

Método para la creación de un campo de luz con cámaras plenópticas

Autores:

Jhon Sebastián Yagama Parra

Juan Sebastián Mora Zarza

Director:

ING.Francisco Calderón Bocanegra. PhD.

Codirector:

Julián David Colorado Montaña, Ph.D.



Pontificia Universidad Javeriana de Bogotá
Facultad de ingeniería, Departamento de Electrónica
2023

Abstract

This degree project proposes a method for obtaining lightfields in Levoy's notation, and a method for evaluating these lightfields using the Lightfield Toolbox libraries in Matlab for plenoptic cameras, especially type 2.0 cameras such as the Raytrix R42. The method for obtaining the lightfields is based on the implementation of an algorithm designed in Python that uses concepts such as computer vision, image processing, artificial intelligence, and image reconstruction. On the other hand, the evaluation method is based on explaining, demonstrating and taking advantage of the libraries that this tool for lightfields offers, focusing on developing a method for evaluating lightfields oriented to the synthetic refocusing libraries.

Resumen

Este trabajo de grado propone un método para la obtención de campos de luz en notación de Levoy y un método para evaluar estos campos de luz con ayuda de las librerías del Lightfield Toolbox de Matlab para cámaras plenópticas, en especial las cámaras tipo 2.0 como la Raytrix R42. El método para la obtención de los campos de luz se basa en la implementación de un algoritmo diseñado en Python que emplea conceptos como la visión artificial, procesamiento de imágenes, inteligencia artificial y reconstrucción de imágenes. Por otra parte, el método de evaluación se basa en explicar, demostrar y aprovechar las librerías que esta herramienta de trabajo para los campos de luz ofrece, centrándose en desarrollar un método de evaluación de los campos de luz orientado a las librerías de reenfoque sintético.

Índice general

1. Introducción	1
1.1. Justificación	1
1.2. Problemática	2
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
1.4. Estado del arte	3
1.5. Marco teórico	8
1.5.1. El campo de luz	8
1.5.2. Cámaras plenópticas	9
1.5.3. Cámara basada en lentes	11
1.5.4. Cámara basada en lentes enfocadas	12
1.5.5. Notacion de Levoy	12
1.5.6. Visión artificial	13
1.5.7. OpenCV	13
2. Diseño de la solución	14
2.1. Obtención de los campos de luz en notación Levoy	15
2.2. Método de Evaluación usando LFToolbox	16
3. Implementación para la creación y obtención de un campo de luz	17
3.1. Calibración de la cámara Raytrix R42 y adquisición de las imágenes	18
3.2. Procesamiento de la imagen en blanco	19
3.2.1. Pre-Thresholding	19
3.2.2. Thresholding por rangos	19
3.2.3. Detección de contornos considerando la jerarquía	20
3.2.4. Detección de centroides para cada lente a partir del momento (M)	21
3.3. Clasificación para cada tipo de lente	23
3.3.1. Parámetros	23
3.3.2. Detección de contornos Padre	23
3.3.3. Clasificación por K-Means	26
3.4. Corrección para la distribución de píxeles	27
3.4.1. Agrupación de centroides en el eje y usando K-means	27
3.4.2. Reorganización en el eje x	29
3.5. Obtención de los campos de luz para cada tipo de lente	30
3.5.1. Reconstrucción de la imagen a partir de una imagen en blanco	30
3.5.2. Verificación del tamaño de la matriz usada en la reconstrucción	31

4. Implementación del método de evaluación usando LFToolbox	33
4.1. Preparación de imagen en Blanco para imágenes Lytro	34
4.2. Decodificación para imágenes Lytro	34
4.3. Calibración y Rectificación para imágenes Lytro	36
4.4. Lectura de campos de luz	37
4.4.1. Lectura de campos de luz Lytro	38
4.4.2. Lectura de campos de luz Raytrix R42	38
4.5. Visualización de campos de luz	39
4.6. Filtrado	41
4.6.1. Filtro de suma y desplazamiento	41
4.6.2. Filtro lineal 2D	42
4.6.3. Filtro plano 4D	42
4.6.4. Filtro Hyperfan 4D	43
5. Análisis y resultados	44
5.1. Dataset de fotos	44
5.1.1. Dataset Raytrix R42 (Anexo 1)	45
5.1.2. Dataset cámara Lytro (Anexo 1)	46
5.2. Variación de calidad en imágenes blancas obtenidas por la Raytrix R42	48
5.3. Variación de lentes en la cámara Raytrix R42 para la creación del campo de luz	51
5.3.1. Lente 58001	51
5.3.2. Lente 59872	53
5.4. Variación del campo de luz para cada tipo de lente	55
5.5. Análisis de los filtros de LFToolbox	56
5.5.1. Filtro de suma y desplazamiento	56
5.5.2. Filtro de línea 2D	57
5.5.3. Filtro plano 4D	58
5.5.4. Filtro Hyperfan 4D	59
5.6. Sharepoint	61
6. Conclusiones y trabajos futuros	64
6.1. Conclusiones	64
6.2. Trabajos futuros	64
7. Anexos	67
Bibliografía.	68

Índice de figuras

1.1.	Resultados de reenfoque sintético con Lytro 1 [1].	4
1.2.	Solución propuesta en demultiplexación de campo de luz [2].	5
1.3.	Decodificación de la imagen del sensor 2D sin procesar en un campo de luz 4D [4].	7
1.4.	Parametrización de un campo de luz [3].	9
1.5.	Estructura básica de una cámara plenóptica enfocada [4].	10
1.6.	Varios arreglos MLA (a) ortogonales con solo tipo de lente (b) hexagonal, lo que proporciona un mejor factor de relleno (c) hexagonal con tres tipos de lentes, cada uno con una distancia focal diferente [4].	10
1.7.	La cámara plenóptica basada en lentes: luz de un objeto en la escena (derecha) pasa a través de la lente principal (azul) y se enfoca en una matriz de lentes (verde). las lentillas dividen la luz entrante en muchos píxeles (izquierda), según la dirección de llegada. Un solo píxel es resaltado en amarillo, que ilustra el tamaño de apertura de una cámara convencional con la misma profundidad de campo [4].	11
2.1.	Diagrama de bloques de la solución propuesta	14
2.2.	Diagrama del sub-bloque para la obtención de los campos de luz	15
2.3.	Diagrama del sub-bloque para el método de evaluación de los campos de luz	16
3.1.	Variables mecánicas de apertura y foco del lente <i>HF-1218-12M</i> de la Cámara Raytrix R42	18
3.2.	a: Muestra de calibración de la foto en blanco para la apertura de las micro lentes usada en los resultados. b: calibración de foco usado para el ejemplo, en este caso la flor está enfocada y el plano del fondo desenfocado.	18
3.3.	a: Muestra de la foto en blanco original. b: Muestra de la foto en blanco en escala de grises.	19
3.4.	a: Muestra de la foto en escala de grises. b: Muestra de la foto al ser umbralizada por rangos.	20
3.5.	Muestra de los dos contornos hallados para cada tipo de micro lente. Nota: Los contornos 1, 3 y 5 son hijos de los contornos 0, 2 y 4 respectivamente y estos a su vez son sus contornos padres.	20
3.6.	a: Muestra de la foto al ser umbralizada por rangos. b: Muestra de la foto original con los contornos dobles hallados para cada micro lente.	21
3.7.	Muestra de la foto en blanco con los centroides marcados para cada micro lente.	22
3.8.	Muestra de una foto en blanco que enseña únicamente las zonas fuera de los enfoques de los lentes en escala de grises y a la derecha se presenta un acercamiento de los tres tipos de lentes señalando el perímetro usado.	23
3.9.	Diagrama de flujo del método implementado para la detección de contornos padre y filtrado de datos basura o ruido tomando como ejemplo para los datos de entrada la tabla 3.1. Siendo la variable <i>contorno</i> todos los contornos calculados, por otra parte, la variable <i>a</i> es el contorno primer hijo y la variable <i>b</i> es el contorno padre, esto para cada contorno procesado.	24
3.10.	Gráfica de los 3 clústers generados dando así la clasificación para cada lente.	26

3.11. a: Muestra de la foto en blanco original. b: Muestra de la foto en blanco luego de ser clasificado cada micro lente, esta resalta los contornos en tres distintos colores dependiendo del tipo de lente que sea.	26
3.12. a: Muestra de la foto en blanco original donde se observan imperfecciones. b: Muestra de la foto en blanco con las imperfecciones que pueden afectar el algoritmo señaladas.	27
3.13. Acercamiento de la gráfica de los 176 clústers generados para clasificar cada fila de la matriz de contornos, cada punto hace referencia a un clúster en el cual se agrupa las coordenadas de los puntos centrales de cada contorno.	27
3.14. Diagrama de flujo para la organización del eje Y.	28
3.15. Diagrama de flujo para la organización del eje X.	29
3.16. a: Fotografía original sin procesar. b: imágenes reconstruidas a partir de los 3 tipos de lentes	30
3.17. Mosaico del campo de luz completo y creación de carpetas para almacenamiento.	31
3.18. Matriz 15x15 usada para la reconstrucción del campo de luz de cada micro lente.	32
3.19. Ejemplo del campo de luz al usar una matriz mayor a la recomendada por el algoritmo. . .	32
4.1. Modelo de cuadrícula dado por LFToolbox donde se exponen las 4 esquinas de la imagen en blanco creada y su centro.	34
4.2. Proceso de decodificación de una imagen 2D	35
4.3. Fotografías antes y después de la corrección de color.	35
4.4. Puntos hallados en el set de fotografías del patrón de ajedrez.	36
4.5. muestra del patrón con las medidas de entrada pedidas por la función.	36
4.6. Fotografías antes y después de la calibración y rectificación, donde se puede observar la corrección de algunas rectas en las imágenes.	37
4.7. Ejemplo de la lectura de un campo de luz implementada para cámaras Lytro.	38
4.8. Ejemplo de la lectura de un campo de luz implementada para cámaras Raytrix R42.	39
4.9. Visualización de los campos de luz creado con la función <i>LFDisp</i> . a: Campo de luz creado por la cámara Lytro. b: Campo de luz creado por la cámara Raytrix R42.	39
4.10. Visualización de los campos de luz creado con la función <i>LFDispTiles (stuv)</i> tipo mosaico. a: Campo de luz creado por la cámara Lytro. b: Campo de luz creado por la cámara Raytrix R42.	40
4.11. Visualización del campo de luz creado con la función <i>LFDispTiles(uvst)</i>	40
4.12. Visualización del campo de luz creado con la función <i>LFDispTiles(uvst)</i> visto en seis subfiguras dadas en pares de distintas dimensiones.	40
4.13. Representación visual del desplazamiento creado por el filtro.	41
4.14. a: Filtro lineal 2D pasabandas implementado para los campos de luz creados por la cámara Lytro. b: Filtro pasabandas implementado para los campos de luz creados por la cámara Raytrix R42.	42
4.15. a: Filtro plano 4D pasabandas implementado para los campos de luz creados por la cámara Lytro. b: Filtro pasabandas implementado para los campos de luz creados por la cámara Raytrix R42.	43

4.16. a: Filtro Hyperfan 4D pasabandas implementado para los campos de luz creados por la cámara Lytro. b: Filtro pasabandas implementado para los campos de luz creados por la cámara Raytrix R42.	43
5.1. Software de calibración cámara Raytrix R42	45
5.2. Fotografías capturadas con la cámara Raytrix R42	46
5.3. Software para exportar las imágenes de la cámara Lytro	47
5.4. Imágenes capturadas con la cámara lytro, exportadas a través del software [5].	47
5.5. Muestra de tres distintas fotografías tomadas para la verificación de calidad del algoritmo, en estas se ve una fotográfica con una gran apertura, una fotografía con apertura media y una fotografía con apertura reducida.	48
5.6. Muestra de una fotográfica blanca con una apertura ampliada.	48
5.7. Muestra de una fotográfica blanca con una apertura ideal.	49
5.8. Muestra de una fotográfica blanca con una apertura reducida.	50
5.9. Lente 58001 y muestra de la fotográfica blanca tomada este.	51
5.10. Gráfica de clasificación de los micro lentes para la fotografía tomada con el lente principal 58001.	52
5.11. Muestra de una esquina de la fotográfica blanca tomada con el lente 58001, donde se evidencia el cierre progresivo de los micro lentes.	52
5.12. Lente 59872 y muestra de la fotográfica blanca tomada este.	53
5.13. Gráfica de clasificación de los micro lentes para la fotografía tomada con el lente principal 59872.	53
5.14. a: Muestra de la reconstrucción de la imagen para el lente tipo 3. b: Muestra de la reconstrucción de la imagen para el lente tipo 2. c: Muestra de la reconstrucción de la imagen para el lente tipo 1.	54
5.15. a: Muestra de la reconstrucción de la imagen para el lente tipo 3. b: Muestra de la reconstrucción de la imagen para el lente tipo 2. c: Muestra de la reconstrucción de la imagen para el lente tipo 1.	55
5.16. a: Muestra de la reconstrucción de la imagen para el lente tipo 3. b: Muestra de la reconstrucción de la imagen para el lente tipo 2. c: Muestra de la reconstrucción de la imagen para el lente tipo 1.	55
5.17. Comparación entre el campo de luz de entrada y el filtro suma y desplazamiento para muestras de Lytro. a: Muestra del campo de luz de entrada. b: Filtro con una pendiente de $-2/9$. c: Filtro con una pendiente de $6/9$	56
5.18. Comparación entre el campo de luz de entrada y el filtro suma y desplazamiento para muestras de Raytrix. a: Muestra del campo de luz de entrada. b: Filtro con una pendiente de 0.	57
5.19. Comparación entre el campo de luz de entrada y el filtro lineal 2D para muestras de Lytro.	57
5.20. Comparación entre el campo de luz de entrada y el filtro lineal 2D para muestras de Raytrix.	58
5.21. Salida del filtro plano 4D para muestras de Lytro.	59
5.22. Comparación entre el campo de luz de entrada y el filtro lineal 4D para muestras de Raytrix.	59
5.23. Salida del filtro Hyperfan 4D para muestras de Lytro.	60

5.24. Comparación entre el campo de luz de entrada y el filtro Hyperfan 4D para muestras de Raytrix.	61
5.25. Pagina principal Sharepoint	61
5.26. a:Justificación del proyecto en Sharepoint, b: Estados del arte del proyecto en Sharepoint. .	62
5.27. Paso a paso para emplear campos de luz con LFToolbox.	62
5.28. Sección de resultados en el Sharepoint.	63
5.29. Sección de conclusiones en el Sharepoint.	63

Índice de tablas

3.1. Ejemplo del orden jerárquico que retornaría la función para los contornos de la figura 3.5. . .	21
5.1. Resultados en la detección de micro lentes para una apertura ampliada.	49
5.2. Resultados en la detección de micro lentes para una apertura ideal.	49
5.3. Resultados en la detección de micro lentes para una apertura reducida.	50
5.4. Resultados en la detección de micro lentes para fotografías en blanco tomadas por el lente 59872.	54

CAPÍTULO 1

Introducción

Contenido

1.1. Justificación	1
1.2. Problemática	2
1.3. Objetivos	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos	3
1.4. Estado del arte	3
1.5. Marco teórico	8
1.5.1. El campo de luz	8
1.5.2. Cámaras plenópticas	9
1.5.3. Cámara basada en lentes	11
1.5.4. Cámara basada en lentes enfocadas	12
1.5.5. Notacion de Levoy	12
1.5.6. Visión artificial	13
1.5.7. OpenCV	13

La importancia de la ingeniería radica en la aplicación del conocimiento científico en base a la creación, el diseño, la implementación y la gestión, materializando soluciones para los desafíos cotidianos de nuestra sociedad. En este capítulo se aborda la problema principal del proyecto de grado y el enfoque asumido a través del cual se busca desarrollar nuevos métodos para la creación, desarrollo y manejo de campos de luz creados por camaras plenópticas, orientado a las plenópticas 2.0.

1.1. Justificación

La agricultura es un campo de conocimiento esencial que ha evolucionado a lo largo de la historia para satisfacer las necesidades alimentarias de la sociedad en constante crecimiento. El desarrollo agrícola a nivel mundial puede constituir uno de los medios más importantes para poner fin a la pobreza extrema, impulsar la prosperidad colectiva y alimentar a una población que se espera llegue

a 9700 millones de habitantes en 2050, así mismo, el crecimiento de la agricultura es entre dos y cuatro veces más eficaz que el de otros sectores para incrementar los ingresos de los más pobres, además de que el 65 % de los adultos pobres que trabajan viven de la agricultura. [6]

En el contexto Colombiano, la agricultura aun es un proceso básico, rudimentario y tosco; esto hace que la productividad en este sector se desarrolle tardíamente con respecto al desarrollo de esta área en potencias mundiales; y aunque la agricultura ha sido testigo de avances tecnológicos significativos que han transformado la forma en que cultivamos y producimos alimentos, como la implementación de tecnologías innovadoras, por ejemplo sensores de monitoreo y drones que ha mejorado la eficiencia y la productividad en el sector agrícola; sigue existiendo una brecha bastante grande entre el desarrollo tecnológico y la implementación de prácticas agrícolas modernas en las diferentes regiones del país. Esta desconexión representa un desafío para alcanzar un sistema agrícola sostenible y eficiente.

Es en este contexto que proyectos como Omicas, en la Universidad Javeriana de Cali, han surgido para abordar estas problemáticas y contribuir al avance de la agricultura de precisión; el cual busca, a través de siete proyectos, desarrollar e implementar estrategias científico-tecnológicas para el mejoramiento de variedades agrícolas con el objetivo de aportar a la seguridad alimentaria y una producción sostenible a nivel mundial.

1.2. Problemática

El proyecto de fenómica llevado a cabo por OMICAS busca desarrollar e implementar una plataforma multimodal de bajo costo y distribuida para fenotipado de cultivos, que integre características relevantes, específicamente de plantas, para la medición y la cuantificación de las variables que afectan a la respuesta de cultivos en el ambiente. [7] Para el desarrollo de esta plataforma multimodal es necesaria una gran base de datos que integre investigaciones en las áreas de creación, implementación y desarrollo de campos de luz, esto enfocado en las cámaras con las que cuenta la universidad, ya que, no cuentan con la suficiente documentación y desarrollo en dichas áreas.

Por esta razón, se planteó el desarrollo de un software que permite la creación y evaluación de campos de luz usando las camaras plenópticas Lytro y Raytrix R42, por medio de una robusta documentación que facilite el entendimiento de esta área y del diseño e implementación de algoritmos capaces de crear y evaluar los campos de luz capturados dando acceso a su uso en futuros proyectos.

1.3. Objetivos

1.3.1. Objetivo general

Diseñar un método que permita, a partir de una imagen tomada con la cámara Raytrix R42 y Lytro, la creación de un campo de luz, en la notación de Levoy de 1996, compatible con la librería de Matlab Light Field toolbox y documentar el desarrollo para posteriores investigaciones.

1.3.2. Objetivos específicos

- Adquirir un conjunto de datos de por lo menos 60 imágenes de muestras vegetales calibradas usando la cámara Raytrix R42 y Lytro.
- Diseñar un algoritmo que reciba de entrada una imagen del dataset de la cámara Raytrix R42 y Lytro y retorne un conjunto de imágenes que use la notación de Levoy para la modelación de su campo de luz.
- Evaluar el campo de luz en notación de Levoy sintetizado a partir de la captura de la cámara Raytrix R42 y Lytro, utilizando las herramientas de reenfoque sintético del Light Field toolbox de Matlab.
- Documentar en una página web de sharepoint todos los archivos de captura del software de la cámara plenóptica Raytrix R42 y Lytro. Donde se evidencie toda la documentación, resultados, análisis y conclusiones.

1.4. Estado del arte

La agricultura de precisión se ha consolidado como una herramienta para lograr una producción más eficiente y sostenible. Sin embargo, a pesar de los avances en este campo, aún existen desafíos por abordar. Este documento de trabajo de grado se enfoca en una de estas problemáticas, la falta de información acerca de softwares de generación de campo de luz y su manejo en el contexto de la fenotipificación de cultivos. Con el objetivo de contribuir al desarrollo de nuevas metodologías en este ámbito, se llevará a cabo un estudio detallado de los antecedentes y estado del arte en este tema.

Procesamiento de señales plenópticas

Teniendo lo anterior en cuenta un gran precedente para el desarrollo de la investigación en el campo del procesamiento de señales plenópticas ha sido impulsado por los trabajos previos en el área. Uno de los mayores avances ha sido el trabajo “*Plenoptic signal processing*” [8], que combinó el procesamiento de señales multidimensionales con la representación basada en imágenes. Esta combinación es esencial para el desarrollo de un modelado plenoptico de luz.

La representación basada en imágenes se ha convertido en una alternativa atractiva para simplificar la representación de gráficos por computadora, ya que cambia el enfoque del modelo de capturar la geometría y la textura de una escena a modelar el comportamiento de la luz que la transmite. Este enfoque basado en datos permite una representación muy rápida y eficiente de datos arbitrarios, puntos de vista y con cámaras arbitrarias, lo que es esencial para los consumidores actuales de las cámaras plenópticas. Por lo tanto, la investigación en la representación de los campos de luz es un área que sigue siendo activa y en constante evolución, y es importante para garantizar que las tecnologías plenópticas cumplan con las expectativas de los consumidores y proporcionen soluciones innovadoras y efectivas en este campo [8].

Enfoque sintético con cámara Lytro (Marc Levoy)

En 1996 el ingeniero Marc Levoy comenzó una investigación con el objetivo de realizar enfoque sintético a fotografías tomadas con una cámara de mano, obteniendo de esta forma en 2005 un avance mediante un diseño que permite capturar campos de luz densos con la cámara de mano, este diseño permitió que las fotografías cotidianas se reenfocaran después de ser capturadas, en 2011 casi 15 años después del inicio de su investigación salió al mercado la cámara Lytro 1, con la cual el ingeniero realizo el reenfoque sintético con imágenes de distancia cercana y lejana (ver figura 1.1) [1].



Figura 1.1: Resultados de reenfoque sintético con Lytro 1 [1].

Finalmente obteniendo resultados de baja resolución debido a la definición de la cámara, teniendo en cuenta que eran reenfocables, de esta manera estas cámaras no tuvieron éxito y eventualmente salieron del mercado dando paso los modos de retrato de los dispositivos móviles, si bien las imágenes desenfocadas sintéticamente por los móviles no son físicamente correctas es muy complicado hallar la diferencia [1].

Demultiplexación de campo de luz y estimación de disparidad

En el ámbito de procesamiento posterior de imágenes capturadas por cámaras plenópticas, en el artículo “*Light-Field Demultiplexing and Disparity Estimation*” [2], en particular en el desarrollo con cámara lytro. Los autores presentan un enfoque innovador para estimar los centros de las microlentes y desmultiplexar la imagen bruta sin desmosaicarla previamente, lo que permite evitar artefactos en la imagen. Además, han desarrollado un nuevo algoritmo de coincidencia de bloques para estimar disparidades entre vistas plenópticas no desmosaicadas. Este algoritmo se basa en la configuración de la vista dada por la cámara plenóptica, que permite una estimación más precisa de las disparidades.

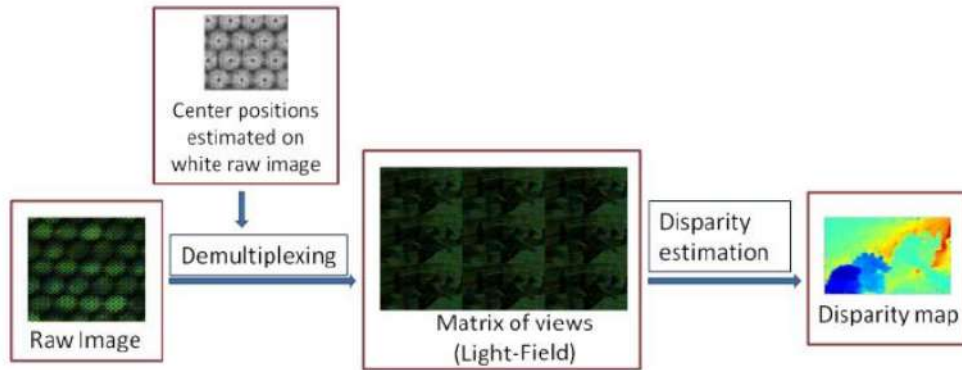


Figura 1.2: Solución propuesta en demultiplexación de campo de luz [2].

La figura 1.2 sigue un proceso llamado "pipeline". Para hacerlo más fácil de ver, se muestra una parte de las imágenes y las vistas en una matriz. Posteriormente obtienen el campo de luz (matriz de vistas) dividiendo los datos en mosaicos y usando la posición de la imagen central. Después, estiman la diferencia de distancia para una vista de referencia a partir del campo de luz. Los resultados demuestran que la estimación de disparidad es más precisa cuando la imagen bruta se desmultiplexa sin desmosaicar previamente. Este estudio es relevante para el campo de la representación de los campos de luz y puede ser utilizado como antecedentes en futuros trabajos de investigación en este ámbito [2].

Decodificación, calibración y rectificación para plenópticas basadas en lentes

La tecnología plenóptica ha tenido una evolución rápida en los últimos años, permitiendo la captura de información en profundidad y luz en una sola imagen. Esto ha llevado a una amplia gama de aplicaciones en áreas como la fotografía, la realidad virtual y la realidad aumentada. Sin embargo, antes de poder utilizar esta información, es necesario realizar un procesamiento posterior, incluyendo la calibración y la rectificación de las imágenes capturadas.

El estado del arte en la rectificación y la calibración de imágenes plenópticas incluye una amplia gama de métodos, desde métodos basados en geometría hasta métodos basados en aprendizaje automático. Cada enfoque tiene sus propias fortalezas y debilidades, y la elección del método adecuado depende de la aplicación específica y los requisitos de precisión.

La rectificación y la calibración son cruciales para la utilización eficiente de la información capturada por cámaras plenópticas y han sido objeto de intensa investigación en los últimos años. El artículo “*Decoding, Calibration and Rectification for Lenselet-Based Plenoptic Cameras*” [9], describe un procedimiento de decodificación, calibración y rectificación para cámaras plenópticas basadas en lenteletas adecuado para una variedad de aplicaciones de visión por computadora. Se deriva una matriz intrínseca 4D novedosa y físicamente basada que relaciona cada píxel grabado con su rayo correspondiente en el espacio 3D. Además, se propone un modelo de distorsión radial y una función objetiva práctica basada en la reproyección de rayos. El modelo de cámara de 15 parámetros es de mucha menor dimensionalidad que los modelos de matriz de cámaras y representa más estrechamente la física de las cámaras basadas en lenteletas.

Los resultados incluyen la calibración de una cámara comercialmente disponible utilizando tres tamaños de cuadrícula de calibración en cinco conjuntos de datos. Los errores típicos de reproyección de rayos RMS son 0.0628, 0.105 y 0.363 mm para cuadrículas de calibración de 3.61, 7.22 y 35.1 mm, respectivamente. Los ejemplos de rectificación incluyen objetivos de calibración e imágenes del mundo real [9].

Procesamiento de señales plenópticas para RobustVision en robótica de campo

El trabajo de grado “*Plenoptic Signal Processing for RobustVision in Field Robotics*” [10], propone el uso de cámaras plenópticas para mejorar la robustez y simplicidad de la visión por máquina en aplicaciones de robótica de campo. El polvo, la lluvia, la niebla, la nieve, el agua turbia y la luz insuficiente pueden hacer que incluso los sistemas de visión más sofisticados fallen. Las cámaras plenópticas ofrecen una alternativa atractiva a la imaginería convencional, reuniendo significativamente más luz sobre un campo de profundidad más amplio y capturando una estructura rica de campo de luz.

Las contribuciones clave de este trabajo consisten en explorar las propiedades de las señales plenópticas y desarrollar algoritmos para explotarlas. Comienza estableciendo los cimientos para el despliegue de cámaras plenópticas en aplicaciones de robótica de campo, proponiendo un nuevo modelo de cámara y esquemas para la decodificación, calibración y rectificación adecuados a dispositivos compactos basados en lentes.

La forma de la señal plenóptica en el dominio de frecuencia se elabora como la intersección de un hipercono 4D altamente selectivo y un ventilador 2D. Esta forma fundamentalmente cuatridimensional informa la construcción de filtros lineales eficientes que mantienen el campo de profundidad enfocándose en un volumen en lugar de un plano. Se mostro que estos filtros mejoran el contraste en condiciones de poca luz y a través de la atenuación de medios como agua turbia y niebla, mientras reduce el impacto de obstaculizadores como nieve, lluvia y materia particulada submarina.

Se consideran las propiedades de una escena estática vista por una cámara plenóptica móvil. Una derivación geométrica da lugar a una serie de métodos para realizar odometría visual sin características de 6 grados de libertad, generando modelos de escena 3D como un producto útil. La derivación culmina en una generalización en forma cerrada del flujo óptico que estima directamente el movimiento de la cámara a partir de las derivadas plenópticas de primer orden. Se demuestra una elegante adaptación de este llamado flujo plenóptico a la imaginería basada en lentes, así como un método sencillo y aditivo para generar vistas noveles [10].

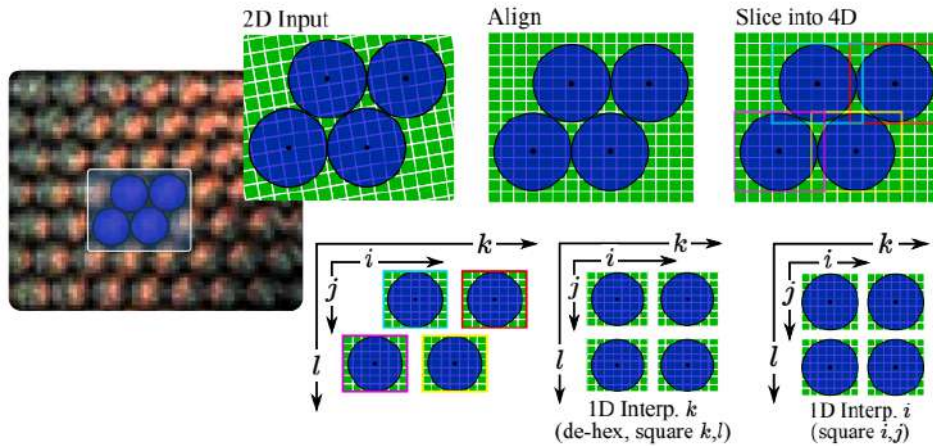


Figura 1.3: Decodificación de la imagen del sensor 2D sin procesar en un campo de luz 4D [4].

La figura 1.3 describe un método para decodificar una imagen 2D sin procesar a un campo de luz 4D. Primero, se corrige el viñeteado dividiendo la imagen por una imagen blanca como se observa en la parte izquierda de la 1.3. Luego continuando a la derecha, las imágenes de las lentes se encuentran en una cuadrícula girada en relación a los píxeles de la imagen. Por lo tanto, se vuelve a muestrear la imagen rotándola y escalándola para que los centros de las lentes coincidan con los centros de los

píxeles en la imagen remuestreada. Este paso requiere una escala que no sea cuadrada, por lo que los píxeles alineados son rectangulares [4].

1.5. Marco teórico

1.5.1. El campo de luz

Inicialmente, para comprender la naturaleza de un campo de luz, es necesario abordar su representación física-teórica, dada por la función plenóptica, y su visualización en imágenes digitales. La función plenóptica nace de un concepto de rayos-ópticos que asigna valores de luminosidad a los rayos de luz que se propagan en todas las direcciones sobre las tres dimensiones espaciales de una escena.

Estos rayos pueden ser bloqueados, atenuados o dispersados mientras se propagan entonces, la función plenóptica es una función de siete dimensiones denotada como:

$$L(x, y, z, \theta, \phi, \varphi, t) \quad (1.1)$$

Tres dimensiones espaciales (x, y, z) , dos dimensiones angulares (θ, ϕ) , una dimensión de color (φ) y una dimensión temporal (t) , que expresa en otras palabras, la luminosidad espectral por unidad de tiempo. Los campos de luz se derivan de la función plenóptica al introducir las siguientes restricciones [3]:

- Se consideran estáticos en el tiempo, es decir, una integración sobre el periodo de exposición de luz elimina la dimensión temporal de la función plenóptica. Sin embargo, con el avance de la tecnología los videos de campos de luz han sido realizables y lideran el estado del arte actual en la aplicación de estos para la realidad virtual.
- Se consideran monocromáticos, es decir, una integración de sensibilidad espectral sobre los píxeles del sensor de imagen, usando el Mosaico de Bayer o filtro de color, elimina la dimensión espectral de la función plenóptica.
- Se asume que los rayos de luz se propagan por el vacío, excepto aquellos que constatan el interior de la escena capturada por el lente de la cámara virtual. Por ende, se asume que la luminosidad es constante a lo largo de la región exterior a la escena, eliminando la tercera dimensión espacial de los rayos propagados [3].

Finalmente, un campo de luz se reduce a cuatro dimensiones de la función plenóptica:

$$L(x, y, z, \theta, \phi) \quad (1.2)$$

En la práctica, este se representa parametrizando los rayos en el espacio a partir de la intersección de estos en un primer plano espacial uv , y en un segundo plano angular st (ver figura 1.4, los rayos de luz son enfocados por un lente principal y después los rayos de luz son demultiplexados por un arreglo de micro lentes) [3]. Entonces, un campo de luz se define como:

$$L(u, v, s, t) \quad (1.3)$$

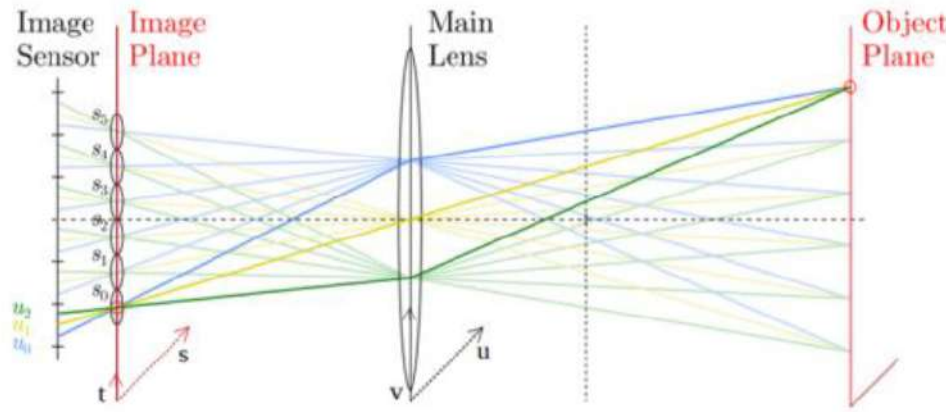


Figura 1.4: Parametrización de un campo de luz [3].

1.5.2. Cámaras plenópticas

Una cámara plenóptica es un dispositivo de adquisición de campos de luz por multiplexación espacial de imágenes; este sistema codifica el campo de luz de 4D en un sensor de imagen de 2D demultiplexando el dominio angular dentro del dominio espacial a través de un arreglo de microlentes, permitiendo capturas con un solo sensor de imagen y en una sola exposición, similar al esquema demostrado en la figura 1.4. Este mecanismo impone una compensación directa entre la resolución angular y la resolución espacial del campo de luz capturado [11].

El objetivo de una cámara plenóptica es capturar información direccional, y se logra añadiendo una estructura de matriz de micro lentes (MLA) a un sistema de cámara convencional. La configuración general de una cámara plenóptica se muestra en la Figura 1.5, donde la lente principal crea una imagen que luego es proyectada por las micro lentes en el plano de imagen [4].

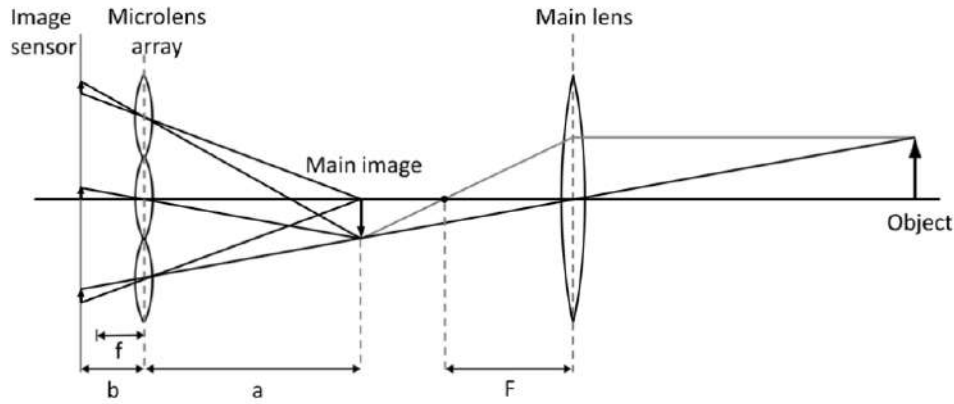


Figura 1.5: Estructura básica de una cámara plenótica enfocada [4].

En base a si la imagen capturada por el sensor es enfocada o borrosa, la cámara resultante se llama cámara plenótica convencional o basada en lentes enfocadas. Es importante destacar que la imagen también puede ser virtual, es decir, formada detrás del plano de imagen, lo que ayuda a reducir el tamaño de la cámara plenótica y es la configuración preferida, como en la cámara Raytrix R29. En una cámara plenótica basada en lentes enfocadas, cada micro lente re-imagen el objeto desde una perspectiva ligeramente diferente, lo que significa gastar la resolución espacial en comparación con una cámara convencional y obtener la información angular basada en la disparidad entre las imágenes micro vecinas.

La MLA en una cámara plenótica puede tener un arreglo ortogonal, hexagonal (con un factor de llenado mejorado) o cualquier otro posible (ver Figura 1.6). La geometría, ubicación y calidad de la MLA afectan fuertemente las propiedades de la cámara plenótica [4].

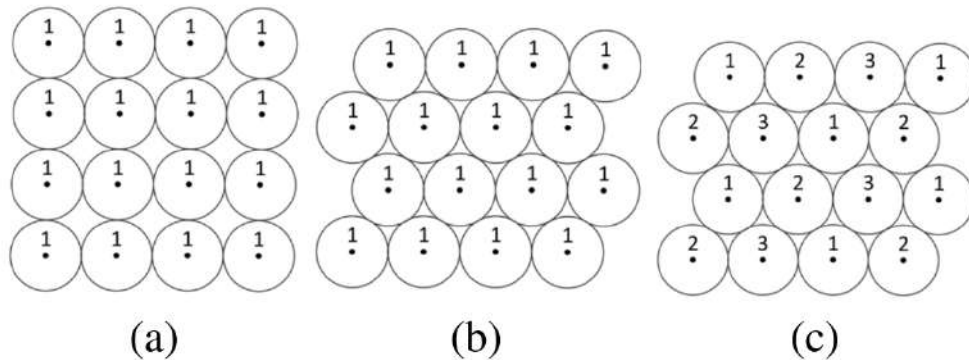


Figura 1.6: Varios arreglos MLA (a) ortogonales con solo tipo de lente (b) hexagonal, lo que proporciona un mejor factor de relleno (c) hexagonal con tres tipos de lentes, cada uno con una distancia focal diferente [4].

La cámara Raytrix R29 utiliza una matriz de micro lentes 2D que consiste en tres MLA hexagonales entrelazadas, cada una con una distancia focal diferente. Las distancias focales de las micro lentes se eligen de manera que el campo de profundidad (DoF) de diferentes tipos de micro lentes solo tocan y así extienden el DoF global de la cámara plenóptica de enfoque múltiple R29. Debido a la estructura compleja de las cámaras plenópticas en general y de las cámaras plenópticas de enfoque múltiple en particular, se utilizan suposiciones y aproximaciones simplificadas para extraer las propiedades de alto nivel de estas cámaras mediante un enfoque analítico [4].

1.5.3. Cámara basada en lentes

Una cámara plenóptica basada en lentes como la cámara Lytro es un tipo de cámara que utiliza un sistema de lente micro para capturar información direccional. Estas cámaras tienen una estructura de arreglos de micro lentes (MLA, por sus siglas en inglés) que se introduce en un sistema de cámara convencional. La lente principal genera una imagen que luego es proyectada por las lentes micro en el plano de imagen.

En una cámara convencional, la imagen que captura el sensor es nítida y enfocada, mientras que en una cámara basada en lentes (ver figura 1.7), la imagen capturada es borrosa. Esta última cámara gana información angular, que es la información relacionada con el ángulo de vista, a costa de la resolución espacial [12].

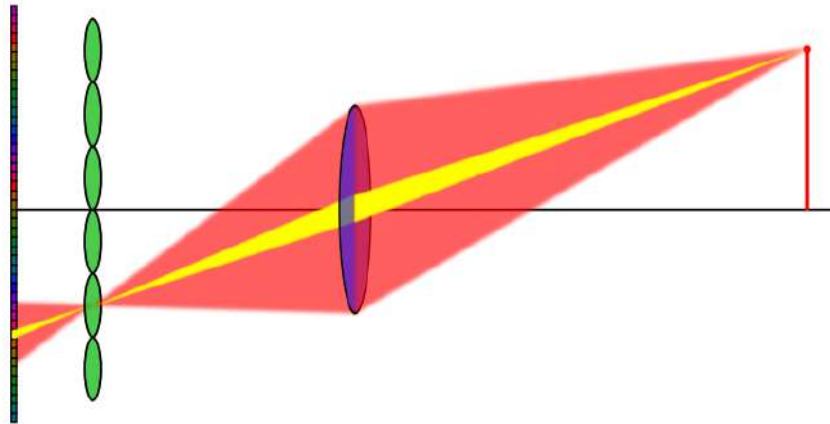


Figura 1.7: La cámara plenóptica basada en lentes: luz de un objeto en la escena (derecha) pasa a través de la lente principal (azul) y se enfoca en una matriz de lentes (verde). las lentillas dividen la luz entrante en muchos píxeles (izquierda), según la dirección de llegada. Un solo píxel es resaltado en amarillo, que ilustra el tamaño de apertura de una cámara convencional con la misma profundidad de campo [4].

El arreglo de micro lentes en una cámara plenóptica puede tener una disposición ortogonal, hexagonal u otra posible disposición. La geometría, la ubicación y la calidad del MLA afectan fuertemente las propiedades de la cámara plenóptica. Además, debido a la estructura compleja

de las cámaras plenópticas, se utilizan suposiciones y aproximaciones simplificadoras para extraer propiedades de alto nivel de estas cámaras mediante un enfoque analítico [12].

1.5.4. Cámara basada en lentes enfocadas

Las cámaras plenópticas con lentes enfocadas, son un tipo especial de cámaras que utilizan una malla de micro lentes para capturar información direccional en el campo de luz de una escena. En lugar de tener una única lente principal, estas cámaras tienen una malla de micro lentes que proyectan la imagen generada por la lente principal en la placa de imagen.

Cada micro lente en la malla proyecta una imagen ligeramente diferente, lo que permite a la cámara recopilar información angular sobre la escena. Esta información angular es esencial para el procesamiento posterior y la creación de vistas noveles. En este tipo de cámaras, cada micro lente está diseñada para enfocar en una profundidad específica en la escena. De esta manera, todas las vistas capturadas por la malla de micro lentes están enfocadas en diferentes profundidades, lo que resulta en una mayor profundidad de campo y una mayor capacidad de capturar información direccional.

La cámara Raytrix R42 es un buen ejemplo de cámara plenóptica con lentes enfocadas. Utiliza una malla de micro lentes con diferentes distancias focales para enfocar en diferentes profundidades en la escena. Después de capturar la imagen, se pueden utilizar algoritmos de procesamiento posterior para crear vistas noveles y extraer información angular sobre la escena [12].

1.5.5. Notación de Levoy

La notación de Levoy se refiere a una representación matemática y conceptual de los campos de luz en la fotografía plenóptica. Marc Levoy, un investigador en gráficos por computadora, propuso esta notación en su tesis doctoral en 1996.

La notación de Levoy describe un campo de luz como una función de cinco dimensiones, donde tres dimensiones corresponden a las coordenadas espaciales (x, y, z) y las otras dos dimensiones corresponden a las coordenadas angulares (θ, ϕ) que indican la dirección de los rayos de luz. En lugar de representar una imagen tradicional en dos dimensiones, la notación de Levoy captura información adicional sobre la dirección de los rayos de luz que llegan a la cámara.

Esta representación de cinco dimensiones permite una mayor flexibilidad en el procesamiento y análisis de la información de luz capturada por una cámara plenóptica. Al utilizar la notación de Levoy, es posible realizar operaciones como el reenfoque sintético, la generación de vistas múltiples y la extracción de información tridimensional a partir de un solo conjunto de datos plenópticos [13].

1.5.6. Visión artificial

La visión por computadora es una rama de la inteligencia artificial que le permite a un sistema reconocer y ver el mundo que lo rodea. En este campo, una imagen o video se representa como una matriz de píxeles con valores numéricos que definen sus colores. Estos valores pueden ser utilizados como características por los modelos de aprendizaje automático para procesarlos, analizarlos y extraer información significativa y lógica.

La visión por computadora tiene muchas aplicaciones prácticas, como la organización de contenido en fotos y videos, la extracción de texto de documentos, y el análisis de espacios. Por ejemplo, las redes sociales pueden utilizar la visión por computadora para identificar personas y objetos en las fotos y etiquetarlos, mientras que el procesamiento de facturas puede utilizar la tecnología para extraer información estructurada de documentos. Además, la visión por computadora puede ser utilizada para analizar un espacio y brindar información sobre las personas, objetos y movimientos en ese espacio [14].

1.5.7. OpenCV

OpenCV es una biblioteca de software de visión artificial y aprendizaje automático de código abierto que se ha desarrollado con el objetivo de proporcionar una infraestructura común para aplicaciones de visión artificial y para acelerar el uso de la percepción de la máquina en los productos comerciales. La biblioteca cuenta con más de 2500 algoritmos optimizados que incluyen un conjunto completo de algoritmos de visión por computadora y aprendizaje automático clásicos y de última generación, lo que la hace altamente versátil y adaptable a una amplia gama de aplicaciones.

Esta biblioteca se utiliza en una amplia gama de aplicaciones, desde unir imágenes de Street-view, detectar intrusiones en videos de vigilancia, monitorear equipos mineros, ayudar a los robots a navegar y recoger objetos, detección de accidentes de ahogamiento en piscinas, ejecutar arte interactivo, revisar las pistas en busca de escombros e inspeccionar etiquetas en productos en fábricas de todo el mundo.

OpenCV cuenta con interfaces C++, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS. La biblioteca se enfoca principalmente en aplicaciones de visión en tiempo real y aprovecha las instrucciones MMX y SSE cuando están disponibles. En la actualidad, se están desarrollando activamente interfaces CUDA y OpenCL con todas las funciones. Además, OpenCV está escrito de forma nativa en C++ y cuenta con una interfaz con plantillas que funciona a la perfección con contenedores STL [15].

CAPÍTULO 2

Diseño de la solución

Contenido

2.1. Obtención de los campos de luz en notación Levoy	15
2.2. Método de Evaluación usando LFToolbox	16

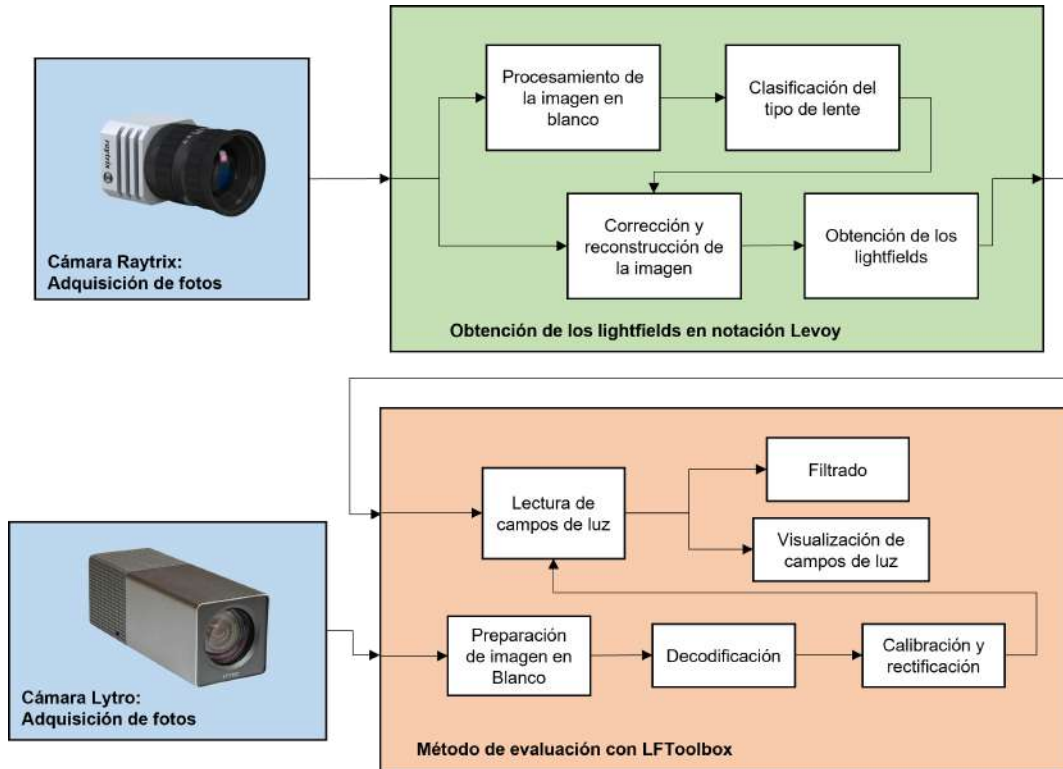


Figura 2.1: Diagrama de bloques de la solución propuesta

Como solución a la problemática, se desarrolló un método que permite la obtención de 3 distintos campos de luz a partir de una fotografía capturada por la cámara Raytrix R42, para luego ser procesada y evaluada por el Lightfield toolbox de matlab. La solución propuesta se divide en 2 bloques principales, de los cuales, se componen de bloques secundarios como se observa en la figura 2.1.

2.1. Obtención de los campos de luz en notación Levoy

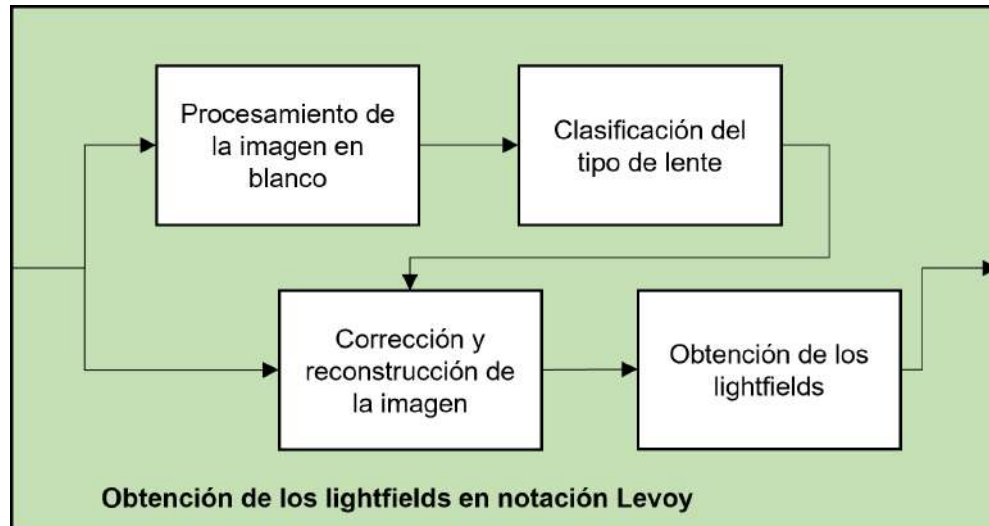


Figura 2.2: Diagrama del sub-bloque para la obtención de los campos de luz

Para poder obtener los campos de luz de la cámara Raytrix R42 se deben tener a disposición dos imágenes tomadas por esta, la primera es la imagen en blanco y la segunda es la imagen la cual se quiere procesar para hallar los campos de luz. Esto es necesario ya que el algoritmo, a partir de la imagen en blanco, toma los contornos y los centroides de los micro lentes para así poder reconstruir las vistas de los campos de luz.

Para lograr esto se implementó un algoritmo en Python que consta de 4 partes como se puede observar en la figura 2.2. El procesamiento de la imagen en blanco, ya que se debe ajustar antes de poder usarla para el análisis correspondiente, la clasificación del tipo de lente, debido a que la cámara cuenta con 3 tipos de lente distintos y esto hace que se pueda obtener un campo de luz con propiedades diferentes para cada tipo de lente, la corrección y reconstrucción de la imagen a partir de los datos obtenidos de la imagen en blanco, puesto que el algoritmo obtiene aleatoriamente el orden de los datos debido a cambios mínimos en los parámetros de la cámara a la hora de calibrarla o simples imperfecciones en los micro lentes. Y finalmente la obtención de los campos de luz para cada tipo de lente.

2.2. Método de Evaluación usando LFToolbox

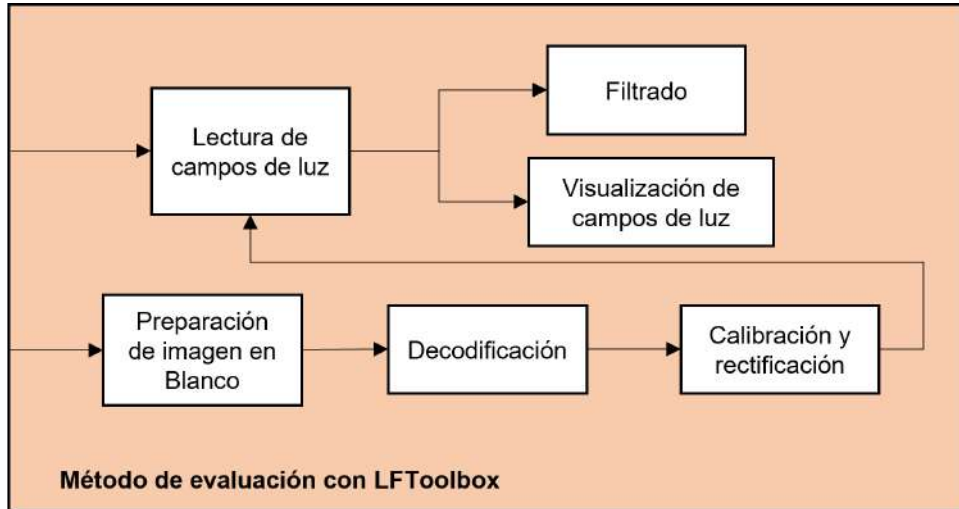


Figura 2.3: Diagrama del sub-bloque para el método de evaluación de los campos de luz

Luego de obtener los campos de luz correspondientes se creó un algoritmo usando el LFToolbox de Matlab, el cual evalúa estos campos de luz mediante herramientas de reenfoque sintético. Para esto se hizo uso de las librerías de decodificación, calibración y rectificación, debido al pre procesamiento requerido por los campos de luz obtenidos por la cámara Lytro; también se usaron librerías de lectura de campos de luz, las cuales permiten obtener lecturas de fotografías tomadas por la cámara Lytro, lecturas de campos de luz en notación de Levoy, entre otros; a su vez se usaron los distintos filtros que esta herramienta posee, como el filtro de suma y desplazamiento, el filtro de lineal 2D, el filtro plano 4D y el filtro Hyperfan 4D, los cuales serán analizados más adelante. Por último, se implementaron las librerías de visualización para diferentes perspectivas del resultado del reenfoque sintético.

Además de los anteriores métodos mencionados que son usados para el reenfoque sintético, este toolbox cuenta con otras herramientas que son útiles para el análisis y el procesamiento del campos de luz, como se puede ver en la figura 2.3.

CAPÍTULO 3

Implementación para la creación y obtención de un campo de luz

Contenido

3.1. Calibración de la cámara Raytrix R42 y adquisición de las imágenes .	18
3.2. Procesamiento de la imagen en blanco	19
3.2.1. Pre-Thresholding	19
3.2.2. Thresholding por rangos	19
3.2.3. Detección de contornos considerando la jerarquía	20
3.2.4. Detección de centroides para cada lente a partir del momento (M)	21
3.3. Clasificación para cada tipo de lente	23
3.3.1. Parámetros	23
3.3.2. Detección de contornos Padre	23
3.3.3. Clasificación por K-Means	26
3.4. Corrección para la distribución de píxeles	27
3.4.1. Agrupación de centroides en el eje y usando K-means	27
3.4.2. Reorganización en el eje x	29
3.5. Obtención de los campos de luz para cada tipo de lente	30
3.5.1. Reconstrucción de la imagen a partir de una imagen en blanco	30
3.5.2. Verificación del tamaño de la matriz usada en la reconstrucción	31

En el capítulo 2 se dio una explicación general del método creado para generar los campos de luz para cada tipo de lente, en este capítulo se da una explicación mas detallada del paso a paso y se dan a conocer los detalles que hay detrás de esta implementación; desde la calibración de la cámara y adquisición de las imágenes, pasando por el procesamiento de estas y tomando conceptos de inteligencia artificial para clasificación, hasta la obtención de los campos de luz en notación de Levoy. En el anexo 3 se puede encontrar el código fuente creado para la presente implementación en un repositorio de GitHub.

3.1. Calibración de la cámara Raytrix R42 y adquisición de las imágenes

Para poder obtener buenos resultados a la hora de crear los campos de luz usando este algoritmo, es necesario tener una buena fotografía en blanco y tener presente el enfoque de la cámara con el objeto o plano que requiere ser enfocado; además de las variables físicas como un ambiente controlado, un entorno bien iluminado (o el su defecto una linterna), entre otras. Se debe hacer una correcta calibración de la cámara para poder obtener los resultados esperados, en este caso entra en juego las variables de apertura y foco de la cámara Raytrix R42 como se observa en la figura 3.1. Para tener una vista anticipada del resultado de las fotografías capturadas y guardarlas en el ordenador se usó el programa Raytrix R42.



Figura 3.1: Variables mecánicas de apertura y foco del lente *HF-1218-12M* de la Cámara Raytrix R42

Como primer paso se ajustó el filtro físico que viene con el kit de la Cámara y con este puesto se capturaron todas las muestras de fotos en blanco, hacer este paso previo es recomendable ya que, al usar el filtro, la luz se dispersa de una manera más homogénea y a su vez esta no impacta directamente en las lentes, la apertura se ajustó de tal manera que se pudiera detallar bien cada micro lente y el foco se ajustó a una distancia de 10 centímetros, como se puede observar en la figura 3.2 respectivamente.

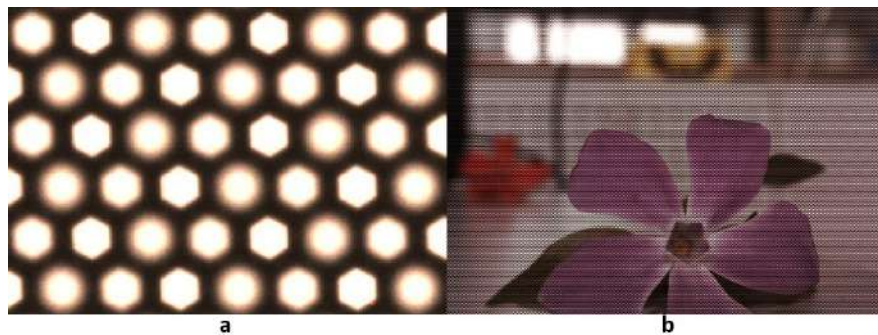


Figura 3.2: a: Muestra de calibración de la foto en blanco para la apertura de las micro lentes usada en los resultados. b: calibración de foco usado para el ejemplo, en este caso la flor está enfocada y el plano del fondo desenfocado.

3.2. Procesamiento de la imagen en blanco

Esta sección se compone del proceso implementado para modificar la imagen en blanco de tal modo que sea posible hacer una óptima clasificación de lentes y la reconstrucción de la muestra requerida, para esto se diseñó un algoritmo que detecta el centroide y los contornos en dos rangos de luminosidad de cada micro lente, esto ya que, al pretender reconstruir una imagen de la cámara plenóptica a partir de la imagen en blanco es necesario conocer el contorno del lente y su punto central dado que, en base a estos datos, el algoritmo hace la reconstrucción del campo de luz; todo esto con ayuda de la librería de OpenCV.

3.2.1. Pre-Thresholding

Luego de obtener la imagen en blanco calibrada se modificó con una función que la transforma a escala de grises como se observa en la Figura 3.3, esto principalmente por dos motivos, el primero es la reducción de dimensiones para una menor complejidad en su manejo ya que en las imágenes RGB hay tres canales de color y por este motivo poseen tres dimensiones, mientras que las imágenes en escala de grises son unidimensionales. El segundo motivo radica en la versatilidad de las funciones para realizar el Thresholding, pocas funciones permiten leer imágenes RGB y a su vez trabajan de una manera más óptima con imágenes unidimensionales

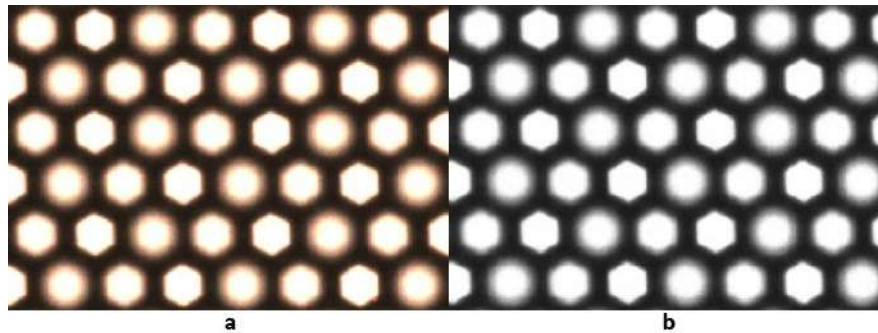


Figura 3.3: a: Muestra de la foto en blanco original. b: Muestra de la foto en blanco en escala de grises.

3.2.2. Thresholding por rangos

Para poder usar la técnica de Thresholding se debe tener en cuenta la función que más conviene para el desarrollo de la solución en el algoritmo, en este caso se usó la función *inRange* de OpenCV ya que esta cuenta con dos rangos de umbralización, lo cual es adecuado para poder hallar los contornos internos y externos de cada micro lente como se puede ver en la Figura 3.4.

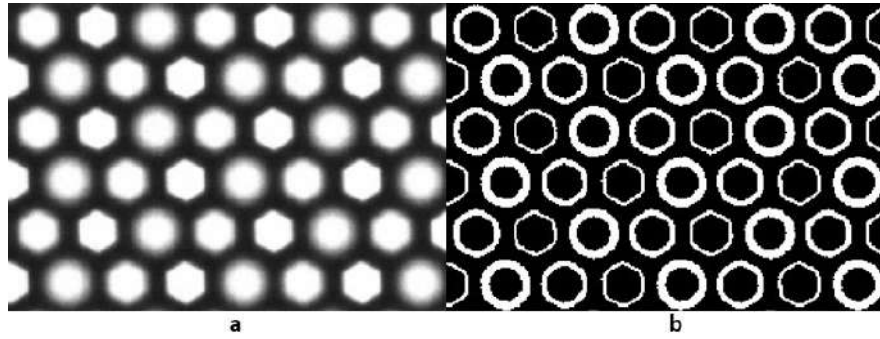


Figura 3.4: a: Muestra de la foto en escala de grises. b: Muestra de la foto al ser umbralizada por rangos.

3.2.3. Detección de contornos considerando la jerarquía

Después de tener la imagen umbralizada, se empleó la función *findContours* para la detección de los contornos dada por OpenCV; esta función cuenta con dos parámetros que se deben tener en cuenta, el primero es el modo de recuperación del contorno, el cual indica el tipo de jerarquía que la función va a retornar y el segundo parámetro es el método de aproximación de contorno que esta función usa para hallarlo, el cual indica si se requiere almacenar todos los datos del contorno o solo sus vértices.

Para esta implementación se empleó como primer parámetro *RETR_TREE* el cual hace que sea retornada una matriz de 4 dimensiones con la jerarquía de cada contorno, el primer valor de la matriz es el número del contorno que halló inmediatamente después del contorno presente, el segundo valor es el numero del contorno inmediatamente después, el tercer valor es el número del primer contorno hijo y el cuarto valor es el número del contorno padre, si alguno de los contornos no posee contorno hijo o contorno padre el valor será asignado como -1, Un ejemplo de esto se puede observar en la tabla 3.1 y en la Figura 3.5.



Figura 3.5: Muestra de los dos contornos hallados para cada tipo de micro lente. Nota: Los contornos 1, 3 y 5 son hijos de los contornos 0, 2 y 4 respectivamente y estos a su vez son sus contornos padres.

Contorno	Contorno siguiente	Contorno anterior	Contorno primer hijo	Contorno padre
0	2	-1	1	-1
1	-1	-1	-1	0
2	4	0	3	-1
3	-1	-1	-1	2
4	-1	2	5	-1
5	-1	-1	-1	4

Tabla 3.1: Ejemplo del orden jerárquico que retornaría la función para los contornos de la figura 3.5.

Como segundo parámetro se usó *CHAIN_APPROX_NONE* el cual hace que sean retornados todos los datos del contorno, sin aproximación alguna. El resultado de los contornos hallados se pueden observar en la figura 3.6.

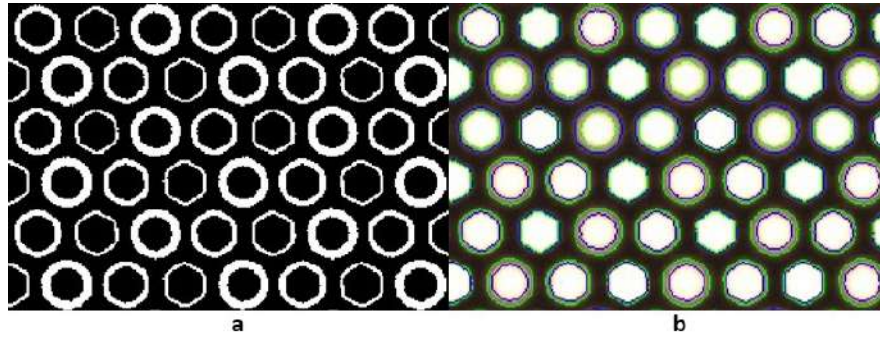


Figura 3.6: a: Muestra de la foto al ser umbralizada por rangos. b: Muestra de la foto original con los contornos dobles hallados para cada micro lente.

3.2.4. Detección de centroides para cada lente a partir del momento (M)

Como ya se mencionó, al querer reconstruir una imagen de la cámara plenóptica a partir de la imagen en blanco, también es necesario conocer el punto central de cada micro lente; para esto se usó la función *moments* la cual proporciona un diccionario de los valores de momento calculados, en este caso, para un contorno en específico.

En esta implementación solo se tuvieron en cuenta los 3 primeros valores del momento ya que, con dichos valores, es posible calcular las coordenadas centrales para cualquier contorno de micro lente. Por otra parte, solo se tuvieron en cuenta los contornos hijos para esta operación ya que únicamente es necesario conocer las coordenadas del centroide de un contorno de los dos contornos que posee cada micro lente y, al ser el contorno que más abarca el espacio útil del lente, es el de mayor utilidad de los dos contornos.

Esta función trabaja con las coordenadas de cada píxel y su intensidad, para hallar el primer momento M_{00} , la función calcula una sumatoria doble del producto de las coordenadas por la intensidad de cada píxel del contorno 3.1.

$$M_{00} = \sum_x \sum_y I(x, y) \quad (3.1)$$

Para hallar el segundo M_{01} y tercer momento M_{10} , la función calcula una sumatoria doble de la coordenada y y la coordenada x respectivamente, multiplicada por el producto de las coordenadas por la intensidad de cada píxel del contorno; o, en otras palabras, la coordenada y para el segundo momento y la coordenada x para el tercer momento por los valores que se hallaron en el primer momento M_{00} , esto para cada coordenada 3.2.

$$M_{01} = \sum_x \sum_y xI(x, y) \quad M_{10} = \sum_x \sum_y yI(x, y) \quad (3.2)$$

Luego de tener los 3 valores del momento para un contorno, se divide el segundo M_{01} y tercer momento M_{10} por el primer momento M_{00} para hallar la coordenada en y y x del centroide respectivamente 3.3. En el algoritmo, el proceso se realizó para todos los contornos que detectó la función *findContours*, dando como resultado dos vectores emparejados donde se encuentran las coordenadas de todos los centroides, en la figura 3.7 se puede encontrar una visualización física del resultado de este proceso.

$$x_{centroide} = \frac{M_{10}}{M_{00}} \quad y_{centroide} = \frac{M_{01}}{M_{00}} \quad (3.3)$$

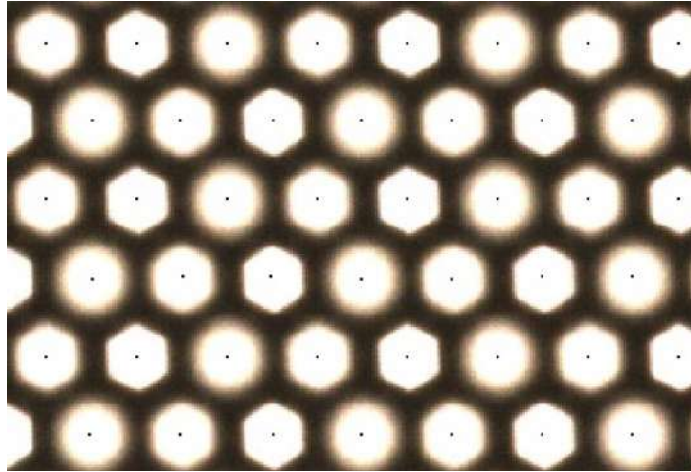


Figura 3.7: Muestra de la foto en blanco con los centroides marcados para cada micro lente.

3.3. Clasificación para cada tipo de lente

En esta sección se da a conocer el proceso de implementación diseñado para crear un clasificador de lentes para imágenes en blanco obtenidas por la cámara Raytrix R42, para esto se usaron un par de parámetros los cuales son el perímetro de los contornos y la diferencia entre las áreas del contorno padre con el contorno hijo de cada micro lente, luego se identificaron los contornos padre y únicamente se tuvieron en cuenta los contornos hijos, en base a estos se hicieron los cálculos de los parámetros, finalmente se ejecuta el algoritmo de clasificación no supervisada K-Means.

3.3.1. Parámetros

Como se mencionó anteriormente, los parámetros usados para la clasificación del K-Means fueron el perímetro de los contornos y la diferencia entre las áreas del contorno padre con el contorno hijo de cada micro lente. La justificación de este diseño se debe a que la mayor diferencia entre los tipos de lentes está ubicada en la zona fuera del enfoque de cada lente como se muestra en la figura 3.8. Por este motivo se hallaron las dos áreas de cada micro lente y se calculó la diferencia. Ya sabiendo que estas dos características tienen el mayor cambio con respecto a cada lente se convirtieron en las mejores características para una buena clasificación.

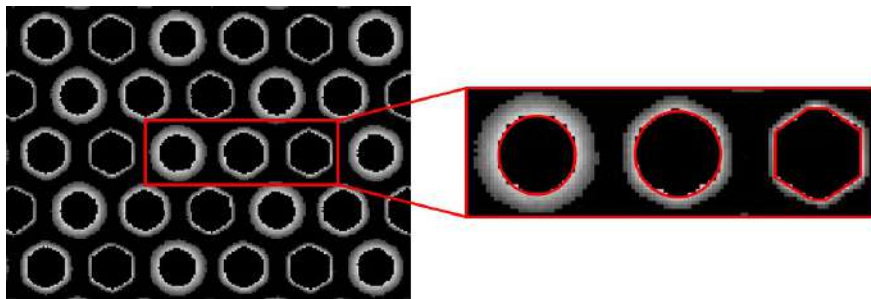


Figura 3.8: Muestra de una foto en blanco que enseña únicamente las zonas fuera de los enfoques de los lentes en escala de grises y a la derecha se presenta un acercamiento de los tres tipos de lentes señalando el perímetro usado.

3.3.2. Detección de contornos Padre

Antes de pasar a la clasificación en sí, se implementó una secuencia que detectaba solo los contornos padres y en base a estos se calculaban las diferencias de áreas con los contornos hijos. Esta implementación se hizo ya que solo era necesario tener un contorno por lente y al tiempo, este proceso reducía el número de contornos a la mitad, disminuyendo así el número de iteraciones necesarias para futuros re-ordenamientos de los arreglos donde se guardaron estos resultados. A

su vez, se implementaron dos secuencias que filtraban datos basura o ruido que, en este caso, se alejaban mucho de los valores más comunes, así como se muestra en la figura 3.9.

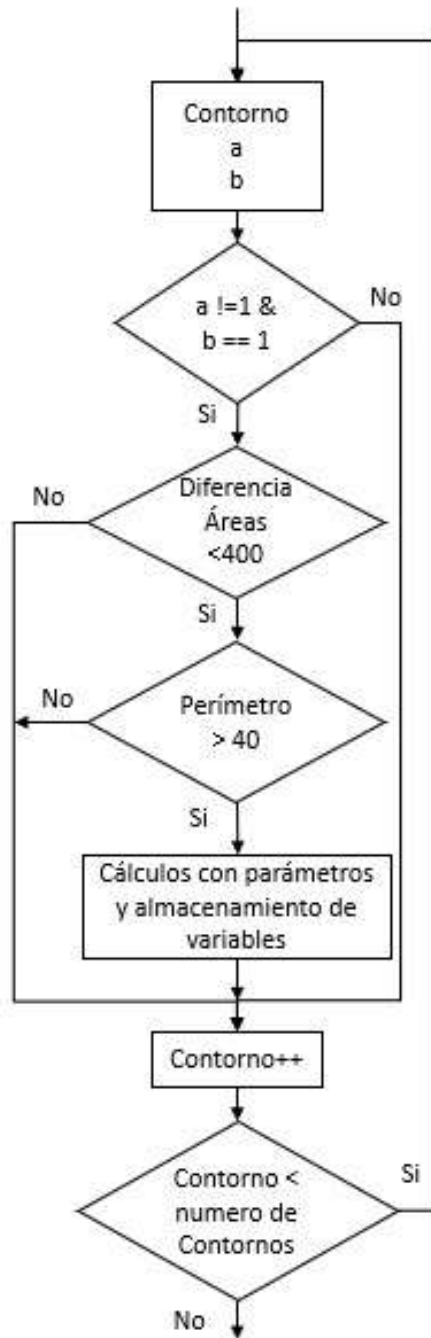


Figura 3.9: Diagrama de flujo del método implementado para la detección de contornos padre y filtrado de datos basura o ruido tomando como ejemplo para los datos de entrada la tabla 3.1. Siendo la variable *contorno* todos los contornos calculados, por otra parte, la variable *a* es el contorno primer hijo y la variable *b* es el contorno padre, esto para cada contorno procesado.

Este método implementado para la detección de contornos padres se basa en confirmar el tercer y cuarto valor de la jerarquía de cada contorno, ya que, si el tercer valor es diferente de -1 y el cuarto valor es igual a -1 se afirma que el actual contorno verificado es un contorno padre. Posteriormente, para todos los cálculos se toma el tercer valor de la jerarquía como el número de contorno hijo para este contorno padre y se guardan los parámetros calculados mas un nuevo arreglo que guarda el contorno hijo. Si el tercer valor de la jerarquía es igual a -1 o el cuarto valor es diferente de -1 se ignora y el algoritmo pasa al siguiente contorno para volver a procesarlo hasta pasar por todos los contornos detectados, como se puede observar en el algoritmo 1.

Algoritmo 1 Algoritmo para el método implementado en la detección de contornos padre; donde la entrada son todos los contornos calculados con su respectiva jerarquía y n es el número total de contornos calculados.

Require: *Contornos*

Require: *Jerarquia*

```

1: for  $i \leftarrow 0, n$  do
2:   if ( $Jerarquia\_Contorno\_hijo_i \neq -1$ ) &
      ( $Jerarquia\_Contorno\_padre_i == -1$ ) then
3:      $Diferencia\_Areas \leftarrow Area(Contorno\_padre_i) - Area(Contorno\_hijo_i)$ 
4:      $Perimetro \leftarrow Perimetro(Contorno\_hijo_i)$ 
5:      $Nuevos\_contornos\_hijos \leftarrow Contorno\_hijo_i$ 
6:   end if
7: end for

```

Finalmente, la implementación de los dos filtros de datos basura o ruido se basa en un rango superior y uno inferior que acotan los limites superiores e inferiores dando así resultados sin valores mayores o menores de estos rangos respectivamente, como se aprecia en el algoritmo 2.

Algoritmo 2 Algoritmo para el método implementado en el filtrado de datos basura o ruido; donde n es el número total de contornos calculados.

Require: *Contornos*

```

1: for  $i \leftarrow 0, n$  do
2:   if ( $Diferencia\_Areas_i < 400$ ) then
3:     if ( $Perimetro_i > 40$ ) then
4:        $Diferencia\_Areas \leftarrow Area(Contorno\_padre_i) - Area(Contorno\_hijo_i)$ 
5:        $Perimetro \leftarrow Perimetro(Contorno\_hijo_i)$ 
6:        $Nuevos\_contornos\_hijos \leftarrow Contorno\_hijo_i$ 
7:     end if
8:   end if
9: end for

```

3.3.3. Clasificación por K-Means

Como último proceso de la clasificación para cada tipo de lente se llevó a cabo el diseño del algoritmo de clasificación no supervisada K-Means. Figura 3.10.

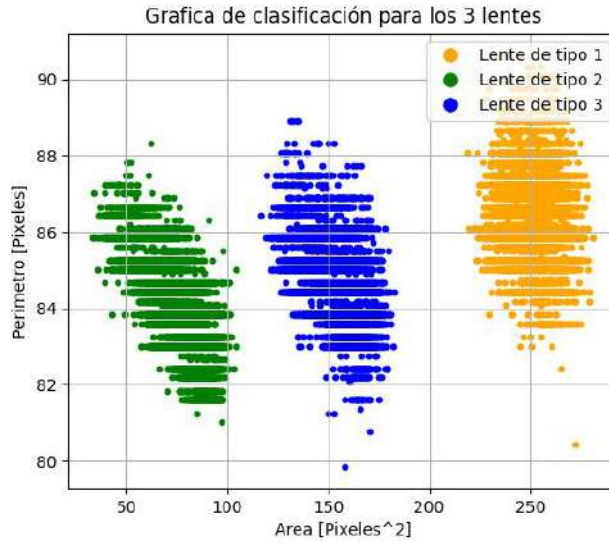


Figura 3.10: Gráfica de los 3 clústers generados dando así la clasificación para cada lente.

El cual se basa en generar $k = 3$ clústers tomando los parámetros propuestos anteriormente. Dando como resultado la gráfica de la figura 3.10. Como se puede examinar, en la leyenda de la gráfica, el Lente de tipo 1 es el lente con menor área de enfoque, el Lente de tipo 2 es el lente con mayor área de enfoque y el Lente de tipo 3 es el lente con el área de enfoque medio. A cada tipo de lente se le asigna una etiqueta para poder distinguirlos y separarlos, teniendo hecha la clasificación se puede pasar a la corrección de píxeles y luego la reconstrucción de la imagen que se explica en las siguientes secciones. En la figura 3.11. Se puede visualizar una muestra del resultado final de esta clasificación.

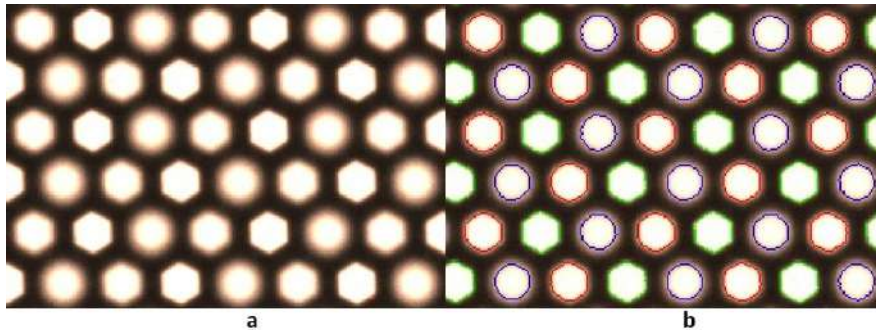


Figura 3.11: a: Muestra de la foto en blanco original. b: Muestra de la foto en blanco luego de ser clasificado cada micro lente, esta resalta los contornos en tres distintos colores dependiendo del tipo de lente que sea.

3.4. Corrección para la distribución de píxeles

Al tener guardados los contornos procesados y sus centroides se implementó una corrección para la distribución de estos datos dividida en una clasificación con K-Means y un algoritmo de reorganización por números de menor a mayor teniendo en cuenta la clasificación mencionada anteriormente, esto debido a la aleatoriedad con la que se hallan los contornos puesto que estos dependen de la calidad de la foto en blanco y los defectos físicos de la cámara plenóptica con la que fue tomada esta foto, como por ejemplo, algunos lentes con imperfecciones debido a un mal manejo de la cámara como se observa en la figura 3.12.

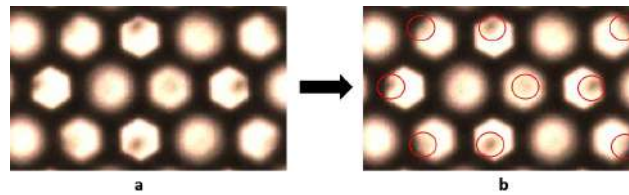


Figura 3.12: a: Muestra de la foto en blanco original donde se observan imperfecciones. b: Muestra de la foto en blanco con las imperfecciones que pueden afectar el algoritmo señaladas.

3.4.1. Agrupación de centroides en el eje y usando K-means

Para dar solución a este problema se diseñó un algoritmo de reorganización para los contornos con sus respectivos centroides, esto con ayuda del algoritmo K-Means con un $K = 176$ dado que es el numero de filas en la matriz de los micro lentes.

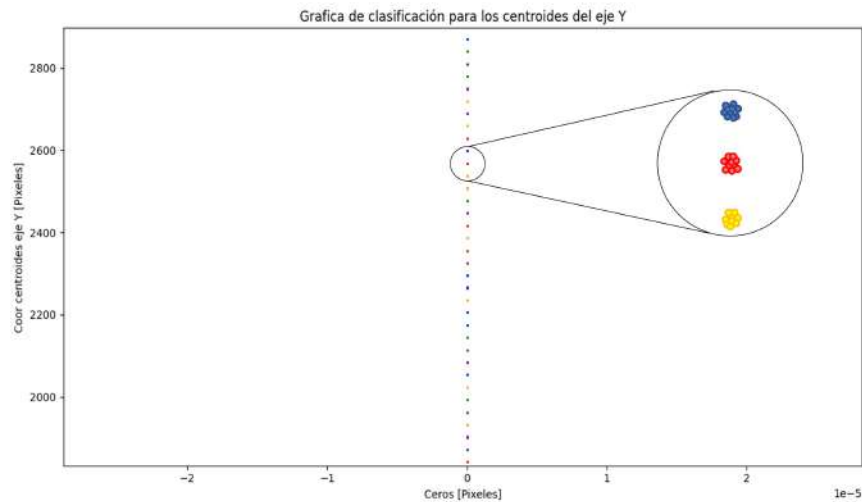


Figura 3.13: Acercamiento de la gráfica de los 176 clústers generados para clasificar cada fila de la matriz de contornos, cada punto hace referencia a un clúster en el cual se agrupa las coordenadas de los puntos centrales de cada contorno.

El algoritmo se implementó para esta solución, ya que, al ser aleatoria la cantidad de micro lentes bien detectados, se debe tener presente cuantos micro lentes se detectaron por fila. En la figura 3.13. Se puede observar la gráfica de los clústers detectados.

En la figura 3.14. se observan los pasos, en forma de diagrama de flujo, que se siguieron para lograr esto. El algoritmo consiste en una búsqueda del menor valor en todo el arreglo y lo guarda, cuando este termina de buscar, crea un intercambio de valores entre el menor valor que encuentra y el valor actual, finalmente cambia al siguiente valor, lo toma como el valor actual y repite el proceso hasta terminar con todos los datos.

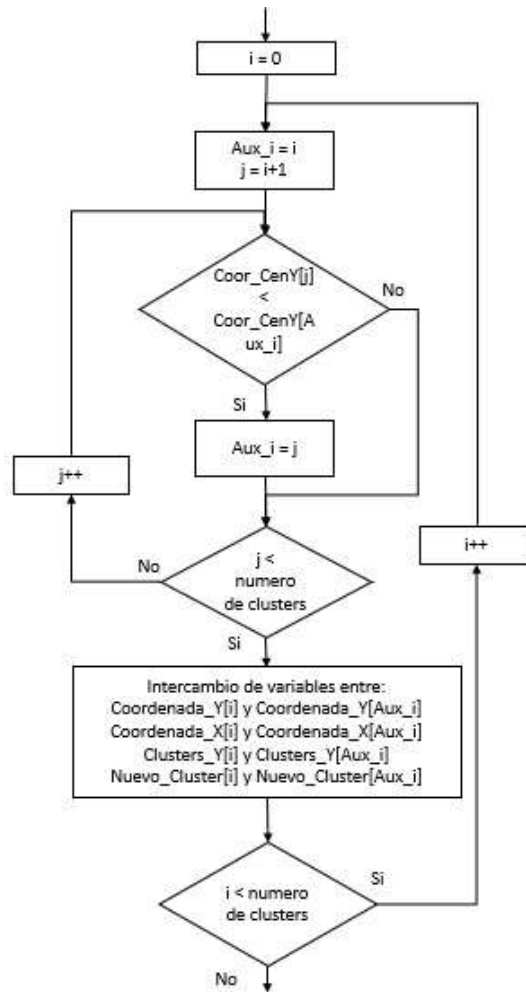


Figura 3.14: Diagrama de flujo para la organización del eje Y.

Finalmente, para la reorganización en el eje X se tomó la misma base del algoritmo anterior, pero con dos importantes modificaciones como se evidencia en la figura 3.15. La primera, se incorporó un bucle situado antes de entrar al bucle que reorganiza los arreglos, este busca los cambios que surgen en el arreglo de los clústers en Y y al producirse ese cambio, el algoritmo cuenta con el número de ciclos producidos antes del cambio, esto se hace para poder saber el número de contornos detectados

para esa fila y poder así organizar los arreglos de las coordenadas de los centroides. El segundo cambio es un bucle que cuenta y guarda la posición del limite anterior, esto para todo el arreglo de centroides y así poder retomar donde quedo el arreglo en la siguiente pasada por este bucle.

3.4.2. Reorganización en el eje x

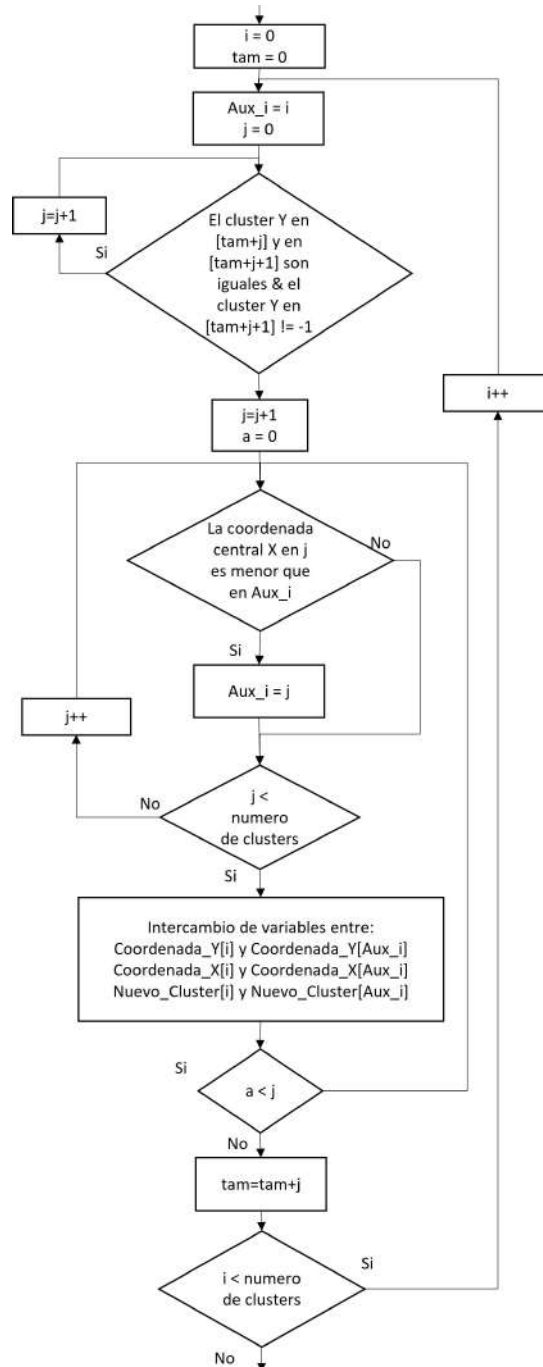


Figura 3.15: Diagrama de flujo para la organización del eje X.

3.5. Obtención de los campos de luz para cada tipo de lente

En esta sección final, para poder pasar de la fotografía original a la reconstrucción de los 3 tipos de imágenes como se muestra en la figura 3.16. Se da a conocer el algoritmo implementado para asignar la etiqueta de cada tipo de lente en su respectiva carpeta, la reconstrucción de la foto sobre imágenes en blanco, el tamaño que debe tener la matriz que se usa para hacer la creación del campo de luz, una rectificación a las imágenes finales y el proceso de guardado, esto para cada tipo de lente.

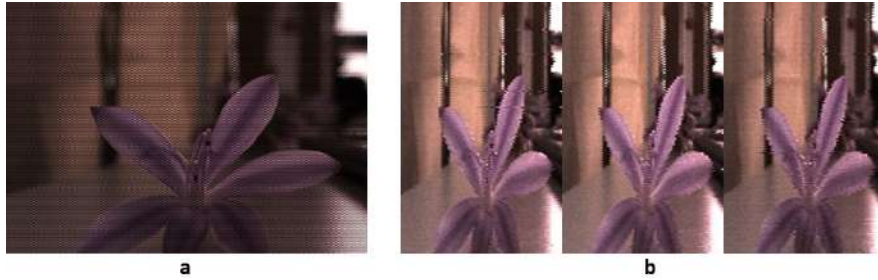


Figura 3.16: a: Fotografía original sin procesar. b: imágenes reconstruidas a partir de los 3 tipos de lentes

Al querer realizar la asignación de etiqueta dependiendo del tipo de lente para cada campo de luz se implementó el mismo algoritmo de reorganización de menor a mayor de la figura 3.14. Sin embargo, primero se hallaron los 3 valores de la diferencia de áreas con su respectivo clúster para luego organizar los valores del área de menor a mayor y así poder saber el orden de los clusters para cada tipo de lente.

3.5.1. Reconstrucción de la imagen a partir de una imagen en blanco

Pasando al proceso de reconstrucción, se crearon 3 imágenes en blanco de 175×75 para dividir la imagen original en las 3 imágenes por cada tipo de micro lente. Luego, se implementó un bucle de búsqueda para identificar el punto en el que se hace un cambio de fila en la matriz de centroides ya organizados, esto debido a que es necesario saber la cantidad de lentes detectados en cada fila para poder hacer la reconstrucción de las imágenes en orden. Posteriormente, se realizó la reconstrucción píxel a píxel, la cual consistió en pasar por cada fila y columna de las imágenes en blanco dependiendo del posicionamiento del píxel a asignar de los lentes de la imagen original, esto para cada tipo de lente. Finalmente, se realizó el almacenamiento de cada imagen en su respectiva carpeta, como se puede visualizar en la figura 3.17.



Figura 3.17: Mosaico del campo de luz completo y creación de carpetas para almacenamiento.

3.5.2. Verificación del tamaño de la matriz usada en la reconstrucción

Algoritmo 3 Algoritmo implementado para la verificación del tamaño de la matriz.

Require: *Contornos*

Require: *Arreglo_X*

Require: *Arreglo_Y*

Ensure: *Tamano_Matriz*

```

1: for  $i \leftarrow 0, 50$  do
2:    $Conteo \leftarrow 0$ 
3:    $Salir \leftarrow 0$ 
4:    $N\_Random \leftarrow N\_Random\_del\_0\_al\_38720$ 
5:   for  $a \leftarrow 0, 20$  do
6:     for  $j \leftarrow 0, Tamano\_Contorno\_Random$  do
7:       if  $Vertices\_de\_la\_matriz == Contorno$  then
8:          $Salir \leftarrow 1$ 
9:       end if
10:    end for
11:    if  $Salir \neq 1$  then
12:       $Conteo \leftarrow Conteo + 1$ 
13:    end if
14:  end for
15:   $Tamano\_Matriz_i \leftarrow Conteo$ 
16: end for

```

Con respecto al tamaño de la matriz usada para definir las dimensiones del mosaico completo del campo de luz, se diseñó teniendo en cuenta la cantidad de píxeles, en un perímetro cuadrado, que no sobresalían del contorno de cada lente. En este caso, se usó una matriz de 15x15 luego de recibir este resultado gracias al algoritmo implementado para definir el tamaño máximo de la matriz a usar, como se puede observar en el algoritmo 3.

Este algoritmo se encarga de tomar 50 contornos aleatorios y medir la matriz cuadrada máxima que se puede usar dentro de cada uno sin que esta sobrepase cualquier punto del contorno, dando como resultado un arreglo de números que se pueden usar como el tamaño para la matriz final. Sin embargo, el algoritmo se encarga de reorganizar el arreglo devolviendo como resultado el valor menor para poder usarlo como tamaño de la matriz para cualquiera de los 3 tipos de micro lentes como se ve en la figura 3.18.

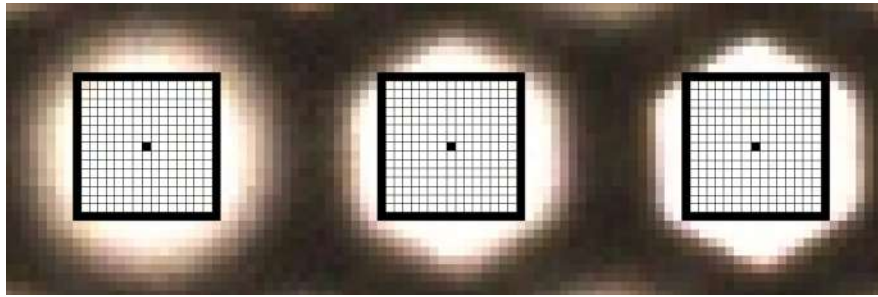


Figura 3.18: Matriz 15x15 usada para la reconstrucción del campo de luz de cada micro lente.

Si se usa una matriz mayor a la recomendada por el algoritmo, los píxeles tomados para la reconstrucción tenderán a salir de los lentes, esto implica un oscurecimiento progresivo en las imágenes del campo de luz hasta tener imágenes completamente en negro como se visualiza en la figura 3.19. Estas darán resultados desfavorables a la hora de pasar por el método de evaluación creado en Matlab.

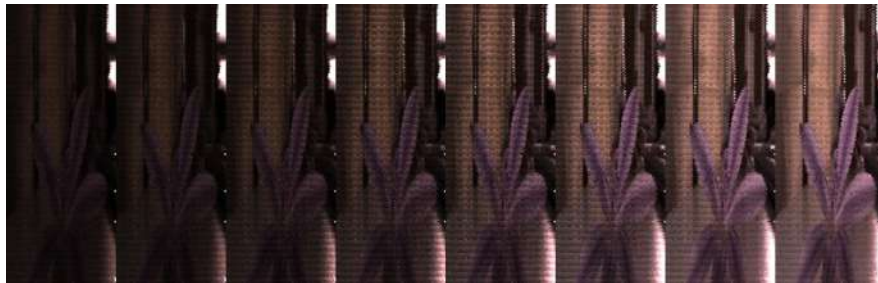


Figura 3.19: Ejemplo del campo de luz al usar una matriz mayor a la recomendada por el algoritmo.

CAPÍTULO 4

Implementación del método de evaluación usando LFToolbox

Contenido

4.1. Preparación de imagen en Blanco para imágenes Lytro	34
4.2. Decodificación para imágenes Lytro	34
4.3. Calibración y Rectificación para imágenes Lytro	36
4.4. Lectura de campos de luz	37
4.4.1. Lectura de campos de luz Lytro	38
4.4.2. Lectura de campos de luz Raytrix R42	38
4.5. Visualización de campos de luz	39
4.6. Filtrado	41
4.6.1. Filtro de suma y desplazamiento	41
4.6.2. Filtro lineal 2D	42
4.6.3. Filtro plano 4D	42
4.6.4. Filtro Hyperfan 4D	43

Al tener los campos de luz creados en la implementación demostrada en el capítulo 3, estos ya se pueden trabajar con varios métodos dependiendo del resultado que se requiera; en el caso de este trabajo de grado, se implementó un algoritmo para evaluar los campos de luz usando las librerías de Lightfield Toolbox de Matlab, orientado en particular hacia las herramientas de reenfoque sintético de esta Toolbox.

Este capítulo expone el desarrollo del método de evaluación, tanto para el manejo de fotografías tomadas por la cámara Lytro, desde la creación de su imagen en blanco, pasando por la decodificación, calibración y rectificación del campo de luz, hasta su respectivo filtrado y visualización; como para los campos de luz obtenidos en la implementación del capítulo 3 desarrollada para la cámara Raytrix R42, desde la lectura del campo de luz, hasta los distintos filtrados y sus respectivas visualizaciones.

En el anexo 4 se puede encontrar el código fuente creado para la presente implementación en un repositorio de GitHub.

En el anexo 5 se puede encontrar el link al manual para el manejo de las librerías de LFToolbox, se recomienda leer primero antes de continuar con la presente sección.

4.1. Preparación de imagen en Blanco para imágenes Lytro

Como primer paso se tuvo que preparar las imágenes en blanco para poder procesar las las fotografías tomadas por la cámara Lytro, para esto fue necesario extraer las imágenes en blanco de los archivos Lytro debido a que, al usar las imágenes de una cámara Lytro por primera vez, las carpetas del Toolbox no cuentan con los archivos *.raw* y *.TXT* donde son leídas las imágenes en blanco con las que cuenta la cámara y la calibración de esta. Esto se logró usando la función dada por el Toolbox *LFUtilUnpackLytroArchive*, ya que extrae las imágenes en blanco y otros archivos de un archivo Lytro de varios volúmenes. Estos archivos Lytro suelen adoptar nombres como *data.C.0*, *data.C.1*, etc. Los cuales contienen imágenes en blanco, metadatos y otra información.

Una vez lograda la extracción de estos archivos, se generó la base de datos de imágenes en blanco usando la función *LFUtilProcessWhiteImages*, la cual genera un modelo de cuadrícula de lentejas para cada una de las imágenes en blanco; luego, los modelos de cuadrícula son guardados como **.grid.json*. Y la base de datos de imágenes en blanco es guardada como *WhiteFileDatabase.mat*, esta base de datos se utiliza para seleccionar la imagen en blanco adecuada para decodificar cada campo de luz. Este modelo de cuadrícula se puede observar en la figura 4.1. La cual muestra un subconjunto de 5 imágenes, una para cada esquina de la cuadrícula y otra para la parte central de la cuadrícula.

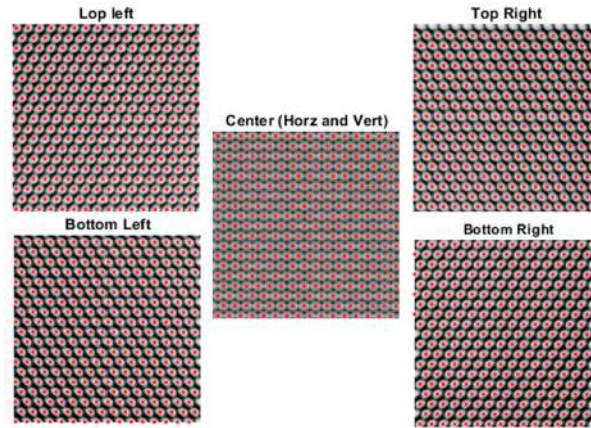


Figura 4.1: Modelo de cuadrícula dado por LFToolbox donde se exponen las 4 esquinas de la imagen en blanco creada y su centro.

4.2. Decodificación para imágenes Lytro

El Toolbox puede decodificar imágenes de campo de luz basadas en lentes en una estructura de campo de luz 4D para poder ser leídos y manejados por las diferentes librerías, esto debido a que la aplicación para exportar imágenes de la cámara Lytro descarga las fotos como campos de luz en

formato *.lfr* el cual significa que es un archivo sin procesar o en crudo y requieren este paso previo para poder tener la estructura de un campo de luz 4D. La función que se usó para la decodificación fue *LFUtilDecodeLytroFolder*, la cual, para cada imagen a decodificar, selecciona una imagen blanca apropiada para esa imagen según el número de serie de la cámara y los ajustes de zoom y enfoque. Para esto, La imagen blanca y la imagen de la lente sin procesar se pasan por la función principal de decodificación que construye el campo de luz 4D. luego, la función *LFSelectFromDatabase* selecciona la imagen blanca apropiada para el campo de luz y es utilizada para la respectiva decodificación, esto implica eliminar viñetas, transformar y cortar la imagen de la lente de entrada para producir una estructura 4D, el proceso visual de esta función se puede ver en la figura 4.2.

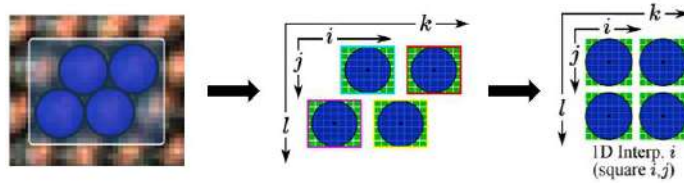


Figura 4.2: Proceso de decodificación de una imagen 2D

Posteriormente se corrió nuevamente la función *LFUtilDecodeLytroFolder* pero aplicando la corrección de color, este es un paso opcional pero recomendable a la hora de decodificar estos tipos de datos como se observa en la figura 4.3. Este proceso aplica la información que se encuentra en los metadatos del campo de luz. La secuencia de comandos realiza un seguimiento de las operaciones que se han aplicado a cada campo de luz, por lo que no repetirá el proceso de decodificación, sino que esta carga cada campo de luz ya decodificado, opera sobre él y lo sobrescribe con el campo de luz corregido por color.



Figura 4.3: Fotografías antes y después de la corrección de color.

4.3. Calibración y Rectificación para imágenes Lytro

Para este apartado se usaron las funciones *LFUtilCalLensletCam*, *LFUtilProcessCalibrations* y *LFUtilDecodeLytroFolder* con la opción de rectificación activada.

Antes de usar la primera función, como paso extra, se corrió la función de decodificación *LFUtilDecodeLytroFolder* vista anteriormente, decodificando así las fotos del patrón de ajedrez. esto debido a que, al ser la primera vez que se hacia la calibración y rectificación de la cámara con dichas fotos del patrón, estas estaban en un formato bruto y no era posible el manejo de estas fotos en Matlab. Al tener las fotos decodificadas, se empleó la función *LFUtilCalLensletCam*, en la cual se usaron las opciones de entrada *ExpectedCheckerSize* y *ExpectedCheckerSpacing-m*; la función principal identifica las imágenes de campo de luz decodificadas, en este caso todas las imágenes del patrón de ajedrez y, en base a estas, realiza la calibración dependiendo de las opciones de entrada como se observa en la figura 4.4.

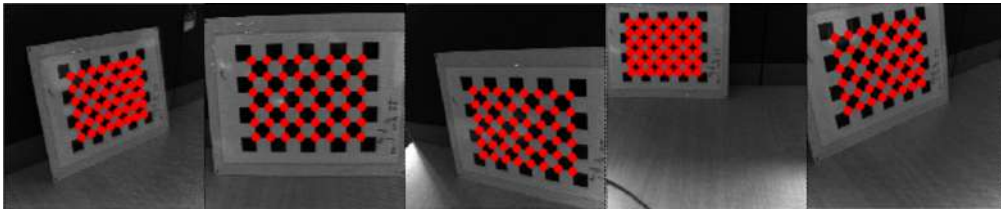


Figura 4.4: Puntos hallados en el set de fotografías del patrón de ajedrez.

La opción de entrada *ExpectedCheckerSize* define la cantidad de esquinas que hay en una matriz de dos dimensiones, excluyendo las esquinas de los bordes, en este caso se usó una matriz de [8,6]. La segunda opción de entrada *ExpectedCheckerSpacing-m* define el espaciado, en milímetros, que hay entre los cuadros del tablero de ajedrez, para esta calibración se usaron unas medidas de $1e-3 \cdot [35.1, 35.0]$. 4.5.

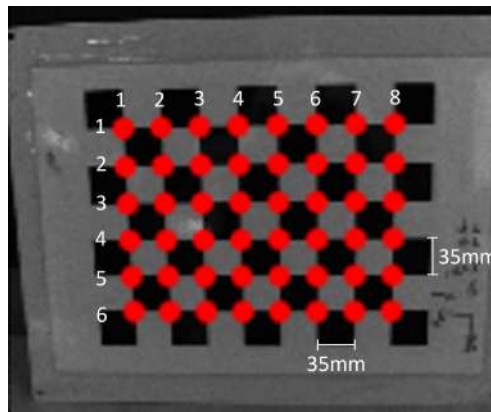


Figura 4.5: muestra del patrón con las medidas de entrada pedidas por la función.

La segunda función que se empleó fue *LFUtilProcessCalibrations*, esta crea un archivo llamado *CalibrationDatabase.mat* que permite la selección de la calibración apropiada para cada campo de luz y, al tener rectificado el campo de luz, este se sobrescribe en el archivo del campo de luz decodificado para que el código de decodificación no repita las rectificaciones ya completadas. Finalmente, se ejecutó *LFUtilDecodeLytroFolder* con la opción de rectificación activada para emplear la calibración creada anteriormente en las imágenes requeridas. En la figura 4.6 se observan dos ejemplos de calibración y rectificación.

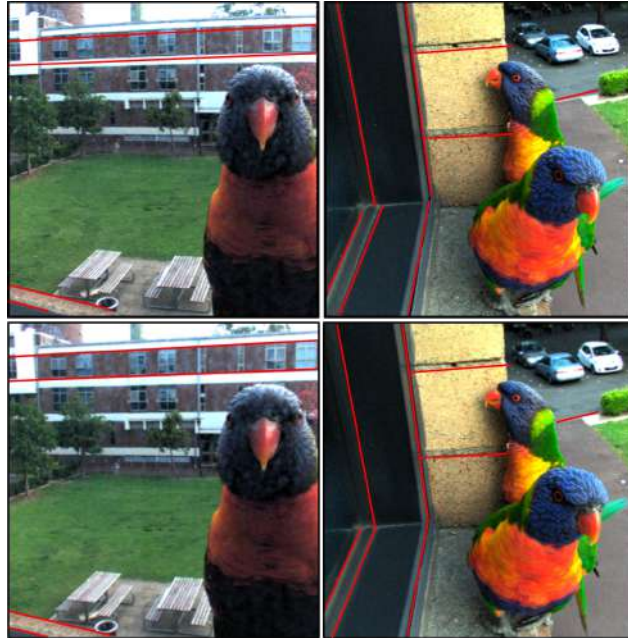


Figura 4.6: Fotografías antes y después de la calibración y rectificación, donde se puede observar la corrección de algunas rectas en las imágenes.

4.4. Lectura de campos de luz

Los campos de luz pueden venir en formatos variados dependiendo de la notación con la que se haya creado este y los datos que lo conformen para su creación, Esta Toolbox permite leer campos de luz en formato *.lfr .lfp .raw* con su respectiva metadata (estos para ser decodificados), archivos *ESLF* y campos de luz en notación de Levoy o estilo gantry. Para esta sección se implementó la lectura de campos de luz en formato *.lfr* y en notación de Levoy, ya que son los archivos de salida de la cámara Lytro y Raytrix R42 respectivamente, luego de su respectivo proceso.

4.4.1. Lectura de campos de luz Lytro

Para la lectura del campo de luz de la cámara Lytro se deben tener dos archivos decodificados, (calibrados y rectificados si es el caso) por las funciones anteriormente implementadas, el primero es una base de datos en formato de MATLAB que guarda todas las variables que requiere el Toolbox y el segundo es una imagen común para verificar los cambios. Estos archivos fueron cargados con la función *load* y almacenados en la variable LF la cual ya puede ser manejada por toda la librería. En la figura 4.7 se muestra una porción del campo de luz cargado.



Figura 4.7: Ejemplo de la lectura de un campo de luz implementada para cámaras Lytro.

4.4.2. Lectura de campos de luz Raytrix R42

Para la lectura del campo de luz de la cámara Raytrix R42 se debe tener la carpeta con los archivos en notación de Levoy que fueron creados en la implementación del capítulo 3, esta carpeta fue cargada con la función *LFReadGantryArray* en la cual se usaron las opciones *UVLimit*, *STSize* y *FnamePattern*. La primera opción ajusta la escala para garantizar una dimensión de imagen máxima, esto usualmente se hace cuando las imágenes de la carpeta tienen un tamaño mayor a 256×256 ; la segunda opción ajusta el tamaño de la matriz de imágenes o mosaico y la tercera opción ajusta el formato de las imágenes de entrada. Para estas opciones se asignaron los siguientes datos: 256, [15, 15] y *.jpg* respectivamente, el resultado de esta lectura se puede visualizar en la figura 4.8.

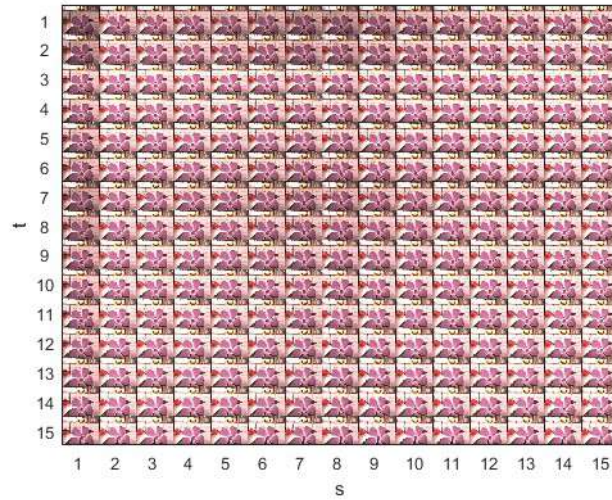


Figura 4.8: Ejemplo de la lectura de un campo de luz implementada para cámaras Raytrix R42.

4.5. Visualización de campos de luz

Antes de pasar al filtrado es una buena practica poder visualizar los campos de luz para tener la seguridad de que están correctamente creados e implementados en el toolbox o simplemente por alguna necesidad externa, para esto existen algunas herramientas de visualización de campos de luz de las cuales 4 fueron usadas en esta implementación.

Visualización por *LFDisp* Esta función muestra un corte 2D en la imagen más central que se toma en s y t de un campo de luz, esta también funciona con matrices de imágenes en 3D, figura 4.9. Si se incluye un argumento de salida, no se genera ninguna visualización, pero en su lugar se devuelve el segmento extraído.



Figura 4.9: Visualización de los campos de luz creado con la función *LFDisp*. a: Campo de luz creado por la cámara Lytro. b: Campo de luz creado por la cámara Raytrix R42.

Visualización por $LFDispTiles(stuv)$ La segunda función muestra un mosaico de un campo de luz 4D en una matriz 2D de cortes 2D con recortes en $[u, v]$ en un mosaico en $[s, t]$, figura 4.10.



Figura 4.10: Visualización de los campos de luz creado con la función $LFDispTiles(stuv)$ tipo mosaico. a: Campo de luz creado por la cámara Lytro. b: Campo de luz creado por la cámara Raytrix R42.

Visualización por $LFDispTiles(uvst)$ La tercera función también muestra un mosaico de un campo de luz 4D en una matriz 2D de cortes 2D como la segunda función pero con recortes en $[s, t]$ en un mosaico en $[u, v]$, figura 4.11.

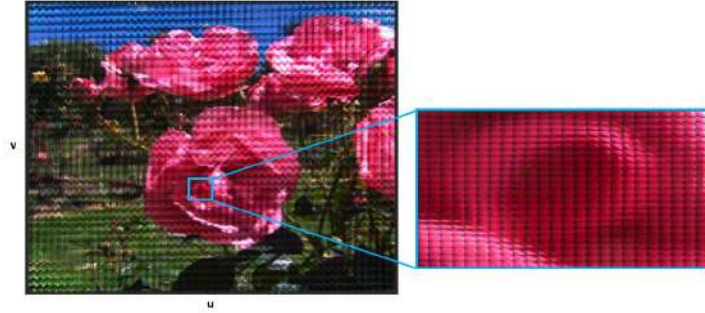


Figura 4.11: Visualización del campo de luz creado con la función $LFDispTiles(uvst)$.

Visualización por $LFDispProjSubfigs$ Finalmente, la cuarta función construye seis subfiguras que muestran proyecciones del campo de luz en pares de distintas dimensiones, figura 4.12.

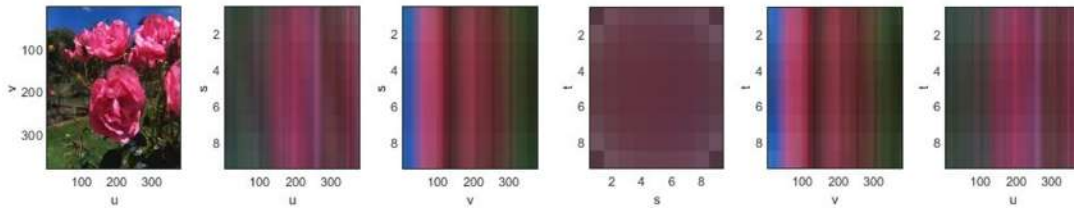


Figura 4.12: Visualización del campo de luz creado con la función $LFDispProjSubfigs$ visto en seis subfiguras dadas en pares de distintas dimensiones.

4.6. Filtrado

Las alternativas para generar imágenes con reenfoque sintético o enfocadas en diferentes profundidades del espacio son variadas, el LFFtoolbox posee 4 implementaciones de filtros para lograr este reenfoque, desde filtros aplicables en el dominio espacial hasta filtros que se trabajan en el espacio transformado de Fourier. En esta sección final de la implementación del método de evaluación de los campos de Luz se da a conocer el funcionamiento y la implementación de los filtros de suma de desplazamiento, el lineal 2D, el plano 4D y el Hyperfan 4D los cuales generan el reenfoque sintético. El resultado de cada uno de los filtros implementados se pueden visualizar en el capítulo 5 de análisis y resultados.

4.6.1. Filtro de suma y desplazamiento

Para el filtro de suma de desplazamiento se implementó la función *LFFiltShiftSum*, esta función se define como un filtro selectivo de profundidad de dominio espacial, es decir, funciona desplazando todas las posiciones en u, v del campo de luz a una profundidad común y luego se suman estas posiciones para obtener una única salida 2D. Esto es una ventaja ya que, al controlar la cantidad de cambio, se puede generar el enfoque en diferentes profundidades. Una representación del filtro se puede observar en la figura 4.13.

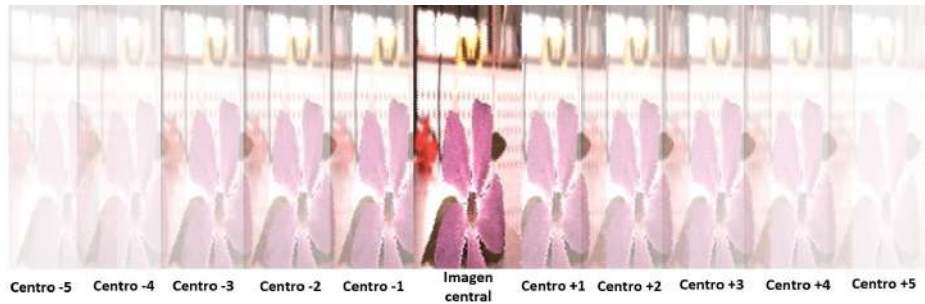


Figura 4.13: Representación visual del desplazamiento creado por el filtro.

La cantidad en la que se deben desplazar los cortes del campo de luz mencionada anteriormente se designa como la pendiente del filtro, esta pendiente con relación a la profundidad depende de la parametrización del campo de luz, pero en general, una pendiente de 0 se encuentra cerca del centro de la profundidad de campo capturada.

4.6.2. Filtro lineal 2D

En el caso del filtro lineal 2D, primero se tuvo que llevar a cabo la construcción del filtro con la función *LFBUILD2DFreqLine* la cual construye un filtro pasa banda lineal 2D en el dominio de la frecuencia, es decir, esta crea una respuesta de magnitud de valor real en 2D, para la cual la banda de paso es lineal. Una vez creado el filtro visualizado en la figura 4.14, este se aplicó al campo de luz utilizando *LFFilt2DFFT*; dicha función se ejecuta en secciones 2D del campo de luz de entrada, en los cuales es aplicado el filtro 2D creado anteriormente, a cada uno de ellos. La función toma la FFT de cada porción del campo de luz de entrada, la multiplica por la respuesta de magnitud proporcionada por el filtro 2D creado y luego esta llama a la FFT inversa.

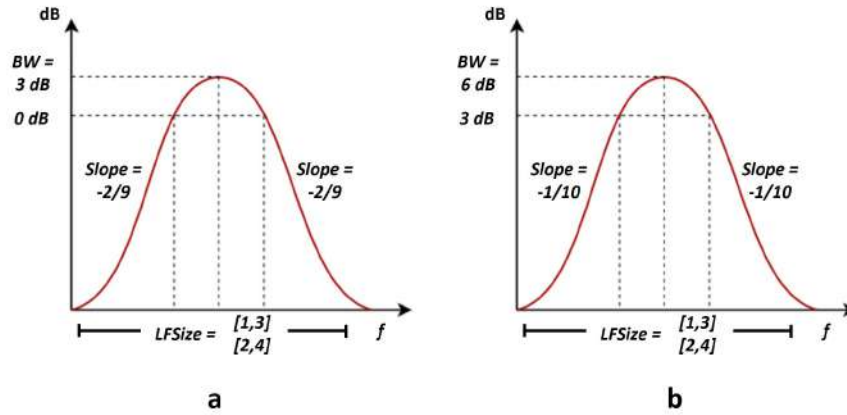


Figura 4.14: a: Filtro lineal 2D pasabandas implementado para los campos de luz creados por la cámara Lytro. b: Filtro pasabandas implementado para los campos de luz creados por la cámara Raytrix R42.

4.6.3. Filtro plano 4D

Para poder implementar el filtro plano 4D, de la misma manera que para el filtro lineal 2D, se tuvo que llevar a cabo primero la construcción de este usando la función *LFBUILD4DFreqPlane* la cual construye un filtro pasa banda plano 4D en el dominio de la frecuencia, o en otras palabras, construye una respuesta de magnitud de valor real en 4D, para la cual la banda de paso es un plano. Esto es útil para seleccionar objetos a una sola profundidad de un campo de luz y tiene un efecto similar al reenfoque utilizando en el filtro de suma y desplazamiento *LFFiltShiftSum*.

Luego de crear el filtro visualizado en la figura 4.15, este se aplicó al campo de luz utilizando *LFFilt4DFFT*, la cual funciona tomando la FFT del campo de luz de entrada, multiplicándola por la respuesta de magnitud proporcionada por el filtro creado anteriormente y luego llamando a la FFT inversa. Así como en el filtro lineal 2D, el tamaño del filtro plano 4D debe coincidir o exceder el tamaño del campo de luz. Así mismo, si el tamaño del filtro es más grande que el campo de luz de entrada, este se rellena con ceros para que coincida con el tamaño del filtro.

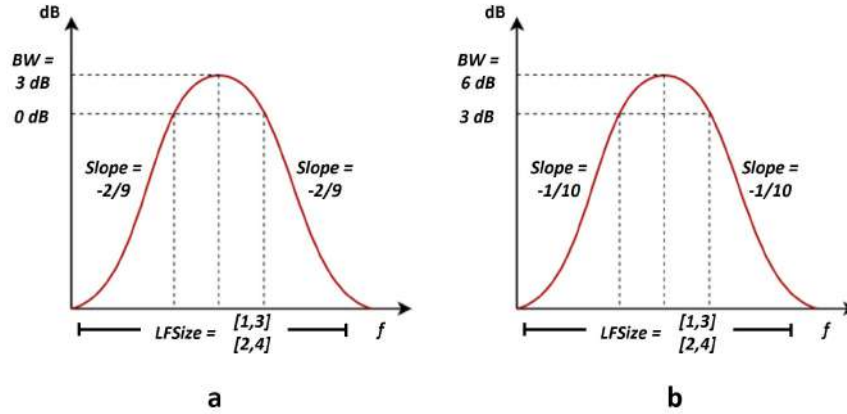


Figura 4.15: a: Filtro plano 4D pasabandas implementado para los campos de luz creados por la cámara Lytro. b: Filtro pasabandas implementado para los campos de luz creados por la cámara Raytrix R42.

4.6.4. Filtro Hyperfan 4D

Por último se implementó el filtro Hyperfan 4D, de la misma manera que para el filtro lineal 2D, se tuvo que llevar a cabo primero la construcción de este usando la función *LFBUILD4DFreqHyperfan* la cual construye un filtro pasa banda hyperfan 4D en el dominio de la frecuencia, es decir, construye una respuesta de magnitud de valor real en 4D, para la cual la banda de paso tiene un estilo de ventilador. Esto es útil para seleccionar objetos en un rango de profundidades desde un campo de luz, es decir, un enfoque volumétrico. Al tener creado el filtro visualizado en la figura 4.16, este se aplicó al campo de luz utilizando *LFFILT4DFFT*, la cual funciona tomando la FFT del campo de luz de entrada, multiplicándola por la respuesta de magnitud proporcionada por el filtro creado anteriormente y luego llamando a la FFT inversa como los anteriores filtros diseñados.

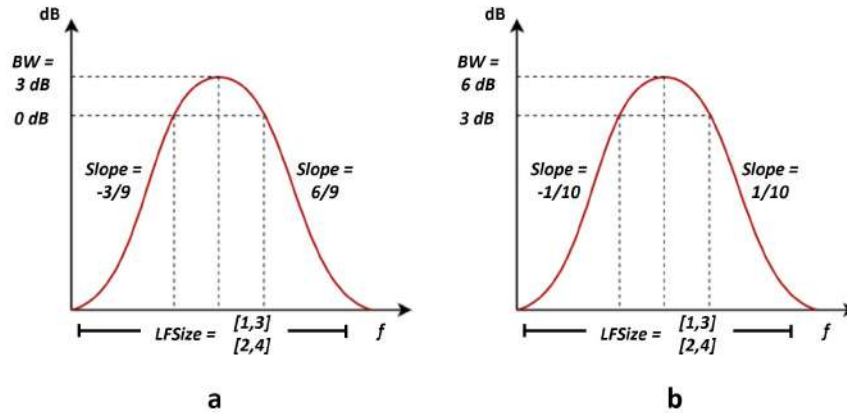


Figura 4.16: a: Filtro Hyperfan 4D pasabandas implementado para los campos de luz creados por la cámara Lytro. b: Filtro pasabandas implementado para los campos de luz creados por la cámara Raytrix R42.

CAPÍTULO 5

Análisis y resultados

Contenido

5.1. Dataset de fotos	44
5.1.1. Dataset Raytrix R42 (Anexo 1)	45
5.1.2. Dataset cámara Lytro (Anexo 1)	46
5.2. Variación de calidad en imágenes blancas obtenidas por la Raytrix R42	48
5.3. Variación de lentes en la cámara Raytrix R42 para la creación del campo de luz	51
5.3.1. Lente 58001	51
5.3.2. Lente 59872	53
5.4. Variación del campo de luz para cada tipo de lente	55
5.5. Análisis de los filtros de LFToolbox	56
5.5.1. Filtro de suma y desplazamiento	56
5.5.2. Filtro de línea 2D	57
5.5.3. Filtro plano 4D	58
5.5.4. Filtro Hyperfan 4D	59
5.6. Sharepoint	61

En este capítulo se dan a conocer los resultados obtenidos en el desarrollo de todo el trabajo de grado incluyendo el dataset obtenido, la reconstrucción de los campos de luz para la cámara Raytrix R42, los resultados obtenidos del método de evaluación usando las librerías de LFToolbox y la pagina web de Sharepoint donde fueron incluidos todos los archivos de captura de las cámaras y su documentación respectiva, así mismo, se da un análisis del desarrollo en cada apartado de la implementación.

5.1. Dataset de fotos

La información para la calibración de la cámara Raytrix R42 se obtuvo del documento en línea [5] y de adicionalmente algunas imágenes del dataset de las cámaras Raytrix R42 y Lytro que se exponen en el siguiente capítulo.

Para la obtención del dataset de imágenes con las cámaras plenópticas, se llevó a cabo el siguiente procedimiento:

5.1.1. Dataset Raytrix R42 (Anexo 1)

- **Paso 1. Preparación del entorno:** Se preparó el entorno donde se realizó la captura de imágenes de las plantas. Se aseguró contar con iluminación adecuada y un fondo apropiado.
- **Paso 2. Configuración de las cámaras:** Se configura la cámara Raytrix R42 de acuerdo a las necesidades específicas de la captura. Esto puede incluir ajustes de apertura, enfoque y otros parámetros relevantes 3.1.
- **Paso 3. Calibración de las cámara:** Se realiza la calibración de las cámara plenóptica Raytrix R42 a través del software de su fabricante, para asegurar la precisión y alineación adecuada de los sensores. Esto implica capturar imágenes de un patrón de calibración y utilizar algoritmos específicos para obtener los parámetros de calibración necesarios (ver figura 5.1).

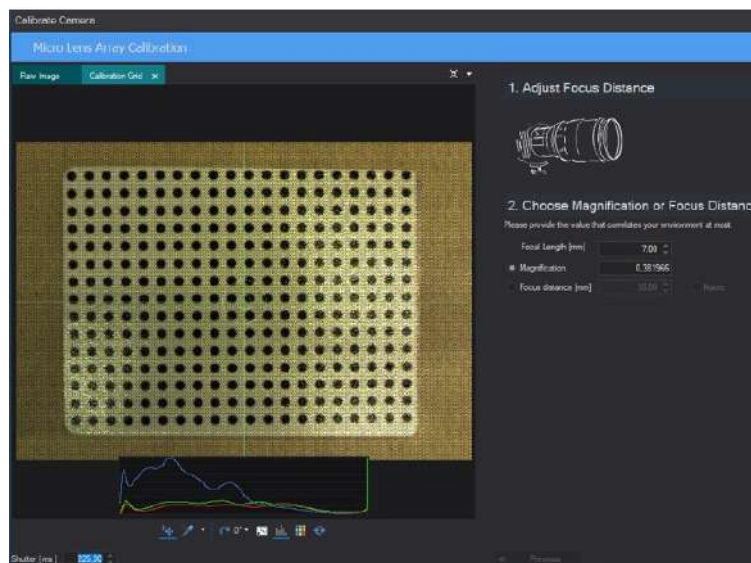


Figura 5.1: Software de calibración cámara Raytrix R42

- **Paso 4. Captura de las imágenes:** Se capturan las imágenes de las plantas utilizando la cámara plenóptica. Se pueden tomaron imágenes desde diferentes ángulos y distancias para obtener una variedad de perspectivas y detalles.
- **Paso 5. Organización y etiquetado de las imágenes:** Una vez capturadas, las imágenes se organizan y etiquetan adecuadamente para facilitar su posterior análisis y procesamiento como se observa en la figura 5.2.



Figura 5.2: Fotografías capturadas con la cámara Raytrix R42

- **Paso 6. Almacenamiento y respaldo de los datos:** Las imágenes capturadas se almacenaron y respaldaron de manera segura para evitar la pérdida de datos.

5.1.2. Dataset cámara Lytro (Anexo 1)

Para la captura de las fotografías del dataset con la cámara Lytro se tuvieron en cuenta los siguientes pasos:

- **Paso 1. Preparación del entorno:** Se preparó el entorno donde se realizó la captura de imágenes de las plantas. Se aseguró contar con iluminación adecuada y un fondo apropiado.
- **Paso 2. Captura de las imágenes:** Se capturan las imágenes de las plantas utilizando la cámara plenóptica. Se pueden tomar imágenes desde diferentes ángulos y distancias para obtener una variedad de perspectivas y detalles.
- **Paso 3. Obtención del software para exportar las imágenes:** Se obtiene el software de la cámara Lytro a través del siguiente foro [16], donde se explica el uso y los pasos a seguir para exportar las imágenes (ver figura 5.3).

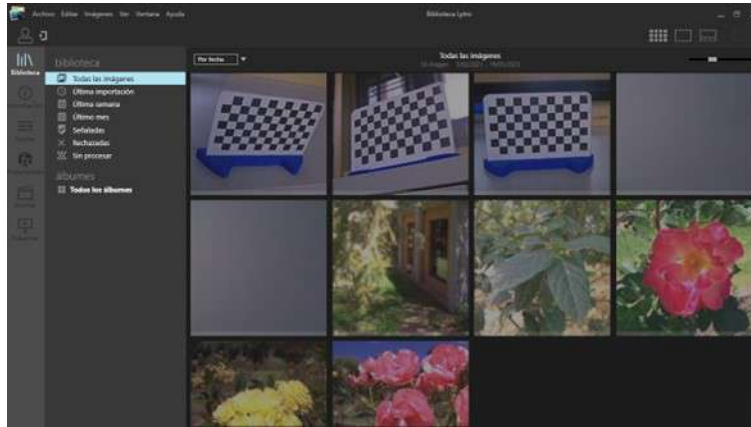


Figura 5.3: Software para exportar las imágenes de la cámara Lytro

- **Paso 5. Organización y etiquetado de las imágenes:** Una vez capturadas, las imágenes se organizan y etiquetan adecuadamente para facilitar su posterior análisis y procesamiento como se observa en la figura 5.4.

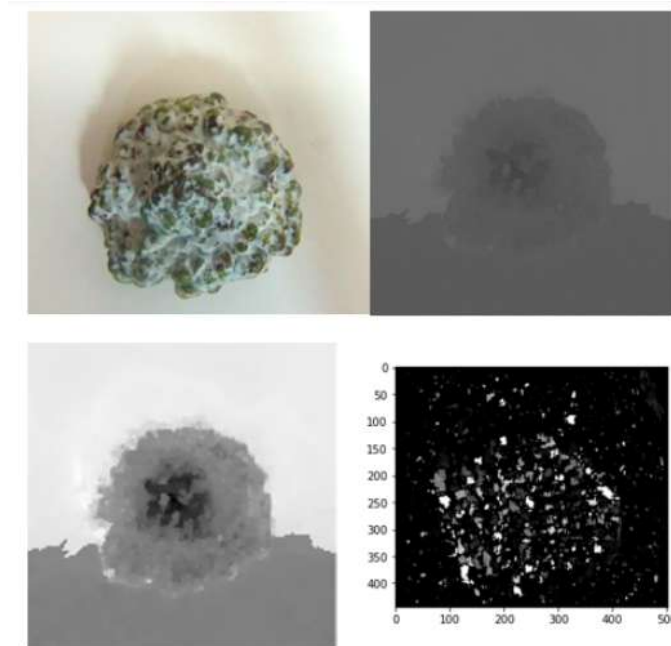


Figura 5.4: Imágenes capturadas con la cámara lytro, exportadas a través del software [5].

5.2. Variación de calidad en imágenes blancas obtenidas por la Raytrix R42

En esta sección se da a conocer la versatilidad del algoritmo para hallar los lentes y clasificarlos por sus tipos, esto gracias a una comparación entre imágenes en blanco tomadas con la cámara Raytrix R42 variando la apertura de los lentes y la cantidad de luz que se usó para tomar estas fotos como se ve en la figura 5.5. Para lograr esto se usó un rango variable en el procesamiento del thresholding por rangos, esto ya que, al tener un rango variable, se puede lograr un ajuste personalizado para cada fotografía en blanco si se requiere.

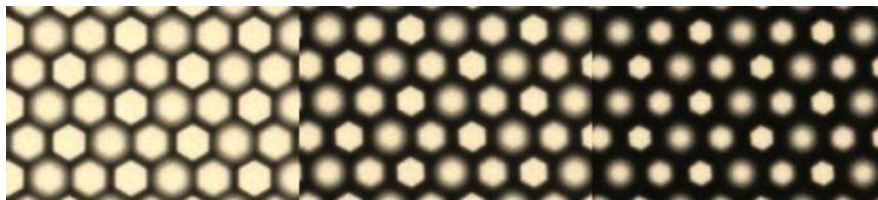


Figura 5.5: Muestra de tres distintas fotografías tomadas para la verificación de calidad del algoritmo, en estas se ve una fotografía con una gran apertura, una fotografía con apertura media y una fotografía con apertura reducida.

Se tomaron 3 fotografías en rangos donde la apertura tuviera grandes variaciones, para esto se obtuvo la primera imagen con una gran apertura, la segunda con una apertura con una medida ideal y la tercera con una apertura reducida, esto con el objetivo de analizar el comportamiento del algoritmo a la hora de entregar los resultados en los hallazgos y clasificación de los distintos lentes.

Imagen con apertura ampliada: Esta fotografía se tomó con una apertura cercana al 80 % de la apertura máxima, en la figura 5.6, se observa la poca separación que hay entre los lentes. Como primer análisis se puede llegar a concluir que esta apertura es beneficiosa para la creación del campo de luz final, sin embargo, en la practica también se encuentra la variación de la luz en el ambiente y, si el entorno no esta totalmente controlado, las fotografías pueden llegar a tener saturación de luz y el software no se podría en ellas.

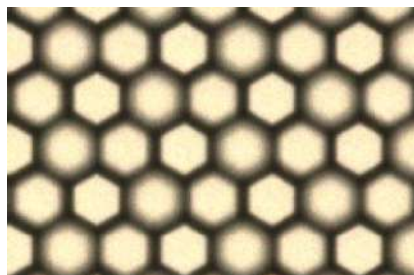


Figura 5.6: Muestra de una fotografía blanca con una apertura ampliada.

En la tabla 5.1 se dan a conocer los rangos usados para el procesamiento de thresholding los cuales fueron de $[80, 150]$, el numero de lentes hallados para cada tipo de micro lente y los errores absoluto y porcentuales de la detección y clasificación de los lentes para la apertura ampliada con respecto a la cantidad de micro lentes exactas que hay en el lente principal.

	Rango inferior	Rango superior	N. de lentes hallados	Error absoluto	Error porcentual
Imagen apertura amplia	80	150	36744	1976	5,10 %
Lente tipo 1	80	150	12441	465	3,60 %
Lente tipo 2	80	150	12372	534	4,14 %
Lente tipo 3	80	150	11931	975	7,55 %

Tabla 5.1: Resultados en la detección de micro lentes para una apertura ampliada.

Imagen con apertura ideal Esta fotografía se tomó con una apertura ideal para el algoritmo, esta es cercana al 60 % de la apertura máxima, en la figura 5.7, se observa una separación media que hay entre los distintos lentes. Esto es ideal ya que en la practica, esta apertura es versátil a los cambios en la intensidad de la luz y no depende de variables externas para un buen funcionamiento ya que cuenta con un rango amplio para trabajar en lugares con sobre-exposición y sub-exposición de luz.

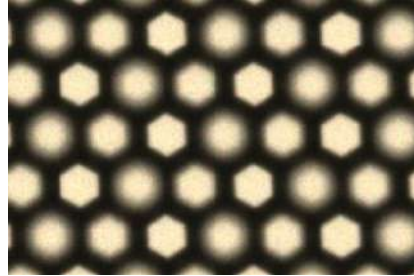


Figura 5.7: Muestra de una fotografía blanca con una apertura ideal.

En la tabla 5.2 se dan a conocer los rangos usados para el procesamiento de thresholding los cuales fueron de $[80, 200]$, el numero de lentes hallados para cada tipo de micro lente y los errores absoluto y porcentuales de la detección y clasificación de los lentes para la apertura ideal con respecto a la cantidad de micro lentes exactas que hay en el lente principal.

	Rango inferior	Rango superior	N. de lentes hallados	Error absoluto	Error porcentual
Imagen apertura media	80	200	38693	27	0,07 %
Lente tipo 1	80	200	12930	24	0,19 %
Lente tipo 2	80	200	12921	15	0,12 %
Lente tipo 3	80	200	12842	64	0,50 %

Tabla 5.2: Resultados en la detección de micro lentes para una apertura ideal.

Imagen con apertura reducida Esta fotografía se tomó con una apertura ideal para el algoritmo, esta es cercana al 60 % de la apertura mínima, en la figura 5.8, se observa una separación

extensa entre los distintos lentes. Esto podría ser ideal para el algoritmo, ya que, al contar con mas separación, los lentes son mas fáciles de detectar. Sin embargo, los resultados a la hora de crear el campo de luz no serán óptimos debido a la poca apertura y por ende, la poca información que los lentes tendrán de la fotografía capturada.

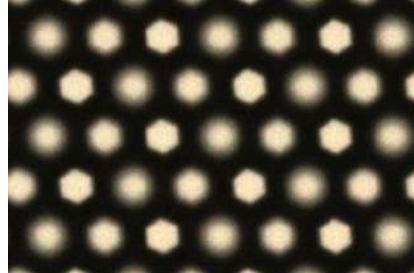


Figura 5.8: Muestra de una fotográfica blanca con una apertura reducida.

En la tabla 5.3 se dan a conocer los rangos usados para el procesamiento de thresholding los cuales fueron de $[65, 140]$, el numero de lentes hallados para cada tipo de micro lente y los errores absoluto y porcentuales de la detección y clasificación de los lentes para la apertura reducida con respecto a la cantidad de micro lentes exactas que hay en el lente principal.

	Rango inferior	Rango superior	N. de lentes hallados	Error absoluto	Error porcentual
Imagen blanca reducida	65	140	38453	267	0,69 %
Lente tipo 1	65	140	12925	19	0,15 %
Lente tipo 2	65	140	12856	50	0,39 %
Lente tipo 3	65	140	12672	234	1,81 %

Tabla 5.3: Resultados en la detección de micro lentes para una apertura reducida.

Análisis de la variación de calidad en imágenes blancas:

Como análisis de los resultados obtenidos anteriormente, para el lente con la apertura ampliada se llego a un error del 5,10 %, hallando así 36744 de 38720 lentes totales que se pueden detectar; para el lente con la apertura ideal se obtuvo un error del 0,07 %, hallando así 38693 de 38720 lentes totales que se pueden detectar; finalmente, para el lente con la apertura ampliada se llego a un error del 0,69 %, hallando así 38453 de 38720 lentes totales que se pueden detectar.

Debido a estos resultados se puede concluir que el algoritmo es útil para un amplio rango de aperturas pero teniendo que ajustar sutilmente el rango del procesamiento de thresholding. Por otra parte, el algoritmo funciona ligeramente mejor para aperturas medias o reducidas que para aperturas muy amplias, ya que se obtuvo un 0,07 % y 0,69 % de error contra el 5,10 % de error en la apertura ampliada. Debido a esto, si se requiere tomar fotografías en espacios con luz intensa, se recomienda usar una amplitud menor a la ideal para los lentes.

En cuanto a los resultados de la clasificación de cada lente se puede concluir que, para los tres casos de variación de apertura, el algoritmo pierde eficacia a la hora de detectar y clasificar los lentes tipo 3, esto se puede observar con la diferencia de errores que se obtuvieron en la clasificación de cada lente, siendo los errores de los lentes tipo 1 y 2 para la apertura ampliada de [3,60 %, 4,14 %], para la apertura ideal de [0,19 %, 0,12 %] y para la apertura reducida de [0,15 %, 0,39 %] contra el error del lente tipo 3 para la apertura ampliada de 7,55 % para la apertura ideal de 0,50 % y para la apertura reducida de 1,81 % respectivamente.

5.3. Variación de lentes en la cámara Raytrix R42 para la creación del campo de luz

Esta sección se dedica a mostrar una comparativa del uso de 2 tipos de lentes principales, además del lente original de la cámara *HF-1218-12M*, para la creación de un campo de luz. para esto se usaron los lentes 58001, 59872.

5.3.1. Lente 58001

Para el lente 58001 se obtuvo la fotografía en blanco de la figura 5.9, esta fotografía se tomó con una apertura ideal luego de hacer la respectiva calibración de la cámara, para proceder con su respectivo procesamiento mediante el algoritmo de creación de campos de luz.

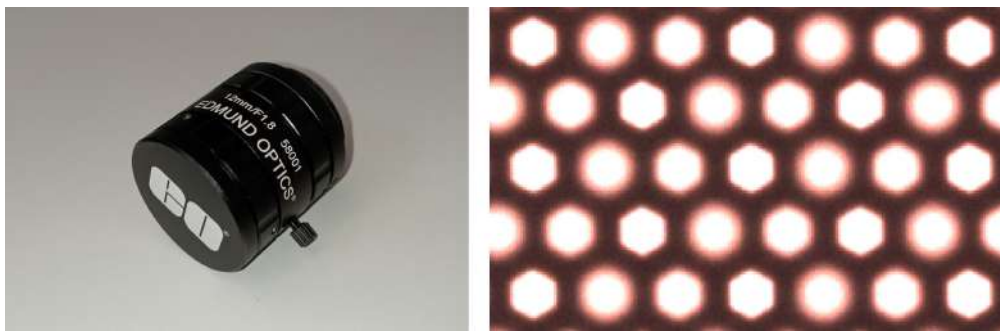


Figura 5.9: Lente 58001 y muestra de la fotografía blanca tomada este.

Esto da como resultado la clasificación de los lentes en la figura 5.10. Como se puede observar, al pasar por el algoritmo de K-Means para la clasificación de cada lente no se definieron bien los clusters, dando así un mal resultado para la posterior reconstrucción de la imagen.

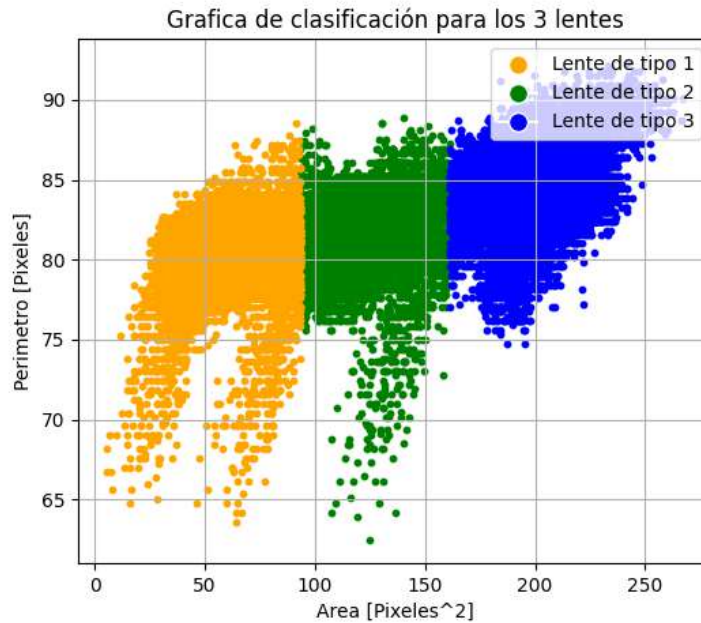


Figura 5.10: Gráfica de clasificación de los micro lentes para la fotografía tomada con el lente principal 58001.

Este resultado en la clasificación se da debido a que los lentes de las esquinas de la imagen en blanco tienden a tomar la forma circular del lente como se puede observar en la figura 5.11, esto puede deberse a que el lente adquirido y la cámara no son compatibles. Como análisis extra, es posible recortar estas imágenes para que las esquinas no interfieran pero se perdería mucha información y la reconstrucción de las imágenes bajarían su resolución aun mas de lo que ya lo hacen.

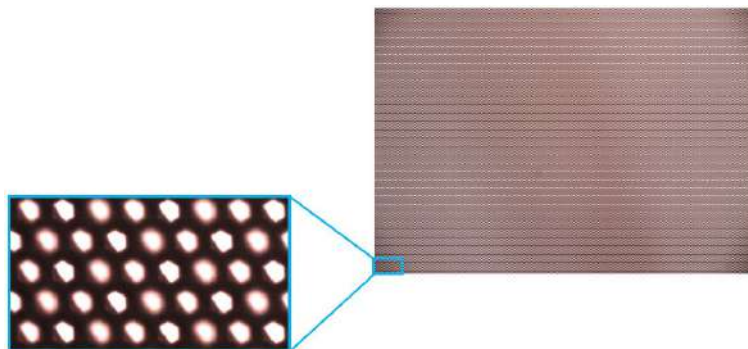


Figura 5.11: Muestra de una esquina de la fotografía blanca tomada con el lente 58001, donde se evidencia el cierre progresivo de los micro lentes.

5.3.2. Lente 59872

Por otro lado, para el lente 59872 se obtuvo la fotografía en blanco de la figura 5.12, esta fotografía se tomó con una apertura ideal luego de hacer la respectiva calibración de la cámara, para proceder con su respectivo procesamiento mediante el algoritmo de creación de campos de luz como con el lente anterior.

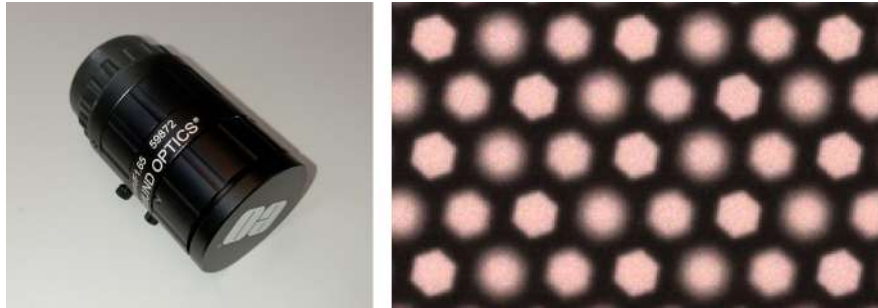


Figura 5.12: Lente 59872 y muestra de la fotografía blanca tomada este.

Esto da como resultado la clasificación de los lentes en la figura 5.13. Como se puede observar, al pasar por el algoritmo de K-Means para la clasificación de cada lente, en este casi si se definieron bien los clusters, lo cual da como resultado una buena clasificación y es posible hacer la reconstrucción de las imágenes para la creación del campo de luz.

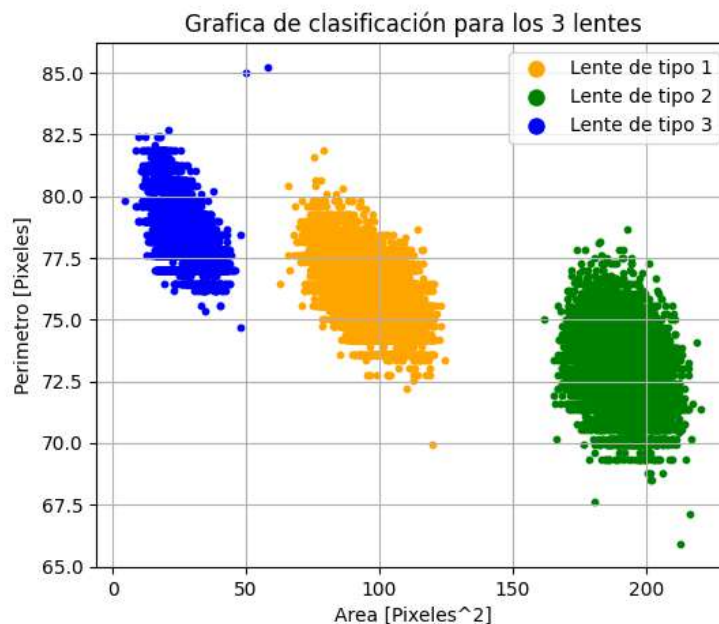


Figura 5.13: Gráfica de clasificación de los micro lentes para la fotografía tomada con el lente principal 59872.

En la tabla 5.4 se muestran los rangos usados para el procesamiento de thresholding los cuales fueron de $[80, 155]$. También se da a conocer el numero de lentes hallados para cada tipo de micro lente y los errores absoluto y porcentuales de la detección y clasificación de los lentes para la apertura reducida con respecto a la cantidad de micro lentes exactas que hay en el lente principal. En el caso del lente 59872, cuenta con el mismo numero de micro lentes que el lente original de la cámara *HF-1218-12M*, a su vez, las dimensiones de la matriz son iguales, esto crea una ventaja ya que no es necesario cambiar el código fuente para el manejo de fotografías tomadas por esta lente.

	Rango inferior	Rango superior	N. de lentes hallados	Error absoluto	Error porcentual
Lente mediano	80	155	38575	145	0,37 %
Lente tipo 1	80	155	13013	107	0,83 %
Lente tipo 2	80	155	13002	96	0,74 %
Lente tipo 3	80	155	12560	346	2,68 %

Tabla 5.4: Resultados en la detección de micro lentes para fotografías en blanco tomadas por el lente 59872.

Con respecto a los errores para la clasificación presentada en la tabla 5.4, se puede observar el mismo caso que se presenta al clasificar los lentes de las imágenes en blanco tomadas por el lente original de la cámara (*HF-1218-12M*). el algoritmo no detecta con la misma precisión los lentes tipo 3 como con los lentes tipo 1 y 2.

En cuanto a los resultados de la reconstrucción de las imágenes y la posterior creación del campo de luz se obtuvo que la creación de este campo para el lente tipo 3 no fue la mas óptima como se observa en la figura 5.14, esto posiblemente debido a la detección de los lentes tipo 3.

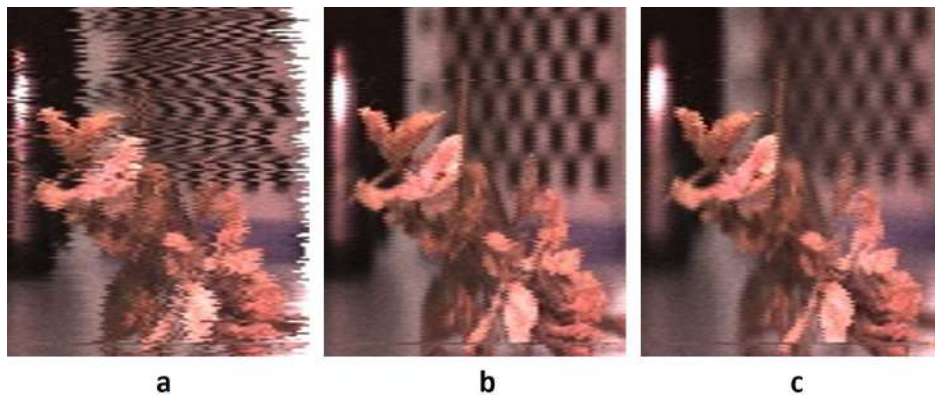


Figura 5.14: a: Muestra de la reconstrucción de la imagen para el lente tipo 3. b: Muestra de la reconstrucción de la imagen para el lente tipo 2. c: Muestra de la reconstrucción de la imagen para el lente tipo 1.

5.4. Variación del campo de luz para cada tipo de lente

Para analizar la variación del campo de luz con respecto a cada tipo de micro lente clasificado en la reconstrucción para las imágenes tomadas por la cámara Raytrix R42 se debe tener en cuenta la cantidad de enfoque que tenga cada tipo de micro lente, esto varia dependiendo del tipo de lente principal que sea usado para capturar las muestras. En este caso, para el campo de luz generado con el lente original de la Raytrix R42 (*HF-1218-12M*), cuenta con una variación poco perceptible como se ve en la figura 5.15.

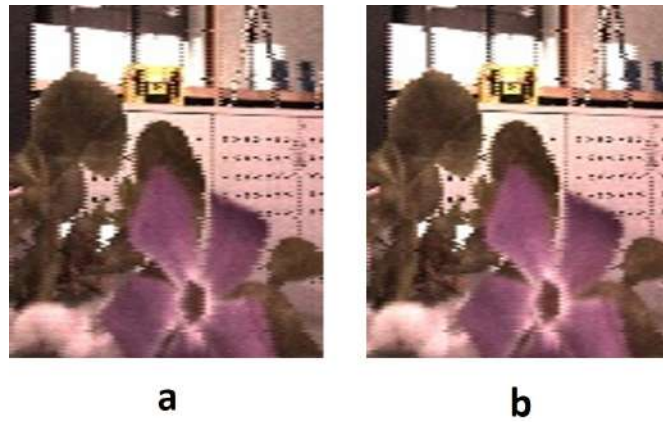


Figura 5.15: a: Muestra de la reconstrucción de la imagen para el lente tipo 3. b: Muestra de la reconstrucción de la imagen para el lente tipo 2. c: Muestra de la reconstrucción de la imagen para el lente tipo 1.

Lo contrario sucede al usar muestras del campo de luz generado por el lente 59872 ya que los enfoques de los micro lentes de esta lente principal cuentan con una mayor diferencia entre si como se puede observar en la figura 5.16.

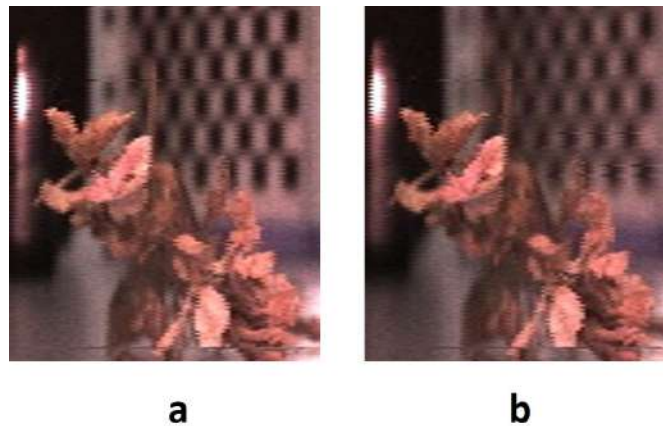


Figura 5.16: a: Muestra de la reconstrucción de la imagen para el lente tipo 3. b: Muestra de la reconstrucción de la imagen para el lente tipo 2. c: Muestra de la reconstrucción de la imagen para el lente tipo 1.

5.5. Análisis de los filtros de LFToolbox

En esta sección se muestran los resultados de los filtros implementados en el capítulo 4 y se analiza el mejoramiento de las imágenes gracias al reenfoque sintético con respecto a las imágenes originales del campo de luz.

5.5.1. Filtro de suma y desplazamiento

Para el caso de los resultados obtenidos por el filtro de suma y desplazamiento se tuvo que definir el valor de la pendiente en dos partes, en la primera se asignó un valor de $-2/9$ y en la segunda de $6/9$, las cuales se pueden visualizar aplicadas en el campo de luz capturado en la figura 5.17, esto para el campo de luz tomado con la cámara Lytro.

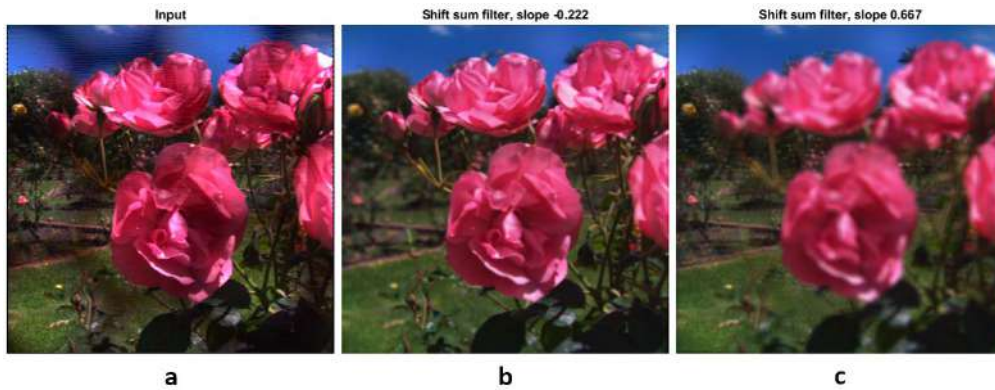


Figura 5.17: Comparación entre el campo de luz de entrada y el filtro suma y desplazamiento para muestras de Lytro. a: Muestra del campo de luz de entrada. b: Filtro con una pendiente de $-2/9$. c: Filtro con una pendiente de $6/9$.

Por otra parte, para la implementación del filtro usado en los campos de luz tomados por la cámara Raytrix R42, se asignó un valor para la pendiente de 0, esto debido a que la flor se encuentra enfocada en el centro de la profundidad del campo. Por ultimo, el filtro cuenta con una opción de superresolución de reenfoque lineal, la cual escala por el factor asignado, para el caso del filtrado del campo de luz se tomó un valor de escala de 3. El resultado de esta implementación se puede verificar en la figura 5.18.

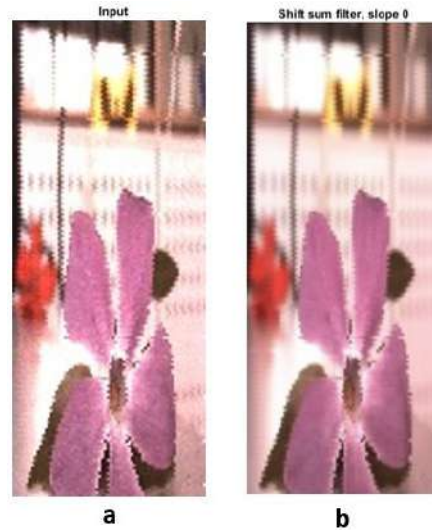


Figura 5.18: Comparación entre el campo de luz de entrada y el filtro suma y desplazamiento para muestras de Raytrix. a: Muestra del campo de luz de entrada. b: Filtro con una pendiente de 0.

5.5.2. Filtro de línea 2D

Para usar el filtro lineal 2D fue necesario especificar las dos dimensiones a lo largo de las cuales se debe aplicar este como se observa en la figura 4.14 del capítulo 4. El tamaño del filtro en el dominio de la frecuencia debe coincidir o superar el tamaño del campo de luz en estas dimensiones. Si el filtro de dominio en frecuencia es más grande que el campo de luz de entrada, el resto del campo de luz se rellena con ceros para que coincida con el tamaño del filtro.

En la implementación del filtro usado en los campos de luz tomados por la cámara Lytro, luego de generar un tamaño mayor que el del campo de luz para el filtro, se asignaron dos valores, el primero de $[1, 3]$ y el segundo de $[2, 4]$, a las dimensiones en las que se aplicó la matriz, para la pendiente del filtro se tomó un valor de $-2/9$ y para el ancho de banda del filtro se asignó un valor de 3 dB, los resultados de este filtro se pueden observar en la figura 5.19.



Figura 5.19: Comparación entre el campo de luz de entrada y el filtro lineal 2D para muestras de Lytro.

Por otro lado, para la implementación de este filtro usando los campos de luz tomados por la cámara Raytrix R42, se asignó un valor para la pendiente de $-1/10$, esta pendiente tan pequeña es debido a que la flor se encuentra enfocada en el centro de la profundidad del campo pero se puede generar un poco de pendiente para generar un desenfoque mayor en el fondo. El resultado de esta implementación se puede verificar en la figura 5.20.

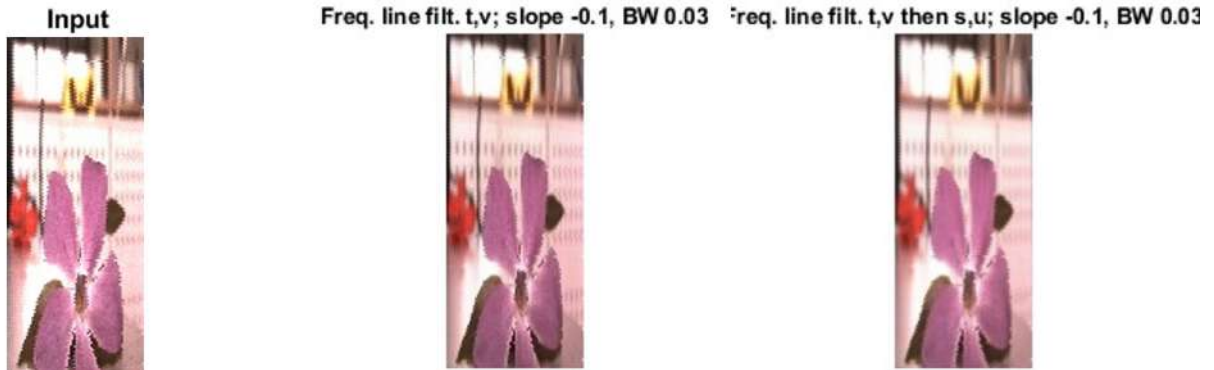


Figura 5.20: Comparación entre el campo de luz de entrada y el filtro lineal 2D para muestras de Raytrix.

5.5.3. Filtro plano 4D

Para usar el filtro plano 4D fue necesario especificar las dos dimensiones a lo largo de las cuales se debe aplicar este como se observa en la figura 4.15 del capítulo 4. El tamaño del filtro en el dominio de la frecuencia debe coincidir o superar el tamaño del campo de luz en estas dimensiones. Si el filtro de dominio en frecuencia es más grande que el campo de luz de entrada, el resto del campo de luz se rellena con ceros para que coincida con el tamaño del filtro como en los demás filtros propuestos.

En la implementación del filtro usado en los campos de luz tomados por la cámara Lytro, luego de generar un tamaño mayor que el del campo de luz para el filtro, se asignaron dos valores, el primero de $[1, 3]$ y el segundo de $[2, 4]$, a las dimensiones en las que se aplicó la matriz, para la pendiente del filtro se tomó un valor de $-2/9$ y para el ancho de banda del filtro se asignó un valor de 3 dB, los resultados de este filtro se pueden observar en la figura 5.21.

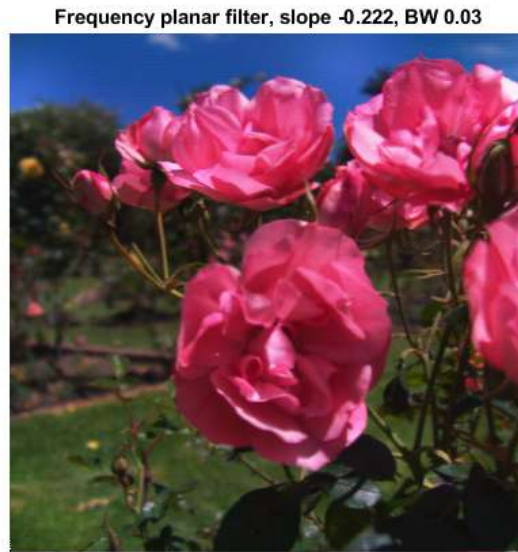


Figura 5.21: Salida del filtro plano 4D para muestras de Lytro.

Por otro lado, para la implementación de este filtro usando los campos de luz tomados por la cámara Raytrix R42, se asignó un valor para la pendiente de $-1/10$. El resultado de esta implementación se puede verificar en la figura 5.22.

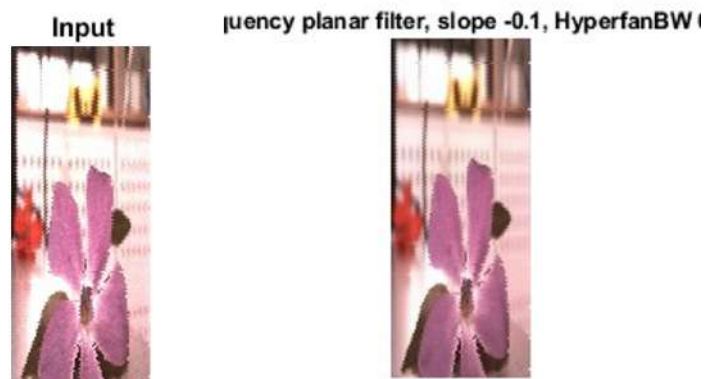


Figura 5.22: Comparación entre el campo de luz de entrada y el filtro lineal 4D para muestras de Raytrix.

5.5.4. Filtro Hyperfan 4D

Para usar el filtro lineal 2D fue necesario especificar las dos dimensiones a lo largo de las cuales se debe aplicar este como se observa en la figura 4.16 del capítulo 4. El tamaño del filtro en el dominio de la frecuencia debe coincidir o superar el tamaño del campo de luz en estas dimensiones.

Si el filtro de dominio en frecuencia es más grande que el campo de luz de entrada, el resto del campo de luz se rellena con ceros para que coincida con el tamaño del filtro.

En la implementación del filtro usado en los campos de luz tomados por la cámara Lytro, luego de generar un tamaño mayor que el del campo de luz para el filtro, se asignaron dos valores, el primero de $[1, 3]$ y el segundo de $[2, 4]$, a las dimensiones en las que se aplicó la matriz, para el rango de las pendientes del filtro se tomaron valores de $-3/9$ y $6/9$, para el ancho de banda del filtro se asignó un valor de 3 dB, los resultados de este filtro se pueden observar en la figura 5.23.



Figura 5.23: Salida del filtro Hyperfan 4D para muestras de Lytro.

Por otro lado, para la implementación de este filtro usando los campos de luz tomados por la cámara Raytrix R42, se asignaron valores para las pendientes de $-1/10$ y $1/10$, este rango de pendientes es debido a que la flor, al ser un objeto tridimensional, se encuentra enfocada en distintas profundidades del campo. El resultado de esta implementación se puede verificar en la figura 5.24.

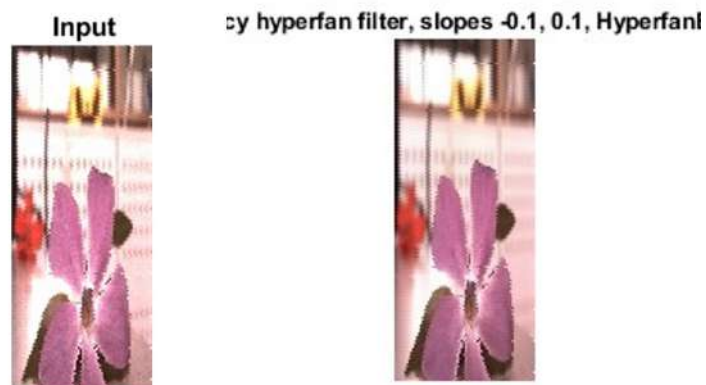


Figura 5.24: Comparación entre el campo de luz de entrada y el filtro Hyperfan 4D para muestras de Raytrix.

5.6. Sharepoint

A continuación se presenta la plataforma web en la cual se documentan todos los archivos relacionados con el proyecto y se presenta la información de manera sintetizada, con una mejor visualización de los resultados obtenidos.



Figura 5.25: Pagina principal Sharepoint

El objetivo de la creación del Sharepoint es mantener la documentación del trabajo de grado en una pagina web que sea de fácil acceso, en la cual se pueda obtener la información acerca del desarrollo de la problemática que se afronto y los resultados obtenidos con el desarrollo completo del proyecto (Ver figura 5.25).

La documentación almacenada en el Sharepoint contiene una introducción sobre el porque del proyecto, partiendo desde la problemática abordada (Ver figura 5.26.a), teniendo en cuenta los objetivos del proyecto, incluyendo tambien los documentos sobre los trabajos que ya existen para solucionar problemáticas similares (Ver figura 5.26.b).

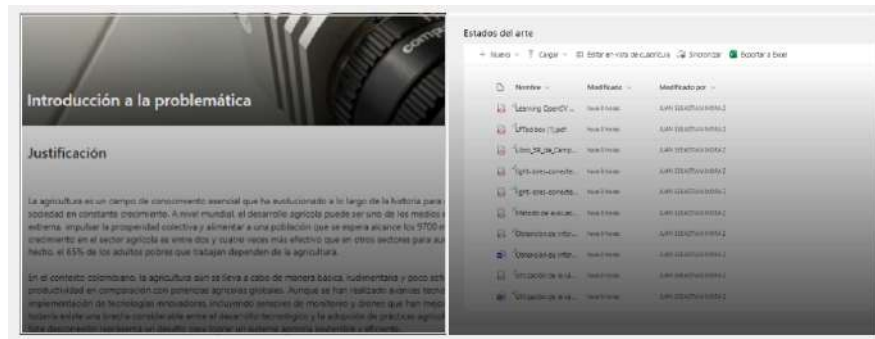


Figura 5.26: a:Justificación del proyecto en Sharepoint, b: Estados del arte del proyecto en Sharepoint.

Posteriormente se deposito la información detallada sobre los métodos de solución abordados en este trabajo de grado, se realiza un paso a paso de los procedimientos sobre ¿Como crear campos de Luz con cámaras plenópticas? y ¿Como emplear campos de luz usando LFToolbox? (Ver figura 5.27).



Figura 5.27: Paso a paso para emplear campos de luz con LFToolbox.

Adicionalmente se presentan los resultados obtenidos para los objetivos abordados con el proyecto, incluyendo el análisis de los mismos y la documentación que se deja del proyecto, para la continuación con trabajos futuros (Ver figura 5.28).



Figura 5.28: Sección de resultados en el Sharepoint.

Finalmente se creo una sección en la cual se incluyen las conclusiones sobre el proyecto y las aportaciones y trabajos futuros que se pueden realizar a partir del desarrollo realizado en este trabajo de grado (Ver figura 5.29).



Figura 5.29: Sección de conclusiones en el Sharepoint.

CAPÍTULO 6

Conclusiones y trabajos futuros

6.1. Conclusiones

- El objetivo principal del proyecto fue la creación de un software que permitiera crear campos de luz usando camaras plenópticas 2.0 para poder ser evaluados utilizan ya que no se encuentra documentación lo suficientemente robusta para esta área, una vez creado los algoritmos, se reveló que las imágenes que componen el campo de luz creado a partir de la cámara Raytrix R42 cuentan con una resolución baja. Esto debido a que el tamaño de las imágenes finales dependen directamente de la cantidad de lentes que la cámara tenga, y, al estar repartida en 3 tipos de micro lentes, la resolución de cada campo de luz se ve reducida en una tercera parte.
- Se evidencio que el algoritmo es óptimo para captar 2 de los 3 tipos de lentes que hay en las imágenes tomadas por la Raytrix R42, esto debido al método utilizado para la obtención de estos ya que el tercer tipo de lente, al contar con mucha área de desenfoque, el algoritmo puede confundir algún tipo de ruido que tenga la imagen con este tipo de lente.
- Al comparar los resultados obtenidos por el lente principal *HF-1218-12M* y el lente 59872 se puede llegar a concluir que el uso del segundo lente es óptimo y se puede usar para la generación de campos de luz con este mentido, teniendo en cuenta los ajustes pertinentes para el uso de este tipo de imágenes.
- Se tuvo la dificultad en el reenfoque sintético para los campos de luz creados con la cámara Raytrix debido a que, al ser captados por una sola lente, estos campos no cuentan con una extensa información con respecto a las dimensiones (s,t,u,v). Este problema se podría solucionar implementando una imagen en blanco para cada tipo de lente directamente en el Toolbox y repartiendo la imagen original en tres imágenes, una para cada lente.

6.2. Trabajos futuros

El proyecto de tesis se concluyó con buenos resultados como se muestran en los capítulos anteriores, pero el sistema aún no es óptimo. Es posible que se deban considerar mejoras en la creación de los campos de luz y optimizar los procesos para ahorrar tiempo; además de una profundización

en las áreas de reenfoque sintético con las librerías de LFToolbox. Esta es la razón por la cual se sugieren trabajos a futuro.

- **Superresolución como técnica que busca mejorar la calidad y nitidez de las imágenes capturadas con la cámara raytrix R42:** Se podrían explorar diferentes enfoques de superresolución, como el uso de técnicas basadas en aprendizaje profundo, como redes neuronales convolucionales, para reconstruir imágenes de alta resolución a partir de las imágenes de baja resolución capturadas por la cámara. También se podrían investigar métodos de fusión de información de múltiples imágenes de campos de luz para obtener una imagen de alta resolución más precisa y detallada.

Este trabajo futuro debería estar respaldado por una revisión exhaustiva de la literatura existente sobre superresolución y fotografía plenóptica, así como por experimentos y pruebas rigurosas para validar y respaldar los resultados obtenidos. Además, se deberían tener en cuenta las limitaciones y desafíos asociados con la aplicación de técnicas de superresolución en imágenes de campos de luz, como la presencia de ruido o artefactos.

- **Generación de una pendiente adecuada para los filtros en el Toolbox:** Actualmente, la configuración de los filtros en el Toolbox puede requerir ajustes manuales y experimentación para obtener resultados satisfactorios. Por lo tanto, se podría investigar y diseñar un algoritmo que analice las características de las imágenes de campos de luz y genere automáticamente una pendiente adecuada para los filtros utilizados en el proceso de procesamiento y análisis.

Para lograr esto, se debería estudiar en detalle la relación entre las características de las imágenes de campos de luz y los resultados deseados en términos de calidad visual, resolución, nitidez u otras métricas relevantes. Esto incluiría considerar factores como el rango dinámico, la distribución de intensidad, la estructura de los campos de luz y las limitaciones específicas del Toolbox.

Una vez comprendida la relación entre las características de la imagen y los resultados deseados, se podrían aplicar técnicas de aprendizaje automático, como el aprendizaje supervisado o el aprendizaje por refuerzo, para entrenar un modelo capaz de generar automáticamente una pendiente adecuada para los filtros del Toolbox. Este modelo podría basarse en datos de entrenamiento que incluyan imágenes de campos de luz con diferentes características y los resultados correspondientes que se consideren óptimos.

El trabajo futuro requeriría recopilar un conjunto de datos adecuado para el entrenamiento del modelo, lo cual implicaría capturar imágenes de campos de luz con diferentes configuraciones y realizar un análisis manual para determinar las pendientes óptimas correspondientes a cada configuración. Luego, se utilizarían estos datos etiquetados para entrenar y ajustar el modelo de generación automática de pendientes.

Una vez que se haya desarrollado el modelo, se debería evaluar y validar su desempeño utilizando un conjunto de datos separado. Esto implicaría comparar las pendientes generadas

automáticamente por el modelo con las pendientes óptimas determinadas manualmente en función de los resultados deseados.

- **Estimación de profundidad y la reconstrucción en 3D:** La estimación de profundidad y la reconstrucción en 3D son aspectos fundamentales en la fotografía plenóptica, ya que permiten obtener información tridimensional a partir de imágenes 2D capturadas desde diferentes perspectivas. Estas técnicas tienen aplicaciones en campos como la realidad virtual, la robótica, la medicina y la visión por computadora.

Para llevar a cabo este trabajo futuro, se requeriría investigar y diseñar algoritmos y técnicas adecuadas para realizar la estimación de profundidad y la reconstrucción en 3D. Esto implicaría analizar las características particulares de las imágenes capturadas por una cámara plenóptica, como los rayos de luz adicionales y la información angular, y utilizar esta información para inferir la profundidad de los objetos en la escena.

Una posible aproximación sería emplear técnicas de visión estéreo, las cuales se basan en el emparejamiento de puntos correspondientes en diferentes vistas para estimar la disparidad y, posteriormente, la profundidad. Sin embargo, se deberían adaptar y mejorar estas técnicas para aprovechar las propiedades específicas de las imágenes plenópticas, como la disponibilidad de información angular y la distribución de los micro-lentes.

Además, se podría considerar la incorporación de métodos de aprendizaje automático, como redes neuronales convolucionales, para mejorar la precisión y la robustez de la estimación de profundidad y la reconstrucción en 3D. Estos enfoques podrían utilizar datos de entrenamiento que incluyan imágenes capturadas con la cámara plenóptica y las correspondientes medidas de profundidad o modelos 3D de referencia.

El trabajo futuro requeriría la implementación y evaluación de los algoritmos propuestos utilizando conjuntos de datos adecuados, que podrían incluir imágenes plenópticas sintéticas o reales junto con información de profundidad o modelos 3D de referencia. Se debería realizar una comparación exhaustiva de los resultados obtenidos con otros métodos existentes y métricas de evaluación establecidas para validar la eficacia y el rendimiento del enfoque propuesto.

CAPÍTULO 7

Anexos

Anexo 1: Dataset

- <https://livejaverianaedu.sharepoint.com/sites/FotografiaPlenoptica2/SitePages/Fotograf%C3%ADas.aspx>

Anexo 2: Pagina web en Sharepoint

- <https://livejaverianaedu.sharepoint.com/sites/FotografiaPlenoptica2/SitePages/ProjectHome.aspx>

Anexo 3: Archivo fuente en repositorio de GitHub para la creación de un campos de luz

- https://github.com/SebastianYagama/Creacion_de_campos_de_luz_para_la_camara_Raytrix_R42.git

Anexo 4: Archivo fuente en repositorio de GitHub para el método de evaluación de un campos de luz

- https://github.com/SebastianYagama/Metodo_de_evaluacion_para_campos_de_luz_usando_LFToolbox.git

Anexo 5: Manual para el manejo de las librerías de LFToolbox

- <https://livejaverianaedu.sharepoint.com/sites/FotografiaPlenoptica2/Manual%20Para%20el%20manejo%20de%20las%20Librerias%20de%20LFToolbox/Forms/AllItems.aspx>

Bibliografía

- [1] M. Levoy, “Marc levoy’s home page.” <https://graphics.stanford.edu/~levoy/>.
- [2] N. Sabater, V. Drazic, M. Seifi, G. Sandri, and P. Pérez, “Light-field demultiplexing and disparity estimation,” 2014.
- [3] I. Ihrke, J. Restrepo, and L. Mignard-Debise, “Principles of light field imaging: Briefly revisiting 25 years of research,” *IEEE Signal Processing Magazine*, vol. 33, no. 5, pp. 59–69, 2016.
- [4] M. Damghanian, R. Olsson, M. Sjöström, A. Erdmann, and C. Perwass, “Spatial resolution in a multi-focus plenoptic camera,” in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 1932–1936, 2014.
- [5] L. Hidalgo, “Calibracion de la cámara.” <https://lvalentinah.github.io/DataPlenoptic/pages/calibracion.html>.
- [6] “Agricultura y alimentos.” <https://www.bancomundial.org/es/topic/agriculture>.
- [7] “Omicas.” <https://www.omicas.co/que-es-omicas>.
- [8] S. Zhou, T. Zhu, K. Shi, Y. Li, W. Zheng, and J. Yong, “Review of light field technologies,” *Visual Computing for Industry, Biomedicine, and Art*, vol. 4, no. 1, p. 29, 2021.
- [9] D. G. Dansereau, O. Pizarro, and S. B. Williams, “Decoding, calibration and rectification for lenselet-based plenoptic cameras,” in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1027–1034, 2013.
- [10] D. G. Dansereau, *Plenoptic Signal Processing for Robust Vision in Field Robotics*. PhD thesis, The University of Sydney, 2014.
- [11] “Raytrix - 3d light field camera technology.” <https://raytrix.de/>.
- [12] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” tech. rep., Stanford University Computer Science Tech Report CSTR, 2005.
- [13] M. Levoy and P. Hanrahan, “Light field rendering,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 31–42, ACM, 1996.
- [14] “¿qué es la visión artificial?.” <https://www.ibm.com/cloud/learn/computer-vision>.
- [15] “Open cv (open source computer vision library).” <https://opencv.org/>.
- [16] “Lytro forum.” <http://lightfield-forum.com/lytro/lytro-archive/>.