



Universidad
de La Laguna

Aplicaciones multimedia de las cámaras plenópticas

Multimedia applications of Plenoptic Cameras

Pablo Salido Luis-Ravelo

Departamento de Estadística, Investigación Operativa y Computación

Escuela Técnica Superior de Ingeniería Informática

Trabajo de Fin de Grado

La Laguna, 07 de septiembre de 2014

D. **Fernando Andrés Pérez Nava**, con N.I.F. 42091420-V profesor Titular de Universidad y D. **Alejandro Pérez Nava**, con N.I.F. 43821179-S profesor Asociado de Universidad, adscritos al Departamento de Estadística, Investigación Operativa y Computación de la Universidad de La Laguna

CERTIFICAN

Que la presente memoria titulada:

“Aplicaciones Multimedia de las Cámaras Plenópticas ”

ha sido realizada bajo su dirección por D. Pablo Salido Luis-Ravelo, con N.I.F. 78641169-Y.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 7 de septiembre de 2014.

D. Fernando Pérez Nava

D. Alejandro Pérez Nava

Agradecimientos

A mi familia, por enseñarme que todos los propósitos se pueden conseguir con esfuerzo y constancia.

A mis amigos, por ayudarme a que este camino no pareciera tan duro.

Agradecimiento a todos los profesores de la ETSII, en especial a Fernando Pérez Nava por su gran labor docente durante la realización del presente Trabajo de Fin de Grado.

Resumen

En los últimos años han aparecido dispositivos novedosos e innovadores relacionados con el campo de la fotografía. Estos aparatos son las cámaras plenópticas, que entre sus características principales están el reenfoque o el cambio de perspectiva después de haber capturado la imagen, y los monitores autoestereoscópicos, que permiten observar contenido 3D sin necesidad de utilizar gafas o dispositivos especiales.

En este contexto, se plantea el desarrollo de un programa informático que unifique la captura y visualización de dispositivos 3D. Para ello tendrá como entrada una serie de imágenes capturadas con cámaras estereoscópicas y cámaras plenópticas. Tras la ejecución de esta aplicación se producirá un conjunto de imágenes para su visualización en monitores estereoscópicos y autoestereoscópicos. El programa principal está escrito en el lenguaje de programación Java, y la mayoría de los algoritmos de procesamiento de las imágenes son funciones de MATLAB. Estas funciones han sido adaptadas a Java para poder utilizarlas en la aplicación principal. Paralelamente, se pretende realizar un estudio básico que establezca una comparativa entre los dispositivos y que defina cuando es mejor la utilización de uno u otro.

Palabras clave

Fotografía, cámara plenóptica, cámara estereoscópica, monitor estereoscópico, monitor autoestereoscópico, contenido 3D, Java, MATLAB.

Abstract

We have recently witnessed the arrival of a high amount of novel and innovative devices related to the photography field. These items are plenoptic cameras and autostereoscopic displays. Among plenoptic camera main features, we can highlight the refocus or the possibility of changing the perspective after taking a picture. Also, autostereoscopic displays allow watching 3D content without using glasses or other special items.

In this context, we decided to develop a computer program that unifies the capture and visualization of 3D devices. This application uses a set of images taken by stereoscopic and plenoptic camera and generates a set of images which can be visualized in stereoscopic and autostereoscopic displays. The main program is written in Java programming language and most of the image processing algorithms are MATLAB functions. These functions have been adapted to Java in order to be used in the main application. Simultaneously, it is our aim to realize a basic study which set a comparison between devices and decide which is better using one or the other.

Keywords

Photography, plenoptic camera, stereo camera, stereoscopic display, autostereoscopic display, 3D content, Java, MATLAB.

ÍNDICE GENERAL

Capítulo 1. Introducción	16
1.1. Objetivos	16
1.2. Alcance	16
1.3. Antecedentes	17
Capítulo 2. Dispositivos de captura y visualización 3D	21
2.1. Sensores 3D	21
2.1.1. Cámaras estereoscópicas	21
a. Historia	21
b. Aplicaciones	22
c. Fujifilm 3D	23
2.1.2. Cámaras plenópticas	24
a. Historia	25
b. La función plenóptica	25
c. Funcionamiento	26
d. Aplicaciones	27
e. La cámara Lytro	28
2.2. Dispositivos de visualización 3D	29
2.2.1. Monitores estereoscópicos	29
a. Televisor LG 3D	31
2.2.2. Monitores autoestereoscópicos	32
a. Barrera de paralaje	32
b. Lámina lenticular	33
c. Monitor Samsung con lámina lenticular	34
Capítulo 3. Procesamiento de imágenes	35
3.1. Procesar imagen MPO para visualizar en Televisor 3D	35
3.2. Procesar imagen Lytro para visualizar en Televisor 3D	35

3.2.1.	Extrayendo “white images”	36
3.2.2.	Decodificando la imagen	36
3.2.3.	Estructura del campo de luz decodificado	36
3.2.4.	Creando la imagen 3D	37
3.3.	Procesamiento de la imagen Lytro para su visualización en un monitor autoestereoscópico	37
3.4.	Procesamiento de la imagen MPO para su visualización en un monitor autoestereoscópico	38
3.4.1.	Algoritmo de interpolación de vistas	39
Capítulo 4. Desarrollo		43
4.1.	Plan de trabajo	43
4.2.	Tecnologías empleadas	44
4.3.	Librerías	45
4.4.	Herramientas	45
4.5.	Ejecución del programa	46
4.6.	Funcionamiento del programa	47
4.6.1.	Procesamiento MPO para visualizar en Televisor 3D	47
4.6.2.	Procesamiento Lytro para visualizar en Televisor 3D	48
4.6.3.	Procesamiento Lytro para visualizar en monitor autoestereoscópico	51
4.6.4.	Procesamiento MPO para visualizar en monitor autoestereoscópico	52
4.7.	Problemas encontrados	56
4.7.1.	Consumo de recursos	56
4.7.2.	Mapa de disparidad	56
4.7.3.	Búsqueda de información	58
Capítulo 5. Aplicabilidad		59
5.1.	Comparando las cámaras Lytro y Fujifilm 3D	59
5.1.1.	Características	59
5.1.2.	Facilidad de toma	61
5.1.3.	Modo de compartir imágenes.	63

5.2. Resumen comparativa Fujifilm 3D – Lytro	63
5.3. Uso del monitor estereoscópico y del monitor autoestereoscópico	65
Capitulo 6. Conclusiones y Trabajo Futuro	66
Capitulo 7. Conclusions and Future	68
Bibliografía	70

ÍNDICE DE FIGURAS

Figura 1. Representación gráfica de la disparidad angular	18
Figura 2. Programa oficial de Lytro para procesar imágenes	19
Figura 3. Dispositivos estereoscópicos. Izquierda: estereoscopio diseñado por Sir Charles Wheatstone. Derecha: cámara 3D diseñada por Sir David Brewster	22
Figura 4. Cámara Fujifilm 3D	23
Figura 5. Ejemplo de fotografía tomada con la cámara Fujifilm 3D	24
Figura 6. Sistema visual artificial con múltiples lentes	25
Figura 7. Representación gráfica del funcionamiento de la cámara plenóptica, en comparación con una cámara convencional	27
Figura 8. Ejemplo del reenfoque que se puede realizar con las cámara plenópticas	27
Figura 9. Estructura interna de la cámara Lytro	29
Figura 10. Representación gráfica del uso de gafas anaglifo	30
Figura 11. Representación gráfica del uso de gafas polarizadas	30
Figura 12. Representación gráfica del uso de gafas de obturadores activos	31
Figura 13. Televisor LG 3D	31
Figura 14. Representación gráfica del funcionamiento de la barrera de paralaje	32
Figura 15. Representación gráfica del funcionamiento de la lámina lenticular	33
Figura 16. Monitor Samsung con lámina lenticular	34
Figura 17. Procesamiento de la imagen de la cámara 3D para visualizar en televisor 3D	35
Figura 18. Procesamiento de la imagen de la cámara Lytro para visualizar en televisor 3D	35
Figura 19. Ejemplo de matriz con todas las imágenes	37
Figura 20. Procesamiento imagen Lytro para visualizar en monitor autoestereoscópico	38
Figura 21. Representación gráfica de las imágenes seleccionadas	38
Figura 22. Procesamiento de imagen de la cámara 3D para visualizar en monitor autoestereoscópico	39
Figura 23. Ejemplo de interpolación de vistas	40
Figura 24. Algoritmo en el que se utilizan los dos mapas de disparidad	41
Figura 25. Representación gráfica de la conversión de funciones MATLAB	46
Figura 26. Imagen "side by side" para mostrar en televisor 3D	48
Figura 27. Imágenes de la cámara Lytro. Izquierda: imagen sin aplicar corrección de color. Derecha: imagen con corrección de color	50

Figura 28. Imágenes seleccionadas para crear la imagen 3D	51
Figura 29. Imagen "side by side" para mostrar en televisor 3D	51
Figura 30. Imagen de ejemplo para mostrar en monitor autoestereoscópico	52
Figura 31. Arriba: imágenes correspondientes a los puntos vista izquierdo y derecho. Abajo: mapas de disparidad de los puntos de vista de referencia	53
Figura 32. Imagen para mostrar en monitor autoestereoscópico	54
Figura 33. Diagrama de flujo del funcionamiento del programa	55
Figura 34. Arriba: imágenes de ejemplo de la cámara Fujifilm 3D. Abajo: mapas de disparidad obtenida al ejecutar el programa principal	57
Figura 35. Imagen autoestereoscópica resultante de la ejecución del programa. Se pueden observar algunos errores	57
Figura 36. Pantalla de ajustes Lytro	59
Figura 37. Comparativa de las pantallas de las cámaras	60
Figura 38. Problema al establecer un obstáculo en el objetivo de la cámara 3D	61
Figura 39. Ejemplo de fotografía 3D con objeto en primer plano	62
Figura 40. Ejemplo de reenfoque con cámara Lytro	62
Figura 41. Aplicación oficial Lytro para smartphones	63

ÍNDICE DE TABLAS

Tabla 1. Distribución temporal de tareas	43
Tabla 2. Descripción de los directorios de entrada y salida del programa.	47
Tabla 3. Resumen de la comparativa de las cámaras Lytro y Fujifilm 3D	64

CAPITULO 1. INTRODUCCIÓN

1.1. Objetivos

La fotografía, el cine y la televisión tridimensionales, proporcionan una forma más natural de percepción visual a los espectadores, añadiendo realismo a la historia que se está desarrollando. Así como a su vez, la fotografía y las películas en color se impusieron a las realizadas en blanco y negro, puesto que la percepción visual humana es capaz de apreciar el color, es razonable suponer que la tecnología 3D se impondrá sobre el formato visual 2D, mayoritario en la actualidad. El objetivo principal de este Trabajo de Fin de Grado consiste en elaborar un sistema de captura y visualización de imágenes 3D a partir de cámaras con dos lentes y cámaras plenópticas en monitores estereoscópicos y autoestereoscópicos. Para conseguir este objetivo principal se tendrá que:

- Estudiar y conocer los distintos dispositivos de captura 3D.
- Estudiar y conocer los distintos dispositivos de visualización 3D.
- Conocer las técnicas de procesamiento 3D.
- Estudiar la aplicabilidad de los dispositivos de captura 3D.
- Estudiar la experiencia visual de los dispositivos de visualización 3D

1.2. Alcance

Se pretende desarrollar un programa implementado en Java, que reciba una serie de imágenes capturadas con la cámara plenóptica Lytro y la cámara estereoscópica Fujifilm 3D. Estas imágenes serán procesadas y se obtendrá un conjunto de imágenes 3D para mostrar en un monitor estereoscópico (televisor LG 3D) y en un monitor autoestereoscópico (monitor Samsung con lámina lenticular).

El programa principal será implementado en la plataforma de desarrollo Eclipse y para la mayoría del procesamiento de las imágenes utilizará una serie de funciones MATLAB, que serán adaptadas al lenguaje Java para poder incorporarlas a la aplicación principal.

Además, con la realización de este proyecto se pretende obtener un estudio básico de la aplicabilidad de los dispositivos de captura 3D, decidiendo cuando es mejor utilizar uno u otro, y de la experiencia visual de trabajar con los dispositivos de presentación de contenido 3D.

1.3. Antecedentes

Aunque parezca que la fotografía y el cine en tres dimensiones es un invento reciente, la idea del cine en 3D no es nueva en absoluto y sus orígenes se sitúan a comienzos del siglo XIX, donde la industria audiovisual era consciente de que llevar la sensación de profundidad e inmersión era un éxito casi seguro. La tecnología 3D aporta realismo a la escena que se desarrolla ante nosotros, puesto que ésta es nuestra forma natural de visión. Este hecho se debe a la evolución natural que provocó que la mayoría de depredadores tengan sus ojos mirando hacia el frente, permitiéndoles percibir la profundidad y ayudándoles a calcular la distancia hacia su presa.

En 1838, se explicó por primera vez la percepción estereoscópica: debido a que cada ojo ve el mundo desde posiciones horizontales ligeramente diferentes, la imagen que se forma en cada ojo es distinta de la otra. De esta manera, los objetos que se sitúan a diferentes distancias de los ojos proyectan imágenes que difieren en su ángulo, lo que conocemos como disparidad angular (Figura 1). El primer sistema de cine en 3D que se patentó fue en 1890 y el siguiente paso llegó diez años después con la cámara de dos lentes. Más tarde llegaría la separación de la imagen basada en dos colores, en la que se introducía el concepto de las gafas con cristales de dos colores que todos asociamos a las tres dimensiones. A pesar de estos desarrollos iniciales y de los esfuerzos por asentar la tecnología 3D, especialmente en la industria cinematográfica de los años 50, no se obtuvo éxito comercial debido a las deficiencias técnicas que producían un efecto 3D de baja calidad. De hecho, las aplicaciones 3D con cierto éxito se encontraban fuera del campo multimedia, como son los simuladores y dispositivos de visualización. En 1986 se lanzó el IMAX 3D, y fue la única excepción en el campo multimedia que

ha tenido cierto éxito comercial y que desde entonces realiza un número limitado de producciones 3D al año. Sin embargo, esta situación comenzó a cambiar de forma gradual a principios de los 90 con la transición de los dispositivos de captura y transmisión analógicos a dispositivos digitales [1].



Figura 1. Representación gráfica de la disparidad angular

Tras los problemas que suponen la utilización de gafas en las tecnologías 3D, aparecen los dispositivos autoestereoscópicos, en los que no se necesita la utilización de las incómodas gafas. En el procedimiento autoestereoscópico, unos filtros ópticos integrados en la pantalla realizan la función de las gafas 3D activas. Esta técnica requiere varias imágenes tomadas desde diferentes ángulos de visión para poder reproducir una única imagen tridimensional.

El uso de cámaras 3D y de dispositivos de visualización de contenidos 3D ha ido aumentando considerablemente en los últimos años. Y con ello, el desarrollo de aplicaciones que se utilizan para procesar estas imágenes. Existe una gran variedad de programas para los distintos sistemas operativos, que se encargan de procesar la imagen y decodificarla en los formatos disponibles de imagen 3D, para

posteriormente visualizarlas en los diferentes tipos de monitores estereoscópicos. Además, debido a la evolución dentro del campo de la fotografía, han ido apareciendo dispositivos novedosos e innovadores, como es el caso de las cámaras plenópticas. Estos dispositivos permiten, gracias a su mecanismo de varias lentes, la posibilidad de reenfocar o cambiar de perspectiva la escena después de haber realizado la fotografía. La principal cámara plenóptica, que se encuentra en el mercado es la desarrollada por la empresa Lytro, aunque varias compañías dedicadas a fabricación de dispositivos móviles se encuentran trabajando en la posibilidad de incorporar esta tecnología en sus aparatos. En cuanto a las imágenes obtenidas con la cámara Lytro, el programa principal que se encarga de procesarlas es el oficial de Lytro (Figura 2). Con este programa, podemos visualizar las imágenes capturadas con la Lytro, reenfocar cualquier elemento que aparezca en la escena, cambiar la perspectiva e incluso obtener la imagen 3D correspondiente para visualizarla en un monitor estereoscópico [2].

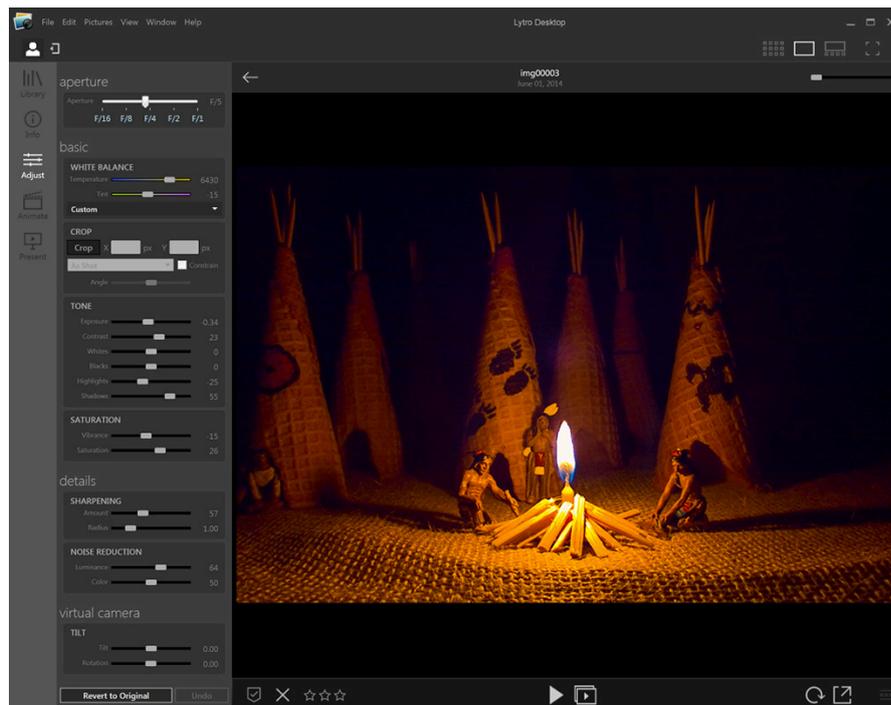


Figura 2. Programa oficial de Lytro para procesar imágenes

La aparición de los dispositivos autoestereoscópicos no se produce hasta hace unos años, por lo que no se conocen programas o aplicaciones comerciales que conviertan las imágenes realizadas con una cámara 3D o con una cámara plenóptica para visualizarlas en un monitor autoestereoscópico.

CAPITULO 2. DISPOSITIVOS DE CAPTURA Y VISUALIZACIÓN 3D

2.1. Sensores 3D

2.1.1. Cámaras estereoscópicas

Las cámaras estereoscópicas están basadas en un principio básico, el de la visión humana. Nuestros ojos están separados unos 65 mm el uno del otro, por lo que cada ojo ve la imagen con distinto ángulo. El cerebro se encarga de unir las dos imágenes obteniendo el efecto de profundidad. Si capturamos dos imágenes con una separación similar al de la visión humana, estaremos imitando el comportamiento de nuestros ojos [3]. Utilizando visores especiales o proyecciones polarizadas, dejando ver a cada ojo sólo la imagen que le corresponde, nuestro cerebro creará el efecto de visión 3D. Los dispositivos de captura 3D más habituales suelen ser un par de cámaras 2D, una cámara 2D junto con una cámara de profundidades o un conjunto de n cámaras 2D. Actualmente, la mayoría de contenido 3D se genera utilizando un par de cámaras o utilizando una cámara 2D dotada de una óptica especial. A diferencia de la utilización de un par de cámaras, el uso de un conjunto de n cámaras tiene la ventaja de ofrecer contenido 3D desde diversos puntos de vista, lo que permite al usuario que pueda escoger el punto de vista para ver la escena. Sin embargo, la utilización de varias cámaras tiene el inconveniente de que todas deben ser consistentes tanto en la respuesta del color como en el sincronismo [1].

a. Historia

Euclides y Leonardo da Vinci observaron y estudiaron el fenómeno de la visión binocular, siendo considerados los pioneros en este tema. El famoso astrónomo Kepler también realizó estudios sobre la estereoscopia [4].

La fotografía estereoscópica surge unos 10 años después de inventarse la fotografía convencional, hacia 1850 es cuando se capturan y se dan a conocer las primeras imágenes tridimensionales. La invención de este dispositivo puede ser atribuida a

Sir Charles Wheatstone quien inventó en 1838 el estereoscopio y posteriormente a Sir David Brewster como el primero constructor de una cámara 3D en 1849.



Figura 3. Dispositivos estereoscópicos. Izquierda: estereoscopio diseñado por Sir Charles Wheatstone. Derecha: cámara 3D diseñada por Sir David Brewster

b. Aplicaciones

Las aplicaciones más conocidas de las cámaras estereoscópicas son la fotografía, el cine y la televisión 3D. Sin embargo, esta tecnología también es utilizada en varios campos científicos:

- *Topografía y estudio del terreno.* Si tomamos dos fotografías aéreas de un área del terreno con cierta distancia entre ellas, obtendremos una imagen estéreo que podremos visualizar con relieve en un estereoscopio especial, permitiendo calcular las elevaciones del terreno.
- *Estudio de planetas.* La NASA ha realizado vistas tridimensionales de fotografías de planetas del sistema solar. También se utiliza para calcular las distancias y tamaños de superficies.
- *Medicina.* Es uno de los campos donde más se usa. Si a un microscopio estéreo se conectan dos cámaras de video, se puede ofrecer una representación 3D. También ofrece grandes posibilidades en el campo de la microcirugía, en la técnica de endoscopia y en la visualización de imágenes o modelos del interior del cuerpo humano.
- *Ingeniería molecular.* La tecnología estereoscópica ofrece gran ayuda en las estaciones de diseño para crear nuevas moléculas complejas.

- *Eventos sociales y culturales.* En la actualidad, muchos de los eventos deportivos, musicales y culturales ya se transmiten utilizando la tecnología 3D [5].

c. Fujifilm 3D

La cámara Fujifilm 3D (Figura 4) es el primer dispositivo diseñado como cámara digital estéreo con el tamaño de las digitales convencionales [6]. Cuenta con dos lentes Fujinon, con una separación de 75 mm entre ellas. Cada sensor tiene resolución de 10 Megapíxeles y el formato de salida de las imágenes es *MPO* (Figura 5). Esta cámara posee un monitor LCD 2D/3D que permite observar las fotografías 3D en el instante en que han sido realizadas. Igualmente permite realizar fotografías convencionales (2D) y tiene modo de captura dual, en el que se realizan dos tomas con diferentes configuraciones. Otras características de la cámara son el flash incorporado, micrófonos estéreos y encaje para trípode.



Figura 4. Cámara Fujifilm 3D

Entre las ventajas con las que cuenta la cámara cabe destacar la portabilidad, la facilidad de uso, el visor LCD 3D, la posibilidad de utilizar el modo avanzado 3D y el formato de salida de las imágenes (MPO) compatible con la mayoría de dispositivos que utilizan la tecnología 3D. Sin embargo, cuenta con una serie de inconvenientes como pueden ser la pérdida de calidad de imagen al realizar zoom, obteniendo pérdida de nitidez y efectos cromáticos, la obtención de imagen granulada cuando se realiza con poca luz, las lentes pueden verse afectadas por el polvo y la humedad, es complicado de ver el visor cuando hay una alta cantidad de brillo solar y el precio elevado [7].



Figura 5. Ejemplo de fotografía tomada con la cámara Fujifilm 3D

2.1.2. Cámaras plenópticas

El diseño de la mayoría de las cámaras comerciales está basado en el ojo de la mayoría de los animales vertebrados. El ojo de los animales vertebrados cuenta con gran sensibilidad y resolución, pero tiene el inconveniente de tener gran peso y volumen y de capturar un campo de visión pequeño. Para los animales invertebrados, este tipo de ojo era muy pesado y con necesidades de procesamiento en el cerebro muy elevadas, por lo que su ojo se desarrolló de forma diferente. En lugar de curvar hacia adentro la superficie fotoreceptora como en nuestra retina, la evolución dio lugar a que estas criaturas tuvieran la superficie curvada hacia fuera generando un ojo compuesto, formado por una serie de ojos elementales que capturan la luz a partir de un determinado ángulo de incidencia y que contribuyen a una porción de la imagen final. La imagen final del entorno se compone de la suma de luz capturada por todos los ojos elementales produciendo un mosaico que representa la visión del mundo. En los últimos años, se han propuesto varios diseños de sistemas de imágenes que utilizan el funcionamiento de este ojo compuesto (Figura 6). Desde el punto de vista biológico, el diseño óptico de la cámara plenóptica puede interpretarse como tener un ojo humano (cámara) y sustituir su retina con un ojo compuesto (una combinación de microlentes y una matriz de fotosensores). Esto permite que la cámara plenóptica obtenga el campo de luz 4D compuesto por la intensidad y la dirección de todos los rayos de luz en una escena. [8].

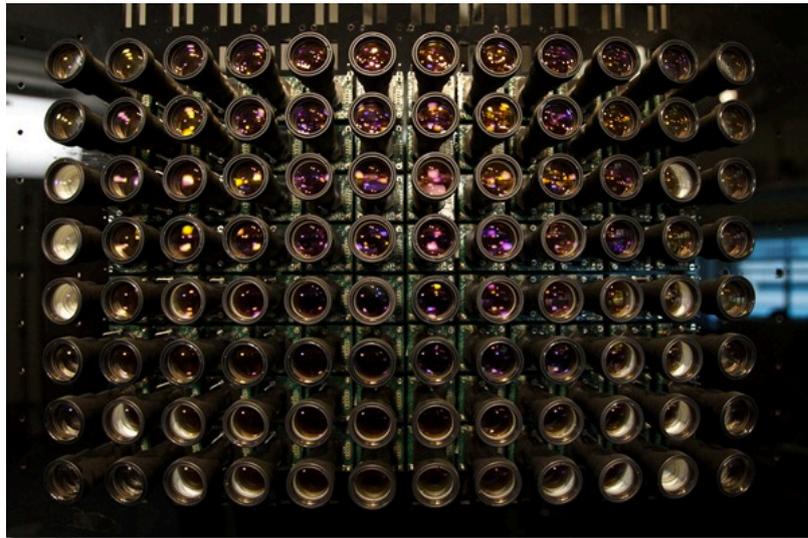


Figura 6. Sistema visual artificial con múltiples lentes

a. Historia

Las ideas fundamentales que están detrás del uso de las cámaras plenópticas aparecieron a principios del siglo pasado con los trabajos de Lippmann y de Ives sobre fotografía [8]. Una de las primeras cámaras plenópticas basada en estos principios fue propuesta en el campo de la Visión por Computador por Adelson y Wang. Otros sistemas similares se construyeron utilizando matrices de microlentes. Georgiev también utilizó una cámara compuesta por una matriz de lentes y prismas colocada en frente de las lentes de una cámara convencional. Otro sistema de imagen es el sensor de Shack-Hartmann que se utiliza para medir las anomalías en los sistemas ópticos. La tecnología plenóptica experimentó su impulso definitivo cuando Ng presentó en 2005 una cámara plenóptica portátil.

b. La función plenóptica

La información visual se puede describir en términos de una función plenóptica dimensional, que actúa como intermediario entre el mundo y el ojo [8]. Esta función describe los rayos de luz presentes en una escena mediante parámetros

geométricos espaciales y direccionales, el tiempo y el espectro de color. Matemáticamente se puede describir como una función 7D:

$$P(x, y, z, \theta, \phi, \lambda, t)$$

donde (x,y,z) son las coordenadas espaciales, (θ, ϕ) denota cualquier dirección dada por las coordenadas esféricas, λ es la longitud de onda y t indica el tiempo. Bajo ciertas asunciones generales [8] La dimensión de la función se puede reducir a un subconjunto de 4 parámetros, haciendo que los requisitos de almacenamiento asociados a la representación de la función sean más manejables:

$$L(x, y, \theta, \phi)$$

c. Funcionamiento

La cámara plenóptica captura el campo de luz 4D mediante la colocación de una matriz de microlentes entre la lente de la cámara y el sensor de imagen para medir la intensidad y la dirección de todos los rayos de luz de cada escena. Su configuración óptica cuenta con una lente fotográfica principal, con una matriz $N_x \times N_x$ de microlentes y una matriz de fotosensores con $N_u \times N_u$ píxeles detrás de cada microlente. Los rayos de luz de un solo punto en el espacio convergen en el plano de la matriz de microlentes (Figura 7). Para maximizar la resolución, las microlentes se enfocan en el plano de la lente principal. Puede parecer que esto requiere un cambio dinámico en la separación entre el plano fotosensor y la matriz de microlentes puesto que la lente principal se mueve durante el enfoque y el zoom. Sin embargo, las microlentes son extremadamente pequeñas en comparación con la lente principal, por lo que independientemente del zoom o de la configuración del enfoque, la lente principal se encuentra fijada en el infinito óptico de las microlentes [8].

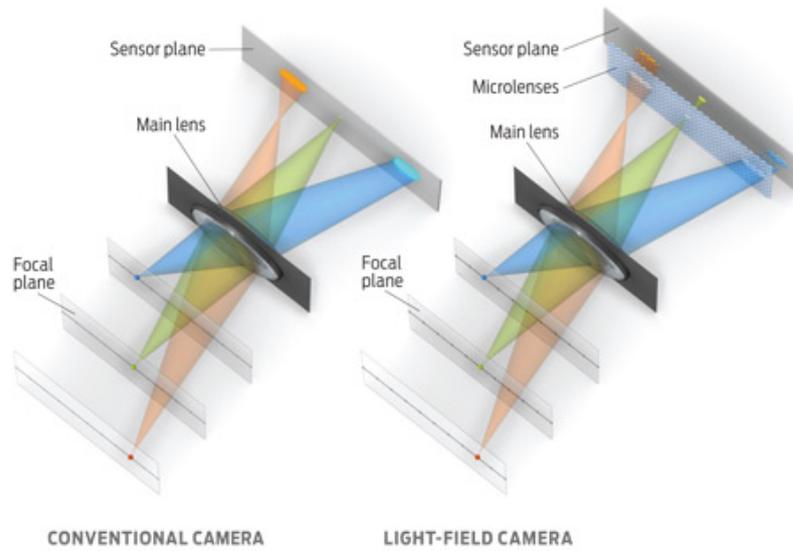


Figura 7. Representación gráfica del funcionamiento de la cámara plenótica, en comparación con una cámara convencional

d. Aplicaciones

- *Reenfoco.* Con la cámara plenótica el campo de luz se muestrea direccionalmente y el dispositivo proporciona al usuario la capacidad de volver a enfocar las imágenes después del momento de la exposición (Figura 8).



Figura 8. Ejemplo del reenfoque que se puede realizar con las cámara plenóticas

- *Cambio de perspectiva.* Es posible ver la escena desde diferentes perspectivas, lo que permite al espectador un mejor conocimiento angular de la escena.
- *Estimación de profundidad.* Las cámaras plenópticas pueden proporcionar información 3D acerca de la escena, indicando la profundidad de cada pixel de la imagen.
- *Super-resolución.* Uno de los principales inconvenientes de las cámaras plenópticas consiste en que la captura de una cierta cantidad de resolución direccional requiere la reducción de la resolución espacial del sensor. Existen una serie de métodos que utilizando las imágenes de estas cámaras, obtienen imágenes con super-resolución.

e. La cámara Lytro

La cámara plenóptica tiene unas microlentes que capturan toda la información de la imagen real desde distintas posiciones y con todos los rayos de luz posibles (Figura 9). A diferencia de las cámaras normales que integran rayos de luz de diversos puntos en el espacio, captura varios rayos de luz en distintas direcciones. Con toda esa información, y con la ayuda de un software de ordenador se pueden convertir todas esas mini-imágenes en una imagen completa, que puede ser enfocada en el punto que deseemos. En la realización de este proyecto utilizaremos la cámara Lytro.

En la actualidad, la cámara Lytro es la cámara plenóptica más vendida. Tiene una pantalla LCD touchscreen de 1.46 pulgadas, lo que permite enfocar la zona deseada de manera táctil. Posee una lente con luminosidad f2.2 y zoom de 8 aumentos. El formato de las imágenes es LFP (Light Field Picture) [9].

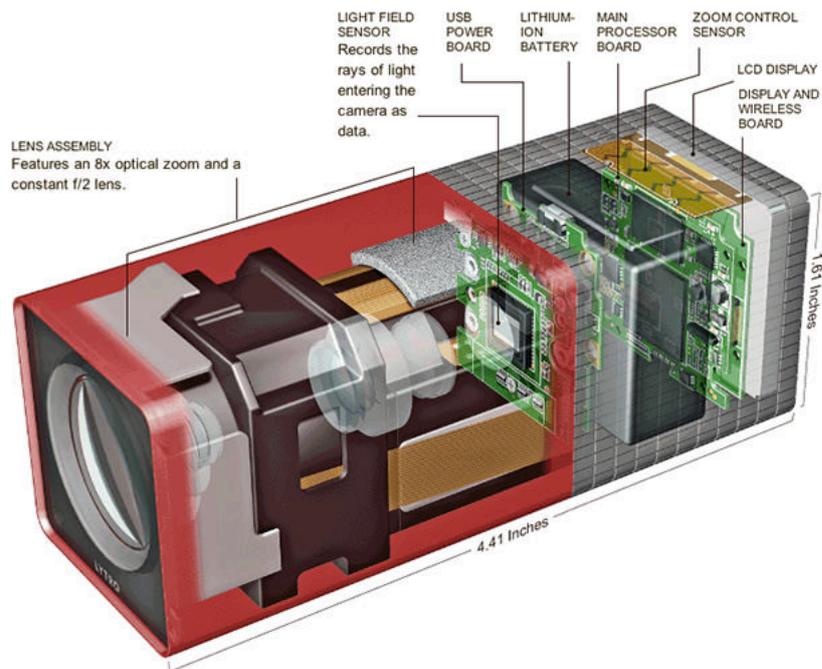


Figura 9. Estructura interna de la cámara Lytro

2.2. Dispositivos de visualización 3D

El último paso para mostrar el contenido 3D al usuario lo constituye el proceso de visualización. El contenido 3D es decodificado y postprocesado para el dispositivo de visualización 3D. Todos los dispositivos utilizan la idea de Wheatstone de enviar imágenes distintas a cada ojo. Las imágenes que han sido capturadas con tecnología 3D se pueden mostrar en dos tipos de dispositivos de visualización: los monitores estereoscópicos y los autoestereoscópicos. Los primeros necesitan unas gafas especiales, mientras que los otros permiten disfrutar de la sensación 3D sin ningún tipo de complemento.

2.2.1. Monitores estereoscópicos

Este tipo de sistemas necesitan el uso de gafas para una correcta visualización. Su funcionamiento se basa en que se emiten dos imágenes diferentes (captadas con una cámara estereoscópica), y cada ojo capta su imagen correspondiente mediante gafas especiales, para así tener una sensación de profundidad [10]. El primer tipo de gafas que se utilizó fueron las **gafas anaglifo** (rojo/azul). Los colores actúan

como filtros que permiten enviar información distinta a cada ojo (Figura 10). El principal inconveniente de este tipo de gafas es la mala reproducción del color y la obtención de un efecto 3D de mediana calidad.



Figura 10. Representación gráfica del uso de gafas anaglifo

Para solucionar estos problemas, aparecieron las **gafas polarizadas** (Figura 11). Su funcionamiento se basa en mostrar sobre la pantalla dos imágenes superpuestas a través de diferentes filtros polarizados ortogonales. Las gafas polarizadas contienen también un par de filtros polarizados con la misma orientación que la pantalla, que bloquean en cada ojo la luz polarizada en la dirección ortogonal. Los principales inconvenientes de estas gafas son la pérdida de luminosidad de la imagen 3D y el precio, superior al de las gafas anaglifo. Estas gafas son la que se utilizan principalmente en los cines, aunque también se utilizan en algunas pantallas de televisión 3D.

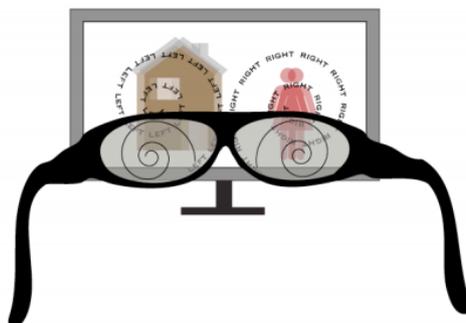


Figura 11. Representación gráfica del uso de gafas polarizadas

La última alternativa la constituyen las **gafas de obturadores activos** (Figura 12). Cada lente contiene un cristal líquido transparente que tiene la propiedad de oscurecerse cuando se le aplica corriente eléctrica. Las gafas son controladas con

un transmisor que envía una señal a las gafas indicando qué ojo oscurecer, dependiendo de que la imagen que se muestra en la pantalla sea la correspondiente al ojo derecho o al izquierdo. Este tipo de gafas es el que se utiliza principalmente en los televisores 3D. Cuenta con los inconveniente de que las gafas son las más caras, necesitan recargar su batería y la imagen 3D sufre pérdida de luminosidad [1].

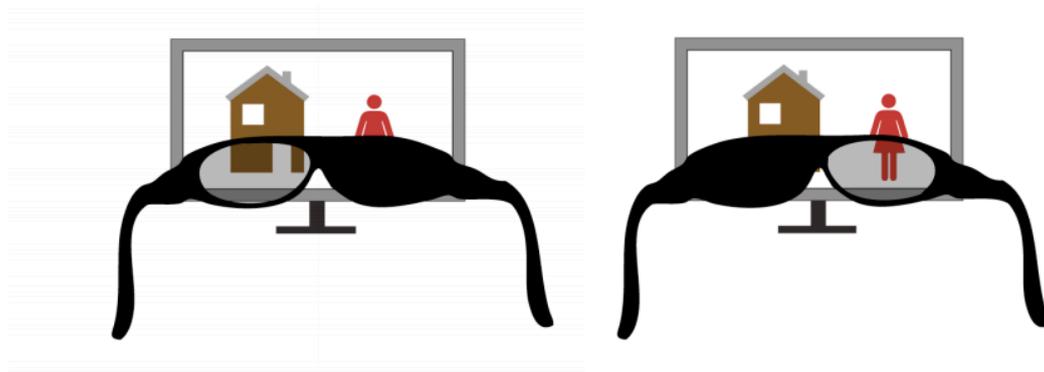


Figura 12. Representación gráfica del uso de gafas de obturadores activos

a. Televisor LG 3D

En este proyecto se ha utilizado el dispositivo estereoscópico LG 27MT93. Se trata de un televisor 3D de 27 pulgadas, con conexión HDMI, posibilidad de entrada de datos por USB y dispone de gafas polarizadas para poder disfrutar de la tecnología 3D. La distancia óptima para visualización de 3D es de 100 cm. Entre los formatos de imagen 3D que admiten se encuentra el MPO y el side-by-side (Figura 13).



Figura 13. Televisor LG 3D

2.2.2. Monitores autoestereoscópicos

Otra alternativa para la visualización 3D la constituyen los dispositivos autoestereoscópicos, en los que no es necesario la utilización de gafas complementarias. En la actualidad, existen dos tipos principales: el de barrera y el lenticular [10].

a. Barrera de paralaje

Es una de las tecnologías más utilizadas para la creación de imágenes estereoscópicas sin necesidad de ningún tipo de dispositivo (gafas) para su visualización. La barrera construirá una imagen diferente para cada ojo mediante un dispositivo colocado delante de nuestra fuente de imagen, generando finalmente una imagen tridimensional. Una barrera de paralaje se basa en un patrón periódico de regiones opacas “no transmisibles” y regiones transparentes “transmisibles” colocadas en frente de un dispositivo de visualización, como podría ser una pantalla de cristal líquido, generando finalmente diferentes imágenes para cada ojo según nuestra visualización de las regiones. Las regiones tienden a ser paralelogramos debido a su fácil distribución en forma de columnas y siguiendo una simetría para que cada ojo visualice únicamente una imagen (Figura 14).

Para la generación de la imagen 3D se muestran dos imágenes 2D en la pantalla LCD. Estas dos imágenes son intercaladas progresivamente a lo largo de la pantalla. Cuando el observador se coloca en la ventana de visualización, el ojo derecho percibirá una imagen 2D, mientras que el ojo izquierdo percibirá otra imagen 2D diferente. En ese instante, nuestro cerebro procesará las imágenes percibiendo el efecto 3D.

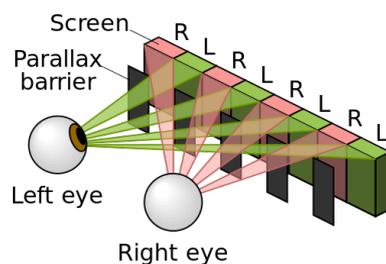


Figura 14. Representación gráfica del funcionamiento de la barrera de paralaje

Entre las desventajas podemos encontrar que el usuario tiene que permanecer en una posición fija para poder ver con claridad la imagen, el ángulo de visión siempre es limitado y la imagen tiende a perder de luminosidad. Podemos agrupar en dos tipos de uso las barreras de paralaje:

- *Sin seguimiento de múltiples vistas.* Caracterizado por los múltiples puntos de vista de los que dispone independientemente de los espectadores. La generación de las vistas se basa en la división de la pantalla para múltiples imágenes intercaladas entre sí. El gran inconveniente es que la resolución de las imágenes es limitada.
- *Seguimiento de dos vistas.* Se utiliza para un único usuario y se basa en la posición del observador. Para solucionar el aspecto del movimiento del usuario se emplean técnicas de detección y seguimiento facial, con el fin de saber qué regiones observará cada ojo. La calidad aumentará considerablemente, pero nuestro sistema estará limitado a aplicaciones monousuario.

b. Lámina lenticular

La alternativa a la barrera de paralaje es el uso de una lámina lenticular, que consiste en una hoja de lentes finas y largas. Esta hoja lenticular contiene una serie de lentes cilíndricas moldeadas en un sustrato plástico. Se diseña la imagen que se muestra tras la lámina lenticular de forma que los rayos de luz de la escena se dirijan a cada ojo. (Figura 15).

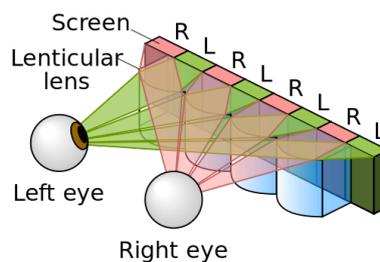


Figura 15. Representación gráfica del funcionamiento de la lámina lenticular

El factor fundamental para la visualización de las imágenes autoestereoscópicas basadas en láminas lenticulares es la calidad y la uniformidad de las lentes. A diferencia del método de barrera, la superficie entera de la lente irradia luz, por lo que no hay zonas opacas.

c. Monitor Samsung con lámina lenticular

Uno de los inconvenientes de los monitores autoestereoscópicos es su elevado precio. Es por ello que en este proyecto se ha adaptado un monitor convencional superponiéndole una lámina lenticular.

El dispositivo utilizado ha sido un monitor LCD *Samsung SyncMaster901N* de 19 pulgadas. A este monitor se le ha añadido una lámina lenticular de 19" y de resolución 1280 x 1024 (Figura 16).



Figura 16. Monitor Samsung con lámina lenticular

CAPITULO 3. PROCESAMIENTO DE IMÁGENES

3.1. Procesar imagen MPO para visualizar en Televisor 3D

Aunque el televisor 3D es capaz de mostrar imágenes en formato MPO, se ha preferido utilizar una imagen “side by side”, ya que en este formato las imágenes ocupan menos espacio sin perder calidad de imagen. Una imagen MPO ocupa en torno a 3.5MB, mientras que una imagen “side by side” con formato JPS, puede llegar a ocupar 2MB, según la cantidad de detalles de la imagen.. Además, las imágenes izquierda y derecha resultantes, también se utilizarán en el proceso de visualizar las imágenes en el monitor autoestereoscópico. El fichero de salida de la cámara contiene dos imágenes, que corresponden a cada uno de los objetivos de la cámara (izquierdo y derecho), por lo que tenemos que extraer estas dos imágenes y crear una nueva en la que estén una al lado de la otra (imagen “side by side”) (Figura 17).

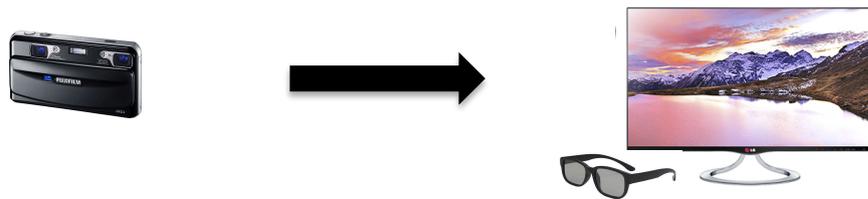


Figura 17. Procesamiento de la imagen de la cámara 3D para visualizar en televisor 3D

3.2. Procesar imagen Lytro para visualizar en Televisor 3D

Este proceso se basa en decodificar imágenes de la cámara Lytro en una estructura de campo de luz 4D. Las entradas en este proceso son una imagen y una “white image” (Figura 18)

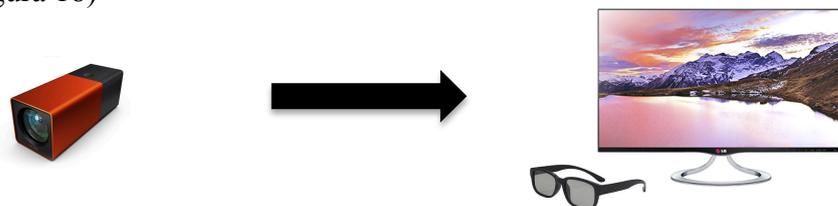


Figura 18. Procesamiento de la imagen de la cámara Lytro para visualizar en televisor 3D

3.2.1. Extrayendo “white images”

Las “white images” son imágenes tomadas a través de un difusor y se utilizan para corregir el viñeteado (oscurecimiento en los bordes de las imágenes) y para construir una red que representa la ubicación de los centros de las microlentes. Las cámaras Lytro vienen precargadas con una serie de “white images” que son copiadas al ordenador la primera vez que conectamos la cámara. Estas imágenes están capturadas con una variedad de ajustes de zoom y enfoque. Cuando se decodifica la imagen, se selecciona la “white image” que más se acerque a los ajustes de zoom y enfoque de la cámara en el momento en que se realizó la fotografía.

Antes de decodificar la imagen, hay que analizar las “white images”. Esto produce una serie de modelos de rejilla, uno por cada “white image”, y una base de datos con las imágenes disponibles.

3.2.2. Decodificando la imagen

Una vez que hemos extraído y analizado las “white images”, debemos extraer los datos necesarios de las imágenes de la cámara Lytro. Para cada imagen, se selecciona una “white image” según el número de serie de la cámara y la configuración de zoom y enfoque. La “white image” y el fichero raw que contiene la imagen son utilizados por la función de decodificación, que construye el campo de luz 4D.

3.2.3. Estructura del campo de luz decodificado

Un campo de luz está formado fundamentalmente por una estructura de 4 dimensiones como se detalló en la sección 2.1.2c. En términos generales, cada píxel corresponde a un rayo de luz, y dos dimensiones definen la posición de ese rayo, mientras que los otros dos definen su dirección. Dos dimensiones seleccionan una imagen, y las otras dos dimensiones seleccionan un píxel dentro de esa imagen. Tras la decodificación, obtenemos una estructura de 11x11 imágenes, y 380x380 píxeles en cada imagen. Si examinamos el fichero de salida de la decodificación,

comprobamos que es un array de 5 dimensiones y de tamaño [11 x 11 x 380 x 380 x 3], donde la ultima dimensión corresponde al canal de color RGB (Figura 19).

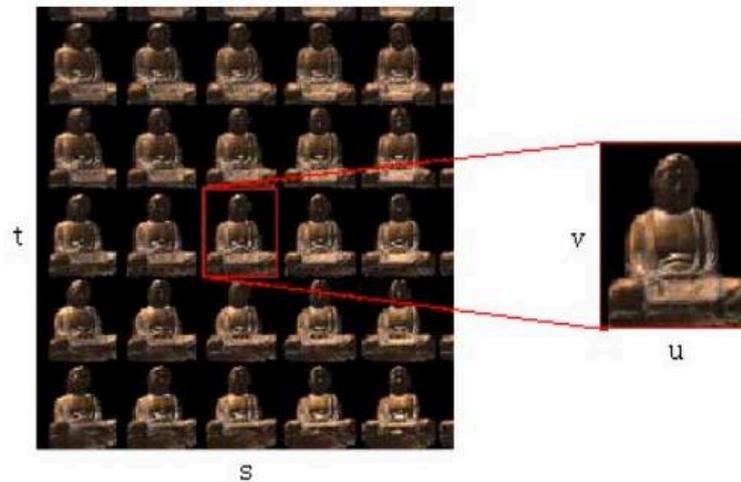


Figura 19. Ejemplo de matriz con todas las imágenes

3.2.4. Creando la imagen 3D

Una vez tenemos el campo de luz decodificado, podemos acceder a las filas y columnas de la matriz y extraer la imagen izquierda y derecha. Para obtener las mejores imágenes, en el eje vertical debemos seleccionar las imágenes de la mitad de la matriz, y en el eje horizontal las que están mas alejadas entre ellas, es decir, la primera y la última. Así se producirá el efecto de profundidad. Por último, solo hay que unir las en una imagen “side by side” para poder visualizarlas en el televisor 3D [11].

3.3. Procesamiento de la imagen Lytro para su visualización en un monitor autoestereoscópico

Para construir la imagen que se mostrará en el monitor autoestereoscópico, debemos seleccionar el mayor número de vistas diferentes de la imagen (Figura 20). A partir de la matriz del campo de luz que obteníamos en el paso anterior, podemos aprovechar las 9 imágenes del centro horizontal, ya que cada una de ellas tiene una perspectiva diferente (Figura 21).

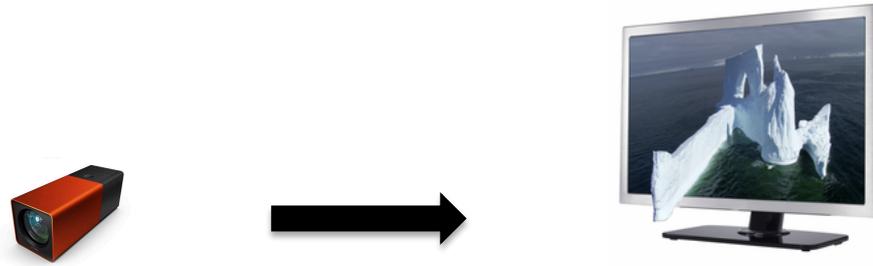


Figura 20. Procesamiento imagen Lytro para visualizar en monitor autoestereoscópico

Creamos una matriz del tamaño de la resolución del monitor autoestereoscópico (en este caso 1280 x 1024), y hacemos un mapeado utilizando las 9 imágenes para darle el color específico a cada pixel de la pantalla y que se observe la sensación de profundidad.

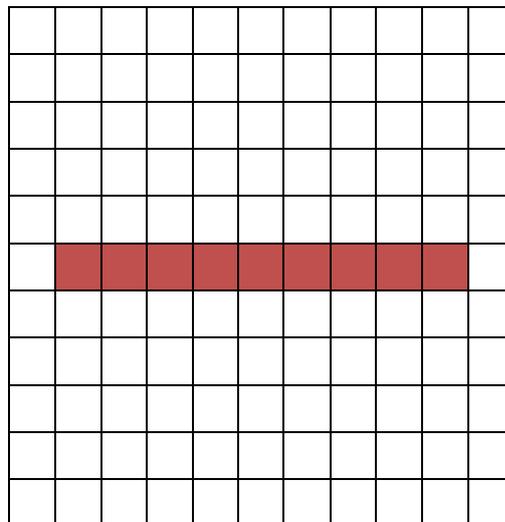


Figura 21. Representación gráfica de las imágenes seleccionadas

3.4. Procesamiento de la imagen MPO para su visualización en un monitor autoestereoscópico

Para poder observar en el monitor autoestereoscópico las imágenes que han sido realizadas con la cámara Fujifilm 3D, necesitamos procesar previamente la imagen (Figura 22). Es el proceso más complejo y el que más recursos consume de los cuatro. Para poder realizar el mapeado que se referencia en el apartado anterior, necesitamos tener el mayor número de vistas posibles de la imagen. Sin embargo, con la cámara estereoscópica sólo obtenemos las dos imágenes correspondiente al

objetivo izquierdo y al derecho, por lo que es necesario aplicar un algoritmo de interpolación de vistas.

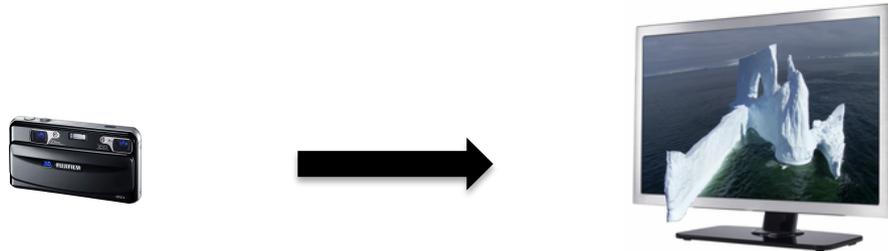


Figura 22. Procesamiento de imagen de la cámara 3D para visualizar en monitor autoestereoscópico

3.4.1. Algoritmo de interpolación de vistas

El proceso de síntesis de nuevas imágenes de una escena con puntos de vista diferentes a los que ya tenemos, se conoce como interpolación de vistas. Existen varias técnicas para conseguir este objetivo, que van desde renderizados basados en modelos 3D hasta renderizados basados en una imagen. Este trabajo se centra en el uso de al menos dos imágenes de una escena que se utilizan como referencia, utilizando algoritmos para recuperar información de profundidad y generar nuevos puntos de vista (Figura 23). Los algoritmos estéreos tienen que lidiar con una gran cantidad de problemas para recuperar con precisión la información 3D de cada píxel. La mayoría de los problemas provienen de la falta de correspondencia entre elementos en las dos imágenes incluyendo la presencia de oclusiones en la imagen, es decir, los píxeles que corresponden a los objetos visible sólo en algunos de los puntos de vista de referencia. En la utilización de dos cámaras, la información de profundidad en las discontinuidades 3D prácticamente no se pueden recuperar, a menos que dispongamos de los mapas de profundidad de antemano. Estos problemas se resuelven parcialmente incorporando información a priori sobre la estructura 3D como su suavidad local.

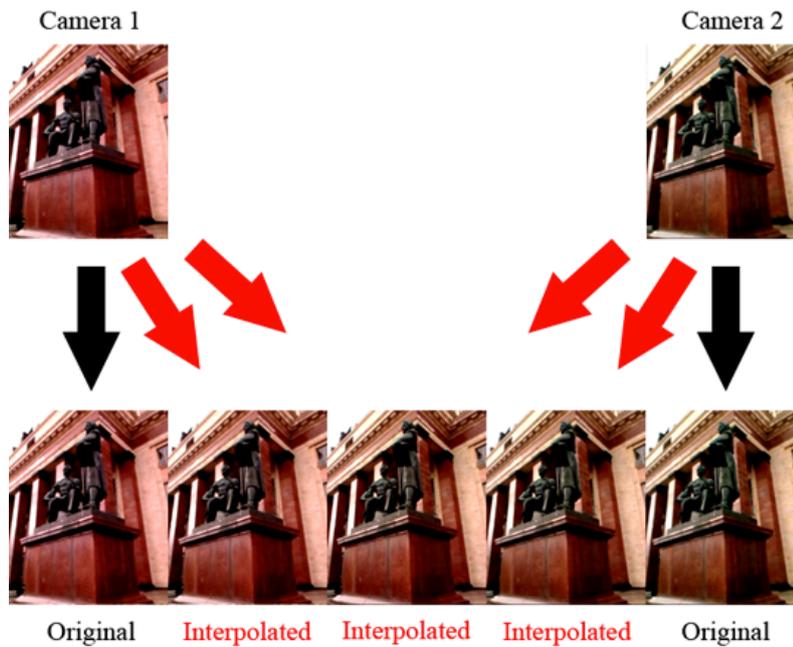


Figura 23. Ejemplo de interpolación de vistas

Para realizar la interpolación de vistas debemos suponer que las dos imágenes de referencia que vamos a utilizar están rectificadas y perfectamente alineadas. A partir de dos imágenes (I_L , I_R), la interpolación de imágenes tiene como objetivo generar imágenes que corresponden a los puntos de vista que se extienden sobre la línea que une los dos puntos de vista de referencia. D_L y D_R se referirán a los mapas de disparidad para las vistas izquierda y derecha, y D_M hará referencia al mapa de disparidad de la vista interpolada. α corresponde al coeficiente que representa la distancia que hay entre la cámara virtual y las cámaras izquierda y derecha. Si $\alpha = 0$, entonces la cámara virtual estaría situada en el mismo lugar que la izquierda, si $\alpha = 1$, estaría en la posición de la derecha [12].

- **Forward Mapping:** cuando tenemos los mapas de disparidad izquierdo D_L y derecho D_R , se propone la utilización de cada uno de ellos para escoger disparidades en un tercer mapa D_M . El algoritmo utiliza sólo el mapa de disparidad izquierdo usando la siguiente fórmula para cada píxel de la imagen interpolada:

$$I_M(x - \alpha D_L(x)) = I_L(x)$$

Este proceso produce mapas de disparidad donde la disparidad se puede

escoger en una posición que se encuentra entre dos píxeles. Si finalmente utilizamos los dos mapas de disparidad, entonces debemos recurrir al siguiente algoritmo (Figura 24):

```

1: for all  $x$  do
2:    $vis(x) \leftarrow visibility(D_L, D_R)$ 
3:   if  $vis = left$  then
4:      $I_M(x - \alpha D_L(x)) = I_L(x)$ 
5:   else if  $vis = right$  then
6:      $I_M(x + (1 - \alpha)D_R(x)) = I_R(x)$ 
7:   else
8:      $I_M(x + (1 - \alpha)D_R(x)) = \alpha I_L(x) + (1 - \alpha) I_R(x)$ 
9:   end if
10: end for

```

Figura 24. Algoritmo en el que se utilizan los dos mapas de disparidad

La función *visibility* se utiliza para elegir que cámara se puede escoger para elegir el color adecuado.

La principal ventaja de este método, es que una vez que hemos calculado los dos mapas de disparidad correspondientes, las nuevas vistas se podrán generar fácilmente [12].

- **Backward mapping:** este método proyecta disparidades que se obtienen de los puntos de vista de la cámara virtual localizada entre las dos cámaras reales. La siguiente fórmula indica cómo podríamos hacerlo:

$$I_M(x) = \alpha I_L(x + D_M(x)) + (1 - \alpha) I_R(x - D_M(x))$$

Debemos observar que cada píxel de la vista virtual $I_M(x)$, está relacionado con las cámaras izquierda y derecha, y el valor de intensidad final es una mezcla lineal de las intensidades de las imágenes. Puede parecer que se obtengan malos resultados, porque cuando se ocultan los objetos, se espera que los píxeles que los representan van a ser diferentes en ambas imágenes, ya que no son completamente visibles. Sin embargo, de forma práctica esto no sucede. La disparidad obtenida para cada píxel de un objeto oculto, hace que este píxel sea visible y se le asigna un color coherente entre las imágenes utilizadas. Teniendo en cuenta la coherencia de la textura, este

color estará cerca del real. El principal problema es encontrar el mapa de disparidad, y el algoritmo utiliza la siguiente ecuación para calcular una disparidad en cada píxel de la vista virtual:

$$D_M(x) = \arg \min_{d \in R} f(I_L(x - \alpha d), I_R(x + (1 - \alpha)d))$$

donde R es el rango de disparidades válidas y f es la función de coincidencia entre dos valores de intensidad. La principal desventaja de este método es que requiere el cálculo de un mapa de disparidad completo para cada vista que deseemos interpolar, haciéndolo más complejo de forma computacional. Sin embargo, como la búsqueda de la mejor disparidad se hace para cada píxel de la vista virtual, se produce un mapa de disparidad denso, sin huecos que llenar [12].

CAPITULO 4. DESARROLLO

4.1. Plan de trabajo

Para la realización del proyecto se ha seguido un plan de trabajo (Tabla 1), en el que se distribuyen durante 16 semanas todas las tareas del Trabajo de Fin de Grado.

Distribución temporal de tareas	
Semana 1-3	Estudio general de los métodos de captura y representación 3D, estudio de los dispositivos de captura y de visualización.
Semana 4-6	Diseño del sistema: Lectura de ficheros de los dispositivos de captura. Procesamiento 3D. Lectura de ficheros de salida para mostrar en los dispositivos de visualización.
Semana 7-8	Implementación del procesamiento de los ficheros de las cámaras Fuji 3D y Lytro, para visualización en monitor estereoscópico LG.
Semana 9	Implementación del procesamiento de los ficheros de la cámara Lytro para visualización en monitor autoestereoscópico Samsung.
Semana 10-13	Implementación del procesamiento de los ficheros de la cámara Fuji 3D para visualización en monitor autoestereoscópico Samsung.
Semana 14	Integración de las tareas realizadas anteriormente.
Semana 15-16	Evaluación de la experiencia visual en los dispositivos de visualización y de la facilidad de captura de las cámaras. Elaboración de los documentos de la Memoria del TFG y de la presentación de la Defensa oral.

Tabla 1. Distribución temporal de tareas

4.2. Tecnologías empleadas

El programa principal de este proyecto ha sido realizado utilizando el lenguaje de programación *Java*. Este lenguaje de programación destaca por ser orientado a objetos, concurrente y basado en clases, además fue diseñado para tener el mínimo de dependencias de implementación posibles. Se intenta que los desarrolladores de aplicaciones escriban el programa y lo ejecuten en cualquier dispositivo, por lo que el código que es ejecutado en una plataforma no tiene que ser recompilado en otra. Se han utilizado funciones de MATLAB ya existentes para el procesamiento de las imágenes. Estas funciones han sido convertidas a clases de Java para poder incorporarlas al programa principal.

Aunque la mayoría de funciones de procesamiento de imágenes están escritas en MATLAB, se ha optado por una implementación del programa principal basada en Java. Esto se debe, a que el lenguaje de programación Java ofrece una serie de ventajas, entre ellas:

- Es un lenguaje independiente de la plataforma, es decir, cualquier programa creado a través de Java podrá funcionar correctamente en ordenadores de todo tipo y con sistemas operativos distintos. Así, cualquier programador podrá acceder desde otro ordenador y aunque tenga instalado un sistema operativo diferente, podrá realizar cualquier modificación.
- Si queremos ampliar nuestro programa y realizar una interfaz gráfica de usuario (GUI), el uso de Java nos ayudará en muchos aspectos.
- Sus características de memoria liberan a los programadores de una familia entera de errores, ya que se ha prescindido por completo de los punteros y de la liberación explícita de memoria.

Además, se ha utilizado un script desarrollado en Python para la extracción de datos de las imágenes de la cámara Lytro.

4.3. Librerías

- **Monte Media Library:** se trata de una librería para Java que se utiliza en el procesamiento de datos audiovisuales. Puede trabajar con formatos como imágenes, vídeos, audios y metadatos. En este proyecto se ha utilizado para separar las imágenes de la cámara Fujifilm 3D que tienen formato MPO, en dos imágenes en formato JPG y que corresponden al ojo izquierdo y ojo derecho [13].
- **LFP Reader:** librería y script desarrollado en Python para leer y extraer datos de los archivos de la cámara Lytro que tienen el formato LFP [14].
- **Light Field Toolbox:** librería utilizada en Matlab para trabajar en el procesamiento de imágenes realizadas con Lytro. Incluye codificación, corrección de color, visualización de las imágenes, calibración de la cámara y rectificación de la imagen [12].

4.4. Herramientas

- **Eclipse:** potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como aplicaciones Java, programas en C++, sitios web o aplicaciones para smartphones. Se trata de un entorno de desarrollo integrado (IDE) en el que se pueden encontrar todas las herramientas y funciones necesarias para realizar un proyecto. Aunque las funciones de esta plataforma son de carácter general, las características del programa se pueden ampliar y mejorar mediante el uso de plug-ins. Originalmente fue desarrollado por IBM, pero en la actualidad está desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto [15].
- **MATLAB:** herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio. Está disponible para las plataformas Linux, Windows y Mac. Entre sus prestaciones básicas podemos encontrar: la manipulación de matrices, la representaciones de datos y funciones, la implementación de algoritmos, la

creación de interfaces de usuario y la comunicación con programas en otros lenguajes y con otros dispositivos hardware [16].

- **MATLAB Builder JA:** herramienta que pertenece al software matemático MATLAB y que se utiliza para convertir funciones que han sido desarrolladas en el lenguaje de programación de MATLAB en clases de Java. Estas clases pueden ser integradas directamente en nuestro programa desarrollado en Java [17] (Figura 25).

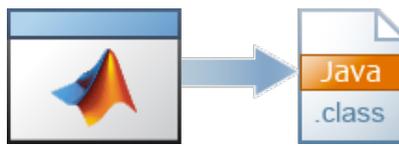


Figura 25. Representación gráfica de la conversión de funciones MATLAB

4.5. Ejecución del programa

La ejecución de este programa se debe de hacer mediante línea de comandos, para ello, el usuario tiene que introducir a través de la consola el siguiente comando:

java ProcesarImagenes

Además, el usuario tiene la opción de escoger si quiere procesar únicamente las imágenes de la cámara Lytro o sólo las de la cámara Fujifilm 3D. Tiene que introducirlo como parámetro a la hora de ejecutar la aplicación. El usuario introducirá **java ProcesarImagenes fujifilm3D** si desea procesar las imágenes de la cámara 3D o **java ProcesarImagenes lytro** si sólo quiere procesar las de la cámara Lytro. Si el usuario no escribe ningún parámetro, el programa procesará todas las imágenes independientemente de la cámara. En caso de que se introduzca un parámetro desconocido se obtendrá el siguiente mensaje de error:

Argumento x desconocido.

Uso: java ProcesarImagenes [lytro | fujifilm3D]

Una vez el programa termine de procesar las imágenes, aparecerá un aviso informando de que la ejecución ha finalizado.

4.6. Funcionamiento del programa

Este programa tiene seis directorios principales reservados para el almacenamiento de los ficheros. Dos de ellos están destinados para los ficheros de entrada y cuatro para los de salida (Tabla 2).

Tipo	Nombre	Descripción
Entrada	MPO/	Contiene las imágenes realizadas con la cámara 3D Fujifilm.
Entrada	LYTRO/	Contiene las imágenes de la cámara Lytro.
Salida	MPOTO3DTV/	Reservado para las imágenes de la cámara 3D Fujifilm, ya procesadas, para visualizarlas en el televisor 3D.
Salida	LYTRO3DTV/	Se almacenarán las imágenes de la cámara Lytro, ya procesadas, para visualizarlas en el televisor 3D.
Salida	LYTRO3DLCD/	Reservado para las imágenes de la cámara Lytro, ya procesadas, para visualizarlas en el monitor LCD.
Salida	TIFF/	Se almacenarán las imágenes de la cámara 3D Fujifilm, ya procesadas, para visualizarlas en el monitor LCD.

Tabla 2. Descripción de los directorios de entrada y salida del programa.

4.6.1. Procesamiento MPO para visualizar en Televisor 3D

La primera función del programa es *mpoToTV3D()*, se encarga de separar la imagen realizada con la cámara Fujifilm 3D, que tiene formato MPO, en dos imágenes JPG y que corresponden a la visión del ojo izquierdo y derecho. Dentro de esta función se llama a otras dos funciones: *splitMPO()* y *sideBySideMPO()*.

- *splitMPO()*: en ella se hace una llamada a la función *splitMPOFile()*, que pertenece a la librería Monte Media Library. Es la función encargada de extraer las dos imágenes del fichero con extensión MPO, y guardarlas en el directorio *MPO/leftright/* con la terminación “_l.JPG” o “_r.JPG” dependiendo de si se trata de la visión izquierda o derecha.

- *sideBySideMPO()*: esta función se encarga de unir, en una imagen “side by side”, las imágenes que hemos obtenido en el paso anterior. La imagen resultante tiene extensión .jps para poder ser visualizado en el televisor 3D y es almacenada en el directorio MPO3DTV/ (Figura 26).



Figura 26. Imagen "side by side" para mostrar en televisor 3D

4.6.2. Procesamiento Lytro para visualizar en Televisor 3D

- **Extrayendo “white images”**: el primer paso para procesar las imágenes de la Lytro es extraer las “white images”. Cuando conectamos la cámara por primera vez al ordenador, estos ficheros se guardan en el directorio:
`\Users\<username>\AppData\Local\Lytro\cameras\sn-<serial_number>`.
 Este directorio contiene ficheros con el nombre data.C.0, data.C.1 y data.C.3. Los ficheros se encuentran en el formato específico de almacenamiento de Lytro, por lo que debemos procesarlo usando el script de Python *LFP Reader*. El comando que debemos utilizar es “*lfp-storage.py export data.C.**”, y se nos guardarán en el directorio `Cameras/numero_de_serie_de_la_cámara/WhiteImages/`. Los ficheros con formato raw resultante corresponden a las “white images” y los ficheros txt contienen metadatos que se necesitan para distinguirlas. En el programa principal, la función encargada de ejecutar este script es *extractWhiteImages()*.
- **Analizando “white images”**: para analizar las “white images” debemos ejecutar la función de MATLAB perteneciente al Light Field Toolbox:

LFUtilProcessWhiteImages. Previamente, debemos convertir esta función para que pueda ser utilizada en Java. Para ello utilizamos la herramienta MATLAB Builder JA, con la que creamos la clase *WhiteImages* que contiene la función *LFUtilProcessWhiteImages*. En el programa principal, llamamos a la función de la siguiente manera: *wImages.LFUtilProcesswhiteImages()*.

- **Extrayendo datos de las imágenes:** el siguiente paso es extraer los datos necesarios que contienen las imágenes con el formato propio de las cámaras Lytro (.lfp). Para ello, debemos utilizar el comando del script LFP Reader “*lfp-picture.py export img_0.lfp*”, y los datos se guardarán en el directorio *Images/*. Por cada imagen, se extraerá: el raw que contiene los datos del campo de luz (*img_0__frame.raw*), la descripción de los ajustes de la cámara cuando se realizó la fotografía, que incluye ajustes de zoom y enfoque e información de corrección de color (*img_0__frame_metadata.json*) y por último el número de serie del dispositivo (*img_0__frame_private_metadata.json*). En el programa principal, la función encargada de ejecutar el script es *exportLFP(int numb)*, donde “numb” es el número identificativo de la imagen.
- **Decodificando la imagen:** para decodificar la imagen hay que ejecutar la función de MATLAB que pertenece al Light Field Toolbox: *LFUtilDecodeLytroFoler*. Además, se le pasará el parámetro ‘ColourCorrect’ para aplicarle a las imágenes una corrección de color. Esta función ha sido convertida para poder utilizarla en el programa Java principal, y la contiene la clase *LFDecode*, por lo que la llamada a esta función se realiza de la siguiente manera: *lfDeco.decodingLF()*.



Figura 27. Imágenes de la cámara Lytro. Izquierda: imagen sin aplicar corrección de color.
Derecha: imagen con corrección de color

- **Creando la imagen 3D:** tras decodificar la imagen, obtendremos el campo de luz decodificado y se almacenará en el directorio Images/ con el nombre de la imagen original y con la terminación “__Decoded.mat”. La función de nuestro programa principal que se encarga de trabajar con este fichero es *createLytro3D()*, en la que se llama a la función *ImageExtract()* de MATLAB que ha sido previamente transformada para poder utilizarla en Java. Esta función se encarga de cargar el fichero que contiene la matriz de imágenes y de escoger la segunda imagen y la penúltima de forma vertical y que se encuentran en la mitad horizontal. No se escogen las de los extremos porque pueden contener errores de viñeteado en la imagen. Los recuadros coloreados de (Figura 28) , corresponden a las imágenes seleccionadas para crear la imagen 3D.

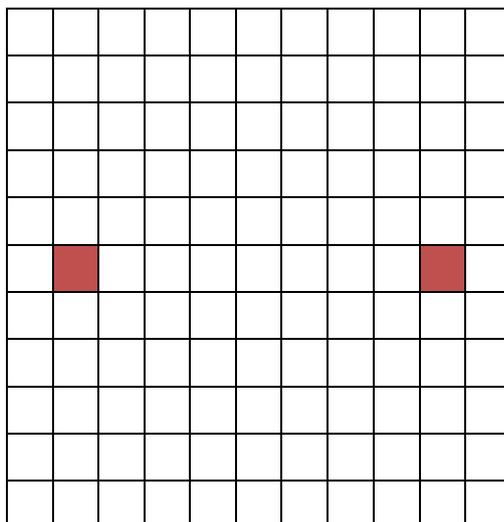


Figura 28. Imágenes seleccionadas para crear la imagen 3D

Al ejecutar esta función, obtendremos en el directorio Images/ dos imágenes con la extensión “_l.jpg” y “_r.jpg”, que pertenecen a la vista izquierda y derecha. En la misma función *createLytro3D()*, se llama a la función *sideBySideLytro()*, que se encarga de unir cada par de imágenes en una sola imagen “side by side”. La imagen resultante tiene extensión .jps para poder ser visualizado en el televisor 3D y es almacenada en el directorio LYTRO3DTV/ (Figura 29).



Figura 29. Imagen "side by side" para mostrar en televisor 3D

4.6.3. Procesamiento Lytro para visualizar en monitor autoestereoscópico

La función de MATLAB *convert3DLCD()*, contiene el algoritmo de mapeo para establecer el color de cada píxel del monitor y que la imagen aporte la sensación de

profundidad a través de la lámina lenticular. Para ello, escoge las 9 imágenes del centro horizontal de la matriz resultante de la decodificación de la imagen, que pudimos observar en el apartado anterior, y las utiliza para realizar el mapeado. Esta función de MATLAB, ha sido convertida para poder utilizarla en el programa principal escrito en Java. La función que realiza la llamada es *lytroToLCD()*, y la ejecuta de la siguiente manera *convertLCD.convert3DLCD()*. Una vez ejecutada esta función, obtenemos las imágenes correspondientes con extensión .tif en el directorio LYTRO3DLCD/ (Figura 30).



Figura 30. Imagen de ejemplo para mostrar en monitor autoestereoscópico

4.6.4. Procesamiento MPO para visualizar en monitor autoestereoscópico

Ya que con la cámara Fujifilm sólo tenemos dos puntos de vista de la escena, necesitamos a través de la interpolación de vistas la creación de los

siete puntos de vista restantes para poder realizar el mapeado de la imagen. Como la ejecución de este proceso es compleja, se ha reducido el tamaño de las imágenes de la cámara 3D en un 7% para que el uso de recursos no fuera tan elevado.

La última función del programa principal es *interpolation()*, en la que se llama a la función de MATLAB *runStereoMatcher()*. Esta función se encarga de procesar tanto la imagen izquierda como la derecha y crear dos imágenes que contienen los mapas de disparidad de ambos puntos de vista. La ejecución de esta función hace que se almacenen en el directorio MPO/results/ los correspondientes mapas de disparidad (Figura 31).

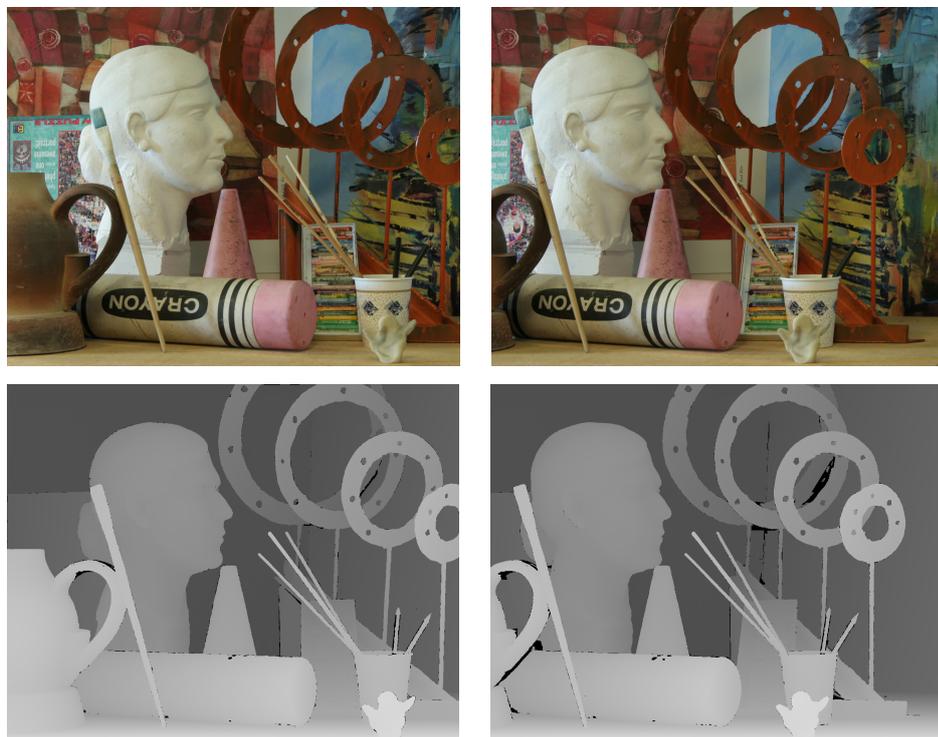


Figura 31. Arriba: imágenes correspondientes a los puntos de vista izquierdo y derecho.
Abajo: mapas de disparidad de los puntos de vista de referencia

A continuación, se llama a la función MATLAB *sample_test()*, que es la encargada de realizar la interpolación de vistas a partir de las dos imágenes (izquierda y derecha) y sus correspondientes mapas de disparidad. Esta función crea un mapa de disparidad final a partir de la nueva vista generada

y los mapas de disparidad de las vistas de referencia. Con este mapa de disparidad final y con la nueva vista generada, se crea la matriz que contiene los 9 puntos de vista de la escena. La ejecución de esta función produce un fichero *synth.mat* en el directorio MPO/results/.

La última función a la que se llama desde *interpolation()* es la función de MATLAB *synth()*. Esta función se encarga de cargar el fichero que contiene la matriz, y realiza el mapeado de forma similar al que observamos en el apartado anterior. Una vez ejecutada esta función, obtenemos las imágenes correspondientes con extensión .tif en el directorio TIFF/. Al mostrar estas imágenes en el monitor autoestereoscópico con lámina lenticular se observa el efecto 3D (Figura 32).

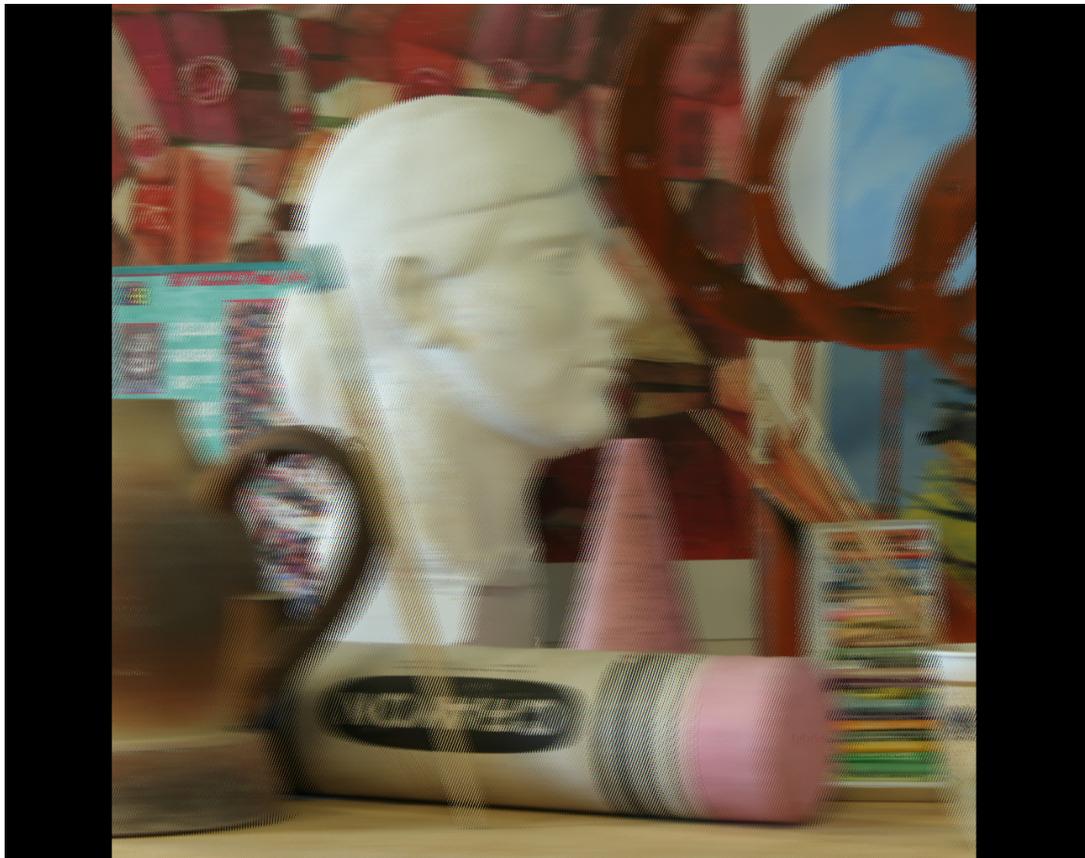


Figura 32. Imagen para mostrar en monitor autoestereoscópico

El funcionamiento del programa, lo podríamos representar mediante un diagrama de flujo (Figura 33).

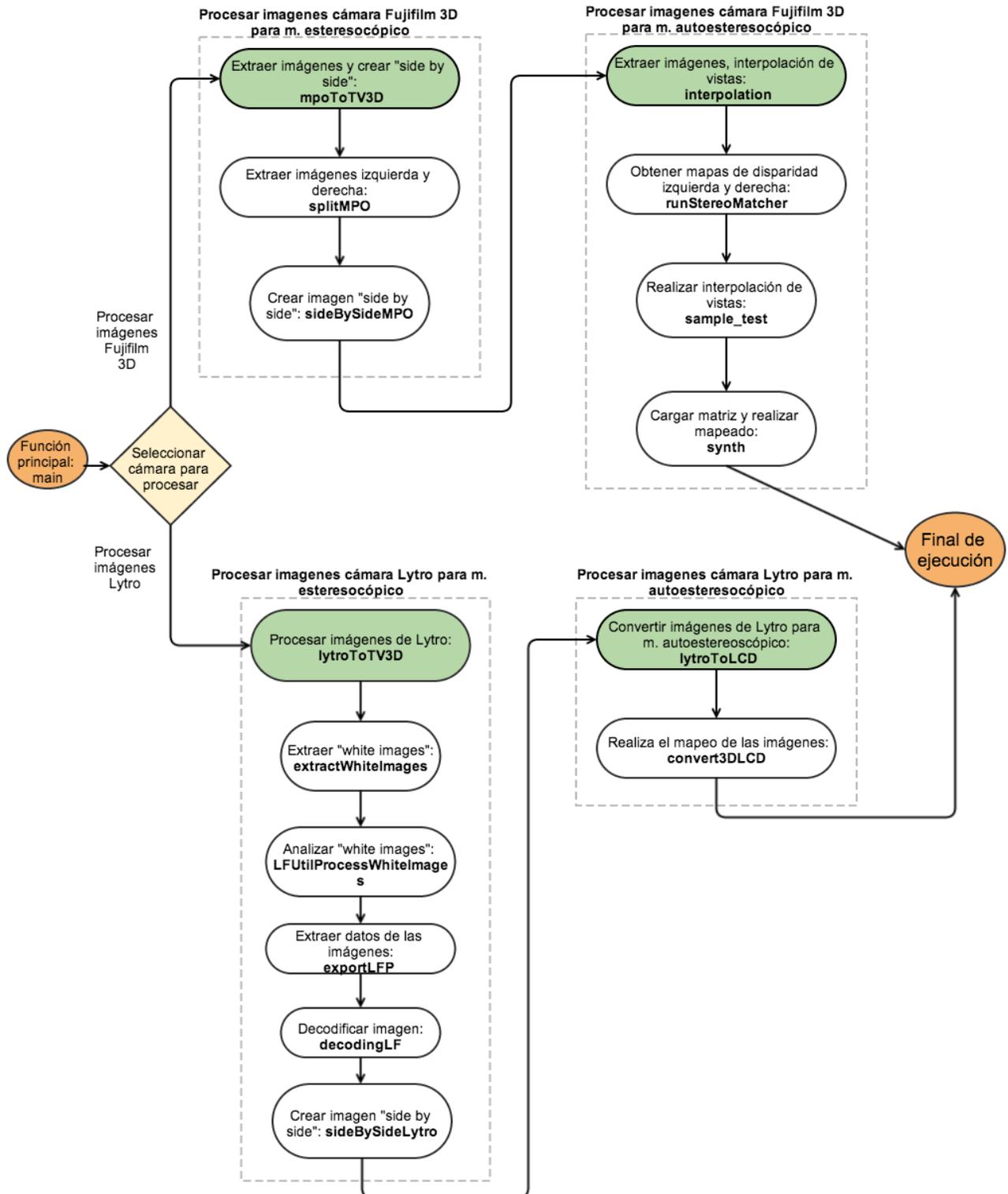


Figura 33. Diagrama de flujo del funcionamiento del programa

4.7. Problemas encontrados

4.7.1. Consumo de recursos

Las funciones utilizadas de MATLAB son bastante complejas y su ejecución produce un elevado consumo de recursos. Por ejemplo, para procesar las imágenes de la cámara Lytro el programa tarda alrededor de 1:05 minutos en trabajar con cada imagen. Mientras que en la obtención del mapa de disparidad de las imágenes de la cámara Fujifilm 3D y en la elaboración de la imagen final autoestereoscópica, el programa tarda aproximadamente 4:20 minutos con cada imagen. Aparte del consumo de recursos computacionales, esto lleva a que se dificulte la tarea de realización de pruebas mientras se está implementando el programa.

Las primeras etapas del programa se estaban ejecutando sobre la plataforma Windows en su versión de 32 bits. Pero las complejas funciones de MATLAB no eran soportadas por esta versión, por lo que se tuvo que trasladar toda la implementación a un ordenador con la versión de 64 bits, lo que produjo un retraso con respecto a la planificación de tareas establecida.

4.7.2. Mapa de disparidad

Existen cámaras estereoscópicas que a parte de realizar las imágenes desde los puntos de vista izquierdo y derecho, también generan un mapa de disparidad de la escena que se está fotografiando. Pero este no es el caso de la cámara Fujifilm 3D, que sólo genera las imágenes de ambas vistas. Esto produce que la ejecución de la interpolación de vistas contenga bastantes errores a la hora de generar la imagen autoestereoscópica, sobre todo si la imagen cuenta con pocos detalles (superficies con colores lisos, cielo, paredes, etc.) (Figura 34) (Figura 35).

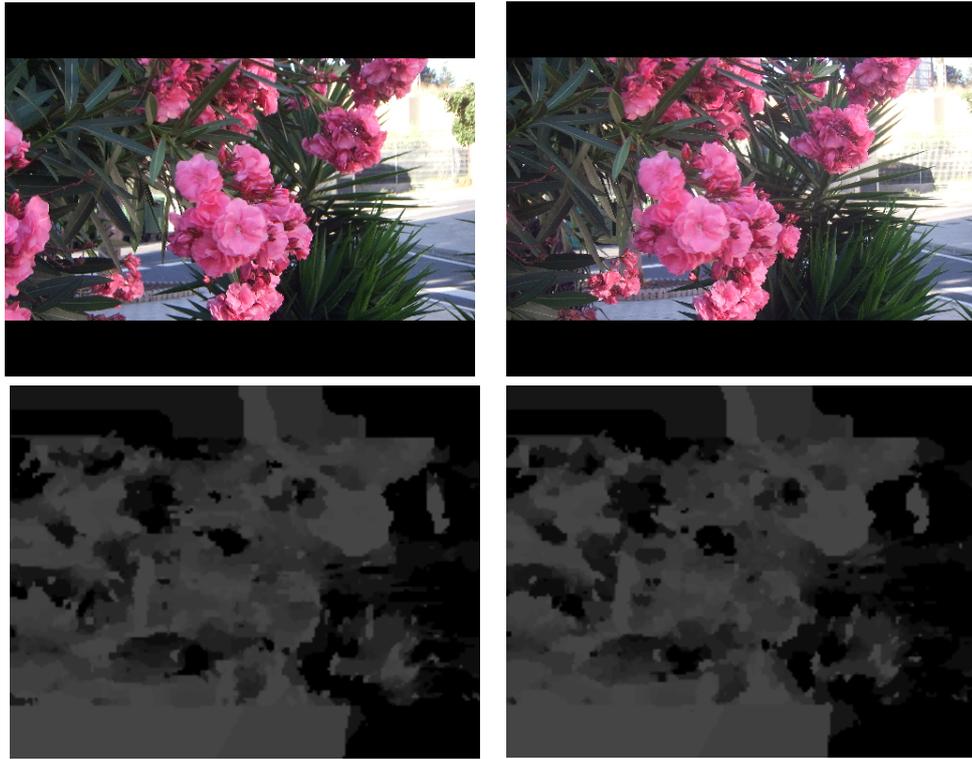


Figura 34. Arriba: imágenes de ejemplo de la cámara Fujifilm 3D. Abajo: mapas de disparidad obtenida al ejecutar el programa principal

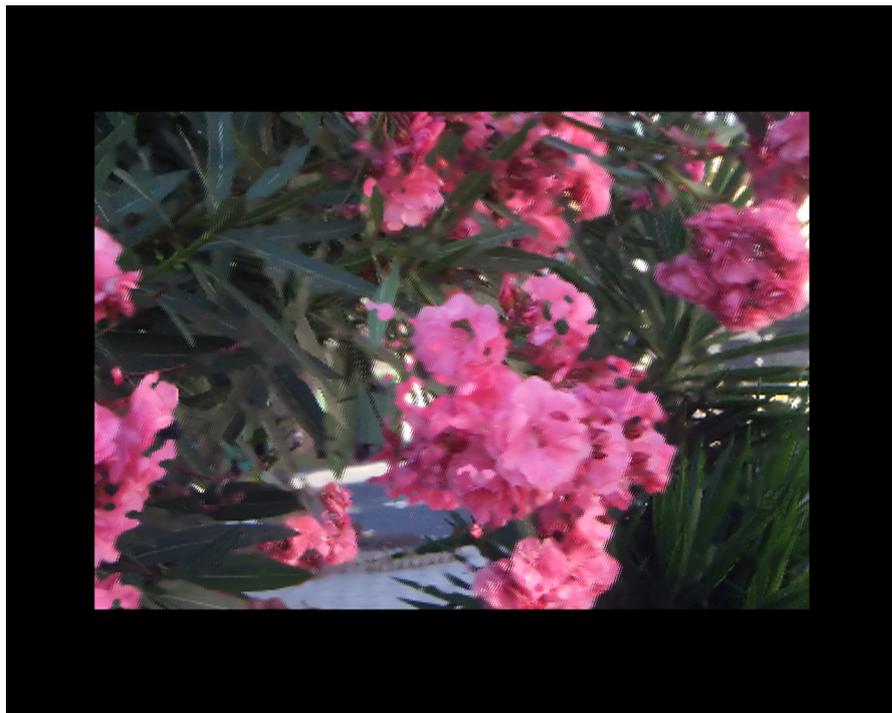


Figura 35. Imagen autoestereoscópica resultante de la ejecución del programa. Se pueden observar algunos errores

4.7.3. Búsqueda de información

Los dispositivos autoestereoscópicos son una tecnología bastante reciente, en la que aún se está trabajando. Esto produce que la cantidad de bibliografía disponible relacionada con esta tecnología sea algo escasa.

CAPITULO 5. APLICABILIDAD

5.1. Comparando las cámaras Lytro y Fujifilm 3D

Durante la realización de este proyecto, se han realizado una serie de pruebas con las distintas cámaras, obteniendo resultados suficientes para poder llevar a cabo una breve comparativa entre ambas.

5.1.1. Características

- *Modos de captura.* La cámara Fujifilm 3D permite utilizar distintos modos, entre los que destacan: el modo automático, el modo manual (establece el control de la velocidad del obturador y la abertura), el modo de exposición automática con ajuste manual de la abertura, el modo automático con ajuste programado y la toma 3D individual (permite realizar una fotografía 3D a partir de dos tomas con cada obturador individual). Mientras que la cámara Lytro solo ofrece al usuario la posibilidad de activar el “modo creativo”, que permite mayor control de la cantidad de desenfoque en la escena, simplemente tocando en la pantalla la zona que se desea enfocar. También, la cámara Lytro permite acceder manualmente a los ajustes de la velocidad del obturador y al ajuste de la sensibilidad ISO.



Figura 36. Pantalla de ajustes Lytro

- *Sensibilidad ISO.* La sensibilidad ISO marca la cantidad de luz que necesita nuestra cámara para hacer una fotografía. La cámara Lytro tiene mayor sensibilidad a la luz que la cámara Fujifilm 3D, ya que cuenta con mayor índice ISO (3,200 ISO frente a 1,600 ISO). Así, las imágenes realizadas en escenas con poca luz, quedarán mejor las que han sido tomadas con la cámara Lytro.
- *Flash.* Aunque la cámara Lytro tiene mayor sensibilidad ISO, no tiene flash incorporado, por lo que si necesitamos realizar alguna fotografía en la que sea imprescindible el uso del flash sólo la podremos realizar con la cámara Fujifilm 3D.
- *Resolución.* Si lo que el usuario busca es trabajar con resolución alta en las fotografías, la mejor opción es la Fujifilm 3D, ya que las imágenes realizadas con este dispositivo tienen mayor resolución que las realizadas con la Lytro.
- *Pantalla.* La pantalla de la cámara Fujifilm 3D (3.5”) es mucho más práctica que la pantalla de la Lytro (1.5”) y la imagen se ve mejor en este dispositivo. Además, en la cámara Fujifilm 3D se puede observar el efecto de profundidad directamente, sin necesidad de visualizar la imagen en un dispositivo que admita la tecnología 3D. Sin embargo, la cámara Lytro al ser táctil, ofrece facilidades en relación al enfoque de los elementos de la escena (Figura 37).



Figura 37. Comparativa de las pantallas de las cámaras

5.1.2. Facilidad de toma

- *Ergonomía.* La ergonomía es la disciplina que se encarga del diseño de lugares de trabajo, herramientas y tareas, de modo que coincidan con las características fisiológicas, anatómicas, psicológicas y las capacidades del usuario. Con respecto a este aspecto, la cámara Fujifilm 3D es mucho más ergonómica que la Lytro, ya que la toma de imágenes se realiza de forma similar a las cámaras digitales convencionales. Sin embargo, se debe tener especial cuidado con no dejar algún dedo delante de alguno de los objetivos, o el efecto 3D se verá afectado (Figura 38).

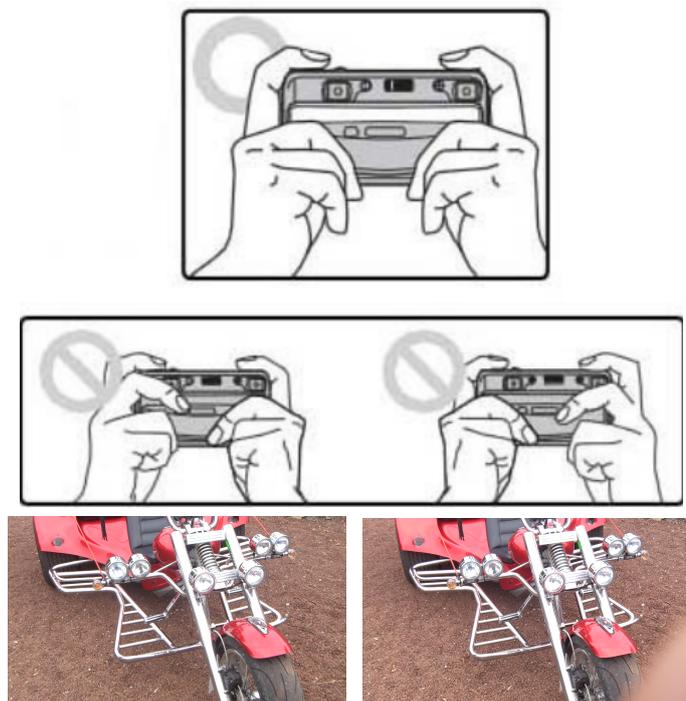


Figura 38. Problema al establecer un obstáculo en el objetivo de la cámara 3D

- *Planos.* Si queremos realizar una fotografía en 3D con un objeto en primer plano (Figura 38), la cámara con la que obtendremos el mejor resultado es la Lytro, ya que si al utilizar la Fujifilm 3D intentamos obtener un objeto en primer plano, la imagen queda desenfocada. Si se realiza una foto 3D con la cámara Fujifilm y el factor de zoom es demasiado elevado o el sujeto se encuentra mas cerca del rango recomendado (1,5 – 2 m), la imagen se verá difuminada. En la siguiente imagen se observa un elemento en primer

plano, que no podría haberse realizado con la cámara Fujifilm porque el elemento se encuentra demasiado cerca de la cámara ($< 1,5$ m).



Figura 39. Ejemplo de fotografía 3D con objeto en primer plano

Sin embargo, si queremos realizar una imagen 3D de una escena con plano general, el efecto 3D se mostrará de forma óptima en las imágenes que se han realizado con la Fujifilm 3D, mientras que en las realizadas con la Lytro se notará muchos menos (Figura 39).

- *Reenfoco y cambio de perspectiva.* Se trata del punto fuerte de la cámara Lytro, por lo que si, tras realizar la toma de la imagen, queremos volver a enfocar un elemento diferente o cambiar la perspectiva, sólo lo podremos hacer con la cámara Lytro (Figura 40).



Figura 40. Ejemplo de reenfoque con cámara Lytro

5.1.3. Modo de compartir imágenes.

La cámara que ofrece mayores facilidad para compartir las imágenes es la Lytro. Gracias a la incorporación de tecnología wifi, podremos compartir directamente las fotografías con nuestro smartphone (a través de la aplicación oficial de Lytro) o con nuestro ordenador personal sin la necesidad de utilizar cables. Sin embargo, para un usuario de nivel básico, será más fácil visualizar las imágenes de la cámara Fujifilm 3D en cualquier dispositivo que admita tecnología 3D, ya que para visualizar las imágenes de la Lytro es necesario descargar el software oficial de esta compañía (Figura 41).

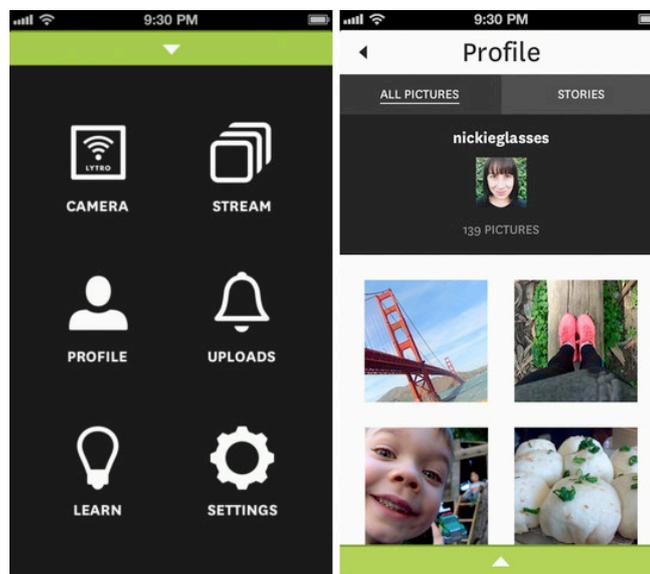


Figura 41. Aplicación oficial Lytro para smartphones

5.2. Resumen comparativa Fujifilm 3D – Lytro

La comparativa de las diferentes características de la cámaras la podríamos resumir en una tabla (Tabla 3).

	Lytro	Fujifilm 3D
Modos de captura	- Modo creativo - Modo manual (velocidad del obturador y sensibilidad ISO)	- Modo automático - Modo manual (velocidad obturador y la abertura) - Modo de exposición automática con ajuste manual de la abertura - Modo automático con ajuste programad - Toma 3D individual
Sensibilidad ISO	3,200 ISO - Mayor sensibilidad a la luz.	1,600 ISO - Poca sensibilidad a la luz.
Flash	No incorporado. No permite hacer fotos donde sea imprescindible.	Incorporado. Se pueden realizar fotos donde el flash se necesario.
Resolución	Imágenes con resolución más baja que las de Fujifilm 3D.	Mayor resolución que la cámara Lytro.
Pantalla	1.5" – Táctil con facilidades para enfocar.	3.5" – Permite visualizar en el momento el efecto 3D.
Ergonomía	Dificultad para realizar fotografías por su diseño.	Más ergonómica que la Lytro. Pero con posibilidad de tapar uno de los objetivos.
Planos	- Posibilidad de sacar objetos en primer plano. - En escenas de plano general, no se observa tanto el efecto 3D.	- Objetos en primer plano producen imagen desenfocada. El elemento más cercano (+ 1.5 m). - En escenas de plano general, se ve perfectamente el efecto 3D.
Reenfoque y cambio de perspectiva (tras la toma).	Permitido.	No permitido.

Tabla 3. Resumen de la comparativa de las cámaras Lytro y Fujifilm 3D

5.3. Uso del monitor estereoscópico y del monitor autoestereoscópico

A la hora de visualizar las imágenes 3D realizadas con la cámara Fujifilm 3D, se observará mejor la sensación de profundidad en el monitor estereoscópico antes que en el autoestereoscópico. Esto se debe a que la cámara, sólo realiza dos puntos de vista de la escena (izquierda y derecha) y para visualizar una imagen en un monitor autoestereoscópico necesitamos más puntos de vista de la escena. Como es necesario aplicar un algoritmo de interpolación de vistas, y la cámara Fujifilm 3D no genera los mapas de disparidad, el algoritmo no funcionará de manera óptima, obteniendo errores en la imagen.

Sin embargo, la sensación de profundidad en las imágenes que han sido realizadas con la Lytro se percibe mejor en el monitor autoestereoscópico.

CAPITULO 6. CONCLUSIONES Y TRABAJO FUTURO

El programa principal de este proyecto tiene como entrada una serie de imágenes realizadas con una cámara estereoscópica y con una cámara plenóptica y que se utilizan para visualizarlas en monitores estereocópicos y autoestereoscópicos. La creación de este programa ha sido una experiencia muy gratificante y con la que he podido aprender en muchos aspectos, tanto a nivel de programación, cuestiones de aplicabilidad de los dispositivos e incluso en el área de la fotografía computacional. Con la realización de este proyecto, he podido poner en práctica muchos de los conocimientos de Java, Python y el lenguaje de MATLAB que he ido adquiriendo a lo largo de los años en los que he estado en la carrera de Ingeniería Informática. Además, me ha motivado el hecho de poder trabajar con dispositivos bastante novedosos, como son las cámaras plenópticas y los monitores autoestereoscópicos, ya que, por ejemplo, hace tan solo 3 años no nos planteábamos la posibilidad de volver a enfocar distintos elementos de una escena después de haber realizado una fotografía. También, he podido incorporar al programa principal métodos innovadores como, por ejemplo, el algoritmo de interpolación de vistas.

Desde mis comienzos en Informática he intentado encontrar alguna tarea o proyecto relacionado con el campo de la Fotografía, y este es un hecho que pude llevar a cabo gracias a mi participación en este Trabajo de Fin de Grado.

En cuanto a la propuesta de trabajos futuros, se debe centrar en seguir trabajando en los dispositivos más actuales, como son la cámara plenóptica y el monitor autoestereoscópico. Varias compañías que fabrican smartphones están trabajando para crear sus propias cámaras plenópticas e incorporarlas a sus dispositivos. Sería interesante continuar investigando con las cámaras plenópticas y los dispositivos móviles, lo que llevaría a la creación de aplicaciones que se utilicen para mejorar las características de los dispositivos actuales: mejorar el reconocimiento de rostros, creación de imágenes 3D con el móvil, incorporar pantallas de visualización 3D a los dispositivos, etc. En cuanto a la tecnología de los dispositivos autoestereoscópicos, se debe mejorar el algoritmo de interpolación de vistas. Esto ayudaría a que la imagen resultante de aplicar la interpolación de vistas

no contenga ningún error y se vea perfectamente en un monitor autoestereoscópico, incluso cuando la cámara no proporcione el mapa de disparidad de la escena. Por último, se podría trabajar con los videos que produce la cámara Fujifilm 3D y estudiar como se podrían visualizar en un monitor autoestereoscópico.

CAPITULO 7. CONCLUSIONS AND FUTURE

The project's main program uses a set of images taken by stereoscopic and plenoptic cameras. These images are used to be shown in stereoscopic and autostereoscopic displays. This program development has been a very exciting experience in which I have been able to learn aspects related to devices applicability, programming and even in a computational photography way.

By working on this project I have put into practice my knowledge in Java, Python and MATLAB language that I have been gained through all the years studying Computer Engineer degree. Besides, I have found the fact of working with such modern devices very stimulating. Just three years ago we didn't thought about the chance of refocusing different items from a scene after taking a picture. I have also been able to add innovating methods such as the algorithm of view interpolation to the main program.

I have attempted to find any project or task related to photography field since my start in the degree. Finally I have reached this opportunity thanks to my participation in this final project.

For proposed future work, should be focus to further working on newer devices, such as plenoptic camera and autostereoscopic display. Several companies that manufacture smartphones are working to create their own plenoptic cameras and incorporate them into their devices. It would be interesting to keep further investigating plenoptic cameras and mobile devices, leading to the creation of applications that would be used as an improve for today's devices: enhancing face-recognition, creating 3D images with mobiles, incorporating 3D screen displays, etc. About autostereoscopic devices technology, the algorithm of view interpolation must be improved. This would help the resulting image not containing an error, and because of that, it would be perfectly displayed on an autostereoscopic monitor, even when camera doesn't provide the scene's disparity map. Finally, it could be investigate with videos taken by Fujifilm 3D and study how might display them on autostereoscopic displays.

BIBLIOGRAFÍA

- [1] Pérez Nava, F.; Rodríguez Valido, M.; Magdaleno, E. “Tecnología y matemáticas en el cine y la televisión en tres dimensiones”. *Matematicalia*, vol. 7, nº 4 (2011), p.1-10.
- [2] *Lytro*. Living Pictures and Lytro desktop. [Consulta: 15 agosto 2014]. Disponible en: <https://www.lytro.com/downloads/>
- [3] *Cámara estereoscópica*. [Consulta: 15 agosto 2014] Disponible en: <http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/394-c%C3%A1mara-estereosc%C3%B3pica>
- [4] *La fotografía estereoscópica*. [Consulta: 15 agosto 2014] Disponible en: <http://www.fotonostra.com/biografias/estereoscopica.htm>
- [5] *Aplicaciones de la estereoscopia*. [Consulta: 15 agosto 2014] Disponible en: <http://www.red3i.es/stereoweb/aplica.htm>
- [6] *FinePix Real 3D W3*. Especificaciones cámara 3D Fujifilm. [Consulta: 15 agosto 2014]. Disponible en: <http://www.fujifilm.eu/es/products/digital-cameras/fun-and-easy/model/finepix-real-3d-w3/>
- [7] *The Fuji 3D Camera*. A presentation by George Themelis. [Consulta: 15 agosto 2014]. Disponible en: <http://www.drt3d.com/W06-Fuji3d.pdf>
- [8] Pérez Nava, F.; Pérez Nava, A.; Rodríguez Valido, M.; Magadelono Castellò. “*Plenoptic Sensors*”. Aparecerá en “Biologically inspired Vision”. Springer. (2014)
- [9] *Lytro*. User manual. [Consulta: 15 agosto 2014] Disponible en: <https://www.lytro.com/downloads/resources/Lytro-Official-User-Guide.pdf>

- [10] *3-D displays. A review of current technologies.* Disponible en: <http://www.dgp.toronto.edu/~gf/Research/Volumetric%20UI/3-D%20Displays%20A%20review%20of%20current%20technologies.htm>
- [11] Light Field Toolbox v.02. Disponible en: <http://www.mathworks.com/matlabcentral/fileexchange/41511-light-field-toolbox-v0-2>
- [12] Fast view interpolation from Stereo. Disponible en: <http://www.cc.gatech.edu/conferences/3DPVT08/Program/Papers/paper213.pdf>
- [13] Monte Media Library. Librería para Java. Disponible en: <http://www.randelshofer.ch/monte/>
- [14] LFP File Reader. Disponible en: <http://code.behnam.es/python-lfp-reader/>
- [15] Eclipse. Plataforma de implementación y desarrollo. Disponible en: <https://www.eclipse.org/home/index.php>
- [16] MATLAB. Plataforma desarrollo matemático. Disponible en: <http://www.mathworks.es/products/matlab/>
- [17] MATLAB JA Builder. Plataforma transformación funciones MATLAB a Java. Disponible en: <http://www.mathworks.es/products/javabuilder/>