

UNIVERSIDAD DE LA LAGUNA

TESIS DOCTORAL

---

Recuperación de la geometría de una  
escena a partir de imágenes plenópticas  
aplicando técnicas locales

---

*Autor:*  
Jonas Philipp Lüke

*Director:*  
Dr. Fernando Rosa González

2014





D. Fernando Rosa González, Doctor en Ciencias Físicas y Profesor Titular del área de conocimiento de “Teoría de la Señal y Comunicaciones”, perteneciente al Departamento de Física Fundamental y Experimental, Electrónica y Sistemas de La Universidad de La Laguna, como director

**HACE CONSTAR**

Que D. Jonás Philipp Lüke, con N.I.E. X-0581666L, alumno del programa de doctorado “Física e informática. Orientaciones: física de materiales, informática, tecnología del medio ambiente” de la Universidad de La Laguna, ha realizado bajo mi dirección la Tesis Doctoral titulada: “Recuperación de la geometría de una escena a partir de imágenes plenópticas aplicando técnicas locales”, para optar al grado de Doctor y

**AUTORIZA**

la presentación y defensa pública de la memoria de Tesis Doctoral en La Laguna, a 22 de enero de 2014.

El Director

Fernando Rosa  
González



# Prólogo

Mis estudios de doctorado se iniciaron en el curso 2006-2007 al incorporarme al programa de doctorado de Física e Informática de la Universidad de La Laguna. Ese curso comenzó con los cursos de doctorado y el año siguiente se realizaron los trabajos de investigación para la obtención del Diploma de Estudios Avanzados, que finalmente obtuve en 2008. Desde ese momento, me he encontrado en la fase de elaboración de la tesis doctoral que se ha extendido hasta este año, 2013.

Simultáneamente, a partir del año 2006, me uní al *Grupo de Tecnologías Integradas* de la Universidad de La Laguna en el que inicialmente desempeñé labores de becario y luego de contratado de investigación, asociadas al proyecto de investigación de bioacústica denominado “Creación de herramientas informáticas para la significación de la cultura vocal del *Orcinus Orca*”, financiado por Loro Parque Fundación.

A partir de 2008 me sumé al proyecto de desarrollo de la

cámara de fase-distancias (CAFADIS), que se lleva desarrollando en la Universidad de La Laguna desde el año 2005 y que ha dado lugar a dos patentes (Rodríguez Ramos et al. 2007; Rodríguez-Ramos et al. 2009a), compaginando ambos proyectos. Mi participación se ha concretado en varios proyectos del plan nacional de I+D+i, además de un proyecto con financiación autonómica, concedidos al grupo de investigación y que se han centrado principalmente en dos frentes.

Por un lado, la recuperación de la fase del frente de onda a partir de sensores plenópticos como CAFADIS, y por otro, la estimación de distancias a partir del *light field*. Personalmente, he estado más implicado en este segundo frente, colaborando con otros miembros del grupo en el desarrollo de algoritmos e implementándolos sobre unidades de procesamiento gráfico. Todos estos trabajos, así como los realizados en el proyecto de bioacústica, han dado lugar a varias comunicaciones a congresos, así como a publicaciones en revistas.

En el ámbito de la docencia universitaria, en 2010, tomé posesión de una plaza de profesor contratado del departamento de Física Fundamental y Experimental, Electrónica y Sistemas, por lo que tuve que repartir mi dedicación entre las actividades docentes e investigadoras.

Fue a partir de 2011, tras la valoración de las limitaciones de los modos existentes para obtener distancias de modo pasivo, que comenzamos a ver la necesidad de desarrollar nuevas técnicas de estimación de distancias, que resolviesen los problemas del método que habíamos desarrollado anteriormente. Uno de estos problemas es la enorme carga computacional de las técnicas globales. De modo que

se comenzó a explorar la posibilidad de utilizar técnicas locales, más directas y fácilmente paralelizables. Los frutos de ese trabajo se vieron sólo un año más tarde dando lugar a la esencia de los métodos que se presentan en esta tesis doctoral.



# Agradecimientos

Quiero expresar mi agradecimiento a Fernando Rosa González, mi director de tesis, por la ayuda, los consejos y, especialmente, las correcciones durante el desarrollo de este trabajo. También quiero agradecer a los miembros del Grupo de Tecnologías Integradas su buena disposición y el buen trabajo que han realizado durante estos años. Tampoco me puedo olvidar del resto de compañeros del Departamento de Física Fundamental y Experimental, Electrónica y Sistemas. Todos, sin duda, han contribuido al buen ambiente de trabajo que ha sido necesario para la realización de esta tesis.

Además, me gustaría agradecer a Loro Parque Fundación el hecho de financiar el proyecto “Creación de herramientas informáticas para la significación de la cultura vocal del Orcinus Orca” del que fui becario durante los primeros años de doctorado.

No hace falta decir que las personas que nos rodean nos dan la fuerza necesaria para ir salvando los obstáculos que nos pone la vida. Por eso, debo agradecer especialmente a mis padres y a mi hermano,

sin olvidar a mis amigos, la ayuda y el apoyo que me han prestado, tanto en los momentos buenos, como en los malos.

Sin embargo, esta tesis debo dedicársela a Cristina, con quién he compartido mi vida durante el tiempo que he tardado en elaborar este trabajo. Gracias Cristina, por ser como eres, por quererme, por apoyarme, por los amigurumis que aparecen en algunos *light fields* que se muestran en esta memoria, por leer y corregir el texto de esta memoria, y por muchas cosas más.

Jonas Philipp Lüke



# Resumen

En los últimos años, se han producido notables avances en la captura y procesamiento de imágenes con la finalidad de obtener la información tridimensional de la escena frente a la cámara. Para ello se utiliza el concepto de *light field*, una función de cuatro dimensiones que recoge la intensidad de los rayos de luz que atraviesan el espacio vacío en cada punto del mismo. Dicha función se puede capturar mediante diversos dispositivos. Una vez capturado, el *light field* permite sintetizar imágenes reenfocadas a distintas distancias o desde distintos puntos de vista. Además estos datos se pueden utilizar para recuperar la información tridimensional de la escena, ya que se preserva la información angular.

Esta tesis doctoral se ha centrado en el procesamiento del *light field* para la obtención de estimaciones de distancias mediante métodos locales.

Inicialmente, se expusieron los métodos empleados para obtener un conjunto de *light fields*, tanto reales como sintéticos, que permitieron

comprobar la eficacia del método de estimación de distancias propuesto. Seguidamente se desarrolló un nuevo el operador local para la medida de distancias a partir del *light field*, que constituye la contribución central de esta tesis doctoral. Dicho operador local se basa en la estimación de la orientación de los planos generados por los objetos de la escena en el *light field*, utilizando para ello la información que contienen los gradientes, teniendo en cuenta la naturaleza 4D del *light field*, así como la ligadura entre ambos vectores directores de estos planos.

Esta forma de proceder permite obtener una medida de distancia para cada rayo del *light field*, salvo en zonas con insuficiente textura. Como el objetivo era obtener un único mapa de distancias, se propuso un método para combinar los múltiples mapas obtenidos en un marco de referencia común.

Aún disponiendo de un número elevado de medidas, el mapa resultante de la combinación puede contener zonas vacías, sin medidas. Por ello, se propuso rellenar estas zonas utilizando el modelo TV-L1 optimizado mediante el método split Bregman. Dicho modelo ha sido aplicado anteriormente en la regularización de mapas de distancias y el método split Bregman también es un método conocido para optimizar el modelo TV-L1. Sin embargo, la aplicación de este último a la regularización de mapas de distancias no es conocida y aporta la ventaja de su fácil paralelización e implementación sobre GPU. También se han explorado algunas variantes del método para aliviar el cálculo, utilizando una técnica de jerarquización en el caso de imágenes estáticas y, por otro lado, encadenando resultados de *frames* sucesivos

en el procesamiento de secuencias temporales.

Las pruebas, realizadas sobre distintos conjuntos de datos, revelan que el método propuesto mejora a otros métodos de estimación de distancias a partir de *light fields*, tanto en precisión como en tiempos de ejecución, existiendo aún posibilidad de mejora en estos últimos. Además, el método permite obtener la información de la distancia utilizando un número reducido de muestras angulares, sin que esto afecte significativamente a la precisión de los resultados. Por último, la complejidad computacional del método propuesto tampoco tiene dependencia de la resolución de la variable distancia.



# Abstract

In recent years, there have been significant advances in the capture and processing of images in order to obtain the 3D information about the scene in front of the camera. For this purpose, the concept of light field is used. The light field is a four dimensional function which grabs the intensity of light rays traversing the empty space. This function can be captured by several devices. Once captured, the light field may be used to synthesize images refocused at different depths or images from different viewpoints. Furthermore, these data can be used to retrieve three-dimensional information, since the angular information is kept.

This dissertation focuses on the processing of the light field to obtain depth estimates using local methods.

Initially, we show the methods employed to obtain a set of real and synthetic light fields, which allowed to test the effectiveness of the proposed depth estimation method. Then a novel local operator for measuring depths for was proposed. This operator is the main contribution of this thesis. The local operator is based on estimating the orientation of the planes generated by the objects of the scene in the light field. For this purpose, the information contained in the

gradients is used, taking into account the 4D nature of the light field, as well as the relation between the direction vectors of these planes.

This approach allows to obtain a depth measure for each ray of the light field, except in areas with insufficient texture. Since the goal was to obtain a single depth map, a method was proposed to combine the multiple maps into one map.

Even by merging a large number of measurements, the map resulting from the combination may contain void areas, without measures. Therefore, it was proposed to fill these areas using the TV-L1 model optimized by split Bregman method. This model has been applied earlier on depth map regularization and split Bregman method is also a known method to optimize the TV-L1 model. But the application of the latter to depth map regularization is not known and it has advantages for implementation and parallelization on GPU. Also some variants of the method for reducing computations were proposed. In the case of static images a hierarchical implementation was proposed and in the case of temporal sequences the reuse of the results of previous frames was explored.

The tests performed over different datasets show that the proposed method improves other methods which estimate depths from light fields. The improvement affects both, accuracy and execution times. The latter can still be improved. Furthermore, the method allows to obtain depth information using a reduced number of angular samples, without a significant effect on accuracy. Finally, the computational complexity of the proposed method does not depend on the resolution of the depth variable.

# Índice general

|   |          |
|---|----------|
| <b>1. Introducción</b>  | <b>1</b> |
| <b>2. Antecedentes</b>  | <b>9</b> |
| 2.1. <i>Light field</i> o función plenóptica . . . . .            | 10       |
| 2.1.1. Rotaciones y subespacios del espacio plenóptico            | 16       |
| 2.2. Dispositivos de adquisición del <i>light field</i> . . . . . | 19       |
| 2.2.1. Arrays de múltiples cámaras . . . . .                      | 19       |
| 2.2.2. Multiplexación temporal . . . . .                          | 21       |
| 2.2.3. Multiplexación espacial . . . . .                          | 22       |
| 2.2.4. Multiplexación en frecuencia . . . . .                     | 26       |
| 2.2.5. Compressive sensing . . . . .                              | 26       |
| 2.3. Repositorios de <i>light fields</i> . . . . .                | 27       |
| 2.4. Reconstrucción de distancias a partir del light field . .    | 27       |
| 2.5. Estimación de orientaciones . . . . .                        | 31       |
| 2.5.1. Estimación de gradientes . . . . .                         | 31       |

|   |           |
|---|-----------|
| 2.5.2. Detección de estructuras lineales en imágenes<br>epipolares . . . . .                          | 33        |
| 2.6. Regularización de mapas de distancias . . . . .  | 36        |
| 2.7. Fotografía a partir del <i>light field</i> . . . . .   | 40        |
| 2.8. Hardware gráfico (GPU) . . . . .   | 48        |
| <b>3. Medida y simulación de <i>light fields</i></b>  | <b>55</b> |
| 3.1. Adquisición mediante multiplexación temporal . . . . .   | 56        |
| 3.2. Light fields obtenidos mediante la cámara CAFADIS . . . . .                                      | 60        |
| 3.2.1. Descripción del montaje óptico . . . . .   | 60        |
| 3.2.2. Relación entre pendientes y distancias . . . . .   | 64        |
| 3.2.3. Descripción de los <i>light fields</i> adquiridos . . . . .                                    | 66        |
| 3.3. <i>Light fields</i> obtenidos mediante el simulador de escenas<br>virtuales (SIDEV) . . . . .    | 68        |
| 3.4. Conjunto de datos del <i>Heidelberg Collaboratory for<br/>        Image Processing</i> . . . . . | 73        |
| <b>4. Estimación de distancias mediante un detector de<br/>pendientes en 4D</b>                       | <b>77</b> |
| 4.1. Detección de estructuras planas en cuatro dimensiones  | 79        |
| 4.2. Análisis de errores . . . . .  | 86        |
| 4.2.1. Estimación de orientación sobre las imágenes<br>epipolares . . . . .                           | 88        |
| 4.2.2. Estimación de la orientación en el <i>light field</i> 4D                                       | 92        |
| 4.2.3. Comparativa entre ambos operadores . . . . .   | 98        |
| 4.3. Establecimiento de umbrales sobre la medida de<br>confianza . . . . .                            | 99        |



|   |            |
|---|------------|
| 4.4. Agregación de las medidas locales . . . . .  | 100        |
| 4.5. Detalles de implementación . . . . .   | 106        |
| <b>5. Estimación de mapas de distancias densos</b>  | <b>113</b> |
| 5.1. Método Split Bregman para el modelo TV-L1 . . . . .                                      | 114        |
| 5.2. Implementación jerárquica de grueso a fino . . . . .                                     | 118        |
| 5.3. Aplicación a secuencias temporales . . . . .   | 122        |
| 5.4. Detalles de implementación . . . . .   | 126        |
| <b>6. Resultados</b>  | <b>129</b> |
| 6.1. Imágenes adquiridas mediante multiplexación temporal                                     | 130        |
| 6.2. Imágenes adquiridas mediante la cámara CAFADIS . .                                       | 134        |
| 6.3. Escena de los cubos . . . . .  | 137        |
| 6.4. Conjunto de datos del <i>Heidelberg Collaboratory for<br/>Image Processing</i> . . . . . | 138        |
| 6.4.1. Aplicando sólo el operador local en cuatro<br>dimensiones . . . . .                    | 140        |
| 6.4.2. Regularización TV-L1 . . . . .   | 144        |
| 6.5. Discusión . . . . .  | 148        |
| <b>7. Conclusiones</b>  | <b>155</b> |
| <b>Apéndices</b>  | <b>157</b> |
| <b>A. Formación de imágenes y su relación con el <i>light field</i></b>                       | <b>159</b> |
| <b>B. Cálculo de errores en la estimación de distancia</b>                                    | <b>163</b> |

|  |            |
|--|------------|
| B.1. Cálculo errores aleatorios en la estimación de la distancia a partir de imágenes epipolares . . . . . | 164        |
| B.2. Cálculo de errores en la estimación de distancia en el light field 4D . . . . .                       | 167        |
| <b>Bibliografía</b>  | <b>171</b> |

# Índice de figuras

|  |    |
|--|----|
| 2.1. Parametrizaciones de la función plenóptica. . . . .   | 12 |
| 2.2. Parametrización del <i>light field</i> mediante dos planos. . .   | 13 |
| 2.3. Relación entre la profundidad de un punto y la<br>pendiente de la línea generada en el <i>light field</i> . . . . . | 14 |
| 2.4. Rotaciones del espacio plenóptico. . . . .  | 17 |
| 2.5. Ejemplos de imágenes epipolares. . . . .  | 18 |
| 2.6. Ejemplos de arrays de cámaras. . . . .  | 20 |
| 2.7. Ejemplos de pórticos de cámaras. . . . .  | 21 |
| 2.8. Ejemplos de cámaras con multiplexación espacial. . . .  | 23 |
| 2.9. Proceso de mapeo de las coordenadas de un plano<br>imagen al espacio de rayos. . . . .                              | 42 |
| 2.10. Reparametrización dinámica de rayos para dos<br>superficies focales. . . . .                                       | 43 |
| 2.11. Reenfoque digital aplicando el método del <i>slice de<br/>Fourier</i> (Extraído de Ng et al. (2005)) . . . . .     | 45 |

|  |    |
|--|----|
| 2.12. Capacidad de cómputo en GFLOP/s para la CPU y la GPU. . . . .  | 51 |
| 3.1. Esquema del montaje óptico de la multiplexación temporal. . . . .   | 57 |
| 3.2. Fotografía del montaje utilizado para adquirir <i>light fields</i> mediante multiplexación temporal. . . . .  | 58 |
| 3.3. Imágenes centrales de los tres <i>light fields</i> adquiridos por multiplexación temporal. . . . .  | 59 |
| 3.4. Esquema del montaje óptico de la cámara plenóptica. . . . .   | 61 |
| 3.5. <i>Light field</i> obtenido con la cámara CAFADIS. . . . .  | 68 |
| 3.6. Interfaz gráfica del simulador de cámara plenóptica . . . . .   | 69 |
| 3.7. Esquema de un array de cámaras <i>pin-hole</i> equivalente a la cámara plenóptica . . . . .   | 72 |
| 3.8. <i>Light field</i> obtenido con el SIDEV. . . . .   | 74 |
| 3.9. Imagen central y mapas de distancias para los <i>light fields</i> sintéticos del HCI dataset. . . . .   | 75 |
| 3.10. Imagen central y mapas de distancias para los <i>light fields</i> reales del HCI dataset. . . . .  | 75 |
| 4.1. Representación de los errores sistemáticos del estimador de orientación 2D en función de la pendiente y la frecuencia. . . . .  | 89 |
| 4.2. Desviación típica del error aleatorio en la estimación de la pendiente en imágenes bidimensionales en función de la desviación típica de el ruido de la imagen. . . . . | 91 |

|  |     |
|--|-----|
| 4.3. Desviación típica del error aleatorio en la estimación de la pendiente en imágenes bidimensionales en función de $\zeta$ . . . . .  | 92  |
| 4.4. Desviación típica del error aleatorio en la estimación de la pendiente en imágenes bidimensionales en función de la pendiente a estimar. . . . .                            | 93  |
| 4.5. Errores sistemáticos para el operador local en cuatro dimensiones . . . . .   | 94  |
| 4.6. Desviación típica del error aleatorio en la estimación de la pendiente frente a la desviación típica del ruido en la imagen para el operador en cuatro dimensiones. . . . . | 95  |
| 4.7. Desviación típica del error aleatorio en la estimación de la pendiente frente a $\zeta$ para el operador en cuatro dimensiones. . . . .                                     | 96  |
| 4.8. Desviación típica del error aleatorio en la estimación de la pendiente frente a la pendiente para el operador en cuatro dimensiones. . . . .                                | 97  |
| 4.9. Cuadrado de la confianza $C(\theta)^2$ frente al cociente $\frac{\zeta}{\sigma_L}$ . . . . .  | 100 |
| 4.10. Múltiples mapas de distancias obtenidos estimando la orientación en 4D. . . . .  | 102 |
| 4.11. Representación de las medidas mediante nubes de puntos. . . . .  | 103 |
| 4.12. Mapa de distancias verdadero generado con el simulador . . . . .   | 105 |
| 4.13. Mapas de distancia obtenidos mediante distintos métodos de agregación . . . . .  | 109 |

|  |     |
|--|-----|
| 4.14. Diagrama de flujo de datos entre los <i>kernels</i> ejecutados en la implementación en CUDA. . . . .   | 110 |
| 4.15. Tiempos de ejecución sobre GPU para la obtención de un mapa de distancias disperso a partir del <i>light field</i> . . . . .                         | 111 |
| 5.1. Resultados obtenidos en los distintos niveles de la jerarquía para la escena de los cubos con 12 iteraciones en cada nivel. . . . .                   | 121 |
| 5.2. Resultado obtenido para la escena de los cubos realizando 36 iteraciones sin jerarquizar. . . . .   | 122 |
| 5.3. Función de costo en cada <i>frame</i> en función del número de iteraciones para el procesamiento independiente y el procesamiento encadenado. . . . . | 124 |
| 5.4. Mapas de pendientes regularizados obtenidos para el <i>frame</i> 21 de la secuencia de los cubos. . . . .   | 126 |
| 5.5. Tiempos de ejecución por iteración de split Bregman en función del tamaño del mapa de salida en millones de píxeles. . . . .                          | 128 |
| 6.1. Resultados obtenidos para la escena <i>ovillos</i> . . . . .  | 131 |
| 6.2. Resultados obtenidos para la escena <i>flores3</i> . . . . .  | 132 |
| 6.3. Resultados obtenidos para la escena <i>tortuga</i> . . . . .  | 133 |
| 6.4. Resultados obtenidos para la escena <i>muñecos-24</i> . . . . .   | 135 |
| 6.5. Resultados obtenidos al aumentar la escala $\times 7$ para la escena <i>muñecos-15</i> . . . . .  | 136 |
| 6.6. Mapas de distancias regularizados para la escena de los cubos. . . . .  | 139 |

|       |   |     |
|-------|---|-----|
| 6.7.  | Resultados obtenidos sobre las imágenes sintéticas del HCI dataset, aplicando directamente el operador local propuesto. . . . . | 142 |
| 6.8.  | Resultados obtenidos sobre las imágenes reales del HCI dataset, aplicando el operador local propuesto . . . . .                 | 143 |
| 6.9.  | Resultados obtenidos sobre las imágenes sintéticas del HCI dataset, aplicando la regularizacion TV-L1. . . . .                  | 146 |
| 6.10. | Resultados obtenidos sobre las imágenes reales del HCI dataset, aplicando la regularizacion TV-L1. . . . .                      | 147 |
| A.1.  | Esquema del proceso de formación de imagen de una lente delgada. . . . .  | 161 |





# Índice de tablas

|  |     |
|--|-----|
| 3.1. Dimensiones y condiciones de adquisición de los <i>light fields</i> adquiridos con multiplexación temporal. . . . .               | 59  |
| 3.2. Descripción de los parámetros ópticos de la cámara plenóptica . . . . .   | 62  |
| 3.3. Características de las microlentes utilizadas en el montaje de la cámara plenóptica. . . . .                                      | 67  |
| 3.4. Características de la lente principal utilizada en el montaje de la cámara plenóptica. . . . .                                    | 67  |
| 3.5. Parámetros de la cámara plenóptica simulada. . . . .  | 73  |
| 3.6. Dimensiones de los mapas de pendientes del HCI Dataset, junto con la pendiente máxima y mínima en cada uno. . . . .               | 76  |
| 6.1. Parámetros empleados en la ejecución del algoritmo sobre las escenas adquiridas con el método de multiplexación temporal. . . . . | 130 |

|  |     |
|--|-----|
| 6.2. Parámetros empleados en la ejecución del algoritmo sobre las escenas adquiridas con CAFADIS. . . . .                      | 137 |
| 6.3. Métricas de error para la escena de los cubos. . . . .  | 138 |
| 6.4. Medidas de error para el conjunto de datos del HCI aplicando el operador local en cuatro dimensiones propuesto. . . . .   | 141 |
| 6.5. Parámetros del algoritmo split Bregman al ejecutarlo sobre los datos del HCI dataset. . . . .                             | 145 |
| 6.6. Medidas de error para el conjunto de datos del HCI regularizando con TV-L1. . . . .                                       | 145 |
| 6.7. Error cuadrático medio multiplicado por 100 optimizando los parámetros del algoritmo para cada conjunto de datos. . . . . | 148 |
| 6.8. Clasificación de algoritmos de estimación de distancias de Wanner et al. (2013) . . . . .                                 | 153 |

# Capítulo 1

## Introducción

*“The recipe for great work is: very exacting taste, plus the ability to gratify it.”*

---

Paul Graham

Podría decirse que el deseo o la necesidad de representar el mundo es casi tan antiguo como la humanidad misma. Desde la prehistoria los seres humanos han tratado de representar partes de su mundo, tal como atestiguan las pinturas rupestres. Pero hasta la aparición de las primeras técnicas fotográficas a principios del siglo XIX, esto requería la mano experta de un artista que dibujaba, pintaba o esculpía su visión particular de la realidad. Con la aparición de la técnica fotográfica surge la posibilidad de captar de forma gráfica y duradera la realidad, “tal y como es”, utilizando para ello la luz que reflejan o emiten los objetos. Sin embargo, una fotografía sólo capta un instante

y únicamente dos de las tres dimensiones del espacio, por lo que sigue siendo, de alguna manera, una representación incompleta. A finales del siglo XIX, con el surgimiento del cine se añade el tiempo como una nueva dimensión.

A principios del siglo XX es cuando aparece la posibilidad de proyectar en tres dimensiones, lo que junto con la aparición de las películas en color permitiría observar una grabación del mundo como si fuese el mundo mismo. Sin embargo, el éxito comercial de cine 3D inicialmente es limitado.

A partir de los años ochenta, con el surgimiento de las tecnologías digitales y el impresionante desarrollo de la informática, se han abierto nuevas posibilidades tecnológicas, que han permitido gestionar de un modo distinto los datos capturados y que en los últimos años han permitido introducir el cine tridimensional y la televisión 3D en los hogares. En la actualidad, los gráficos por ordenador permiten crear contenidos 3D de apariencia casi real, sin la necesidad de que estos sean capturados de la realidad. Esto se consigue mediante la creación de un modelo del mundo que contiene la geometría de los objetos así como las propiedades de sus materiales, tales como color, textura, brillo, viscosidad, etc. Ese modelo permite generar tantas imágenes del mundo como sean necesarias e incluso permite interactuar con el mismo, como sucede en el caso de los juegos de ordenador.

Sin embargo, la creación de este tipo de modelos es un proceso complejo que podría simplificarse si dispusiésemos de instrumentos que pudiesen captar representaciones tridimensionales a partir de la

---

realidad. Ya no se trataría únicamente de capturar imágenes, sino de capturar la geometría tridimensional junto con las propiedades cromáticas de los objetos, emulando la escultura, lo cual supondría un avance comparable al que se produjo con el surgimiento de la fotografía, que emulaba la pintura. Con este tipo de instrumentos podrían mezclarse modelos capturados de la realidad con modelos sintéticos, o modificar los modelos capturados de la realidad.

Es fácil pensar como se beneficiaría la industria del entretenimiento, pero no sería la única beneficiada. Este tipo de avances también suponen un impacto significativo sobre otros campos, como son la robótica, la automoción o la aeronáutica, donde la posibilidad de generar un modelo suficientemente preciso del entorno permitiría automatizar parte de la navegación de un robot o nave no tripulada. En campos como la arquitectura o la arqueología se podrían obtener réplicas virtuales de objetos de valor, edificios, monumentos, incluso yacimientos arqueológicos o ciudades enteras. En el amplio campo de la industria, medir con precisión una pieza es fundamental, y tiene una enorme relevancia en los procesos industriales de control de calidad. Sin embargo, la generación de modelos tridimensionales es también de interés para la medicina, ya que los modelos del cuerpo humano pueden facilitar diagnósticos o robotizar intervenciones quirúrgicas.

Durante las últimas décadas se han realizado numerosos avances para crear sistemas que permitan generar modelos precisos, existiendo numerosas soluciones aplicables para fines concretos en ámbitos

específicos. Pero a día de hoy, este problema no se puede considerar resuelto. Se trata de un problema de gran complejidad cuya solución se puede abordar mediante métodos activos o pasivos. En el caso de los métodos activos, el sistema de visión 3D introduce en la escena información con la que interactúa con los objetos presentes, al contrario que en los sistemas pasivos, en los que se utiliza la luz presente en la escena y reflejada por los objetos, para obtener información sobre la geometría. No siempre es posible utilizar sistemas activos, como es el caso de iluminar con un láser un depósito de combustible inflamable.

El problema que se ha abordado en esta tesis doctoral es la estimación de distancias a partir de la adquisición de *light fields* obtenidos por diferentes medios pasivos. La solución al problema de estimación de distancias es la obtención de la tercera coordenada, que sirve como paso previo a la construcción de un modelo del mundo.

Tradicionalmente, el problema de estimación de distancias de forma pasiva, se ha abordado empleando técnicas de estereovisión que se basan en la captura de imágenes procedentes de dos cámaras alineadas y calibradas convenientemente. Una vez capturados los pares de imágenes, se debe resolver el problema de correspondencia consistente en identificar un mismo punto en ambas imágenes. Conocida la correspondencia de los puntos, la tercera coordenada se puede obtener por medio de un cálculo sencillo. Una evolución del método del estéreo es la multiestereovisión, que consiste en utilizar más de dos cámaras para la captura de datos. La característica común es

---

que la separación entre las distintas cámaras es relativamente grande, esto es, del orden de los centímetros, lo que condiciona los métodos empleados para resolver el problema de correspondencia. También cabe mencionar que el manejo de estos sistemas no es igual de sencillo que el de una cámara convencional debido a su tamaño y su peso.

En los últimos años, ha cobrado importancia otro dispositivo, la cámara plenóptica, que permite capturar datos con información tridimensional con una cámara de un único cuerpo. Estas cámaras capturan imágenes plenópticas que contienen información multivista. Sin embargo, la distancia entre puntos de vista es del orden unos cientos de micras y la resolución de cada una de las vistas es menor que la de una fotografía convencional, al compartir todas el mismo sensor. Esto ha suscitado gran interés y ha dado lugar al desarrollo de numerosas técnicas para el tratamiento de los datos adquiridos, sobre todo a partir del año 2005. Tanto es así, que actualmente existen dos versiones comerciales de este dispositivo.

Las imágenes plenópticas permiten la síntesis de imágenes enfocadas a distintas distancias después de haber tomado la imagen, para lo que se han desarrollado varias técnicas. La síntesis de imágenes a partir de los datos de la imagen plenóptica se conoce como fotografía computacional y consiste en el uso de nuevos sensores que retrasan el proceso de formación de imagen para llevarlo a cabo computacionalmente.

Existen técnicas de estimación de distancias a partir de imágenes plenópticas que utilizan un conjunto imágenes enfocadas en distintos

planos, a partir de las que se obtiene una estimación de la distancia. Para cada píxel de las imágenes reenfocadas se trata de estimar en cual de ellas está más enfocado y asignar la distancia de enfoque correspondiente a ese píxel en el mapa de distancias. Sin embargo, esto implica que previamente a la estimación de distancias, hay que computar las imágenes reenfocadas, y hacerlo eficientemente es un reto desde el punto de vista computacional, debido a la enorme cantidad de datos a tratar. Esta forma de proceder, además, tiene la desventaja de que el cálculo eficiente de imágenes reenfocadas da lugar a un conjunto discreto de planos de enfoque que, dependiendo de la configuración óptica, puede ser más o menos reducido, limitando la resolución en distancia. Además, la síntesis de imágenes, y por tanto de mapas de distancias, con un tamaño similar a las obtenidas por una cámara convencional, tampoco está exento de problemas y hacerlo de forma satisfactoria requiere conocer las distancias que en este caso son el resultado final a estimar.

Los antecedentes de la investigación realizada durante el desarrollo de esta tesis doctoral se encuentran expuestos de forma detallada en el capítulo 2. A continuación, en el capítulo 3, se explicarán los procedimientos y materiales empleados para generar de los *light fields* necesarios para determinar la eficacia del método propuesto. En el capítulo 4, se abordará el problema de estimación de distancias, en el que se centrará esta tesis doctoral. No se tratará el problema de la síntesis de imágenes a partir del *light field*, ni ningún otro aspecto de la fotografía computacional. Uno de los objetivos es abordar el problema de forma directa, sin tener que pasar por el cálculo de imágenes



---

reenfocadas, estimando las distancias a partir de los datos de la imagen plenóptica. Además, las técnicas antes mencionadas requieren un gran número de datos para producir un valor de distancia, por lo que se tratará de abordar la estimación de forma local utilizando únicamente los datos en el entorno del punto para calcular una medida de distancia.

Un *light field* contiene una gran cantidad de datos, pudiendo superar la decena de megapíxeles, haciendo imprescindible la aplicación de computación masivamente paralela para conseguir unas velocidades de procesamiento aceptables. Esto refuerza los argumentos a favor de una técnica local que además, por la propia naturaleza de los algoritmos involucrados, es sencilla de paralelizar. Sin embargo, hay que aclarar que el objetivo final no es la implementación óptima sobre hardware paralelo, sino el desarrollo de un método que facilite esta implementación.

Como ya se ha indicado, las distancias obtenidas pueden ser de utilidad en varios campos. Los requerimientos de exactitud y precisión de la medida realizada pueden variar enormemente según la aplicación, lo que obliga a estudiar las posibles fuentes de error que puedan afectar al método utilizado para determinar si es o no apto para cada aplicación.

El uso de técnicas locales sobre la imagen plenóptica puede dar lugar a múltiples medidas de distancia con redundancia para el mismo punto del mapa de distancias final. Para utilizar estas redundancias será necesario agregar o fusionar las distintas medidas correspondientes al mismo punto. Este aspecto no puede ser eludido y,

aunque no es el objetivo principal, también se abordará en esta tesis doctoral.

Es fácil comprender que, dependiendo de la escena, no es posible recuperar información tridimensional para todos los puntos a partir de una imagen plenóptica. Considérese, por ejemplo, el caso de una escena consistente en una pared completamente blanca, sin ningún tipo de textura ni rugosidad, y cuyos bordes no son visibles a la cámara. En este caso, ni siquiera el cerebro humano podría determinar a que distancia está la pared. Para el caso de zonas de la imagen en las que no se puede determinar de forma directa una medida de distancia se debe llevar a cabo una inferencia a partir de las medidas cercanas, siempre teniendo en cuenta que los valores inferidos nunca serán medidas reales, sino tan sólo inferencias. El proceso de inferencia, que se abordará en el capítulo 5, se puede llevar a cabo mediante diversas técnicas, que generalmente implican un modelo de la escena por medio de conocimiento previo, como por ejemplo una restricción sobre la suavidad de las superficies. En esta tesis se aplicará un modelo conocido y un método de inferencia para el relleno de los huecos en los que no se dispone de una medida fiable, teniendo nuevamente en cuenta nuestra preferencia por que sea viable su paralelización.

Los resultados obtenidos al aplicar el método local propuesto para la obtención de distancias a partir de los *light field* adquiridos con los métodos de captura propuestos en el capítulo 3, se muestran en el capítulo 6. Finalmente, en el capítulo 7, se recogerán las conclusiones que se han obtenido.

# Capítulo 2

## Antecedentes

*“Oigo y olvido; veo y recuerdo; hago y comprendo”*

---

Confucio

Comenzaremos describiendo los antecedentes de la investigación que se presenta en esta memoria de tesis doctoral. Los objetivos propuestos en el capítulo anterior se basan en la realización de medidas pasivas de distancias a partir de *light fields*. Es por ello que primero se introduce la función plenóptica y el concepto de *light field* con el fin de definir el tipo de datos de entrada que alimentarán a los algoritmos desarrollados en los capítulos posteriores. Una vez definido el concepto de *light field*, se pasa a describir las diferentes técnicas y dispositivos que permiten capturar este tipo de datos. Además, se

indican distintos repositorios en los que se pueden encontrar datos que pueden usarse como referencia. A continuación, se revisan los trabajos previos relacionados con la reconstrucción de distancias a partir del *light field* y se describe la técnica de detección de orientaciones en 2D que inspira la técnica propuesta en esta tesis doctoral. Seguidamente, se revisan las técnicas de regularización de mapas de distancias. Tras lo cual se repasan los métodos para crear imágenes a partir del *light field* y, finalmente, se introducen las unidades de procesamiento gráfico como hardware paralelo sobre el que implementar los métodos presentados.

## 2.1. *Light field* o función plenóptica

El concepto de *light field* se refiere a la función que describe la cantidad de luz que atraviesa un punto del espacio tridimensional en todas las direcciones. Este término fue acuñado por primera vez por Gershun (1936). Sin embargo, fue en el trabajo de Adelson y Bergen (1991) donde se introdujo este concepto en el campo de la visión por computador con la denominación de función plenóptica. Más tarde, en Levoy y Hanrahan (1996) utilizan el nombre de *light field*, para un caso concreto de la función plenóptica, mientras que en Gortler et al. (1996) lo denominan *lumigraph*.

Para describir la luz que viaja por el espacio libre de una escena, en Adelson y Bergen (1991) emplean una función de siete dimensiones denotada con  $P(\Theta, \Phi, \lambda, t, V_x, V_y, V_z)$ , donde  $(\Theta, \Phi)$  son los ángulos de

incidencia de los rayos de luz sobre el punto de observación, cuyas coordenadas son  $(V_x, V_y, V_z)$ . La variable  $t$  indica el instante de tiempo y  $\lambda$  es la longitud de onda de la luz incidente. Éstas dos últimas se suelen eliminar para simplificar la notación y se explicitan sólo cuando es necesario.

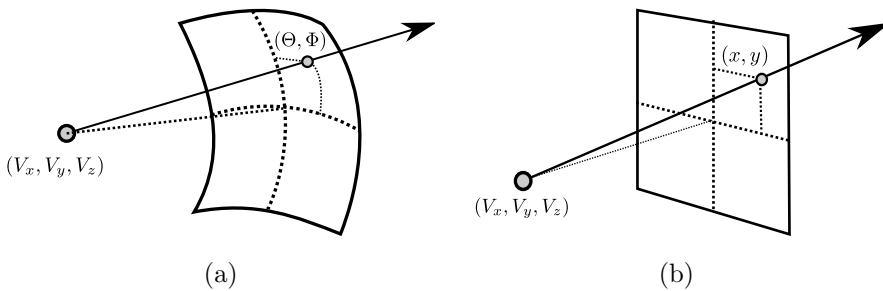
Tal y como indican Adelson y Bergen en su artículo, la función plenóptica es un concepto ideal y no es posible especificarla completamente para una escena real, ya que no es posible medir la luz que pasa a través de todos los puntos del espacio, en todas las direcciones, en todos los instantes de tiempo y en todo el espectro. Los objetos de una escena tridimensional iluminados con la luz presente, rellenan el espacio libre mediante un patrón de rayos que constituyen la función plenóptica y el observador toma muestras de la misma. De este modo, la función plenóptica sirve de vínculo entre los objetos y sus imágenes, aspecto que se describe detalladamente en el apéndice A.

Si bien la función plenóptica puede parecer complicada al ser una función de siete dimensiones, es una función estructurada que contiene un alto grado de redundancia. Su estructura está relacionada con la geometría de los objetos de la escena y con las propiedades ópticas de los materiales que los componen. La detección de estas estructuras a partir de las muestras de la función plenóptica permite, al menos en ciertos casos, distinguir y extraer características geométricas de los objetos. Esta idea también fue introducida por Adelson y Bergen al definir las tablas periódicas de la visión temprana (*early vision*).

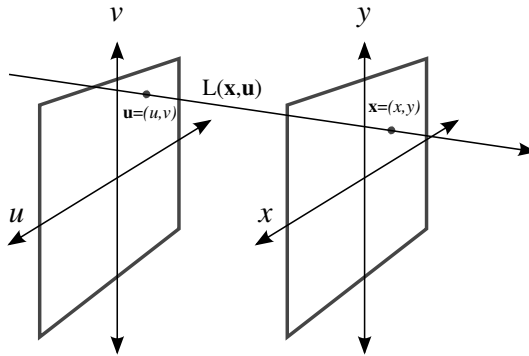
La mayoría de los métodos pasivos de visión 3D emplean un

subconjunto de información contenida en la función plenóptica, ya sean los métodos de estéreo-visión, multiestéreo-visión, *depth from focus* o *depth from video*. Esto hace que la función plenóptica sirva como marco común a todas las técnicas de visión tridimensional. El subconjunto de información utilizado depende del método empleado y la selección de la información tiene lugar por medio de la selección del patrón de muestreo.

Para poder diseñar algoritmos que utilicen los datos de la función plenóptica habrá que llevar a cabo una adecuada discretización, así como una parametrización de la misma. La discretización está ligada al montaje óptico que se utiliza para adquirirla. Además, los rayos que atraviesan un determinado punto  $(V_x, V_y, V_z)$  se pueden parametrizar por medio de coordenadas angulares o en coordenadas cartesianas como se muestra en la figura 2.1.



**Figura 2.1:** *Parametrizaciones de la función plenóptica: (a) Parametrización del rayo que pasa por un punto utilizando coordenadas angulares; (b) Parametrización del rayo que pasa por un punto utilizando coordenadas cartesianas.*

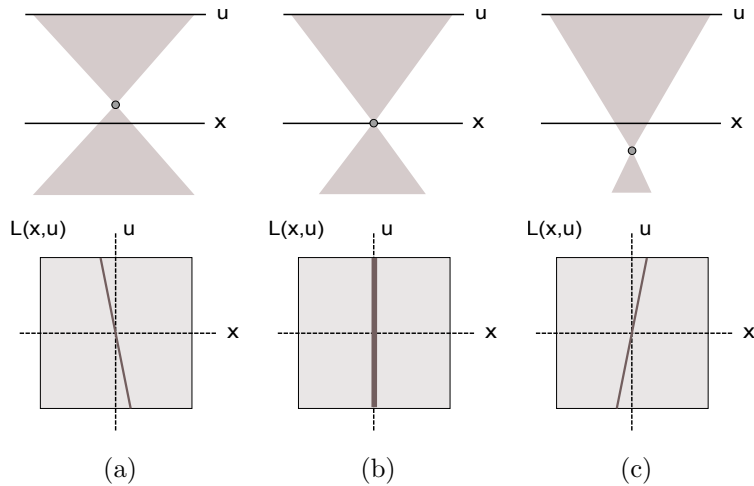


**Figura 2.2:** *Parametrización del light field mediante dos planos.*

La representación de la función plenóptica en espacio abierto, se puede pasar de cinco dimensiones, es decir,  $P(\Theta, \Phi, V_x, V_y, V_z)$ , obviando  $t$  y  $\lambda$ , a una representación 4D que en Levoy y Hanrahan (1996) se denominó *light field*. Esto es consecuencia de asumir que la radiancia a lo largo de una línea recta en el espacio libre no cambia. La reducción a 4D se realiza parametrizando cada rayo por medio de sus puntos de corte con dos planos paralelos, tal y como se muestra en la figura 2.2. El primer plano se parametriza con las coordenadas  $(u, v)$  y el segundo se parametriza con las coordenadas  $(x, y)$ . Esta parametrización, conocida como parametrización de dos planos, fue propuesta, simultáneamente, en Levoy y Hanrahan (1996) y en Gortler et al. (1996). En ambos trabajos se utiliza una notación diferente a la que se utilizará en este documento. En esta memoria se empleará la notación de Ng (2006), con lo que la función plenóptica se denotará con  $L(x, y, u, v)$  o con  $L(\mathbf{x}, \mathbf{u})$ , siendo  $\mathbf{x} = (x, y)$  y  $\mathbf{u} = (u, v)$ .

En Gu et al. (1997), del que también es coautor Gortler, se

profundiza en la parametrización de dos planos, y se relacionan algunas estructuras geométricas como puntos y segmentos, con las estructuras que generan en la función plenóptica parametrizada con dos planos. Estas relaciones se obtienen asumiendo un mundo plano (*flatland*), lo que reduce la función plenóptica a dos dimensiones facilitando su representación en papel. Para ilustrar esta idea, en la figura 2.3 se muestra la relación entre un punto en el espacio y la línea que éste genera en el espacio plenóptico.



**Figura 2.3:** Asociación entre la profundidad de un punto y la pendiente de la línea correspondiente en el light field : (a) El punto se encuentra delante del plano  $\mathbf{x}$  - pendiente negativa; (b) El punto se encuentra sobre del plano  $\mathbf{x}$  - pendiente infinita; (c) El punto se encuentra detrás del plano  $\mathbf{x}$  - pendiente positiva.

Evidentemente, para trabajar con imágenes de escenas reales la representación en *flatland* no es suficiente. Por ello en Gu et al.



(1997) hacen una extensión para el *light field* de cuatro dimensiones (parametrizado con dos planos), donde la estructura correspondiente a un punto tridimensional  $(p_x, p_y, p_z)$  se convierte en un subconjunto bidimensional del espacio  $(x, y, u, v)$ , como se muestra a continuación<sup>1</sup>:

$$subconjunto(a, b) = \left( \left(1 + \frac{1}{p_z}\right) a - \frac{p_x}{p_z}, \left(1 + \frac{1}{p_z}\right) b - \frac{p_y}{p_z}, a, b \right) \quad (2.1)$$

La detección de esta estructura sobre imágenes plenópticas parametrizadas por medio de dos planos permitiría conocer la posición de un punto en el espacio. Esta idea se hace extensible a otros esquemas de captura, siempre que se conozca la estructura o variedad que genera un punto del espacio 3D en el espacio plenóptico y además ésta sea detectable (Berent y Dragotti, 2007).

En este trabajo se asumirá que el comportamiento de las superficies de los objetos es lambertiano, es decir, que el brillo aparente de éstas, es el mismo independientemente del punto de vista del observador (Lambert y Anding, 1892). Bajo esta condición, los rayos pertenecientes a las líneas que se generan para el *flatland* o a los subconjuntos definidos por la ecuación 2.1 en el *light field* 4D, deben ser todos del mismo color e intensidad. Esta suposición, aunque no es válida para todo tipo de materiales, si se cumple en gran medida, lo que permite la detección de estas estructuras y por ende la detección de distancias.

---

<sup>1</sup> Con el fin de mantener una notación uniforme se ha cambiado el orden de las variables con respecto a Gu et al. (1997), donde la expresión figura como  $subconjunto(a, b) = \left( a, b, \left(1 + \frac{1}{p_z}\right) a - \frac{p_x}{p_z}, \left(1 + \frac{1}{p_z}\right) b - \frac{p_y}{p_z} \right)$

### 2.1.1. Rotaciones y subespacios del espacio plenóptico

Asumiendo un *light field* parametrizado mediante dos planos como se ha descrito previamente, se pueden identificar algunas rotaciones y subespacios del espacio 4D que pueden tener cierto interés para el desarrollo de algoritmos que tengan como entrada los datos del *light field*.

En la sección anterior, las coordenadas del *light field* 4D vienen dadas en el siguiente orden  $(x, y, u, v)$  donde  $(x, y)$  son las coordenadas espaciales y  $(u, v)$  son la parametrización de las dimensiones angulares. Una vez discretizada la función  $L(\mathbf{x}, \mathbf{u})$ , estas coordenadas se pueden mapear sobre un espacio bidimensional para su representación, siendo el resultado la imagen  $I(x + u \cdot N_{uv}, y + v \cdot N_{uv}) = L(x, y, u, v)$ . En el mapeo anterior  $N_{uv}$  es el número de puntos de vista discretos capturados, tanto en horizontal como en vertical, y  $N_x$  y  $N_y$  son el número de píxeles a lo ancho y a lo alto, de una imagen tomada con un cierto punto de vista  $(u, v)$ .

Una ordenación alternativa de las coordenadas, que no es más que una rotación del espacio plenóptico 4D, sería  $(u, v, x, y)$ . La imagen mapeada correspondiente vendría dada por  $I(u \cdot N_x + x, v \cdot N_y + y) = L(x, y, u, v)$ . Esta imagen sería el equivalente a la captura obtenida mediante un array de cámaras.

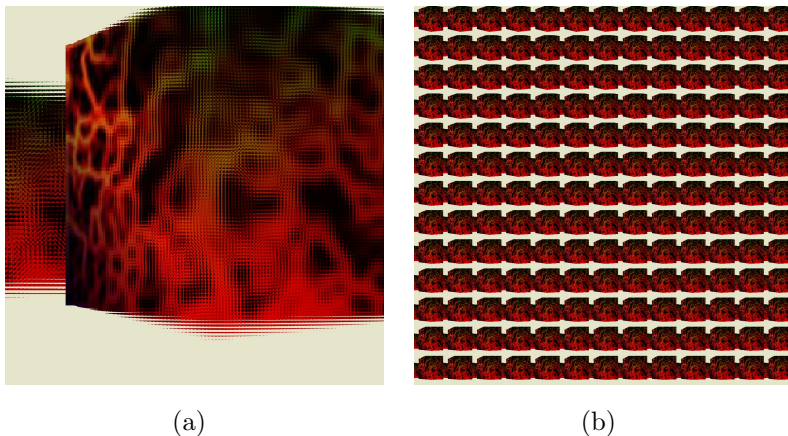
Por otro lado, también puede ser útil la visualización del subespacio  $(x, u)$ , fijando el resto de las coordenadas en  $(y_0, v_0)$ , y el subespacio

## 2.1 *Light field* O FUNCIÓN PLENÓPTICA

---

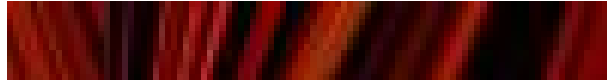
$(y, v)$ , fijando el resto de las coordenadas en  $(x_0, u_0)$ . Ambos subespacios se pueden representar como imágenes epipolares o EPI-imágenes (Bolles et al. 2008), que se obtienen como  $I_{(y_0, v_0)}(x, u) = L(x, y_0, u, v_0)$  para el subespacio  $(x, u)$  y  $I_{(x_0, u_0)}(y, v) = L(x_0, y, u_0, v)$  para el subespacio  $(y, v)$ .

En la figura 2.4, se muestran las imágenes obtenidas rotando el espacio plenóptico para un *light field* de una escena formada por dos cubos situados a distintas distancias, con  $N_{uv} = 13$  y  $N_x = N_y = 100$ . Por otro lado, en la figura 2.5 se muestran dos EPI-imágenes obtenidas a partir del *light field* de la figura 2.4. La primera para  $(x_0 = 50, u_0 = 6)$  y la segunda para  $(y_0 = 50, v_0 = 6)$ .

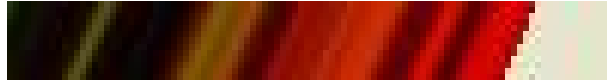


**Figura 2.4:** Rotaciones del espacio plenóptico : (a)  $I(x+u \cdot N_{uv}, y+v \cdot N_{uv}) = L(x, y, u, v)$ ; (b)  $I(u \cdot N_x + x, v \cdot N_y + y) = L(x, y, u, v)$ . En ambos casos:  $N_{uv} = 13$ ,  $N_x = N_y = 100$ .

En la figura 2.4, se puede observar por un lado la disposición, que



(a)



(b)

**Figura 2.5:** Ejemplos de imágenes epipolares del light field que se muestra en la figura 2.4 : (a)  $I_{(y_0=50, v_0=6)}(x, u) = L(x, y_0, u, v_0)$ ; (b)  $I_{(x_0=50, u_0=6)}(y, v) = L(x_0, y, u_0, v)$ . En ambos casos:  $N_{uv} = 13$ ,  $N_x = N_y = 100$ .

por su semejanza con la imagen capturada por una cámara plenóptica, se puede denominar como disposición de subaperturas, y por otro, la disposición de multi-imágenes. La presencia de distintas distancias se puede apreciar especialmente en la figura 2.5(a), en la que se observa el patrón de líneas inclinadas con distintas pendientes que dependen de la distancia. Por ello, no es de extrañar que algunos trabajos utilicen las imágenes epipolares para la extracción de la distancia (Dansereau, 2003; Wanner y Goldluecke, 2012). Por otra parte, se pueden encontrar trabajos que se centran exclusivamente en la generación de representaciones epipolares a partir del *light field* (Wanner et al. 2011).

## 2.2. Dispositivos de adquisición del *light field*

Un dispositivo de captura del *light field* debe ser capaz de muestrear la función plenóptica, en todas sus dimensiones u obviando algunas de ellas. El muestreo del *light field* se puede llevar a cabo mediante diversos dispositivos, cada uno de los cuales debe, a su modo, conservar al menos parte de la información angular de la función plenóptica.

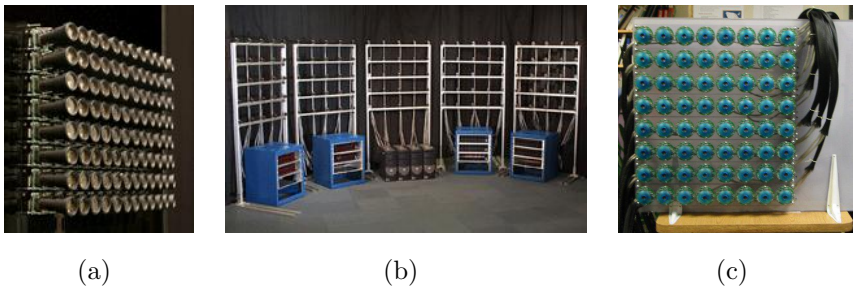
En Zhang y Chen (2006), se describen los aspectos teóricos relacionados con distintos patrones de muestreo del *light field*. En Wetzstein et al. (2013), se revisan distintos métodos de multiplexación y reconstrucción. Por su parte, Wetzstein et al. (2011) contiene una excelente revisión con una sección que describe distintas técnicas para la adquisición del *light field* y dispositivos. También se puede encontrar una revisión más antigua y menos extensa en Levoy (2006).

En las siguientes secciones, se describen diferentes técnicas y se mencionan algunos dispositivos asociados.

### 2.2.1. Arrays de múltiples cámaras

Esta técnica consiste en utilizar múltiples cámaras convencionales situadas en distintas posiciones espaciales para conservar la información angular. Este tipo de dispositivos suelen ser bastante

aparatosos dificultando su manejo. Asimismo, se pueden encontrar distintas configuraciones que condicionan el patrón de muestreo de la función plenóptica. Algunos de estos montajes se muestran en la figura 2.6.



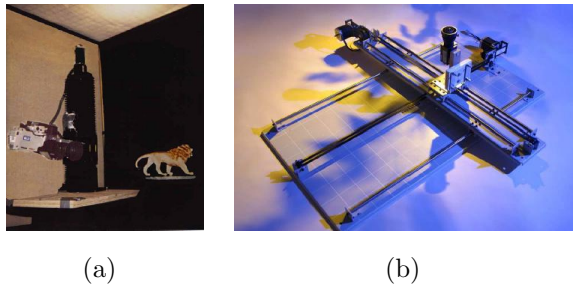
**Figura 2.6:** *Ejemplos de arrays de cámaras: (a) Array de cámaras (Wilburn et al. 2005); (b) Cámaras, tarjetas de procesamiento y 4 PCs de control (Wilburn et al. 2005); (c) Cámara light field distribuida (Yang et al. 2002).*

Construidos específicamente para la captura del *light field* se pueden encontrar los arrays de cámaras de Stanford (Wilburn et al. 2002; Wilburn et al. 2005) o los arrays de cámaras distribuidos de Yang et al. (2002). También se pueden incluir en esta categoría otros dispositivos como los arrays de cámaras flexibles de Nomura et al. (2007), empleados para crear collages de imágenes tomadas desde distintos puntos de vista. Además, se podrían considerar parte de esta categoría las cámaras estereoscópicas y multiestereoscópicas clásicas (Hartley y Zisserman, 2004; Ma et al. 2003).

### 2.2.2. Multiplexación temporal

Esta técnica, aplicable exclusivamente a escenas estáticas, se basa en una cámara en movimiento que captura imágenes desde distintos puntos de vista en distintos instantes de tiempo.

El recorrido que realiza la cámara puede ser controlado mediante brazos robóticos o pórticos (*gantries*) y la captura de imágenes suele estar restringida a posiciones concretas. En Levoy y Hanrahan (1996) y en Unger et al. (2003) se pueden encontrar algunos ejemplos de este enfoque, que se muestran en la figura 2.7. Si el recorrido de la cámara durante la toma de imágenes es totalmente libre se obtienen *light fields* no estructurados (Gortler et al. 1996; Buehler et al. 2001; Davis et al. 2012).



**Figura 2.7:** Ejemplos de pórticos de cámaras: (a) Levoy y Hanrahan (1996); (b) Unger et al. (2003).

Otra forma de multiplexación temporal es el enfoque que se adoptó en Liang et al. (2008), consistente en interponer una serie de máscaras entre la lente y el sensor de una cámara convencional, de

forma que a partir de un conjunto de imágenes tomadas con diferentes máscaras, en diferentes instantes de tiempo, se pueda recuperar el *light field*.

Finalmente, en Ihrke et al. (2008) mantienen la cámara fija y fotografían un espejo móvil en ángulo que refleja la luz procedente de la escena.

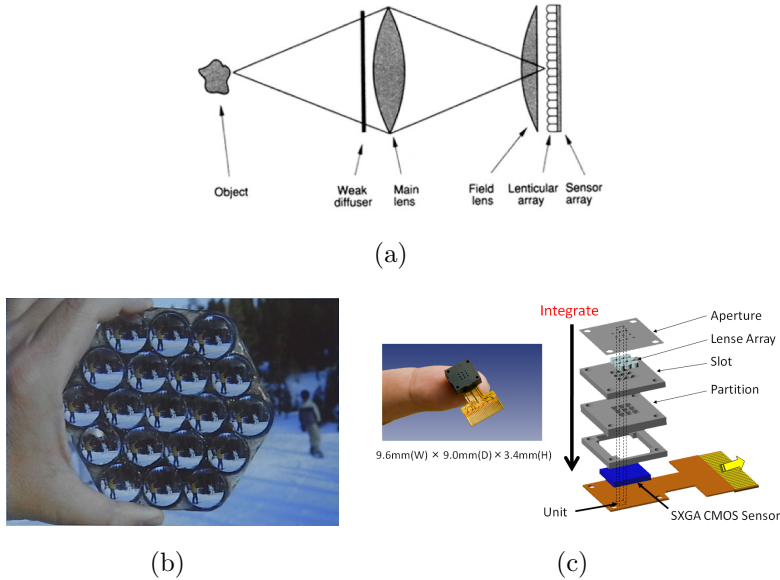
### **2.2.3. Multiplexación espacial**

Estos métodos emplean un único sensor sobre el que se multiplexa espacialmente la información angular. En la figura 2.8 se muestran tres ejemplos de este tipo de dispositivos, que, al utilizar un único sensor, se caracterizan por la necesidad de alcanzar un compromiso entre resolución espacial y resolución angular (Georgiev et al. 2006b). La ventaja de los adquisidores que utilizan multiplexación espacial es que, al usar un único sensor, son menores en cuanto a tamaño comparados por ejemplo con los arrays de cámaras. Esto se puede apreciar comparando los montajes de las figuras 2.6 y 2.8.

La multiplexación espacial se consigue por medio del uso de máscaras o barreras de paralaje, patentadas por Ives en 1903 (Ives, 1903), o por medio de arrays de microlentes interpuestos entre la lente principal y el sensor, de forma que la información angular queda repartida sobre diferentes píxeles espaciales del sensor. El primero en proponer un dispositivo de este tipo fue Lippmann en 1908 (Lippmann, 1908). Esta idea fue posteriormente recuperada en Adelson y Wang



## 2.2 DISPOSITIVOS DE ADQUISICIÓN DEL *light field*



**Figura 2.8:** Ejemplos de cámaras con multiplexación espacial:(a) Esquema óptico de la cámara plenótica (Adelson y Wang, 1992);(b) Array hexagonal de lentes y prismas(Georgiev e Intwala, 2006);(c) Fotografía y esquema de TOMBO (Tanida, 2007).

(1992), donde este dispositivo fue denominado cámara plenótica, y luego fue desarrollado en profundidad por Ren Ng en su tesis doctoral (Ng, 2006). A partir de la publicación de los trabajos de Ren Ng, la cámara plenótica capta el interés de la comunidad científica, desarrollándose algunas variantes y publicándose numerosos trabajos que tratan de procesar los datos capturados.

En cuanto al diseño de la cámara plenótica, en Fife et al. (2006) se presenta un sensor con microlentes integradas y en (Levoy

et al. 2006) se aplican los conceptos de la cámara plenóptica a la microscopía. En Lumsdaine y Georgiev (2008), se propone la versión 2.0 de la cámara plenóptica, también conocida como cámara plenóptica enfocada (*Focused plenoptic camera*), consistente en una ligera modificación en el montaje óptico, quedando el diseño de Ren Ng rebautizado como cámara plenóptica 1.0. En Ng (2006) también se indica la posibilidad de esta modificación óptica, aunque se denomina cámara plenóptica generalizada. Más adelante, en el año 2012, surge otra variante conocida como cámara plenóptica multifoco, que se describe en Georgiev y Lumsdaine (2012) y en Perwass y Wietzke (2012), y que consiste en utilizar un array de microlentes con distintas focales. En Drazic et al. (2012) se puede encontrar un estudio en profundidad sobre los parámetros ópticos a optimizar para utilizar la cámara plenóptica como cámara de distancias.

La primera cámara plenóptica comercial, basada en las ideas de Georgiev sobre la cámara plenóptica 2.0, aparece en el año 2010 fabricada por la empresa alemana Raytrix (Raytrix, 2012). Un año más tarde, nace la empresa Lytro, liderada por Ren Ng (Lytro, 2012).

Aparte de estas dos propuestas comerciales, también cabe destacar el diseño propuesto por la Universidad de La Laguna conocido como CAFADIS (CÁmara de FAse-DIStancias). Este desarrollo se ha llevado a cabo a lo largo de varios proyectos de I+D+i, cuyo objetivo era la creación de sensores pasivos de distancias y de recuperación de la fase del frente de onda. Durante el desarrollo de estos proyectos, además del montaje óptico, se ha desarrollado un conjunto de algoritmos para la

## 2.2 DISPOSITIVOS DE ADQUISICIÓN DEL *light field*

---

síntesis de imágenes (Pérez Nava et al. 2008; Pérez Nava y Lüke, 2009a; Pérez Nava y Lüke, 2009b; Marichal-Hernández et al. 2009; Marichal-Hernández, 2012), la estimación de distancias (Pérez Nava y Lüke, 2009b; Lüke et al. 2010; Marichal-Hernández, 2012), y la recuperación de la fase del frente de onda, lo que ha permitido el uso de la cámara plenóptica en Astrofísica (Rodríguez-Ramos et al. 2008; Rodríguez-Ramos et al. 2009b; Rodríguez-Ramos et al. 2010; Montilla et al. 2010; Rodríguez-Ramos et al. 2012).

Además de la cámara plenóptica, se pueden encontrar otros ejemplos de multiplexación espacial. En Tanida et al. (2001), Tanida (2007) y Yamada y Takahashi (2008) se emplea un conjunto de lentes dispuestas de forma similar a un ojo de mosca para multiplexar espacialmente varias imágenes tomadas desde distintos puntos de vista sobre un mismo sensor. Los resultados son similares a los que pueden ser obtenidos con una cámara plenóptica. Sin embargo, en vez de situar un array de microlentes tras una lente principal se utiliza directamente un array de unas  $3 \times 3$  microlentes. (ver figura 2.8(c)).

Existe otro conjunto de dispositivos de multiplexación espacial que se añaden externamente a la lente principal como los prismas y las lentes propuestas en Georgiev e Intwala (2006) (ver figura 2.8(b)).

Por último, en Ueda et al. (2008) se propone un dispositivo para capturar imágenes desde distintos puntos de vista y con distintas focales. En Unger et al. (2003), Lanman et al. (2006) y Taguchi et al. (2010) se fotografían arrays de espejos para adquirir el *light field*.

#### 2.2.4. Multiplexación en frecuencia

Otra forma de capturar múltiples imágenes en un único sensor es la multiplexación en frecuencia. En este caso, en vez de repartir los píxeles de las imágenes entre los distintos puntos de vista, se lleva a cabo una partición en el dominio frecuencial. Estos métodos suelen basarse en el uso de máscaras que codifican la componente angular. Sin embargo, estas técnicas son menos directas que las anteriores, ya que requieren de un paso de procesado para reconstruir el *light field* a partir de los datos capturados. Se pueden encontrar algunos ejemplos de este tipo de cámaras en Georgiev et al. (2006a) y en Veeraraghavan et al. (2007).

#### 2.2.5. Compressive sensing

El *light field* de una escena puede contener un alto grado de redundancia. Basándose en esta idea, existen métodos de captura que tratan de capturar una versión codificada y comprimida de la señal para reconstruirla posteriormente. Los métodos se conocen con el nombre de compressive sensing y, al igual que los métodos de multiplexación en frecuencia, se basan en el uso de máscaras. Sin embargo, la máscara sirve como base dispersa de la señal, realizándose el proceso de compresión ópticamente. Se pueden encontrar algunos ejemplos de esta técnica en Babacan et al. (2009), en Babacan et al. (2012), en Marwah et al. (2012) y en Kamal et al. (2012).

### 2.3. Repositorios de *light fields*

Con el fin de validar los resultados que se obtienen al procesar la imagen plenóptica, es conveniente recurrir a *light fields* de referencia de repositorios públicos de Internet. De este modo, es posible comparar los resultados obtenidos frente a otros trabajos en los que se ha empleado el mismo repositorio de pruebas.

Se pueden encontrar varios repositorios como el de la Universidad de Stanford que contiene varios *light fields* capturados con pórtico de cámaras, el repositorio del MIT formado únicamente por *light fields* sintéticos (Wetzstein, 2013), el repositorio de la UCSD/MERL con datos reales pero con el inconveniente de que sólo tienen una dimensión angular (MERL, 2007), o el repositorio del *Heidelberg Collaboratory for Image Processing* que contiene *light fields* sintéticos y reales acompañados de un mapa de distancias de referencia (Goldlücke, 2013). Este último fue el empleado para validar los resultados obtenidos en esta tesis doctoral.

### 2.4. Reconstrucción de distancias a partir del *light field*

En esta tesis doctoral se aborda la reconstrucción de distancias a partir del *light field*. En esta sección se describen algunos trabajos previos en esta dirección.

Existe un conjunto de métodos genéricos de estimación de distancias que pueden ser aplicables al *light field*. Por un lado, como éste puede verse como un conjunto de imágenes adquiridas desde múltiples puntos de vista situados sobre un plano se pueden aplicar técnicas geométricas como las descritas en Hartley y Zisserman (2004). Por otro lado, se puede sintetizar un conjunto de imágenes reenfocadas a distintas distancias a partir del *light field*, denominado focal stack, que sirve como entrada de un algoritmo de *depth from focus/defocus* o stereo confocal (Favaro y Soatto, 2006; Hasinoff y Kutulakos, 2009). Un ejemplo de esta aproximación se puede encontrar Marichal-Hernández (2012), donde el autor aplica un operador de contraste sobre el focal stack con el fin de determinar qué imagen está más enfocada y así obtener una estimación de la distancia para cada píxel.

Aunque estos métodos generales pueden ser perfectamente aplicables, existe un conjunto técnicas específicas que se describirán a continuación.

El primer método para la reconstrucción de distancias a partir de imágenes plenópticas fue propuesto en Adelson y Wang (1992). El algoritmo busca el desplazamiento entre dos imágenes utilizando para ello un operador diferencial cuyo cuadrado se integra en un parche de las imágenes. Además proponen una medida de confianza para determinar la fiabilidad de los resultados de distancia obtenidos en cada píxel. Finalmente, Adelson y Wang extienden el operador que proponen a cuatro dimensiones.

En Dansereau (2003), el autor propone un método basado en la

## 2.4 RECONSTRUCCIÓN DE DISTANCIAS A PARTIR DEL LIGHT FIELD

extracción de imágenes epipolares a las que aplica un operador para detectar estructuras lineales cuya pendiente está relacionada con la distancia.

Una técnica alternativa es la desarrollada por los autores de Berent y Dragotti (2007), que consiste en detectar las estructuras plenópticas por medio de técnicas de segmentación en el *light field*. Para ello desarrollan una técnica de segmentación específica para segmentar un conjunto de estructuras lineales como la especificada en la ecuación 2.1.

En Pérez Nava y Lüke (2009b), Lüke et al. (2010) y Pérez Nava (2010) se proponen distintas variantes de un método basado en el cálculo del focal stack de varianzas y su uso para la construcción de un funcional de energía. Este funcional de energía se interpreta en términos de un campo aleatorio de Markov y se minimiza utilizando el método de *Belief Propagation* (Felzenszwalb y Huttenlocher, 2006; Szeliski et al. 2008).

También se han desarrollado técnicas para obtener mapas de distancia densos a resolución completa (Bishop et al. 2009; Bishop y Favaro, 2011). Bishop y Favaro formulan una función de energía basada en las correspondencias entre subimágenes, a la que se añade un término de regularización basado en la variación total del mapa de distancia. La optimización la llevan a cabo mediante el método del gradiente conjugado.

Otro conjunto de técnicas es el desarrollado por T. Georgiev y

que están asociadas a la obtención de imágenes de alta resolución a partir de la cámara plenóptica 2.0. En Georgiev y Lumsdaine (2010), proponen un método basado en el cómputo de la correlación entre la zona central de una imagen de microlente con las imágenes de las microlentes colindantes, sin aplicar ningún tipo de regularización. Por otra parte, en Atanassov et al. (2011), del que también es coautor T. Georgiev, se propone un método de estimación de distancias basado en contenido.

Finalmente, en Wanner y Goldluecke (2012), proponen detectar estructuras lineales a partir de imágenes epipolares de forma similar a Dansereau (2003). Una vez detectadas estas estructuras aplican un método de optimización global satisfaciendo las restricciones de visibilidad y regularizan la solución. Tanto el operador propuesto en Dansereau (2003), como el operador propuesto en Wanner y Goldluecke (2012), son operadores que tratan de estimar la orientación de las estructuras plenópticas que quedan reflejadas en las imágenes epipolares, como se puede observar en la figura 2.5(a).

Los operadores propuestos en Adelson y Wang (1992), en Dansereau (2003) y en Wanner y Goldluecke (2012) son operadores de naturaleza local, ya que emplean un conjunto reducido de datos procedente del entorno del punto para estimar el desplazamiento entre imágenes o la orientación, y por ende la distancia. En esta tesis doctoral se persigue como objetivo la detección de la distancia por medio de operadores de naturaleza local, por lo que se han tomado estos trabajos como punto de partida. En la siguiente sección se



detallan algunos aspectos sobre la detección de orientaciones de forma local en dos dimensiones.

### 2.5. Estimación de orientaciones

El problema de estimación de distancias a partir de las imágenes epipolares extraídas del *light field*, tal y como se propone en Dansereau (2003) y Wanner y Goldluecke (2012), equivale a estimar la orientación de estructuras lineales como las que se observan en la figura 2.5(a). Estos métodos permiten estimar la orientación de forma local utilizando únicamente un entorno del *light field*. Esta idea se tomará como punto de partida para desarrollar un método que permita la estimación de las orientaciones directamente sobre entornos del *light field* de cuatro dimensiones y que permita su implementación con algoritmos paralelizables.

En la siguientes secciones se describen los elementos necesarios para estimar los gradientes de las imágenes epipolares, así como una técnica para estimar la orientación de patrones lineales que se tomará como base para el desarrollo de un operador en cuatro dimensiones.

#### 2.5.1. Estimación de gradientes

El gradiente en un punto de una señal multidimensional se obtiene calculando la derivada parcial en cada una de las direcciones. En el

caso de que la función sea una señal muestreada y limitada en banda, las derivadas parciales se obtienen convolucionando las muestras con la derivada parcial del filtro de reconstrucción.

$$\frac{\partial L_f(\mathbf{x})}{\partial x_i} = \widetilde{L}_f * \frac{\partial K(\mathbf{x})}{\partial x_i} \quad (2.2)$$

El problema que surge es que, en el caso de las señales limitadas en banda, el filtro de reconstrucción  $K$  es una función sinc cuyo dominio es infinito. Esto obliga a realizar la operación en el dominio de la frecuencia o sustituir el filtro derivativo ideal por una versión truncada aproximada. En la literatura se han definido diversos operadores diferenciales consistentes en el ajuste de una aproximación polinomial de la función original y su diferenciación, o bien en el filtrado con la derivada de un núcleo de suavizado truncado.

El uso de filtros aproximados genera una serie de errores sistemáticos que afectan a los resultados. Dichos errores se pueden minimizar optimizando los coeficientes del filtro respecto a una función de error. En Farid y Simoncelli (2004) se propone un procedimiento para obtener un conjunto de filtros derivativos multidimensionales separables optimizados respecto a una función de error que también se propone en ese trabajo. Seguidamente se describen algunos detalles de este procedimiento, ya que se utilizará para calcular las derivadas implicadas en el método de estimación en cuatro dimensiones que se desarrollará en el capítulo 4.

Un filtro derivativo separable multidimensional se puede escribir

de la siguiente manera:

$$\mathcal{F} \left\{ \frac{\partial K(\mathbf{x})}{\partial x_i} \right\} = D_1(\omega_i) \prod_{j \neq i} P(\omega_j) \quad (2.3)$$

donde  $D_1(\omega)$  es la respuesta al impulso del filtro a aplicar en la dirección de la derivada y  $P(\omega)$  es la respuesta al impulso del filtro a aplicar en las direcciones perpendiculares de la derivada.

Para obtener los coeficientes óptimos de los filtros  $P$  y  $D_1$ , Farid y Simoncelli utilizan la siguiente función de costo:

$$E\{P, D_1\} = \frac{\int_{\omega} |j\omega P(\omega) - D_1(\omega)|^2 d\omega}{\int_{\omega} |P(\omega)|^2 d\omega} \quad (2.4)$$

A continuación discretizan esta función y optimizan los coeficientes de los filtros imponiendo además una restricción adicional de simetría. Utilizando este procedimiento obtienen los siguientes filtros de tamaño 3:

$$\begin{aligned} \vec{p} &= \langle 0,229879 & 0,540242 & 0,229879 \rangle \\ \vec{d}_1 &= \langle -0,425287 & 0,000000 & 0,425287 \rangle \end{aligned} \quad (2.5)$$

### 2.5.2. Detección de estructuras lineales en imágenes epipolares

Las imágenes epipolares descritas en la sección 2.1.1 no son más que cortes bidimensionales del espacio plenóptico y presentan una estructura de líneas cuya pendiente depende de la geometría de la

escena. Por tanto, la detección de las pendientes en las distintas imágenes epipolares que se pueden formar a partir del *light field* daría lugar a dos estimaciones de distancia por cada uno de los rayos.

En procesamiento de imágenes se han propuesto diversas técnicas para la detección de estructuras lineales en imágenes bidimensionales. La primera técnica para la detección de patrones orientados es la propuesta por Bigün y Granlund (1987). Ambos métodos se basan el cálculo del vector gradiente de los puntos de la imagen y en la construcción del tensor de estructura que se obtiene como sigue:

$$T_{I(x,y)} = \begin{pmatrix} \widehat{I}_x^2 & \widehat{I}_x \widehat{I}_y \\ \widehat{I}_x \widehat{I}_y & \widehat{I}_y^2 \end{pmatrix} \quad (2.6)$$

donde  $\widehat{I}_x^2 = \int_{\Omega} \omega(\mathbf{x} - \mathbf{x}_0) |I_x(\mathbf{x})|^2 d\mathbf{x}$ ,  $\widehat{I}_y^2 = \int_{\Omega} \omega(\mathbf{x} - \mathbf{x}_0) |I_y(\mathbf{x})|^2 d\mathbf{x}$ , y  $\widehat{I}_x \widehat{I}_y = \int_{\Omega} \omega(\mathbf{x} - \mathbf{x}_0) I_x(\mathbf{x}) I_y(\mathbf{x}) d\mathbf{x}$ . La función  $\omega(\mathbf{x} - \mathbf{x}_0)$  es una función de peso centrada en el punto a analizar  $\mathbf{x}_0$ , mientras que  $I_x(x, y) = \frac{\partial I(x,y)}{\partial x}$  y  $I_y(x, y) = \frac{\partial I(x,y)}{\partial y}$  son las derivadas parciales de la imagen respecto de  $x$  e  $y$ .

Los autovalores del tensor de estructura 2D vienen dados por:

$$\lambda_{0,1} = \frac{1}{2} \widehat{I}_y^2 + \widehat{I}_x^2 \pm \sqrt{\left(\widehat{I}_y^2 - \widehat{I}_x^2\right)^2 + 4 \widehat{I}_x \widehat{I}_y^2} \quad (2.7)$$

Para un patrón lineal, el autovector unitario  $\langle k_x, k_y \rangle$  correspondiente al autovalor más pequeño  $\lambda_0$  cumple que:

$$\left(\widehat{I}_x^2 - \lambda_0\right) k_x - \widehat{I}_x \widehat{I}_y k_y = 0 \quad (2.8)$$

## 2.5 ESTIMACIÓN DE ORIENTACIONES

---

La orientación del patrón lineal viene dada por el ángulo doble  $\theta = \arctan(k_x, k_y)$  del autovector de lo que se puede deducir que:

$$2\theta = \arctan\left(\widehat{I}_y^2 - \widehat{I}_x^2, 2\widehat{I}_x\widehat{I}_y\right) \quad (2.9)$$

Finalmente, la pendiente de la estructura lineal vendría dada por:

$$s = \tan\left(\frac{1}{2} \arctan\left(\widehat{I}_y^2 - \widehat{I}_x^2, 2\widehat{I}_x\widehat{I}_y\right)\right) \quad (2.10)$$

Además, se define una medida de confianza para la medida de orientación obtenida mediante la ecuación 2.10 teniendo en cuenta las siguientes restricciones sobre los autovalores: Cuando el patrón de textura es perfectamente lineal se tiene que  $\lambda_0 = 0$  y  $\lambda_1 \gg 0$ , cuando ambos autovalores son nulos, se trata se una textura uniforme, y cuando se tiene que  $\lambda_0 \gg 0$  y  $\lambda_1 \gg 0$  hay dos direcciones predominantes. Una medida de confianza que refleja estas propiedades es la de la siguiente ecuación:

$$C = \left(\frac{\lambda_1 - \lambda_0}{\lambda_1 + \lambda_0}\right)^c = \left(\frac{\sqrt{(\widehat{I}_y^2 - \widehat{I}_x^2)^2 + 4\widehat{I}_x\widehat{I}_y^2}}{\widehat{I}_x^2 + \widehat{I}_y^2}\right)^c \quad (2.11)$$

donde  $c$  es una constante positiva que permite controlar el rango dinámico de la medida de confianza. En lo sucesivo se tomara  $c$  igual a 2.

## 2.6. Regularización de mapas de distancias

Según el siguiente teorema, postulado en Baker et al. (2003), la reconstrucción de distancias a partir del *light field* no es posible en todos los casos:

*Suppose that  $S_1$  is a Lambertian Scene and  $L_1$  a light field of that scene with the following properties:*

- 1.  $L_1$  is defined on a finite collection of continuously differentiable surface patches in  $[0, 1]^3$ ,*
- 2.  $L_1$  is open or, equivalently, every ray in  $L_1$  is contained in a 4D open subset of rays in  $L_1$  (i.e., across the 2D surface on which  $L_1$  is defined and the 2D space of directions),*
- 3. every ray in  $L_1$  intersects  $S_1$  somewhere,*
- 4. every point in free space and on the surface of  $S_1$  is visible in  $L_1$  somewhere, and*
- 5. no ray in  $L_1$  is tangent to  $S_1$  or to the surface patch on which the light-field is defined.*

*Then, there is another Lambertian Scene  $S_2$  (i.e., that has a different occupied volume) that also has light-field  $L_1$  if and only if  $L_1$  is constant in a 4D open subset.*

La consecuencia de este teorema es que si en el *light field* existe una zona constante, sin textura, existen múltiples superficies que pueden

## 2.6 REGULARIZACIÓN DE MAPAS DE DISTANCIAS

---

haber generado ese *light field*, es decir, que el problema no tiene una solución única y por tanto no se puede generar directamente un mapa de distancias denso. Para poder generar un mapa de distancias denso es necesario restringir el espacio de soluciones incorporando conocimiento previo sobre la escena. Este conocimiento previo se puede incorporar, por ejemplo, por medio de restricciones de suavidad o planaridad de la superficie. Rellenar los huecos que se producen en las zonas sin textura requiere de un método de inferencia que permita añadir estas restricciones.

Un método para introducir conocimiento previo sobre la escena en el proceso es formular el problema de estimación como un problema de minimización de energía:

$$\min_u \{R(u) + F(u, \hat{s})\} \quad (2.12)$$

donde  $F(u, \hat{s})$  es un término de fidelidad que mide la diferencia entre los datos, que en este caso será un mapa de pendientes medido que se denotará con  $\hat{s}$ , y el mapa de pendientes estimado  $u$ . La función de fidelidad suele ser el error cuadrático o la diferencia absoluta. Además, como el problema de estimación de distancias en general es un problema mal condicionado, se incluye el término de regularización  $R(u)$  para modelar el conocimiento a priori sobre la escena. La técnica concreta para solucionar el problema de optimización depende de los datos de entrada, las funciones empleadas, así como de la convexidad del mismo. Según la variante concreta del problema, se han aplicado métodos como *Graph Cuts* (Boykov et al. 2001) o *Belief Propagation* (Yedidia et al. 2003; Felzenszwalb y Huttenlocher, 2006; Szeliski et

al. 2008). Ambos métodos formulan el problema de estimación en el marco de los campos aleatorios de Markov. Generalmente, se convierte el problema en un problema de etiquetado en el que a cada píxel se le asigna una etiqueta relacionada con la distancia o la disparidad que se selecciona de un conjunto discreto de etiquetas candidatas para minimizar el funcional de energía. Esta fue la aproximación por la que se optó en Pérez Nava y Lüke (2009b) y Lüke et al. (2010).

El principal inconveniente, achacable a las dos técnicas mencionadas, es el uso de un conjunto discreto de distancias candidatas, que implica el mantenimiento de un volumen de datos 3D correspondiente a la función de costo. Además, esto requiere de un uso intensivo de memoria que limita las resoluciones de los mapas de distancias tratables con estos algoritmos.

Los métodos variacionales, que se hicieron populares tras la publicación de Horn y Schunck (1980), son una alternativa que permiten el uso de valores reales sin incurrir en un uso excesivo de memoria. Las velocidades de cómputo dependen en gran medida del modelo concreto y del método minimización empleado. Existen numerosos modelos aplicados a problemas de reducción de ruido, de segmentación, de relleno y de deconvolución de imágenes. Dos modelos muy utilizados son el modelo de Mumford-Shah (Mumford y Shah, 1989) y el modelo Rudin-Osher-Fatemi (ROF) (Rudin et al. 1992), de los que existen numerosas adaptaciones y variantes. Desde la publicación del trabajo de Rudin et al. la variación total se ha utilizado ampliamente como término de regularización en numerosos problemas,



## 2.6 REGULARIZACIÓN DE MAPAS DE DISTANCIAS

---

entre otros, a la estimación de disparidades estéreo (Kumar et al. 1997; Slesareva et al. 2005; Miled y Pesquet, 2006; Pock et al. 2008).

El término de fidelidad utilizado en el modelo ROF es el error cuadrático, lo que hace que las imágenes regularizadas sufran de una pérdida de contraste, que se trata en Strong y Chan (2000) y en Chan et al. (2005), y que no es conveniente para la regularización de mapas de distancias. Un modelo que no presenta este problema y que se caracteriza por su robustez frente a *outliers*, es el modelo TV-L1 desarrollado en Alliney (1997) y en Chan y Esedoglu (2005).

El modelo TV-L1 consiste en el uso del error absoluto como término de fidelidad y de la variación total como término de regularización:

$$\arg \min_u \int_{\Omega} \lambda |u - \widehat{s}| d\mathbf{x} + TV(u) \quad (2.13)$$

donde  $TV(u)$  es la variación total del mapa de distancias estimado  $u$ , esto es,  $TV(u) = \int_{\Omega} |\nabla u(x)| d\mathbf{x}$ .  $\lambda$  es un valor dependiente de la posición que permite balancear la influencia del término de fidelidad y del de regularización en cada píxel.

Se ha propuesto el uso de diferentes métodos y aproximaciones para minimizar el modelo TV-L1. Por ejemplo, en Chan y Esedoglu (2005) se modifica ligeramente el modelo, introduciendo términos de regularización, para que éste sea diferenciable en todo el dominio y se aplica el método del gradiente descendente para la minimización. En Zach et al. (2007) se utiliza un enfoque basado en dualidad. Desde la publicación de Goldstein y Osher (2009), se ha comenzado a emplear el método split Bregman para la minimización del modelo ROF. También

se pueden encontrar ejemplos de la aplicación de este método al modelo TV-L1 en Wei et al. (2010) y en Getreuer (2012).

El mayor inconveniente del uso de modelos basados en la variación total es que asumen que los mapas de distancias están compuestos por un conjunto de parches constantes, lo que puede dar lugar a la aparición del “efecto escalera” cuando se aplica a superficies inclinadas respecto de la cámara. Para solventar este inconveniente, se ha propuesto la variación total generalizada, que consiste en añadir términos basados en las derivadas de orden superior con el fin de modelar superficies planas o polinomiales (Bredies et al. 2010).

## 2.7. Fotografía a partir del *light field*

La vinculación entre el *light field* y la imagen de una escena se puede entender fácilmente pensando en el funcionamiento de un sensor de imagen. Un sensor de imagen integra las dimensiones angulares de la función plenóptica para cada punto de su superficie. En el caso concreto del *light field*, el sensor integraría en un punto las intensidades de todos los rayos que lo atraviesan. La relación entre la función plenóptica y la formación de imágenes se describe con mayor detalle en el apéndice A.

La existencia de dispositivos de captura del *light field*, como los descritos en la sección 2.2, permite la síntesis de imágenes en diversas condiciones a partir de los datos capturados, utilizando técnicas de

## 2.7 FOTOGRAFÍA A PARTIR DEL *light field*

---

*Image Based Rendering* (IBR). Las técnicas IBR permiten sintetizar diversos tipos de imágenes como imágenes con distintas aperturas, imágenes a distintas distancias focales, imágenes desde distintos puntos de vista, o incluso imágenes con una superficie de enfoque variable (Isaksen et al. 2000), a partir de un *light field*.

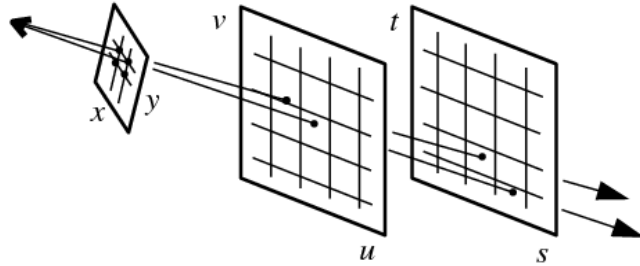
Las primeras técnicas para sintetizar imágenes a partir del *light field* fueron presentadas en Levoy y Hanrahan (1996) y en Gortler et al. (1996). En Levoy y Hanrahan (1996) se explica cómo sintetizar imágenes a partir del *light field* de cuatro dimensiones definido en ese artículo. Los autores aplican una transformación de las coordenadas de un plano imagen, situado en una cierta posición del espacio, a las coordenadas de los rayos<sup>2</sup> en  $L(x, y, u, v)$ . Una vez mapeados los puntos del plano imagen al espacio de rayos realizan una interpolación cuatrilineal para obtener el valor de la intensidad del rayo. Este proceso se ilustra en la figura 2.9 extraída de Levoy y Hanrahan (1996).

En Gortler et al. (1996) se propone una idea similar para la generación de nuevas imágenes, pero utiliza información geométrica de las escena en el proceso de interpolación del *light field*.

En Isaksen et al. (2000) se generaliza el proceso de renderizado. La diferencia con respecto a Levoy y Hanrahan (1996) y Gortler et al. (1996) es que Isaksen y McMillan proponen utilizar una superficie focal independiente. De este modo, permiten cambiar dinámicamente la superficie focal sin cambiar la organización de la base de datos de rayos

---

<sup>2</sup>Los autores de Levoy y Hanrahan (1996) emplean la notación  $L(s, t, u, v)$  para parametrizar el espacio de rayos.

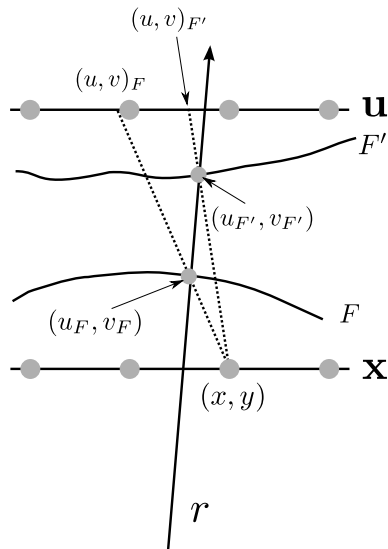


**Figura 2.9:** Proceso de mapeo de las coordenadas de un plano imagen al espacio de rayos (Levoy y Hanrahan, 1996).

subyacente. Esto requiere un mapeo  $M_F : (\hat{x}, \hat{y}, u_F, v_F) \mapsto (u, v)$  para obtener el rayo  $(x, y, u, v)$  correspondiente de la base de datos de rayos<sup>3</sup>. En la figura 2.10 se muestra como obtener el mapeo  $M_F$ . Se busca una recta entre el punto  $(u_F, v_F)$  de la superficie focal y los puntos  $(x, y)$  del plano trasero de la parametrización de dos planos. Luego para cada par de puntos  $(x, y)$  y  $(u_F, v_F)$  se obtiene el punto de corte con el plano delantero  $\mathbf{u}$  en  $(u, v)_F$ . De este modo, el rayo a recuperar de la base de datos de rayos queda como  $(x, y, u, v)|_{(u,v)=(u,v)_F}$ . En la figura se muestra dicha recuperación para dos superficies focales  $F$  y  $F'$ .

Uno de los usos del IBR sobre el *light field* es la síntesis del *focal stack*, que consiste en calcular un conjunto de imágenes enfocadas a distintas distancias. La base para el cálculo del focal stack es el operador fotográfico que consiste en integrar el *light field* con

<sup>3</sup>La notación seguida por Isaksen y McMillan es  $M_F : (\hat{s}, \hat{t}, u_F, v_F) \mapsto (u, v)$



**Figura 2.10:** *Reparametrización dinámica de rayos para dos superficies focales.*

distintas pendientes sobre las dimensiones angulares, como se muestra a continuación:

$$P_\alpha[L](\mathbf{x}') = \frac{1}{\alpha^2 F^2} \int L \left( \mathbf{u} \left( 1 - \frac{1}{\alpha} \right) + \frac{\mathbf{x}'}{\alpha}, \mathbf{u} \right) d\mathbf{u} \quad (2.14)$$

En la ecuación 2.14,  $F$  es la distancia entre los dos planos que parametrizan el *light field* y  $\alpha$  indica la posición del plano de enfoque respecto del plano delantero de la parametrización de dos planos. El valor de  $\alpha$  viene dado por  $\alpha = \frac{F'}{F}$ , siendo  $F'$  la posición del plano de enfoque (Ng, 2006).

Si bien el focal stack se puede obtener aplicando métodos de fuerza bruta como los de Levoy y Hanrahan (1996), Gortler et al. (1996) o Isaksen et al. (2000), se han desarrollado métodos de cómputo eficientes. El primero en desarrollar un algoritmo eficiente para el cálculo del focal stack fue Ren Ng en su tesis doctoral (Ng, 2006). Para ello, aplicó la técnica del *Slice de Fourier* sobre el *light field* adquirido mediante una cámara plenóptica. En la figura 2.11 se muestra el resultado obtenido por Ren Ng con su método.

Tras la publicación de la tesis doctoral de Ren Ng en 2006, aumentó de forma significativa el interés en las investigaciones relacionadas con el *light field* y las cámaras plenópticas, desarrollándose también diversos algoritmos para el cálculo del focal stack. En Pérez Nava et al. (2008) los autores presentan un algoritmo basado en la transformada fraccionaria de Fourier para el cálculo de la transformada discreta del *focal stack*. Posteriormente, en Pérez Nava y Lüke (2009a) se introduce una mejora de este

## 2.7 FOTOGRAFÍA A PARTIR DEL *light field*

---



**Figura 2.11:** *Reenfoco digital aplicando el método del slice de Fourier (Extraído de Ng et al. (2005)) .*

algoritmo desde el punto de vista computacional, mientras que en Marichal-Hernández et al. (2009) se presenta el algoritmo rápido de la transformada aproximada del focal stack, basado en la extensión a cuatro dimensiones de la transformada de Radón. Desde un punto de vista computacional este es el algoritmo más eficiente conocido hasta la fecha, al computar un conjunto completo de imágenes enfocadas a distintas distancias en tiempo lineal con el número de píxeles de la imagen plenóptica<sup>4</sup>. De este último, se han publicado algunas extensiones en Marichal-Hernández et al. (2011) y es una de las partes fundamentales de la tesis doctoral de J.G. Marichal-Hernández (Marichal-Hernández, 2012).

Los algoritmos anteriormente citados, se caracterizan por que la resolución de las imágenes que generan coincide con el número de microlentes del array empleado en la construcción de la cámara plenóptica. Esta limitación se debe al compromiso entre la resolución espacial y la resolución angular de la cámara plenóptica (Georgiev et al. 2006b). Por ejemplo, en Ng et al. (2005), se utiliza un array de microlentes de  $292 \times 292$  microlentes. Comparado con la imagen obtenida con una cámara fotográfica de 14 mega-píxeles o una cámara de vídeo de alta definición ( $1080 \times 720$  píxeles) este número es claramente insuficiente para las exigencias actuales.

En Lumsdaine y Georgiev (2008) se presentan los resultados del primer método para aumentar la resolución de las imágenes obtenidas, utilizando para ello *light fields* adquiridos mediante una cámara

---

<sup>4</sup>Light field capturado mediante una cámara plenóptica



## 2.7 FOTOGRAFÍA A PARTIR DEL *light field*

---

plenóptica 2.0, presentada en ese mismo trabajo. En Lumsdaine y Georgiev (2009), se rebautiza la cámara plenóptica 2.0 como cámara plenóptica enfocada y se describe el método de renderizado con mayor detalle, mostrando además resultados de su aplicación sobre imágenes obtenidas con la cámara de Ng et al. (2005). El algoritmo de Lumsdaine y Georgiev (2009) genera un conjunto de imágenes cuyo tamaño depende de la distancia de enfoque. Esto fue resuelto en Pérez Nava y Lüke (2009b), donde se presentó un algoritmo que permitía obtener un focal stack cuyas imágenes eran de un tamaño aproximadamente igual a la mitad del sensor utilizado para capturar la imagen plenóptica.

Estas técnicas de superresolución<sup>5</sup> tienen el inconveniente de que las zonas no enfocadas de la imagen presentan artefactos de alta frecuencia, conocidos como artefactos plenópticos, en vez de desenfoque. Para abordar este problema, Georgiev recurre a una serie de correcciones basadas en la distancia a la que están situados los objetos de la escena (Georgiev y Lumsdaine, 2010). Otra técnica de superresolución<sup>6</sup> basada en estimaciones de distancia es la presentada en Bishop et al. (2009).

Disponer de una estimación de la distancia permite la síntesis de

---

<sup>5</sup>En este caso la palabra superresolución no se refiere a las técnicas de resolución convencionales como las descritas en Ng y Bose (2003), sino al hecho de que con estas técnicas se obtienen imágenes de mayor tamaño que con las técnicas de *light field* rendering convencionales.

<sup>6</sup>En este caso si se aplican técnicas de superresolución convencional (Ng y Bose, 2003).

imágenes totalmente enfocadas seleccionando los píxeles adecuados del conjunto de imágenes del focal stack. Se pueden encontrar ejemplos de la aplicación de esta idea en Pérez Nava y Lüke (2009b) y Marichal-Hernández (2012). Por otro lado, se pueden encontrar técnicas para la síntesis de la imagen totalmente enfocada como la presentada en Takahashi et al. (2003) y Kubota et al. (2004) que no precisan una medida directa de distancias.

## 2.8. Hardware gráfico (GPU)

La evolución del hardware gráfico en los últimos años ha permitido que las unidades de procesamiento gráfico hayan pasado de ser un hardware para un propósito específico, como el renderizado de gráficos, a ser coprocesadores masivamente paralelos y programables. Las primeras generaciones de unidades de procesamiento gráfico (GPU) fueron chips con arquitecturas específicas con cauces de procesamiento y funciones fijas cuyas salidas estaban limitadas a colores RGB de 8 bits por canal. Hoy en día se trata de arquitecturas flexibles y programables que permiten operar en punto flotante de precisión simple o incluso doble precisión. Asimismo, su funcionalidad se aleja cada vez más del procesamiento de gráficos y permite realizar cálculos de propósito general (Owens et al. 2008).

El camino hacia la programabilidad de las tarjetas gráficas se abre en la década del 2000, cuando los fabricantes comenzaron a producir modelos que permitían programas de fragmento y de vértices.

## 2.8 HARDWARE GRÁFICO (GPU)

---

La posibilidad de crear programas de propósito general surge con la creación de lenguajes ensambladores para programas de vértice y fragmento ejecutables sobre este hardware (Olano y Lastra, 1998; Lindholm et al. 2001). La programabilidad se queda de manera definitiva con la incorporación de las primitivas `ARB_vertex_program` y `ARB_fragment_program` al estándar *OpenGL* en 2002.

Los primeros usos de unidades de procesamiento gráfico para cómputo de propósito general consistieron en traducir manualmente los problemas generales a formas compatibles con el procesamiento gráfico, utilizando las primitivas gráficas para ejecutar los cálculos. La dificultad de este proceso hizo que se desarrollaran lenguajes específicos para el cómputo de propósito general sobre GPUs que admitían una serie de primitivas generales que luego eran traducidas a primitivas gráficas por el compilador. Algunos ejemplos de este tipo de lenguajes son *BrookGPU* (Buck et al. 2004), *Shader algebra* (McCool et al. 2004) o *Accelerator* (Tarditi et al. 2006).

En 2007, nVIDIA lanzó CUDA (*Compute Unified Device Architecture*), una arquitectura de GPU que permitía realizar cálculos de propósito general y de altas prestaciones de forma nativa. Dicha arquitectura fue incorporada a la totalidad de la línea de productos de nVIDIA y actualmente se encuentra en su tercera generación. nVIDIA también lanzó la línea de productos *Tesla* (Lindholm et al. 2008) destinada exclusivamente al cómputo de altas prestaciones. Además, estas nuevas arquitecturas estaban dotadas de un kit de desarrollo, el nVIDIA CUDA SDK (NVIDIA Corporation,

2012), facilitando así el desarrollo de software de propósito general sobre la GPU. Asimismo, desde el lanzamiento de CUDA se han desarrollado interfaces de programación para diversos lenguajes de programación como C, Fortran, Java o Python, entre otros.

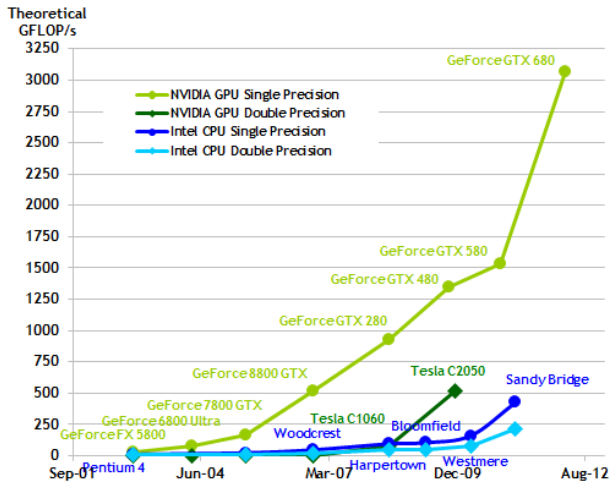
Por otra parte, AMD (*Advanced Microdevices, Inc*) absorbe al fabricante de GPUs ATI en 2006, comenzando a comercializar la línea de productos *FireStream* (Advanced Microdevices, Inc. 2012a) también acompañada de su kit de desarrollo *AMD APP Software Development Kit* (Advanced Microdevices, Inc. 2012b).

También han cobrado importancia otros estándares abiertos como *OpenCL* (Khronos OpenCL Working Group, 2012), cuya primera especificación se completó en 2009. OpenCL es un conjunto de primitivas de programación para sistemas heterogéneos. Dichas primitivas fueron incorporadas tanto por nVIDIA como por AMD a sus plataformas de desarrollo y actualmente cuentan con un amplio soporte por parte de ambos fabricantes.

Los desarrollos de esta tesis doctoral se han realizado sobre CUDA, utilizando el lenguaje CUDA C, ya que se disponía de experiencia previa con desarrollos sobre esta plataforma. La motivación de realizar los desarrollos sobre GPU es que los incrementos de velocidad pueden ser notables frente a los desarrollos sobre CPUs convencionales. En la figura 2.12 se muestra la evolución en el tiempo de la capacidad de cómputo de los diferentes modelos de GPUs fabricados por nVIDIA. También se muestra esta evolución para las CPUs de Intel, pudiéndose observar cómo la capacidad de cómputo de las GPUs es

## 2.8 HARDWARE GRÁFICO (GPU)

---



**Figura 2.12:** Capacidad de cómputo en GFLOP/s para la CPU y la GPU (Figura tomada de NVIDIA Corporation (2012)).

considerablemente mayor que la de las CPUs.

CUDA es una arquitectura SIMT (*Single Instruction Multiple Threads*) en la que un mismo programa es ejecutado por múltiples hilos de ejecución sobre datos distintos. Esto hace que esta arquitectura se adapte especialmente bien a aquellos problemas que permitan un elevado paralelismo a nivel de datos. Los problemas de procesamiento de imágenes suelen adaptarse bien a estas arquitecturas cuando los píxeles de una imagen de salida pueden calcularse de forma independiente al resto (NVIDIA Corporation, 2012).

Un programa ejecutado por la GPU se denomina *kernel*. La ejecución de un *kernel* consiste en la ejecución de las mismas instrucciones sobre distintos datos, lo cual es posible gracias a los

*streaming multiprocessors* (SM). Cada SM es capaz de ejecutar un conjunto de hilos, denominado bloque. Los hilos de un bloque comparten una pequeña memoria caché que puede ser directamente administrada por el programador o por medio de un mecanismo de caché automático<sup>7</sup>. Dicha memoria compartida permite ocultar las latencias de la memoria global de la GPU.

Para su ejecución, los hilos de un *kernel* se organizan en bloques que a su vez se organizan en lo que se denomina un *grid*. Los hilos y los bloques se identifican por medio de identificadores que pueden tener hasta tres dimensiones. A través de dichos identificadores se determinan los datos sobre los que se ejecutan las instrucciones. Esta organización permite la escalabilidad de los programas a medida que aparecen nuevos modelos de GPU en el mercado, ya que, dependiendo del número de SMs de la GPU, el grado de paralelismo es mayor o menor.

Aunque los modelos más recientes permiten mapear directamente la memoria RAM del ordenador en el espacio de memoria de la GPU, generalmente se realiza un proceso de copia de datos desde la memoria RAM del ordenador a la memoria global de la GPU, luego se produce la ejecución de uno más *kernels* y finalmente se procede a la copia de vuelta de los resultados.

A la hora de diseñar algoritmos para esta plataforma es necesario tener en cuenta el grado de paralelismo de datos que se puede alcanzar,

---

<sup>7</sup>La administración automática sólo es posible a partir de la *Compute capability 2.0*

## 2.8 HARDWARE GRÁFICO (GPU)

---

así como las transferencias de memoria que se puedan producir entre las memorias de la CPU y la GPU, o viceversa. Si bien esto no siempre es posible, se ha tratado de adaptar los algoritmos para que sean ejecutables sobre arquitecturas paralelas como CUDA.





# Capítulo 3

## Medida y simulación de *light fields*

*“How does somebody know what they want if they haven’t even seen it?”*

---

Steve Jobs

En este capítulo se concretarán algunos aspectos sobre los procedimientos que se utilizarán para obtener los *light fields* sobre los que se aplicará el método de estimación de distancias que se desarrollará en los capítulos 4 y 5.

Los *light fields* obtenidos estarán parametrizados mediante dos planos usando el método que se describió en la sección 2.1. El número de muestras espaciales horizontales y verticales se denotarán con  $N_x$  y  $N_y$  respectivamente. El de muestras angulares se representará con  $N_{uv}$

lo que nos indicará que el *light field* tiene  $N_{uv} \times N_{uv}$  puntos de vista.

En las siguientes secciones se exponen los cuatro conjuntos de datos obtenidos. los dos primeros fueron adquiridos mediante el método de multiplexación temporal y la cámara plenóptica CAFADIS respectivamente. Los dos últimos corresponden a *light fields* que fueron simulados.

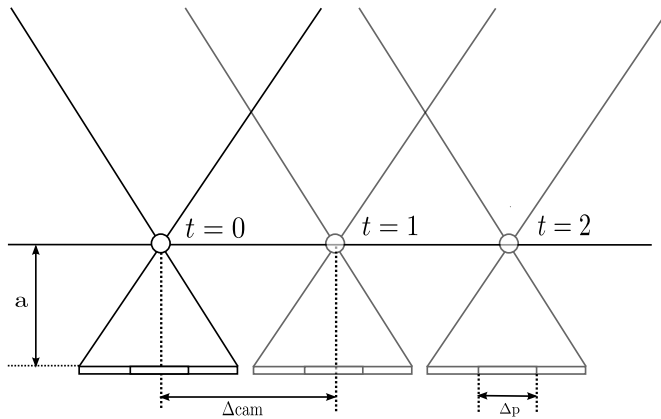
### 3.1. Adquisición mediante multiplexación temporal

La captura de *light fields* mediante la técnica de multiplexación temporal permite obtener datos con un montaje óptico sencillo, utilizando una cámara fotográfica convencional y un mecanismo para desplazarla de forma precisa en dos direcciones perpendiculares sobre un plano, aunque tiene la desventaja de que la escena debe permanecer inmóvil durante el tiempo de medida.

En la figura 3.1 se muestran las magnitudes que caracterizan un montaje de multiplexación temporal y que servirán para relacionar las medidas tomadas sobre las imágenes con la distancia del espacio físico.

Una vez finalizada la captura, se deben detectar las pendientes de las estructuras plenópticas que, para esta configuración, se relacionan

### 3.1 ADQUISICIÓN MEDIANTE MULTIPLEXACIÓN TEMPORAL



**Figura 3.1:** Esquema del montaje óptico de la multiplexación temporal.

con la distancia  $z$  través de la siguiente expresión:

$$z = \frac{1}{\hat{s}} \frac{\Delta_{\text{cam}}}{\Delta p} a \quad (3.1)$$

donde  $\Delta_{\text{cam}}$  es la separación entre puntos de vista,  $\Delta p$  es el tamaño del píxel,  $a$  es la separación entre el centro óptico de la cámara y el sensor asumiendo un modelo de cámara *pin-hole*, y  $\hat{s}$  es la pendiente medida en píxeles por vista, estimada a partir de los datos adquiridos.

En este caso, se ha adquirido un conjunto de *light fields* mediante una cámara Nikon D60 con un objetivo AF-S Nikkor 18-55 mm 1:3.5-5.6 G. El sensor de esta cámara permite adquirir imágenes de  $3872 \times 2592$  píxeles, siendo el tamaño del píxel de  $6,09\mu\text{m}$ . Para mover con precisión la cámara y adquirir las vistas necesarias se ha utilizado un micrómetro X-Z, modelo Ultralign 651D del fabricante Newport. La distancia entre vistas fue de  $500\mu\text{m}$ , tanto en horizontal como en



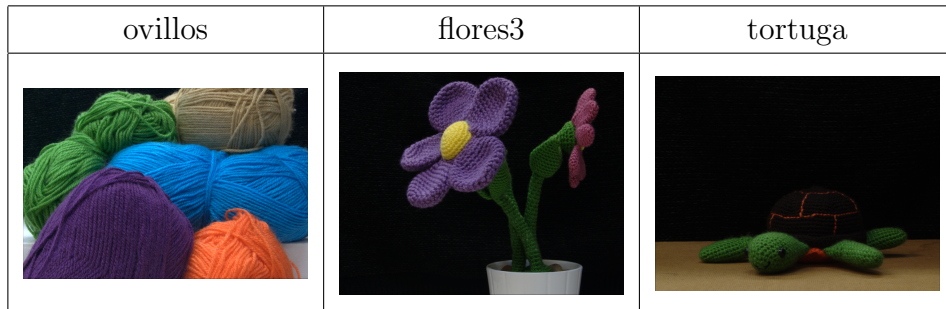
**Figura 3.2:** Fotografía del montaje utilizado para adquirir *light fields* mediante multiplexación temporal.

vertical. Con este método se obtuvieron 3 *light fields* de  $5 \times 5$  puntos de vista, que se utilizarán para obtener resultados sobre imágenes reales con pocas muestras angulares. En la figura 3.2 se muestra una fotografía del montaje.

La resolución espacial de cada uno de los puntos de vista se ha recortado para obtener un encuadre adecuado de la escena y posteriormente se ha reducido su tamaño mediante un filtro de media.

En la figura 3.3, se muestra la imagen central de cada uno de los *light fields* adquiridos, junto con la etiqueta asignada, con el fin de facilitar su identificación posterior. Por otro lado, en la tabla 3.1 se muestran las dimensiones de cada uno de ellos y las condiciones particulares en las que fueron adquiridos.

### 3.1 ADQUISICIÓN MEDIANTE MULTIPLEXA-CIÓN TEMPORAL



**Figura 3.3:** *Imágenes centrales de los tres light fields adquiridos por multiplexación temporal.*

| LF      | Res. espacial | Res. angular | Tiempo de exposición | Número F# | Focal   | Tam. filtro |
|---------|---------------|--------------|----------------------|-----------|---------|-------------|
| ovillos | 954 × 654     | 5 × 5        | 1.6 seg.             | F/32      | 55.0 mm | 4 × 4       |
| flores3 | 726 × 567     | 5 × 5        | 4 seg.               | F/25      | 38.0 mm | 4 × 4       |
| tortuga | 641 × 483     | 5 × 5        | 5 seg.               | F/25      | 55.0 mm | 5 × 5       |

**Tabla 3.1:** *Dimensiones y condiciones de adquisición de los light fields adquiridos con multiplexación temporal.*

Para que las imágenes estuvieran enfocadas en el mayor rango posible se utilizaron números F# grandes y se aumentó el tiempo de exposición.

## 3.2. Light fields obtenidos mediante la cámara CAFADIS

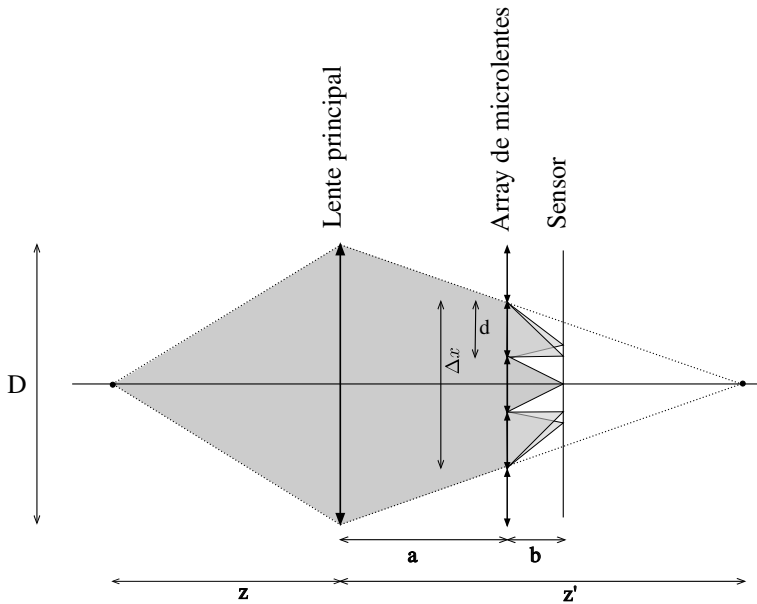
La cámara CAFADIS es una cámara plenóptica que se viene desarrollando durante los últimos años en la Universidad de La Laguna, con el objetivo que sirva tanto como sensor de distancias como de sensor de la fase del frente de onda. Tal y como se ha descrito en la sección 2.2.3, se trata de un método de multiplexación espacial en el que sobre el mismo sensor se reparten muestras espaciales y angulares. Esto hace que la resolución espacial de cada uno de los puntos de vista sea mucho menor que la que se puede obtener con otros métodos como los de multiplexación temporal que utilizan todo el sensor para cada punto de vista.

A continuación, se describen los parámetros del montaje óptico de una cámara plenóptica y, seguidamente, se detallan los *light fields* adquiridos y las condiciones de su adquisición.

### 3.2.1. Descripción del montaje óptico

La apariencia de una cámara plenóptica no es muy distinta de la de una cámara fotográfica convencional, ya que se conservan las partes esenciales del montaje óptico: Una lente principal y un sensor. La diferencia estriba en que entre ambos componentes se emplaza un array de microlentes, como se muestra en la figura 3.4. La función de

## 3.2 LIGHT FIELDS OBTENIDOS MEDIANTE LA CÁMARA CAFADIS



**Figura 3.4:** Esquema del montaje óptico de la cámara plenóptica.

este array de microlentes es multiplexar la luz incidente en cada punto desde distintas direcciones sobre el sensor bidimensional.

La cámara plenóptica mide el *light field* en el interior de la cámara y consecuentemente el *light field* de una porción del mundo 3D en el exterior de la misma. El *light field* en el interior de la cámara se denota como  $L(\mathbf{x}, \mathbf{u})$ , siendo  $\mathbf{x}$  la posición en el plano de las microlentes y  $\mathbf{u}$  la posición en el plano de la lente principal.

El patrón de muestreo del *light field* continuo, tanto en el interior como en el exterior de la cámara, depende de los parámetros del sistema óptico. En la tabla 3.2 se describen los parámetros implicados

| Parámetro  | Descripción   |
|------------|---|
| $D$        | Diámetro de la lente principal  |
| $d$        | Diámetro de cada microlente   |
| $a$        | Distancia entre la lente principal y el array de microlentes                                    |
| $b$        | Distancia entre las microlentes y el sensor   |
| $F$        | Longitud focal de la lente principal  |
| $f$        | Longitud focal de las microlentes   |
| $z$        | Distancia entre la lente principal y el objeto enfocado   |
| $z'$       | Distancia entre la lente principal y la imagen del objeto a través de sólo esta lente           |
| $\Delta x$ | Ancho sobre el plano de las microlentes del cono generado por el objeto situado a distancia $z$ |

**Tabla 3.2:** Descripción de los parámetros ópticos de la cámara plenóptica



## 3.2 LIGHT FIELDS OBTENIDOS MEDIANTE LA CÁMARA CAFADIS

---

y en la figura 3.4 se muestran asociados con el montaje óptico.

La distancia entre el array de microlentes y el sensor, denotada con  $b$ , cambia el patrón de muestreo del *light field* en el interior de la cámara. Por otro lado, la distancia entre la lente principal y el array de microlentes, denotada con  $a$ , afecta al patrón de muestreo en el exterior de la cámara, puesto que esta distancia determina el mapeo entre el *light field* delante a la lente principal y el *light field* detrás de la lente principal.

En función de la distancia  $b$ , se pueden distinguir los dos casos que ya se mencionaron en la sección 2.2.3. El primer caso es la cámara plenóptica 1.0, que se obtiene cuando  $b = f$  y el segundo es la cámara plenóptica 2.0 que se obtiene cuando  $b \neq f$ . El efecto de esta modificación es que cambia el patrón de muestreo del *light field* en el interior de la cámara. En el caso de la cámara plenóptica 1.0, que será la que se emplee en este trabajo, el resultado de la medida es un conjunto de valores de intensidad, cada uno correspondiente a una de las distintas direcciones de los rayos de luz incidentes sobre un punto espacial  $\mathbf{x}$  sobre el plano de las microlentes. Cada píxel tras una microlente captura la intensidad de la luz incidente en la posición de la microlente en  $\mathbf{x}$  y que proviene desde una dirección.

El efecto de cambiar  $a$  es mover la posición del plano conjugado de las microlentes, lo cual también puede conseguirse modificando la distancia focal de la lente principal  $F$ . Considerando el caso de la lente delgada y asumiendo una imagen enfocada a la distancia  $z'$  tras la lente principal, el efecto de modificar  $a$  o  $F$  hace que también cambie la

distancia  $z$  delante de la lente principal, colocando el plano de enfoque a otra distancia.

Un aspecto a tener en cuenta para la adquisición de *light fields*, es que las microimágenes formadas tras cada una de las microlentes deben ser nítidas. Esto es necesario para poder reconstruir correctamente las imágenes de color enfocadas, ya que sólo se podrá obtener una imagen enfocada aplicando el operador fotográfico, descrito en la sección 2.7, sobre microimágenes nítidas. Además, para asegurar el máximo aprovechamiento del sensor, al mismo tiempo que evitamos la superposición indeseable de microimágenes, conviene hacer iguales los números  $F\#$  de trabajo de la lente principal y de las microlentes (Ng et al. 2005; Perwass y Wietzke, 2012).

### 3.2.2. Relación entre pendientes y distancias

Como ya se ha dicho, la imagen capturada mediante una cámara plenóptica es un *light field* de la escena, parametrizado con dos planos y mapeado sobre un plano imagen con una disposición como la que se mostró en la figura 2.4(a). Para obtener una estimación de distancias a partir de la imagen plenóptica es necesario detectar la pendiente de las estructuras planas definidas en la ecuación 2.1.

El plano delantero de la parametrización se corresponde con el plano de la lente principal y el plano trasero coincide con el plano de las microlentes (ver figura 3.4). La pendiente de la recta en el *light field* 2D discreto, viene dada por la expresión  $s = \frac{\Delta x}{\Delta u}$ , donde  $\Delta u$  es igual a

## 3.2 LIGHT FIELDS OBTENIDOS MEDIANTE LA CÁMARA CAFADIS

---

$D$  y  $\Delta x$  es la distancia que abarca, sobre el plano de las microlentes, el cono de luz generado por un punto situado a una cierta distancia  $z$  frente a la lente. La imagen de este punto generada por la lente principal estaría a una distancia  $z'$  tras la misma.

Suponiendo que se dispone del valor de la pendiente  $s$ , su relación con la posición de la imagen,  $z'$ , se puede obtener fácilmente observando la figura 3.4 y aplicando la siguiente semejanza de triángulos.

$$\frac{\Delta u}{z'} = \frac{\Delta x}{z' - a} \quad (3.2)$$

Reordenando la ecuación 3.2 se obtiene:

$$\frac{1}{z'} = \frac{1}{a} (1 - s) \quad (3.3)$$

Por último, si definimos  $\alpha$  como la relación<sup>1</sup> entre  $z'$  y  $a$  se obtiene que  $s = (1 - \frac{1}{\alpha})$  y  $z' = a\alpha$ .

Una vez se conoce la posición de la imagen respecto de la lente principal es sencillo mapearla hacia el espacio 3D, conociendo la configuración óptica de la lente principal, que normalmente es un sistema de lentes.

Es importante señalar que la medida de  $s$  se hará en una escala distinta, correspondiente a la de los píxeles de la imagen plenóptica, por lo que se medirá en píxeles/vista. Esta medida se denotará con  $\widehat{s}$

---

<sup>1</sup>En Ng et al. (2005)  $\alpha$  se define como  $\frac{F'}{F}$ , siendo  $F'$  el equivalente de  $z'$

y el valor de  $s$  en función de  $\hat{s}$  es aproximadamente:

$$s = \hat{s} \cdot \frac{d}{D} N_{uv} \quad (3.4)$$

Esta expresión implica los siguientes elementos: el número de puntos de vista, que coincide con el número de píxeles tras una microlente,  $N_{uv}$ ; el tamaño de cada píxel, que coincide con el diámetro de las microlentes,  $d$ ; y finalmente, depende del diámetro de la lente principal,  $D$ .

### 3.2.3. Descripción de los *light fields* adquiridos

Se ha utilizado el prototipo de cámara plenóptica CAFADIS, mencionado en la sección 2.2.2, para adquirir datos de prueba con los que validar los métodos desarrollados en esta tesis doctoral.

El prototipo está compuesto por una cámara de cine RED One (Red Digital Cinema, 2011), cuyo sensor permite obtener imágenes de  $4096 \times 2304$  píxeles. Las características de las microlentes empleadas se detallan en la tabla 3.3. Finalmente, la lente principal empleada fue el Red Zoom Lens 50-150mm T2.8 Compact, cuyas características básicas se detallan en la tabla 3.4.

La adquisición de la imagen se realizó ajustando la focal de la lente principal a 150 mm y conjugando el plano de las microlentes a 1 m. Frente a la lente se situó una escena compuesta por muñecos de plástico, obteniendo un *light field* con  $N_x = 166$ ,  $N_y = 93$  y  $N_{uv} = 24$ . Debe tenerse en cuenta que parte de las microlentes del array no se utilizaron puesto que se sobrepasaba la superficie del sensor. Además,

### 3.2 LIGHT FIELDS OBTENIDOS MEDIANTE LA CÁMARA CAFADIS

---

|                             |                     |
|-----------------------------|---------------------|
| Material                    | <i>Fused Silica</i> |
| Focal trasera               | 380 $\mu$ m         |
| Focal delantera             | 1000 $\mu$ m        |
| F <sub>#</sub>              | 8.3                 |
| Distancia entre microlentes | 130 $\mu$ m         |
| Tamaño del array            | 31 $\times$ 21mm    |

**Tabla 3.3:** Características de las microlentes utilizadas en el montaje de la cámara plenóptica.

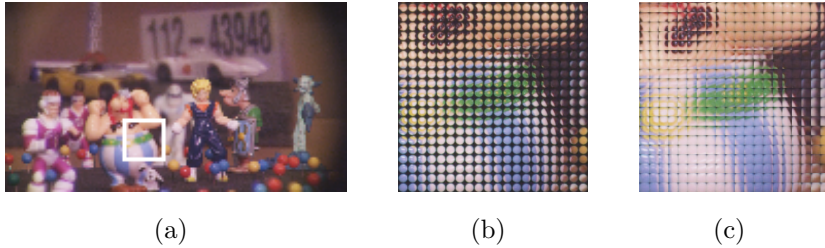
|                  |           |
|------------------|-----------|
| Longitud focal   | 50-150 mm |
| T-Stop           | T2.8      |
| Enfoque mínimo   | 1 m       |
| Diámetro frontal | 76 mm     |

**Tabla 3.4:** Características de la lente principal utilizada en el montaje de la cámara plenóptica.

las microlentes de los bordes están afectadas por viñeteo. Este primer *light field* se denominará muñecos-24.

Con el fin de comprobar la influencia de los efectos de borde de las microlentes, se generó un segundo *light field*, que se obtuvo, a partir de la misma imagen plenóptica que el primero, recortando un cuadrado en el interior de las imágenes de microlentes circulares. En este caso el valor de  $N_{uv}$  fue 15 y el *light field* se denominará muñecos-15.

La imagen central junto con una vista detallada de los *light fields* muñecos-24 y muñecos-15 se muestra en la figura 3.5.



**Figura 3.5:** Light field obtenido con la cámara CAFADIS. (a) Imagen central. (b) Detalle del light field *muñecos-24*. (c) Detalle del light field *muñecos-15*.

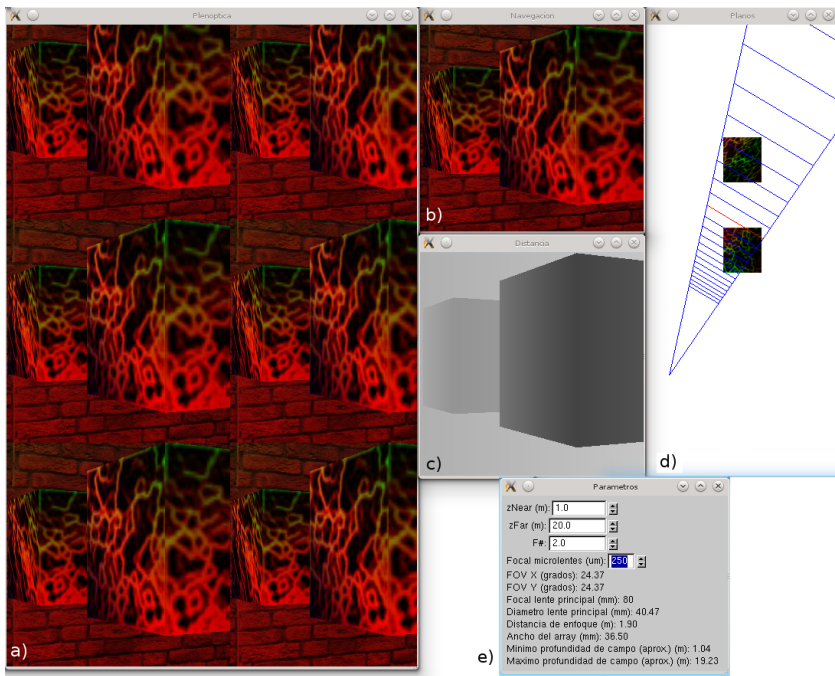
### 3.3. *Light fields* obtenidos mediante el simulador de escenas virtuales (SIDEV)

Para validar las medidas de distancia obtenidas por los algoritmos, será necesario emplear *light fields* sintéticos acompañados de un mapa de distancias. Para generar estos datos, se ha utilizado un simulador de entornos virtuales (SIDEV), que permite cargar una escena virtual en formato VRML y navegar a través de la misma. Mientras el usuario navega se van generando el *light field* de la vista actual y el mapa de distancias correspondiente.

El desarrollo del SIDEV fue iniciado por el Dr. J.G. Marichal-Hernández en el año 2004 y ha sido modificado a lo largo del tiempo. Por ejemplo, durante el desarrollo de esta tesis doctoral, se ha añadido un módulo que permite introducir algunos parámetros ópticos de la

### 3.3 *Light fields* OBTENIDOS MEDIANTE EL SIMULADOR DE ESCENAS VIRTUALES (SIDEV)

cámara plenóptica a simular, con el fin de producir el mismo resultado que produciría una cámara real con esos parámetros. Además se consiguió que los parámetros ópticos permitieran abarcar todas las profundidades de la escena generando pendientes dentro del rango  $[-1, 1]$ .



**Figura 3.6:** Interfaz gráfica del simulador de cámara plenóptica: a) Múltiples vistas. b) Ventana de navegación. c) Mapa de distancias verdadero. d) Vista cenital de los objetos de la escena. e) Ventana para el manejo de parámetros de la cámara plenóptica.

En la figura 3.6 se puede observar una captura de pantalla del simulador de cámara plenóptica. Esta interfaz permite navegar de

forma interactiva por la escena como si de un videojuego se tratase y realizar capturas de imágenes plenópticas sintéticas, junto con su mapa de distancias y los parámetros de la cámara en ese momento. También dispone de una ventana para ajustar los parámetros de la cámara plenóptica. Además se ha añadido una vista cenital de la escena que facilita la navegación.

La implementación de este simulador se basa en la librería OpenGL (The Kronos Group, 2013). Para simular la cámara plenóptica, se calculan los parámetros de un array de cámaras equivalente a la cámara plenóptica que se desea simular. En cada toma de la escena se desplaza una cámara *pin-hole* por las distintas posiciones del array de cámaras y se ajustan los parámetros de la misma para tomar las imágenes necesarias. Una vez capturados todos los puntos de vista, es necesario realizar una rotación como la descrita en la sección 2.1.1 con el fin de obtener una imagen plenóptica final.

Para expresar los parámetros de la cámara OpenGL en función de los de la cámara plenóptica, se asumirá que ésta consta de un array de  $N_x \times N_y$  microlentes, que tras cada microlente se encuentran  $N_{uv} \times N_{uv}$  píxeles y que se conocen los valores de los parámetros ópticos de la cámara plenóptica (tabla 3.2). A partir de estos parámetros se determina la resolución en píxeles, la distancia focal, la posición, el campo de visión y el *frustum* a aplicar, para cada cámara OpenGL.

El número de puntos de vista a adquirir es de  $N_{uv} \times N_{uv}$  y el número de píxeles de cada una de las cámaras es  $N_x \times N_y$ . La distancia focal de cada una de las cámara estenopéicas es igual a la distancia  $a$  entre



### 3.3 *Light fields* OBTENIDOS MEDIANTE EL SIMULADOR DE ESCENAS VIRTUALES (SIDEV)

---

la lente principal y el sensor.

Las posiciones desde las que se toma cada una de las imágenes de cada punto de vista, se distribuyen sobre el plano de la lente principal. La separación entre ellas dependerá del diámetro de la apertura  $D$  y del número de píxeles detrás de cada microlente, de la siguiente manera:

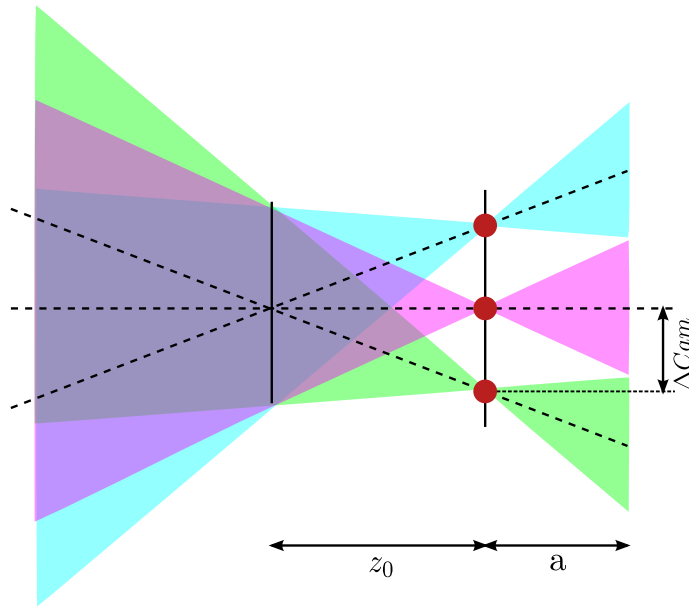
$$\Delta Cam = \frac{D}{N_{uv}} \quad (3.5)$$

Es necesario imponer una restricción adicional sobre el *frustum* de cada una de las cámaras para que el sensor de cada una de ellas esté desplazado con respecto al eje óptico, de forma que exista una línea que parte del centro del sensor hasta el punto de enfoque, pasando por el centro de la cámara *pin-hole* (línea punteada de la figura 3.7). Esto hará que, a la distancia conjugada del plano de las microlentes, el campo visual de todas las cámaras sea coincidente y que el campo de visión de cada cámara sea asimétrico respecto de su eje óptico, como se puede apreciar la figura 3.7.

Finalmente, el ángulo de visión horizontal total de cada cámara, aunque su campo visual sea asimétrico, viene dado por la siguiente expresión:

$$fov_x = 2 \arctan \left( \frac{(N_x - 1) \cdot d}{2a} \right) \quad (3.6)$$

Utilizando la técnica anteriormente descrita se han obtenido dos conjuntos de datos simulando una cámara plenóptica con los parámetros ópticos que se muestran en la tabla 3.5. Con esta



309

**Figura 3.7:** Esquema de un array de cámaras pin-hole equivalente a la cámara plenóptica. Los círculos rojos son el centro de proyección de las cámaras con focal  $a$ .

configuración óptica las pendientes del rango  $[-1, 1]$  quedan confinadas entre 1 metro y 20 metros. Los objetos de la escena son dos cubos de 50 centímetros de lado con una pared al fondo.

El primer conjunto de datos, que se muestra en la figura 3.8, es un *light field* estático con  $292 \times 292$  microlentes y  $13 \times 13$  píxeles tras cada una de ellas. El segundo es una secuencia de 21 imágenes de la misma escena pero la cámara realiza un movimiento de traslación durante los 10 primeros *frames* y un movimiento rotacional durante los 10 siguientes.

### 3.4 CONJUNTO DE DATOS DEL *Heidelberg Collaboratory for Image Processing*

---

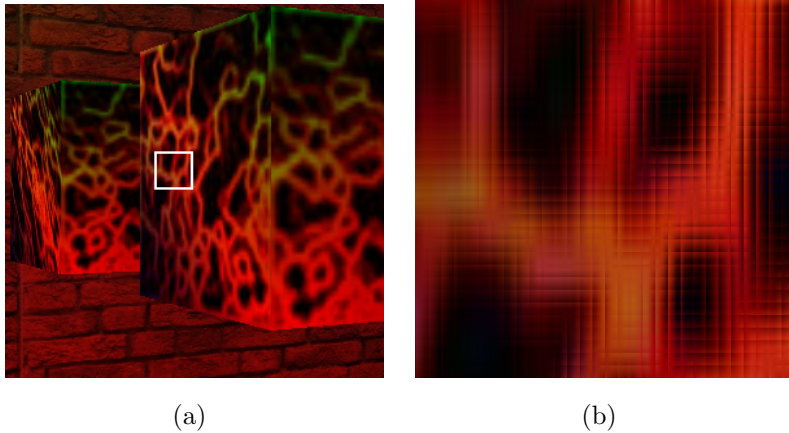
| Parámetro | Valor             |
|-----------|-------------------|
| D         | 40.27 mm          |
| F         | 80 mm             |
| f         | 500 $\mu\text{m}$ |
| d         | 250 $\mu\text{m}$ |
| b         | 500 $\mu\text{m}$ |
| F#        | 2.0               |

**Tabla 3.5:** *Parámetros de la cámara plenóptica simulada.*

## 3.4. Conjunto de datos del *Heidelberg Collaboratory for Image Processing*

Además de los datos generados con el SIDEV, se ha decidido recurrir al conjunto de datos elaborado en el *Heidelberg Collaboratory for ImageProcessing* (HCI) para la evaluación de algoritmos aplicables a *light fields* (Wanner et al. 2013). Se trata del único conjunto públicamente disponible que, además de los *light fields*, proporciona información de distancias, lo que permitirá comparar el método desarrollado con los resultados de otros trabajos que utilizan este conjunto de datos.

El conjunto de datos está formado por siete *light fields* sintéticos generados con Blender (Blender Foundation, 2013) y cinco imágenes reales obtenidas mediante un pórtico de cámaras (véase la sección



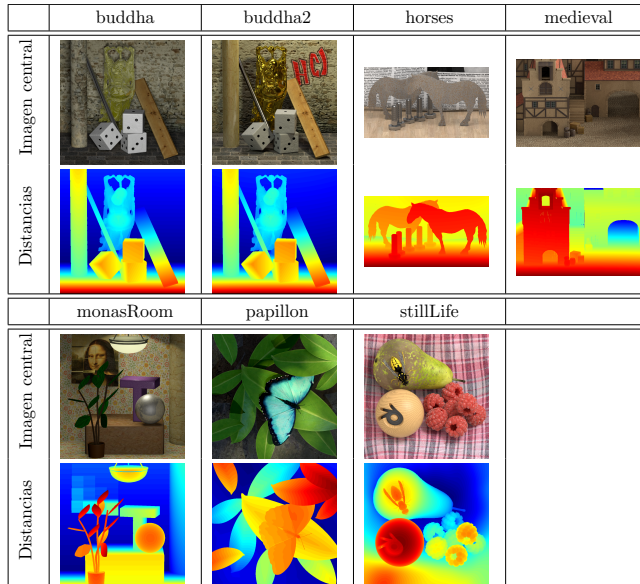
**Figura 3.8:** Light field obtenido con el SIDEV. (a) Imagen central. (b) Detalle del light field de los cubos.

2.2.2). Todos los *light fields* del conjunto vienen acompañados de los mapas de distancias para cada una de las vistas, así como de los metadatos necesarios para su manipulación, que se describen detalladamente en el trabajo de Wanner et al. (2013).

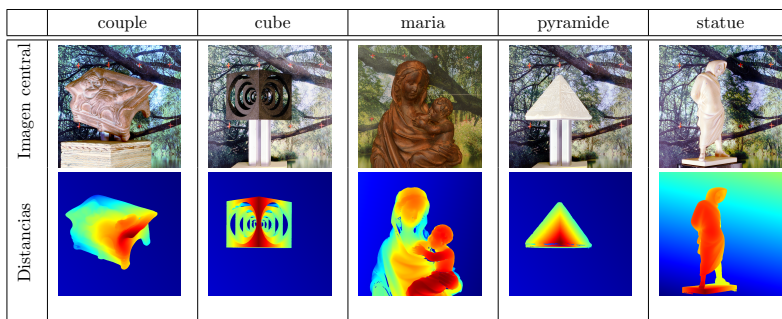
Las figuras 3.9 y 3.10 muestran la imagen central de cada uno de los *light fields*, así como el mapa de distancias asociado. Además, la tabla 3.6 muestra los tamaños en píxeles de cada una de las imágenes del *light field*, así como la pendiente mínima y máxima presente en cada uno de ellos. Todas las escenas fueron tomadas con  $9 \times 9$  puntos de vista, aunque para la escena *medieval* y para el grupo de escenas tomadas con el pórtico el número de vistas se ha limitado a  $7 \times 7$  para evitar un desbordamiento de memoria en la GPU.

### 3.4 CONJUNTO DE DATOS DEL *Heidelberg Collaboratory for Image Processing*

---



**Figura 3.9:** Imagen central y mapas de distancias para los light fields sintéticos del HCI dataset.



**Figura 3.10:** Imagen central y mapas de distancias para los light fields reales del HCI dataset.

| LF        | Tamaño   | $\hat{s}_{\min}$ | $\hat{s}_{\max}$ |
|-----------|----------|------------------|------------------|
| buddha    | 768x768  | -0,85            | 1,54             |
| buddha2   | 768x768  | -0,70            | 1,20             |
| horses    | 1024x576 | -3,70            | 0,94             |
| medieval  | 1024x720 | -1,93            | 0,89             |
| monasRoom | 768x768  | -0,79            | 0,72             |
| papillon  | 768x768  | -1,17            | 0,89             |
| stillLife | 768x768  | -2,71            | 2,56             |
| couple    | 898x898  | -1,11            | 1,25             |
| cube      | 898x898  | -1,11            | 0,80             |
| maria     | 926x926  | -0,60            | 0,43             |
| pyramide  | 898x898  | -1,11            | 0,79             |
| statue    | 898x898  | -2,27            | 0,54             |

**Tabla 3.6:** Dimensiones de los mapas de pendientes del HCI Dataset, junto con la pendiente máxima y mínima en cada uno. El primer grupo son las escenas generadas con Blender y el segundo grupo son las obtenidas con el pórtico.

# Capítulo 4

## Estimación de distancias mediante un detector de pendientes en 4D

*“Be conservative in what you do, be liberal in what you accept from others.”*

---

Jon Postel

Como se ha indicado en las secciones 3.1 y 3.2.2, existe una relación matemática entre la pendiente de las estructuras que los objetos de la escena generan en el *light field* y su distancia. El problema de estimar la distancia a la que se encuentra un punto se reduce por tanto a estimar las pendientes  $\hat{s}$  de la estructura que genera a partir de los datos del *light field*. Al estimar las pendientes sobre el *light field* discreto, éstas se expresan en píxeles/vista, existiendo así una analogía con las

disparidades estimadas en los métodos estéreo.

En este capítulo se desarrollará un nuevo método para estimar las pendientes  $\hat{s}$ , utilizando para ello un operador local que favorezca su paralelización y que simultáneamente nos permita evaluar la confianza de la estimación. Al ser el *light field* parametrizado por dos planos, una función de cuatro dimensiones, el objetivo del operador en cada posición será estimar las pendientes de las estructuras planas asociadas a cada punto visible del objeto usando información del entorno. En contraste a los métodos de detección bidimensionales, como los descritos en la sección 2.5.2, se utilizarán operadores extendidos a cuatro dimensiones. Posteriormente, se realizará un análisis de error para ambos operadores que nos permitirá su comparación.

Como veremos, los operadores locales permiten obtener en la práctica una medida por cada una de las posiciones del *light field*. La combinación de estas medidas redundantes con el fin de obtener un único mapa de distancias se abordará al final de este capítulo donde también se discuten algunos detalles de la implementación del algoritmo sobre GPU.

Además, cabe destacar una característica interesante que presentan los métodos locales, como el que se desarrollará en este capítulo, frente a métodos previos, como el propuesto en Pérez Nava y Lüke (2009b). Este método asume que, al tratarse de una escena lambertiana, todos los píxeles del plano 4D deben ser iguales y compara la totalidad de los píxeles de un conjunto de datos candidatos para determinar qué plano se aproxima más a la hipótesis lambertiana. Esto obliga a



que la BRDF (Bidirectional Reflectance Distribution Function) de los materiales de la escena sea estrictamente lambertiana. Sin embargo, con el enfoque local sólo es necesario que la hipótesis lambertiana se cumpla localmente, admitiendo así una variedad de BRDFs más amplia.

### 4.1. Detección de estructuras planas en cuatro dimensiones

La detección de haces de planos en un espacio de cuatro dimensiones puede hacerse por medio del método propuesto para los haces de líneas en dos dimensiones en Bigün y Granlund (1987) o con el propuesto en Di Zenzo (1986) con el mismo propósito. En el primero se trabaja directamente en el espacio de los datos mediante el tensor de momentos de inercia locales de la imagen y en el segundo en el espacio gradiente mediante su tensor de correlación local, también denominado tensor de estructura local de la imagen. La ventaja de trabajar en el espacio gradiente es que aporta una mayor insensibilidad a los valores absolutos de la imagen y buscando incluir esta ventaja es por lo que se decide la utilización de este método como base para la generalización a las cuatro dimensiones. De modo que para la aplicación generalizada de este método se debe calcular el tensor de estructura del *light field* en cuatro dimensiones. Como veremos, el estudio de los autovalores de este tensor nos permitirá determinar la presencia de estructuras planas.

El tensor de estructura sobre el *light field* de cuatro dimensiones se define para cada punto  $\mathbf{p} = (x, y, u, v)$ , como sigue:

$$T_L(\mathbf{p}) = \begin{pmatrix} \widehat{L_x^2} & \widehat{L_x L_y} & \widehat{L_x L_u} & \widehat{L_x L_v} \\ \widehat{L_y L_x} & \widehat{L_y^2} & \widehat{L_y L_u} & \widehat{L_y L_v} \\ \widehat{L_u L_x} & \widehat{L_u L_y} & \widehat{L_u^2} & \widehat{L_u L_v} \\ \widehat{L_v L_x} & \widehat{L_v L_y} & \widehat{L_v L_u} & \widehat{L_v^2} \end{pmatrix} \quad (4.1)$$

donde  $\widehat{L_{a_i} L_{a_j}} = \omega(\mathbf{r}) * (L_{a_i}(\mathbf{p})L_{a_j}(\mathbf{p}))$ ,  $L_{a_n}(\mathbf{p}) = \frac{\partial L}{\partial a_n} |_{\mathbf{p}}$  y  $a_i \in [x, y, u, v]$ . La función  $\omega(\mathbf{r})$  es una función de peso de cuatro dimensiones que balanceará la importancia en el cálculo de los puntos en torno al punto bajo análisis.

Si estudiamos los autovalores del tensor y su relación con las estructuras en el entorno del punto de estudio podemos identificar cuatro tipos de estructuras. Para ello, ordenando los autovalores de mayor a menor:  $\lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \lambda_3$ , obtenemos:

- Hiperplano: Cuando  $\lambda_0 > 0, \lambda_1 = \lambda_2 = \lambda_3 = 0$ , el entorno del punto bajo análisis presenta una estructura de hiperplanos tridimensionales, paralelos con intensidades constantes y distintas.
- Plano: Cuando  $\lambda_0 > 0, \lambda_1 > 0, \lambda_2 = \lambda_3 = 0$ , el entorno del punto bajo análisis es de estructura de plano, es decir, un haz de planos paralelos con intensidades constantes y distintas.
- Línea: Cuando  $\lambda_0 > 0, \lambda_1 > 0, \lambda_2 > 0, \lambda_3 = 0$ , el entorno del

## 4.1 DETECCIÓN DE ESTRUCTURAS PLANAS EN CUATRO DIMENSIONES

---

punto bajo análisis tiene una estructura de línea en el espacio de cuatro dimensiones, entendida como un haz de líneas paralelas con intensidades constantes y distintas.

- Isótropa: Cuando  $\lambda_0 = \lambda_1 = \lambda_2 = \lambda_3 = 0$ . el entorno del punto bajo análisis tiene una estructura isótropa, es decir, que su intensidad es constante en todo el entorno.

Suponiendo una escena lambertiana y sin oclusiones, en su *light field*, localmente sólo podrán encontrarse planos y regiones isótropas. Los planos serán como los definidos por la ecuación 2.1 y los puntos pertenecientes a los mismos aparecerán con la misma intensidad y color, mientras que las regiones isótropas corresponderán al apilamiento de varios planos asociados a puntos de la escena del mismo color e intensidad.

Si en esta escena se calculasen el tensor de estructura y sus autovalores, el número de autovalores distintos de cero nos indicará, en caso de ser dos, que se trata de un plano definido por los autovectores correspondientes a los autovalores nulos, y en el caso de que los cuatro autovalores fuesen nulos sería una región sin textura, y que por tanto, no se puede determinar la distancia.

La estructura plenóptica generada por la ecuación 2.1 es un plano, pero con una restricción adicional, que reduce el número de grados de libertad de dos a uno, ya que ambos vectores directores dependen de un mismo parámetro. Sin embargo, el método general detectaría cualquier plano sin tener en cuenta la ligadura existente entre los

vectores directores que sí se tendrá en cuenta aquí.

Para derivar el nuevo operador se utiliza un método de energías similar al empleado en Weijer (2005, pág. 37) para el detector de dos dimensiones, pero teniendo en cuenta las cuatro dimensiones y el modelo de vector gradiente de un *light field*, que establece una ligadura entre los vectores directores por medio de la orientación del plano.

Un vector gradiente en un punto de un *light field* ideal, que no esté afectado por ruido, puede ser descrito mediante una combinación lineal de dos vectores ligados por la distancia a la que se encuentra el punto del objeto y que se representa por  $s = \tan(\theta)$ , la pendiente de la estructura plenóptica subyacente, como se muestra en la siguiente ecuación:

$$\nabla L = \langle L_x, L_y, L_u, L_v \rangle = \alpha \langle \cos \theta, 0, \sin \theta, 0 \rangle + \beta \langle 0, \cos \theta, 0, \sin \theta \rangle \quad (4.2)$$

donde los vectores  $\langle \cos \theta, 0, \sin \theta, 0 \rangle$  y  $\langle 0, \cos \theta, 0, \sin \theta \rangle$  forman una base local del subespacio gradiente.

Extendiendo este subespacio generado por dos vectores con otros dos vectores ortogonales adicionales, se tiene una base completa del espacio gradiente:

$$\begin{aligned} e_1(\theta) &= \langle \cos \theta & 0 & \sin \theta & 0 \rangle^\top \\ e_2(\theta) &= \langle 0 & \cos \theta & 0 & \sin \theta \rangle^\top \\ e_3(\theta) &= \langle -\sin \theta & 0 & \cos \theta & 0 \rangle^\top \\ e_4(\theta) &= \langle 0 & -\sin \theta & 0 & \cos \theta \rangle^\top \end{aligned} \quad (4.3)$$

#### 4.1 DETECCIÓN DE ESTRUCTURAS PLANAS EN CUATRO DIMENSIONES

---

Ya que el objetivo del operador local es medir el valor del ángulo  $\theta$  que nos es desconocido, se plantea que el vector gradiente medido debe estar dentro de un subespacio gradiente como el descrito para algún ángulo. Se trata por tanto de definir una función cuadrática que en un extremo se ajuste al modelo.

Para ello, a partir de esta base local del espacio gradiente del punto  $\mathbf{p}$ , con pendiente  $s = \tan(\theta)$ , se define una energía, función del ángulo, correspondiente a la componente proyectada sobre el subespacio del modelo de la ecuación 4.2 y otra energía para la parte restante, que se interpreta como ruido de medida y errores de discretización, y por tanto se entiende como energía residual.

$$E_f(\theta) = \int (e_0(\theta)^\top \nabla L(\mathbf{r}))^2 \omega(\mathbf{r}) d\mathbf{r} + \int (e_1(\theta)^\top \nabla L(\mathbf{r}))^2 \omega(\mathbf{r}) d\mathbf{r} \quad (4.4)$$

$$E_r(\theta) = \int (e_2(\theta)^\top \nabla L(\mathbf{r}))^2 \omega(\mathbf{r}) d\mathbf{r} + \int (e_3(\theta)^\top \nabla L(\mathbf{r}))^2 \omega(\mathbf{r}) d\mathbf{r} \quad (4.5)$$

Reordenando y asociando adecuadamente los elementos de las ecuaciones 4.4 y 4.5 se obtiene:

$$\begin{aligned}
 E_f(\theta) = & e_0(\theta)^\top \left( \int \nabla L(\mathbf{r}) \nabla L(\mathbf{r})^\top \omega(\mathbf{r}) d\mathbf{r} \right) e_0(\theta) \\
 & + e_1(\theta)^\top \left( \int \nabla L(\mathbf{r}) \nabla L(\mathbf{r})^\top \omega(\mathbf{r}) d\mathbf{r} \right) e_1(\theta)
 \end{aligned} \tag{4.6}$$

$$\begin{aligned}
 E_r(\theta) = & e_2(\theta)^\top \left( \int \nabla L(\mathbf{r}) \nabla L(\mathbf{r})^\top \omega(\mathbf{r}) d\mathbf{r} \right) e_2(\theta) \\
 & + e_3(\theta)^\top \left( \int \nabla L(\mathbf{r}) \nabla L(\mathbf{r})^\top \omega(\mathbf{r}) d\mathbf{r} \right) e_3(\theta)
 \end{aligned} \tag{4.7}$$

donde  $\int \nabla L(\mathbf{r}) \nabla L(\mathbf{r})^\top \omega(\mathbf{r}) d\mathbf{r}$  no es más que el tensor de estructura en cada punto  $\mathbf{p}$ , definido en la ecuación 4.1, por lo que se puede escribir:

$$E_f(\theta) = e_0(\theta)^\top T_L e_0(\theta) + e_1(\theta)^\top T_L e_1(\theta) \tag{4.8}$$

$$E_r(\theta) = e_2(\theta)^\top T_L e_2(\theta) + e_3(\theta)^\top T_L e_3(\theta) \tag{4.9}$$

Desarrollando las ecuaciones anteriores y sustituyendo los valores del tensor de estructura y los vectores definidos en la ecuación 4.2 se obtiene que:

$$\begin{aligned}
 E_f(\theta) = & \left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) \cos^2(\theta) + \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right) \sin^2(\theta) \\
 & + 2 \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right) \sin(\theta) \cos(\theta)
 \end{aligned} \tag{4.10}$$

$$\begin{aligned}
 E_r(\theta) = & \left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) \sin^2(\theta) + \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right) \cos^2(\theta) \\
 & - 2 \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right) \sin(\theta) \cos(\theta)
 \end{aligned} \tag{4.11}$$

#### 4.1 DETECCIÓN DE ESTRUCTURAS PLANAS EN CUATRO DIMENSIONES

---

Seguidamente, las energías  $E_f(\theta)$  y  $E_r(\theta)$ , se expresan en función del ángulo doble <sup>1</sup>:

$$\begin{aligned}
 E_f(\theta) = & \frac{1}{2} \left( \widehat{L}_x^2 + \widehat{L}_y^2 + \widehat{L}_u^2 + \widehat{L}_v^2 \right) \\
 & + \frac{1}{2} \left( \left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) - \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right) \right) \cos(2\theta) \quad (4.12) \\
 & + \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right) \sin(2\theta)
 \end{aligned}$$

$$\begin{aligned}
 E_r(\theta) = & \frac{1}{2} \left( \widehat{L}_x^2 + \widehat{L}_y^2 + \widehat{L}_u^2 + \widehat{L}_v^2 \right) \\
 & - \frac{1}{2} \left( \left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) - \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right) \right) \cos(2\theta) \quad (4.13) \\
 & - \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right) \sin(2\theta)
 \end{aligned}$$

Conocida la expresión para ambas energías se define una función que mida la cantidad relativa de energía es residual y cual es válida, cuyo extremo máximo nos indicará el ángulo adecuado. Por otro lado esta función nos servirá como medida de la confianza en el dato medido en su ajuste al modelo de *light field*. Su expresión a partir de las energías es:

$$C(\theta) = \frac{E_f(\theta) - E_r(\theta)}{E_f(\theta) + E_r(\theta)} \quad (4.14)$$

Sustituyendo los valores de ambas energías la medida de confianza

---

<sup>1</sup>Se aplican las relaciones trigonométricas:  $\cos^2(\theta) = \frac{1+\cos(2\theta)}{2}$ ,  $\sin^2(\theta) = \frac{1-\cos(2\theta)}{2}$  y  $\sin(2\theta) = 2 \sin(\theta) \cos(\theta)$

queda:

$$C(\theta) = \frac{\left( \left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) - \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right) \right) \cos(2\theta) + 2 \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right) \sin(2\theta)}{\widehat{L}_x^2 + \widehat{L}_y^2 + \widehat{L}_u^2 + \widehat{L}_v^2} \quad (4.15)$$

A continuación se optimiza la orientación para obtener la máxima confianza, ya que ese ángulo será el que mejor se ajusta a los datos. El resultado de esta optimización es el siguiente:

$$\widehat{\theta} = \frac{1}{2} \arctan \left( \frac{2 \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right)}{\left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) - \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right)} \right) \quad (4.16)$$

Sustituyendo el valor del ángulo óptimo obtenido con la ecuación 4.16 en la ecuación 4.15 se obtiene que:

$$C^2 = \frac{a^2 + b^2}{d^2} \quad (4.17)$$

donde  $a = \left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) - \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right)$ ,  $b = 2 \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right)$  y  $d = \widehat{L}_x^2 + \widehat{L}_y^2 + \widehat{L}_u^2 + \widehat{L}_v^2$ .

## 4.2. Análisis de errores

En esta sección se presenta un análisis errores, tanto para el método de estimación de orientaciones sobre las imágenes epipolares, descrito



## 4.2 ANÁLISIS DE ERRORES

---

en la sección 2.5, como para el estimador de la orientación del *light field* 4D propuesto en la sección anterior. La finalidad de este análisis es comparar la resistencia al ruido de ambos métodos.

Las estimaciones de orientación obtenidas por ambos métodos están afectadas por errores sistemáticos que se introducen durante el cálculo de los gradientes cuando se utilizan operadores derivativos aproximados como los que se han descrito en la sección 2.5.1. Por otra parte, estas estimaciones están afectadas por errores aleatorios que dependen del ruido en las medidas del *light field*. Estos errores determinan la exactitud y la precisión de las medidas.

Denotando el error sistemático con  $\delta_{\hat{s}}$  y la desviación típica del error aleatorio con  $\sigma_{\hat{s}}$  el error total de la medida, que se denotará con  $E_{\hat{s}}$ , viene dado por la siguiente expresión:

$$E_{\hat{s}}^2 = \delta_{\hat{s}}^2 + \sigma_{\hat{s}}^2 \quad (4.18)$$

Una vez conocido el error cometido en la estimación de  $\hat{s}$ , se puede propagar el error a las medidas del mundo teniendo en cuenta el montaje óptico concreto empleado para su adquisición.

### 4.2.1. Estimación de orientación sobre las imágenes epipolares

En esta sección se presenta un análisis de los errores cometidos por el estimador de pendiente en imágenes epipolares descrito en la sección 2.5.2, suponiendo que para ello se emplea el filtro de derivada de  $3 \times 3$  de Farid y Simoncelli (2004). Dicho análisis, restringido al rango  $[-1, 1]$ , se ha publicado en una contribución a congresos (Lüke et al. 2013).

El error sistemático depende del contenido de frecuencias espaciales de la señal a derivar y, por tanto, no puede hallarse mediante una expresión analítica sencilla. Por ello, en Farid y Simoncelli (2004) se hace un estudio experimental, aplicando el estimador de orientación sobre un conjunto de datos en escala de grises con orientación y frecuencia conocida. En la figura 4.1 se ha representado este error para la estimación de la pendiente<sup>2</sup>,  $\hat{s}$ , en función de la frecuencia y la pendiente la imagen.

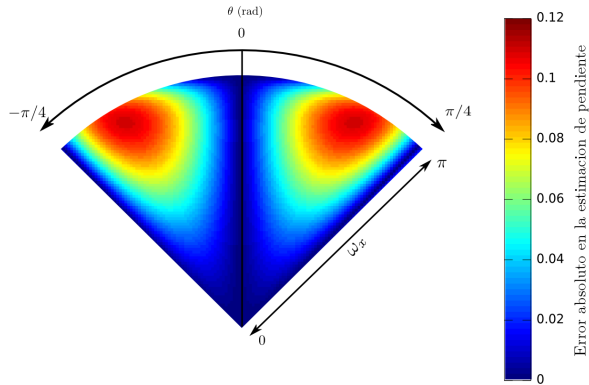
Dicha representación del error se ha obtenido aplicando el operador de la ecuación 2.10 sobre un conjunto de datos de entrada simulados mediante un patrón senoidal a distintas frecuencias y pendientes. En general, el error sistemático depende del contenido en frecuencias de la imagen y, por tanto, no es sencillo de predecir.

Además de estos errores sistemáticos, debidos a la discretización,

---

<sup>2</sup>En Farid y Simoncelli (2004) se muestra una figura similar con el error angular.

## 4.2 ANÁLISIS DE ERRORES



**Figura 4.1:** Representación de los errores sistemáticos del estimador de orientación 2D en función de la pendiente y la frecuencia.

se pueden producir errores aleatorios debidos al ruido en la imagen. Aunque en el trabajo de Farid y Simoncelli (2004) no se abordaron estos errores, juegan un papel en la robustez del método propuesto y por ello se estudiarán aquí. Para ello se ha asumido que los datos de entrada vienen afectados por ruido blanco gaussiano con desviación típica  $\sigma_I$  y se ha asumido que el numerador y el denominador del cociente de la ecuación 2.10 son estadísticamente independientes. Utilizando las expresiones para la propagación de errores de Ku (1966), en el apéndice B, se ha obtenido una expresión para calcular la desviación típica del error en la estimación de pendientes conocida la desviación típica del ruido en la imagen:

$$\sigma_{\hat{s}} = |\omega_{2D}| |p| |d_1| (1 + \hat{s}^2) \frac{\sigma_I}{\zeta} \quad (4.19)$$

donde  $\sigma_I$  es la desviación típica del ruido en la imagen y  $\zeta^2 =$

$d^2 = \widehat{I}_x^2 + \widehat{I}_y^2$ . Nótese además que esta expresión se ha obtenido asumiendo que todos los pesos de la operación  $\widehat{\cdot}$  son iguales y los valores promediados están decorrelados, con lo que  $|\omega_{4D}| = \frac{1}{\sqrt{N}}$  siendo  $N$  el número de puntos promediados.

Para comprobar la validez de la expresión de la ecuación 4.19, se han contrastado los resultados de ésta con los resultados obtenidos experimentalmente a partir de datos simulados.

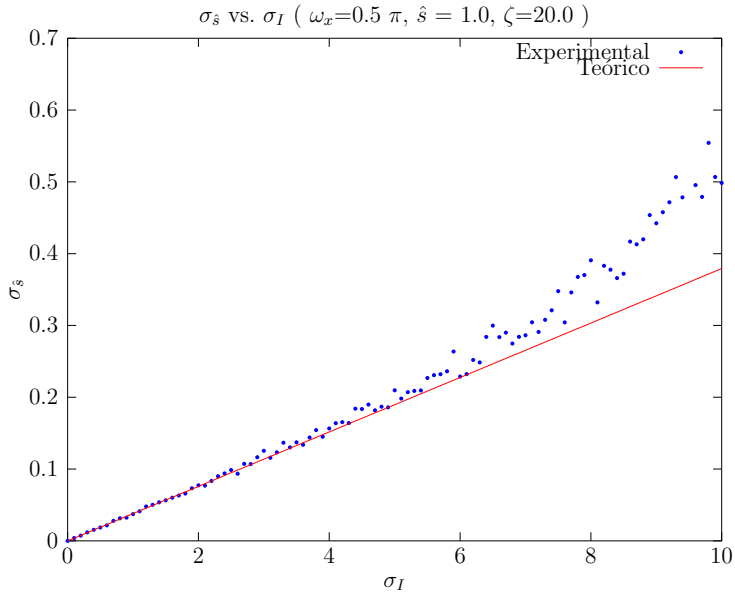
En la figura 4.2 se ha representado la desviación típica del error en la estimación de la pendiente frente a la desviación típica del ruido de la imagen para un patrón senoidal de frecuencia  $\omega_x = 0,5$  y con pendiente  $\widehat{s} = 1$ . Puede observarse como la desviación típica del error aumenta linealmente con la desviación típica del ruido, coincidiendo los datos experimentales con el comportamiento predicho por la ecuación 4.19.

Seguidamente, ha sido analizada la dependencia del error con  $\zeta$ . Se ha representado la desviación típica del error frente a  $\zeta$  manteniendo  $s = 1$  y  $\sigma_I = 5$ . En la figura 4.3, se puede observar el resultado obtenido, coincidiendo los datos experimentales con los teóricos.

Interpretando  $\zeta$  como una medida del contraste en el entorno del punto de la imagen, se puede concluir que la precisión a la hora de determinar la distancia es inversamente proporcional al contraste. Esto significa que a mayor contraste mayor precisión a la hora de determinar la pendiente y, por tanto, la distancia. El hecho de que  $\sigma_{\widehat{s}}$  tienda a infinito en las zonas con poca textura, concuerda con lo descrito en la

## 4.2 ANÁLISIS DE ERRORES

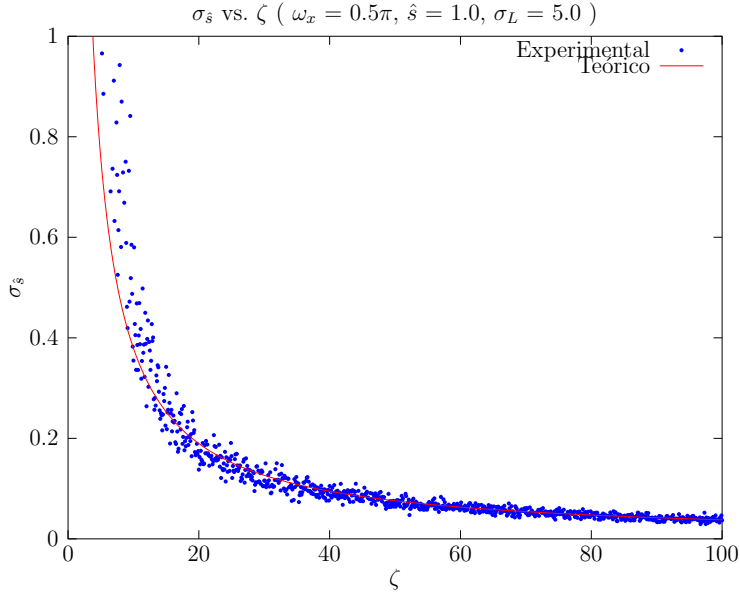
---



**Figura 4.2:** *Desviación típica del error aleatorio en la estimación de la pendiente en imágenes bidimensionales en función de la desviación típica de el radio de la imagen.*

sección 2.6, donde se indica que no es posible recuperar información de profundidades en estas zonas.

La última dependencia del error de estimación es la dependencia cuadrática con la pendiente. Para comprobar esta dependencia, se ha representado  $\sigma_{\hat{s}}$  frente a  $\hat{s}$  para el caso de  $\sigma_I = 5$  y  $\omega_x = 0,5$ . En la figura 4.4 se puede observar dicha representación y cómo los datos experimentales nuevamente coinciden con el comportamiento predicho por la expresión de la ecuación 4.19.

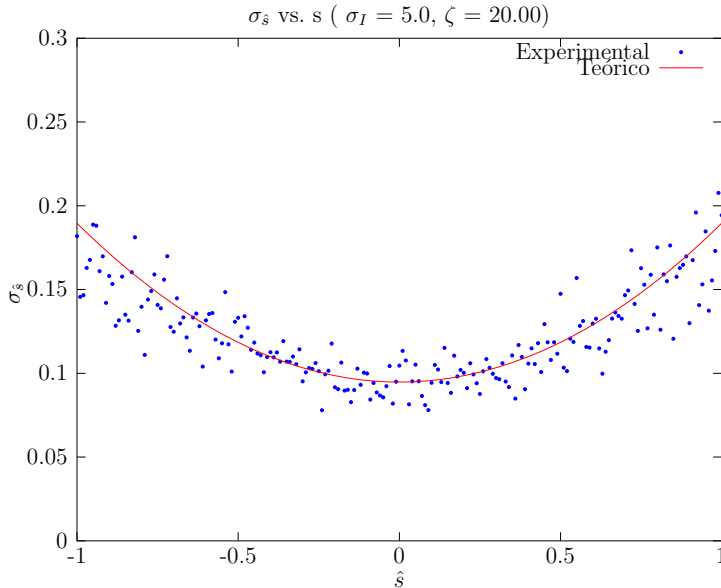


**Figura 4.3:** Desviación típica del error aleatorio en la estimación de la pendiente en imágenes bidimensionales en función de  $\zeta$ .

### 4.2.2. Estimación de la orientación en el *light field* 4D

Se ha realizado un análisis de error para la extensión del operador a cuatro dimensiones. Igual que sucedía en el caso del operador de dos dimensiones, existen errores sistemáticos debidos a la aproximación de los operadores derivativos y a la discretización. En la figura 4.5 se han representado los errores sistemáticos para distintas frecuencias espaciales y pendientes.

## 4.2 ANÁLISIS DE ERRORES



**Figura 4.4:** Desviación típica del error aleatorio en la estimación de la pendiente en imágenes bidimensionales en función de la pendiente a estimar.

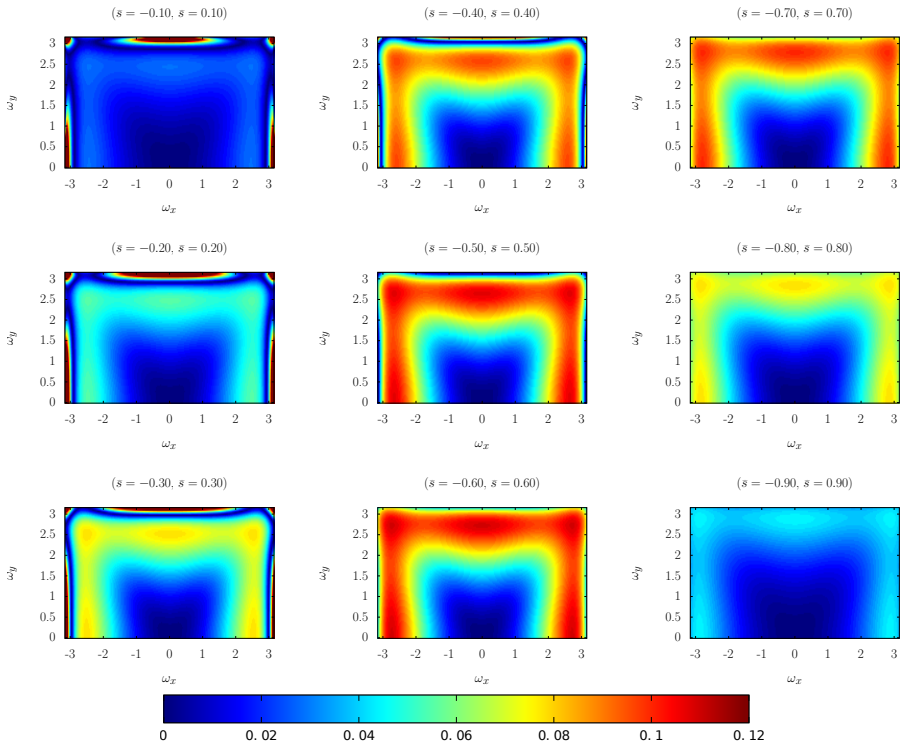
En el caso de los errores aleatorios, que afectan a la precisión del método, la expresión obtenida en el apéndice B, es la siguiente:

$$\sigma_{\hat{s}} = |\omega_{4D}| |p|^3 |d_1| (1 + \hat{s}^2) \frac{\sigma_L}{\zeta} \quad (4.20)$$

donde se redefine  $\zeta$  para cuatro dimensiones, quedando como:  $\zeta^2 = \widehat{L}_x^2 + \widehat{L}_y^2 + \widehat{L}_u^2 + \widehat{L}_v^2$ . Nótese además que esta expresión se ha obtenido asumiendo que todos los pesos de la operación  $\widehat{\cdot}$  son iguales y los valores promediados están decorrelados, con lo que  $|\omega_{4D}| = \frac{1}{\sqrt{N}}$  siendo  $N$  el número de puntos promediados.

## ESTIMACIÓN DE DISTANCIAS MEDIANTE UN DETECTOR DE PENDIENTES EN 4D

---

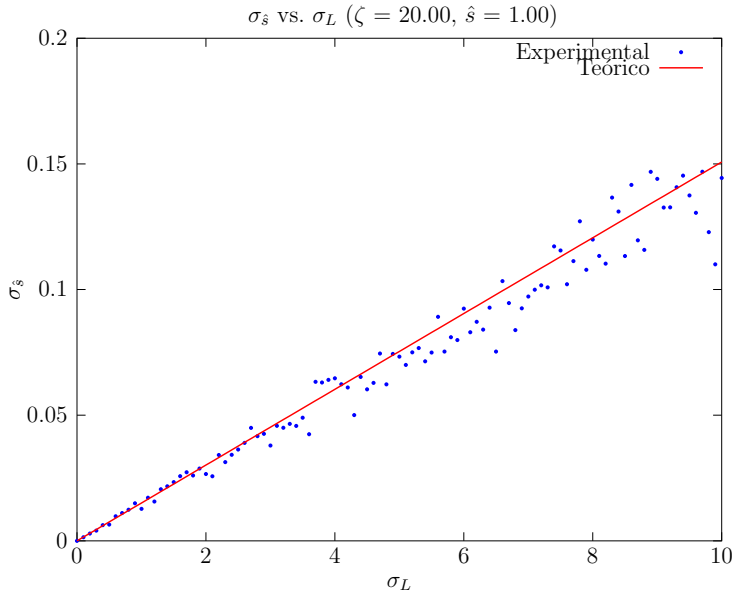


**Figura 4.5:** Errores sistemáticos para el operador local en cuatro dimensiones. El error para la componente de continua es infinito en todos los casos. Los errores para  $\hat{s} = 0$  y para  $\hat{s} = 1$  son nulos excepto en la componente de continua. Conviene recalcar que existe una simetría respecto de  $\hat{s} = 0$ .



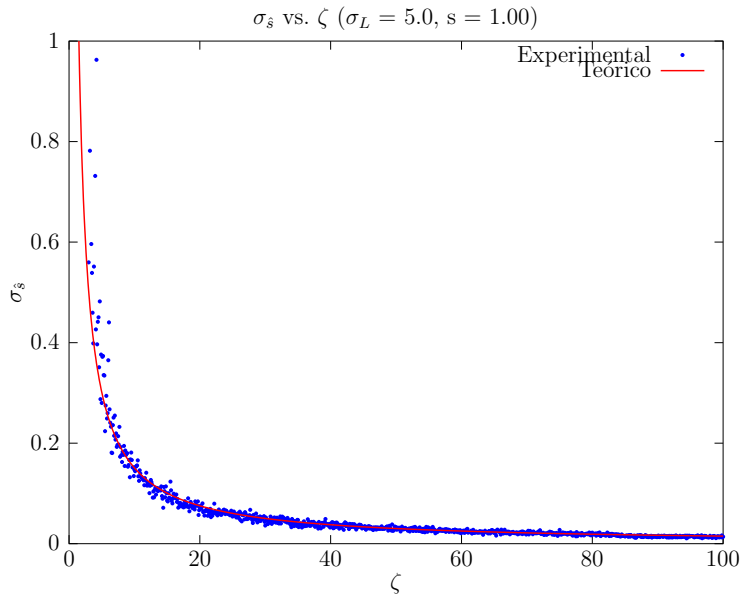
## 4.2 ANÁLISIS DE ERRORES

---



**Figura 4.6:** *Desviación típica del error aleatorio en la estimación de la pendiente frente a la desviación típica del ruido en la imagen para el operador en cuatro dimensiones.*

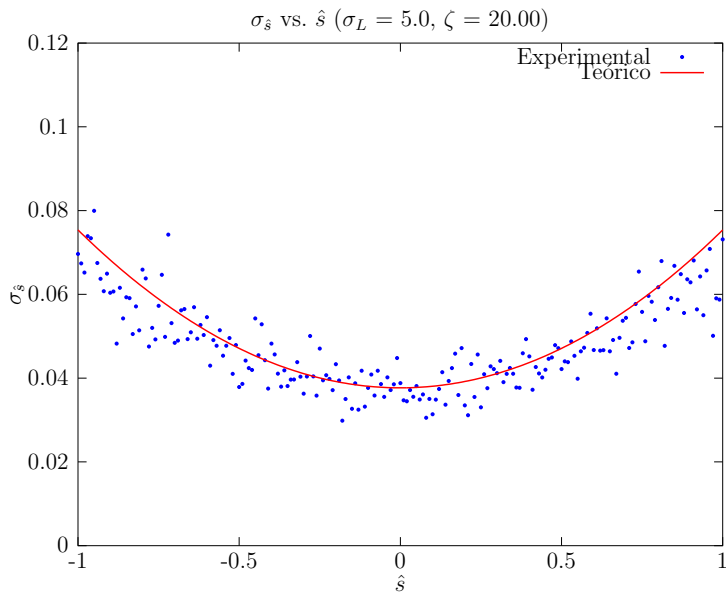
Esta expresión se ha validado nuevamente comparando los resultados obtenidos con resultados experimentales obtenidos al aplicar la expresión 4.16 sobre un conjunto de datos de entrada simulados. En las figuras 4.6, 4.7 y 4.8 se puede observar nuevamente como los resultados experimentales concuerdan con los resultados teóricos, para experimentos similares a los que se realizaron para el operador de dos dimensiones tomando  $\omega_x = 0,5$  y  $\omega_y = 0,5$ .



**Figura 4.7:** Desviación típica del error aleatorio en la estimación de la pendiente frente a  $\zeta$  para el operador en cuatro dimensiones.

## 4.2 ANÁLISIS DE ERRORES

---



**Figura 4.8:** *Desviación típica del error aleatorio en la estimación de la pendiente frente a la pendiente para el operador en cuatro dimensiones.*

### 4.2.3. Comparativa entre ambos operadores

Observando las figuras presentadas en las secciones anteriores se puede observar que el comportamiento de ambos operadores es similar pero el operador de cuatro dimensiones es más resistente al ruido.

En el caso de los errores aleatorios se puede comprobar que para el mismo valor  $\zeta$ , de  $\hat{s}$  y de desviación típica, la relación entre ambos errores es  $\frac{|\omega_{4D}||p|^2}{|\omega_{2D}|}$ . Suponiendo que el promedio pesado correspondiente al operador  $\hat{\cdot}$  se hace únicamente sobre los tres canales de color la relación se reduce a  $|p|^2 \approx 0,4$ . Esto significa que el error cometido con el operador de cuatro dimensiones es aproximadamente el 40 % del error cometido con el operador de dos dimensiones sobre la imagen epipolar.

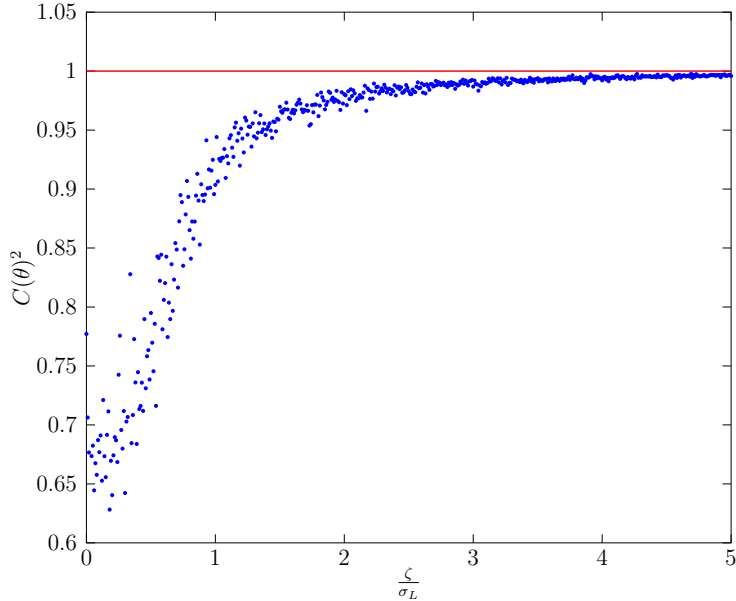
Este resultado debe interpretarse con cautela ya que no conviene olvidar que la definición de  $\zeta$  y de las derivadas es distinta en ambos casos, por lo que el mismo punto calculado con ambos métodos puede presentar valores de  $\zeta$  distintos. En cualquier caso, parece razonable que al incluirse un mayor número de puntos la robustez al ruido del operador extendido sea mayor.

### 4.3. Establecimiento de umbrales sobre la medida de confianza

Las expresiones para el cálculo del error aleatorio, presentadas en la sección anterior, indican que podría ser de interés limitar los puntos a un subconjunto en los que el valor de  $\zeta$  sobrepasa un determinado umbral. Por otra parte, también es conveniente seleccionar los puntos en función de la medida de confianza establecida en la ecuación 4.17.

El hecho de que  $\zeta$  aparezca tanto en la expresión de la confianza en la ecuación 4.17 como en la expresión del error en la ecuación 4.20, lleva a pensar que pudiera haber una función que relacione ambas magnitudes. Por ello, en la figura 4.9 se ha representado el cuadrado de la confianza  $C(\theta)^2$  frente a la relación  $\frac{\zeta}{\sigma_L}$ . Donde  $\zeta$ , mide el tamaño de la señal y  $\sigma_L$ , mide el tamaño del ruido y que se ha obtenido mediante datos simulados. Esta figura se puede interpretar como que los píxeles con mayor confianza generan un menor error en la estimación de  $\hat{s}$ , como era de esperar.

Se puede observar que existe una relación estrictamente creciente entre la confianza y  $\frac{\zeta}{\sigma_L}$ , lo que indica que es suficiente con umbralizar únicamente sobre la medida de confianza. En lo sucesivo este umbral de confianza se denotará con *cthr1*.



**Figura 4.9:** Cuadrado de la confianza  $C(\theta)^2$  frente al cociente  $\frac{z}{\sigma_L}$ .

## 4.4. Agregación de las medidas locales

El operador de estimación de pendientes, desarrollado en la sección 4.1, permite generar una estimación de distancias por cada uno de los puntos del *light field* de entrada, excepto en los bordes. Por ejemplo, para el *light field* de los cubos con dimensión  $292 \times 292 \times 13 \times 13$ , cuyo proceso de generación se ha descrito en la sección 3.3, potencialmente se puede obtener un conjunto de  $290 \times 290 \times 11 \times 11$  medidas de distancia. Esto constituye un conjunto de más de 10 millones de medidas, cada una correspondiente a un punto del espacio tridimensional.

#### 4.4 AGREGACIÓN DE LAS MEDIDAS LOCALES

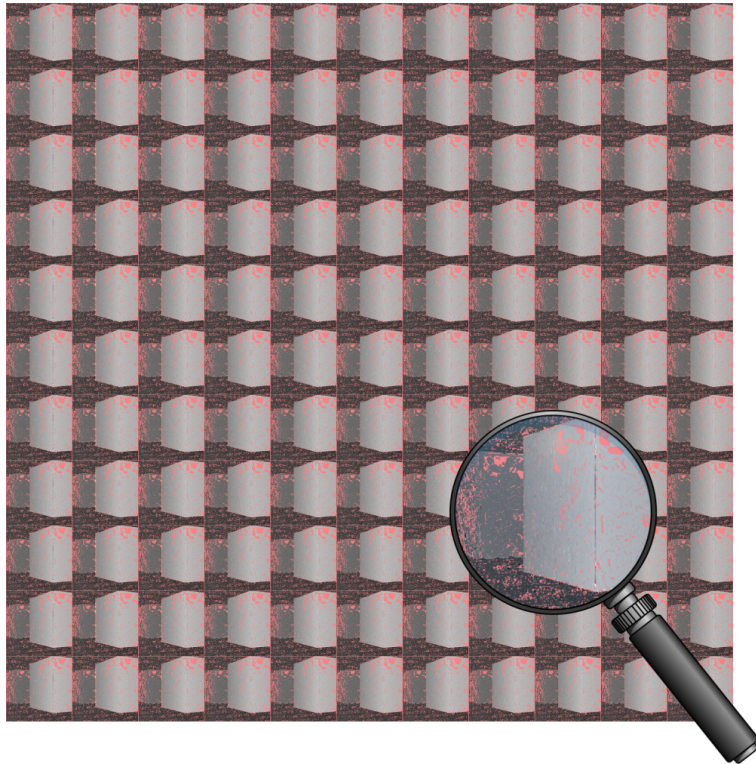
---

El conjunto de medidas 3D obtenido puede representarse mediante un conjunto de mapas de distancia no completos, ya que algunos de los puntos no tienen valores válidos. En la figura 4.10 se presenta el conjunto de  $11 \times 11$  mapas de distancia incompletos generado para la escena de los cubos.

La ausencia de medidas en algunas sitios se debe a la aplicación del umbral *cthr1* con el que se descartan las medidas poco fiables que generalmente aparecen en zonas con poca textura. Estos puntos poco fiables se muestran con color rojo en la figura 4.10. El inconveniente de esta representación es que se dispone de múltiples mapas de distancias para la escena que deben ser combinados.

La forma más sencilla de representar de forma unificada todos estos resultados es mediante una nube de puntos. Para obtener esta representación se proyecta cada punto al espacio 3D y se ponen todos los puntos en un marco de referencia común. La ventaja es que la nube de puntos es sencilla de obtener y además permite asociar a cada punto un vector de color que se obtiene directamente del *light field*. En la figura 4.11, se muestra una nube de puntos formada por 4.8 millones de puntos para la escena de los cubos. A la izquierda se muestran los puntos coloreados con una escala de color que representa la distancia y a la derecha se muestra la misma nube coloreando los puntos con el valor de color obtenido del *light field*.

Otro método para agregar las medidas disponibles en los distintos mapas de distancias es re proyectar cada uno de los puntos hacia el mundo y proyectarlo nuevamente sobre el sensor de una cámara

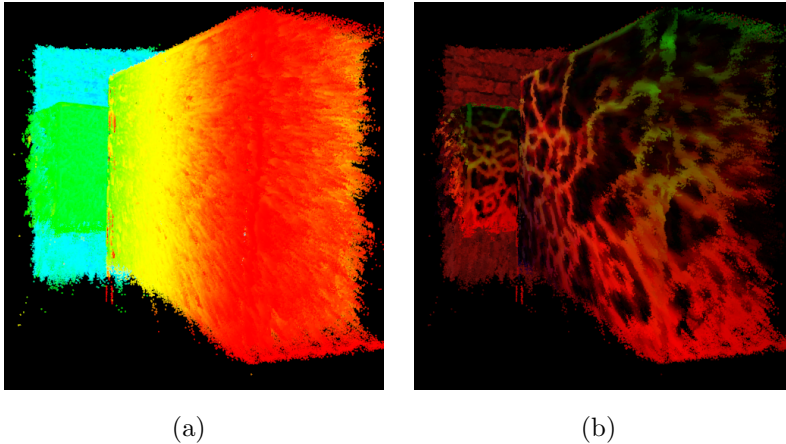


**Figura 4.10:** *Múltiples mapas de distancias obtenidos estimando la orientación en 4D. Los puntos con una confianza inferior a 0,9 se han representado con el color rojo.*



#### 4.4 AGREGACIÓN DE LAS MEDIDAS LOCALES

---



**Figura 4.11:** Representación de las medidas mediante nubes de puntos : (a) Nube de puntos coloreada con una escala de color que refleja la distancia. Los colores cálidos representan puntos cercanos y los colores fríos representan puntos lejanos; (b) Nube de puntos coloreada con los valores de color extraídos del light field.

virtual. La posición de esta cámara suele hacerse coincidir con la de uno de los mapas de distancias medidos que se toma como referencia. Al hacer coincidir la posición de la cámara virtual con uno de los puntos de vista, no es necesario pasar por un cambio de coordenadas al mundo. Esta operación se puede hacer directamente sobre las coordenadas imagen. Además, es posible fijar una resolución del sensor virtual distinta a la de cada uno de los mapas. Finalmente, tras proyectar las medidas sobre el sensor virtual puede haber varias medidas por cada uno de los píxeles, pudiéndose optar por diferentes métodos para obtener una única medida combinada. Antes de la

proyección, es conveniente realizar una selección de los puntos en base a criterios de confianza como los que se han indicado en la sección 4.3. Una vez combinadas las medidas de cada punto, se puede volver a umbralizar la confianza utilizando un nuevo umbral que se denotará con *cthr2*.

Entre los posibles métodos para combinar las distintas medidas que proyectan sobre cada punto se han considerado las siguientes:

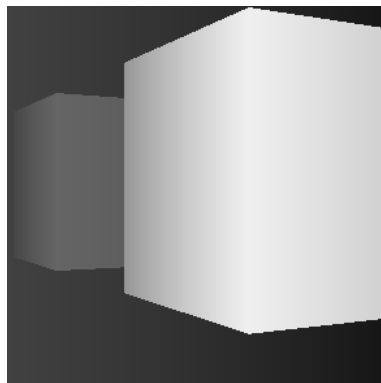
- Media ponderada: Se realiza una media ponderada de los distintos valores de distancia utilizando como pesos la confianza de cada uno de ellos. En este caso, para la medida de la confianza final, se ha utilizado la media de las confianzas, por lo que *cthr2* debe fijarse a un valor mayor que en los demás casos. El resultado de este procedimiento se muestra en la figura 4.13(a).
- Distancia más cercana: Se utiliza como medida final la distancia más cercana y, como medida de confianza final, la confianza asociada a ese punto. Este método se sustenta en la idea de que entre todos los puntos que proyectan sobre un píxel del sensor virtual, sólo el más cercano será visible. Sin embargo, esta técnica introduce un sesgo en el caso de que las medidas tengan ruido, ya que tiende a acercar todos los puntos del mapa final. Este hecho se puede observar en la figura 4.13(b) comparándola con el resto de mapas.
- Distancia con mayor confianza: La medida final del píxel se obtiene seleccionando el punto proyectado de mayor confianza,

#### 4.4 AGREGACIÓN DE LAS MEDIDAS LOCALES

---

siendo esta misma confianza la medida de confianza asociada. El resultado se puede observar en la figura 4.13(c).

- Promedio de los tensores de estructura: En este caso no se promedian directamente las medidas de distancia sino que se promedian los numeradores y los denominadores de la expresión 4.16 por separado y luego se calcula el cociente. Esto es equivalente a promediar los tensores de estructura asociados a cada uno de los puntos en origen. La medida de confianza se calcula utilizando la ecuación 4.17.



**Figura 4.12:** Mapa de distancias verdadero generado con el simulador .

Los resultados obtenidos con cada uno de los métodos se pueden comparar con el mapa de distancias verdadero para la escena de los cubos con fondo, que se muestra en la figura 4.12. En los bordes de los cubos hay dos distancias claramente diferenciadas. Esto induce a errores en los cálculos de las derivadas del *light field* que se propagan

a las estimaciones de pendiente que se realizan posteriormente. Para reducir este efecto se ha optado por utilizar un entorno de  $3 \times 3 \times 3 \times 3$  para el cálculo de la derivada y que el operador  $\hat{\cdot}$  de la ecuación 4.16 únicamente promedie los tres canales de color y sin realizar promedios espaciales. El promedio de los tensores de estructura es el que mejor se comporta en estas situaciones ya que se obtienen mejores valores de confianza, como puede observarse en la figura 4.13(d). Hay que tener en cuenta que este efecto puede aumentar cuando aumenta el ruido de la imagen plenóptica ya que a su vez aumentará el ruido en la estimación de  $\hat{s}$  haciendo que algunos puntos se proyecten sobre píxeles que no se corresponden con su distancia real. El estudio detallado de este efecto se abordará en un futuro trabajo.

## 4.5. Detalles de implementación

En esta sección se aclararan algunos detalles de la implementación sobre GPU del método local de estimación de distancias descrito en este capítulo.

La principal dificultad a la hora de implementar un algoritmo de procesamiento del *light field* sobre GPU, es la localidad de memoria. Hay que tener en cuenta que el diseño hardware de la GPU está optimizado para el procesamiento de imágenes bidimensionales y no para funciones de cuatro dimensiones. Por ello, hay que mapear la función de cuatro dimensiones sobre una imagen de dos dimensiones, mientras que los patrones de acceso a memoria requeridos siguen

siendo en 4D, dando lugar a accesos no coalescentes sobre la imagen 2D. Por eso, los accesos a memoria son el principal problema para la optimización del código.

Lo primero que debe hacerse para poder obtener una estimación de  $\hat{s}$  es el cálculo de las derivadas del *light field* aplicando los núcleos derivativos de la ecuación 2.5. Al ser el filtro derivativo separable, se aplica el filtro  $d_1$  en la dirección de la derivada y el filtro  $p$  en las tres direcciones restantes. A priori, para la obtención de las 4 derivadas deberán hacerse 16 ejecuciones del *kernel conv<sub>i</sub>(f)* que aplica el filtro  $f$  en la dirección  $i$ . Sin embargo, se ha optado por aprovechar parte de los datos calculados previamente reduciendo el número de ejecuciones del *kernel* a 13, lo que supone un 18.75 % menos de tiempo de ejecución a costa de aumentar los requerimientos de memoria. En la figura 4.14 se muestra el flujo de datos entre los distintos *kernels* hasta llegar al *kernel compute\_av\_proj*, que calcula las pendientes y agrega todas las medidas en un único mapa de distancias.

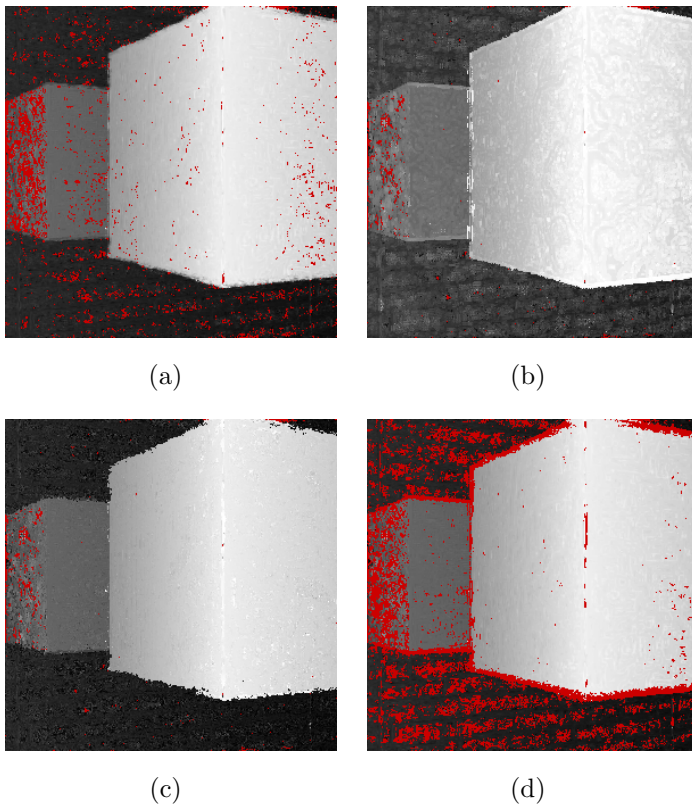
El *kernel* de agregación *compute\_av\_proj*, se ha implementado promediando los tensores de estructura de los rayos que proyectan sobre cada punto del mapa de distancias final. Este *kernel* tiene dos partes. La primera, calcula la distancia a partir de  $L_x$ ,  $L_y$ ,  $L_u$  y  $L_v$  aplicando la ecuación 4.16. La segunda, consiste en, una vez obtenido el valor de la pendiente y de la medida de confianza, seleccionar los puntos que superan el umbral de confianza y proyectarlos sobre el sensor virtual. Este paso puede generar un cuello de botella. Varios puntos pueden proyectar en paralelo sobre el mismo píxel

final, por lo que debe protegerse mediante mecanismos de exclusión mutua. Finalmente, se ejecuta el *kernel compute\_depth*, que calcula la pendiente y la confianza final aplicando la ecuación 4.16 sobre los tensores promediados.

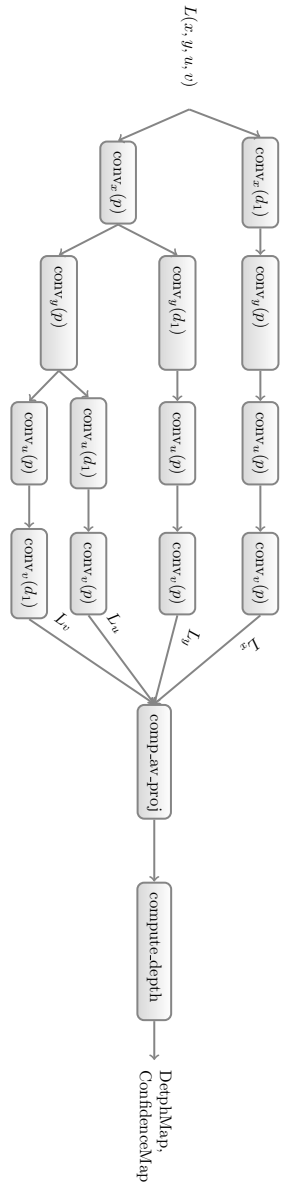
La implementación final se ha realizado sobre un PC con un procesador Intel ® Core™i5-2320 a 3.00GHz dotado de 8 Gb de memoria RAM y sistema operativo Debian GNU/Linux 6.0. La tarjeta gráfica empleada ha sido la NVidia Tesla C2075 y para su programación se ha empleado el NVidia CUDA Toolkit 5.0.

En la figura 4.15, se muestran las medidas de tiempo de ejecución sobre el sistema indicado en función del número de puntos de salida a generar. Considerando una tasa de vídeo de 25 fps, lo que equivale a 40 ms por cuadro, se podrían alcanzar estas tasas generando hasta aproximadamente 3.5 millones de puntos.

Durante el proceso de medida no se han tenido en cuenta los procesos de carga y descarga de datos. Considerando ambos tiempos, el número de puntos de salida a obtener en tiempo real serían menores. Suponiendo por ejemplo, un sensor como el utilizado en Ng et al. (2005), el número de puntos de salida ronda los 10 millones de puntos, siendo el tamaño de las imágenes de salida de  $292 \times 292$  píxeles, lo cual incita a buscar técnicas e implementaciones aún más eficientes o al uso de múltiples GPUs como se hizo en Lüke et al. (2010). Como se ha dicho anteriormente, el principal obstáculo para mejorar los tiempos de ejecución son los patrones de acceso a memoria que se producen al mapear el *light field* 4D sobre un array 2D.



**Figura 4.13:** Mapas de distancia obtenidos mediante distintos métodos de agregación : (a) Mapa de distancia agregado con la media ponderada con la confianza ( $cthr2 = 0,97$ ); (b) Mapa de distancias agregado utilizando el máximo en cada punto ( $cthr2 = 0,95$ ); (c) Mapa de distancias agregado utilizando la medida de confianza máxima en cada punto ( $cthr2 = 0,95$ ); (d) Mapa de distancias agregado mediante el promedio de los tensores de estructura ( $cthr2 = 0,95$ ).

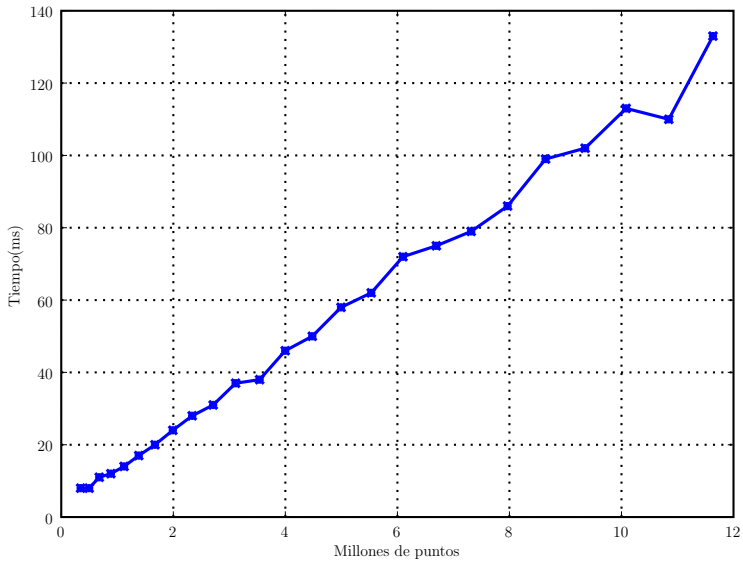


**Figura 4.14:** Diagrama de flujo de datos entre los kernels ejecutados en la implementación en *CUDA*.



## 4.5 DETALLES DE IMPLEMENTACIÓN

---



**Figura 4.15:** *Tiempos de ejecución sobre GPU para la obtención de un mapa de distancias disperso a partir del light field.*

ESTIMACIÓN DE DISTANCIAS MEDIANTE UN DETECTOR DE  
PENDIENTES EN 4D

---

# Capítulo 5

## Estimación de mapas de distancias densos

*“Si buscas resultados distintos, no hagas siempre lo mismo.”*

---

Albert Einstein

Las medidas agregadas obtenidas mediante los procedimientos descritos en la sección 4.4, generan mapas de distancias incompletos. Sin embargo, es deseable obtener mapas de distancia densos, con una medida por cada píxel. Como se ha indicado en la sección 2.6, para ello habrá que aplicar un proceso de regularización.

Debido a su idoneidad para optimizar modelos regularizados con un término con norma L1, como la variación total, se utilizará el método

split Bregman para estimar mapas de distancias densos con el modelo TV-L1. El principal problema de este modelo es que la norma L1 no es derivable. Sin embargo, el método split Bregman permite separar el problema en distintos subproblemas de modo que los términos que contienen normas L1 se pueden resolver mediante una expresión en forma cerrada (Goldstein y Osher, 2009).

El problema de minimizar el modelo TV-L1 queda definido como se ha descrito en la sección 2.6:

$$\arg \min_u \int_{\Omega} \lambda |u - \hat{s}| d\mathbf{x} + TV(u) \quad (5.1)$$

$TV(u) = \int_{\Omega} |\nabla u(x)| d\mathbf{x}$  es la variación total del mapa de pendientes estimado  $u$ .  $\lambda$  se define del siguiente modo a partir de la medida de confianza definida en la ecuación 4.17:

$$\lambda(\mathbf{x}) = \begin{cases} \lambda_f C(\mathbf{x}) & \text{si } C(\mathbf{x}) > cthr2 \\ 0 & \text{en otro caso} \end{cases} \quad (5.2)$$

donde  $\lambda_f$  es un parámetro que permite controlar el peso global del término de fidelidad frente al de regularización.

## 5.1. Método Split Bregman para el modelo TV-L1

En este apartado se aplican los resultados obtenidos en Goldstein y Osher (2009) y se desarrollan los elementos necesarios para obtener

## 5.1 MÉTODO SPLIT BREGMAN PARA EL MODELO TV-L1

---

una solución al problema planteado en la ecuación 5.1 mediante el método split Bregman. En Getreuer (2012) se puede encontrar un desarrollo similar aplicado al problema del relleno y restauración de imágenes (*inpainting*).

El primer paso es replantear el problema de 5.1 como un problema con restricciones, con el siguiente planteamiento:

$$\begin{aligned} & \arg \min_{u,d,z} \int_{\Omega} |d| d\mathbf{x} + \int_{\Omega} \lambda |z - \widehat{s}| d\mathbf{x} \\ & \text{sujeto a dos restricciones } (d = \nabla u) \text{ y } (z = u) \end{aligned} \quad (5.3)$$

A continuación, para convertir el problema con restricciones en un problema sin restricciones se introducen dos términos de penalización, quedando de la siguiente manera:

$$\arg \min_{u,d,z} \int_{\Omega} \left( |d| + \lambda |z - \widehat{s}| + \frac{\gamma_1}{2} \|d - \nabla u\|_2^2 + \frac{\gamma_2}{2} (z - u)^2 \right) d\mathbf{x} \quad (5.4)$$

Luego se añaden las variables de Bregman  $b_1$  y  $b_2$  obteniendo la siguiente iteración split Bregman simplificada (Goldstein y Osher, 2009, ecuaciones 2.9 y 2.10):

$$\begin{aligned} (u^{k+1}, d^{k+1}, z^{k+1}) = \arg \min_{u,d,z} & \left\{ \int_{\Omega} \left( |d| + \lambda |z - \widehat{s}| + \dots \right. \right. \\ & \left. \left. \dots + \frac{\gamma_1}{2} \|d - \nabla u - b_1^k\|_2^2 + \frac{\gamma_2}{2} (z - u - b_2^k)^2 \right) d\mathbf{x} \right\} \end{aligned} \quad (5.5)$$

$$b_1^{k+1} = b_1^k + \nabla u^{k+1} - d^{k+1} \quad (5.6)$$

$$b_2^{k+1} = b_2^k + u^{k+1} - z^{k+1} \quad (5.7)$$

La minimización conjunta sobre las variables  $u$ ,  $d$  y  $z$  de la ecuación 5.5 se ha resuelto aplicando el método de las direcciones alternativas, de modo que en cada paso se minimiza una de las variables manteniendo el resto fijas. Esto hace que se presenten tres subproblemas que deben ser discretizados y a continuación se presenta la solución que se ha aplicado a cada uno de ellos:

- Minimización de  $d$  con  $u$  y  $z$  fijas:

$$\arg \min_d \sum_{i,j} |d_{i,j}| + \frac{\gamma_1}{2} \|d_{i,j} - \nabla u_{i,j} - b_{1_{i,j}}\|_2^2 \quad (5.8)$$

Los elementos de  $d$  son espacialmente independientes por lo que la minimización global consiste en minimizar la expresión anterior para cada posición  $(i, j)$ . Además, este problema admite una solución en forma cerrada aplicando el operador de *soft-thresholding* (Donoho y Johnstone, 1995; Wang et al. 2007):

$$d_{i,j} = \max \left\{ \|\nabla u_{i,j} + b_{1_{i,j}}\| - \frac{1}{\gamma_1}, 0 \right\} \frac{\nabla u_{i,j} + b_{1_{i,j}}}{\|\nabla u_{i,j} + b_{1_{i,j}}\|} \quad (5.9)$$

- Minimización de  $z$  con  $d$  y  $u$  fijas:

$$\arg \min_z \sum_{i,j} \lambda_{i,j} |z_{i,j} - \hat{s}_{i,j}| + \frac{\gamma_2}{2} (z_{i,j} - u_{i,j} - b_{2_{i,j}})^2 \quad (5.10)$$

Este problema es independiente para cada punto  $(i, j)$  y también admite una solución en forma cerrada realizando previamente el cambio de variable  $w = z - \hat{s}$ . Entonces la ecuación 5.10 se

reescribe como:

$$\arg \min_w \sum_{i,j} \lambda_{i,j} |w_{i,j}| + \frac{\gamma_2}{2} (w_{i,j} + \widehat{s}_{i,j} - u_{i,j} - b_{2_{i,j}})^2 \quad (5.11)$$

y finalmente el valor óptimo de  $z$  es:

$$\begin{aligned} z_{i,j} &= \widehat{s}_{i,j} + w_{i,j} = \\ &= \widehat{s}_{i,j} + \text{signo}(r) \max \left\{ r - \frac{\lambda_{i,j}}{\gamma_2}, 0 \right\}, \\ r &= u_{i,j} + b_{2_{i,j}} - \widehat{s}_{i,j} \end{aligned} \quad (5.12)$$

- Minimización de  $u$  con  $d$  y  $z$  fijas:

$$\arg \min_u \sum_{i,j} \frac{\gamma_1}{2} \|d_{i,j} - \nabla u_{i,j} - b_{1_{i,j}}\|_2^2 + \frac{\gamma_2}{2} (z_{i,j} - u_{i,j} - b_{2_{i,j}})^2 \quad (5.13)$$

La solución a este problema satisface:

$$\gamma_2 u - \gamma_1 \Delta u = \gamma_2 (z - b_2) - \gamma_1 \text{div}(d - b_1) \quad (5.14)$$

La ecuación 5.14 es una variante de la ecuación de Poisson cuya solución se puede obtener en el dominio de Fourier, en el dominio de la transformada discreta coseno, o mediante métodos iterativos. En Goldstein y Osher (2009) se propone aplicar una pasada del método de Gauss-Seidel en cada iteración de Bregman. También se opta por esta simplificación en este trabajo.

El nuevo valor de  $u$  se obtiene mediante la siguiente expresión:

$$\begin{aligned}
 G_{i,j} = & \frac{1}{\gamma_2 + 4\gamma_1} (\gamma_1 (u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}) + \dots \\
 & \dots + \gamma_2 (z_{i,j} - b_{2i,j}) + \gamma_1 (d_{i,j} - d_{i-1,j} + d_{i,j} - d_{i,j-1} + \dots \\
 & \dots + b_{1i,j} - b_{1i-1,j} + b_{1i,j} - b_{1i,j-1}))
 \end{aligned}
 \tag{5.15}$$

Una vez resueltos los tres subproblemas se obtiene el algoritmo split Bregman para la minimización del modelo TV-L1 que se aplicará a los mapas de distancia incompletos que se obtienen mediante el método descrito en el capítulo 4. El pseudocódigo se muestra en el algoritmo 1.

## 5.2. Implementación jerárquica de grueso a fino

En el algoritmo 1, el número de iteraciones máximo a aplicar debe aumentar con el número de píxeles considerados en el mapa de distancias. Para reducir este efecto, cuando se regularizan mapas de distancia de imágenes estáticas, se opta por una versión jerárquica del algoritmo.

Esta aproximación consiste en definir una jerarquía de  $L$  niveles sobre los datos de entrada. El nivel 0 son los datos iniciales a regularizar, obtenidos con el método propuesto en el capítulo 4. El nivel  $l$  consiste en realizar una reducción de resolución de  $2^l \times 2^l$  a los



---

**Algoritmo 1:** Iteración split Bregman para la minimización del modelo TV-L1.

---

**Data:** Mapa de distancias incompleto,  $\widehat{s}$

**Result:** Mapa de distancias denso estimado,  $u$

```

1  $k = 0$ 
2  $u = \widehat{s}$ 
3  $b_1 = 0, b_2 = 0, z = 0, d = 0$ 
4 while  $k < max\_iter$  do
5     foreach  $(i, j)$  do
6          $u_{i,j} = G_{i,j}$ 
7          $d_{i,j} = \max \left\{ \|\nabla u_{i,j} + b_{1_{i,j}}\| - \frac{1}{\gamma_1}, 0 \right\} \frac{\nabla u_{i,j} + b_{1_{i,j}}}{\|\nabla u_{i,j} + b_{1_{i,j}}\|}$ 
8          $r = u_{i,j} + b_{2_{i,j}} - \widehat{s}_{i,j}$ 
9          $z_{i,j} = \widehat{s}_{i,j} + \text{signo}(r) \max \left\{ r - \frac{\lambda_{i,j}}{\gamma_2}, 0 \right\}$ 
10         $b_{1_{i,j}} = b_{1_{i,j}} + \nabla u_{i,j} - d_{i,j}$ 
11         $b_{2_{i,j}} = b_{2_{i,j}} + u_{i,j} - z_{i,j}$ 
12    end
13     $k = k + 1$ 
14 end

```

---

datos iniciales. Para una jerarquía de  $L$  niveles, se comienza ejecutando el algoritmo 1 sobre los datos del nivel  $L - 1$  con un cierto número de iteraciones. A continuación, se propagan los resultados obtenidos en un nivel  $l$  al nivel  $l - 1$  utilizando el resultado interpolado de  $l$  como solución inicial de  $l - 1$  y se ejecuta nuevamente el algoritmo 1. Este procedimiento se repite hasta alcanzar el nivel 0, donde se obtiene la solución final al problema con la misma resolución que los datos iniciales.

En la figura 5.1, se muestran los datos para una ejecución jerárquica sobre la escena de los cubos, mientras que la figura 5.2 se muestra el resultado de aplicar directamente el algoritmo 1 a los mismos datos de entrada. Se puede observar como al comparar cualitativamente ambos resultados el resultado del proceso jerárquico es mejor. Esto se debe a que el relleno de los huecos sin datos se hace más rápido en los niveles de menor escala y en los niveles de mayor escala sólo se refina el resultado, por lo que se necesita un menor número de iteraciones. Otra ventaja de este método es que se reduce el tiempo de ejecución. Esta reducción depende del número de iteraciones a realizar en cada uno de los niveles y del propio número de niveles. Suponiendo el mismo número de iteraciones en cada nivel, la aceleración viene dada por la siguiente expresión:

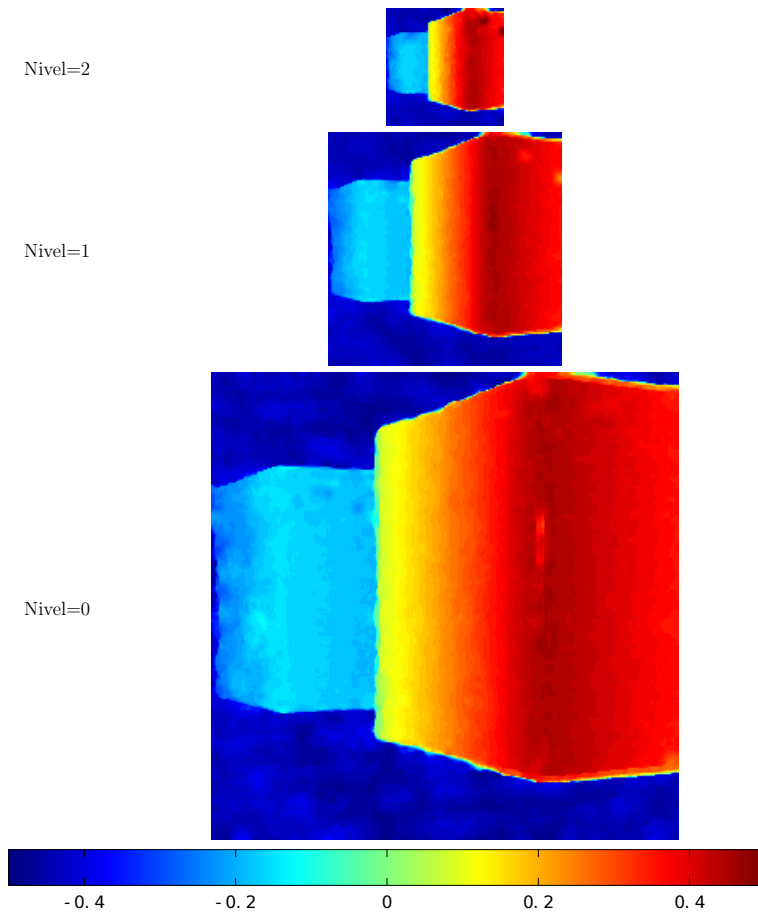
$$\text{Speed Up} = \frac{L \cdot \left(\frac{1}{4} - 1\right)}{2^{-2L} - 1} \quad (5.16)$$

Por ejemplo, para los tres niveles de la figura 5.1 se obtendría una aceleración de 2,28, aunque hay que tener en cuenta que en la práctica

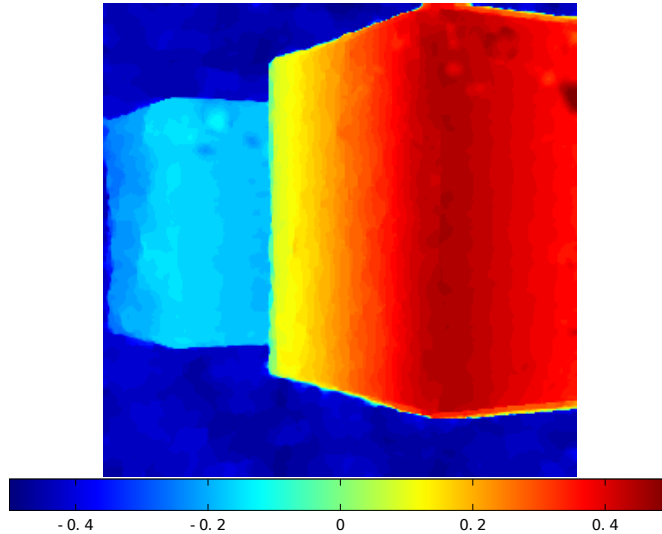
## 5.2 IMPLEMENTACIÓN JERÁRQUICA DE GRUESO A FINO

---

este valor será algo menor debido a que esta expresión no considera los tiempos correspondientes a los remuestreos necesarios para obtener los datos de entrada a cada nivel.



**Figura 5.1:** Resultados obtenidos en los distintos niveles de la jerarquía para la escena de los cubos con 12 iteraciones en cada nivel.



**Figura 5.2:** Resultado obtenido para la escena de los cubos realizando 36 iteraciones sin jerarquizar.

### 5.3. Aplicación a secuencias temporales

El método de estimación presentado en la sección 5.1 parte de una aproximación inicial de la solución que se ha hecho coincidir con las pendientes medidas  $\hat{s}$ . La calidad de dicha aproximación inicial condiciona la velocidad de convergencia y el número de iteraciones necesario para obtener un resultado satisfactorio. Una forma de acelerar la convergencia, cuando el método se aplica a secuencias temporales de imágenes, podría ser el uso de los resultados obtenidos en la imagen anterior como aproximación inicial de la imagen actual. Esto permitiría reducir el número de iteraciones necesario y por tanto

### 5.3 APLICACIÓN A SECUENCIAS TEMPORALES

---

obtener tasas de procesamiento mayores que las que se obtendrían al procesar cada una de las imágenes de manera independiente.

Si la posición y la orientación de la cámara son conocidos en dos instantes de tiempo sucesivos  $t_1$  y  $t_2$ , el resultado obtenido en  $t_1$  se puede transformar al marco de coordenadas de la imagen obtenida en  $t_2$  por medio de una operación lineal, lo que permite utilizar estos datos como solución inicial, aunque este enfoque requiere realizar un seguimiento de la posición y la orientación de la cámara.

Conocidos los parámetros extrínsecos de la cámara, es decir la posición y la rotación, los datos obtenidos en el instante  $t_1$  se transforman al marco de coordenadas de la cámara en el instante  $t_2$  por medio de la siguiente operación (Hartley y Zisserman, 2004):

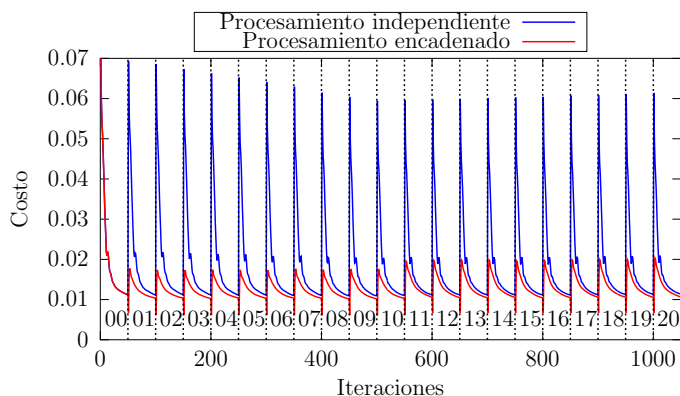
$$\mathbf{X}_2 = \left[ \begin{array}{c|c} R & T \\ \hline 0 & 1 \end{array} \right] \mathbf{X}_1 = \mathbf{H}\mathbf{X}_1 \quad (5.17)$$

donde  $R$  es la matriz de rotación relativa de la cámara,  $T$  es el desplazamiento relativo de la cámara,  $X_1$  es el punto, expresado en coordenadas homogéneas del mundo, respecto del marco de referencia de la cámara en el instante  $t_1$  y  $X_2$  es el punto en coordenadas homogéneas respecto de la cámara en  $t_2$ . Conviene recalcar que, al cambiar el mapa de distancias de  $t_1$  al marco de referencia de la cámara en  $t_2$ , no sólo cambian las posiciones de los píxeles, como ocurre en imágenes convencionales, sino también su intensidad o lo que es lo mismo el valor de distancia asignado a los mismos.

En el caso de que los movimientos relativos de traslación y de

rotación de la cámara en el instante  $t_2$  respecto al instante  $t_1$  sean infinitesimales, la matriz de cambio de coordenadas  $H$  se puede aproximar como la matriz identidad con lo que la aproximación inicial del *frame* actual es justamente el resultado obtenido en el *frame* anterior y se evitaría tener que realizar un seguimiento de los parámetros extrínsecos de cámara.

A continuación, se compararan el procesamiento independiente de cada *frame* y el procesamiento encadenado de los *frames* de una secuencia de 21 imágenes de la escena de los cubos.



**Figura 5.3:** Función de costo en cada frame en función del número de iteraciones para el procesamiento independiente y el procesamiento encadenado.

En la figura 5.3 se muestra la función costo definida en la ecuación 2.13 frente al número de iteraciones del algoritmo split Bregman. Cada una de las ranuras se ha numerado con el número de secuencia de la imagen de entrada y abarca 50 iteraciones. En la ranura 0,

### 5.3 APLICACIÓN A SECUENCIAS TEMPORALES

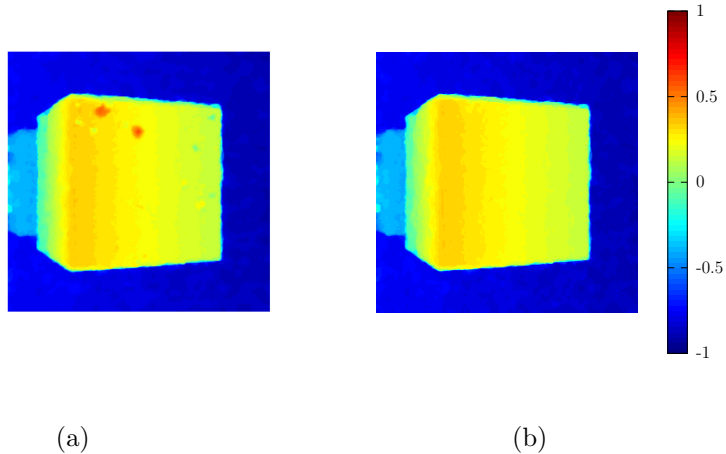
---

se puede observar como ambos tipos de procesamiento coinciden ya que en la primera imagen de la secuencia no se dispone de una aproximación inicial mejor que los propios datos medidos. Las ranuras del 1 al 10 corresponden a un desplazamiento de la cámara a velocidad constante a través de la escena de los cubos. Las ranuras del 11 al 20 corresponden a una rotación horizontal de la cámara.

En la figura se observa como la función de coste siempre es menor para el procesamiento encadenado. Sin embargo, esto sólo se mantiene cuando los movimientos de cámara son infinitesimales. En otro caso, puede ocurrir que al partir de la solución del *frame* anterior el algoritmo se inicie en una situación más lejana a la solución que si se hubiera inicializado con los propios datos de entrada. La cuestión que surge de inmediato es cuan pequeños deben ser los movimientos de cámara para que se puedan considerar infinitesimales. Este aspecto depende de la configuración óptica empleada así como de la propia escena y requiere un estudio más profundo que se abordará en trabajos futuros.

Por otra parte, en la figura 5.4 se muestran los mapas de pendientes obtenidos para la imagen 20 de la secuencia, utilizando procesamiento independiente y procesamiento encadenado con 50 iteraciones en ambos casos. Se observa como el mapa obtenido con procesamiento independiente presenta algunas motas, debidas a que no se han ejecutado suficientes iteraciones de split Bregman. El mapa obtenido con procesamiento encadenado no presenta motas. Estos errores van desapareciendo a medida que se va avanzando en la secuencia, lo cual

es una ventaja respecto del procedimiento independiente.



**Figura 5.4:** Mapas de pendientes regularizados obtenidos para el frame 20 de la secuencia de los cubos. (a) Mapa obtenido con procesamiento independiente. (b) Mapa obtenido con procesamiento encadenado.

## 5.4. Detalles de implementación

La implementación del algoritmo split Bregman se ha realizado sobre el hardware que ya se especificó en la sección 4.5 y se ha dividido en dos *kernels*. El primer *kernel* se encarga de ejecutar la iteración Gauss-Seidel definida en la ecuación 5.15, mientras que el segundo se encarga de realizar el resto de los pasos internos del bucle del algoritmo 1.



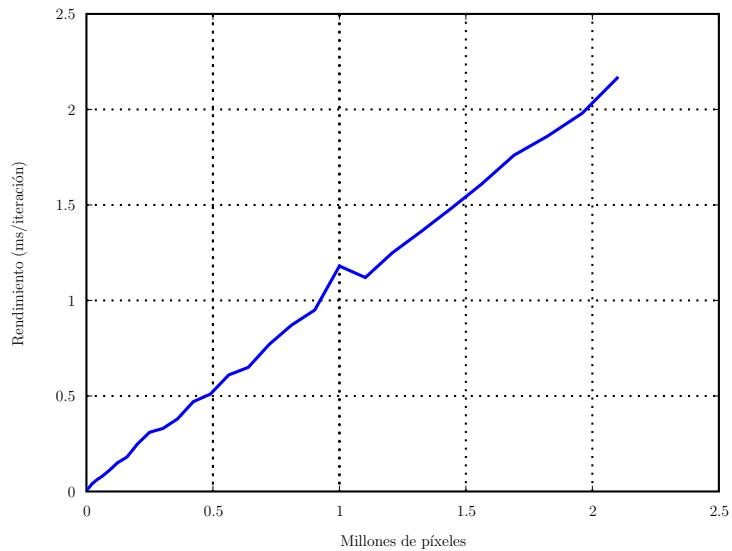
## 5.4 DETALLES DE IMPLEMENTACIÓN

---

La razón de dividir el proceso en dos *kernels* independientes es que el gradiente de los nuevos valores de  $u$ , que se calculan en la línea 6, se utiliza en las líneas 7 y 10. Al hacer la división, se produce una sincronización antes de comenzar con el segundo *kernel* que abarca desde las líneas 7 hasta 11, asegurando que todos los valores de  $u$  se hayan calculado antes de calcular el gradiente.

Llamando a ambos *kernels*, se paraleliza el bucle *for* interno, de manera que cada píxel potencialmente se pueda calcular en paralelo con los demás. El bucle externo, no es paralelizable debido a que los datos que se calculan en una iteración dependen de los resultados de la iteración anterior.

En la figura 5.5 se muestran los tiempos de ejecución por iteración en función del tamaño del mapa de salida en millones de píxeles. Se puede observar un crecimiento lineal con una pendiente aproximada de  $1 \text{ ms}/(\text{iteración} * \text{Mpixel})$ .



**Figura 5.5:** *Tiempos de ejecución por iteración de split Bregman en función del tamaño del mapa de salida en millones de píxeles.*

# Capítulo 6

## Resultados

*“Lo mejor es enemigo de lo bueno.”*

---

Sigmund Freud

En este capítulo se presentan los resultados obtenidos al aplicar el método desarrollado en los capítulos anteriores sobre la serie de *light fields*, tanto sintéticos como reales, presentados en el capítulo 3. El uso de datos sintéticos permitirá contrastar los datos de distancia obtenidos con los datos de la escena que en ese caso son conocidos, mientras, que la aplicación del método a datos obtenidos con cámaras reales permite determinar de forma cualitativa la calidad de los resultados obtenidos.

| LF      | $cthr1$ | $cthr2$ | iteraciones | $\lambda_f$ |
|---------|---------|---------|-------------|-------------|
| ovillos | 0.9     | 0.9     | 1000        | 0.2         |
| flores3 | 0.95    | 0.9     | 5000        | 0.3         |
| tortuga | 0.7     | 0.7     | 2000        | 0.3         |

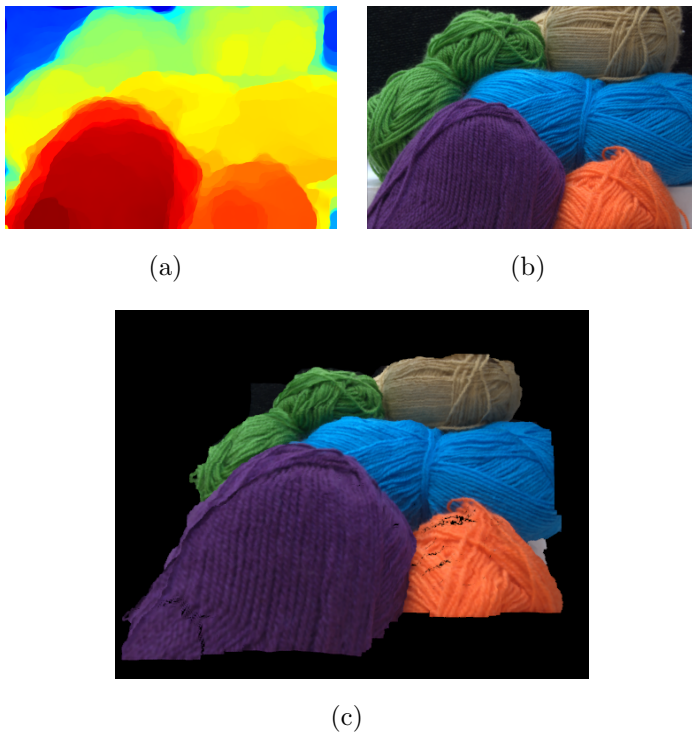
**Tabla 6.1:** *Parámetros empleados en la ejecución del algoritmo sobre las escenas adquiridas con el método de multiplexación temporal.*

## 6.1. Imágenes adquiridas mediante multiplexación temporal

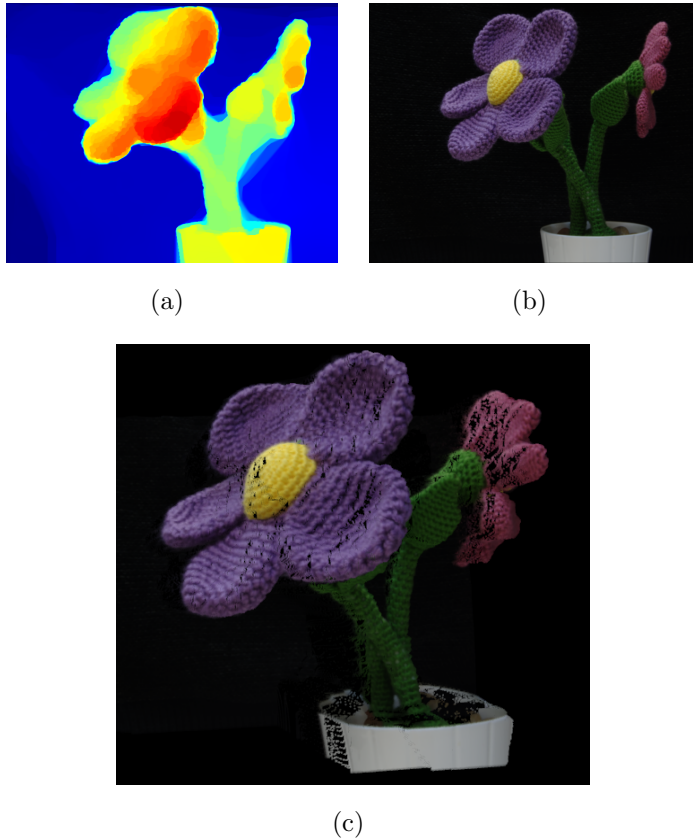
En esta sección se muestran los resultados obtenidos sobre tres imágenes reales adquiridas mediante la técnica de multiplexación temporal descrita en la sección 3.1. En este caso, no se dispone de un mapa de distancias de referencia, por lo que únicamente es posible realizar una evaluación cualitativa de los resultados. En las figuras 6.1, 6.2 y 6.3 se muestran los mapas de distancias obtenidos para los tres *light fields* descritos en la sección 3.1, junto con la imagen totalmente enfocada resultante de utilizar los datos geométricos obtenidos como superficie de enfoque variable en el método de Isaksen et al. (2000). Además, también se muestra una imagen de las nubes de puntos contruidas a partir de los mapas de distancias y las imágenes totalmente enfocadas. Los parámetros de ejecución del algoritmo se indican en la tabla 6.1.

## 6.1 IMÁGENES ADQUIRIDAS MEDIANTE MULTIPLEXACIÓN TEMPORAL

---



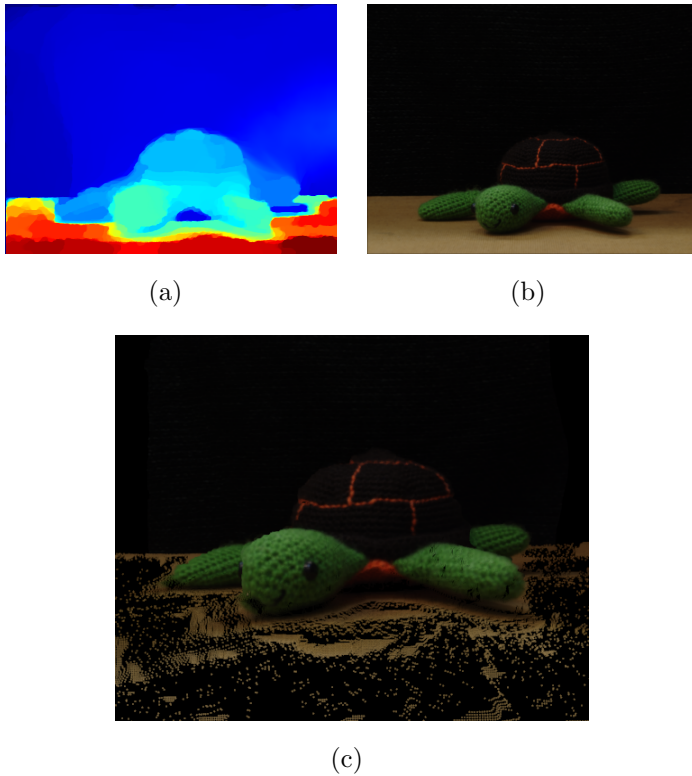
**Figura 6.1:** Resultados obtenidos para la escena *ovillos*. (a) Mapa de pendientes (Colores cálidos: cerca; Colores fríos: lejos). (b) Imagen totalmente enfocada. (c) Nube de puntos en tres dimensiones.



**Figura 6.2:** Resultados obtenidos para la escena *flores3*. (a) Mapa de pendientes (Colores cálidos: cerca; Colores fríos: lejos). (b) Imagen totalmente enfocada. (c) Nube de puntos en tres dimensiones.

## 6.1 IMÁGENES ADQUIRIDAS MEDIANTE MULTIPLEXACIÓN TEMPORAL

---



**Figura 6.3:** Resultados obtenidos para la escena *tortuga*. (a) Mapa de pendientes (Colores cálidos: cerca; Colores fríos: lejos). (b) Imagen totalmente enfocada. (c) Nube de puntos en tres dimensiones.

## 6.2. Imágenes adquiridas mediante la cámara CAFADIS

En esta sección se presentan los resultados obtenidos al aplicar el método propuesto sobre los *light fields* adquiridos con la cámara CAFADIS, cuyos parámetros ópticos se han descrito en la sección 3.2. Los parámetros de ejecución del algoritmo para cada uno de los *light fields* se muestran en la tabla 6.2.

En la figura 6.4, se muestra el mapa de pendientes, la imagen renderizada con el método de Isaksen et al. (2000) y la nube de puntos obtenida a partir del mapa de distancias estimado con el modelo TV-L1, para el *light field* etiquetado como muñecos-24. La resolución en píxeles corresponde con el número de microlentes ( $166 \times 93$ ). En la figura se observa como el efecto de viñeteo genera un oscurecimiento de los bordes de la imagen de color y da lugar a errores en el cálculo de la pendiente en esas zonas.

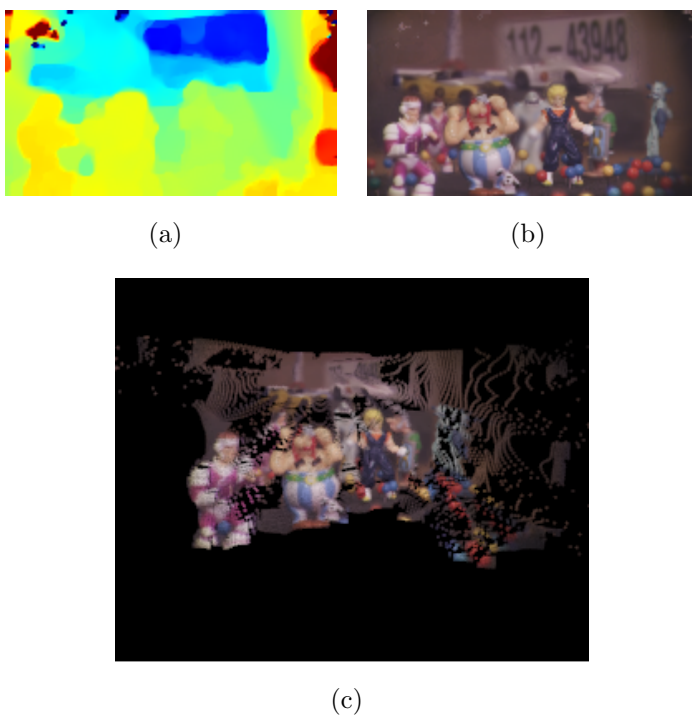
Para generar un mapa de distancias con un número de píxeles mayor, se ha aumentado la escala del sensor virtual sobre el que se proyectan las medidas de distancias. En la figura 6.5, se muestra el resultado del aumento de escala, pudiéndose observar el mapa de distancias, junto a la imagen de color renderizada a partir del mismo.

El inconveniente de aumentar la escala es que la imagen de color puede presentar artefactos en zonas donde la distancia no está bien estimada o donde no hay píxeles en la parametrización cuadrada. Esto

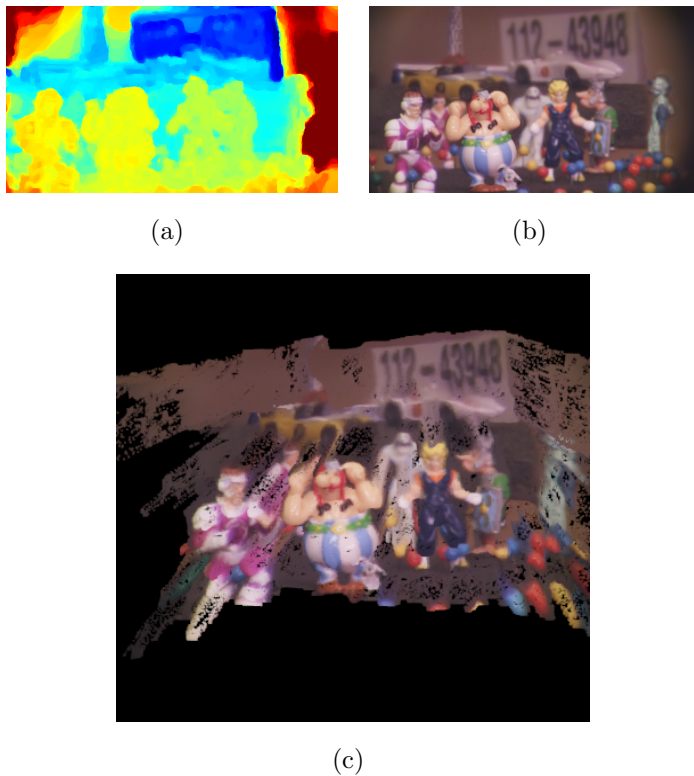


## 6.2 IMÁGENES ADQUIRIDAS MEDIANTE LA CÁMARA CAFADIS

---



**Figura 6.4:** Resultados obtenidos para la escena *muñecos-24*. (a) Mapa de pendientes (Colores cálidos: cerca; Colores fríos: lejos). (b) Imagen renderizada. (c) Nube de puntos en tres dimensiones.



**Figura 6.5:** Resultados obtenidos al aumentar la escala  $\times 7$  para la escena *muñecos-15*. (a) Mapa de pendientes (Colores cálidos: cerca; Colores fríos: lejos). (b) Imagen renderizada. (c) Nube de puntos en tres dimensiones.

### 6.3 ESCENA DE LOS CUBOS

---

| LF         | <i>cthr1</i> | <i>cthr2</i> | iteraciones | $\lambda_f$ |
|------------|--------------|--------------|-------------|-------------|
| muñecos-24 | 0.75         | 0.7          | 100         | 0.8         |
| muñecos-15 | 0.75         | 0.7          | 750         | 0.8         |

**Tabla 6.2:** *Parámetros empleados en la ejecución del algoritmo sobre las escenas adquiridas con CAFADIS.*

último se debe a que la apertura de la cámara es circular. Por ello, se ha utilizado el conjunto `muñecos-15` obtenido al recortar los píxeles que se encuentran en el interior del cuadrado inscrito en la circunferencia de cada microimagen. Además, como queda reflejado en la tabla 6.2, se ha aumentado el número de iteraciones ya que el tamaño de la imagen de salida es mayor.

En la nube de puntos de la figura 6.5(c), se observa el aumento de densidad de los puntos respecto a la figura 6.4(c), ya que la imagen final obtenida tiene  $1162 \times 651$  píxeles.

### 6.3. Escena de los cubos

A continuación, se presentan los resultados obtenidos con el operador local para la escena de los cubos cuyo *light field* se ha obtenido mediante el SIDEV. Los detalles de la adquisición de este *light field* se encuentran descritos en la sección 3.3.

Se han calculado diferentes medidas de error, tanto para el operador de detección de pendientes local en 4D, como para el

| Métrica     | Operador local 4D | Regularización TV-L1 |
|-------------|-------------------|----------------------|
| Tiempo(ms)  | 150               | 200                  |
| MAE         | 0,0708            | 0,0420               |
| MSE         | 0,0215            | 0,0091               |
| BMP(> 0,1)  | 4,60 %            | 0,64 %               |
| BMP(> 0,05) | 34,04 %           | 32,70 %              |

**Tabla 6.3:** Métricas de error para la escena de los cubos.

resultado regularizado con el modelo TV-L1. En la tabla 6.3 se presentan, para ambos casos, el tiempo de ejecución, el error absoluto medio (MAE), el error cuadrático medio (MSE) y el porcentaje de *bad matching pixels* para errores absolutos de pendiente mayores que 0,1 y 0,05. Debe tenerse en cuenta que para el caso del operador local sólo se consideran los píxeles que superan el *cthr2*, que se ha fijado a 0,5.

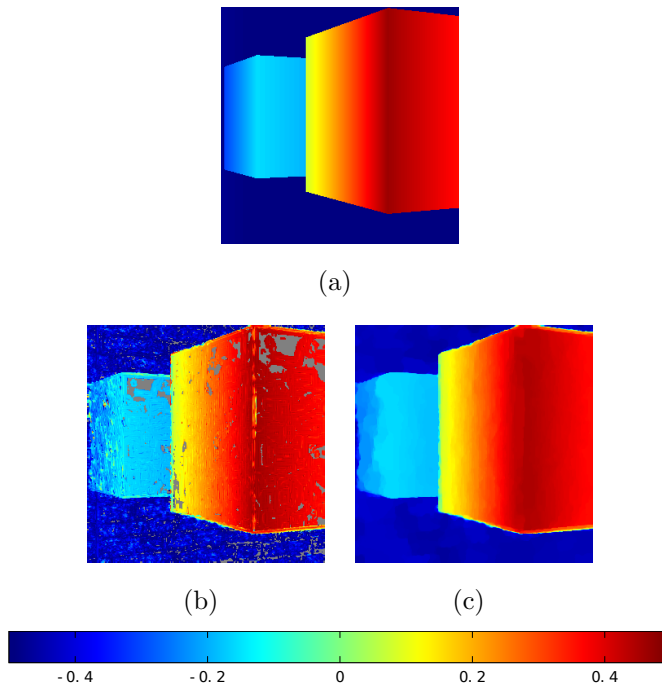
La figura 6.6 muestra el mapa de referencia, el resultado obtenido tras aplicar el operador local y ,finalmente, el resultado que se obtiene al aplicar la regularización TV-L1.

## 6.4. Conjunto de datos del *Heidelberg Collaboratory for Image Processing*

En esta sección se muestran los resultados obtenidos al aplicar los métodos propuestos en los capítulos anteriores sobre el conjunto de datos del *Heidelberg Collaboratory for Image Processing* (Wanner et

## 6.4 CONJUNTO DE DATOS DEL *Heidelberg Collaboratory for Image Processing*

---



**Figura 6.6:** *Mapas de distancias regularizados para la escena de los cubos. (a) Mapa de referencia. (b) Mapa incompleto obtenido tras aplicar el método local 4D. (c) Mapa denso regularizado con el modelo TV-L1.*

al. 2013). Los detalles acerca de este conjunto de datos se han descrito en la sección 3.4.

Para cada escena se han medido los tiempos de ejecución, que pueden variar ligeramente debido al paso de agregación explicado en la sección 4.4. También se ha medido el error absoluto medio (MAE) respecto del mapa de distancias de referencia, el error cuadrático medio (MSE) y el porcentaje de píxeles en los que el error de pendientes  $E_{\hat{s}}$

supera un determinado umbral (BMP). Estas medidas se han tomado para las pendientes obtenidas tras aplicar sólo el operador propuesto en el capítulo 4 y luego a los mapas estimados mediante la optimización del modelo TV-L1. Los datos obtenidos se muestran en las siguientes secciones.

#### 6.4.1. Aplicando sólo el operador local en cuatro dimensiones

Inicialmente, se han obtenido las métricas de error aplicando sólo el método local de detección de pendientes 4D propuesto en el capítulo 4. Los resultados numéricos de comparar el mapa obtenido con el mapa de referencia, se muestran en la tabla 6.4. El valor de  $cthr1$  utilizado fue 0,9 y el valor de  $cthr2$  fue 0,8. Tanto los errores como los porcentajes se han calculado para los píxeles cuya confianza supera  $cthr2$ .

Por otro lado, en la figura 6.7 se muestran los mapas de referencia, la imagen central del *light field*, junto con los mapas incompletos de pendientes obtenidos tras aplicar el método propuesto. Los mapas de pendientes de la figura se encuentran codificados por colores, donde los colores cálidos indican distancias cercanas y los colores fríos distancias lejanas. Los píxeles en gris indican que no se dispone del dato por no superar el umbral de confianza establecido. Por otra parte, en la última columna se muestra la imagen de color en escala de grises, donde los píxeles con matiz verde indican que el error de pendiente  $E_{\hat{s}}$  es menor

## 6.4 CONJUNTO DE DATOS DEL *Heidelberg Collaboratory for Image Processing*

| LF        | Tiempo (ms) | MAE    | MSE    | BMP(> 0,1) | BMP(> 0,05) |
|-----------|-------------|--------|--------|------------|-------------|
| buddha    | 400         | 0,0394 | 0,0151 | 7,04 %     | 15,81 %     |
| buddha2   | 420         | 0,0616 | 0,0114 | 15,09 %    | 36,30 %     |
| horses    | 430         | 0,0904 | 0,0558 | 18,14 %    | 39,29 %     |
| medieval  | 290         | 0,0759 | 0,0239 | 19,53 %    | 44,18 %     |
| monasRoom | 390         | 0,0453 | 0,0152 | 7,66 %     | 20,02 %     |
| papillon  | 370         | 0,0847 | 0,0343 | 20,73 %    | 43,65 %     |
| stillLife | 450         | 0,1612 | 0,1554 | 32,41 %    | 50,65 %     |
| couple    | 330         | 0,0554 | 0,0077 | 12,43 %    | 41,06 %     |
| cube      | 330         | 0,0667 | 0,0257 | 15,59 %    | 36,44 %     |
| maria     | 360         | 0,0538 | 0,0152 | 11,25 %    | 38,25 %     |
| pyramide  | 340         | 0,0498 | 0,0186 | 9,86 %     | 22,81 %     |
| statue    | 340         | 0,1054 | 0,1078 | 22,81 %    | 45,48 %     |
| PROMEDIO  | 371         | 0,0741 | 0,0405 | 16,04 %    | 36,16 %     |

**Tabla 6.4:** *Medidas de error para el conjunto de datos del HCI aplicando el operador local en cuatro dimensiones propuesto.*

que 0,05, los píxeles con matiz rojo indican que el error es mayor, y los píxeles con tono gris indican que el algoritmo no generó un resultado para este punto.

En la figura 6.8 se muestran los resultados obtenidos con los *light fields* reales del conjunto de datos. Hay que tener en cuenta que los datos de referencia fueron obtenidos con un escáner de luz estructurada y postprocesados para introducirlos en Blender, con lo que no se dispone de datos de referencia para todos los puntos de la imagen. Por ello no es posible obtener una medida de error para todos los puntos. Los puntos que no permiten obtener una medida de error se han matizado en azul en el mapa de la última columna.

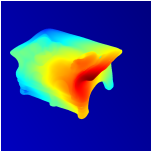

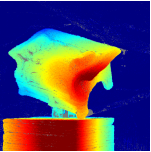
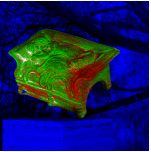
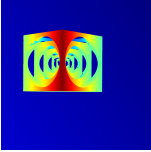

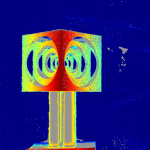
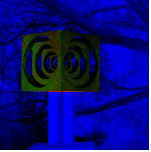
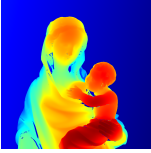

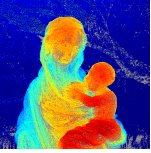
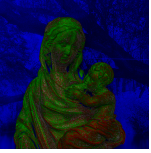
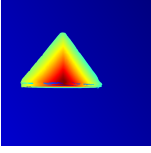

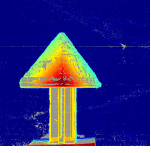
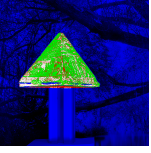
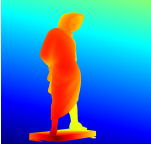

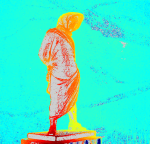
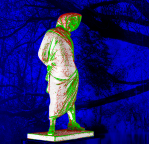
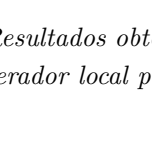
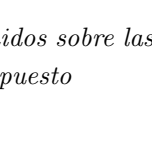
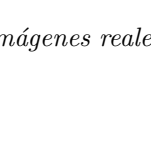
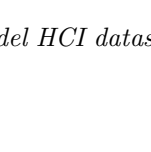












| Dataset   | Datos de referencia |       | Operador de detección propuesto |                             |
|-----------|---------------------|-------|---------------------------------|-----------------------------|
|           | Ground truth        | Color | Mapa de pendientes              | Bad matching pixels (>0.05) |
| buddha    |                     |       |                                 |                             |
| buddha2   |                     |       |                                 |                             |
| horses    |                     |       |                                 |                             |
| medieval  |                     |       |                                 |                             |
| monasRoom |                     |       |                                 |                             |
| papillon  |                     |       |                                 |                             |
| stillLife |                     |       |                                 |                             |

**Figura 6.7:** Resultados obtenidos sobre las imágenes sintéticas del HCI dataset, aplicando directamente el operador local propuesto.



## 6.4 CONJUNTO DE DATOS DEL *Heidelberg Collaboratory for Image Processing*

---

| Dataset  | Datos de referencia   |   | Operador local propuesto  |  |
|----------|---|---|---|--|
|          | Ground truth  | Color   | Mapa de pendientes  | Bad matching pixels(>0.05)   |
| couple   |    |    |    |    |
|          |    |    |    |    |
| cube     |    |    |    |    |
|          |   |   |   |   |
| maria    |  |  |  |  |
|          |  |  |  |  |
| pyramide |  |  |  |  |
|          |  |  |  |  |
| statue   |  |  |  |  |
|          |  |  |  |  |

**Figura 6.8:** Resultados obtenidos sobre las imágenes reales del HCI dataset, aplicando el operador local propuesto

### 6.4.2. Regularización TV-L1

A continuación se muestran los resultados obtenidos al aplicar la regularización TV-L1 a las medidas obtenidas con el operador local. El parámetro  $\lambda_f$  del algoritmo split Bregman se ha elegido para optimizar el promedio del MSE para todos los *light field* del *dataset*, mientras que para el número de iteraciones se ha elegido un valor que garantice la convergencia en todos los casos. Los valores de  $\gamma_1$  y  $\gamma_2$  no afectan significativamente al resultado final, por lo que se han elegido empíricamente. Los valores concretos utilizados se muestran en la tabla 6.5.

La tabla 6.6 recoge los resultados para distintas métricas de error tras aplicar la regularización mediante el modelo TV-L1 a los datos iniciales. En las figuras 6.9 y 6.10 se muestran los mismos datos que se mostraron en las figuras 6.7 y 6.8 para el método local sin regularizar, pero aplicando ahora la regularización TV-L1. Finalmente, en la tabla 6.7, se muestran los errores cuadráticos medios multiplicados por 100 utilizando la mejor combinación de parámetros para cada uno de los *light fields* del conjunto de datos.

## 6.4 CONJUNTO DE DATOS DEL *Heidelberg Collaboratory for Image Processing*

---

| Parámetro   | Valor |
|-------------|-------|
| $\lambda_f$ | 0,4   |
| iteraciones | 2000  |
| $\gamma_1$  | 5     |
| $\gamma_2$  | 8     |

**Tabla 6.5:** *Parámetros del algoritmo split Bregman al ejecutarlo sobre los datos del HCI dataset.*

| LF        | Tiempo (ms) | MAE    | MSE    | BMP(> 0,1) | BMP(> 0,05) |
|-----------|-------------|--------|--------|------------|-------------|
| buddha    | 1550        | 0,0262 | 0,0060 | 3,39 %     | 8,91 %      |
| buddha2   | 1560        | 0,0486 | 0,0071 | 8,17 %     | 26,12 %     |
| horses    | 1650        | 0,0620 | 0,0145 | 12,54 %    | 30,80 %     |
| medieval  | 1740        | 0,0411 | 0,0118 | 4,16 %     | 16,85 %     |
| monasRoom | 1550        | 0,0385 | 0,0089 | 6,06 %     | 14,31 %     |
| papillon  | 1520        | 0,0605 | 0,0206 | 8,77 %     | 37,25 %     |
| stillLife | 1600        | 0,0995 | 0,0394 | 29,48 %    | 48,27 %     |
| couple    | 1960        | 0,0413 | 0,0037 | 4,34 %     | 27,51 %     |
| cube      | 1970        | 0,0309 | 0,0070 | 4,47 %     | 9,17 %      |
| maria     | 2040        | 0,0322 | 0,0016 | 0,80 %     | 21,71 %     |
| pyramide  | 1980        | 0,0301 | 0,0041 | 3,01 %     | 8,85 %      |
| statue    | 1980        | 0,0423 | 0,0040 | 8,30 %     | 26,25 %     |
| PROMEDIO  | 1758        | 0,0461 | 0,0107 | 7,79 %     | 23,00 %     |

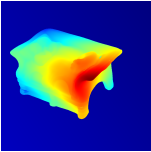

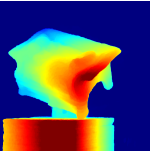
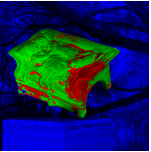
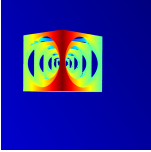

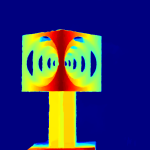
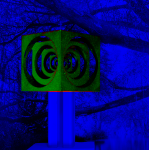
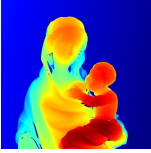

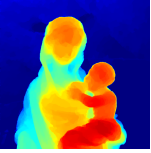
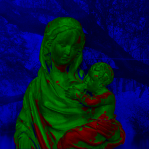
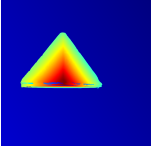

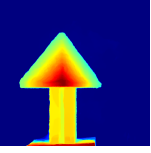
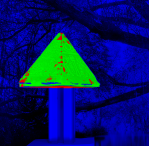
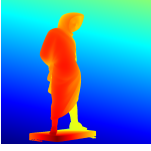

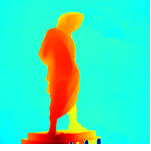

**Tabla 6.6:** *Medidas de error para el conjunto de datos del HCI regularizando con TV-L1.*

| Dataset   | Datos de referencia |       | Regularización TV-L1 |                             |
|-----------|---------------------|-------|----------------------|-----------------------------|
|           | Ground truth        | Color | Mapa de pendientes   | Bad matching pixels (>0.05) |
| buddha    |                     |       |                      |                             |
| buddha2   |                     |       |                      |                             |
| horses    |                     |       |                      |                             |
| medieval  |                     |       |                      |                             |
| monasRoom |                     |       |                      |                             |
| papillon  |                     |       |                      |                             |
| stillLife |                     |       |                      |                             |

**Figura 6.9:** Resultados obtenidos sobre las imágenes sintéticas del HCI dataset, aplicando la regularización TV-L1.

## 6.4 CONJUNTO DE DATOS DEL *Heidelberg Collaboratory for Image Processing*

---

| Dataset | Datos de referencia   |   | Operador local propuesto  |  |
|---------|---|---|---|--|
|         | Ground truth  | Color   | Mapa de pendientes  | Bad matching pixels(>0.05)   |
| couple  |    |    |    |    |
|         |    |    |    |    |
| maria   |    |    |    |    |
|         |   |   |   |   |
| statue  |  |  |  |  |

**Figura 6.10:** Resultados obtenidos sobre las imágenes reales del HCI dataset, aplicando la regularización  $TV-L1$ .

| LF        | MSE  |
|-----------|------|
| buddha    | 0,60 |
| buddha2   | 0,70 |
| horses    | 1,36 |
| medieval  | 1,10 |
| monasRoom | 0,86 |
| papillon  | 2,00 |
| stillLife | 3,90 |
| couple    | 0,33 |
| cube      | 0,70 |
| maria     | 0,17 |
| pyramide  | 0,41 |
| statue    | 0,39 |
| PROMEDIO  | 1,04 |

**Tabla 6.7:** *Error cuadrático medio multiplicado por 100 optimizando los parámetros del algoritmo para cada conjunto de datos.*

## 6.5. Discusión

En las secciones anteriores se han mostrado los resultados obtenidos al aplicar los métodos, desarrollados en los capítulos 4 y 5, a una serie de *light fields* obtenidos con los distintos métodos descritos en el capítulo 3.

Con respecto a los *light field* de entrada generados, es conveniente diferenciar aquellos con un número de muestras angulares elevado, como los obtenidos con la cámara CAFADIS, de los que tienen un número de muestras reducido como los que se han obtenido mediante

## 6.5 DISCUSIÓN

---

el método de multiplexación temporal explicado en la sección 3.1.

Los resultados obtenidos con ambos conjuntos de datos se han descrito en las secciones 6.1 y 6.2. Dichos resultados únicamente permiten una evaluación cualitativa ya que se trata de imágenes reales para las que no se dispone de un mapa de distancias de referencia. Los resultados obtenidos a partir de los *light fields* capturados con multiplexación temporal demuestran que, utilizando el método local propuesto, es posible obtener resultados satisfactorios aún cuando el número de muestras angulares es reducido, con la ventaja poder obtener mayor resolución espacial en la salida. El aumento de la resolución angular en el caso de una cámara plenóptica conlleva una reducción de la resolución espacial. Por ello, al aplicar directamente el método propuesto se obtienen mapas de distancias e imágenes de color de una resolución reducida, como los  $166 \times 93$  píxeles de la figura 6.4. Este efecto se puede contrarrestar aumentando el número de píxeles del sensor virtual sobre el que se proyectan las medidas de distancias como se muestra en la figura 6.5. Sin embargo, no se observa ningún aumento significativo del detalle en el mapa de distancias y además la imagen de color puede contener artefactos. Por tanto, cuando el proceso de adquisición no introduce una cantidad significativa de ruido, no es conveniente el uso de un número elevado de muestras angulares, ya que únicamente parecen aportar información redundante.

Además de realizar pruebas sobre imágenes reales, se han obtenido métricas de error sobre la imagen sintética de los cubos, así como sobre el conjunto de datos de prueba del HCI. Esto ha permitido establecer

algunas limitaciones del método propuesto. En primer lugar, no es posible determinar información de distancias en zonas con textura insuficiente. Esto se puede advertir en la figura 6.7. En los mapas de la tercera columna aparecen zonas marcadas con gris que, en su mayoría, coinciden con las zonas poco texturizadas que aparecen en la imagen de color. El efecto se puede apreciar claramente en la escena *papillon* en la que la superficie de las hojas aparece en gris. Por otra parte, pueden producirse errores significativos cuando el contenido frecuencial de las texturas se encuentra en zonas del espectro en las que el estimador de orientación comete errores sistemáticos altos. Esto se puede observar en el mantel del fondo del *light field* del *stillLife* de la figura 6.7. Este segundo problema podría solventarse mejorando los filtros derivativos.

A pesar de este inconveniente, hay que recordar que el uso de métodos locales, como el método aquí propuesto, tiene la ventaja de es aplicable a escenas no estrictamente lambertianas, reduciendo la restricción lambertiana al entorno local al que se aplica el operador. Otra ventaja añadida del método aquí propuesto es que no se discretizan las medidas de distancia ni se elige la distancia más probable entre un conjunto de distancias candidatas, lo que hace que la complejidad computacional sea independiente de la resolución en distancia.

Para superar las limitaciones en las zonas con insuficiente textura, se propuso el uso del modelo TV-L1 para regularizar los mapas incompletos de distancias. Comparando los resultados obtenidos al



## 6.5 DISCUSIÓN

---

aplicar únicamente el operador local en 4D con los resultados obtenidos tras la regularización TV-L1, se puede observar que en todos los casos se produce una mejora en los errores cuadráticos medios. Pero, hay que tener en cuenta que el error cuadrático medio para el operador local (tabla 6.4) se ha calculado utilizando sólo aquellas medidas en las que *cthr2* superaba 0,8 mientras que para la regularización TV-L1 se ha empleado la totalidad de los píxeles. También se puede comprobar esta mejora en los datos que se muestran en la tabla 6.3 para la escena de los cubos generada con el SIDEV.

El uso de un modelo como el TV-L1 permite inferir un valor para la distancia cuando éste no se ha podido medir, así como eliminar ruido de medida. Estos valores inferidos se obtienen a partir de los valores de los píxeles cercanos. En un área con medidas, el modelo TV-L1 realiza un suavizado que elimina el ruido aleatorio. Sin embargo, si en una zona las medidas tienen una tendencia errónea, el hecho de aplicar la regularización no mejora los resultados. Esto se puede comprobar comparando las zonas marcadas en rojo en las imágenes de la cuarta columna de las figuras 6.7 y 6.9. Las zonas que presentaban un error de medida alto continúan estando presentes tras el proceso de estimación, puesto que el mapa estimado debe ajustarse a las medidas.

Finalmente, se ha comparado la precisión del método propuesto con otros métodos evaluados sobre el HCI dataset. En Wanner et al. (2013) se muestra una comparativa de distintos algoritmos de estimación de distancias. Comparando el MSE promedio, que se muestra en la tabla 6.7, con los algoritmos evaluados en Wanner et al. (2013),

cuyos MSE promedio se muestran en la tabla 6.8, se observa como el MSE del método propuesto sólo es mejorado por uno de los métodos allí evaluados. En cuanto a la velocidad de ejecución, el algoritmo propuesto se puede clasificar en la categoría de rápido con un tiempo de ejecución promedio de 1758 ms, mientras que los algoritmos que lo superan o igualan en precisión caen en las categorías de velocidad media y lenta. Además, hay que considerar que algunos aspectos de la implementación no están optimizados y que los tiempos indicados se reducirían aún más con una implementación jerárquica como la explicada en la sección 5.2.

Hay que hacer notar que estos valores de MSE se han obtenido utilizando el mejor conjunto de parámetros de cada algoritmo sobre cada conjunto de datos, por lo que los datos de la tabla 6.8 pueden ser excesivamente optimistas. Si se compara el MSE promedio de la tabla 6.6, obtenido con los mismos parámetros para todos los *light fields*, se observa que sólo es superado o igualado en precisión por tres de los métodos mostrados en la tabla 6.8, lo que indica que, incluso utilizando una métrica más conservadora, el método propuesto tiene un rendimiento que puede competir con los métodos evaluados en Wanner et al. (2013).

## 6.5 DISCUSIÓN

---

| Algoritmo               | Velocidad     | MSE*100     |
|-------------------------|---------------|-------------|
| <b>Método propuesto</b> | <b>rápida</b> | <b>1,04</b> |
| EPLS                    | rápida        | 1,07        |
| ST_CH_S                 | rápida        | 1,14        |
| ST_AV_S                 | rápida        | 1,19        |
| EPLL                    | rápida        | 1,64        |
| ST_AV_L                 | rápida        | 2,72        |
| ST_CH_L                 | rápida        | 3,54        |
| EPLC                    | media         | 1,04        |
| ST_CH_G                 | lenta         | 1,01        |
| ST_AV_G                 | lenta         | 1,12        |
| EPLG                    | lenta         | 1,18        |

**Tabla 6.8:** *Clasificación de algoritmos de estimación de distancias de Wanner et al. (2013). Velocidad: rápida (menos de cinco segundos), media(cinco segundos a un minuto), y lenta (más de un minuto). MSE\*100: Error cuadrático medio de pendiente multiplicado por 100. Datos extraídos de Wanner et al. (2013).*



# Capítulo 7

## Conclusiones

*“Si lloras por haber perdido el sol, las lágrimas te impedirán ver las estrellas.”*

---

Rabindranath Tagore

Las investigaciones de esta tesis doctoral se han centrado en el desarrollo de un método local para la medida de distancias a partir del *light field*, siendo las conclusiones las siguientes:

- El nuevo método desarrollado permite la medida de distancias de forma directa a partir del *light field*, utilizando un número reducido de vistas, lo que permite aumentar la resolución espacial de cada una de ellas.
- Para extraer la información de distancias se desarrollado un

operador local, por lo que no se necesita que la hipótesis lambertiana se cumpla de modo global, sino solamente en el entorno local lo que implica una mayor robustez frente a métodos globales.

- El operador se ha basado en el diseño de una base local del espacio gradiente en la que se maximiza la cantidad de energía que responde al modelo ideal del *light field*. Lo que simultáneamente permite la estimación de un valor de confianza, que indica el grado de correlación con el modelo.
- La estimación del gradiente local mediante operadores aproximados permite aumentar la eficiencia de los métodos de cálculo, aunque introduciendo errores sistemáticos que dependen del contenido frecuencial de la imagen.
- Se han estudiado los errores sistemáticos introducidos por la aproximación de los estimadores de derivadas en cuatro dimensiones, estableciendo que el comportamiento es similar al de sus homólogos en dos dimensiones.
- Se ha determinado que no es posible estimar una única cota del error sistemático ya que tiene una dependencia directa de los datos, tanto de la distancia, como del color y su distribución en frecuencia.
- La precisión de las medidas de distancias depende del módulo cuadrado del vector gradiente, promediado en color, en el entorno

---

de cada punto, de la distancia a la que se encuentra el objeto al que corresponde el punto y de la varianza del ruido de la imagen.

- Se ha desarrollado un método para combinar múltiples mapas de distancias obtenidos con el método local en un único mapa de distancias. Dicho método es eficiente cuando el número de vistas es reducido, pero debe ser revisado al utilizar un número elevado de vistas.
- Como los mapas de distancias obtenidos con el método local pueden ser incompletos, se ha propuesto utilizar el modelo TV-L1 optimizado mediante el método split Bregman para obtener mapas de distancias densos.
- El número de iteraciones del método split Bregman, depende del área a rellenar. Por tanto, a mayor tamaño de imagen mayor deberá ser el número de iteraciones. Este efecto se reduce aplicando técnicas jerárquicas en el caso de imágenes estáticas o bien realizando un procesamiento encadenado para partir de resultados anteriores en el caso de secuencias temporales.
- La complejidad computacional y los requerimientos de memoria dependen del número de rayos del *light field* y del tamaño del mapa de distancias de salida y no de la resolución en distancias.
- El método propuesto ha sido clasificado siguiendo los criterios propuestos en Wanner et al. (2013), resultando ser el más rápido y preciso de su categoría.





# Apéndice **A**

## Formación de imágenes y su relación con el *light field*

Según lo que se ha descrito en la sección 2.1, la función plenóptica establece una ligadura entre los objetos de la escena y sus imágenes. El proceso de formación de imagen puede asimilarse a un conjunto de operaciones que tiene lugar sobre la función plenóptica, o sobre una parte de la misma, y termina dando lugar a una imagen de la escena. En esta sección se revisarán los elementos que intervienen en el proceso de formación de imágenes, cómo actúan sobre la función plenóptica y las consecuencias en el proceso de reconstrucción de distancias que se describirá en el capítulo 4.

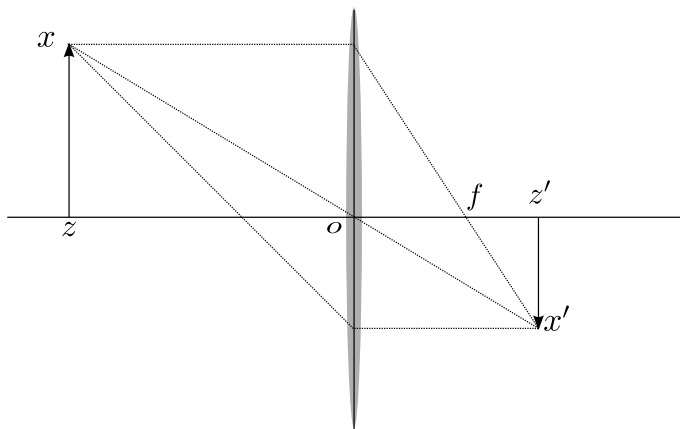
Una imagen podemos entenderla como una función bidimensional

$I(x, y)$  que registra los valores de intensidad de la luz procedente de los objetos de la escena, en una o más bandas del espectro. Un sistema de captura de imágenes generalmente está formado por dispositivos para cambiar la dirección de los rayos de luz, es decir lentes, y un dispositivo que mide la intensidad de la luz que incide sobre él, esto es, un sensor. La combinación de estos elementos forma una cámara.

Las lentes producen un cambio controlado de la dirección de propagación de la luz, que se consigue mediante difracción, refracción y reflexión. El modelo más sencillo de lente es la lente delgada, en el que se obvian los efectos de difracción y reflexión. La lente se define con un eje óptico, un plano focal perpendicular al eje óptico, y una apertura centrada en el centro óptico  $o$ , que coincide con la intersección entre el eje óptico y el plano focal. Matemáticamente toda lente delgada se define con dos parámetros, su longitud focal  $f$  y su diámetro  $d$ , considerando que la lente es circular. Los rayos procedentes de un punto situado a una distancia  $z$  del plano de la lente inciden sobre la superficie de la lente en el plano de la apertura, donde se modifica su dirección haciéndolos converger en un punto imagen situado a una distancia  $z'$ . La relación entre  $z$  y  $z'$  se encuentra en la ecuación de la lente delgada y depende de  $f$ :

$$\frac{1}{z} + \frac{1}{z'} = \frac{1}{f} \quad (\text{A.1})$$

En la figura A.1 se muestra el proceso de formación de imagen de una lente delgada. La lente delgada se caracteriza por que los rayos que



**Figura A.1:** Esquema del proceso de formación de imagen de una lente delgada.

inciden paralelos al eje óptico al atravesar la lente se desvían y pasan por el foco situado a distancia  $f$ . Por otro lado, los rayos que atraviesan el centro óptico no se desvían. Esto permite determinar gráficamente la posición del punto imagen  $(x', z')$  a partir de la posición del punto objeto  $(x, z)$  frente a la lente. La relación entre  $z$  y  $z'$  viene dada por la ecuación A.1, mientras que la relación entre  $x$  y  $x'$  viene dada por:

$$x' = -\frac{z'}{z}x = Mx \quad (\text{A.2})$$

donde  $M$  es la magnificación de la lente.

Las ecuaciones A.1 y A.2 sólo indican en qué posición se formará la imagen de un objeto pero en ningún momento tienen en cuenta la apertura  $d$ . Este parámetro suele relacionarse con la distancia focal  $f$  por medio del número  $F_{\#}$ , que se mencionará más adelante, y que

viene dado por la siguiente expresión:

$$F_{\#} = \frac{f}{d} \quad (\text{A.3})$$

Hay que tener en cuenta que generalmente los sistemas ópticos, como los objetivos fotográficos, requieren de más de una lente para exhibir un comportamiento como el que se ha descrito. Sin embargo, su especificación suele venir dada por la distancia focal  $f$  y el número  $F_{\#}$ .

El segundo elemento de un sistema de captura de imágenes es el sensor, que es un dispositivo que integra la intensidad de la luz en el tiempo, en una banda del espectro y angularmente para cada punto  $(x, y)$  de su superficie. El funcionamiento de un sensor podría modelarse relacionando la función plenóptica tras una lente con la imagen que éste registra. Este proceso se indica en la siguiente expresión:

$$I(x, y) = \int_{t_0}^{t_1} \int_{\lambda_0}^{\lambda_1} \int \int P(x, y, \Theta, \Phi, \lambda, t) d\Theta d\Phi d\lambda dt \quad (\text{A.4})$$

donde  $t_1 - t_0$  se corresponde con el tiempo de exposición y el intervalo  $[\lambda_0, \lambda_1]$  es la banda del espectro a la que es sensible dicho sensor.

La ecuación A.4 muestra como el proceso de formación de imagen en una cámara convencional, al integrarse en ángulo, destruye la información angular. Sin embargo, es necesario disponer de dicha información angular para obtener medidas de distancia a partir de la función plenóptica, ya que se necesitan al menos dos muestras de las dimensiones angulares para poder localizar un punto en el espacio.

# Apéndice **B**

## Cálculo de errores en la estimación de distancia

En este apéndice se aborda la propagación de los errores debidos a ruido en las imágenes plenópticas hasta la obtención de un resultado en el mapa de distancias. Suponiendo que las imágenes plenópticas están alteradas por ruido blanco gaussiano, éste se propaga a través de la expresión para estimar la distancia para así obtener el error cometido. Las propagaciones de error se hacen teniendo en cuenta las fórmulas de propagación del NIST (Ku, 1966).

## B.1. Cálculo errores aleatorios en la estimación de la distancia a partir de imágenes epipolares

La estimación de distancia se realiza por medio de la expresión 2.10 que se muestra a continuación:

$$\hat{\theta} = \frac{1}{2} \arctan \left( \frac{2\widehat{I}_x\widehat{I}_u}{\widehat{I}_x^2 - \widehat{I}_u^2} \right) \quad (\text{B.1})$$

Para mayor claridad se denotarán el numerador y el denominador con  $A$  y  $B$ :

$$\begin{aligned} A &= 2\widehat{I}_x\widehat{I}_u \\ B &= \widehat{I}_x^2 - \widehat{I}_u^2 \end{aligned} \quad (\text{B.2})$$

La derivada parcial de una imagen epipolar respecto de una de sus variables se calcula como:

$$I_{x_i} = \sum_{\mathbf{x}} a_{x_i}(\mathbf{x})I(\mathbf{x}) \quad (\text{B.3})$$

Suponiendo que la varianza del error todos los puntos de la imagen epipolar es  $\sigma_I$  y que el ruido en las distintas muestras no está correlado, la varianza del error en la derivada parcial viene dada por:

$$\sigma_{I_{x_i}}^2 = \left[ \sum_{\mathbf{x}} a_{x_i}(\mathbf{x})^2 \right] \sigma_I^2 = |p|^2 |d_1|^2 \sigma_I^2 \quad (\text{B.4})$$

## B.1 CÁLCULO ERRORES ALEATORIOS EN LA ESTIMACIÓN DE LA DISTANCIA A PARTIR DE IMÁGENES EPIPOLARES

---

Ahora propagando el error obtenido en la ecuación B.4 en el cálculo del producto de dos derivadas se obtiene:

$$\sigma_{I_x I_u}^2 = |p|^2 |d_1|^2 \sigma_I^2 [I_x^2 + I_u^2] \quad (\text{B.5})$$

A continuación, se obtiene el error en el numerador propagando el resultado de la ecuación B.5 en la expresión del numerador teniendo en cuenta el operador  $\hat{\cdot}$  se obtiene como resultado:

$$\sigma_A^2 = 4|p|^2 |d_1|^2 \sigma_I^2 \eta^2 \quad (\text{B.6})$$

con  $\eta^2 = \sum_{i,j} \omega(i,j)^2 [I_x(i,j)^2 + I_u(i,j)^2] + \varphi$ , siendo  $\varphi$  un término dependiente de la correlación entre los términos del sumatorio.

Por otra parte, la propagación del error de la derivada en el cálculo de la derivada al cuadrado resulta como sigue:

$$\sigma_{I_{x_i}^2}^2 = 4I_{x_i}^2 |p|^2 |d_1|^2 \sigma_I^2 \quad (\text{B.7})$$

De este modo, el error en el denominador viene dado por:

$$\sigma_B^2 = 4|p|^2 |d_1|^2 \sigma_I^2 \eta^2 \quad (\text{B.8})$$

Conocidos los errores en el numerador (B.6) y del denominador (B.8) se propagan al cociente. Aumiendo que el numerador y el denominador no están correlados, el error resultante para el cociente es el siguiente:

$$\sigma_{\frac{A}{B}}^2 = 4|p|^2 |d_1|^2 \sigma_I^2 \eta^2 \left(\frac{A}{B}\right)^2 \left(\frac{1}{A^2} + \frac{1}{B^2}\right) \quad (\text{B.9})$$

Sustituyendo los valores ideales de  $A$  y  $B$  en la expresión anterior y simplificando se obtiene:

$$\sigma_{\frac{A}{B}}^2 = |p|^2 |d_1|^2 \frac{\eta^2}{\zeta^4} \frac{4}{\cos^4(2\theta)} \quad (\text{B.10})$$

Propagando el error  $\sigma_{\frac{A}{B}}^2$  en la expresión de la pendiente se obtiene el valor del error de ésta:

$$\sigma_{\hat{s}^2} = \left( \frac{1}{2(x^2 + 1)} \left[ 1 + \tan^2 \left( \frac{\arctan(x)}{2} \right) \right] \right)^2 \sigma_x^2 \quad (\text{B.11})$$

Sustituyendo  $x = \frac{A}{B} = \tan 2\theta$  y  $\sigma_x^2 = \sigma_{\frac{A}{B}}^2$  por su valor y simplificando las expresiones trigonométricas resultantes se obtiene:

$$\sigma_{\hat{s}^2} = |p|^2 |d_1|^2 (1 + \tan^2(\theta))^2 \frac{\eta^2}{\zeta^4} \sigma_I^2 \quad (\text{B.12})$$

donde  $\zeta^2 = \sum_{i,j} \omega(i,j) [I_x(i,j)^2 + I_u(i,j)^2]$ .

Finalmente hay que tener en cuenta que  $\hat{s} = \tan(\theta)$  con lo que:

$$\sigma_{\hat{s}} = |p| |d_1| (1 + \hat{s}^2) \frac{\eta}{\zeta^2} \sigma_I \quad (\text{B.13})$$

En el caso particular en el que todas las componentes del vector de pesos,  $\omega$ , de la operación  $\hat{\cdot}$  sean iguales, estos sumen uno, y  $\varphi$  sea asuma nulo, esta expresión se reduce a:

$$\sigma_{\hat{s}} = |\omega| |p| |d_1| (1 + \hat{s}^2) \frac{\sigma_I}{\zeta} \quad (\text{B.14})$$

donde  $|\omega| = \frac{1}{\sqrt{N}}$ , siendo  $N$  el número de puntos promediados.



## B.2. Cálculo de errores en la estimación de distancia en el light field 4D

La estimación de distancia se realiza por medio de la expresión siguiente expresión:

$$\hat{\theta} = \frac{1}{2} \tan^{-1} \left( \frac{2 \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right)}{\left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) - \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right)} \right) \quad (\text{B.15})$$

Para mayor claridad se denotarán el numerador y el denominador con  $A$  y  $B$ :

$$\begin{aligned} A &= 2 \left( \widehat{L}_x \widehat{L}_u + \widehat{L}_y \widehat{L}_v \right) \\ B &= \left( \widehat{L}_x^2 + \widehat{L}_y^2 \right) - \left( \widehat{L}_u^2 + \widehat{L}_v^2 \right) \end{aligned} \quad (\text{B.16})$$

La derivada parcial del light field respecto de una de sus variables se calcula como:

$$L_{x_i} = \sum_{\mathbf{x}} a_{x_i}(\mathbf{x}) L(\mathbf{x}) \quad (\text{B.17})$$

Suponiendo que la varianza del error todos los puntos de  $L$  es  $\sigma_L$  y que el ruido den las distintas muestras de  $L$  no está correlado, la varianza del error en la derivada parcial viene dada por:

$$\sigma_{L_{x_i}}^2 = \left[ \sum_{\mathbf{x}} a_{x_i}(\mathbf{x})^2 \right] \sigma_L^2 = |p|^6 |d|^2 \sigma_L^2 \quad (\text{B.18})$$

Ahora propagando el error obtenido en la ecuación B.18 en el

cálculo del producto de dos derivadas se obtiene:

$$\sigma_{L_{x_i}L_{x_j}}^2 = |p|^6 |d|^2 \sigma_L^2 \left[ L_{x_i}^2 + L_{x_j}^2 \right] \quad (\text{B.19})$$

A continuación se obtiene el error en el numerador propagando el resultado de la ecuación B.19 en la expresión del numerador dando como resultado:

$$\sigma_A^2 = 4|p|^6 |d|^2 \sigma_L^2 \zeta^2 \quad (\text{B.20})$$

donde  $\eta^2 = \sum_{i,j} \omega(i,j)^2 [L_x^2 + L_y^2 + L_u^2 + L_v^2] + \varphi$ , siendo  $\varphi$  un término dependiente de la correlación entre los términos del sumatorio.

Por otra parte, la propagación del error de la derivada en el cálculo de la derivada al cuadrado resulta como sigue:

$$\sigma_{L_{x_i}^2}^2 = 4L_{x_i}^2 |p|^6 |d|^2 \sigma_L^2 \quad (\text{B.21})$$

De este modo, el error en el denominador viene dado por:

$$\sigma_B^2 = 4|p|^6 |d|^2 \sigma_L^2 \eta^2 \quad (\text{B.22})$$

Conocidos los errores en el numerador (B.20) y del denominador (B.22) se propagan al cociente. Asumiendo que el numerador y el denominador no están correlados, el error resultante para el cociente es el siguiente:

$$\sigma_{\frac{A}{B}}^2 = 4|p|^6 |d|^2 \sigma_L^2 \eta^2 \left( \frac{A}{B} \right)^2 \left( \frac{1}{A^2} + \frac{1}{B^2} \right) \quad (\text{B.23})$$

## B.2 CÁLCULO DE ERRORES EN LA ESTIMACIÓN DE DISTANCIA EN EL LIGHT FIELD 4D

---

Sustituyendo los valores ideales de  $A$  y  $B$  en la expresión anterior y simplificando se obtiene:

$$\sigma_{\frac{A}{B}}^2 = \frac{4|p|^6|d|^2\sigma_L^2\eta^2}{\cos^4(2\theta)\zeta^4} \quad (\text{B.24})$$

Propagando el error  $\sigma_{\frac{A}{B}}^2$  en la expresión de la pendiente se obtiene el valor del error de ésta:

$$\sigma_{\hat{s}^2} = \left( \frac{1}{2(x^2 + 1)} \left[ 1 + \tan^2 \left( \frac{\arctan(x)}{2} \right) \right] \right)^2 \sigma_x^2 \quad (\text{B.25})$$

Sustituyendo  $x = \frac{A}{B} = \tan 2\theta$  y  $\sigma_x^2 = \sigma_{\frac{A}{B}}^2$  por su valor y simplificando las expresiones trigonométricas resultantes se obtiene:

$$\sigma_{\hat{s}^2} = (1 + \tan^2(\theta))^2 |p|^6 |d|^2 \frac{\eta^2}{\zeta^4} \sigma_L^2 \quad (\text{B.26})$$

donde  $\zeta^2 = \sum_{i,j} \omega(i,j) [I_x(i,j)^2 + I_u(i,j)^2]$ .

Finalmente hay que tener en cuenta que  $\hat{s} = \tan(\theta)$  con lo que:

$$\sigma_{\hat{s}} = |p|^3 |d| (1 + \hat{s}^2) \frac{\eta}{\zeta^2} \sigma_L^2 \quad (\text{B.27})$$

En el caso particular en el que todas las componentes del vector de pesos,  $\omega$ , de la operación  $\hat{\cdot}$  sean iguales, estos sumen uno, y  $\varphi$  sea asuma nulo, esta expresión se reduce a:

$$\sigma_{\hat{s}} = |\omega| |p|^3 |d_1| (1 + \hat{s}^2) \frac{\sigma_I}{\zeta} \quad (\text{B.28})$$

donde  $|\omega| = \frac{1}{\sqrt{N}}$ , siendo  $N$  el número de puntos promediados.



# Bibliografía

- Adelson, E. H. y J. R. Bergen (1991). «The Plenoptic Function and the Elements of Early Vision». En: *Computational Models of Visual Processing*. MIT Press, págs. 3-20.
- Adelson, E. H. y J. Y. A. Wang (1992). «Single Lens Stereo with a Plenoptic Camera». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, pág. 9.
- Advanced Microdevices, Inc. (2012a). *AMD FireStream? GPU Compute Accelerators*. URL: <http://www.amd.com/us/products/workstation/graphics/legacy/Pages/firestream.aspx> (visitado 20-01-2012).
- (2012b). *Accelerated Parallel Processing (APP) SDK*. URL: <http://developer.amd.com/tools/heterogeneous-computing/amd-accelerated-parallel-processing-app-sdk/> (visitado 20-01-2012).

- Alliney, S. (1997). «A Property of the Minimum Vectors of a Regularizing Functional Defined by Means of the Absolute Norm». En: *Trans. Sig. Proc.* 45.4, págs. 913-917. ISSN: 1053-587X.
- Atanassov, K., S. Goma, V. Ramachandra y T. Georgiev (2011). «Content-based depth estimation in focused plenoptic camera». En: San Francisco Airport.
- Babacan, S. D., R. Ansorge, M. Luessi, R. Molina y A. K. Katsaggelos (2009). «Compressive sensing of light fields». En: *Proceedings of the 16th IEEE international conference on Image processing. ICIP'09.* Piscataway, NJ, USA: IEEE Press, págs. 2313-2316. ISBN: 978-1-4244-5653-6.
- Babacan, S., R. Ansorge, M. Luessi, P. Mataran, R. Molina y A. Katsaggelos (2012). «Compressive Light Field Sensing». En: *Image Processing, IEEE Transactions on* 21.12, págs. 4746 -4757. ISSN: 1057-7149.
- Baker, S., T. Sim y T. Kanade (2003). «When Is the Shape of a Scene Unique Given Its Light-Field: A Fundamental Theorem of 3D Vision?» En: *IEEE Trans. Pattern Anal. Mach. Intell.* 25.1, págs. 100-109.
- Berent, J. y P. Dragotti (2007). «Plenoptic Manifolds». En: *Signal Processing Magazine, IEEE* 24.6, págs. 34 -44. ISSN: 1053-5888.
- Bigün, J. y G. H. Granlund (1987). «Optimal Orientation Detection of Linear Symmetry». En: *Proceedings of the IEEE First International Conference on Computer Vision.* London, Great Britain, págs. 433-438.

## BIBLIOGRAFÍA

---

- Bishop, T. E. y P. Favaro (2011). «Full-resolution depth map estimation from an aliased plenoptic light field». En: *Proceedings of the 10th Asian conference on Computer vision - Volume Part II. ACCV'10*. Queenstown, New Zealand: Springer-Verlag, págs. 186-200. ISBN: 978-3-642-19308-8.
- Bishop, T., S. Zanetti y P. Favaro (2009). «Light Field Superresolution». En: *Proc. ICCP '09. IEEE International Conference on Computational Photography*.
- Blender Foundation (2013). *Blender*. URL: <http://www.blender.org>.
- Bolles, R. C., H. H. Baker y D. H. Marimont (2008). «Epipolar-plane image analysis: An approach to determining structure from motion.» En: *International Journal of Computer Vision* 1.1, págs. 7-55.
- Boykov, Y., O. Veksler y R. Zabih (2001). «Fast Approximate Energy Minimization via Graph Cuts». En: *IEEE Trans. Pattern Anal. Mach. Intell.* 23.11, págs. 1222-1239. ISSN: 0162-8828.
- Bredies, K., K. Kunisch y T. Pock (2010). «Total Generalized Variation.» En: *SIAM J. Imaging Sciences* 3.3, págs. 492-526.
- Buck, I., T. Foley, D. Horn, J. Sugerman, K. Fatahalian, M. Houston y P. Hanrahan (2004). «Brook for GPUs: Stream Computing on Graphics Hardware». En: *ACM Transactions on Graphics* 23, págs. 777-786.
- Buehler, C., M. Bosse, L. McMillan, S. Gortler y M. Cohen (2001). «Unstructured lumigraph rendering». En: *Proceedings of the 28th annual conference on Computer graphics and interactive*

- 
- techniques*. SIGGRAPH '01. New York, NY, USA: ACM, págs. 425-432. ISBN: 1-58113-374-X.
- Chan, T., S. Esedoglu, F. Park y A. Yip (2005). «Recent developments in total variation image restoration». En: *In Mathematical Models of Computer Vision*. Springer Verlag.
- Chan, T. F. y S. Esedoglu (2005). «Aspects of total variation regularized L 1 function approximation». En: *SIAM J. Appl. Math.*
- Dansereau, D. (2003). «4D Light Field Processing and its Application to Computer Vision». Tesis de lic. Calgary, Alberta: University of Calgary.
- Davis, A., M. Levoy y F. Durand (2012). «Unstructured Light Fields». En: *Comp. Graph. Forum* 31.2pt1, págs. 305-314. ISSN: 0167-7055.
- Di Zenzo, S. (1986). «A Note on the Gradient of a Multi-Image». En: *Computer Vision, Graphics, and Image Processing* 33, págs. 116-125.
- Donoho, D. L. e I. M. Johnstone (1995). «Adapting to unknown smoothness via wavelet shrinkage». En: *Journal of the American Statistical Association*, págs. 1200-1224.
- Drazic, V., J.-J. Sacré, A. Schubert, J. Bertrand y E. Blondé (2012). «Optimal design and critical analysis of a high-resolution video plenoptic demonstrator». En: *Journal of Electronic Imaging* 21.1, págs. 011007-1-011007-10.
- Farid, H. y E. P. Simoncelli (2004). «Differentiation of Discrete Multidimensional Signals». En: *IEEE Transactions on Image Processing* 13, págs. 496-508.



## BIBLIOGRAFÍA

---

- Favaro, P. y S. Soatto (2006). *3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur*. Secaucus, NJ, USA: Springer-Verlag New York, Inc. ISBN: 1846281768.
- Felzenszwalb, P. F. y D. P. Huttenlocher (2006). «Efficient Belief Propagation for Early Vision». En: *International Journal of Computer Vision* 70.1, págs. 41-54.
- Fife, K., A. E. Gamal y H. s. Philip Wong (2006). «A 3d multi-aperture image sensor architecture». En: *In Custom Integrated Circuits Conference, 2006. CICC '06. IEEE*, págs. 281-284.
- Georgiev, T. y A. Lumsdaine (2010). «Reducing Plenoptic Camera Artifacts». En: *COMPUTER GRAPHICS forum* 00.0, págs. 1-14.
- Georgiev, T. y C. Intwala (2006). *Light Field Camera Design for Integral View Photography*. Inf. téc. Adobe Systems, Inc.
- Georgiev, T. y A. Lumsdaine (2012). «The Multi-Focus Plenoptic Camera». En: *IS&T/SPIE Electronic Imaging*.
- Georgiev, T., C. Intwala y D. Babacan (2006a). *Light-Field Capture by Multiplexing in the Frequency Domain*. Inf. téc. Adobe Systems, Inc.
- Georgiev, T., K. C. Zheng, B. Curless, D. Salesin, S. Nayar y C. Intwala (2006b). «Spatio-Angular Resolution Tradeoff in Integral Photography». En: *In Eurographics Symposium on Rendering*, págs. 263-272.
- Gershun, A. (1936). «The light field». Trad. por P. Moon y G. Timoshenko. En: *J. Math. and Physics* 18, págs. 51-151.
- Getreuer, P. (2012). «Total Variation Deconvolution using Split Bregman». En: *Image Processing On Line* 2012.

- Goldlücke, B. (2013). *HCI Light Field Archive*. URL: [http://hci.iwr.uni-heidelberg.de/HCI/Research/LightField/lf\\_archive.php](http://hci.iwr.uni-heidelberg.de/HCI/Research/LightField/lf_archive.php) (visitado 30-06-2013).
- Goldstein, T. y S. Osher (2009). «The Split Bregman Method for L1-Regularized Problems». En: *SIAM Journal on Imaging Sciences* 2.2, págs. 323-343. eprint: <http://epubs.siam.org/doi/pdf/10.1137/080725891>.
- Gortler, S. J., R. Grzeszczuk, R. Szeliski y M. F. Cohen (1996). «The lumigraph». En: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. SIGGRAPH '96. New York, NY, USA: ACM, págs. 43-54. ISBN: 0-89791-746-4.
- Gu, X., S. J. Gortler y M. F. Cohen (1997). «Polyhedral Geometry and the Two-Plane Parameterization». En: *In Rendering Techniques 97 (Proceedings of Eurographics Rendering Workshop*. Eurographics, págs. 1-12.
- Hartley, R. I. y A. Zisserman (2004). *Multiple View Geometry in Computer Vision*. Second. Cambridge University Press, ISBN: 0521540518.
- Hasinoff, S. W. y K. N. Kutulakos (2009). «Confocal Stereo». En: *International Journal of Computer Vision* 81.1, págs. 82-104.
- Horn, B. K. y B. G. Schunck (1980). *Determining Optical Flow*. Inf. téc. Cambridge, MA, USA.
- Ihrke, I., T. Stich, H. Gottschlich, M. Magnor y H.-P. Seidel (2008). «Fast Incident Light Field Acquisition and Rendering». En: *Journal of WSCG (WSCG'08)* 16.1-3, to appear.

## BIBLIOGRAFÍA

---

- Isaksen, A., L. McMillan y S. J. Gortler (2000). «Dynamically reparameterized light fields». En: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., págs. 297-306. ISBN: 1-58113-208-5.
- Ives, F. (1903). «Parallax Stereogram and Process of Making Same». Pat. US patent 725567.
- Kamal, M. H., M. Golbabaee y P. Vandergheynst (2012). «Light Field Compressive Sensing in Camera Arrays». Anglais. En: *The 37th International Conference on Acoustics, Speech, and Signal Processing*. Kyoto, Japon.
- Khronos OpenCL Working Group (2012). *The OpenCL Specification, version 1.2*.
- Ku, H. H. (1966). «Notes on the use of propagation of error formulas». En: *Journal of Research of the National Bureau of Standards. Section C: Engineering and Instrumentation* 70C.4, págs. 263-273. ISSN: 0022-4316.
- Kubota, A., K. Takahashi, K. Aizawa y T. Chen (2004). «All-focused light field rendering». En: *Rendering Techniques*, págs. 235-242.
- Kumar, A., C. Vogel, S. Haker, A. Tannenbaum y S. Zucker (1997). «Stereo disparity and L1 minimization». En: *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*. Vol. 2, 1125-1129 vol.2.
- Lambert, J. y E. Anding (1892). *Lamberts Photometrie: (Photometria, sive De mensura et gradibus luminis, colorum et umbrae) (1760)*.

- Ostwalds Klassiker der exakten Wissenschaften v. 1-2. W. Engelmann.
- Lanman, D., M. Wachs, G. T. Fern y O. Cukierman (2006). «Reconstructing a 3d line from a single catadioptric image». En: *In Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission*, págs. 89-96.
- Levoy, M. (2006). «Light Fields and Computational Imaging». En: *Computer* 39.8, págs. 46-55. ISSN: 0018-9162.
- Levoy, M. y P. Hanrahan (1996). «Light field rendering». En: SIGGRAPH '96. New York, NY, USA: ACM, págs. 31-42. ISBN: 0-89791-746-4.
- Levoy, M., R. Ng, A. Adams, M. Footer y M. Horowitz (2006). «Light field microscopy». En: *ACM Trans. Graph.* 25.3, págs. 924-934. ISSN: 0730-0301.
- Liang, C.-K., T.-H. Lin, B.-Y. Wong, C. Liu y H. Chen (2008). «Programmable Aperture Photography: Multiplexed Light Field Acquisition». En: *ACM Transactions on Graphics* 27.3, 55:1-55:10.
- Lindholm, E., M. J. Kilgard y H. Moreton (2001). «A user-programmable vertex engine». En: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. SIGGRAPH '01. New York, NY, USA: ACM, págs. 149-158. ISBN: 1-58113-374-X.
- Lindholm, E., J. Nickolls, S. Oberman y J. Montrym (2008). «NVIDIA Tesla: A Unified Graphics and Computing Architecture». En: *IEEE Micro* 28.2, págs. 39-55. ISSN: 0272-1732.

## BIBLIOGRAFÍA

---

- Lippmann, G. (1908). «Épreuves réversibles donnant la sensation du relief». French. En: *J. Phys. Theor. Appl.* 7.1, págs. 821-825.
- Lüke, J. P., F. P. Nava, J. G. Marichal-Hernández, J. M. Rodríguez-Ramos y F. Rosa (2010). «Near Real-Time Estimation of Super-resolved Depth and All-In-Focus Images from a Plenoptic Camera Using Graphics Processing Units». En: *International Journal of Digital Multimedia Broadcasting*, pág. 12.
- Lüke, J., F. Rosa, J. Sanluis, J. Marichal-Hernández y J. Rodríguez-Ramos (2013). «Error analysis of depth estimations based on orientation detection in EPI-representations of 4D light fields». En: *Information Optics (WIO), 2013 12th Workshop on*, págs. 1-3.
- Lumsdaine, A. y T. Georgiev (2008). *Full resolution lightfield rendering*. Inf. téc. Adobe Systems, Inc.
- (2009). «The Focused Plenoptic Camera». En: *In Proc. IEEE ICCP*, págs. 1-8.
- Lytro (2012). *Lytro, Inc.* URL: <http://www.lytro.com> (visitado 26-12-2012).
- MERL (2007). *UCSD/MERL Light field repository*. URL: <http://vision.ucsd.edu/datasets/lfarchive/lfs.shtml> (visitado 30-06-2013).
- Ma, Y., S. Soatto, J. Kosecka y S. S. Sastry (2003). *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag. ISBN: 0387008934.
- Marichal-Hernández, J., J. Lüke, F. Rosa, F. Pérez Nava y J. Rodríguez-Ramos (2009). «Fast approximate focal stack

- transform». En: *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, págs. 1 -4.
- Marichal-Hernández, J. G., J. P. Lüke, F. L. Rosa y J. M. Rodríguez-Ramos (2011). «Fast approximate 4D:3D discrete Radon transform, from light field to focal stack with  $O(N^4)$  sums». En: *Proc. SPIE*, 78710G-78710G-15.
- Marichal-Hernández, J. G. (2012). «Obtención de información tridimensional de una escena a partir de sensores plenópticos usando procesamiento de señales con hardware gráfico». Tesis doctoral. Universidad de La Laguna.
- Marwah, K., G. Wetzstein, A. Veeraraghavan y R. Raskar (2012). «Compressive light field photography». En: *SIGGRAPH Talks*, pág. 42.
- McCool, M., S. Du Toit, T. Popa, B. Chan y K. Moule (2004). «Shader algebra». En: *ACM SIGGRAPH 2004 Papers*. SIGGRAPH '04. Los Angeles, California: ACM, págs. 787-795.
- Miled, W. y J.-C. Pesquet (2006). «Disparity Map Estimation Using A Total Variation Bound.» En: *CRV*. IEEE Computer Society, pág. 48. ISBN: 0-7695-2542-3.
- Montilla, I., M. Reyes, B. Femenía y J. M. Rodríguez-Ramos (2010). «Multiconjugate adaptive optics with plenoptic cameras and the Fourier transform reconstructor». En: págs. 773641-773641-8.
- Mumford, D. y J. Shah (1989). «Optimal approximations by piecewise smooth functions and associated variational problems». En: *Communications on Pure and Applied Mathematics* 42.5, págs. 577-685. ISSN: 1097-0312.

## BIBLIOGRAFÍA

---

- NVIDIA Corporation (2012). *NVIDIA CUDA Programming Guide*. NVIDIA Corporation.
- Ng, M. y N. Bose (2003). «Mathematical analysis of super-resolution methodology». En: *Signal Processing Magazine, IEEE* 20.3, págs. 62-74. ISSN: 1053-5888.
- Ng, R. (2006). «Digital Light Field Photography». Tesis doctoral. CA, USA: Stanford University.
- Ng, R., M. Levoy, M. Bredif, G. Duval, M. Horowitz y P. Hanrahan (2005). *Stanford Tech Report CTSR 2005-02 Light Field Photography with a Hand-held Plenoptic Camera*. Inf. téc.
- Nomura, Y., L. Zhang y S. Nayar (2007). «Scene Collages and Flexible Camera Arrays». En: *Proceedings of Eurographics Symposium on Rendering*.
- Olano, M. y A. Lastra (1998). «A shading language on graphics hardware: the pixelflow shading system». En: *Proceedings of the 25th annual conference on Computer graphics and interactive techniques. SIGGRAPH '98*. New York, NY, USA: ACM, págs. 159-168. ISBN: 0-89791-999-8.
- Owens, J., M. Houston, S. Luebke D. Green, J. Stone y J. Phillips (2008). «GPU Computing». En: *Proceedings of the IEEE* 96.5, págs. 879 -899. ISSN: 0018-9219.
- Pérez Nava, F. (2010). «Super-Resolution in Plenoptic Cameras by the Integration of Depth from Focus and Stereo». En: *Computer Communications and Networks (ICCCN), 2010 Proceedings of 19th International Conference on*, págs. 1 -6.

- Pérez Nava, F. y J. Lüke (2009a). «An  $O(n^2 \log(n))$  per plane fast discrete focal stack transform». En: *Proceedings of the Optical 3D Measurement Techniques*.
- (2009b). «Simultaneous estimation of super-resolved depth and all-in-focus images from a plenoptic camera». En: *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, 2009*, págs. 1 -4.
- Pérez Nava, F., J. Marichal-Hernández y J. Rodríguez-Ramos (2008). «The Discrete Focal Stack Transform». En: *16th European Signal Processing Conference (EUSIPCO 2008)*. Luasanne (Switzerland).
- Perwass, C. y L. Wietzke (2012). «Single lens 3D-camera with extended depth-of-field». En: págs. 829108-829108-15.
- Pock, T., T. Schoenemann, G. Graber, H. Bischof y D. Cremers (2008). «A Convex Formulation of Continuous Multi-label Problems». En: *Computer Vision ? ECCV 2008*. Ed. por D. Forsyth, P. Torr y A. Zisserman. Vol. 5304. Lecture Notes in Computer Science. Springer Berlin Heidelberg, págs. 792-805. ISBN: 978-3-540-88689-1.
- Raytrix (2012). *Raytrix GmbH*. URL: <http://www.raytrix.de/> (visitado 26-12-2012).
- Red Digital Cinema (2011). *Red ONE Operation Guide*. URL: <https://www.red.com/downloads>.
- Rodríguez-Ramos, J. M., B. Femenía Castelló, F. Pérez Nava y S. Fumero (2008). «Wavefront and distance measurement using the CAFADIS camera». En: 70155Q-70155Q-10.
- Rodríguez-Ramos, J., B. Femenía, I. Montilla, L. Rodríguez-Ramos, J. Marichal-Hernández, J. Lüke, R. López, J. Díaz e Y. Martín (2010).



## BIBLIOGRAFÍA

---

- «The CAFADIS camera: a new tomographic wavefront sensor for Adaptive Optics». En: *1st AO4ELT conference, 05011*.
- Rodríguez Ramos, J. M., F. Rosa González y J. G. Marichal-Hernández (2007). «Wavefront aberration and distance measurement camera». Patent WO/2007/082975.
- Rodríguez-Ramos, J. M., J. G. Marichal-Hernández, F. Rosa González y F. Pérez Nava (2009a). «Method and camera for the real-time acquisition of visual information from three-dimensional scenes». Patent WO/2009/090291.
- Rodríguez-Ramos, L. F., I. Montilla, J. P. Lüke, R. López, J. G. Marichal-Hernández, J. Trujillo-Sevilla, B. Femenía, M. López, J. J. Fernández-Valdivia, M. Puga, F. Rosa y J. M. Rodríguez-Ramos (2012). «Atmospherical wavefront phases using the plenoptic sensor (real data)». En: págs. 83840D-83840D-9.
- Rodríguez-Ramos, L. F., Y. Martín, J. J. Díaz, J. Piqueras y J. M. Rodríguez-Ramos (2009b). «The plenoptic camera as a wavefront sensor for the European Solar Telescope (EST)». En: págs. 74390I-74390I-9.
- Rudin, L. I., S. Osher y E. Fatemi (1992). «Nonlinear total variation based noise removal algorithms». En: *Phys. D* 60.1-4, págs. 259-268. ISSN: 0167-2789.
- Slesareva, N., A. Bruhn y J. Weickert (2005). «Optic Flow Goes Stereo: A Variational Method for Estimating Discontinuity-Preserving Dense Disparity Maps». En: *Pattern Recognition*. Ed. por W. Kropatsch, R. Sablatnig y A. Hanbury. Vol. 3663. Lecture Notes in

- Computer Science. Springer Berlin Heidelberg, págs. 33-40. ISBN: 978-3-540-28703-2.
- Strong, D. M. y T. F. Chan (2000). «Edge-preserving and Scale-dependent Properties of Total Variation Regularization». En: *Inverse Problems*, págs. 165-187.
- Szeliski, R., R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen y C. Rother (2008). «A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.6, págs. 1068-1080. ISSN: 0162-8828.
- Taguchi, Y., A. Agrawal, A. Veeraraghavan, S. Ramalingam y R. Raskar (2010). «Axial-cones: modeling spherical catadioptric cameras for wide-angle light field rendering». En: *ACM SIGGRAPH Asia 2010 papers*. SIGGRAPH ASIA '10. Seoul, South Korea: ACM, 172:1-172:8. ISBN: 978-1-4503-0439-9.
- Takahashi, K., A. Kubota y T. Naemura (2003). «All in-Focus View Synthesis from Under-Sampled Light Fields». En: *ICAT*.
- Tanida, J. (2007). «Artificial Compound-eye Camera and Its Application to Visual Information Processing». En: *Lasers and Electro-Optics, 2007. CLEO 2007. Conference on*, págs. 1 -2.
- Tanida, J., T. Kumagai, K. Yamada, S. Miyatake, K. Ishida, T. Morimoto, N. Kondou, D. Miyazaki e Y. Ichioka (2001). «Thin Observation Module by Bound Optics (TOMBO): Concept and Experimental Verification». En: *Appl. Opt.* 40.11, págs. 1806-1813.

## BIBLIOGRAFÍA

---

- Tarditi, D., S. Puri y J. Oglesby (2006). «Accelerator: using data parallelism to program GPUs for general-purpose uses». En: *ASPLOS*, págs. 325-335.
- The Kronos Group (2013). *OpenGL API Documentation Overview*. URL: <http://www.opengl.org/documentation/> (visitado 30-06-2013).
- Ueda, K., T. Koike, K. Takahashi y T. Naemura (2008). «Adaptive integral photography imaging with variable-focus lens array». En: 68031A-68031A-9.
- Unger, J., A. Wenger, T. Hawkins, A. Gardner y P. Debevec (2003). «Capturing and rendering with incident light fields». En: *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering*. Leuven, Belgium: Eurographics Association, págs. 141-149. ISBN: 3-905673-03-7.
- Veeraraghavan, A., R. Raskar, A. Agrawal, A. Mohan y J. Tumblin (2007). «Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing». En: *ACM Trans. Graph* 26.
- Wang, Y., W. Yin e Y. Zhang (2007). *A Fast Algorithm for Image Deblurring with Total Variation Regularization*.
- Wanner, S. y B. Goldluecke (2012). «Globally Consistent Depth Labeling of 4D Lightfields». En: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Wanner, S., J. Fehr y B. Jaehne (2011). «Generating EPI Representations of 4D Light Fields with a Single Lens Focused

- Plenoptic Camera». En: *International Symposium on Visual Computing (ISVC, oral presentation)*.
- Wanner, S., S. Meister y B. Goldluecke (2013). «Datasets and Benchmarks for Densely Sampled 4D Light Fields». En: *Vision, Modelling and Visualization (VMV)*.
- Wei, W., Z. Pan, J. Sun y Z. Zhao (2010). «Split Bregman Algorithm of TV-L1 Model for Image Diffusion». En: *Pattern Recognition (CCPR), 2010 Chinese Conference on*, págs. 1-5.
- Weijer, J. van de (2005). «Color Features and Local Structure in Images». Tesis doctoral. University of Amsterdam.
- Wetzstein, G., I. Ihrke, D. Lanman y W. Heidrich (2011). «Computational Plenoptic Imaging». En: *Computer Graphics Forum (to appear)*, págs. 1-25.
- Wetzstein, G. (2013). *MIT Media Lab*. URL: <http://web.media.mit.edu/~gordonw/SyntheticLightFields/> (visitado 30-06-2013).
- Wetzstein, G., I. Ihrke y W. Heidrich (2013). «On Plenoptic Multiplexing and Reconstruction». En: *International Journal of Computer Vision* 101 (2).
- Wilburn, B., M. Smulski, K. Lee y M. A. Horowitz (2002). «The light field video camera». En: *in Media Processors 2002*, págs. 29-36.
- Wilburn, B., N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz y M. Levoy (2005). «High performance imaging using large camera arrays». En: *SIGGRAPH '05*. Los Angeles, California: ACM, págs. 765-776.
- Yamada, K. y H. Takahashi (2008). «Small 3D image capturing system by TOMBO». En: págs. 680512-680512-8.

## BIBLIOGRAFÍA

---

- Yang, J. C., M. Everett, C. Buehler y L. McMillan (2002). «A Real-Time Distributed Light Field Camera». En: págs. 77-86.
- Yedidia, J. S., W. T. Freeman e Y. Weiss (2003). «Exploring Artificial Intelligence in the New Millennium». En: ed. por G. Lakemeyer y B. Nebel. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Cap. Understanding Belief Propagation and Its Generalizations, págs. 239-269. ISBN: 1-55860-811-7.
- Zach, C., T. Pock y H. Bischof (2007). «A duality based approach for realtime TV-L1 optical flow». En: *Proceedings of the 29th DAGM conference on Pattern recognition*. Heidelberg, Germany: Springer-Verlag, págs. 214-223. ISBN: 978-3-540-74933-2.
- Zhang, C. y T. Chen (2006). *Light field sampling*. Synthesis lectures on image, video, and multimedia processing. Morgan & Claypool Publishers. ISBN: 9781598290769.