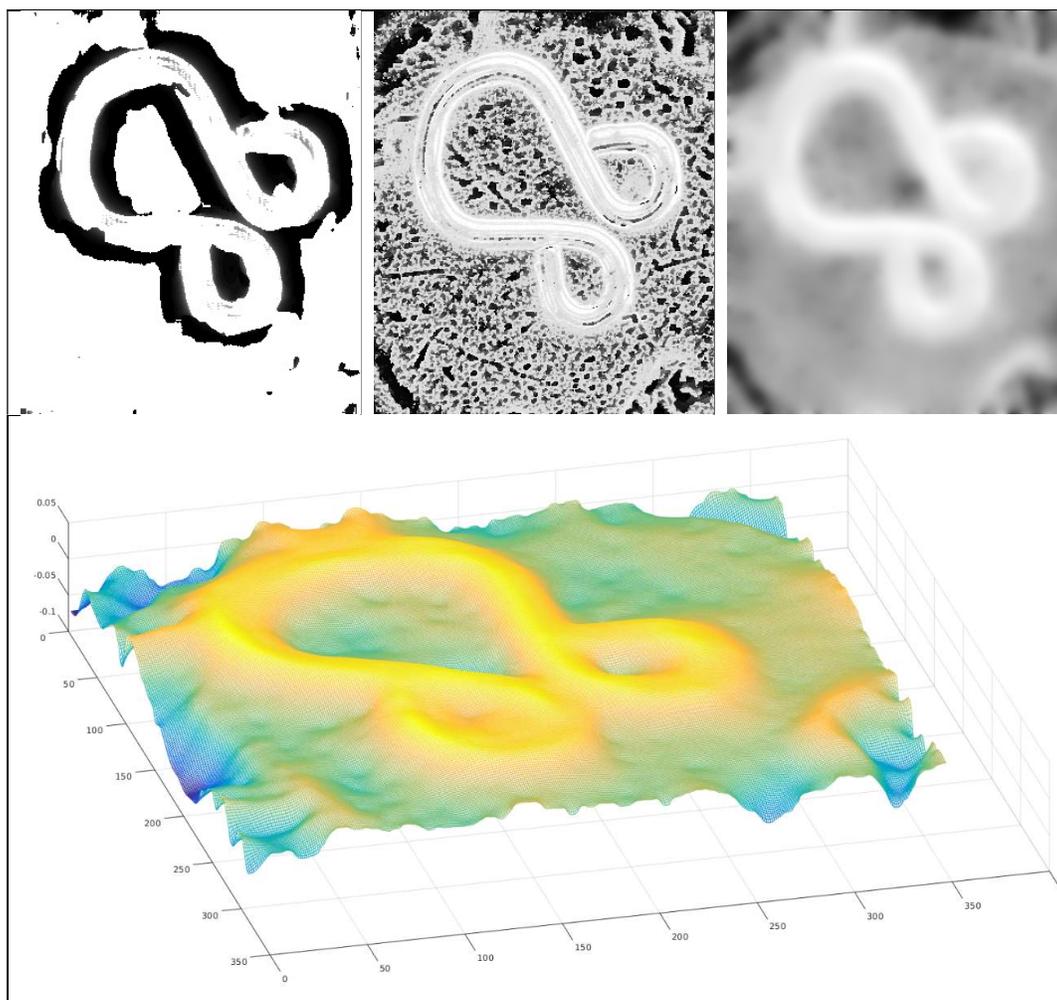


Figura 4.10. Arriba: de izquierda a derecha, mapa de desenfoco, correspondencia y combinado obtenidos con el algoritmo de Tao. Abajo: mapa de profundidades combinado, en formato de malla 3D.

Capítulo 5

Evaluación de Resultados

En este Capítulo mostramos las pruebas hechas con los algoritmos estudiados, seleccionados en base a algunos criterios que se detallarán más adelante. Cabe destacar que para el caso de la arquitectura Plenoptic 2.0, no contamos con imágenes propias por lo que utilizamos capturas de uso libre puestas a disposición en la web por el Dr. Georgiev [40]. Evaluamos los resultados de acuerdo con algunos criterios que consideramos importantes, tales como influencia de los parámetros de los algoritmos en los tiempos de cómputo, efecto del tamaño de las imágenes a procesar en dichos tiempos, preservación de las características originales de la captura (brillo, balance de color, contraste, nivel de detalle). Finalmente reunimos los resultados en un cuadro comparativo en base a los criterios antes mencionados.

5.1. Diseño del experimento

Los datos consisten en un conjunto de capturas con cámara Lytro® F01 descrita en el Capítulo 3. Como escenario para la toma, ordenamos siete muñecos Lego® sobre una hoja cuadriculada, a los fines de medir su separación. Colocamos un patrón de barras detrás de los Lego e hicimos tomas a diferentes distancias de dicho patrón y diferentes espaciamentos entre muñecos. Utilizamos iluminación natural proveniente del cielo, en horas del mediodía, con el escenario a cubierto bajo una galería. La razón de haber utilizado juguetes Lego fue aprovechar los detalles pintados sobre el plástico para observarlos cuando se hace foco sobre ellos, a la vez que pueden ser fácilmente manipulados.

Dada la excesiva cantidad de variables a combinar, decidimos hacer las tomas sin zoom a distancia focal constante y apertura constante, en modo automático para que la cámara eligiera las características del ISO y tiempo de apertura de acuerdo a la iluminación de la escena. La Lytro posee pantalla táctil sobre la que se puede seleccionar el lugar de la escena donde se desea que la cámara enfoque (foco óptico). Optamos por enfocar todas las tomas en el centro de la escena, dadas las recomendaciones brindadas en el sitio [85] para las capturas con Lytro F01. En una de las series de imágenes, ubicamos un primer muñeco a 45 cm del patrón de barras. Desde esta posición desplegamos el resto: a 46 cm el segundo, a 47 cm el tercero y así sucesivamente hasta el último. A su vez separamos los muñecos lateralmente 1.5 cm para que no se ocluyeran entre sí ante la cámara, que posicionamos a 70 cm del patrón. También obtuvimos otras series de imágenes combinando separaciones entre muñecos de 2, 3 y 4 cm y manteniendo la separación lateral de 1.5 cm para evitar oclusiones, todas estas distancias controladas sobre la hoja cuadriculada. Asimismo, hicimos capturas repitiendo las posiciones y separaciones de los muñecos, pero con la cámara colocada a 80 cm del patrón de barras. Mostramos algunas fotos de la composición de la escena en la Figura 5.1.

Otra escena fue montada con un patrón en escalera, con barras blancas y negras. Los escalones tienen 1 cm de altura y 1 cm de profundidad. Se tomaron fotos a diferentes distancias y diferentes angulaciones para evaluar la discriminación en profundidad que proveen los algoritmos. La Figura 5.2 muestra detalles de la composición de la escena.

A los efectos de comprender algunas limitaciones en los resultados que veremos más adelante, es importante en este momento describir las limitaciones de la cámara y demás dispositivos utilizados. En el Capítulo 3 comentamos que la cámara Lytro F01 permite la recuperación de una imagen reenfocada de sólo 378 x 378 píxeles de 8 bits por canal de color, resolución espacial muy baja que afecta la capacidad de reproducir las altas frecuencias espaciales asociadas con las regiones en foco. Por otra parte, el fantoma de escalera descrito previamente presenta pequeños defectos de fabricación en los bordes de escalones, dado que se basa en piezas de encastre con irregularidades en su ajuste, a la vez que las bandas negras y blancas son de cartón que en algunos escalones presentan deformaciones por pliegues o problemas del adhesivo. Tenidas en cuenta todas estas limitaciones y a pesar de ellas, consideramos que los resultados evidencian diferencias que nos permitirían comparar y evaluar la performance de los algoritmos propuestos. Dado que la F01 es la versión más básica de las cámaras plenópticas Lytro, el fabricante brinda recomendaciones en [85] y concluye diciendo: “La mejor manera de aprender a tomar fotografías de un campo de luz es practicando tomarlas” (*“The very best way to learn how to take light field pictures is to practice taking them”*). En este Capítulo presentamos resultados de algunas de las numerosas series de imágenes capturadas.

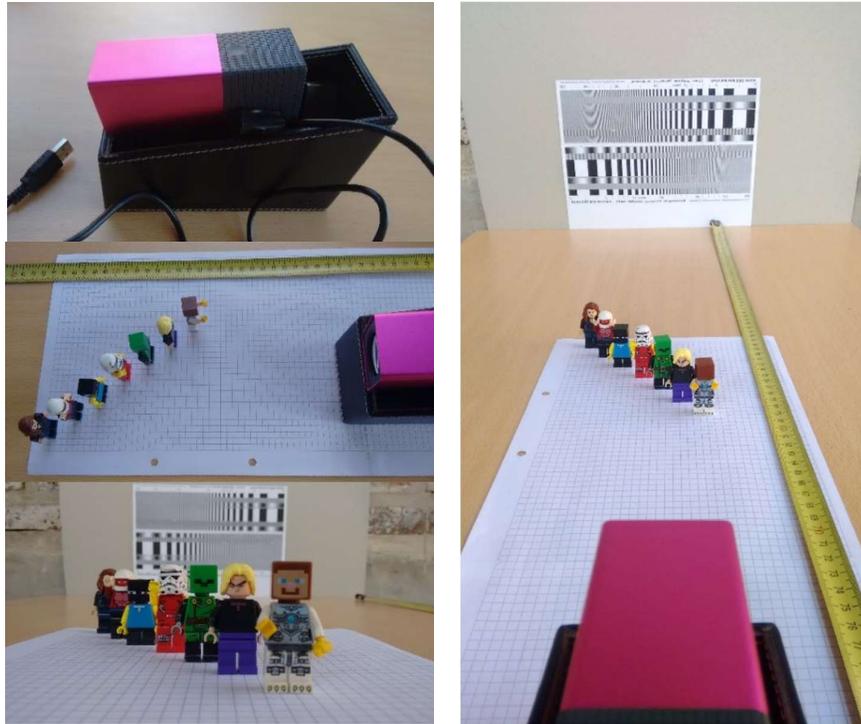


Figura 5.1. Algunas fotografías de la escena preparada para obtener el set de datos. El tamaño de las cuadrículas de la hoja es de 5mm x 5mm.



Figura 5.2. Montaje de la escena con fantoma escalera. Son 31 peldaños de 1cm de frente y 1 cm de altura.

El equipo de computación empleado para correr los algoritmos fue una laptop con las siguientes capacidades:

Processor: Intel® Core™ i7-6700HQ CPU @ 2.60GHz 2.59GHz

Installed Memory (RAM): 16.0 GB (15.8 GB usable)

System type: 64-bit Operating System, x64-based processor.

5.2. Criterios de selección de los algoritmos

En el Capítulo 3 presentamos varios algoritmos, a los que podemos dividir de acuerdo a su propósito en 3 grupos:

- a) **Codificación de la radiancia:** `read_LFfor` y `read_LFvect`
- b) **Modificación de la Perspectiva:** para radiancias de arquitectura Plenoptic 1.0, directamente con la función `squeeze`; para radiancias de arquitectura Plenoptic 2.0, parámetros `offsetX,offsetY` de `BasicFullRender`
- c) **Refocalización:**
 - para radiancias de arquitectura Plenoptic 1.0:
 - en dominio espacial: `desplaz_suma`, `RefocusRadiancia`, `computesubsetFDFST` (Transformada Rápida Discreta del Focal Stack)
 - en dominio frecuencial: `RefocusEspectro`, `LFBuild4DFreqPlane`, `LFBuild4DFreqHyperfan`
 - para radiancias de arquitectura Plenoptic 2.0: `CorteBloques`, `BasicFullRender` y `renderblock_blend`

Codificación de la radiancia

Del grupo a) indicamos en el inciso 3.1.2 del Capítulo 3, que la vectorización de la codificación incrementa la velocidad del proceso en un factor de aproximadamente 1000 comparada con la codificación basada en bucles, por lo que `read_LFvect` es el método seleccionado. A modo de ejemplo, para codificar una imagen cruda de 4624 x 4624 x 3 uint8, 4.414MB, $N_{xy}=289$ y $N_{uv}=16$ se requieren 0.16 segundos.

Modificación de la Perspectiva

Una vez obtenido el LF 4D, la visualización de perspectivas correspondientes al grupo b) no presenta mayores inconvenientes. Como comentamos en el Capítulo 3, inciso 3.2.1, la resolución angular en la arquitectura Plenoptic 1.0 depende exclusivamente del tamaño de las microimágenes, es decir, la cantidad de píxeles debajo de cada microlente. Por ejemplo, la cámara Lytro F01 posee 11x11 píxeles debajo de cada lenticula, siendo en consecuencia la cantidad de imágenes de subapertura igual a 121. Por otra parte, la visualización de perspectivas para el caso de Plenoptic 2.0 representa un compromiso entre el tamaño del bloque seleccionado para el reenfoque y la resolución angular. En el inciso 3.3.1.3 estudiamos que cuanto mayor es el tamaño del bloque, el plano de foco se ubica más lejano a la cámara. Y dicho bloque según se desplaza en los límites de la microlente, permite la visualización de diferentes perspectivas. Es obvio que cuanto mayor es el tamaño del bloque, más restringido es su movimiento dentro de la lenticula: he aquí la relación de compromiso. Imágenes enfocadas cerca de la cámara (bloques pequeños) tienen mayor resolución angular que imágenes enfocadas lejos (bloques grandes). La visualización en perspectiva para Plenoptic 2.0 es parte del algoritmo de refocalización `BasicFullRender`. Simplemente los bloques de tamaño M para el reenfoque, se desplazan del centro una cantidad `OffsetX,OffsetY` y este desplazamiento es el que marca la perspectiva, por lo que debido a esta razón constructiva del algoritmo, no podemos estudiar la visualización de perspectivas por separado.

El tiempo de cómputo de perspectivas para el caso de Plenoptic 1.0 es insignificante ya que se basa en una función MATLAB vectorial optimizada. Por ejemplo, una imagen capturada con cámara Lytro Illum® de 433x625 lenticulas, cada una con 15x15 pixeles, insume para su visualización en perspectiva 0.29s, promedio de 30 pruebas. En sintaxis de MATLAB se observa el siguiente resultado:

```
>> tic;imshow(squeeze(LFin(:,:,2,7,:)));toc
Elapsed time is 0.289460 seconds.
```

Si comparamos este resultado con una captura con cámara Lytro F01 de 378x379 lenticulas, cada una con 11x11 pixeles, el tiempo que lleva visualizar una perspectiva para este campo de luz es de 0.03s promedio.

En la Figura 5.3 observamos un desplazamiento en perspectiva vertical de un mismo campo de luz, hecho evidente en el cambio de posición de la rama en el fondo de la escena respecto de las hojas en el plano focal.



Figura 5.3. Visualización de 2 perspectivas diferentes, cámara Lytro Illum, LF(433,625,15,15,3). Izq.: selección del pixel (renglón,columna)=(2,7) para cada microlente. Centro: selección del pixel (renglón,columna)=(13,7) para cada microlente. Der.: diferencia de las posiciones de los elementos de la imagen por cambio de perspectiva en comparación por superposición de las dos imágenes anteriores. Imagen de acceso libre en [62].

A los efectos de comparar las vistas en diferentes perspectivas, utilizamos una función de MATLAB denominada `imshowpair` que superpone en la misma imagen las dos que se comparan, presentando en falso color las estructuras que no coinciden en posición.

Refocalización

Para la selección de los algoritmos de refocalización del grupo c), un primer criterio es que su tiempo de procesamiento no se extienda, por ejemplo, más de 5 minutos. Este máximo puede ser razonable para post procesos que no requieran resultado urgente, tal como construir la pila focal de una fotografía cualquiera que será guardada y accedida luego. Para procesos en tiempo real este máximo es inadmisibles y deberá recurrirse a códigos más cercanos a lenguaje de máquina o procesamiento paralelo. Basados en dichas condiciones de tiempo de procesamiento y de acuerdo con lo visto en el Capítulo 3 inciso 3.3.1.5, descartamos el algoritmo Transformada Rápida Discreta del Focal Stack, que para obtener una pila focal de 5 imágenes de 256x256 pixeles cada una, requiere 30 minutos. El resto de los algoritmos de refocalización estudiados satisface el requerimiento de tiempo de proceso propuesto, o sea, menos de 5 minutos.

Dentro del grupo de algoritmos del dominio de la frecuencia en Plenoptic 1.0, `RefocusEspectro` y `LFBuild4DFreqPlane` se basan en el mismo principio de corte del

espectro 4D con un plano. Dado que `LFBuild4DFreqPlane` pertenece al `LFToolbox` de Dansereau y que `RefocusEspectro` fue implementado por nosotros, preferimos someter este último a los ensayos. También nos interesa ensayar `LFBuild4DFreqHyperfan`, que es otra de las propuestas de Dansereau, dada la novedad del foco volumétrico que provee como salida.

Finalmente, para la arquitectura Plenoptic 2.0, para idénticos resultados de salida entre `CorteBloques` y `BasicFullRender` elegimos este último dado su menor tiempo de cómputo. El caso del algoritmo `renderblock_blend` es particular, dado que hace uso de `BasicFullRender` para procesar varias vistas en perspectiva para la misma profundidad de foco, de manera que el tiempo de proceso depende del número de vistas promediadas. Dada esta particularidad, optamos por elegir sólo `BasicFullRender` para evaluar imágenes de cámaras plenópticas de arquitectura Plenoptic 2.0.

Por lo argumentado en los párrafos anteriores, elegimos para los ensayos los algoritmos `desplaz_suma`, `RefocusRadiancia`, `RefocusEspectro`, `LFBuild4DFreqHyperfan` y `BasicFullRender`.

5.3. Resultados experimentales

A continuación, presentamos los ensayos realizados sobre los algoritmos seleccionados, basados en las configuraciones de escenas mostradas en los párrafos anteriores. Proponemos algunas estrategias para hacer posibles comparaciones y medir la performance en cada caso.

5.3.1. Ensayos Algoritmo Desplazamiento Suma (`desplaz_suma`)

Tiempos de cómputo

Los tiempos de cómputo del algoritmo `desplaz_suma` son los mostrados en el Gráfico A, para un LF de LytroF01 de tamaño:

```
>> size(LFin)
ans = 378 379 11 11 3
```

Se promediaron los tiempos de 30 corridas por cada parámetro alfa, para alfa entre 0.6 y 2.4 en pasos de 0.2 (`alfa = 0.6 : 0.2 : 2.4` en sintaxis MATLAB) y como era de esperar, no se registraron cambios significativos en los tiempos de corrida para diferentes alfa, ya que este parámetro sólo afecta la cantidad de pixeles que se desplaza cada imagen de subapertura antes de ser promediadas para obtener la imagen reenfocada. El tiempo promedio para todo alfa es de 2.17888 segundos, para las 121 aperturas de la imagen de prueba. Hay una disminución significativa del tiempo de cómputo cuando alfa es entero y atribuimos esto a cómo computa MATLAB, según trabaja con diferentes clases numéricas.

Ensayamos el algoritmo con imágenes de la misma cantidad de lenticulas, pero diferente cantidad de pixeles por microlente, para observar si el incremento de tiempo es lineal o no. Deberíamos esperar un incremento de tiempo proporcional a la diferencia de subaperturas que se promedian. Contamos con LF de 15x15 pixeles/lenticula (Lytro Illum) y de 20x20 pixeles/lenticula (imagen cruda de acceso libre en [86]). Dado que una de las imágenes está limitada a 114x114 microlentes,

trabajamos con esta cantidad de lenticulas para las tres imágenes de prueba. Mostramos los resultados en el Gráfico B. Como es de esperar, el incremento es lineal con el número de subaperturas a promediar y no depende del parámetro alfa (excepto cuando es entero, donde muestra un menor tiempo de cómputo).

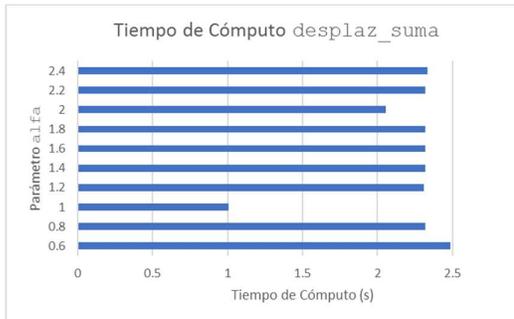


Gráfico A. Tiempos de cómputo para algoritmo Desplazamiento-Suma, con diferentes parámetros alfa.

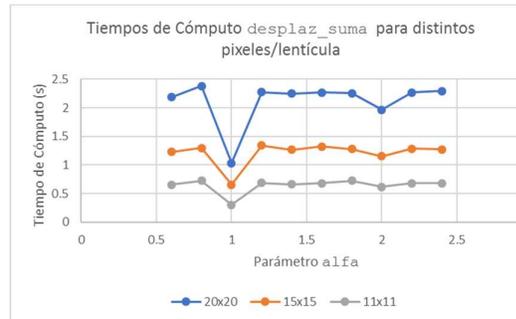


Gráfico B. Tiempos de cómputo para algoritmo Desplazamiento-Suma, imágenes todas de 114x114 microlentes, pero con diferente número de pixeles / lenticula.

Podemos verificar ahora el comportamiento del algoritmo a medida que crece el tamaño de imagen. Lo ensayamos con la misma imagen capturada con cámara Lytro Illum de 433x625 lenticulas, cada una con 15x15 pixeles, de la Figura 3. Partimos de un recorte de 108x156 lenticulas y ensayamos el algoritmo para 216x312, 324x468 y finalmente 433x625, resultados que mostramos en Gráfico C. En el Gráfico D vemos que la relación del tiempo de cómputo con el número de lenticulas de la imagen es lineal.

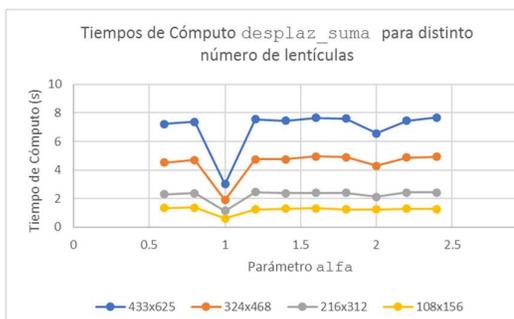


Gráfico C. Tiempos de cómputo para algoritmo Desplazamiento Suma, imágenes todas de 15x15 pixeles / lenticula, pero con diferente número de microlentes.

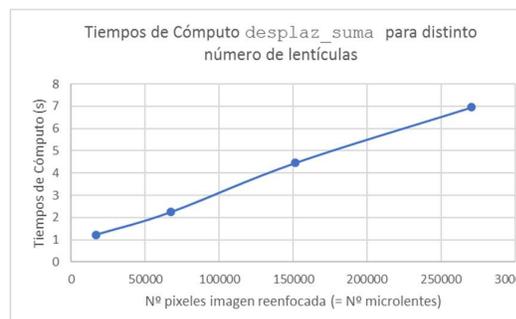


Gráfico D. Tiempo de cómputo promedio para los parámetros alfa ensayados con imágenes de diferente cantidad de lenticulas, con algoritmo Desplazamiento Suma.

Enfoque en profundidad

Nos interesa conocer las regiones enfocadas al aplicar los algoritmos con sus diferentes parámetros. Para determinar el rango de parámetros dentro del cual pueden observarse regiones en foco, acudimos a evaluar perfiles de imagen. En la Figura 5.4 mostramos la comparación entre los perfiles ubicados sobre el patrón de barras en el fondo de la escena Serie 707.mat, correspondientes a la recta roja idéntica para cada imagen, reenfoçadas con alfa de 0.6 y 0.8,

respectivamente. Observamos que para $\alpha=0.8$ el perfil es una onda más cuadrada que para $\alpha=0.6$ y su amplitud pico a pico reproduce mejor el contraste, indicando que está mejor enfocada. Por otra parte, vemos que si las barras fueran negras y blancas, los canales de color debieran verse superpuestos, cosa que no sucede debido a las condiciones de iluminación que no reflejan al blanco como blanco ni asignan al negro el valor teórico de cero. Aún con estas limitaciones, la prueba de los perfiles nos permite definir más objetivamente que el enfoque de fondo comienza con $\alpha=0.8$. De allí en más, a medida que aumenta α , los bordes van quedando definidos en una banda que se parece más a un volumen en foco que a un plano en foco. Después de $\alpha=2$ no observamos cambios apreciables aplicando las mismas pruebas con perfiles. Estos valores experimentales de α están en el rango de los ensayados en [83] con la misma cámara plenóptica.

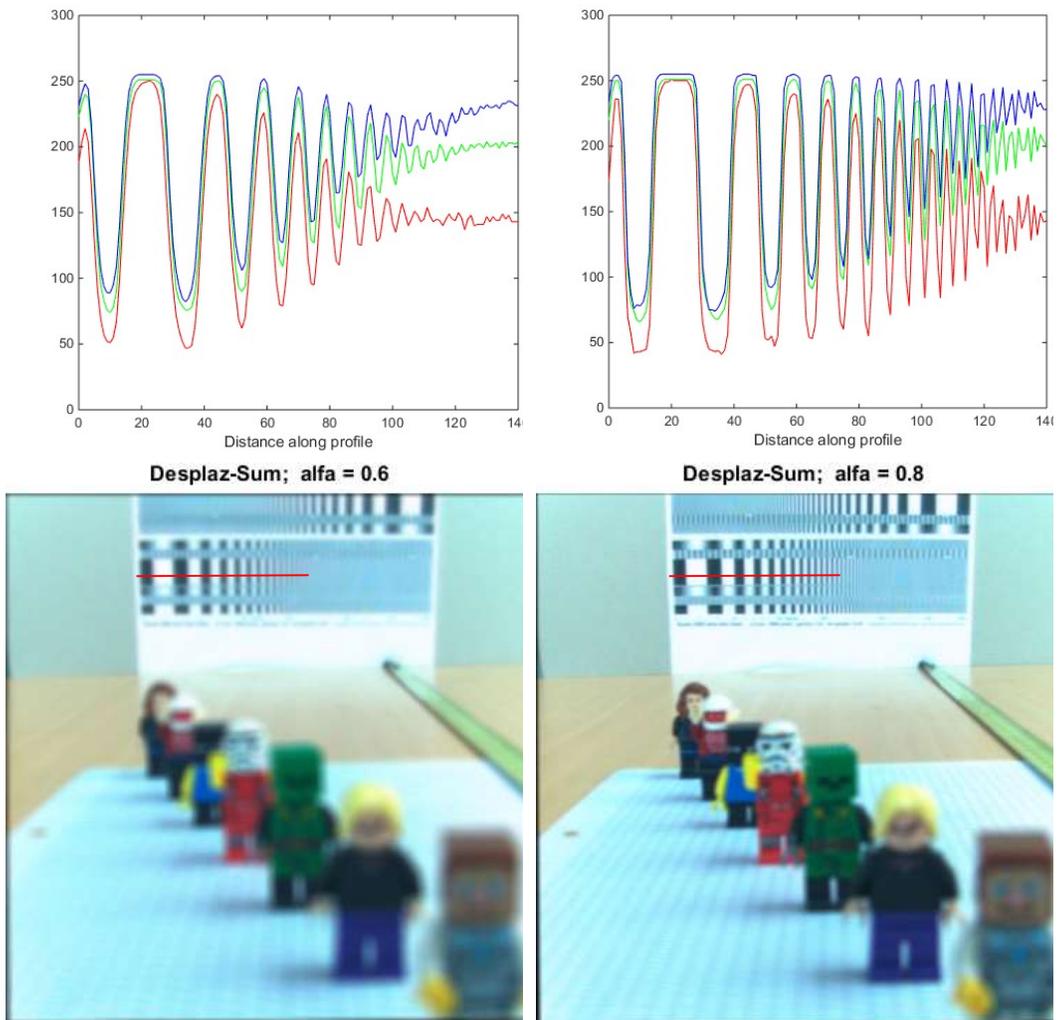


Figura 5.4. Serie 707.mat – Muñecos separados 4cm, distancia cámara-fondo: 70cm. Columna izq.: perfil de imagen de la línea roja del reenfoque por Desplazamiento-Suma para $\alpha=0.6$. Columna der.: perfil de imagen de la línea roja del reenfoque por Desplazamiento-Suma para $\alpha=0.8$. La recta roja se tiende desde el punto (93,58) hasta el punto (233,58) para ambas imágenes.

Una forma rápida de verificar el cambio de plano focal y estimar su posición es realizando los bordes. Sabemos que la imagen en foco presenta buena calidad de bordes mientras que, si está desenfocada, se observa borrosa. Entonces podemos aplicar algún filtro detector de bordes, por ejemplo un Laplaciano, y observar que la región donde los bordes están más definidos corresponde a la zona enfocada. Así como elegimos al Laplaciano, pudo haber sido cualquiera de las numerosas alternativas disponibles, dado que no interesa su eficiencia en términos de la detección en sí misma sino más bien, que permitan comparar cómo cambian los bordes a medida que se enfoca en diferentes posiciones. A los efectos de mejorar la visualización de los bordes por filtrado Laplaciano, en la Figura 5.5 mostramos el complemento del resultado. Observamos que si bien se notan cambios, no es sencilla la comparación entre dos imágenes reenfocadas contiguas.

Otra alternativa para observar las regiones en foco es sustraerle a la imagen reenfocada, la imagen de foco extendido obtenida con el algoritmo `dism_apertura`. La región donde la sustracción es cercana a cero, es la región en foco. La Figura 5.6 muestra los pares Filtrado Laplaciano y Sustracción Enfocada, para un mismo valor de `alfa`. Observamos que la Sustracción Enfocada presenta ruido, ya que la imagen de foco extendido es ruidosa debido a la disminución de apertura. Por eso decimos que la diferencia es “cercana a cero”, es decir, hay mayor coincidencia en las regiones enfocadas. La Figura 5.7 presenta la misma imagen reenfocada de la Figura 5.5 y complemento de la diferencia reenfocada - infocus, con valores de `alfa` entre 0.8 y 2.2.

De manera similar a lo que sucede con el Laplaciano, observamos que para el set de muñecos no es sencillo establecer de manera objetiva la región focal, aunque sí podemos inferir que va desplazándose a medida que cambiamos el parámetro alfa. Con el set de imágenes del fantoma escalera lo que pretendemos es verificar en los escalones donde se hace foco, la calidad de los bordes. El propósito es ser más objetivos en la verificación de los desplazamientos de las regiones en foco a medida que se varían los parámetros de los algoritmos.

En la Figura 5.8 observamos los pares Imagen Reenfocada-Laplaciano para valores de alfa desde 0.9 (foco en el fondo) hasta 1.8 (foco en el frente). El Laplaciano permite ver cómo las rectas oscuras, mejor definidas, se desplazan hacia la cámara a medida que alfa crece. Hacemos crecer alfa en pasos de 0.1 y se observan diferencias, que quizás es difícil distinguir en la Figura 5.8 pero se hacen evidentes cuando se ven una a continuación de otra, en un formato de video del tipo MPEG, por ejemplo. Las distancias entre los bordes de escalón correspondientes al ángulo con que fue hecha la captura, son de 1.4cm y se observan variaciones menores a un escalón.

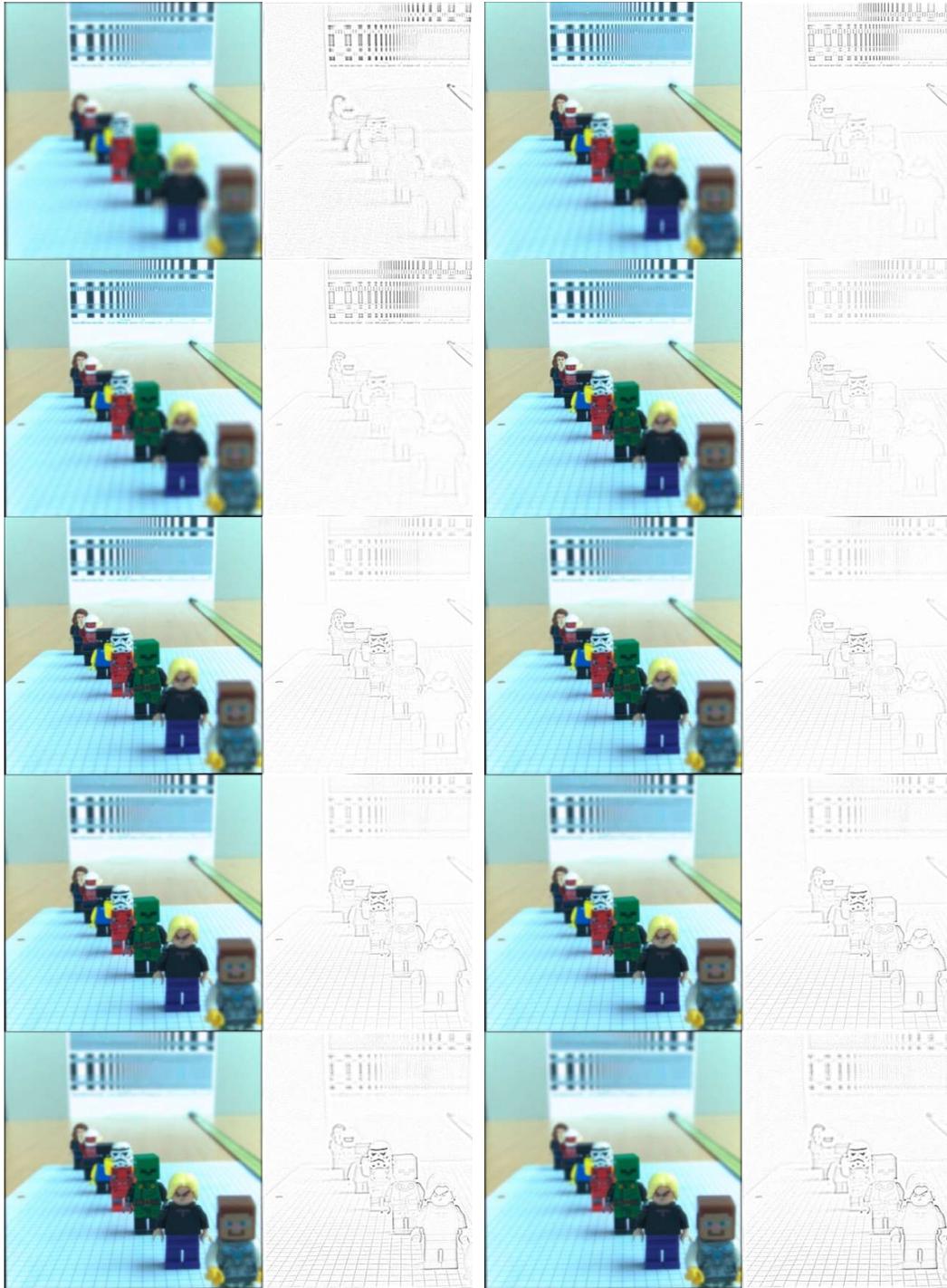


Figura 5.5. Serie 707.mat – Muñecos separados 4cm, distancia cámara-fondo: 70cm. La secuencia se desarrolla por renglones, desde arriba, mostrando imagen reenfocada y su par filtrado Laplaciano, con los siguientes valores de α : 0.6 – 0.8, 1.0 – 1.2, 1.4 – 1.6, 1.8 – 2.0, 2.2 – 2.4.

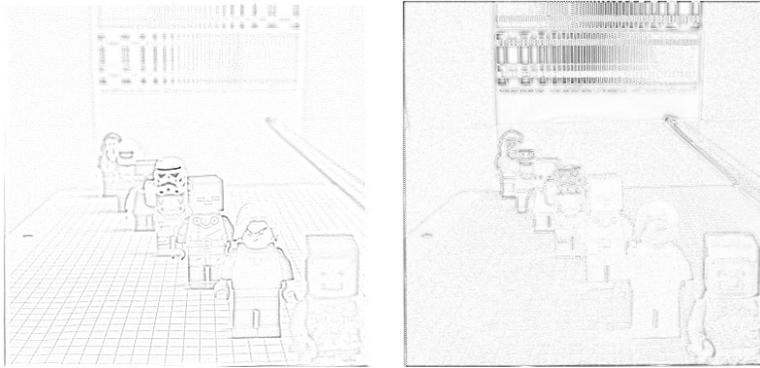


Figura 5.6. Izq.: complemento del laplaciano de la imagen enfocada con algoritmo `desplaz_suma`, con $\alpha=1.8$. Der.: complemento de la imagen diferencia entre la enfocada con algoritmo Desplazamiento Suma ($\alpha=1.8$) e imagen de foco extendido ('infocus') obtenida con `dism_apertura`.

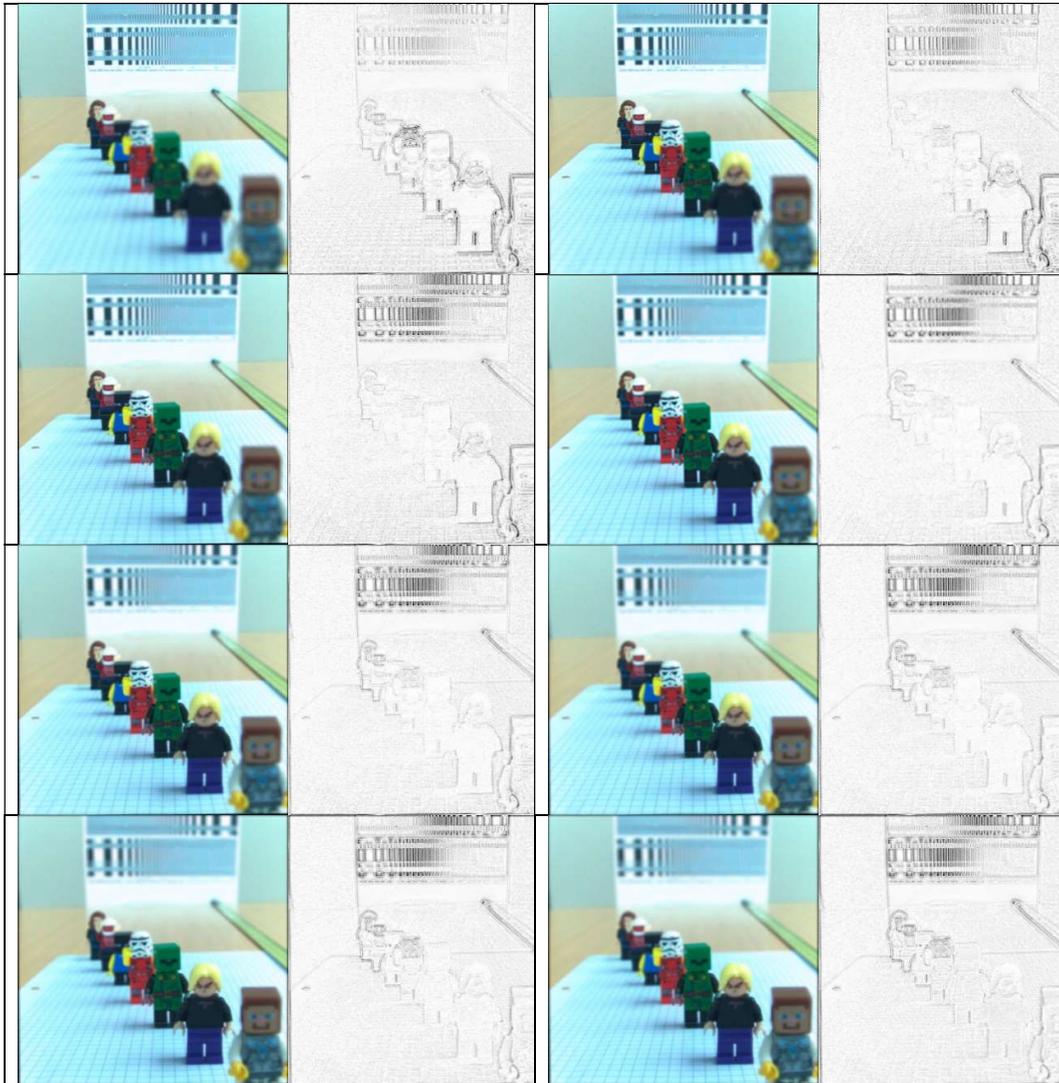


Figura 5.7. Serie 707.mat. La secuencia se desarrolla por renglones, desde arriba, mostrando imagen reenforcada y complemento de la diferencia reenforcada - infocus, con los siguientes valores de α : 0.8 - 1.0, 1.2 - 1.4, 1.6 - 1.8, 2.0 - 2.2.

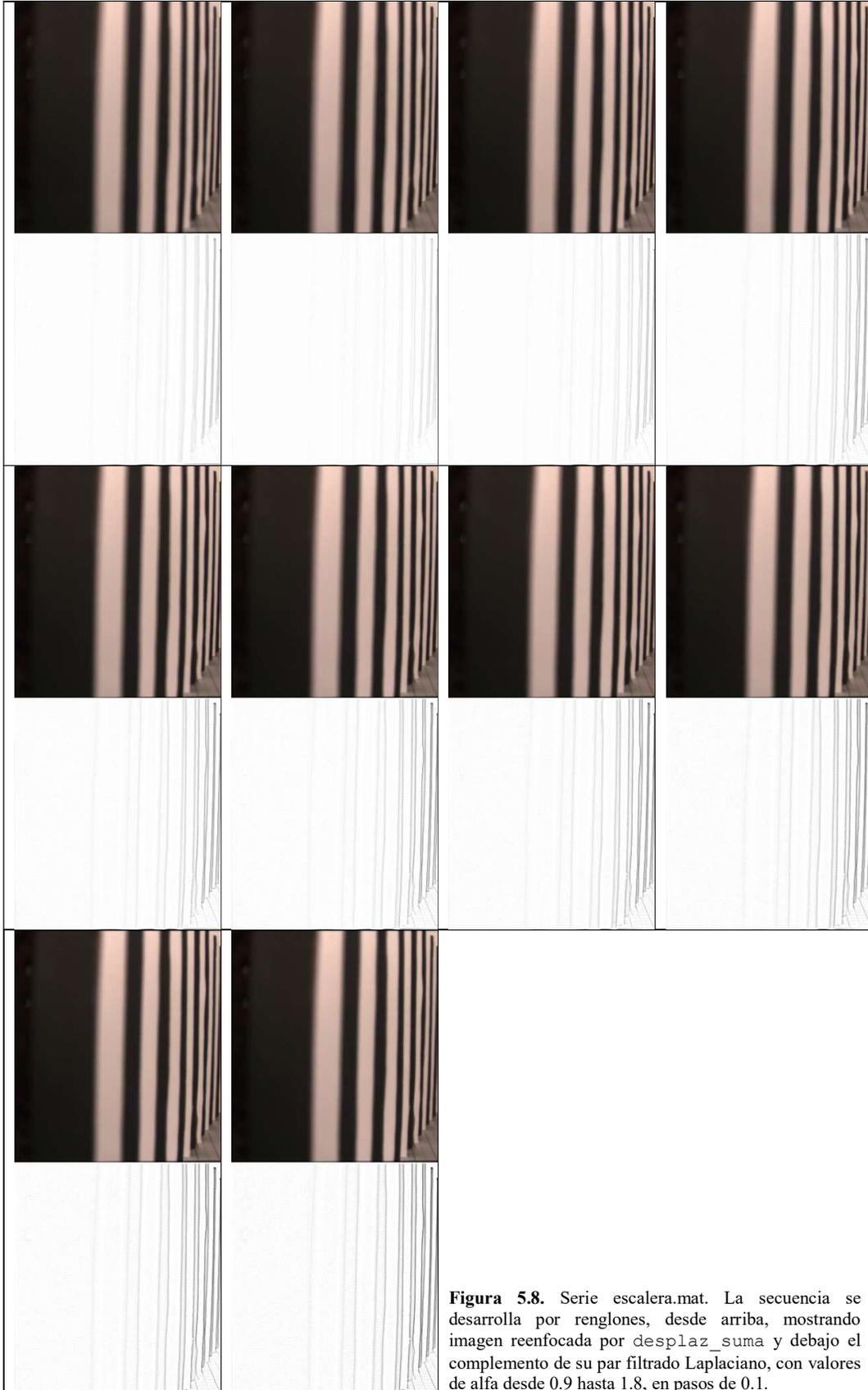


Figura 5.8. Serie escalera.mat. La secuencia se desarrolla por renglones, desde arriba, mostrando imagen reenfocada por `desplaz_suma` y debajo el complemento de su par filtrado Laplaciano, con valores de alfa desde 0.9 hasta 1.8, en pasos de 0.1.

Para hacer más evidentes los cambios antes mencionados, ensayamos diferentes detectores de bordes que nos permitan visualizar como imagen binaria sólo una región en foco y comparar lo que sucede entre dos imágenes reenfocadas con parámetros consecutivos. Sometimos a prueba la mayoría de los detectores de borde clásicos: Sobel, Roberts, Prewitt, Canny y Laplaciano del Gaussiano (LoG), cada uno con diferentes umbrales según lo admitía cada método. Sobel, Robert y Prewitt utilizan la magnitud del gradiente para detectar discontinuidades en intensidad usando la derivada primera. Difieren en algunos de los valores que sus máscaras de convolución utilizan. Canny también se basa en el gradiente local pero antes suaviza la imagen con un gaussiano y luego determina un rango de umbrales dentro del cual valida el borde. Canny es un muy potente detector y en algunas aplicaciones su alta sensibilidad marca demasiados detalles como bordes. Una completa descripción de los filtros mencionados puede hallarse en [87]. De los detectores ensayados, el que nos permitió visualizar con mayor claridad los cambios fue el LoG. Mencionaremos brevemente que los filtros Laplacianos detectan los bordes aplicando la derivada segunda y detectando los cruces por cero, pero son muy sensibles al ruido. Entonces antes de aplicarlos, suavizamos la imagen con un Gaussiano y luego hacemos el Laplaciano. Cabe aclarar que los bordes visibles varían según el umbral del filtro (en este caso, 0.02) y por lo tanto no indican la extensión de este “volumen en foco”; es una estrategia al sólo efecto de visualizar los pasos del foco en función de alfa.

En la Figura 5.9 mostramos los resultados del par Imagen Enfocada – LoG para un rango de alfa entre 0.9 y 1.8, en pasos de 0.1.

Observamos en la Figura 5.10 que los LoG van desplazándose muy levemente hacia al frente a medida que avanzamos con el parámetro alfa desde 0.9 hacia 1.8. Dado que los escalones tienen 1cm de profundidad por 1cm de altura y que la captura se hizo en diagonal, la distancia aproximada en profundidad entre borde y borde es de 1.4cm. Las ínfimas diferencias entre los LoG indican que los pasos de enfoque se mueven distancias del orden de un par de milímetros, para esta configuración de escena. Pasos menores a 0.1 de alfa no evidencian cambios entre dos imágenes reenfocadas consecutivas.

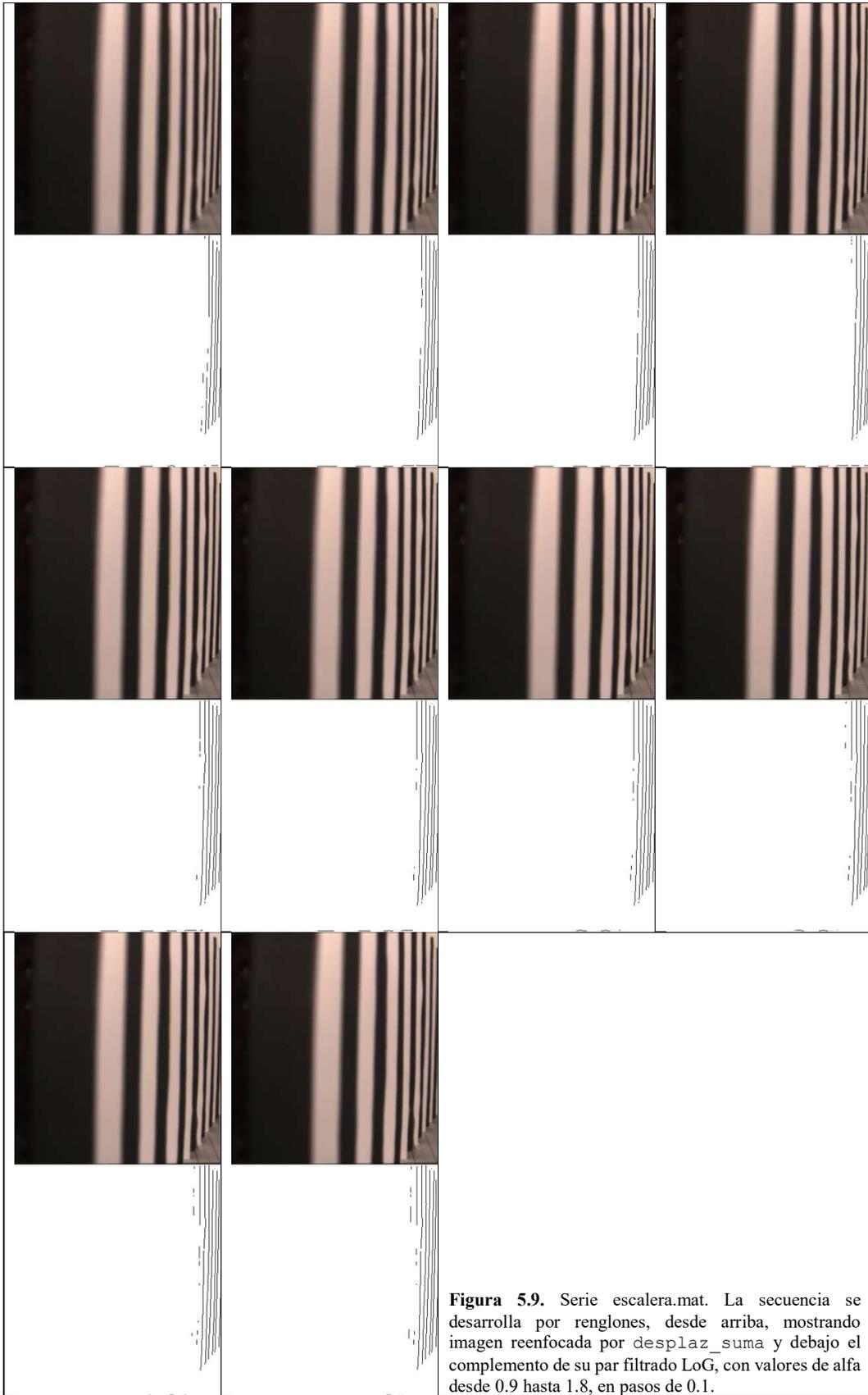


Figura 5.9. Serie escalera.mat. La secuencia se desarrolla por renglones, desde arriba, mostrando imagen reenfocada por `desplaz_suma` y debajo el complemento de su par filtrado LoG, con valores de alfa desde 0.9 hasta 1.8, en pasos de 0.1.

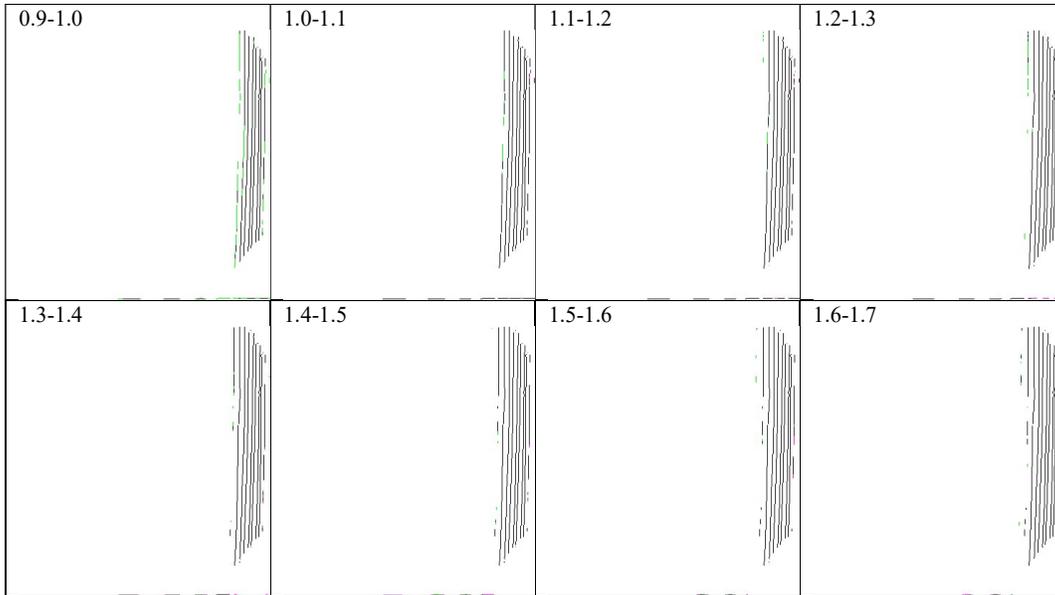


Figura 5.10. Imágenes fusionadas, combinación de los LoG del algoritmo `desplaz_suma` para parámetros alfa contiguos, expresados como diferencias `alfa_x, alfa_y` en cada imagen. Donde no hay diferencias se mantiene la imagen binaria; en verde y magenta se ven las diferencias que aportan las imágenes superpuestas.

5.3.2. Ensayos Algoritmo Sesgado de la Radiancia (`RefocusRadiancia`)

Tiempos de cómputo

Los tiempos de cómputo del algoritmo `RefocusRadiancia` son los mostrados en el Gráfico E, para el mismo LF de LytroF01 con el que ensayamos `desplaz_suma`. Se promediaron los tiempos de 30 corridas por cada parámetro Theta, para $\text{Theta} = 5 : 1 : -5$. El promedio de corrida fue de 31.66s y no se apreciaron diferencias significativas de tiempo para diferentes Theta, siendo el mínimo de 27.99s y el máximo de 32.57s para este ejemplo. Cabe recordar que $\text{Theta}=0$ significa que la radiancia no sufre ningún sesgado y por lo tanto la imagen queda enfocada con el foco óptico de la captura original. Desde el punto de vista del proceso, el algoritmo en este caso no realiza ninguna interpolación y por lo tanto disminuye el tiempo de cómputo.

Ensayamos el algoritmo con imágenes de idéntica cantidad de lenticulas, pero diferente cantidad de pixeles por microlente. Usamos los mismos LF con las que probamos el algoritmo `desplaz_suma`. Los resultados volcados en Gráfico F indican que el incremento de tiempo de cómputo es lineal con el número de pixeles por lenticula.

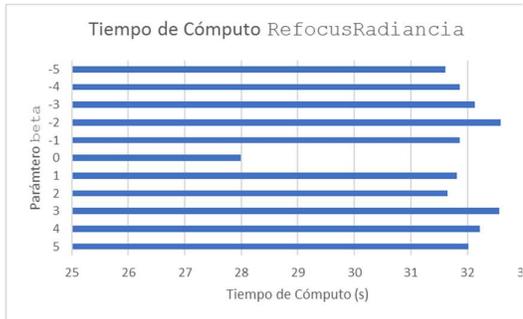


Gráfico E. Tiempos de cómputo para algoritmo Sesgado de la Radiancia, con diferentes parámetros θ .

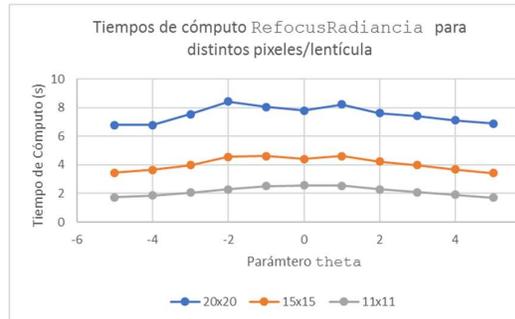


Gráfico F. Tiempos de cómputo para algoritmo Sesgado de la Radiancia, imágenes todas de 114x114 microlentes, pero con diferente número de pixeles / lentejuela.

Finalmente, probamos el algoritmo para diferente cantidad de lentejuelas del mismo tamaño (cantidad de pixeles), para determinar la dependencia del tiempo de cómputo con el tamaño de la matriz de microlentes. Los resultados mostrados en el Gráfico G y H indican que el tiempo de cómputo de `RefocusRadiancia` no es lineal con el número de microlentes, mostrando una tendencia exponencial.

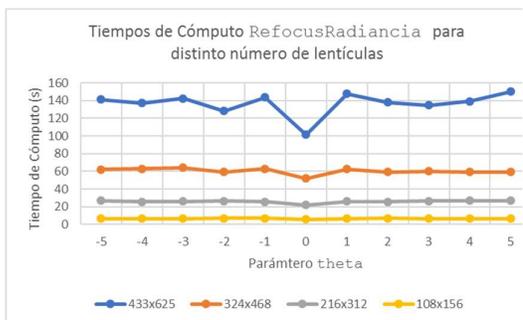


Gráfico G. Tiempos de cómputo para algoritmo Sesgado de la Radiancia, imágenes todas de 15x15 pixeles / lentejuela, pero con diferente número de microlentes.

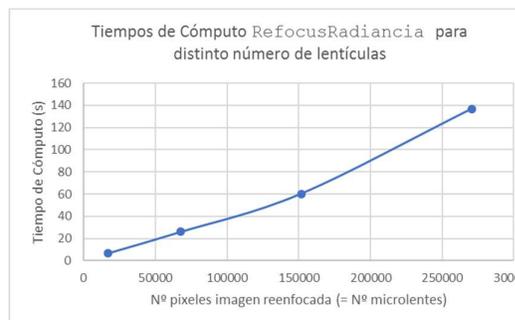


Gráfico H. Relación entre los Tiempos de cómputo y el tamaño de las imágenes reenfoçadas para algoritmo Sesgado de la Radiancia.

Enfoque en profundidad

En cuanto a la performance de `RefocusRadiancia` para refocalizar, ensayamos la misma Serie 707.mat con valores de Theta entre 4 y -4 en pasos unitarios. Estos valores son experimentales, en función de la visualización del fondo y frente en foco. Vemos resultados similares a los obtenidos con Desplazamiento-Suma, aunque se repite la dificultad para comparar cómo varía el foco. A los efectos de determinar las zonas de foco y dados los resultados obtenidos con el algoritmo Desplazamiento-Suma, mostraremos los ensayos realizados con el fantoma escalera. En la Figura 5.11 observamos las imágenes reenfoçadas y su correspondiente LoG, para un rango de theta entre 0 y -4.5 (enfoque más cercano a la cámara), en pasos de 0.5. Aplicamos el LoG a la imagen reenfoçada con el mismo umbral de 0.02 con el que trabajamos con Desplazamiento-Suma y el paso de 0.5 es el mínimo con el que obtuvimos diferencia apreciable entre reenfoçadas contiguas.

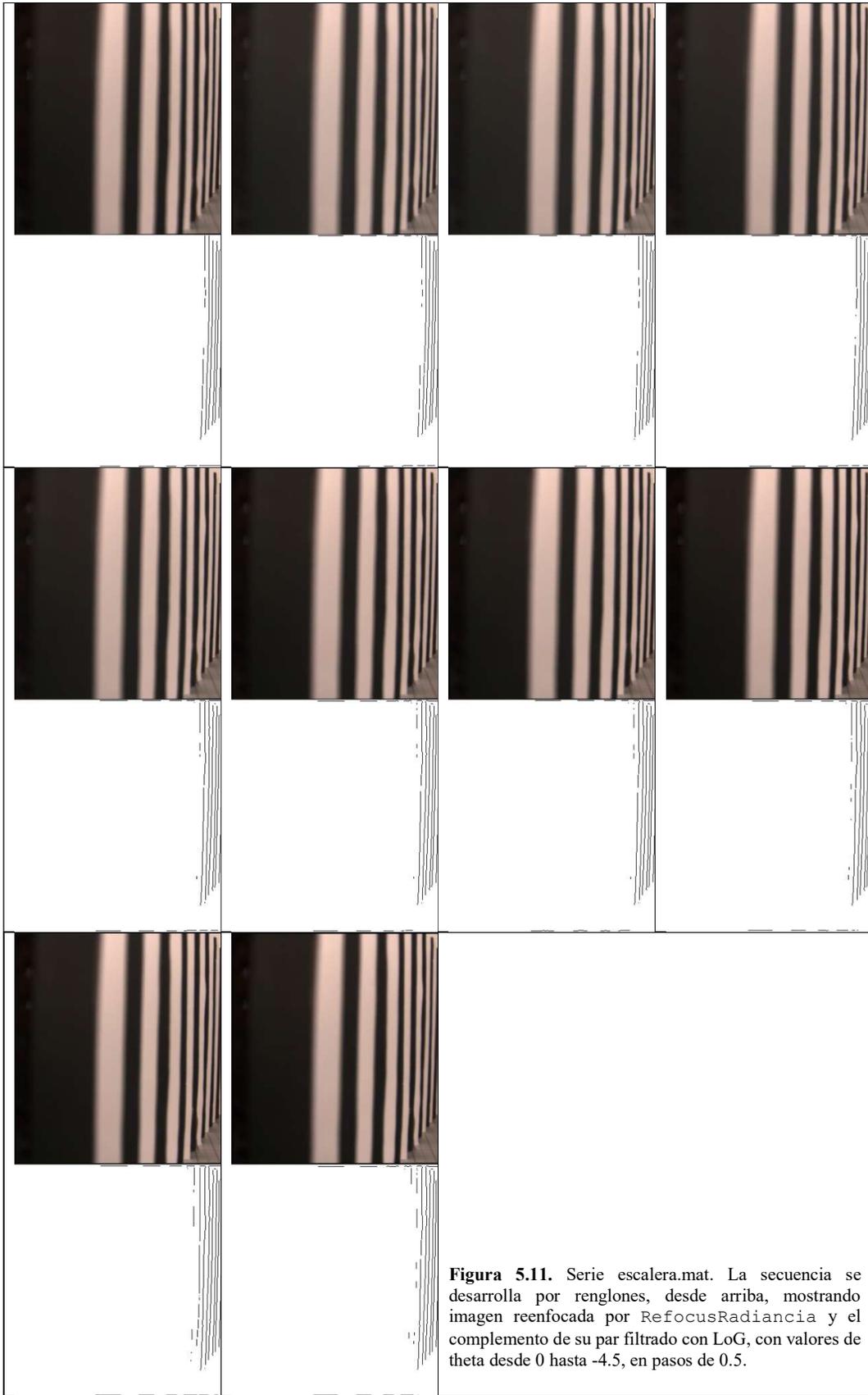


Figura 5.11. Serie escalera.mat. La secuencia se desarrolla por renglones, desde arriba, mostrando imagen reenocada por `RefocusRadiancia` y el complemento de su par filtrado con LoG, con valores de theta desde 0 hasta -4.5, en pasos de 0.5.

Analizando la Figura 5.12 observamos que los resultados son similares a Desplazamiento-Suma, aunque aparece alguna discontinuidad en el rango theta entre -2.0 y -3.0, donde no se aprecia un corrimiento de la región en foco hacia la cámara. Asimismo, hacia el final de la serie (rango -4.0,-4.5) hay una retracción de la zona enfocada hacia atrás. Por otra parte, cabe notar que si bien observamos el desplazamiento del foco hacia la cámara, no lo acompaña el mismo corrimiento en el fondo, lo que da como resultado un ensanchamiento de la región enfocada. Además de estos detalles, el costo del Sesgado de la Radiancia es su elevado tiempo de cómputo.

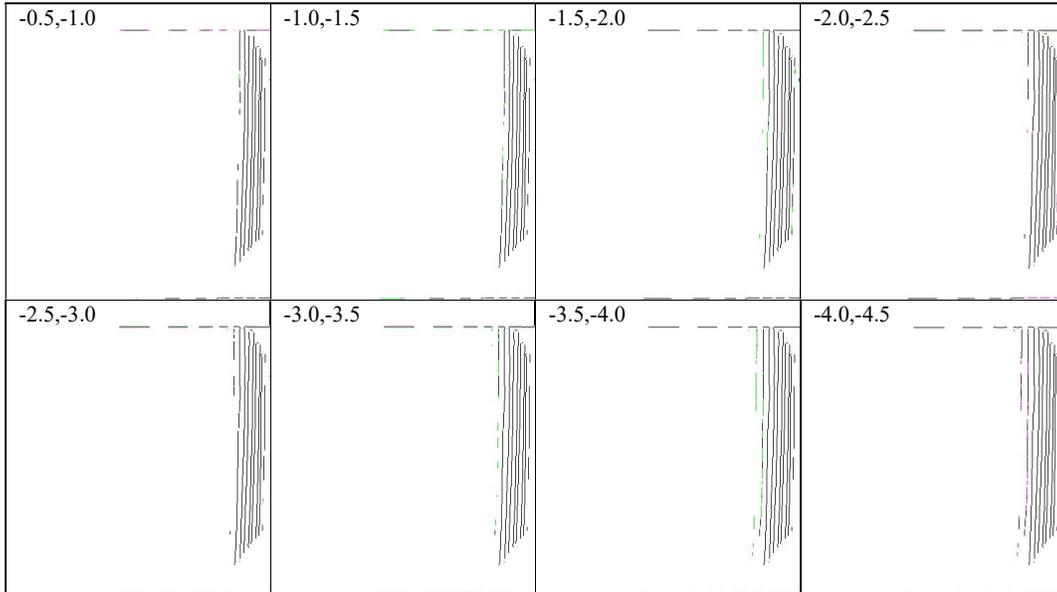


Figura 5.12. Imágenes compuestas, combinación de los LoG del algoritmo `RefocusRadiancia` para parámetros theta contiguos, expresados como diferencias θ_x, θ_y en cada imagen.

Atributos fotométricos

En relación con los efectos de los algoritmos sobre el balance de color y el brillo, comparamos las imágenes reenocadas con ambos algoritmos `Desplazamiento-Suma` y `Sesgado de la Radiancia`, con la imagen original renderizada (algoritmo `render`, estudiado en Capítulo 3). Observamos en la Figura 5.13 que, a simple vista, los reenfoques no introducen cambios en estos atributos.

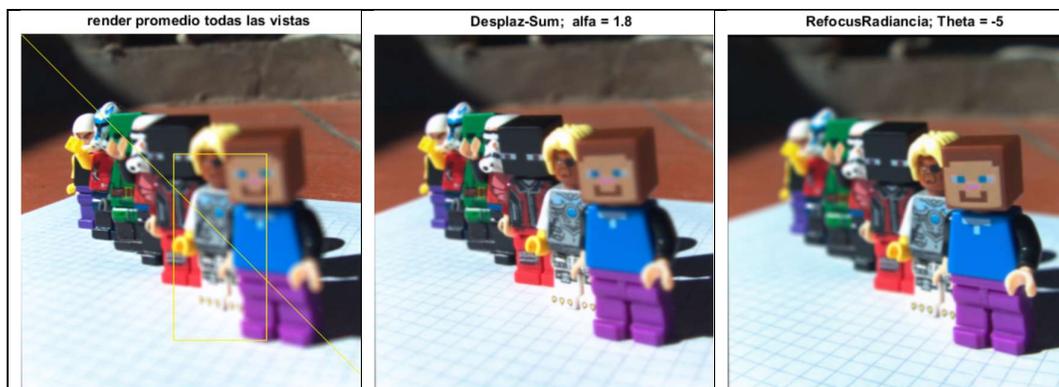


Figura 5.13. Preservación de los atributos de brillo, contraste y balance de color. La imagen original con su foco óptico (izq.) y dos ejemplos de reenfoque (centro: `Desplazamiento-Suma` y der.: `Sesgado de la Radiancia`).

En la Figura 5.14 mostramos los perfiles de imagen e histogramas para la diagonal y sector mostrados en la primera imagen de la Figura 5.13. Observamos que no hay diferencias apreciables entre los perfiles de los reenfoques. Por otra parte, los histogramas tampoco presentan cambios sustanciales, por lo que podemos concluir que no se ven afectados brillo, contraste ni color por los reenfoques de Desplazamiento-Suma ni Sesgado de la Radiancia.

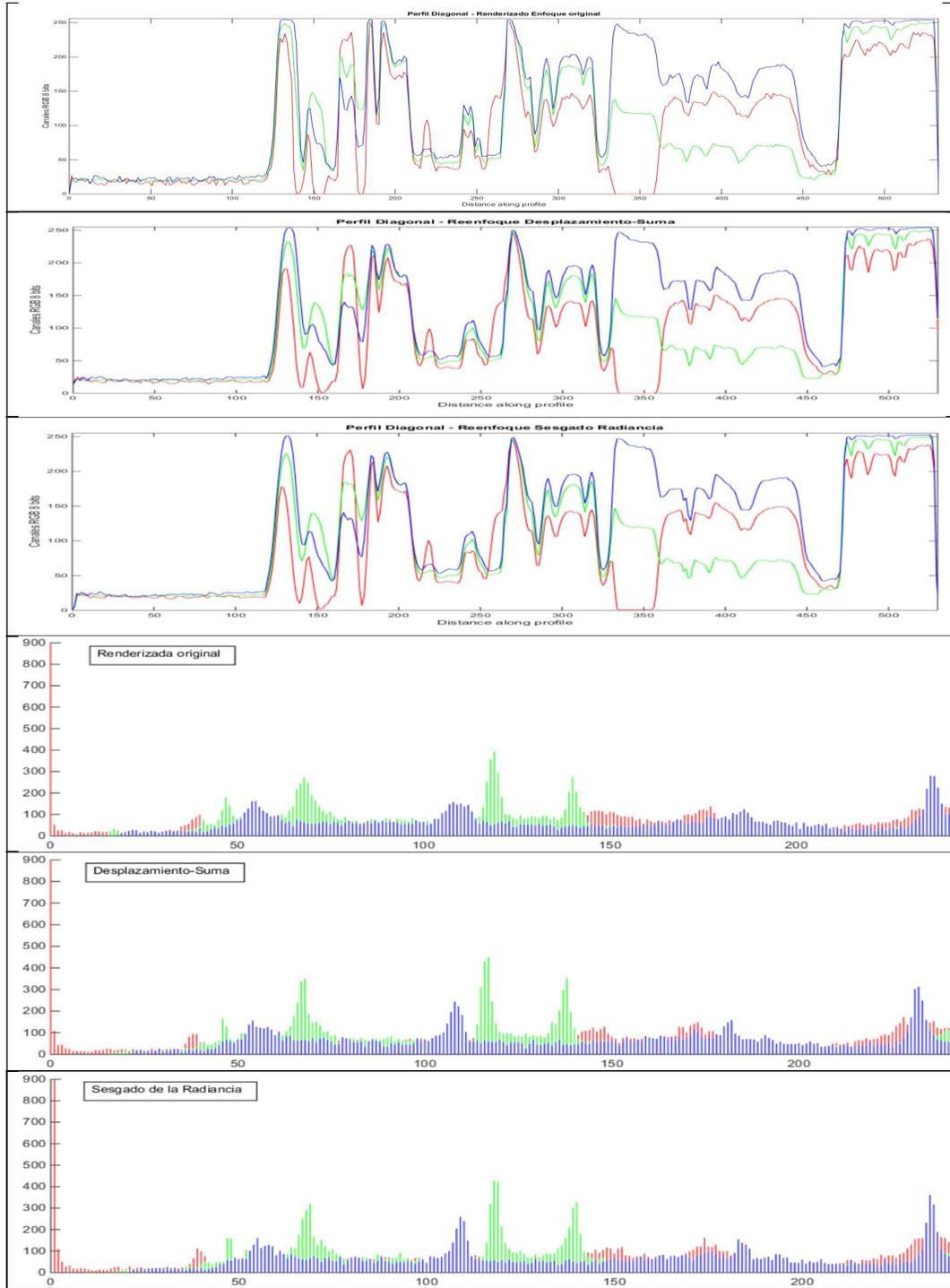


Figura 5.14. Preservación de los atributos de brillo, contraste y balance de color. De arriba hacia abajo: Perfiles de imagen por canal de color para imagen de enfoque óptico original y las reenfocadas por Desplazamiento-Suma y Sesgado de la Radiancia. Histogramas de las mismas imágenes.

5.3.3. Ensayos Algoritmo Corte del Espectro (RefocusEspectro)

Tiempos de cómputo

Evaluamos el tiempo de cómputo para la misma imagen 707.mat y obtuvimos los valores volcados en Gráfico I, para un rango de beta entre -0.020 y +0.020, en pasos de 0.005. Estos valores son experimentales y en este rango se focaliza en el fondo y al frente, respectivamente. Observamos que hay mayor dispersión de los tiempos, comparado con los algoritmos anteriormente ensayados, aunque estamos hablando de una diferencia máxima de 0.24 segundos. Aun cuando deben interpolarse los espectros para los tres planos de color, es notable la eficiencia en términos de tiempo de cómputo.

Ensayamos a continuación el algoritmo para diferente cantidad de pixeles por lenticula, como venimos haciendo con los algoritmos anteriores. Observamos en el Gráfico J que el tiempo de cómputo es lineal con el número de pixeles por lenticula.

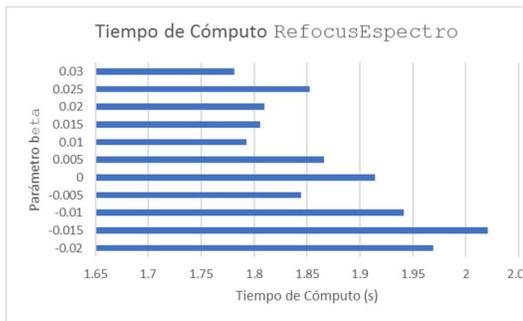


Gráfico I. Tiempos de cómputo para algoritmo Corte del Espectro, con diferentes parámetros beta.

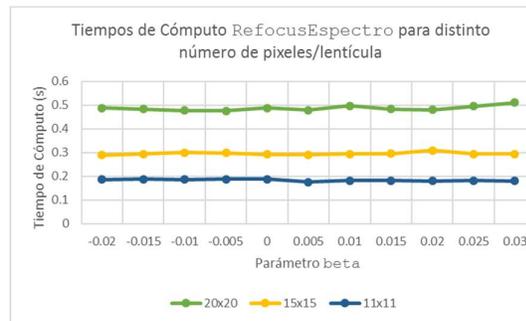


Gráfico J. Tiempos de cómputo para algoritmo Corte del Espectro, imágenes todas de 114x114 microlentes, pero con diferente número de pixeles / lenticula.

Mostramos ahora en el Gráfico K el resultado de ensayar RefocusEspectro para imágenes de diferente cantidad de lenticulas de igual tamaño y la relación entre los tiempos de cómputo y el tamaño de las imágenes reenfoadas en Gráfico L. Observamos que de manera similar a como sucede con el Segado de la Radiancia, el comportamiento del tiempo de cómputo a medida que se incrementa el número de microlentes es exponencial.

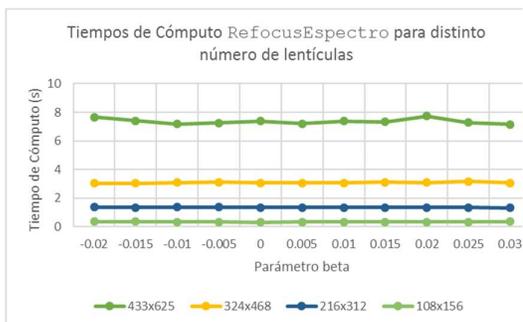


Gráfico K. Tiempos de cómputo para algoritmo Corte del Espectro, imágenes todas de 15x15 pixeles / lenticula, pero con diferente número de microlentes.

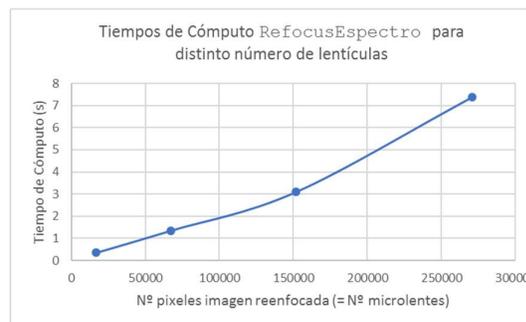
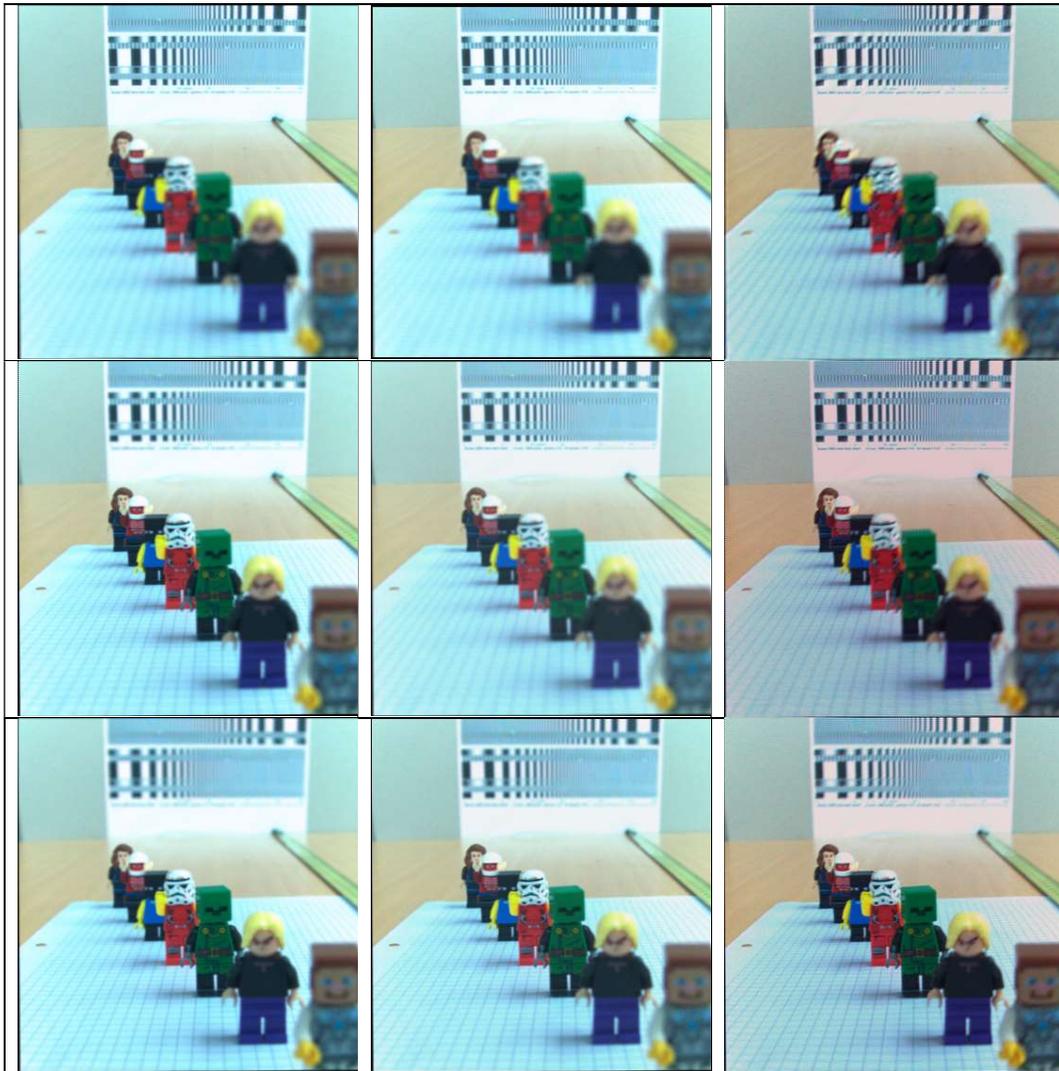


Gráfico L. Relación entre los Tiempos de cómputo y el tamaño de las imágenes reenfoadas para algoritmo Corte del Espectro.

Atributos fotométricos

Para el análisis del reenfoque de Corte del Espectro, vale la pena comparar algunas imágenes de la serie 707.mat para observar las diferencias colorimétricas de los resultados. En la Figura 5.15 mostramos algunos reenfoques obtenidos con Desplazamiento-Suma, Sesgado de la Radiancia y Corte del Espectro y observamos que este último algoritmo, si bien introduce alguna variación en el balance de color, muestra resultados mejor contrastados y de mayor nivel de detalle en comparación con los algoritmos de dominio espacial.



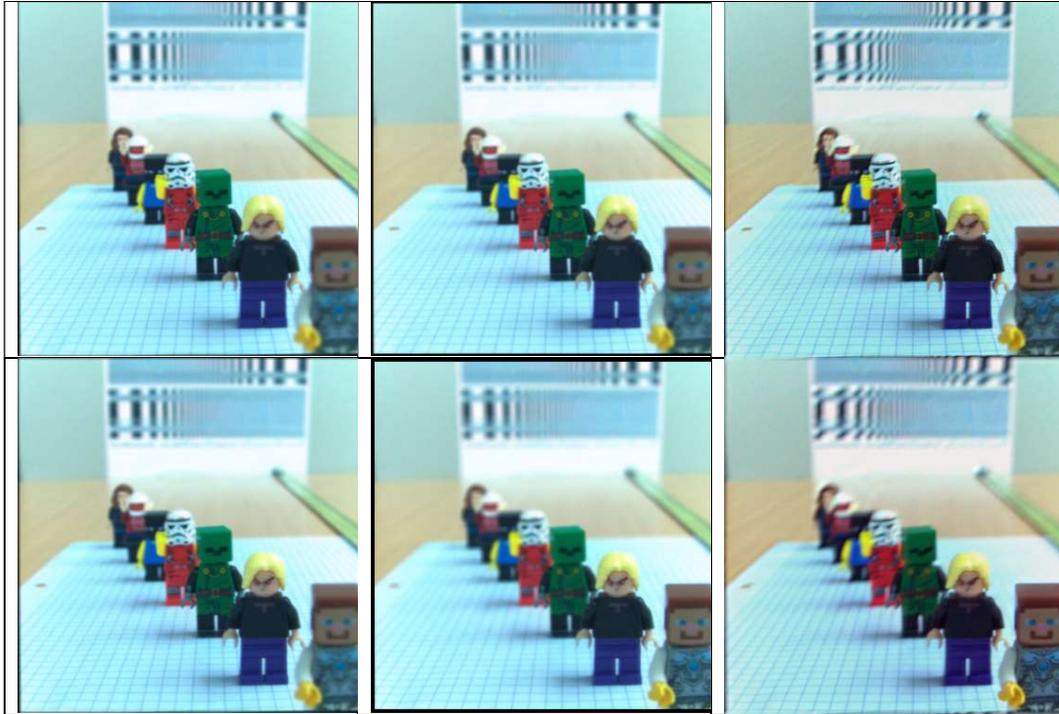
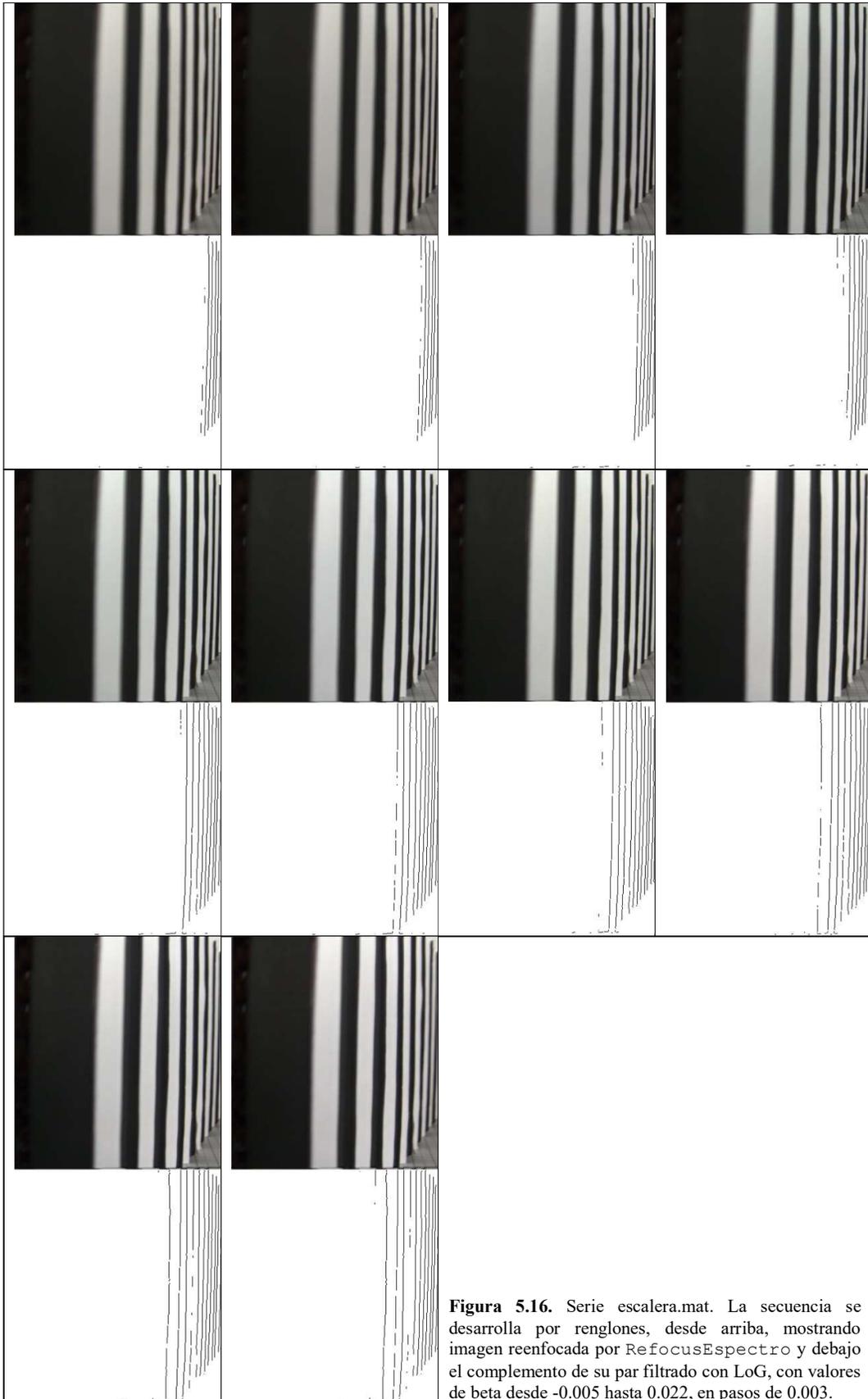


Figura 5.15. Primera columna: reenfoques con `desplaz_sum` ($\alpha = 0.8:0.3:2$). Segunda columna: reenfoques con `RefocusRadiancia` ($\theta = 4:-3:-10$). Tercera columna: reenfoques con `RefocusEspectro` ($\beta = -0.015:0.01:0.025$).

Enfoque en profundidad

Para observar las variaciones de la posición del foco en relación con la variación de los parámetros de `RefocusEspectro`, mostramos en la Figura 5.16 los resultados de reenfoque para β en el rango -0.005 hasta 0.022 , en pasos de 0.003 .

La Figura 5.17 nos presenta el detalle de las diferencias entre un paso y el siguiente del parámetro β . De manera similar a lo que sucede con `RefocusRadiancia`, si bien observamos el desplazamiento de la región en foco hacia la cámara a medida que aumenta β , no hay una correspondencia de dicho corrimiento en el fondo. Vemos que la región enfocada va ensanchándose.



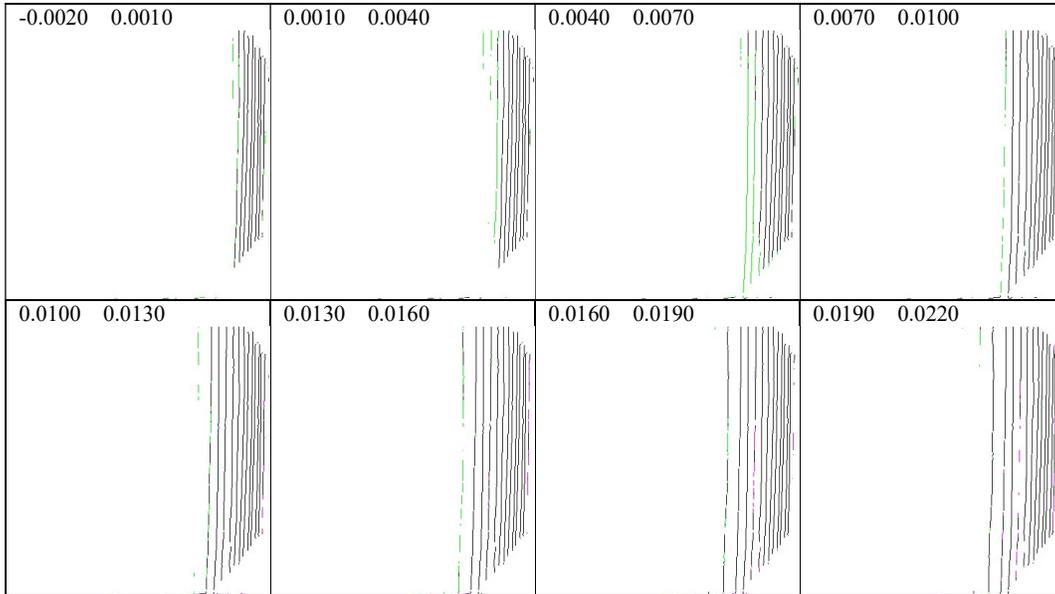


Figura 5.17. Imágenes compuestas, combinación de los LoG del algoritmo `RefocusEspectro` para parámetros beta contiguos, expresados como diferencias `beta_x`, `beta_y` en cada imagen.

5.3.4. Ensayos Algoritmo Foco Volumétrico (`LFBUILD4DFREQHYPERFAN`)

Comentaremos brevemente que este script perteneciente al `LFToolbox v0.4`, construye un filtro 4D en el dominio frecuencial que se aplica luego con la función `LFFILT4DFFT`. El resultado es una imagen reenfocada en un rango de profundidades en la escena, a lo que su autor Dansereau denomina “foco volumétrico” [73].

Aplicamos dicho filtro a la serie `707.mat` a los efectos de visualizar en primera instancia, la relación entre las pendientes (`slope1`, `slope2`) que requiere como parámetros el filtro y la posición del foco volumétrico en la escena. Recordamos que la distancia entre la cámara y el fondo es de 70cm. Luego de varias pruebas con diferentes valores de pendientes, hallamos que el rango entre el fondo y el frente va desde 1 (fondo) hasta -1 (frente).

En cuanto a los tiempos de cómputo, para las imágenes de Lytro F01 (378x379x11x11) el tiempo promedio de cómputo para un volumen focal es de 7.93 segundos pero presenta una dispersión cuando varían los parámetros `slope1`, `slope2`, como podemos observar en Gráfico M. Al igual que como hicimos con los demás algoritmos, ensayamos Foco Volumétrico para diferentes tamaños de imagen (diferente cantidad de lenticulas con idéntico número de pixeles por microlente), resultados que presentamos en el Gráfico N.

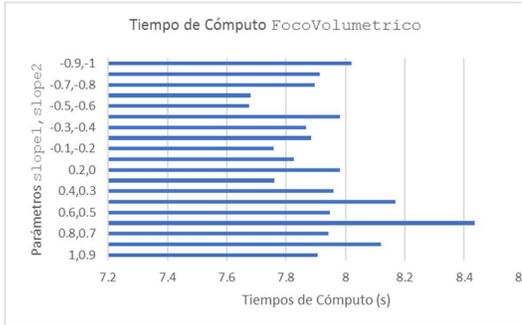


Gráfico M. Tiempos de cómputo para algoritmo Foco Volumétrico, con diferentes parámetros slope1, slope2.

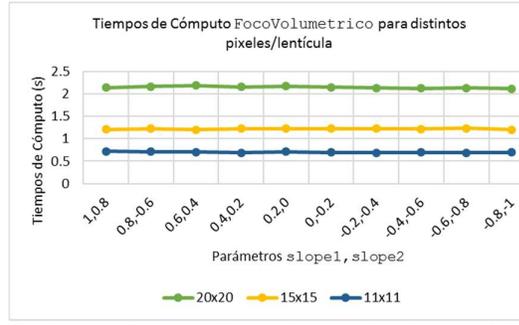


Gráfico N. Comparación de tiempos de cómputo del algoritmo Foco Volumétrico para diferente cantidad de píxeles por microlente, a igual número de microlentes.

También evaluamos la relación entre los tiempos de proceso de refocalización y el tamaño de las imágenes reenfocadas. En los Gráficos O y P mostramos que dicha relación es lineal.

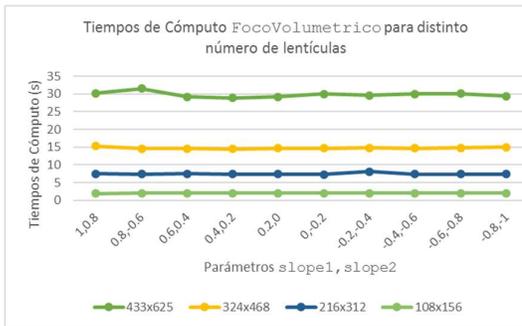


Gráfico O. Comparación de tiempos de cómputo del algoritmo Foco Volumétrico para diferente cantidad de microlentes, a igual número de píxeles por microlente.

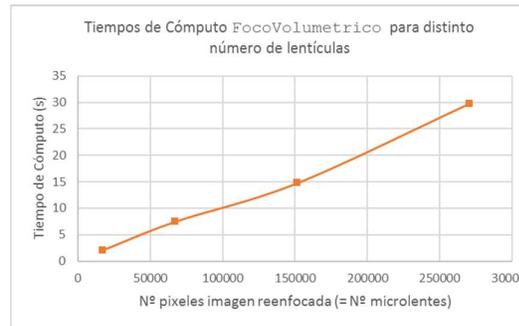


Gráfico P. Relación entre los Tiempos de cómputo y el tamaño de las imágenes reenfocadas para algoritmo Foco Volumétrico.

En la Figura 5.18 observamos la escena dividida en 5 zonas en foco, desde el fondo hasta el frente. La quinta imagen muestra que no hay foco más allá de la pendiente -1. La sexta imagen tomó como pendientes la correspondiente al fondo y al frente, de modo que la escena está en foco en toda su profundidad.

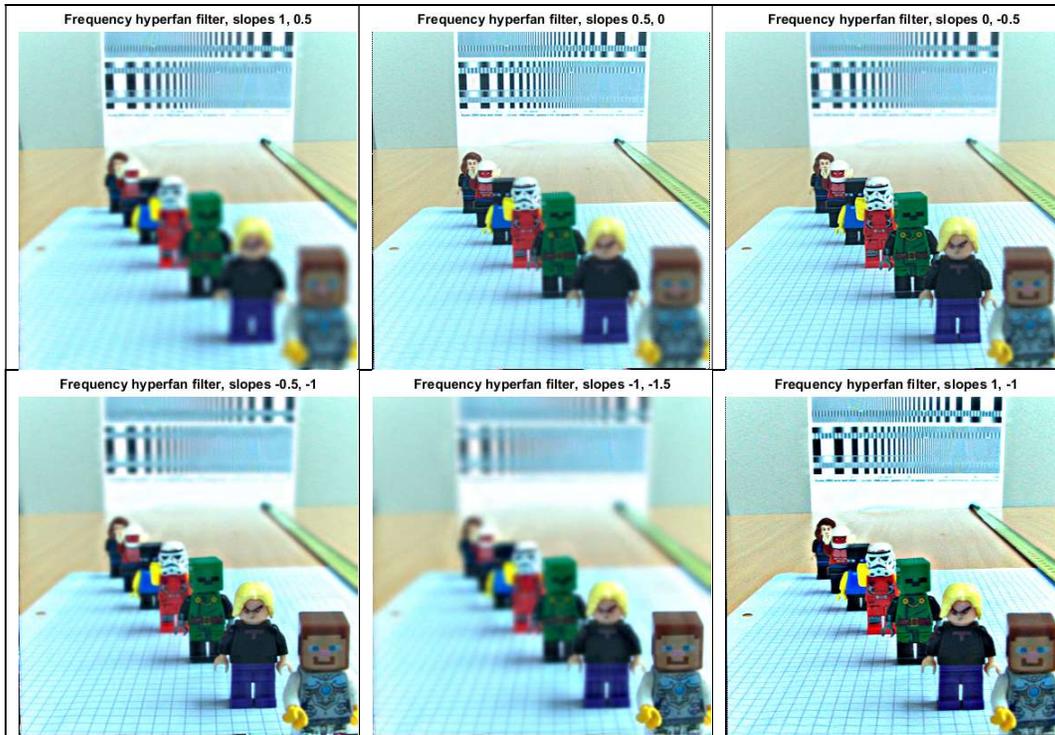


Figura 5.18. Serie 707.mat. Foco Volumétrico para diferentes rangos $slope_1, slope_2$. Ordenadas por renglones de arriba hacia abajo, va reenfocando desde el fondo hacia el frente. La última imagen presenta el resultado de reenfoque entre 1 y -1 (toda la profundidad en la escena).

Ensayamos la serie 713.mat que consiste en la toma de los Lego separados la misma distancia (4 cm) pero con 80 cm fondo-cámara. El rango útil de foco fue el mismo que para la serie anterior. Para confirmar esta prueba evaluamos entonces una fotografía de una escena con mayores profundidades (imagen provista junto con el LFTtoolbox de Dansereau). Los resultados son los mismos: el rango útil de foco va entre 1 y -1. La Figura 5.19 muestra los resultados.

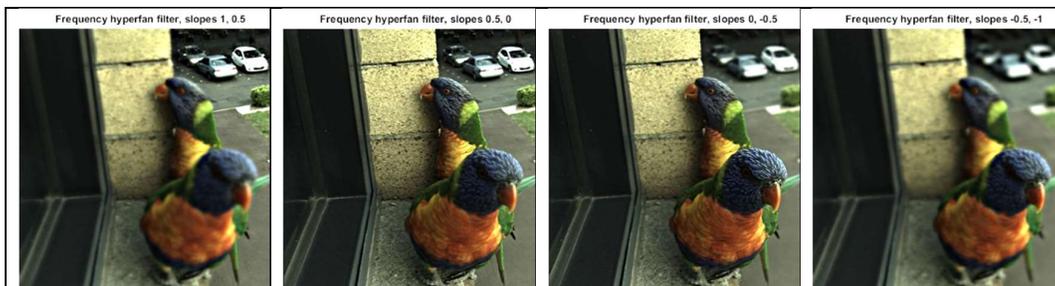


Figura 5.19. Ensayo de la imagen IMG_0005__Decoded.mat (LFTtoolbox) con el algoritmo de Foco Volumétrico, para un rango total 1 a -1 de profundidades. Imagen de acceso libre en [62].

En relación a la preservación del balance de color, brillo y contraste originales, el algoritmo de Foco Volumétrico no altera estas propiedades, comparadas las imágenes de reenfoque con la imagen renderizada aplicado el algoritmo `render`.

A los efectos de evaluar la relación entre los parámetros $slope_1, slope_2$ y su imagen reenfocada, sometemos al algoritmo a un filtrado LoG con el mismo umbral de 0.02 aplicado en

los casos anteriores. Observamos en la Figura 5.20 que en los extremos de la escena (mayor profundidad y mayor cercanía a la cámara) el LoG no detecta bordes para el nivel de umbral 0.02. Otro resultado es que el algoritmo en esta escena, no desplaza el volumen en foco a medida que varía el delta $slope1, slope2$ a la vez que observamos, al final de la serie cuando el reenfoque se acerca a la cámara, que no se logra hacer foco.

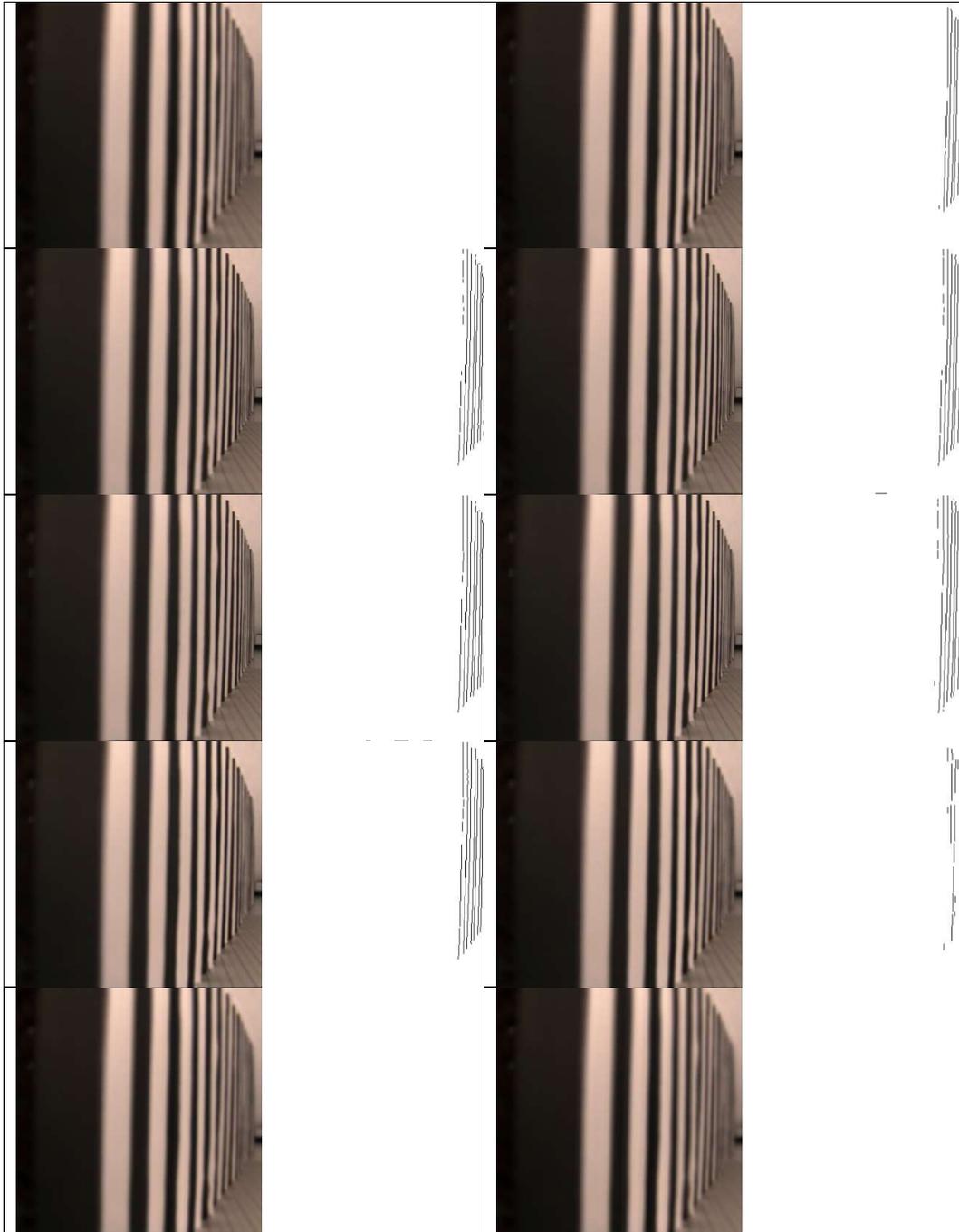


Figura 5.20. Serie escalera.mat. La secuencia se desarrolla por renglones, desde arriba, mostrando imagen reenfocada por *FocoVolumetrico* y el complemento de su par filtrado con LoG, con valores de *delta* desde 0.60-0.45 hasta -0.60-0.75, en pasos de 0.15.

5.3.5. Ensayos Algoritmo Renderizado Plenoptic 2.0 por Bloques (BasicFullRender)

Utilizamos nuestro script `BasicFullRender` que tiene como parámetros el tamaño de la microimagen (en pixeles de lado, para microimagen cuadrada) y tamaño del bloque, que determinará la profundidad donde se hará foco. También incluye los desvíos en x e y respecto del centro donde se ubica el bloque. Dado que no interesa en esta prueba la perspectiva, estos parámetros `offsetX`, `offsetY` se colocan en cero, lo que significa que los bloques de tamaño M quedan centrados en la microimagen de tamaño μ .

Ensayamos la imagen `472.jpg` disponible en [40] que es la radiancia capturada por una cámara de arquitectura Plenoptic 2.0. Es una imagen de $5433 \times 7240 \times 3$ uint8 de 72×96 microimágenes de 75×75 pixeles cada una. La salida es una imagen enfocada a una profundidad dependiente del tamaño M del bloque seleccionado: cuanto mayor es M , más cercano a la cámara se hace foco. Observamos los resultados en la Figura 5.21. Dado que no contamos con fantasmas similares a la escalera en arquitectura Plenoptic 2.0, es difícil medir la resolución en profundidad que proveen los cambios en tamaño de bloques. Sí podemos afirmar, observando la Figura 19, que a medida que se enfoca al frente (mientras va aumentando M), se desenfoca al fondo, lo que indica que la región en foco va desplazándose correspondientemente.

Medimos el tiempo de cómputo en relación con el tamaño de bloque. Se promediaron los tiempos de 30 corridas por tamaño de bloque. El Gráfico Q muestra los resultados. Por otra parte, evaluamos la relación entre dichos tiempos y el tamaño de la imagen, recortando la misma y corriendo el algoritmo. La imagen original que ensayamos tiene 5433×7240 pixeles y la recortamos al 75, 50 y 25% de su tamaño, promediando para cada caso 30 corridas con bloques desde 2×2 hasta 16×16 pixeles. Vemos en el Gráfico R que la relación entre el tamaño de la imagen procesada y el tiempo de cómputo es lineal.

En términos de tiempo computacional las diferencias cuando se enfoca con diferentes tamaños de bloque son insignificantes y no guardan relación con dicho tamaño. Estos resultados son esperables dado que el algoritmo encuentra los índices (posiciones) de los pixeles calculando de manera vectorial los saltos dependientes de M (tamaño del bloque). El número de índices coincide con el tamaño de la matriz original por lo que no hay dependencia del valor de M . La Figura 5.21 muestra el cambio de foco en función del tamaño de bloque.

Las características de la imagen (brillo, balance de color) se mantienen idénticas a la radiancia capturada, puesto que la imagen refocalizada surge de reubicar posiciones de los pixeles de acuerdo al tamaño de bloque M , sin alterar sus valores. Recordemos que `BasicFullRender` no utiliza interpolación sino reasignación de posiciones por vectorización (de aquí su performance en tiempos de cómputo).

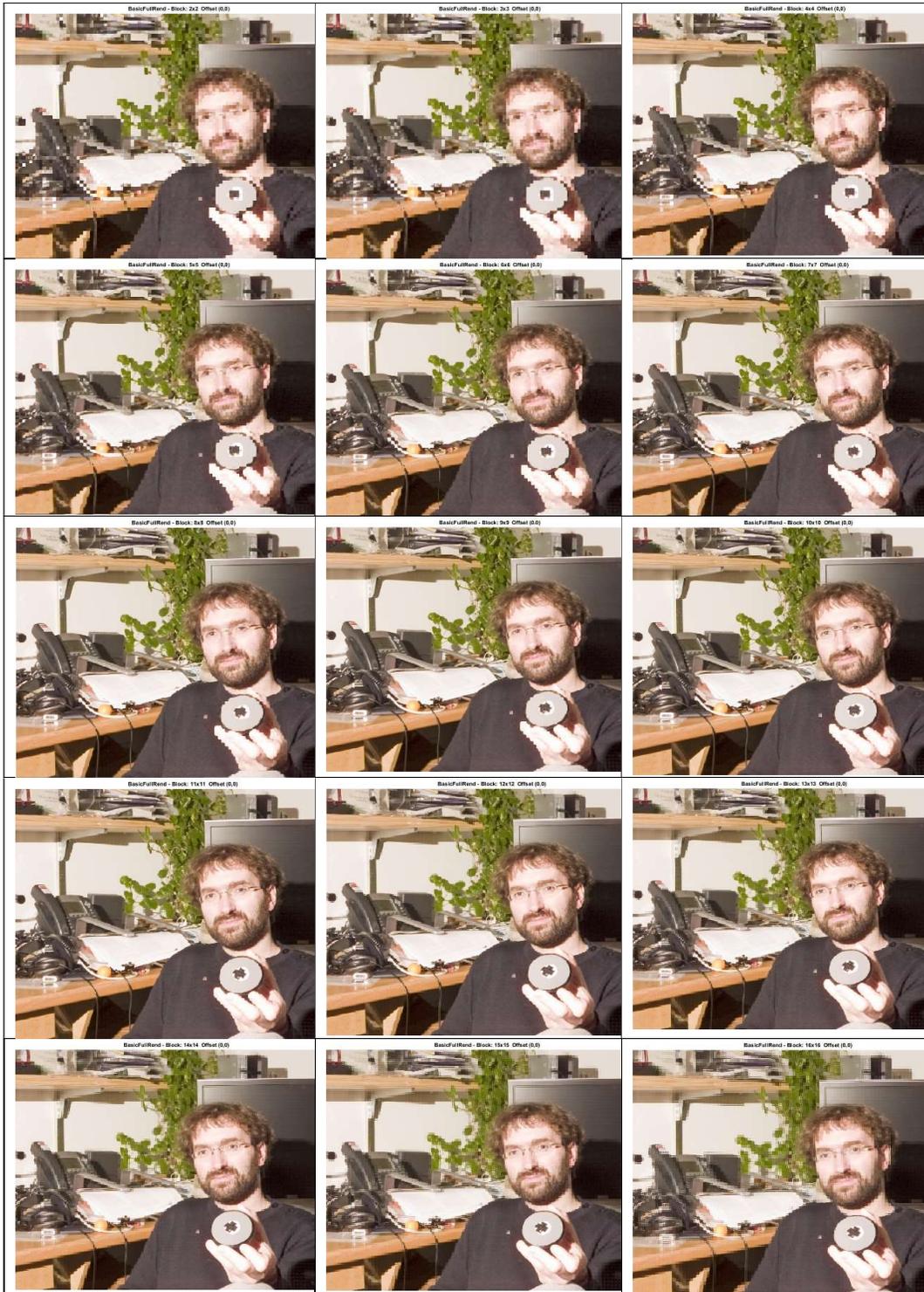


Figura 5.21. Refocalizaciones con algoritmo BasicFullRender para diferentes tamaños de bloque $M \times M$, desde 2×2 hasta 16×16 recorriendo por renglones la secuencia de imágenes. Imagen 472.jpg de dominio público en [40].

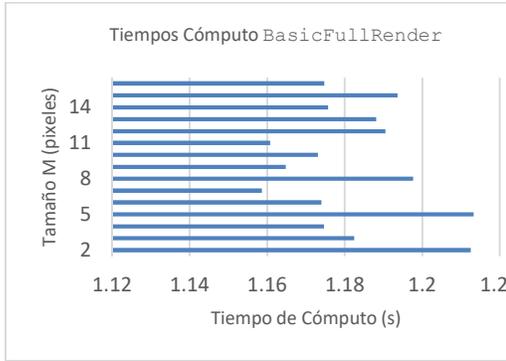


Gráfico Q. Tiempos de cómputo para algoritmo de renderizado para Plenoptic 2.0, con diferentes tamaños de bloque.

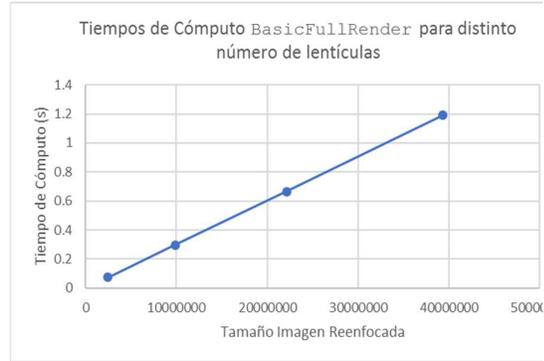


Gráfico R. Tiempos de cómputo para algoritmo de renderizado para Plenoptic 2.0, con diferentes tamaños de imagen reenfocada.

5.4. Comparación de los Resultados

En la Tabla I volcamos los resultados de los ensayos, a los efectos de compararlos.

Limitaciones constructivas del fantoma sumadas a las limitaciones de resolución de la cámara condujeron a algunas imprecisiones vistas como discontinuidades en las imágenes de borde, motivo por el cual hacemos una valoración cualitativa de los rangos de enfoque. Comparamos cómo se comportan dichos rangos a medida que pasamos de un valor del parámetro del algoritmo evaluado al siguiente, observando si se desplazan y si mantienen su profundidad. Establecemos una valoración subjetiva para este ítem que denominamos Resolución en Profundidad, de tres niveles: Mala, Regular y Buena.

El tiempo de cómputo promedio para las refocalizaciones corresponde a una imagen RGB de 4169 x 4158 píxeles, organizados en 379 x 378 microimágenes de 11 x 11 píxeles cada una. La excepción es el caso de Plenoptic 2.0 dado que contamos con radiancias capturadas con un único dispositivo, color RGB de 5433x7240 píxeles organizados en 72 x 96 microimágenes de 75 x 75 píxeles.

Dado que no contamos con una cámara del tipo Plenoptic 2.0, no tenemos capturas de las mismas escenas con las que comparamos la performance de los algoritmos para Plenoptic 1.0. Entonces hay algunos ítems que no podemos aplicar al caso particular de Plenoptic 2.0, en cuyo caso decimos que no tenemos datos (S/D). Pero como se ha estudiado en Capítulos anteriores, Plenoptic 2.0 ofrece ventajas muy importantes en comparación con Plenoptic 1.0, como son la obtención de imágenes reenfoçadas del mismo tamaño que la original, la posibilidad de la super-resolución [88], [89], [90] y los tiempos de proceso que, en nuestros ensayos, lo convierten en el más rápido de los algoritmos evaluados.

Si comparamos los primeros cuatro algoritmos del cuadro, observamos que Sesgado de la Radiancia, Corte del Espectro y Foco Volumétrico presentan características similares excepto el desbalance de color que introduce el Corte del Espectro. A la vez este último presenta el menor tiempo de cómputo de los tres. Si el objetivo es elaborar un mapa de profundidades a partir de la pila focal, no interesa si hay problemas de color. En este caso, de los tres algoritmos nos inclinamos por el más veloz.

TABLA I
COMPARACIÓN DE RESULTADOS DE LOS ALGORITMOS ENSAYADOS

Algoritmo	Tiempo Promedio de Refocalización [s]	Relación de Tiempo de Cómputo – Tamaño microimagen	Relación de Tiempo de Cómputo – Tamaño imagen reenfozada	Preservación del balance color	Preservación brillo y contraste	Resolución en Profundidad
Desplazamiento Suma	2.17888	lineal	lineal	sí	sí	buena
Sesgado de la Radiancia	31.66433	lineal	no lineal	sí	sí	regular
Corte del Espectro	1.87245	lineal	no lineal	no	no	regular
Foco Volumétrico	7.93300	lineal	no lineal	sí	sí	regular
Plenoptic 2.0 por Bloques	1.18228	S/D	lineal	sí	sí	S/D

Los mejores resultados que obtuvimos en resolución de enfoque en profundidad fueron con el algoritmo Desplazamiento Suma. Esto, sumado a que no altera las características fotométricas de la imagen original, lo convierte en una opción a elegir entre los algoritmos ensayados. Asimismo, podemos agregar que en términos de velocidad el Desplazamiento Suma está muy cerca del Corte del Espectro. Por lo antes dicho, Desplazamiento Suma ha sido el algoritmo que tuvo la mejor performance en los ensayos realizados.

Capítulo 6

Conclusiones

En la presente Tesis determinamos las posibilidades de la utilización del modelo de campos de luz para la generación de nuevas representaciones de una escena 3D analizando la información espacio-angular que contiene la función plenóptica y la manera de codificarla en una matriz 4D, seleccionando la parametrización de dos planos paralelos.

Analizamos los resultados de los autores Ng, Lumsdaine, Georgiev, Pérez Nava y Dansereau para la visualización en diferentes perspectivas y el reenfoque digital de campos de luz. Implementamos en MATLAB la codificación de imágenes crudas de cámaras plenópticas en matrices 4D y obtuvimos reenfoque en el dominio espacial con los algoritmos Desplazamiento-Suma y Sesgado de la Radiancia de Ng y Transformada de la Pila Focal de Pérez Nava, para arquitectura Plenoptic 1.0. Por otra parte, implementamos el algoritmo de Renderizado por Bloques con y sin suavizado de Lumsdaine y Georgiev para arquitectura Plenoptic 2.0. Asimismo, trabajamos en el dominio frecuencial implementando el algoritmo de reenfoque basado en el Teorema del Corte de Fourier y en el mismo dominio, obtuvimos resultados con el algoritmo de Foco Volumétrico implementado por Dansereau.

Para generar imágenes propias y poder comparar los resultados con los autores arriba mencionados, desarrollamos un dispositivo experimental utilizando una cámara plenóptica Lytro F01, muñecos Lego y un fantoma de escalera para poder generar imágenes con objetos separados a diferentes distancias.

Los criterios adoptados para evaluar nuestras implementaciones fueron: tiempos de proceso, preservación de los atributos fotométricos de la escena y rangos de reenfoque.

Para los LF capturados con arquitectura Plenoptic 1.0, el algoritmo Corte del Espectro fue el más veloz con un tiempo promedio de reenfoque de 1.87 s, seguido por Desplazamiento-Suma con 2.17 s. El tiempo promedio de proceso de 31.66 s obtenido con el algoritmo Sesgado de la Radiancia, lo inhabilita para requerimientos de trabajo en tiempo real.

El algoritmo Desplazamiento-Suma fue el que preservó las características fotométricas de la escena, evaluadas mediante perfiles de intensidad contra posición para cada canal color RGB e histogramas también de cada canal, medidos sobre las imágenes de los muñecos. El algoritmo Sesgado de la Radiancia causó una leve disminución en intensidad, mientras que el algoritmo Corte del Espectro introdujo desbalances tanto en los canales de color como en la intensidad. Esto demuestra que los dos primeros algoritmos son útiles como herramientas de reenfoque en aplicaciones donde el color es un descriptor, en tanto que el último, no.

Obtuvimos resultados cualitativos de la calidad del enfoque posicionado en distintos rangos de profundidad de campo, basados en el análisis de bordes. Encontramos que los bordes en las imágenes obtenidas para evaluar las regiones en foco muestran claramente que las posiciones de dichas regiones van desplazándose a medida que cambiamos los parámetros de los algoritmos ensayados. Para el algoritmo Desplazamiento-Suma variamos el parámetro alfa desde 0.9 hasta 1.8, en pasos de 0.1, comprobando que valores de alfa inferiores a 0.1 no evidencian cambios entre dos imágenes reenfoçadas consecutivas. El algoritmo Sesgado de la Radiancia con el parámetro theta variando desde 0 hasta -4.5, en pasos de 0.5, si bien mostró el desplazamiento esperado de la región en foco, fue ensanchando la profundidad de campo a medida que se hacía más negativo. El algoritmo Corte del Espectro, ensayado con el parámetro beta variando desde -0.005 hasta 0.022, en pasos de 0.003, también mostró un comportamiento similar al algoritmo anterior.

El algoritmo Desplazamiento-Suma resultó ser el de mejor performance promedio para la arquitectura Plenoptic 1.0, con un tiempo de refocalización de 2.17 s para imágenes crudas de 16Mp, relación lineal del tiempo de cómputo con el tamaño de la imagen reenfoçada, preservación de los atributos fotométricos de color, brillo y contraste de la escena y el mejor comportamiento comparativo de rangos de enfoque.

Pese a las limitaciones de la cámara Lytro F01, pudimos capturar imágenes útiles para ser procesadas con nuestros algoritmos, de donde se infiere que para comenzar a dar los primeros pasos en el aprendizaje de esta técnica, esta cámara es apropiada.

Accedimos a imágenes crudas de 40Mp de arquitectura Plenoptic 2.0 disponibles en internet, que nos permitieron ensayar el algoritmo de Renderizado por Bloques, no aplicable para arquitectura Plenoptic 1.0. Con dichas imágenes procesadas por este algoritmo, obtuvimos reenfoque y vistas desde diferentes ángulos de manera simultánea, con tiempos promedio de proceso de 1.18 s.

Encontramos que las estrategias de vectorización para la paralelización de los bucles en la codificación del LF, incrementaron la velocidad de proceso en un factor de aproximadamente 1000 con respecto a la que se obtiene con bucles anidados.

Para la implementación de los algoritmos utilizamos MATLAB, pero una migración del código a plataformas de trabajo paralelo basadas en GPU permitirá obtener resultados en tiempo cuasi real.

Referencias

- [1] C. Wheatstone, “Contributions to the Physiology of Vision. Part the First. On Some Remarkable, and Hitherto Unobserved, Phenomena of Binocular Vision,” *Philos. Trans. R. Soc. London*, vol. 128, no. 0, pp. 371–394, Jan. 1838.
- [2] Marr, “Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.,” *Book*, 1982.
- [3] C. M. Martínez *et al.*, “Multi-view fundus camera.” Google Patents, 2014.
- [4] S. Baghdadchi *et al.*, “An innovative system for 3D clinical photography in the resource-limited settings,” *J. Transl. Med.*, vol. 12, no. 1, p. 169, 2014.
- [5] N. O. A. H. B. Edard, T. I. S. Hope, and A. L. H. Oberman, “Light field otoscope design for 3D in vivo imaging of the middle ear,” vol. 8, no. 1, pp. 9529–9534, 2017.
- [6] A. Shademan, R. S. Decker, J. Opfermann, S. Leonard, P. C. W. Kim, and A. Krieger, “Plenoptic cameras in surgical robotics: Calibration, registration, and evaluation,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 708–714.
- [7] H. Navarro Fructuoso, “Captura y reproducción de imágenes 3D mediante sistemas de Imagen Integral. Diseño, implementación y aplicaciones.,” Universitat de Valencia, 2014.
- [8] S. Tumu, V. Avasarala, S. T. Jonnalagadda, and P. Wadekar, “Monocular-Cues Based 3-D Reconstruction,” in *3-D Surface Geometry and Reconstruction*, no. 2009, IGI Global, 2009, pp. 181–196.
- [9] Peter Hohenstatt, *Leonardo da Vinci : 1452 - 1519*. Hamburg, Germany: Köln Könnemann, 1998.
- [10] Epelbaum S., “Historia de la Estereoscopia y sus aplicaciones,” *Arch. Oftalmol. B. Aires*, vol. 81, no. 2, pp. 62–67, 2010.
- [11] B. Julesz, “Binocular Depth Perception of Computer-Generated Patterns,” *Bell Syst. Tech. J.*, vol. 39, no. 5, pp. 1125–1162, Sep. 1960.
- [12] J. Adur and J. Diaz-Zamboni, “Digital deconvolution microscopy: Development, evaluation and utilization in 3D quantitative studies of E-cadherin expression in skin of Bufo arenarum tadpoles,” *Mod. Res. Educ. Top. Microsc.*, no. 120, pp. 906–916, 2007.
- [13] J. G. McNally, T. Karpova, J. Cooper, and J. A. Conchello, “Three-Dimensional Imaging by Deconvolution Microscopy,” *Methods*, vol. 19, no. 3, pp. 373–385, Nov. 1999.
- [14] M. Levoy, “Stanford Spherical Gantry,” 2002. [Online]. Available: <https://graphics.stanford.edu/projects/gantry/>. [Accessed: 06-May-2017].
- [15] MIT Media Lab: VisMod Group, “The Structure from Motion Task,” *3D Structure from 2D Motion*, 1999. [Online]. Available: <https://www.media.mit.edu>. [Accessed: 06-May-2017].
- [16] A. et al. Adams, “The Stanford Multi-Camera Array,” 2011. [Online]. Available: <https://graphics.stanford.edu/projects/array/>. [Accessed: 06-May-2017].
- [17] B. Wilburn *et al.*, “High performance imaging using large camera arrays,” in *ACM*

SIGGRAPH 2005 Papers on - SIGGRAPH '05, 2005, p. 765.

- [18] F. Ives, "A novel stereogram," *J. Franklin Inst.*, vol. 153, no. 1, pp. 51–52, 1902.
- [19] G. Lippmann, "Épreuves réversibles donnant la sensation du relief," *J. Phys. Théorique Appliquée*, vol. 7, no. 1, pp. 821–825, 1908.
- [20] G. Lippmann, "Épreuves réversibles. Photographies intégrales," *Comptes Rendus l'Académie des Sci.*, vol. 146, pp. 446–451, 1908.
- [21] G. Lippmann, "Integral Photography," *J. Soc. Fr. Phys*, vol. 69, 1912.
- [22] E. H. Adelson and J. Y. A. Wang, "Single lens stereo with a plenoptic camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 99–106, 1992.
- [23] G. Lippmann, "Integral Photography: a new discovery by Professor Lippmann," *Sci. Am.*, p. 164, 1911.
- [24] P. P. Sokolov, "Autostereoscopy and Integral Photography by Professor Lippmann's Method," *Moscow State Univ. Press*, 1911.
- [25] H. Ives, "Parallax panoramagrams made with a large diameter lens," *Josa*, vol. 20, no. 6, pp. 332–340, 1930.
- [26] H. Ives, "Optical properties of a Lippmann lenticulated sheet," *J. Opt. Soc. Am.*, vol. 21, pp. 171–176, 1931.
- [27] C. Kanolt, "Production of Stereoscopic Picture," 1934.
- [28] D. F. Coffey, "Apparatus for Making a Composite Stereograph," US Patent 2.063.985, 1936.
- [29] Y. A. Dudnikov, "Autostereoscopy and Integral Photography," *Opt. Tech.*, vol. 37, no. 7, 1970.
- [30] D. Roberts and T. Smith, "The history of integral print methods," *Lens Array Print Tech.*, pp. 1–21, 2003.
- [31] H. Navarro, A. Dorado, G. Saavedra, and M. M. Corral, "Three-Dimensional Imaging and Display through Integral Photography," *J. Inf. Commun. Converg. Eng.*, vol. 12, no. 2, pp. 89–96, Jun. 2014.
- [32] J. R. Adelson, Edward H and Bergen, "The Plenoptic Function and the Elements of Early Vision," *Comput. Model. Vis. Process. Cambridge, MA MIT Press*, pp. 3–20, 1991.
- [33] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. ACM SIGGRAPH*, 1996.
- [34] A. Gershun, "The light field. Translated by P. Moon and G. Timoshenko.," *J. Math. Physics*, vol. XVIII, pp. 51–151, 1936.
- [35] L. D. Stroebe, J. Compton, and I. B. Current, *Basic Photographic Materials and Processes*, Second. Boston. U. S.: Focal Press, 2000.
- [36] R. Ng, M. Levoy, G. Duval, M. Horowitz, and P. Hanrahan, "Light Field Photography with a Hand-held Plenoptic Camera," *Informational*, pp. 1–11, 2005.
- [37] R. Ng, "Digital light field photography," *Stanford Univ.*, pp. 1–203, 2006.

- [38] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *2009 IEEE International Conference on Computational Photography (ICCP)*, 2009, pp. 1–8.
- [39] A. Lumsdaine, "Focused plenoptic camera and rendering," *J. Electron. Imaging*, vol. 19, no. 2, p. 21106, Apr. 2010.
- [40] T. Georgiev, "Gallery and Lightfield Data," 2010. [Online]. Available: <http://www.tgeorgiev.net/Gallery>. [Accessed: 08-May-2017].
- [41] J. Keats, "The Sharpest - Shooting Camera," *Popular Science*, pp. 56–57, Apr-2007.
- [42] LightField Forum, "Adobe LightField Camera Prototypes," 2009. [Online]. Available: <http://lightfield-forum.com/light-field-camera-prototypes/adobe-lightfield-camera-prototypes/>. [Accessed: 09-May-2017].
- [43] C. Group, "CAFADIS: Plenoptic technology research.," 2017. [Online]. Available: <http://www.cafadis.uill.es/>. [Accessed: 09-May-2017].
- [44] LightField Forum, "CAFADIS LightField Lens Prototype," 2010. [Online]. Available: <http://lightfield-forum.com/light-field-camera-prototypes/cafadis-lightfield-lens-prototype/>. [Accessed: 09-May-2017].
- [45] LightField Forum, "Pelican Imaging Array Camera: Light Field Module for Smartphones," 2014. [Online]. Available: <http://lightfield-forum.com/light-field-camera-prototypes/pelican-imaging-array-camera-light-field-module-for-smartphones/>. [Accessed: 09-May-2017].
- [46] Disney Research, "David DiFrancesco," 2012. [Online]. Available: <http://www.disneyresearch.com/people/david-difrancesco/>.
- [47] LightField Forum, "Pixar Super LightField Lens," 2013. [Online]. Available: <http://lightfield-forum.com/light-field-camera-prototypes/pixar-super-lightfield-lens/>. [Accessed: 09-May-2017].
- [48] LightField Forum, "Toshiba LightField Camera Module," 2014. [Online]. Available: <http://lightfield-forum.com/light-field-camera-prototypes/toshiba-lightfield-camera-module/>. [Accessed: 09-May-2017].
- [49] LightField Forum, "Nvidia Near-Eye Light Field Display," 2014. [Online]. Available: <http://lightfield-forum.com/light-field-camera-prototypes/nvidia-near-eye-light-field-display/>. [Accessed: 09-May-2017].
- [50] LightField Forum, "Mems|Cam: Software Refocus for Smartphones without LightField Technology," 2013. [Online]. Available: <http://lightfield-forum.com/2013/03/memscam-software-refocus-for-smartphones-without-lightfield-technology/>. [Accessed: 09-May-2017].
- [51] Lytro, "Lytro, Inc," 2017. [Online]. Available: <https://lytro.com/>. [Accessed: 09-May-2017].
- [52] Raytrix GmbH, "3D VISION immersive 4D light field," 2017. [Online]. Available: <https://www.raytrix.de/>. [Accessed: 09-May-2017].
- [53] AVEGANT, "Introducing AVEGANT Light Field," 2017. [Online]. Available: <https://www.avegant.com/blog/company-news/introducing-avegant-light-field>. [Accessed: 09-May-2017].
- [54] Rachel Metz, "A More Realistic Augmented Reality," *MIT Technology Review*, 2017.

- [Online]. Available: https://www.technologyreview.com/s/604072/a-more-realistic-augmented-reality/?utm_campaign=add_this&utm_source=twitter&utm_medium=post. [Accessed: 09-May-2017].
- [55] U. of Michigan, “Nanoelectronics and Nanophotonics Lab,” 2017. [Online]. Available: <http://vhosts.eecs.umich.edu/zhonglab/>. [Accessed: 09-May-2017].
- [56] Katherine McAlpine, “A better 3D camera with clear, graphene light detectors,” *U-M Press Release*, 2016. [Online]. Available: <http://eecs.umich.edu/eecs/about/articles/2016/a-better-3d-camera-with-clear-graphene-light-detectors.html>. [Accessed: 09-May-2017].
- [57] LightField Forum, “New Light Field Tech to use Transparent Sensor Layers instead of Microlenses,” 2017. [Online]. Available: <http://lightfield-forum.com/2016/03/new-light-field-tech-to-use-transparent-sensor-layers-instead-of-microlenses/#more-9413>. [Accessed: 09-May-2017].
- [58] A. Lumsdaine, G. Chunev, and T. Georgiev, “Plenoptic rendering with interactive performance using GPUs,” 2012, p. 829513.
- [59] D. G. Dansereau, O. Pizarro, and S. B. Williams, “Decoding, calibration and rectification for lenselet-based plenoptic cameras,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013.
- [60] O. Johannsen, C. Heinze, B. Goldluecke, and C. Perwaß, “On the Calibration of Focused Plenoptic Cameras,” 2013, pp. 302–317.
- [61] Z. Zhang and M. Levoy, “LFDdisplay: a real-time system for light field microscopy,” 2008. [Online]. Available: <http://graphics.stanford.edu/software/LFDdisplay/>. [Accessed: 09-May-2017].
- [62] D. G. Dansereau, “Light Field Toolbox for MATLAB,” 2015. [Online]. Available: <http://dgd.vision/Tools/LFTtoolbox/>. [Accessed: 10-May-2017].
- [63] A. Lumsdaine and T. Georgiev, “Full resolution lightfield rendering,” 2008.
- [64] F. P. Nava and J. P. Lüke, “An $O(n^2 \log n)$ per plane fast discrete focal stack transform,” *Opt. 3D Meas. Tech.*, 2009.
- [65] F. P. Nava, J. G. Marichal-Hernández, and J. M. Rodríguez-Ramos, “The discrete focal stack transform,” *Proc. 16th Eur. Signal Process. Conf. (EUSIPCO 2008)*, 2008.
- [66] R. Ng, “Fourier slice photography,” *ACM Trans. Graph.*, vol. 24, no. 3, p. 735, Jul. 2005.
- [67] R. N. Bracewell, “Strip integration in radio astronomy,” *Aust. J. Phys.*, vol. 9, pp. 198–217, 1956.
- [68] L. T. Bruton, “Selective filtering of spatio-temporal plane waves using 3-D cone filter banks,” in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, 2001, pp. 67–70.
- [69] D. Dansereau and L. Bruton, “A 4D frequency-planar IIR filter and its application to light field processing,” in *Proceedings of the 2003 International Symposium on Circuits and Systems, 2003. ISCAS '03.*, vol. 4, p. IV-476-IV-479.
- [70] D. Dansereau and L. Bruton, “Gradient-based depth estimation from 4D light fields,” in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat.*

No.04CH37512), p. III-549-52.

- [71] D. Dansereau and L. T. Bruton, "A 4-D Dual-Fan Filter Bank for Depth Filtering in Light Fields," *IEEE Trans. Signal Process.*, vol. 55, no. 2, pp. 542–549, Feb. 2007.
- [72] D. G. Dansereau, "Plenoptic signal processing for robust vision in field robotics," The University of Sydney, 2014.
- [73] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Linear Volumetric Focus for Light Field Cameras," *ACM Trans. Graph.*, vol. 34, no. 2, pp. 1–20, Mar. 2015.
- [74] S. Nayar, "Shape from focus," *Cmu-Ri-Tr-89-27*, vol. 16, no. 8, pp. 824–831, 1989.
- [75] P. Favaro and S. Soatto, *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Springer-Verlag London, 2007.
- [76] P. Favaro, "Depth from focus/defocus," 2002. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FAVARO1/dfdutorial.html. [Accessed: 10-May-2017].
- [77] A. S. Malik, T. S. Choi, and H. Nisar, *Depth Map and 3D Imaging Applications: Algorithms and Technologies*, 1st Editio. IGI Global, 2011.
- [78] S. K. Nayar and Y. Nakagawa, "Shape from focus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 8, pp. 824–831, 1994.
- [79] S. K. Nayar, "Shape from focus system," in *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992, pp. 302–308.
- [80] P. Favaro and S. Soatto, "Learning shape from defocus," *Comput. Vision—ECCV 2002*, pp. 823–824, 2002.
- [81] P. Favaro, A. Mennucci, and S. Soatto, "Observing Shape From Defocused Images," *Int. J. Comput. Vis.*, vol. 52, no. 1, pp. 25–43, 2003.
- [82] P. Favaro and S. Soatto, "A geometric approach to shape from defocus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 406–417, 2005.
- [83] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from Combining Defocus and Correspondence Using Light-Field Cameras," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 673–680.
- [84] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from Combining Defocus and Correspondence Using light-Field Cameras," *Computer Graphics University of California - Berkeley*, 2013. [Online]. Available: <http://graphics.berkeley.edu/papers/Tao-DFC-2013-12/index.html>. [Accessed: 10-May-2017].
- [85] Lytro, "How can I take even better living pictures with my first generation Lytro camera?," *Lytro Support*, 2015. [Online]. Available: <https://support.lytro.com/hc/en-us/articles/200864430-How-can-I-take-even-better-living-pictures->. [Accessed: 10-May-2017].
- [86] T. Anderson *et al.*, "Stanford Light Field Microscope Project," *Stanford Computer Graphics Laboratory*, 2015. [Online]. Available: <http://graphics.stanford.edu/projects/lfmicroscope/>. [Accessed: 10-May-2017].
- [87] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using*

MATLAB, 2nd ed. McGraw-Hill Education, 2009.

- [88] Zhan Yu, Jingyi Yu, A. Lumsdaine, and T. Georgiev, “An analysis of color demosaicing in plenoptic cameras,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 901–908.
- [89] T. Georgiev, G. Chunev, and A. Lumsdaine, “Superresolution with the focused plenoptic camera,” 2011, p. 78730X.
- [90] T. G. Georgiev and A. Lumsdaine, “Superresolution with Plenoptic 2.0 Cameras,” in *Frontiers in Optics 2009/Laser Science XXV/Fall 2009 OSA Optics & Photonics Technical Digest*, 2009, p. STuA6.