



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE CIENCIAS

**RECONOCIMIENTO DE MODOS ESPACIALES DE
LUZ ASISTIDO POR INTELIGENCIA ARTIFICIAL**

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

FÍSICO

P R E S E N T A :

TOMAS TRINIDAD ARVIZU FLORES



DIRECTOR DE TESIS:

DR. ROBERTO DE JESÚS LEÓN MONTIEL

CIUDAD UNIVERSITARIA, CD. MX. 2023



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Resumen

El presente trabajo describe la generación experimental de modos espaciales de luz afectados por medios de fase desordenados y recuperados a su salida mediante redes neuronales en el laboratorio de micro y nanofotónica del Instituto de Ciencias Nucleares de la UNAM. La investigación partió de diversos artículos sobre la generación de modos espaciales de luz usando hologramas binarios, es decir, hologramas que contienen únicamente dos valores, ceros y unos. Parte del trabajo de tesis fue describir paso a paso el método de Lee, que es un método de generación de hologramas binarios usado en gran cantidad de los artículos, pero que no se encuentra desarrollado paso a paso como en el capítulo 2: *Generación de modos espaciales: holografía digital*. Durante la investigación, se decidió usar haces Laguerre-Gauss y se ideó el montaje experimental.

Una vez concluida la investigación, se desarrolló la parte computacional previa al experimento. Se comenzó con la programación de las funciones para la creación de campos Laguerre-Gauss y para la generación de hologramas binarios dado cualquier campo. Dichas funciones sirvieron para la simulación del montaje experimental: la fuente de luz, la iluminación de los hologramas, la transformada de Fourier mediante lentes y el filtraje del primer orden de difracción buscando en todo momento recrear las condiciones experimentales. La simulación sirvió como una guía para el experimento. Sin embargo, se requirieron de diversos ajustes para llevarlo a cabo: la expansión del haz inicial de luz, la utilización de una pantalla secundaria para monitorear si se estaban desplegando correctamente los hologramas, el montaje del iris sobre una plataforma que se movía en el orden de micras, y el diseño e impresión en 3D de monturas especiales.

La parte experimental terminó con la captura en cámara de los modos espaciales deseados. Después, se retomó la parte computacional para simular la deformación y reconocimiento de los modos espaciales. Por un lado, la deformación se hizo mediante máscaras de Kolmogorov que tienen una variación aleatoria en su fase. Por otro lado, el reconocimiento se realizó con redes neuronales entrenadas a partir de los modos espaciales deformados, arrojando resultados que demuestran que estos algoritmos pueden ser utilizados en protocolos de comunicación a través de medios desordenados como las fibras ópticas.

Índice general

1. Introducción	6
2. Generación de modos espaciales: holografía digital	8
2.1. Codificación de campos escalares usando hologramas de amplitud binarios	8
2.2. Haces Laguerre-Gauss: descripción matemática	13
3. Generación experimental de los haces Laguerre-Gauss	16
3.1. Planteamiento del problema	16
3.2. Dispositivo Digital de Microespejos	16
3.3. Generación de los hologramas de amplitud	17
3.4. Empleo del DMD	17
3.5. Montaje experimental	18
3.6. Resultados	26
4. Identificación de modos espaciales usando inteligencia artificial	28
5. Conclusiones	33
A. Códigos en MATLAB	36
A.1. Generación de haces Laguerre-Gauss	36
A.2. Generación de hologramas	37
A.3. Simulación: reconstrucción de haces Laguerre-Gauss	37
A.4. Experimento: despliegue de hologramas y captura de imágenes	39
A.4.1. Configuración de la cámara	39
A.4.2. Configuración de la ventana de MATLAB en el DMD y la pantalla auxiliar	40
A.4.3. Captura de las imágenes en la cámara	41
A.5. Reconocimiento de haces deformados con inteligencia artificial	42
A.5.1. Deformación de los haces mediante máscaras de Kolmogorov	42
A.5.2. Procesamiento de imágenes	44
A.5.3. Entrenamiento de la red neuronal	45
B. Manipulación del módulo de evaluación DLP3310	47
B.1. Desmontaje y extracción	47
B.2. Impresión de las monturas auxiliares	50
C. Modos Laguerre-Gauss experimentales y computacionales	53

Índice de figuras

2.1.	Corte transversal de la binarización de la onda a lo largo del eje x . Si la suma de cosenos toma un valor positivo a lo largo de la dirección de propagación z , entonces la binarización toma el valor 1 y si el campo toma un valor negativo, entonces la binarización toma el valor 0.	10
2.2.	Ejemplos de la distribución de intensidad en diferentes haces Laguerre-Gauss. Los campos están denotados de la forma LG_{lp} donde l es el índice azimutal y p es el índice radial.	15
3.1.	Partes del DLP LightCrafter Display 3310 EVM. Imagen extraída de la guía de usuario [1, Figura 1-1].	18
3.2.	Interfaz de usuario del DLP 3310 que se usó para controlar algunas funciones del módulo durante el experimento.	19
3.3.	El disipador, el DMD, el FFC con su tarjeta en un extremo y la tarjeta secundaria. Elementos por separado 3.3a. Elementos unidos mediante la montura visto lateralmente 3.3b.	19
3.4.	Proceso de montaje común a todos los elementos ópticos ejemplificado con un bloque de plástico. Se muestra el poste atornillado a la base en 3.4a y se muestra el elemento atornillado al poste en 3.4b.	20
3.5.	Fuente láser modelo <i>S1FC660</i> con una longitud de onda de 660 nm conectada a un colimador optimizado para una longitud de onda de 633 nm. Frente al colimador están dos lentes que sirven para expandir el haz a un tamaño que cubre de mejor forma la superficie del DMD.	21
3.6.	Alineaciones de cerca del colimador y el lente plano-convexo. Se muestra como a la regla se pegó una tarjeta graduada para poder ver mejor el láser en el fondo blanco.	22
3.7.	(a) Vista esquemática del montaje experimental, donde L_1 y L_2 son las lentes que expanden el haz y sus distancias focales son f_1 y f_2 , respectivamente; mientras que L_3 y L_4 son las lentes transformadoras de Fourier y sus distancias focales son f_3 y f_4 , respectivamente. (b) Imagen del holograma desplegado en el DMD.	23
3.8.	Imagen del DMD montado en las bases construidas mediante impresión 3D	24
4.1.	Matrices de confusión de las predicciones de la red neuronal contra las salidas reales. La figura 4.1a muestra los resultados antes de que la red fuera entrenada y la figura 4.1b muestra los resultados luego de que esta fue entrenada.	31

B.1.	Diferentes vistas del DLP LightCrafter Display 3310 EVM en su estado inicial. En la vista frontal B.1a se observa el proyector. En la vista lateral izquierda B.1b se pueden observar dos de los tres ventiladores. En la vista inferior B.1c se puede observar la placa que más adelante se manipulará para sujetarse a la mesa óptica. En la vista trasera se puede observar el tercer ventilador, los conectores a los tres leds y los conectores a los motores (etiquetados del uno al tres). En la vista lateral derecha se puede observar el cable plano flexible y el disipador térmico que se conectan al DMD. En la vista superior se puede observar a la <i>Display Board</i> que contiene, entre otros chips, al controlador del DMD.	48
B.2.	Desmontaje de los ventiladores del DLP LightCrafter Display 3310 EVM. Se señalan los tornillos que hay que remover en las figuras B.2a, B.2b y B.2c para liberar la montura de los ventiladores tal y como se muestra en la figura B.2d. Se muestran en B.2e los cables que hay que desconectar para separar totalmente la montura de los ventiladores y quede por separado como en la figura B.2f.	49
B.3.	Pasos finales del desmontaje del módulo de evaluación. La imagen de la izquierda muestra cómo se desconectó el cable plano flexible y la imagen del lado derecho muestra el subsistema <i>Light Engine</i> con el lugar donde se encontraba el DMD encerrado en un recuadro azul. . .	50
B.4.	Herramientas usadas para el diseño e impresión de las monturas auxiliares. Del lado derecho se muestra el vernier digital que se uso para medir las dimensiones, en medio se muestra la impresora 3D donde se imprimieron las monturas que se muestran en la figura y del lado izquierdo se muestra un ejemplo de la manipulación del DMD con pinzas de plástico. Todas las herramientas fueron necesarias para la creación de las monturas mostradas en la figura B.5.	51
B.5.	Diseño de las monturas en AutoCAD del lado izquierdo y las monturas ya impresas del lado derecho. Se muestran la montura que unía a los componentes del DMD, la plataforma con la cual se fijaba el módulo de evaluación a la mesa óptica y uno de los bloques que afianzaban la montura del DMD a los postes, en ese orden.	52

Capítulo 1

Introducción

Durante décadas, experimentalmente se ha logrado generar una gran variedad de modos espaciales entre los que están los Laguerre-Gauss, los Airy y los Bessel, por dar algunos ejemplos. En años recientes, se ha mostrado que usando la generación de estos modos se puede codificar información en bits mediante fotones individuales, lo cual ha llevado a su vez a la creación de nuevos modos de encriptación y protocolos de comunicación segura [2] [3] [4]. De hecho, esta tecnología se ha convertido en la llave para aplicaciones importantes tales como la creación de sistemas de comunicación más seguros, la imagenología de alta resolución y la micromanipulación, todas tecnologías cuánticas con un potencial impacto en diversas áreas de la óptica. Los métodos propuestos por la comunidad científica para generar modos espaciales suelen probarse en el espacio libre, incluso aquellos como el método de Lee del que ya se tiene registro desde finales de los años 70 y que es ampliamente usado [5] [6] [7]. Los modos espaciales al pasar por medios con dispersión son susceptibles a modificaciones de fase. Se ha solucionado esta dificultad para las fibras de vórtice, que son fibras ópticas especiales, pero estas soluciones tienen un alto costo y están limitadas a distancias de alrededor de un kilómetro [8] [9]. Este trabajo propone una posible solución a la propagación de modos espaciales a través de canales ruidosos. El método se basa en hacer pasar haces Laguerre-Gauss por un vidrio esmerilado simulado, cuya distribución de intensidad original es reconocida a través de algoritmos de inteligencia artificial. Este trabajo sienta las bases para un proyecto más ambicioso, el cual consistirá en el desarrollo de nuevos protocolos de menor costo que usen fibras ópticas multimodo de propósito general y que funcionen en distancias necesarias para una instalación de fibra óptica desde algunos kilómetros hasta cientos de ellos.

Los medios con desorden deforman los modos espaciales de luz de tal manera que no son fácilmente reconocibles a su salida. Sin embargo, el desarrollo reciente de la inteligencia artificial hace del reconocimiento de imágenes una tarea más sencilla siempre que existan patrones en los datos. Se describe en esta tesis cómo es que se lograron entrenar algoritmos de aprendizaje automático supervisado con imágenes de los modos espaciales distorsionados por el medio con dispersión para que en base a éstas imágenes deformadas, la red neuronal reconociera el modo original. Las redes entrenadas inclusive pueden ser usadas para detectar individualmente diferentes modos espaciales enviados en secuencia y separados por intervalos de tiempo en un orden aleatorio, por lo cual podrían ser usadas en protocolos más seguros para el envío de información a través de medios tales como la fibra óptica.

La presente tesis está organizada de forma que se describen a priori los fundamentos teóricos tanto de los modos espaciales a codificar como del método de generación de los hologramas. Mientras que otros conceptos teóricos como la propagación de Fresnel y las máscaras de Kolmogorov son aclarados de manera suficiente cuando es necesario.

El capítulo 2 describe cómo se puede plasmar la información de un campo en un holograma binario partiendo de su amplitud y de su fase. Debido al método de construcción del mismo holograma se llega a la conclusión de que su transformada de Fourier es una suma de órdenes de difracción y que dentro de cualquiera de estos órdenes se puede codificar el campo objetivo. Después, en el mismo capítulo se pasa a la descripción de los campos objetivo que son un tipo de campos a los que se les ha dado especial atención porque transportan un momento angular orbital alrededor de su eje, los haces Laguerre-Gauss. La descripción matemática de los haces que se van a generar y del método por el que se generarán fundamenta los resultados obtenidos tanto experimentalmente como computacionalmente.

El capítulo 3 comienza explicando el funcionamiento del dispositivo más importante del montaje experimental que es conocido como *Dispositivo Digital de Micro-espejos* y por qué se pueden desplegar los hologramas en él. Una vez hecha esta explicación se procede a detallar todo el montaje experimental paso por paso, desde la expansión del haz de luz proveniente del láser, pasando por el despliegue de los hologramas hasta llegar a la captura de los haces en la cámara.

Los resultados experimentales obtenidos en el capítulo anterior sugieren que se pueden generar modos Laguerre-Gauss y deformarlos al hacerlos pasar por un medio desordenado antes de ser capturados por la cámara. El capítulo 4 demuestra computacionalmente que los haces deformados pueden ser reconocidos mediante una inteligencia artificial. Se describe el proceso de entrenamiento de la inteligencia artificial que abarca la generación de los haces deformados, el tratamiento de las imágenes, el diseño de la red neuronal y termina con el desempeño de la misma.

Finalmente, el capítulo 5 concluye la presente tesis a partir de los resultados computacionales y experimentales de los capítulos anteriores y plantea algunos experimentos que se pueden realizar tomando a este como base. Entre los futuros trabajos se incluyen probar experimentalmente el reconocimiento de los haces ya deformados y transmitir mensajes que no solo sean caracteres.

Capítulo 2

Generación de modos espaciales: holografía digital

2.1. Codificación de campos escalares usando hologramas de amplitud binarios

El problema fundamental que resuelve la holografía es el de grabar y después reconstruir tanto la amplitud como la fase de una onda que llega de un objeto iluminado coherentemente. La holografía se ha vuelto digital con el desarrollo de nuevas tecnologías de proyección desde el siglo pasado. Estas tecnologías requieren la digitalización de los hologramas, es por ello que a continuación se detalla el método de Lee que es un método para generar hologramas binarios.

Un holograma convencional está formado a partir de la interferencia entre dos ondas: una onda plana $R \exp(-j2\pi x/x_0)$ con R como su amplitud constante y x_0 como su período en la dirección x ; y una onda objetivo $\mathcal{A}(x, y) \exp[j\varphi(x, y)]$ con $\mathcal{A}(x, y)$ como su amplitud y $\varphi(x, y)$ como su fase, mientras que en ambas ondas j denota a la unidad imaginaria. Su transmitancia es proporcional a la intensidad de la interferencia,

$$\begin{aligned} t(x, y) &\propto \left| R \exp\left(-\frac{j2\pi x}{x_0}\right) + \mathcal{A}(x, y) \exp[j\varphi(x, y)] \right|^2, \\ &\propto R^2 + A^2(x, y) + 2R\mathcal{A}(x, y) \cos\left[\frac{2\pi x}{x_0} + \varphi(x, y)\right]. \end{aligned} \quad (2.1)$$

El tercer sumando en la ecuación (2.1) desempeña la reconstrucción de la onda objetivo. Esta se puede binarizar estableciendo los valores no negativos y negativos de la función como uno y cero, respectivamente. De momento y para simplificar el desarrollo, solo se codificará la información de la fase de la onda objetivo $\varphi(x, y)$ mientras que su amplitud $\mathcal{A}(x, y)$ se contemplará más adelante. La binarización se escribe como

$$T(x, y) = \begin{cases} 1 & \text{para } \cos[2\pi x/x_0 + \varphi(x, y)] \geq 0 \\ 0 & \text{de lo contrario} \end{cases}, \quad (2.2)$$

que valiéndose de la función sgn se puede escribir de la siguiente forma

$$T(x, y) = \frac{1}{2} + \frac{1}{2} \operatorname{sgn} \left\{ \cos \left[\frac{2\pi x}{x_0} + \varphi(x, y) \right] \right\}. \quad (2.3)$$

El centro de las franjas de la función de transmitancia, denotada por $T(x, y)$, están en aquellos valores en que $\cos[2\pi x/x_0 + \varphi(x, y)] = 1$, es decir, donde $2\pi x/x_0 + \varphi(x, y) = 2m\pi$ con $m \in \mathbb{Z}$. Con la finalidad de introducir más adelante la información de la amplitud, en lugar de establecer al cero como el umbral en la binarización se escoge una función como un *bias* de la forma $\cos[\pi w(x, y)]$ donde $w(x, y)$ es el término que involucra a la amplitud y cuya forma será definida más adelante. La suma de la función coseno anterior y el bias tiene ceros que en la binarización coinciden con los bordes de los pulsos, es decir, que los bordes están en los puntos donde la suma de cosenos es cero

$$\cos \left[\frac{2\pi x}{x_0} + \varphi(x, y) \right] - \cos[\pi w(x, y)] = 0, \quad (2.4)$$

despejando a x se obtiene la siguiente condición

$$x = mx_0 - \frac{\varphi(x, y)x_0}{2\pi} \mp \frac{w(x, y)x_0}{2}. \quad (2.5)$$

La binarización se puede ver como un conjunto de pulsos rectangulares, donde cada uno tiene un punto inicial (a), un punto final (b), un ancho ($c = b - a$) y un centro (d) que se describen matemáticamente en las siguientes ecuaciones

$$a = mx_0 - \varphi(x, y)x_0/(2\pi) - w(x, y)x_0/2, \quad (2.6)$$

$$b = mx_0 - \varphi(x, y)x_0/(2\pi) + w(x, y)x_0/2, \quad (2.7)$$

$$c = x_0 w(x, y), \quad (2.8)$$

$$d = mx_0 - \varphi(x, y)x_0/(2\pi) = x_0[m - \varphi(x, y)/(2\pi)]. \quad (2.9)$$

En base a las características de cada pulso (2.6), (2.7), (2.8) y (2.9) se concluye que cada pulso es de la forma

$$\operatorname{rect} \left(\frac{x - x_0[m - \varphi(x, y)/(2\pi)]}{w(x, y)x_0} \right) = \begin{cases} 1 & \text{para } |x - x_0[m - \varphi(x, y)/(2\pi)]| \leq 0 \\ 0 & \text{para cualquier otro caso} \end{cases}. \quad (2.10)$$

El holograma binario, suponiendo que se extiende hasta el infinito se puede expresar como la suma de todos los pulsos rectangulares. Haciendo el cambio de variable $p(x, y) \equiv \varphi(x, y)/(2\pi)$ en la ecuación (2.10), la *función de transmitancia binaria* queda de la siguiente forma

$$T(x, y) = \sum_{m=-\infty}^{\infty} \operatorname{rect} \left[\frac{x - (m + p)x_0}{wx_0} \right]. \quad (2.11)$$

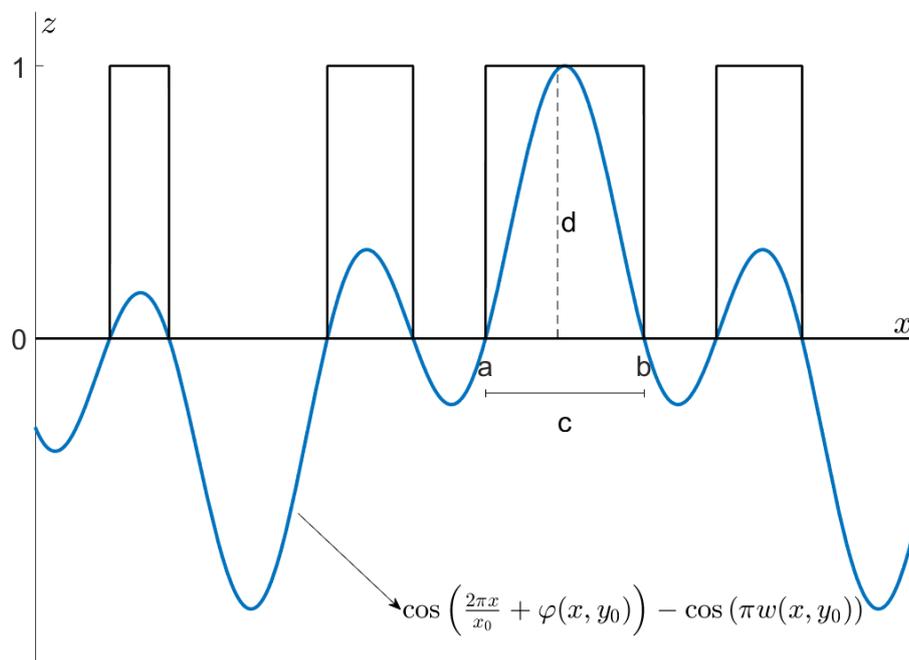


Figura 2.1: Corte transversal de la binarización de la onda a lo largo del eje x . Si la suma de cosenos toma un valor positivo a lo largo de la dirección de propagación z , entonces la binarización toma el valor 1 y si el campo toma un valor negativo, entonces la binarización toma el valor 0.

La representación en series de Fourier de los pulsos binarios son una serie de pulsos rectangulares. Se hace un cambio de variable $z = (x - px_0)/(wx_0)$ para desarrollar en series de Fourier más fácilmente. La variable original $x = zwx_0 + px_0$ entonces se puede sustituir en la función de transmitancia de la siguiente forma

$$\begin{aligned}
 T &= \sum_{m=-\infty}^{\infty} \text{rect} \left[\frac{zwx_0 + px_0 - (m+p)x_0}{wx_0} \right], \\
 &= \sum_{m=-\infty}^{\infty} \text{rect} \left[\frac{zwx_0}{wx_0} - \frac{mx_0}{wx_0} \right], \\
 &= \sum_{m=-\infty}^{\infty} \text{rect} \left[z - \frac{m}{w} \right]. \tag{2.12}
 \end{aligned}$$

Cada uno de los sumandos en la ecuación (2.12) son pulsos unitarios centrados en m/w con un ancho unitario y un período $L = (m+1)/w - m/w = 1/w$, todo esto en la nueva variable z . Para continuar con la expansión de Fourier se calculará en su forma compleja

$$f(\xi) = \sum_{m=-\infty}^{\infty} A_m \exp \left[j \frac{2\pi m \xi}{L} \right], \tag{2.13}$$

donde los coeficientes de Fourier son

$$A_m = \frac{1}{L} \int_{-L/2}^{L/2} f(\xi) \exp \left[-j \frac{2\pi m \xi}{L} \right] d\xi. \tag{2.14}$$

El bias introduce una fluctuación en la amplitud dentro de los distintos coeficientes complejos de la expansión en series de Fourier. Desarrollando (2.14) para la función de transmitancia binaria se obtiene

$$\begin{aligned}
 A_m &= \frac{1}{L} \int_{-L/2}^{L/2} \left\{ \sum_{m=-\infty}^{\infty} \text{rect} \left[z - \frac{m}{w} \right] \right\} \exp \left[-j \frac{2\pi m z}{L} \right] dz, \\
 &= \frac{1}{L} \int_{-L/2}^{L/2} \text{rect}[z] \exp \left[-j \frac{2\pi m z}{L} \right] dz, \\
 &= \frac{1}{L} \int_{-1/2}^{1/2} \exp \left[-j \frac{2\pi m z}{L} \right] dz, \\
 &= \frac{1}{L} \frac{1}{-j \frac{2\pi m}{L}} \left\{ \exp \left[-j \frac{2\pi m z}{L} \right] dz \right\}_{-1/2}^{1/2}, \\
 &= \frac{1}{\pi m} \left\{ \frac{-\exp \left[-j \frac{\pi m}{L} \right] + \exp \left[j \frac{\pi m}{L} \right]}{2j} \right\}, \\
 &= \frac{\sin(\pi m w)}{\pi m}. \tag{2.15}
 \end{aligned}$$

Usando los coeficientes de Fourier (2.15) en la ecuación (2.13), la función de transmitancia binaria queda como

$$\begin{aligned}
 T(x, y) &= \sum_{m=-\infty}^{\infty} \left[\frac{\sin(\pi m w)}{\pi m} \right] \exp \left[j \frac{2\pi m [(x - px_0)/(wx_0)]}{1/w} \right], \\
 &= \sum_{m=-\infty}^{\infty} \left[\frac{\sin(\pi m w)}{\pi m} \right] \exp \left\{ jm \left[\frac{2\pi x}{x_0} + 2\pi p \right] \right\}, \\
 &= \sum_{m=-\infty}^{\infty} \left[\frac{\sin(\pi m w)}{\pi m} \right] \exp \left\{ jm \left[\frac{2\pi x}{x_0} + \varphi \right] \right\}, \\
 &= \sum_{m=-\infty}^{\infty} \left[\frac{\sin(\pi m w)}{\pi m} \right] \exp [jm\varphi] \exp \left[jm \frac{2\pi x}{x_0} \right]. \tag{2.16}
 \end{aligned}$$

La expansión en series de Fourier es útil porque su transformada de Fourier es una suma de deltas de Dirac multiplicadas por los coeficientes de la expansión como se muestra en la ecuación (2.17). Recapitulando, una vez que se codifica el campo mediante la función de transmitancia binaria se debe de hacer la transformada de Fourier de esta última. La transformada se puede conseguir en el campo lejano, en particular, mediante lentes transformadoras de Fourier. [10, Cap. 6]. Matemáticamente la transformada es

$$\begin{aligned}
 T_F(\xi, y) &= \mathcal{F}\{T(x, y)\}, \\
 &= \mathcal{F} \left\{ \sum_{m=-\infty}^{\infty} \left[\frac{\sin(\pi m w)}{\pi m} \right] \exp [jm\varphi] \exp \left[jm \frac{2\pi x}{x_0} \right] \right\}, \\
 &= \sum_{m=-\infty}^{\infty} \left[\frac{\sin(\pi m w)}{\pi m} \right] \exp [jm\varphi] \mathcal{F} \left\{ \exp \left[j \frac{m 2\pi}{x_0} x \right] \right\}, \\
 &= \sum_{m=-\infty}^{\infty} \left[\frac{\sin(\pi m w)}{\pi m} \right] \exp [jm\varphi] \delta \left(\xi - \frac{m}{x_0} \right). \tag{2.17}
 \end{aligned}$$

Una vez con la transformada de la función de transmitancia se pueden usar distintos métodos para grabar la información de la amplitud de la onda objetivo $\mathcal{A}(x, y) \exp [j\varphi(x, y)]$ en hologramas binarios. Un enfoque sencillo consiste en considerar el primer coeficiente o primer orden de la expansión de Fourier $m = 1$, pues este factor incluye directamente al bias $w(x, y)$. La ecuación 2.18 muestra la idea fundamental detrás del método de Lee que consiste en establecer la amplitud de la onda objetivo dentro del bias. Tanto la amplitud como la fase, entonces, quedan grabadas por las posiciones y anchos a lo largo de la frecuencia portadora x_0 .

$$\begin{aligned}
 \frac{\sin[\pi w(x, y)]}{\pi} &= \mathcal{A}(x, y), \\
 \Rightarrow w(x, y) &= \frac{1}{\pi} \arcsin[\mathcal{A}(x, y)]. \tag{2.18}
 \end{aligned}$$

Nótese que la onda objetivo está normalizada, entonces el término π dentro del arcsin de la función auxiliar w se descarta. La función de transmitancia binaria entonces queda como sigue

$$T(x, y) = \begin{cases} 1 & \text{para } \cos[2\pi x/x_0 + \varphi(x, y)] - \cos[\pi w(x, y)] \geq 0 \\ 0 & \text{de lo contrario} \end{cases}, \quad (2.19)$$

que equivalentemente se puede escribir:

$$T(x, y) = \frac{1}{2} + \frac{1}{2} \operatorname{sgn} \left\{ \cos \left[\frac{2\pi x}{x_0} + \varphi(x, y) \right] - \cos[\pi w(x, y)] \right\}. \quad (2.20)$$

Definiendo un par de funciones auxiliares: (i) la función que está relacionada con la fase de la onda objetivo $p(x, y) \equiv \varphi(x, y)/\pi$ que es diferente al cambio de variable hecho en la ecuación (2.11) donde $p(x, y) \equiv \varphi(x, y)/(2\pi)$ y (ii) la función que está relacionada con la amplitud de la onda objetivo $w(x, y) \equiv \arcsin[\mathcal{A}(x, y)]/\pi$, y sustituyendo en la ecuación (2.20) se obtiene la forma final de la función de transmitancia binaria

$$T(x, y) = \frac{1}{2} + \frac{1}{2} \operatorname{sgn} \left\{ \cos \left[\frac{2\pi x}{x_0} + \pi p(x, y) \right] - \cos[\pi w(x, y)] \right\}. \quad (2.21)$$

Se puede codificar cualquier onda siempre que el campo no contenga singularidades y su amplitud esté normalizada para tener como máximo a la unidad. De las más ampliamente codificadas hasta ahora está un conjunto completo y alternativo de soluciones a la ecuación de Helmholtz que es conocido como haces Laguerre-Gauss y son ampliamente usados en la holografía. [6] [11] [10, Cap. 2]

2.2. Haces Laguerre-Gauss: descripción matemática

A continuación se define la amplitud $U_{l,p}$ de un modo Laguerre-Gauss con un índice azimutal l y un índice radial p en la ecuación (2.22). Dichos modos se denotan como LG_{lp} más adelante. Los modos Laguerre-Gauss son obtenidos al escribir la aproximación paraxial de la ecuación de Helmholtz en coordenadas cilíndricas (r, θ, z) y resolviendo con la técnica de separación de variables en r y θ . [12, Cap. 3]

$$U_{l,p}(r, \theta, z) = A_{l,p} \left[\frac{W_0}{W(z)} \right] \left[\frac{\sqrt{2}r}{W(z)} \right]^{|l|} L_p^{|l|} \left(\frac{2r^2}{W^2(z)} \right) \exp \left[-\frac{r^2}{W^2(z)} \right] \times \exp \left[jkz + jk \frac{r^2}{2R(z)} + jl\theta - j(|l| + 2p + 1)\psi(z) \right]. \quad (2.22)$$

El *polinomio generalizado de Laguerre*, también conocido como *polinomio asociado de Laguerre* de orden (l, p) está denotado por $L_p^{|l|}$. La mitad del ancho de la amplitud del perfil es $W(z)$ y el radio de curvatura del frente de onda es $R(z)$. El haz Gaussiano se obtiene al simplificar $l = m = 0$, siendo así el haz Laguerre-Gauss de menor orden. La fase de Gouy $\psi(z)$ sufre un cambio de signo cuando pasa a través

del foco, va desde $-\pi/2$ hasta $\pi/2$. Estos modos están normalizados por constantes $A_{l,p}$ que igualan la potencia total.

$$A_{l,p} = \sqrt{\frac{2p!}{\pi(p+|l|)!}}, \quad (2.23)$$

$$W_0 = \sqrt{\frac{\lambda z_0}{\pi}}, \quad (2.24)$$

$$R(z) = z \left[1 + \left(\frac{z_0}{z} \right)^2 \right], \quad (2.25)$$

$$\psi(z) = \arctan \left(\frac{z}{z_0} \right), \quad (2.26)$$

$$W(z) = W_0 \sqrt{1 + \left(\frac{z}{z_0} \right)^2}. \quad (2.27)$$

En el foco, un haz Laguerre-Gauss tiene una cintura (*waist*, en inglés) de radio W_0 . En la expresión (2.28) para z_0 , es decir, la longitud de Rayleigh, n es el índice de refracción del medio en que se propagan los haces Laguerre-Gauss y λ es la longitud de onda

$$z_0 = \frac{\pi W_0^2 n}{\lambda}. \quad (2.28)$$

Una propiedad interesante de estos modos surge del término de la fase que es proporcional al ángulo azimutal θ . La presencia de este término implica que, cuando el índice azimutal es positivo $l > 0$, el frente de onda del haz tiene un giro helicoidal o *helical twist* (en inglés), este giro aumenta conforme aumenta el índice azimutal. Se puede demostrar que estos haces poseen *momento angular orbital*, en consecuencia pueden impartir torque al objeto que golpeen. Siendo el índice azimutal no nulo $l \neq 0$, el haz tiene intensidad cero en el centro $r = 0$ y un patrón de intensidades en anillos. Cuando la fase circula alrededor de un cero de intensidad se tiene lo que se conoce como “optical vortex” o “phase vortex” [10, Cap. 4] [12, Cap. 3] [13]. A modo de ejemplo, en la fig. 2.2, se muestran haces Laguerre-Gauss de distintos órdenes. Los haces fueron generados por el autor de la tesis.

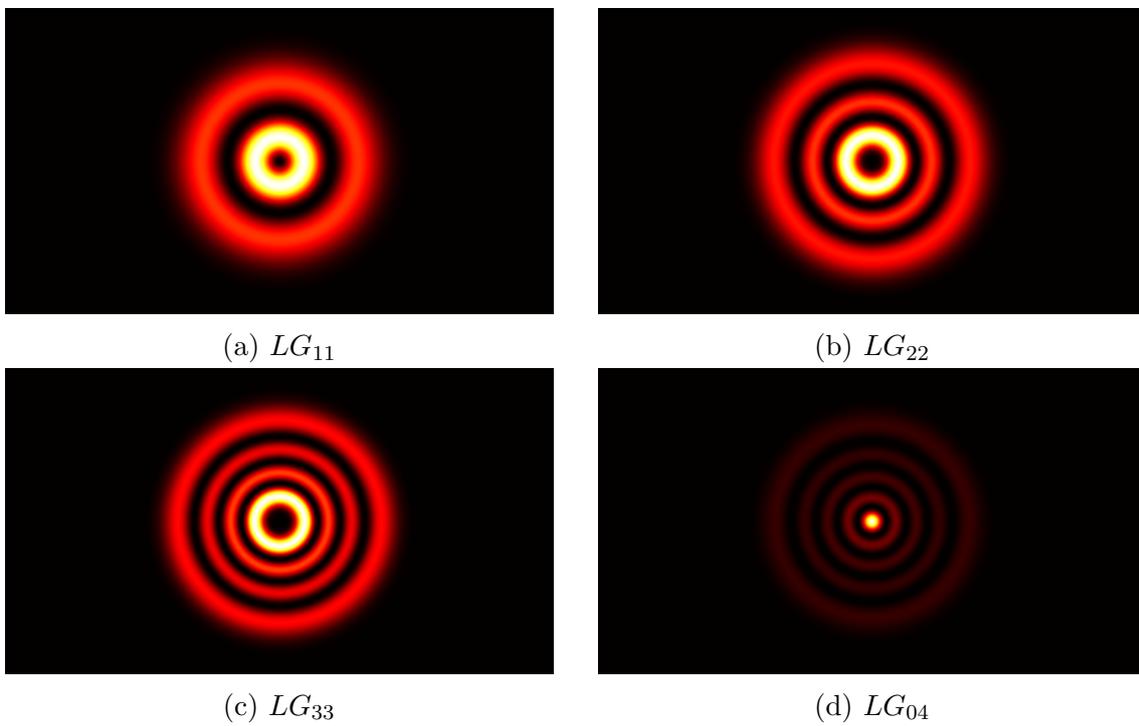


Figura 2.2: Ejemplos de la distribución de intensidad en diferentes haces Laguerre-Gauss. Los campos están denotados de la forma LG_{lp} donde l es el índice azimutal y p es el índice radial.

Capítulo 3

Generación experimental de los haces Laguerre-Gauss

3.1. Planteamiento del problema

Al implementar sistemas de comunicación, la codificación de los mensajes debe ser rápida y, por tanto, la modulación en amplitud es más eficiente que la modulación en fase. Conviene usar entonces un DMD (por sus siglas en inglés, **D**igital **M**icromirror **D**evice) que tiene la capacidad de trabajar en el orden de los MHz mientras que el SLM (por sus siglas en inglés, **S**patial **L**ight **M**odulator) tiene una frecuencia de trabajo de alrededor de 60 Hz. Se reportan las técnicas de codificación de amplitud y fase en un haz de luz usando un DMD. Dichas técnicas logran generar haces con momento angular orbital tales como los Laguerre-Gauss.

3.2. Dispositivo Digital de Microespejos

Desde hace casi dos décadas, la tecnología de *Procesamiento Digital de Luz* o DLP (por sus siglas en inglés, **D**igital **L**ight **P**rocessing) de la compañía Texas Instruments ha hecho avances significativos en tecnologías de la proyección como su Dispositivo Digital de Microespejos o DMD. El DMD es un arreglo de miles de microespejos (píxeles) que tienen una dirección individual y se pueden inclinar independientemente como un «interruptor de luz» basado en semiconductores.

Desde que el DMD tuvo éxito como modulador espacial de luz, docenas de nuevas aplicaciones del DMD hoy en día tienen mayor disponibilidad para los desarrolladores de estas tecnologías. Entre las múltiples aplicaciones está la generación de varios haces de luz de orden superior como los Laguerre-Gauss (LG), Bessel y Airy mediante métodos holográficos.

El desarrollo de esta tecnología comienza con el Deformable Mirror Device en 1977, siendo hasta 1987 que la tecnología evolucionó a un Dispositivo Digital de Microespejos bi-estable. Los DMD están hechos para rotar un ángulo, por ejemplo 12° , entonces los dos estados de los microespejos son a $+12^\circ$ y a -12° . El cambio entre los estados se hace aplicando un voltaje en la dirección de los electrodos para crear una atracción electrostática de tal forma que cada espejo o bien se queda en su posición actual o rápidamente rota al estado opuesto.

Uno de los principales campos en que el DMD tiene potenciales aplicaciones es

la holografía. La holografía es un conjunto de técnicas que numéricamente graba y reconstruye frentes de onda enteros y dentro de estas técnicas está la holografía digital que es un gran campo de interés en la metrología, proyección, almacenamiento de datos y la autenticación de los mismos. El almacenamiento holográfico de datos es bastante interesante desde el punto de vista comercial y científico. Durante muchos años la industria del almacenamiento ha visto esta aplicación con el potencial de lograr una densidad de almacenamiento y tasa de datos más allá del límite de las tecnologías ópticas y magnéticas convencionales [14] [15].

3.3. Generación de los hologramas de amplitud

La generación de los hologramas de amplitud que se proyectan en el DMD para recuperar un campo requiere de la previa creación de dicho campo. La generación de los campos Laguerre-Gauss se puede hacer a partir de la ecuación (2.22), pero conviene simplificar el modelo al generar el modo en su foco, es decir, en $z = 0$ y para lo cual hay que considerar los siguientes puntos: (i) el radio de la cintura es una constante $W(z = 0) = W_0$ que se puede elegir a conveniencia, (ii) el radio de curvatura es infinito $R(z = 0) = \infty$ y (iii) la fase de Gouy se anula dejando únicamente la contribución azimutal, es decir, el momento angular orbital. Cabe mencionar que estas consideraciones no afectan el resultado obtenido en el experimento.

$$U_{l,p}(r, \theta, 0) = \sqrt{\frac{2p!}{\pi(p + |l|)!}} \left[\frac{\sqrt{2}r}{W_0} \right]^{|l|} L_p^{|l|} \left(\frac{2r^2}{W_0^2} \right) \exp \left[-\frac{r^2}{W_0^2} \right] \exp [jl\theta] \quad (3.1)$$

La programación de los haces Laguerre-Gauss en MATLAB según la ecuación simplificada (3.1) se describe en las primeras secciones del apéndice A y, como nota, solo se necesitó la instalación adicional del *Symbolic Math Toolbox*. Por un lado, se generaron y almacenaron haces Laguerre-Gauss para índices azimutales L desde cero hasta tres, para índices radiales P desde uno hasta cuatro y cinturas W_0 desde 0.1 mm hasta 1.0 mm con ayuda de ciclos for y de las funciones descritas en la sección A.1. Por otro lado, en la sección A.2 se muestra la programación de la ecuación (2.21) que en principio puede grabar la información de cualquier campo en un holograma por el método de Lee. Los hologramas de amplitud pueden ser mostrados en cualquier pantalla o proyector, como es el caso del módulo de evaluación.

3.4. Empleo del DMD

El DLP[®] LightCrafter[™] Display 3310 Evaluation Module (EVM) de Texas Instruments se muestra en la figura 3.1 donde se puede observar que el módulo está armado de tal forma que el DMD no puede usarse directamente y tampoco se ve a simple vista. Se desarmó parcialmente este módulo formado por tres subsistemas. El subsistema *Light Engine* es capaz de emitir 300 lúmenes e incluye los elementos ópticos, los leds rojo, azul y verde, y un DMD Full-HD. El subsistema *Display Board* incluye al conjunto de chips DLP que está compuesto de los controladores DLPC3437, el controlador DLPA3000 PMIC y el *LED Driver*. Por último, el subsistema *System Board* incluye al receptor MCP430, al receptor ITE HDMI, al

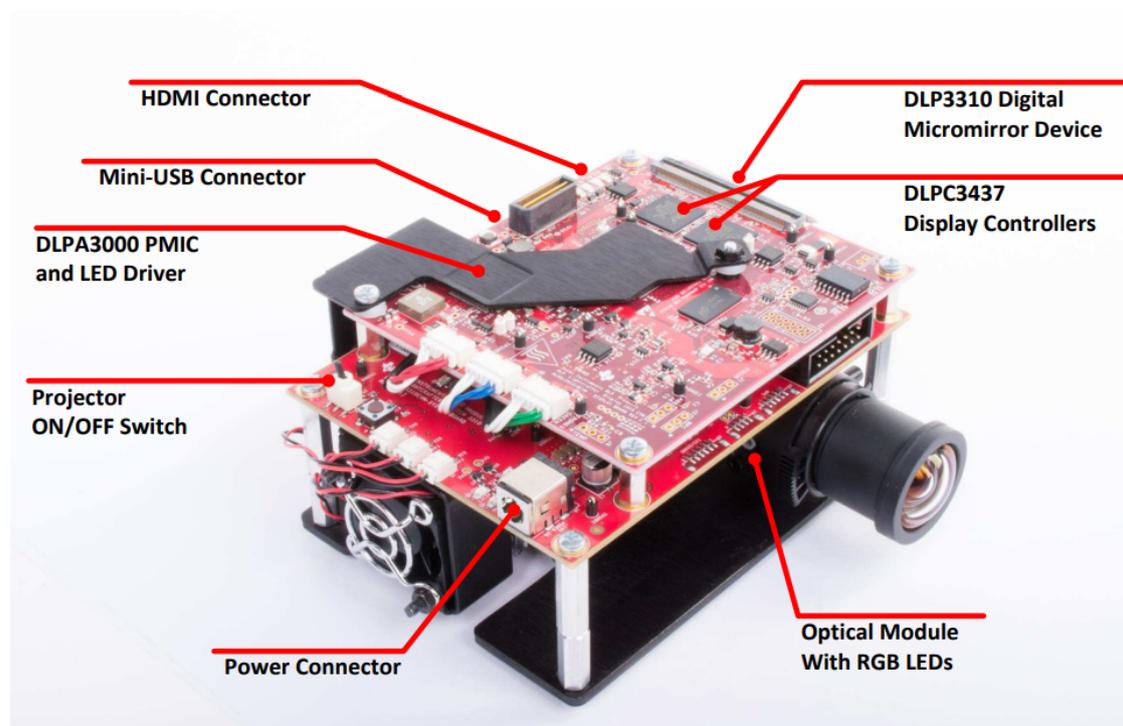


Figura 3.1: Partes del DLP LightCrafter Display 3310 EVM. Imagen extraída de la guía de usuario [1, Figura 1-1]

controlador USB Serial Bridge y varios conectores para entradas externas (HDMI, USB y demás). El objetivo era dejar expuesto el subsistema de la *Light Engine* ya que ahí estaba el DLP3310 (0.33-inch Full-HD DMD) y los leds rojo, azul y verde de Osram que funcionan en conjunto, es decir, si los leds dejan de funcionar entonces el DMD también deja de funcionar y viceversa. El DLP3310 es un DMD que mide diagonalmente 0.33 pulgadas y sus microespejos son de aluminio. Este arreglo de píxeles es de 768 filas por 1368 columnas en una disposición de rejilla cuadrada.

La extracción del DMD se explica a detalle en la sección B.1, una vez extraído se usó una fuente de poder de 19 V y 3.42 A para conectar el módulo a un enchufe de la mesa óptica. También se conectó la entrada Mini-USB del módulo a la entrada USB de la computadora portátil para controlar el dispositivo con ayuda del software DLP LightCrafter Display EVM GUI tool v7.0 Setup del que se muestra una captura de pantalla en la figura 3.2. Cabe mencionar que se tuvo que diseñar un soporte auxiliar para sostener el DMD una vez extraído del módulo, este se muestra en la figura 3.3b y se explica más de él en el apéndice B. En la tabla 3.1 se especifican las principales características del DMD.

3.5. Montaje experimental

El arreglo experimental se montó sobre una mesa óptica *Newport*. Usando los orificios de la mesa óptica se atornillaron las bases modelo THORLABS y dentro de cada base se atornilló el poste a diferentes alturas dependiendo del elemento óptico como se muestra en la figura 3.4a para después atornillar sobre el poste el elemento óptico correspondiente como se muestra en la figura 3.4b.

El arreglo experimental comienza con el láser modelo *S1FC660* que se muestra

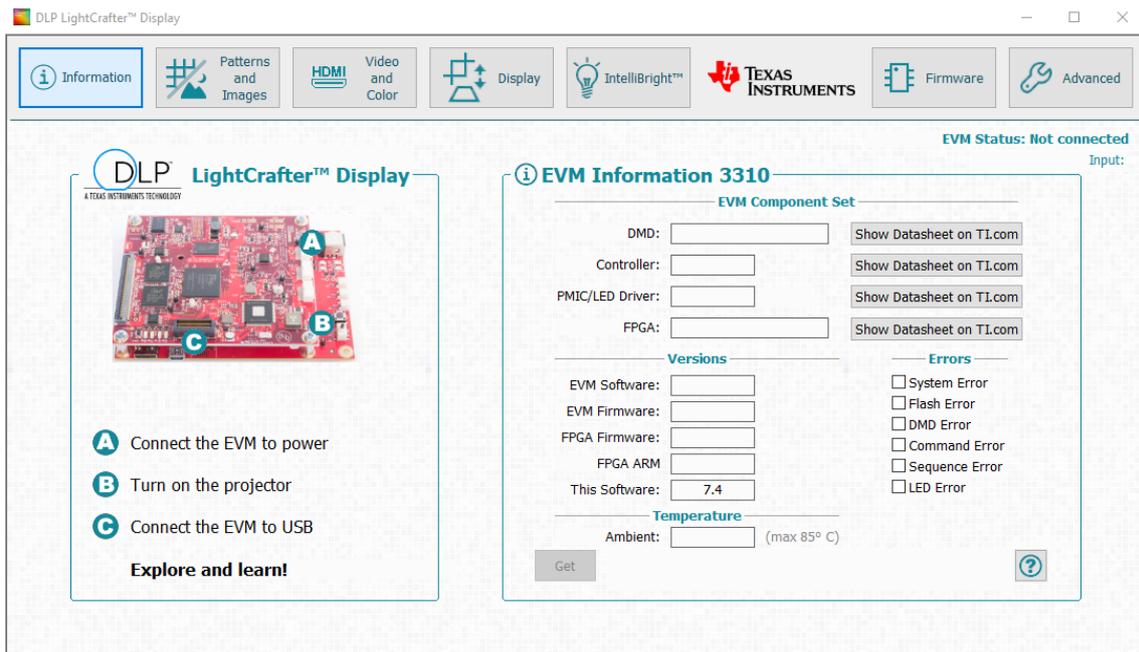


Figura 3.2: Interfaz de usuario del DLP 3310 que se usó para controlar algunas funciones del módulo durante el experimento.

Parámetro	Valor	Unidad
Número de columnas activas	1,368	microespejos
Número de filas activas	768	microespejos
Ancho del microespejo	5.4	μm
Ancho del arreglo de microespejos activo	7.387	mm
Largo del arreglo de microespejos activo	4.147	mm
Ángulo de inclinación de los microespejos	17	grado
Índice de refracción	1.5119	

Tabla 3.1: Principales características del DMD modelo DLP3310.

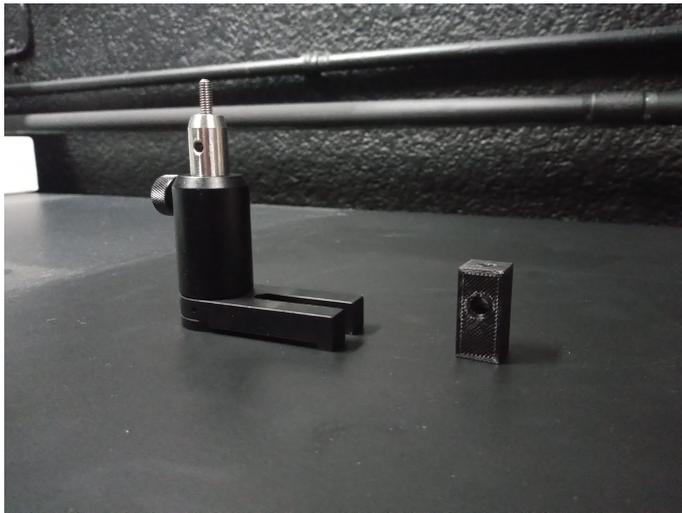


(a) Componentes del DMD por separado



(b) Componentes en la montura

Figura 3.3: El disipador, el DMD, el FFC con su tarjeta en un extremo y la tarjeta secundaria. Elementos por separado 3.3a. Elementos unidos mediante la montura visto lateralmente 3.3b.



(a) Atornillar el poste de cada elemento óptico a su base



(b) Atornillar cada elemento óptico a su poste

Figura 3.4: Proceso de montaje común a todos los elementos ópticos ejemplificado con un bloque de plástico. Se muestra el poste atornillado a la base en 3.4a y se muestra el elemento atornillado al poste en 3.4b.

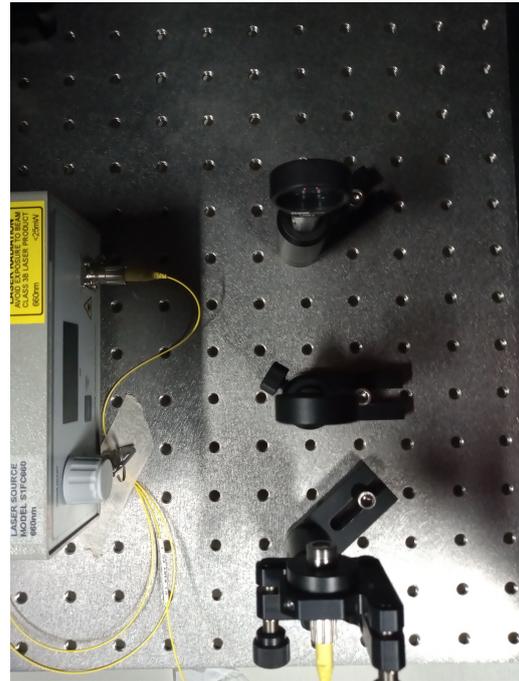
en la figura 3.5a, el cual emite luz con una longitud de onda de 660 nm (por lo que es de un color rojo) . La apertura del láser fue conectada a uno de los extremos de un cable de conexión de fibra óptica multimodo marca THORLABS y modelo P1-630Y-FC-1 como se ilustra en la figura 3.5b. La fibra óptica usada estaba optimizada para operar con longitudes de onda de entre 633 nm y 780 nm. El otro extremo de la fibra óptica fue conectado a un soporte cinemático que sostenía en su centro a un colimador marca THORLABS modelo *F260FC-B* con un foco de 15.5 mm y que estaba optimizado para una longitud de onda de 633 nm que está por debajo de la longitud de onda del láser, pero esta diferencia entre longitudes de onda no afectó significativamente a los resultados obtenidos en el experimento.

Respecto al montaje del colimador, para este elemento se realizó antes un paso extra: ambos tornillos del soporte de la montura cinemática se ajustaron de tal manera que no estuviera atornillado uno más que el otro. Después, como el resto de los elementos ópticos, la montura se colocó en su poste y el poste en su base. Una vez fijo el colimador, se le conectó al final del otro extremo de la fibra óptica y se encendió el láser. Se encendía el láser girando la *keylock*, presionando el botón *Enable* y girando la perilla de ajuste de potencia en el sentido contrario a las manecillas del reloj, todo en ese orden.

Antes de describir la alineación de este primer elemento óptico hay que mencionar que la alineación de los siguientes elementos es similar en el sentido de que primero se tenía que hacer una alineación de cerca y luego una alineación de lejos. La alineación de cerca se hacía tanto verticalmente como horizontalmente: (i) verticalmente, se subía el poste hasta que el haz estuviera a una altura de 90 mm como se muestra en la figura 3.6 auxiliándose de una regla magnética que se adhería a la mesa óptica y una tarjeta blanca graduada que se pegaba a la regla para ver mejor el láser aún a potencias bajas (ii) horizontalmente, se cuidó que el colimador estuviera sobre uno de los orificios de la mesa óptica para después ir colocando el resto de los elementos sobre orificios en la misma línea. La alineación de lejos, es decir, alejando

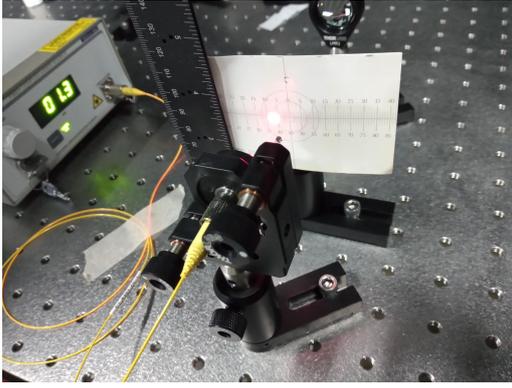


(a) Fuente láser modelo *S1FC660* con una longitud de onda de 660 nm. Se muestra como uno de los extremos de la fibra óptica está conectado al láser.

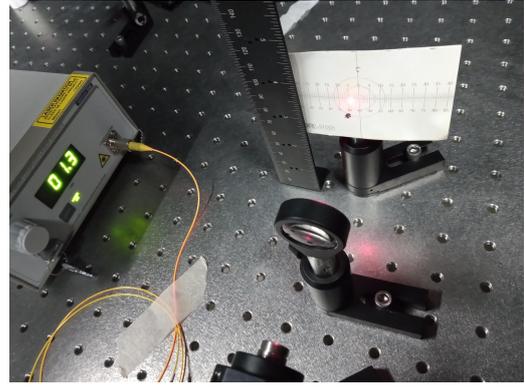


(b) Vista superior del expansor de haz: la fibra óptica conecta el láser con el colimador y frente al colimador están las lentes que expanden el haz.

Figura 3.5: Fuente láser modelo *S1FC660* con una longitud de onda de 660 nm conectada a un colimador optimizado para una longitud de onda de 633 nm. Frente al colimador están dos lentes que sirven para expandir el haz a un tamaño que cubre de mejor forma la superficie del DMD.



(a) Alineación del colimador.



(b) Alineación del lente plano-convexo.

Figura 3.6: Alineaciones de cerca del colimador y el lente plano-convexo. Se muestra como a la regla se pegó una tarjeta graduada para poder ver mejor el láser en el fondo blanco.

la regla hasta el otro extremo de la mesa óptica se hizo también horizontalmente y verticalmente, pero para el caso especial del colimador se podían usar los dos tornillos de la montura cinética: el tornillo superior era para ajustar verticalmente y el tornillo inferior para ajustar horizontalmente.

Luego de que el colimador estuviera alineado, se debían colocar dos lentes para expandir el haz de tal forma que este pudiera cubrir de mejor manera la superficie del DMD y para que la intensidad se disminuyera un poco, evitando en los siguientes pasos la saturación de las imágenes en medida de lo posible. El primer lente era plano-convexo que es etiquetado como L_1 en la figura 3.7 del montaje experimental, es marca THORLABS y modelo $LA1805-A$ con una distancia focal de $f_1 = 30.0$ mm. El lente plano-convexo L_1 se colocó frente al colimador más allá de su distancia focal de 15.5 mm y se procedió a alinearlo. Se alineó de cerca en dos pasos: (i) la alineación horizontal se hacía moviendo la base mientras el lente se mantenía paralelo al elemento óptico anterior que era el colimador y (ii) la alineación vertical se hacía ajustando la altura del poste. Se alineó de lejos también en dos pasos: (i) horizontalmente, se hacía rotar ligeramente el poste y (ii) verticalmente, si estaba desalineado se tenía que volver a ajustar de cerca. Además, cabe mencionar que se contemplaron un par de aspectos extra: (i) se cuidó en todo momento que el láser pasara por el centro del lente y (ii) el reflejo que el lente tenía hacia atrás (hacia el elemento óptico anterior) debía coincidir con el haz del láser.

El segundo lente era biconvexo y es etiquetado como L_2 en la figura 3.7 del montaje experimental, es marca THORLABS y modelo $LB1779-A$ con una distancia focal de $f_2 = 60.0$ mm, por lo cual la distancia entre los lentes era de $f_1 + f_2 = 90.0$ mm. El expensor de haz que formaban los dos lentes es kepleriano, su radio de expansión m es igual a la razón entre la distancia focal de los lentes de la siguiente forma

$$m = \frac{f_2}{f_1} = \frac{60.0 \text{ mm}}{30.0 \text{ mm}} = 2, \quad (3.2)$$

donde f_1 es la distancia focal del primer lente (el plano-convexo) y f_2 es la distancia focal del segundo lente (el biconvexo). Lo anterior dió como resultado que en el experimento el haz pasara de 6 mm a 12 mm luego de atravesar ambos lentes.

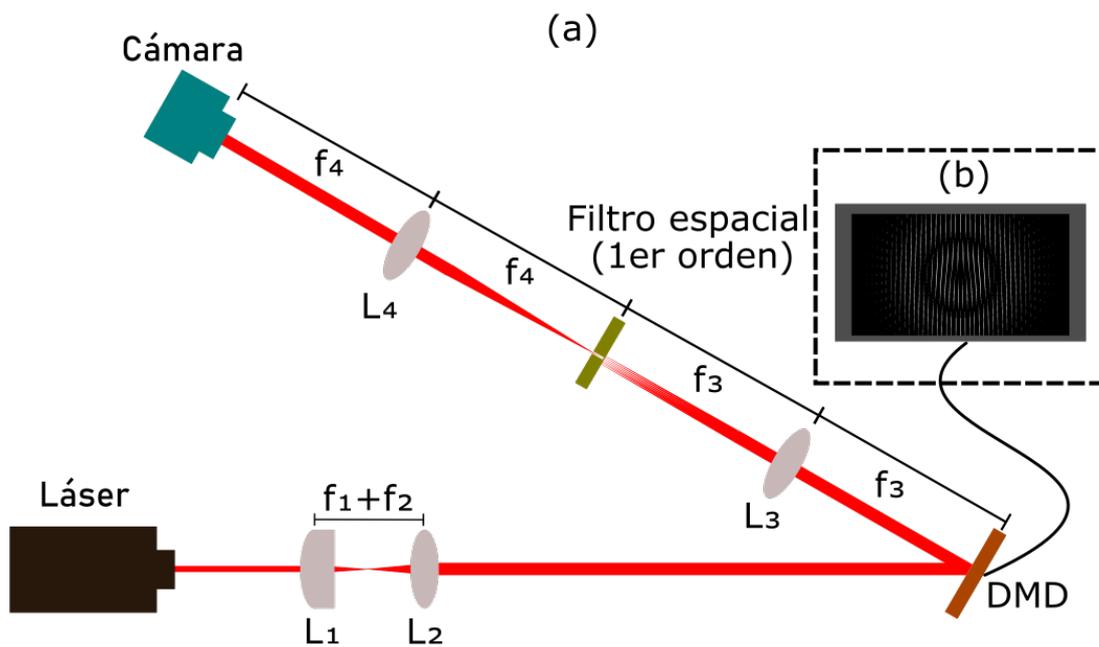


Figura 3.7: (a) Vista esquemática del montaje experimental, donde L_1 y L_2 son las lentes que expanden el haz y sus distancias focales son f_1 y f_2 , respectivamente; mientras que L_3 y L_4 son las lentes transformadoras de Fourier y sus distancias focales son f_3 y f_4 , respectivamente. (b) Imagen del holograma desplegado en el DMD.

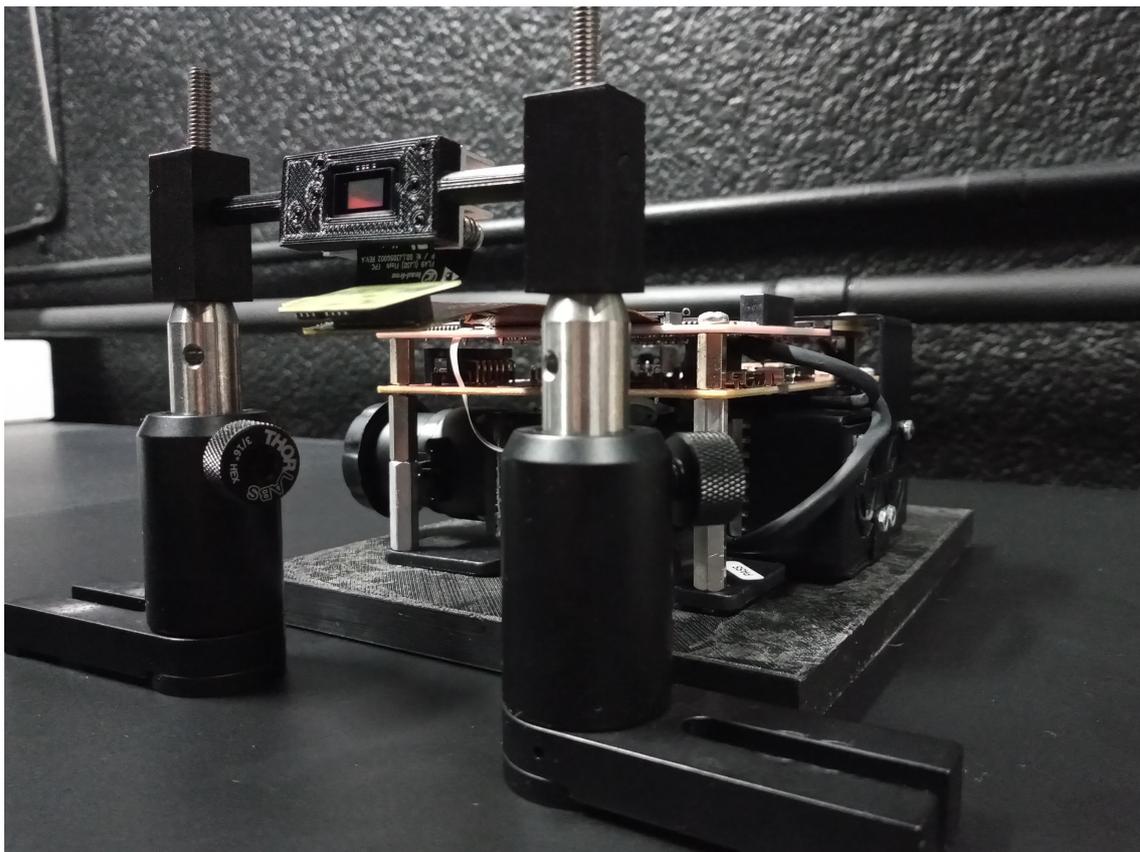


Figura 3.8: Imagen del DMD montado en las bases construidas mediante impresión 3D

El siguiente elemento, el módulo DLP 3310 debía ser fijado a la mesa óptica en el extremo opuesto a donde se habían colocado los dos lentes que expandían el haz. El primer paso para lograr lo anterior fue diseñar en AutoCAD un par de bloques como el que se muestra en la figura 3.4 como se había hecho con la montura para el DMD. Dichos bloques complementarían con el soporte del DMD como se ilustra en la figura 3.8. En la imagen cada uno de los extremos del soporte del DMD atravesaban a cada uno de los bloques. Además, cada bloque tenía orificios extra para que entraran dos tornillos: (i) uno por la parte inferior para atornillar el bloque al poste y (ii) uno por la parte superior para sujetar la extremidad del soporte del DMD y evitar así su rotación. El segundo paso consistió en diseñar e imprimir una base de plástico sobre la cual se atornillaría el módulo y así fijarlo a la mesa óptica. La base era un prisma rectangular con cuatro orificios en sus esquinas que coincidía con los cuatro tornillos que ya tenía debajo el módulo, en este caso se tuvieron que cambiar los tornillos que ya traía el módulo por otros tornillos más largos que atraparan tanto la pieza de plástico como los orificios del dispositivo. Se habla más de estas últimas monturas auxiliares en el apéndice B.

Antes de comenzar con la alineación se tenía que encender el módulo y desplegar una imagen de tal manera que los espejos rotaran un ángulo de 17° o -17° , para ello, desde la interfaz de usuario mostrada anteriormente se desplegaba una imagen pre-determinada y se aprovechaba para desactivar los leds, este procedimiento se hacía cada vez que se movía el DMD para evitar cualquier posible falla por una manipulación excesiva. Primero se alineó de cerca: (i) para alinear verticalmente se fijaron

los extremos de la montura del DMD a los bloques usando los tornillos superiores y manteniendo el DMD lo más vertical posible mientras se subían los postes a la altura apropiada y (ii) respecto a la alineación horizontal, esta no se tuvo que cuidar pues, como se muestra en la figura 3.7 del montaje experimental, el DMD solo se colocaba en un ángulo tal que se aprovechara la mayoría del espacio disponible. Después se alineó de lejos: (i) para la alineación vertical se aprovechó que el DMD rotaba cada vez que se apretaban los tornillos, entonces dependiendo de que tanto se apretaban o aflojaban se lograba esa pequeña rotación que necesita la alineación de lejos y (ii) se tenía que cuidar que al hacer la rotación el haz siguiera reflejándose en los microespejos y si no llegaba a ser así se tenía que reajustar ligeramente la altura de los postes. Al terminar, se dejó una regla de lejos como referencia para alinear horizontalmente los siguientes elementos ópticos. Cabe mencionar que los microespejos reflejaban varios órdenes de difracción y por ello se eligió alinear el más brillante.

De manera similar a los lentes plano-convexo y biconvexo que formaban el expansor de haz, se alineó otro lente biconvexo denotado como L_3 en la figura del montaje experimental 3.7 de marca THORLABS, modelo *LB1056-A* y distancia focal $f_3 = 250.0$ mm. La única diferencia durante el montaje de este lente fue que la alineación horizontal se hizo con ayuda de la regla fijada anteriormente durante la alineación del módulo de evaluación. La separación entre el lente L_3 y el DMD era igual a su distancia focal con el propósito de obtener la transformada de Fourier del holograma. Luego, en el foco posterior de la lente se colocó el filtro espacial o iris encima de una base de traslación micrométrica con la cual se podía mover con precisión el filtro espacial hasta la posición del primer orden de difracción. Cabe mencionar que la posición del primer orden de difracción depende del período de la rejilla utilizada en el diseño del holograma y se encontró que el valor óptimo para dicho período era de 18 microespejos $\approx 97.2 \mu\text{m}$ porque de ser más pequeño el holograma tenía un muestreo insuficiente y de ser más grande los órdenes se juntaban demasiado, esto último se puede inferir de la ecuación (2.17). El proceso para diseñar los hologramas se describe a detalle en el apéndice A.

Tras la filtración del primer orden de difracción se colocó el último lente denotado como L_4 en la figura del montaje experimental 3.7 de marca THORLABS, modelo *LB1779-A* y distancia focal $f_4 = 300.0$ mm que se alineó de manera similar al lente anterior. En el foco posterior de este lente se colocó la cámara marca THORLABS y modelo *DCU224C* para capturar la transformada de Fourier del primer orden de difracción con lo cual se recuperaba el campo inicial.

Se tuvieron que programar desde MATLAB tres de los elementos del montaje experimental, la cámara, el DMD y una pantalla auxiliar. Respecto al DMD y la pantalla auxiliar, estos dos se usaron como segundas pantallas de la computadora portátil que se conectaban desde su salida HDMI mediante un divisor de HDMI que duplicaba la señal logrando que la imagen que se mostraba en el DMD pudiera verse también en la pantalla auxiliar. La pantalla auxiliar era modelo *VTV23615B* de 1366x768 píxeles por lo cuál coincidía con las dimensiones del DMD evitando así diferencias entre lo que se veía en ambas pantallas sobre las cuales se desplegaban ventanas de figura de MATLAB que contenían a los hologramas. Respecto a la cámara, se usaron los programas del manual [16]. Todo esto se detalla en el apéndice A.

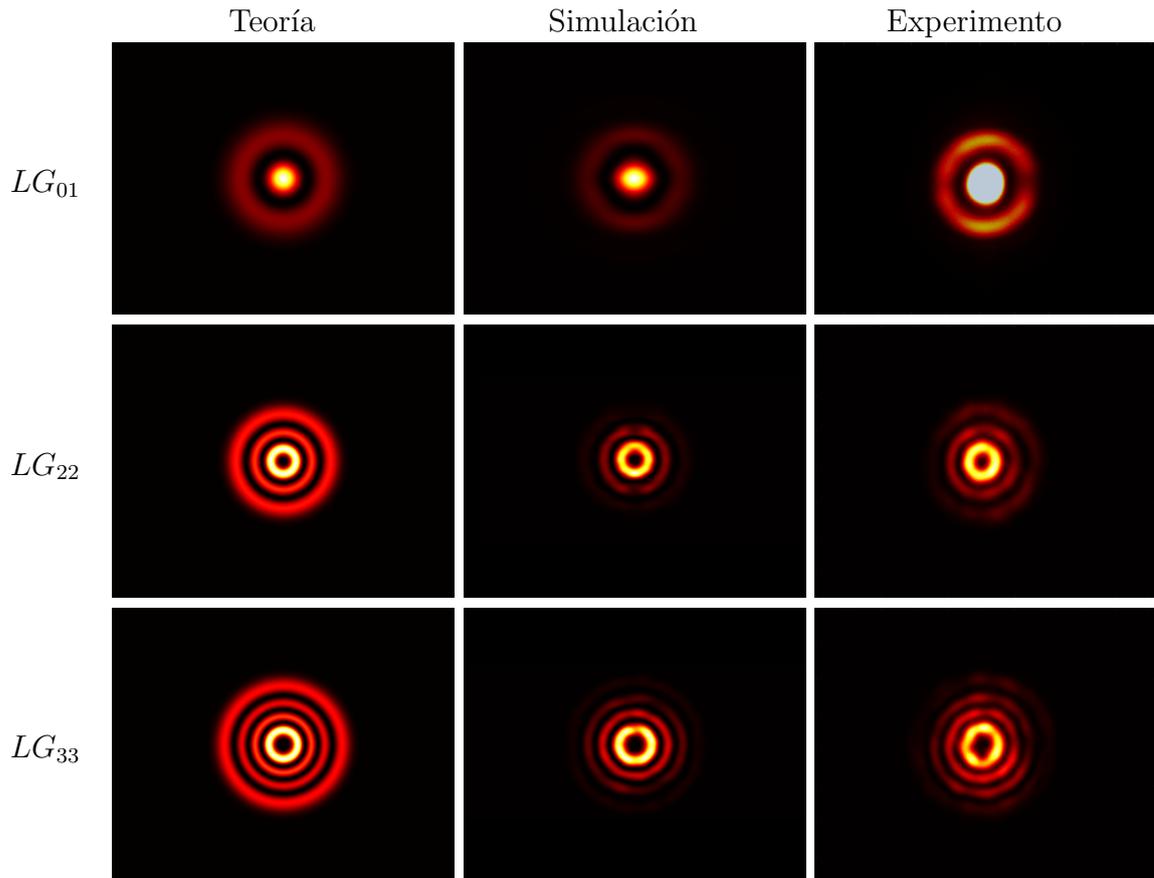


Tabla 3.2: Tabla con algunas de las imágenes capturadas en el experimento comparadas con las generadas en la simulación. La columna izquierda tiene las imágenes de los haces teóricos; la columna del centro contiene a los haces experimentales; y la columna derecha tiene las simulaciones con la distribución de intensidades característica del filtraje experimental.

3.6. Resultados

Durante el experimento se generaron y capturaron haces Laguerre-Gauss con índices azimutales desde $l = 0$ hasta $l = 3$ y con índices radiales desde $p = 1$ hasta $p = 3$ dando un total de 16 de ellos. La tabla 3.2 contiene una muestra de las imágenes de los haces experimentales en comparación con los simulados, solo para $l = 0$ y $p = 1$, para $l = p = 2$ y para $l = p = 3$, es decir, para LG_{01} , LG_{22} y LG_{33} . La simulación partió de una matriz binaria que simulaba a la rejilla de la cual se hizo la transformada de Fourier, se multiplicó por una matriz que representaba al filtro circular y se volvió a hacer su transformada. La primera de las diferencias entre las imágenes experimentales y las imágenes simuladas son las intensidades, en el experimento las intensidades mostraban menor uniformidad entre un anillo y otro, en especial el caso cuando $l = 0$ donde el centro era demasiado brillante. Se llegaba a saturar la cámara a pesar de que la potencia del láser siempre se tuvo en el orden de décimas de miliwatts que era la potencia mínima del láser.

Por otro lado, aunque se pueden observar deformaciones en todas las imágenes, estas deformaciones son aún más notables en las imágenes experimentales que están en la columna central en la tabla 3.2. Después de la primera transformación de

Fourier que hace el lente L_3 , los órdenes de difracción están muy juntos el uno del otro al llegar al iris y, por ello, este último no los filtra de manera eficiente. La condición experimental antes mencionada se logró reproducir en el código ajustando el tamaño del filtraje circular.

Capítulo 4

Identificación de modos espaciales usando inteligencia artificial

La deformación que experimentan los modos espaciales tras ser generados no representa mayor complicación para su reconocimiento. Sin embargo, tras pasar por un medio con dispersión como la fibra óptica, vidrio esmerilado e inclusive la misma atmósfera se pueden deformar hasta el punto de hacerse irreconocibles. La simulación de frentes de onda distorsionados por la atmósfera ha sido una herramienta importante para estudiar la propagación de luz y la imagenología. El punto de partida para la realización del presente trabajo fue asumir que la turbulencia atmosférica sigue el espectro de Kolmogorov y tiene una fase que es estadísticamente uniforme en el intervalo de $-\pi$ a π . Dentro de esta suposición se tiene que las fluctuaciones en el medio causan deformaciones tanto en la fase como en la amplitud del frente de onda. Las distorsiones en la fase tiene efectos más considerables en la calidad de las imágenes en comparación con las distorsiones en la amplitud [17].

La simulación de las deformaciones mediante máscaras de Kolmogorov no solo reproduce el efecto de la atmósfera, sino que recientes trabajos señalan que estas máscaras reproducen los efectos del vidrio esmerilado [18]. Hacer pasar los haces Laguerre-Gauss a través de vidrio esmerilado e incluso fibras ópticas antes de ser capturados por la cámara es el siguiente paso experimental para futuros trabajos, pero como un primer acercamiento, esta simulación parte de una haz Laguerre-Gauss que se deforma al ser multiplicado por una máscara de Kolmogorov con un grado de distorsión transversal controlado a partir del parámetro de Fried $r_0 = 1.4$ m que también es conocido como la longitud de coherencia de Fried y que entre más pequeño es, más distorsión provoca pues este parámetro determina a qué distancia cambia la fase que deforma al frente de onda y, por lo tanto, determina el grado de ruido que se introduce en el campo que interactúa con dicha fase. Una vez deformado el campo, se propagó una cierta distancia y se obtuvo la intensidad al final de la propagación. La diferencia entre un haz distorsionado y uno sin distorsionar se puede ver en la tabla 4.1.

Los modos espaciales, si son reconocidos correctamente aunque estén deformados, pueden hacer el papel de bits para transmitir datos. Tomar el código de caracteres estándar para la transmisión de datos, es decir, el ASCII (siglas del inglés de *American Standard Code for Information Interchange*) sirve como un buen punto de partida. El código ASCII en su versión extendida usa un total de 8 bits o 1 byte para codificar caracteres, por lo cual, un total de ocho haces Laguerre-Gauss con una

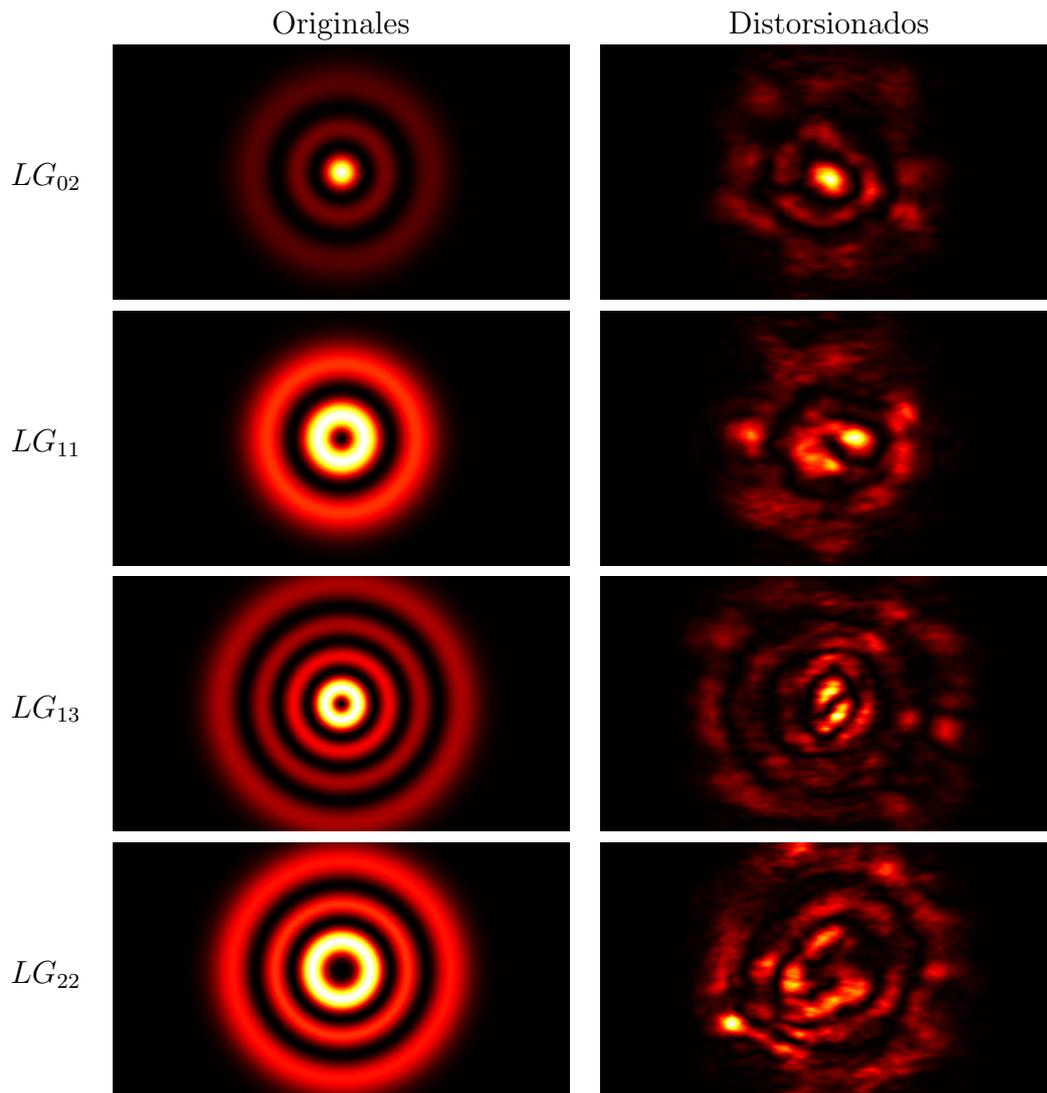


Tabla 4.1: La columna izquierda tiene imágenes de la intensidad de los haces Laguerre-Gauss sin deformaciones y la columna derecha tiene imágenes luego de que estos se deformaran mediante máscaras de Kolmogorov.

Haz Laguerre-Gauss	Enumeración	One-hot encoding
LG_{01}	1	[1 0 0 0 0 0 0]
LG_{02}	2	[0 1 0 0 0 0 0]
LG_{03}	3	[0 0 1 0 0 0 0]
LG_{11}	4	[0 0 0 1 0 0 0]
LG_{12}	5	[0 0 0 0 1 0 0]
LG_{13}	6	[0 0 0 0 0 1 0]
LG_{21}	7	[0 0 0 0 0 0 1]
LG_{22}	8	[0 0 0 0 0 0 0 1]

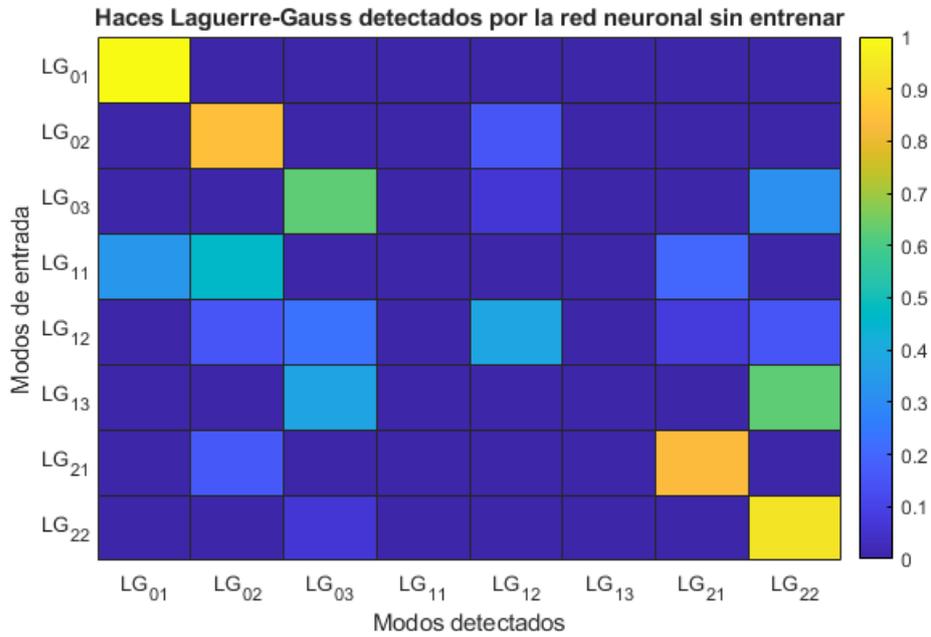
Tabla 4.2: La primer columna de esta tabla contiene el haz Laguerre-Gauss, la segunda columna contiene su enumeración y la tercera contiene su respectivo one-hot encoding.

cintura de $w_0 = 0.0008$ m se enumeraron y a cada número se le asignó un arreglo de bits como se muestra en la tabla 4.2 mediante una técnica conocida en inglés como *one-hot encoding*. Cada uno de los arreglos representa una clase y de cada clase se generaron un total de cien muestras para la simulación.

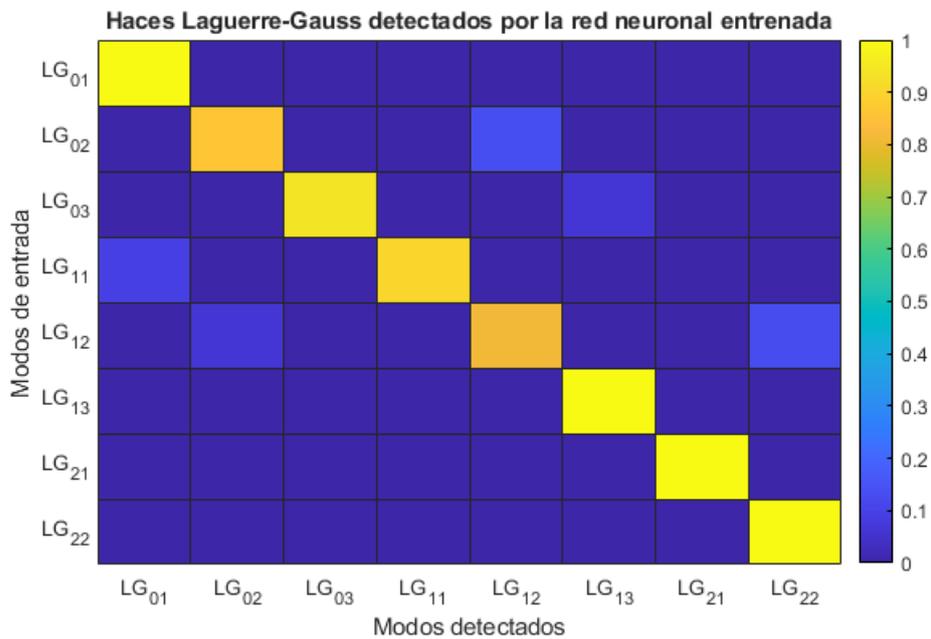
El resultado del one-hot encoding fue un total de 800 arreglos unidimensionales de 8×1 que se concatenaron en un solo arreglo bidimensional de 8×800 para servir como las etiquetas o salidas de la red neuronal. Mientras que las imágenes de 768×1366 se guardaron como elementos de una matriz tridimensional de $800 \times 768 \times 1366$. Luego, cada una de las imágenes fue reducida tomando promedios de ventanas de 37×37 dejando como resultado imágenes de 20×36 que requieren de un menor trabajo de procesamiento. Las reducciones fueron acomodadas en arreglos unidimensionales para servir como entradas de la red neuronal.

Para el reconocimiento inteligente de modos se diseñó una red neuronal con la ayuda del *Deep Learning Toolbox* de MATLAB usando la función *patternnet* donde se especificó que la red tendría un total de tres capas ocultas cada una con 20, 50 y 30 neuronas y una capa de salida con ocho neuronas. Tal configuración se consiguió mediante un proceso de prueba y error buscando optimizar la eficiencia de la red. Se entrenó esta red por 94 épocas, por defecto su función de activación era *Softmax* y su función de costo era *Cross Entropy Loss*. Por otro lado, los datos se dividieron en 70 % para el conjunto de entrenamiento, 15 % para el conjunto de validación y 15 % para el conjunto de prueba. La figura 4.1 muestra matrices de confusión en forma de mapas de calor de las predicciones de la red neuronal sobre el conjunto de prueba comparandolas con las verdaderas salidas. La matriz de confusión superior corresponde a la red antes de ser entrenada y la matriz inferior corresponde a la red neuronal después de ser entrenada. Se puede notar que luego del entrenamiento la red puede reconocer en un alto porcentaje todas las clases, es decir, los ocho tipos de haces Laguerre-Gauss tras ser deformados llegando incluso a eficiencias del 99.2 %.

Una vez asegurado el reconocimiento de haces Laguerre-Gauss individuales, se puede plantear el envío de mensajes un carácter a la vez en su código ASCII. Tomando como ejemplo el carácter «A», se debe de enviar el byte [0 1 0 0 0 0 1] que según la tabla 4.2 necesita a los haces Laguerre-Gauss dos LG_{02} y ocho LG_{22} . Ambos haces pueden ser enviados en dos intervalos de tiempo y en un orden aleatorio por el emisor. El receptor quien tendrá la red neuronal ya entrenada puede reconocerlos inmediatamente, hacer su suma y saber de que carácter se trata. La implementación



(a) Predicciones de la red neuronal antes del entrenamiento



(b) Predicciones de la red neuronal después del entrenamiento

Figura 4.1: Matrices de confusión de las predicciones de la red neuronal contra las salidas reales. La figura 4.1a muestra los resultados antes de que la red fuera entrenada y la figura 4.1b muestra los resultados luego de que esta fue entrenada.

de las alternativas a enviar el carácter como una secuencia de modos espaciales se plantea como un trabajo a futuro.

La continuación natural de este trabajo es probar experimentalmente la eficacia que tiene la red al reconocer los haces luego de que estos atraviesen un vidrio esmerilado. Luego, se pasaría al caso específico de las fibras ópticas multimodo que es sumamente útil para establecer canales de información a larga distancia y a un bajo costo. Se planea además aplicar deformaciones dinámicas en las fibras ópticas mientras el mensaje se encuentra viajando, las deformaciones deben incluir torsiones y cambios de temperatura los cuales son comunes en una instalación ordinaria de estos medios en la vida real.

Se debe modificar además la forma del mensaje. La primera modificación consiste en cambiar una secuencia de haces Laguerre-Gauss y enviar en su lugar a todos los haces al mismo tiempo, es decir, la superposición coherente de todos los haces que codifican al carácter, lo cual aumentaría la capacidad del protocolo para transmitir mensajes. Para reconocer los haces Laguerre-Gauss que conforman a la superposición se requeriría que la red neuronal fuera capaz de reconocer no solo los ocho haces, sino todas sus combinaciones posibles. La segunda modificación consiste en aprovechar el hecho de que el DMD es capaz de trabajar a frecuencias del orden de MHz, lo cual permitiría enviar secuencias de imágenes en el rango de video; pero esto último se encuentra actualmente bajo investigación en el grupo del laboratorio de micro y nanofotónica del ICN.

Capítulo 5

Conclusiones

La presente tesis comprueba la generación de campos escalares arbitrarios mediante la proyección de hologramas binarios en un DMD, hologramas con los que no solo se puede controlar la amplitud del haz, sino también su fase. Al respecto, también se debe de considerar la calidad de los haces obtenidos pues escoger un número insuficiente de microespejos para el período de la rejilla lleva a errores en el modo generado, pero al mismo tiempo un número de microespejos por período muy grande aumenta el ruido de órdenes mayores. En consecuencia, la obtención de modos de una mejor calidad requiere de un DMD con más microespejos y más pequeños que hagan un muestreo adecuado del campo que codifican. Esta mejora en la calidad junto con la velocidad que ya tiene un DMD optimizarían la etapa de codificación de los mensajes en un protocolo de comunicación. Luego de la codificación, la luz viajaría a través de fibra óptica que es un medio muy rápido, pero que al tener tanta dispersión terminaría deformando los haces que conforman el mensaje hasta el punto de dejarlos irreconocibles.

Ante la dificultad de reconocer los haces deformados, se demostró que los algoritmos de inteligencia artificial desarrollados son capaces de reconocer los modos espaciales incluso luego de que estos pasan a través de un medio con desorden simulado mediante máscaras de Kolmogorov. La red neuronal obtenida puede ser usada en un protocolo de comunicación en la que el receptor envía caracteres codificados como una secuencia de ocho haces Laguerre-Gauss y el receptor con la red neuronal ya entrenada traduce la secuencia en caracteres. Se trata de un protocolo de comunicación segura pues la persona que no conozca los pesos sinápticos de la red neuronal será incapaz de reconocer los modos espaciales.

Referencias

- [1] *DLP LightCrafter Display 3310 EVM User's Guide*, Texas Instruments, 12500 TI Blvd., Dallas, Texas 75243 USA, January 2018. [Online]. Available: <https://www.ti.com/lit/pdf/dlpu063>
- [2] A. M. Yao and M. J. Padgett, "Orbital angular momentum: origins, behavior and applications," *Advances in Optics and Photonics*, vol. 3, pp. 161–204, 2011.
- [3] T. Ando, Y. Ohtake, N. Matsumoto, T. Inoue, and N. Fukuchi, "Mode purities of laguerre–gaussian beams generated via complex-amplitudemodulation using phase-only spatial light modulators," *Opt. Lett.*, vol. 34, no. 1, pp. 34–36, Jan 2009. [Online]. Available: <https://opg.optica.org/ol/abstract.cfm?URI=ol-34-1-34>
- [4] C. Zhang, C. Min, and X.-C. Yuan, "Shaping perfect optical vortex with amplitude modulated using a digital micro-mirror device," *Optics Communications*, vol. 381, pp. 292–295, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S003040181630596X>
- [5] M. Krenn, J. Handsteiner, M. Fink, R. Fickler, and A. Zeilinger, "Twisted photon entanglement through turbulent air across vienna," *Proceedings of the National Academy of Sciences*, vol. 112, pp. 14 197 – 14 201, 2015.
- [6] W.-H. Lee, "Binary computer-generated holograms," *Appl. Opt.*, vol. 18, no. 21, pp. 3661–3669, Nov 1979. [Online]. Available: <http://opg.optica.org/ao/abstract.cfm?URI=ao-18-21-3661>
- [7] F. Hufnagel, A. Sit, F. Bouchard, Y. Zhang, D. England, K. Heshami, B. J. Sussman, and E. Karimi, "Investigation of underwater quantum channels in a 30 meter flume tank using structured photons," *New Journal of Physics*, vol. 22, no. 9, p. 093074, sep 2020. [Online]. Available: <https://dx.doi.org/10.1088/1367-2630/abb688>
- [8] Y. Shen, X. Wang, Z. Xie, C. Min, X. Fu, Q. Liu, M. Gong, and X. Yuan, "Optical vortices 30 years on: Oam manipulation from topological charge to multiple singularities," *Light, Science & Applications*, vol. 8, 2019.
- [9] N. Bozinovic, Y. Yue, Y. Ren, M. Tur, P. Kristensen, H. Huang, A. E. Willner, and S. Ramachandran, "Terabit-scale orbital angular momentum mode division multiplexing in fibers," *Science*, vol. 340, pp. 1545 – 1548, 2013.
- [10] J. Goodman, *Introduction to Fourier Optics*. W. H. Freeman, 2017. [Online]. Available: <https://books.google.com.mx/books?id=9zY8DwAAQBAJ>

-
- [11] M. Mirhosseini, O. S. Magaña-Loaiza, C. Chen, B. Rodenburg, M. Malik, and R. W. Boyd, “Rapid generation of light beams carrying orbital angular momentum,” *Optics Express*, vol. 21, no. 25, p. 30196, dec 2013. [Online]. Available: <https://doi.org/10.1364%2Foe.21.030196>
- [12] B. Saleh and M. Teich, *Fundamentals of Photonics*, ser. Wiley Series in Pure and Applied Optics. Wiley, 2007. [Online]. Available: <https://books.google.com.mx/books?id=Ve8eAQAAIAAJ>
- [13] S. Sharifi, Y. Banadaki, E. Siddiqui, S. Cuzzo, N. Bhusal, L. Cohen, A. Kalasky, N. Prajapati, R. Soto-Garcia, S. Brown, I. Novikova, E. Mikhailov, G. Veronis, and J. Dowling, “Identifying laguerre-gaussian modes using convolutional neural network,” in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 475–478.
- [14] D. Dudley, W. M. Duncan, and J. Slaughter, “Emerging digital micromirror device (DMD) applications,” in *MOEMS Display and Imaging Systems*, H. Urey, Ed., vol. 4985, International Society for Optics and Photonics. SPIE, 2003, pp. 14 – 25. [Online]. Available: <https://doi.org/10.1117/12.480761>
- [15] *DLP3310 0.33 1080p DMD datasheet*, Texas Instruments, 12500 TI Blvd., Dallas, Texas 75243 USA, November 2018. [Online]. Available: <https://www.ti.com/lit/gpn/dlp3310>
- [16] *DCC and DCU Cameras: MATLAB Interface Guide*, Thorlabs GmbH, Hans-Böckler-Str.6 85221 Dachau/Munich, March 2013. [Online]. Available: <https://physlab.org/wp-content/uploads/2016/04/DCU223C-Manual.pdf>
- [17] R. G. Lane, A. Glindemann, and J. C. Dainty, “Simulation of a kolmogorov phase screen,” *Waves in Random Media*, vol. 2, no. 3, pp. 209–224, 1992. [Online]. Available: <https://doi.org/10.1088/0959-7174/2/3/003>
- [18] M. Hong, A. Miller, R. d. J. León-Montiel, C. You, and O. S. Magaña-Loaiza, “Engineering photon statistics of spatial light modes,” 2022. [Online]. Available: <https://arxiv.org/abs/2212.01734>
- [19] *DLP LightCrafter Display EVM GUI Tool*, Texas Instruments, 12500 TI Blvd., Dallas, Texas 75243 USA, June 2014. [Online]. Available: <https://www.ti.com/lit/ug/dlpu021b/dlpu021b>

Apéndice A

Códigos en MATLAB

A.1. Generación de haces Laguerre-Gauss

La programación de una función que genere haces Laguerre-Gauss necesita de todas las variables descritas en la ecuación (3.1): el índice radial p que determina el número de anillos y que es un número entero positivo, el índice azumutal l que determina el momento angular orbital y que es un número entero, el radio de la cintura en el foco w_0 que es un flotante y las dos coordenadas cilíndricas R y Phi que son arreglos bidimensionales de flotantes. Respecto a las coordenadas cilíndricas, éstas se pueden calcular a partir de las coordenadas rectangulares X y Y que también son arreglos bidimensionales, pero que resultan más convenientes por la construcción del código como se verá más adelante en éste apéndice: el cálculo de la coordenada radial es $R = \sqrt{X^2 + Y^2}$ y el cálculo de la coordenada azimutal es $Phi = \arctan(X/Y)$ donde se usa la función `atan2` en lugar de `atan` para que el ángulo resultante esté en los cuatro cuadrantes del plano.

Bloque de código A.1: Versión simplificada de la función de MATLAB que genera un haz Laguerre-Gauss

```
function LG = lg(X, Y, l, p, w0)
    R = sqrt(X.^2 + Y.^2);
    Phi = atan2(Y,X);
    Term_0 = sqrt(2 * factorial(p) / (pi * factorial(p +
        abs(l))));
    Term_1 = (sqrt(2) * R / w0).^ abs(l);
    Term_2 = laguerreL(p, abs(l), 2 * R.^2 / w0^2);
    Term_3 = exp(-R.^2 / w0^2);
    Term_4 = exp(-1i * l * Phi);
    LG = Term_0 .* Term_1 .* Term_2 .* Term_3 .* Term_4;
end
```

El bloque de código A.1 calcula cinco factores que se describen a continuación en el orden en que aparecen: (i) la constante de normalización $A_{l,p} = \sqrt{2p! / [\pi(p + |l|)!]}$, (ii) el factor $(\sqrt{2}r/W_0)^{|l|}$, (iii) el polinomio de Laguerre $L_p^{|l|}(2r^2/W_0^2)$ que se calcula usando la función `laguerreL` del *Symbolic Math Toolbox* de MATLAB, (iv) el factor $\exp(-r^2/W_0^2)$ y (v) la fase $\exp(jl\theta)$. El producto de los cinco terminos es el haz Laguerre-Gauss LG en forma de un arreglo bidimensional de las mismas dimensiones que las coordenadas.

A.2. Generación de hologramas

Sobre los haces Laguerre-Gauss generados con la función anterior se aplicó la función de transmitancia binaria descrita en la ecuación (2.21) para poder generar los hologramas binarios. La función de transmitancia binaria utiliza dos funciones auxiliares $w(x, y)$ y $p(x, y)$ que se programaron como se muestra en los bloques de código A.3 y A.2. La primera función recibe la fase de la onda objetivo U_phase para dividirla entre π y la segunda función recibe la amplitud de la onda objetivo $U_amplitude$ para calcular su arcoseno y dividirlo entre π .

Bloque de código A.2: Función que guarda la fase de la onda objetivo en MATLAB

```
function P = p(U_phase)
    P = U_phase / pi;
end
```

Bloque de código A.3: Función que guarda la amplitud de la onda objetivo en MATLAB

```
function W = w(U_amplitude)
    W = asin(U_amplitude) / pi;
end
```

La función de transmitancia se muestra en el bloque de código A.4 y ocupa tres parámetros: (i) un arreglo bidimensional X que representa al eje sobre el que se hará la codificación, (ii) un arreglo bidimensional U que representa a la onda objetivo y un flotante $x0$ que representa al período de la onda plana con la que se codifica a U . En el cuerpo de la función se extraen dos variables de la onda objetivo, por un lado, su fase se calcula con la función *angle* y, por otro lado, su amplitud se calcula con la función *abs*. La fase y la amplitud se pasan como argumentos de las funciones auxiliares $w(x, y)$ y $p(x, y)$. El arreglo T que regresa la función de transmitancia binaria está conformada por unos y ceros.

Bloque de código A.4: Función de transmitancia binaria en MATLAB

```
function T = t(X, U, x0)
    U_phase = angle(U);
    U_amplitude = abs(U);
    T = 2 \ 1 + 2 \ sign( cos(2 * pi * X / x0 + pi * p(
        U_phase)) - cos(pi * w(U_amplitude)) );
end
```

A.3. Simulación: reconstrucción de haces Laguerre-Gauss

Los campos Laguerre-Gauss que se usaron para generar los hologramas pueden ser reconstruidos experimentalmente y, para ello, un haz debe iluminar cada holograma desplegado en el DMD y pasar por dos lentes y por un filtro espacial. La simulación del experimento comienza entonces con la declaración de los parámetros del arreglo de microespejos del DMD: el número horizontal de píxeles Nx , el ancho del arreglo Wx , el ancho de cada microespejo Dx , el número vertical de píxeles Ny ,

el largo del arreglo Wy y el largo de cada microespejo Dy . A partir de los parámetros anteriores se pudieron calcular los arreglos unidimensionales x y y que tienen su origen en el centro del DMD y que sirven a su vez para crear con la función *meshgrid* dos arreglos bidimensionales X y Y que representan a las coordenadas rectangulares. Dado el ancho de cada microespejo Dx , se estableció que el período del holograma x_0 fuera de 18 microespejos $\approx 97.2 \mu\text{m}$ como se menciona en la sección 3.5.

Bloque de código A.5: Variables iniciales para la simulación: parámetros del DMD, coordenadas rectangulares y período del holograma

```
Nx = 1366;
Ny = 768;
Wx = 7.387e-3;
Wy = 4.147e-3;
Dx = Wx / Nx;
Dy = Wy / Ny;

x = (-Wx / 2) : Dx : (Wx / 2 - Dx);
y = (-Wy / 2) : Dy : (Wy / 2 - Dy);

[X,Y] = meshgrid(x,y);

mirror_pitch = Dx;
mirrors_per_period = 18;
x0 = mirrors_per_period * mirror_pitch;
```

El siguiente paso es generar un haz Gaussiano que representa el haz proveniente de un láser y se muestra en el bloque de código A.6 donde la cintura del haz en el foco es $\sigma = 1.2 \text{ mm}$ y el filtro circular *circ* representa a la salida del láser con un radio de $w = 1.5 \text{ mm}$.

Bloque de código A.6: Haz Gaussiano que ilumina al holograma

```
sigma = 1.2e-3;
w = 1.5e-3;
Gaussian_Beam = exp(-(X.^2 + Y.^2)/(2*sigma^2)) .* circ(
    sqrt( X.^2 + (Y).^2 ) / w );
```

La generación e iluminación del holograma se ejemplifica en el bloque de código A.7. El ejemplo parte de un haz Laguerre-Gauss *LG* que tiene $l = 1$, $p = 3$ y $w_0 = 0.1 \text{ mm}$ al que se le aplica la función de transmitancia binaria para obtener el holograma *LG_T*. Se obtiene el holograma iluminado *LG_T_Illuminated* al multiplicar *LG_T* por el haz *Gaussian_Beam*.

Bloque de código A.7: Generación del holograma binario e iluminación del mismo con un haz Gaussiano

```
L = 1;
P = 3;
W0 = 1e-4;

LG = lg(X,Y,L,P,W0);
```

```
LG_T = t(X, LG, x0);
LG_T_Illuminated = LG_T .* Gaussian_Beam;
```

Resta hacer pasar el reflejo del holograma por un lente, un filtro espacial y otro lente. El paso del haz a través de un lente se realiza en la primera línea del bloque de código A.8 al hacer su transformada de Fourier con la función *fft2* que es la transformada rápida de Fourier para arreglos bidimensionales de MATLAB seguida de la función *fftshift* que desplaza el componente cero de la frecuencia al centro del arreglo. Tras pasar por el primer lente, se tienen varios órdenes de la expansión de Fourier como se muestra en la ecuación (2.17). El filtro espacial para el primero de los órdenes *Filter_1st_Order_LG* se realiza en la segunda línea de código con la función *circ* donde se desplaza 0.41 mm el filtro circular a lo largo de la coordenada *X* y su radio se establece como 0.095 mm. El radio es lo suficientemente grande para dejar pasar un poco del ruido proveniente de otros órdenes y así simular mejor las condiciones experimentales en las que el filtro espacial no era lo suficientemente pequeño. La siguiente línea obtiene el primer orden ya filtrado *F_1st_Order_LG* y la última línea simula el paso del primer orden a través del segundo lente con lo cual se recupera el haz Laguerre-Gauss original *LG_Recovered*. Los ejemplos de haces Laguerre-Gauss reconstruidos se encuentran en el apéndice C.

Bloque de código A.8: Simulación del haz reflejado en el holograma pasando por los lentes y el filtro espacial

```
F_LG_T = fftshift( fft2(LG_T_Illuminated) );

Filter_1st_Order_LG = circ( sqrt( (X - 0.410e-3).^2 + Y
    .^2 ) / 0.095e-3 );

F_1st_Order_LG = F_LG_T .* Filter_1st_Order_LG;

LG_Recovered = fft2( fftshift( F_1st_Order_LG ) );
```

A.4. Experimento: despliegue de hologramas y captura de imágenes

Previo a comenzar con el experimento se descargó e instaló el software *ThorCam Software for Scientific and USB Cameras* que se encuentra en el siguiente [enlace](#) para controlar la cámara desde MATLAB. Partiendo del programa mostrado en el manual para programar la cámara, se realizó su configuración y la captura de imágenes.

A.4.1. Configuración de la cámara

El bloque de código para la configuración de la cámara es el A.9, en él se muestra en las primeras tres líneas como se cambió la ruta de la librería *uc480DotNet.dll* para luego crear un objeto que representara a la cámara y, finalmente, se inicializa la primer y única cámara disponible. Las últimas tres líneas del bloque establecen

que la cámara mostrara las imágenes como un mapa de bits, que el modo de color será RGB en ocho bits y que se adquirirá una sola imagen por captura.

Bloque de código A.9: Configuración inicial de la cámara

```
NET.addAssembly('C:\Program Files\Thorlabs\Scientific Imaging\DCx-Camera-Support\Develop\DotNet\uc480DotNet.dll');
import uc480DotNet.*;

cam = uc480.Camera;
cam.Init(0);
cam.Display.Mode.Set(uc480.Defines.DisplayMode.Mono);
cam.PixelFormat.Set(uc480.Defines.ColorMode.RGBA8Packed);
cam.Trigger.Set(uc480.Defines.TriggerMode.Software);
```

A.4.2. Configuración de la ventana de MATLAB en el DMD y la pantalla auxiliar

Antes de comenzar a capturar las imágenes con la cámara configurada, se debían desplegar los hologramas en el DMD. El DMD y la pantalla auxiliar estaban conectadas como segundas pantallas de la computadora portátil mediante un divisor de HDMI de tal forma que lo que se veía en DMD se veía también en la pantalla auxiliar como se detalla en la sección 3.5. Existen cuatro formas distintas en que se puede hacer la proyección en las segundas pantallas, entonces en la computadora portátil se presionó la tecla con el logotipo de Windows seguida de la tecla p y dentro de la ventana emergente se eligió el modo *Ampliar*. Una vez configurada la proyección en las pantallas, se obtuvieron las dimensiones de la computadora portátil *laptop_dims* en píxeles para de ahí obtener el ancho de la pantalla *laptop_width* que es justamente donde termina la primera pantalla y comienza la segunda. Se creó una figura de MATLAB justo donde termina la pantalla de la computadora portátil y empieza la segunda pantalla de ancho Nx y largo Ny por lo cual esta figura abarcaba todo el DMD y la pantalla auxiliar. Además, a la figura se le quitaron todos los elementos gráficos excepto por el área del gráfico sobre la cual se proyectó el holograma *LG_T* como se muestra en las últimas dos líneas del bloque de código [16, Capítulo 6].

Bloque de código A.10: Configuración de la ventana de la figura de MATLAB en la segunda pantalla de la computadora portátil, es decir, el DMD y la pantalla auxiliar

```
laptop_dims = get(0, 'ScreenSize');
laptop_width = laptop_dims(3);

figure_dmd = figure('Position', [laptop_width 0 Nx Ny], 'MenuBar', 'none', 'ToolBar', 'none', 'WindowState', 'fullscreen');
axis off
set(gca, 'position', [0 0 1 1], 'Visible', 'off')
figure(figure_dmd);
imagesc(LG_T);
```

A.4.3. Captura de las imágenes en la cámara

Una vez que el haz del láser se reflejaba en el DMD y pasaba por los lentes y el filtro espacial, este llegaba a la cámara para ser capturado. El bloque de código A.11 muestra como primero se creaba una figura para más adelante graficar las capturas de la cámara y en seguido se establecían algunos parámetros para la captura: el número de captura *pics_num*, el tiempo de exposición *exposure*, la velocidad a la que los sensores de las celdas en la cámara se pueden leer *clock* y la velocidad de los fotogramas *rate*. Los últimos tres parámetros mencionados se le dan a la cámara al inicio de cada iteración del ciclo. Luego, se asigna un espacio en la memoria de la cámara para la imagen y se obtiene el identificador de dicho espacio de memoria *MemId*. El identificador de la memoria sirve para localizar el espacio y adquirir las características de la imagen como el ancho *Width*, la altura *Height* y el número de bits *Bits*, tras lo cual se le da la indicación al programa de que espere a que la imagen sea adquirida. Una vez adquirida la imagen, se procede a copiarla en un arreglo temporal unidimensional *tmp* que luego se reacomoda en un arreglo tridimensional tal como una imagen RGB con ayuda de las funciones *reshape* y *permute*.

Bloque de código A.11: Captura de múltiples imágenes

```
figure_camera = figure ;

pics_num = 50;
exposure = 30;
clock = 1;
rate = 1;

for i=1:pics_num

    cam.Timing.Exposure.Set(exposure);
    cam.Timing.PixelClock.Set(clock);
    cam.Timing.Framerate.Set(rate);

    [~, MemId] = cam.Memory.Allocate(true);

    [~, Width, Height, Bits, ~] = cam.Memory.Inquire(
        MemId);

    cam.Acquisition.Freeze(uc480.Defines.DeviceParameter
        .Wait);

    [~, tmp] = cam.Memory.CopyToArray(MemId);

    Data = reshape(uint8(tmp), [Bits/8, Width, Height]);
    Data = permute(Data, [3,2,1]);

    DataN = zeros(Width, Height);
    DataN = Data(:, :, 3);

figure(figure_camera);
```

```

    imagesc(DataN) ;
    colormap hot
    pause(0.1)
end

```

El arreglo *DataN* reserva a priori la memoria necesaria para después guardar uno de los tres canales de las imágenes RGB. El canal capturado se muestra en la figura *figure_camera* creada al principio para las capturas y, por último, se deja una pequeña pausa para asegurar que el holograma en el DMD se actualice.

A.5. Reconocimiento de haces deformados con inteligencia artificial

Los haces Laguerre-Gauss después de ser deformados deben de propagarse, por lo cuál se establecieron las variables del espacio de Fourier y la longitud de onda del láser $\lambda = 660$ nm como se muestra en el bloque de código A.12.

Bloque de código A.12: Establecimiento de las variables en el espacio de frecuencias y la longitud de onda del láser

```

lambda = 660e-9;
Dfx = 1/Wx; Dfy = 1/Wy;
kx = Dfx*(-Nx/2) : Dfx : Dfx*(Nx/2-1);
ky = Dfy*(-Ny/2) : Dfy : Dfy*(Ny/2-1);
[U, V] = meshgrid(kx, ky);

```

A.5.1. Deformación de los haces mediante máscaras de Kolmogorov

El bloque de código A.13 muestra la generación de los haces deformados. El primer paso para crear una red neuronal que clasifique haces Laguerre-Gauss tras ser deformados es determinar las clases. Se establece en la primera línea del código que el total de clases es de ocho ($LG_{01}, LG_{02}, LG_{03}, LG_{11}, LG_{12}, LG_{13}, LG_{21}$) y la segunda línea establece que de cada clase se crearan un total de 100 muestras. El número de muestras o entradas es entonces $num_inputs = 800$ que se almacenaran en un arreglo tridimensional *Input* y 800 etiquetas o salidas que se almacenarán en un arreglo bidimensional *Output* como se explica en el capítulo 4. Las siguientes líneas del bloque de código inicializan tanto los contadores para los ciclos posteriores como el radio de la cintura en el foco $w_0 = 0.8$ mm que será igual para todos los haces generados. El primer ciclo interno itera sobre los índices radiales desde el uno hasta el tres para cada uno de los índices azimutales que van desde el cero hasta el dos en el ciclo externo. Se itera sobre un total de nueve combinaciones de índices radiales y azimutales, pero se especifica justo en la primer sentencia condicional dentro de los ciclos anidados que en la última combinación ($l = 2$ y $p = 3$) se rompa con esa iteración sin hacer nada más.

Bloque de código A.13: Deformación de los haces Laguerre-Gauss con máscaras de Kolmogorov

```

num_classes = 8;
sample_num = 100;
num_inputs = num_classes * sample_num;

Input = uint8(zeros(num_inputs, Ny, Nx));
Output = uint8(zeros(num_classes, num_inputs));

input_index = 1;
output_index = 1;
w0 = 0.0008;

for l = 0:2
    for p = 1:3

        if (l == 2) && (p == 3)
            break
        end

        Gini = lg(X,Y,l,p,w0);

        for sample = 1:sample_num
            r0 = 1.4;
            Medio = kolmo(Nx,Ny,r0);

            G1 = Gini.*Medio;
            F_G1 = fftshift(fft2(G1));

            z1 = 0.5;
            FP_prop = exp(1i*(2*pi*z1/lambda) * sqrt
                (1-((lambda^2)*(U.^2+V.^2))));

            F_G2 = F_G1.*FP_prop;
            G2 = ifft2(fftshift(F_G2));

            A_G2 = abs(G2);
            I_G2 = uint8(255 * mat2gray(A_G2.^2));

            Input(input_index, :, :) = I_G2;
            Output(output_index, input_index) = 1;
            input_index = input_index + 1;
        end
        output_index = output_index + 1;
    end
end
end

```

Después de la sentencia condicional que evita generar una novena clase, se pasa directo a la creación de cada haz Laguerre-Gauss inicial *Gini* a partir del cual se pueden generar las cien deformaciones correspondientes a la clase con ayuda del segundo ciclo. Dentro del ciclo en cuestión, se establece que el parámetro de Fried

sea $r_0 = 1.4$ m y se da como argumento a la función *kolmo* que devuelve la máscara de Kolmogorov usando la ecuación (A.1). El campo deformado G_1 se propaga una distancia $z_1 = 0.5$ m y para ello, se calcula la transformada del campo deformado F_{G1} y la transformada de la función de transferencia a través del espacio libre FP_{Prop} . El producto de las dos transformadas de Fourier del paso anterior es la transformada del campo propagado F_{G2} . Finalmente, se recupera el campo propagado G_2 aplicando la transformada inversa de Fourier y de ahí se calcula su amplitud A_{G2} y su intensidad I_{G2} . La intensidad entonces se guarda como un elemento de la matriz *Input*, mientras que en su correspondiente salida en la matriz *Output* se cambia uno de los ceros por un uno de acuerdo a la tabla del one-hot encoding 4.2.

$$\Phi = \Re(\mathcal{F}^{-1}(\mathbb{M}\sqrt{\phi})) \quad (\text{A.1})$$

La función *kolmo* se muestra en el bloque de código A.14. Se reciben tres parámetros: el número de píxeles a lo ancho N_x , en número de píxeles a lo largo N_y y el parámetro de Fried r_0 . A partir de las dimensiones se crean las coordenadas rectangulares X y Y que a su vez sirven para calcular la coordenada radial R . Dada la coordenada radial y el índice de deformación transversal, se pueden hacer los cálculos de la ecuación (A.1) donde \mathbb{M} es una matriz con coeficientes aleatorios imaginarios, ϕ es la densidad espectral de potencia aproximada como $0.023r_0^{-5/3}r^{-11/3}$, \Re calcula la parte real de los números imaginarios y \mathcal{F}^{-1} es la transformada inversa de Fourier [17].

Bloque de código A.14: Función que crea máscaras de Kolmogorov a partir del parámetro de Fried, la dimensión a lo largo y la dimensión a lo ancho

```
function Phase_screen = kolmo(Nx,Ny,r0)
    x = -Nx/2:1:Nx/2-1;
    y = -Ny/2:1:Ny/2-1;
    [X,Y] = meshgrid(x,y);
    R = sqrt(X.^2 + Y.^2);

    phi = 0.023*(r0.^(-5/3))*R.^(-11/3);
    phi(Ny/2 + 1, Nx/2 + 1) = 0;
    M = randn(Ny, Nx) + 1i*randn(Ny, Nx);
    Phi = real(fft2(fftshift(sqrt(phi).*M)));
    Phase_screen = exp(1i*2*pi*Phi);
end
```

A.5.2. Procesamiento de imágenes

Las imágenes de los haces distorsionados tienen más de un millón de píxeles, por lo cual conviene reducirlas para facilitar su evaluación en la red neuronal. El bloque de código A.15 reduce las imágenes al dividir la imagen en ventanas cuadradas y sacar un solo valor de todas ellas con un simple promedio. El tamaño de las ventanas parte de establecer número deseado de filas *processed_rows* = 21 que debe tener la imagen una vez reducida y de ahí se calcula al número de píxeles que debe tener la ventana cuadrada *window_size*. Luego, se calcula tanto el ancho que debe tener la imagen ya reducida P_x como su largo P_y . El ciclo a continuación itera sobre

cada una de las imágenes en el arreglo bidimensional y con ayuda de la convolución para dos dimensiones *conv2* hace el promedio de ventanas para después guardar las reducciones en un nuevo arreglo tridimensional *Processed_Input*.

Bloque de código A.15: Reducción de las imágenes de los haces deformados

```

processed_rows = 21;
window_size = round(Ny/processed_rows);

Px = fix(Nx/window_size);
Py = fix(Ny/window_size);
Processed_Input = zeros(num_inputs, Py, Px);

for sample = 1:num_inputs
    reduction = conv2(squeeze(Input(sample, :, :)), ones(
        window_size), 'valid');
    Processed_Input(sample, :, :) = reduction(1:
        window_size:end, 1:window_size:end)/window_size^2;
end

```

Antes de usar las reducciones como entradas de la red neuronal, estas tienen que acomodarse como arreglos unidimensionales como se muestra en el bloque de código A.16 donde se usa la función *reshape* para que las imágenes sean una sola columna y estas columnas se guardan en un arreglo bidimensional *Resized_Input*. Nótese que estos arreglos unidimensionales tienen que ser columnas con un número de filas igual a $P_x \times P_y$.

Bloque de código A.16: Imágenes redimensionadas como arreglos unidimensionales

```

Resized_Input = zeros(Py * Px, num_inputs);

for input = 1:num_inputs
    Resized_Input(:, input) = reshape(squeeze(
        Processed_Input(input, :, :)), 1, []);
end

```

A.5.3. Entrenamiento de la red neuronal

Las entradas y salidas para alimentar a la red neuronal están listas en este punto. Basta ahora con diseñar la red neuronal y dividir el conjunto de datos como se muestra en el bloque de código A.17. La red neuronal se obtiene con la función *patternnet* donde se especificó que su diseño constara de tres capas ocultas de 20, 50 y 30 neuronas. Por otro lado, el conjunto de datos se puede dividir automáticamente usando atributos adecuados para la red neuronal, entonces se especificó que el tamaño del conjunto de entrenamiento fuera del 70% = 5600, que el tamaño del conjunto de validación fuera del 15% = 1200 y que el tamaño del conjunto de prueba fuera también del 15% = 1200.

Bloque de código A.17: Diseño de la red neuronal y división del conjunto de datos

```

net = patternnet([20 50 30]);

```

```
net.divideParam.trainRatio = 0.70;
net.divideParam.valRatio = 0.15;
net.divideParam.testRatio = 0.15;
```

El siguiente bloque de código muestra como se alimenta a la red neuronal con las las entradas procesadas *Resized_Input* y las salidas *Output* usando la función *train* que devuelve la red ya entrenada y una estructura *tr* con atributos que se usarán para la visualización de resultados. Cabe mencionar que el entrenamiento es controlado por el número máximo de épocas que en este caso se estableció como 100 y que se puede reducir para obtener una red con menor entrenamiento. El siguiente paso es usar los índices del conjunto de prueba *tr.testInd* para obtener las entradas *test_input* y la salidas *test_output* sobre las cuales se pondrá a prueba la red neuronal. Las predicciones *net_output* se comparan con las salidas reales en una matriz de confusión que tiene diferentes opciones para visualizarse, pero que en este caso se decidió hacer mediante un mapa de calor.

```
net.trainParam.epochs = 100;
[net, tr] = train(net, Resized_Input, Output);

test_input = Resized_Input(:, tr.testInd);
test_target = Output(:, tr.testInd);

net_output = net(test_input);
[c, cm, ind, per] = confusion(test_target, net_output);
```

Apéndice B

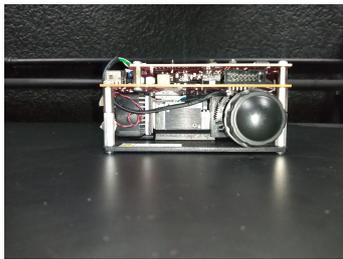
Manipulación del módulo de evaluación DLP3310

B.1. Desmontaje y extracción

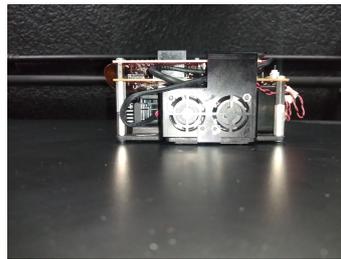
Previo a comenzar con el desmontaje del módulo de evaluación se probó su funcionamiento con las precauciones y el procedimiento indicados en la guía de usuario [1]. Las precauciones incluían: no tocar el módulo de evaluación de la parte superior por la alta temperatura que podía alcanzar, no mirar fijamente la luz emitida por el proyector por la posibilidad de radiación óptica dañina para los ojos y asegurarse de que los ventiladores estuvieran funcionando para evitar el sobrecalentamiento y garantizar un funcionamiento confiable. Se adquirió una fuente de poder externa de corriente directa Toshiba modelo PA3917U-1ACA que cumple con el estándar de seguridad CCC. La fuente de alimentación proporcionaba 19 V de voltaje nominal de salida y una corriente de 3.42 A siendo la máxima corriente permitida para este módulo de evaluación mientras que la mínima era de 2.5 A. Sobre la mesa óptica se colocó el módulo de evaluación, se conectó la fuente de poder de la entrada $J4$ en el subsistema *Display Board* a un enchufe situado a un lado de la mesa óptica. Luego, se movió el switch *SW2* en el mismo subsistema de la posición *OFF* a la posición *ON* para encender el dispositivo haciendo que el proyector mostrara imágenes predeterminadas sobre una pared.

En una computadora portátil se descargó la más reciente versión de la Interfaz Gráfica de Usuario o GUI (por sus siglas en inglés, **G**raphical **U**ser **I**nterface) llamada *DLP LightCrafter Display EVM GUI tool v7.0 Setup.exe* y se siguieron las instrucciones de instalación proporcionadas por el mismo programa. Debido a que el controlador del módulo de evaluación se instala junto con la GUI, no fue necesaria alguna otra instalación. Después, se adquirió un cable que conectara la entrada Mini-USB del módulo de evaluación a la entrada USB de la computadora portátil y, una vez conectados estos dos, se probaron distintas imágenes en la proyección como lo indica el manual de usuario de la GUI [19].

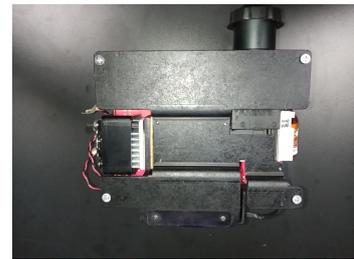
Tras comprobar un correcto funcionamiento, se prosiguió a desconectar y desmontar el dispositivo para obtener el DMD. Además de las precauciones anteriormente mencionadas, se usaron guantes de nitrilo para evitar ensuciar algún componente óptico con las manos y se usó una pulsera electrostática marca Steren conectada a tierra para evitar una descarga que pudiese dañar alguno de los componentes tales como los chips o el mismo DMD. Se tomaron dos medidas importantes para la



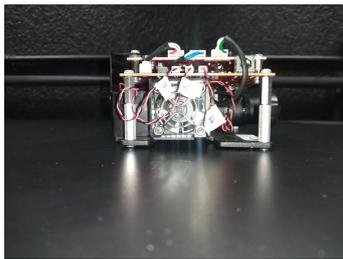
(a) Vista frontal



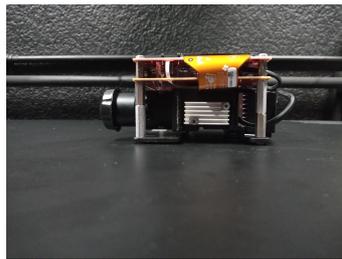
(b) Vista lateral izquierda



(c) Vista inferior



(d) Vista trasera



(e) Vista lateral derecha



(f) Vista superior

Figura B.1: Diferentes vistas del DLP LightCrafter Display 3310 EVM en su estado inicial. En la vista frontal B.1a se observa el proyector. En la vista lateral izquierda B.1b se pueden observar dos de los tres ventiladores. En la vista inferior B.1c se puede observar la placa que más adelante se manipulará para sujetarse a la mesa óptica. En la vista trasera se puede observar el tercer ventilador, los conectores a los tres leds y los conectores a los motores (etiquetados del uno al tres). En la vista lateral derecha se puede observar el cable plano flexible y el disipador térmico que se conectan al DMD. En la vista superior se puede observar a la *Display Board* que contiene, entre otros chips, al controlador del DMD.

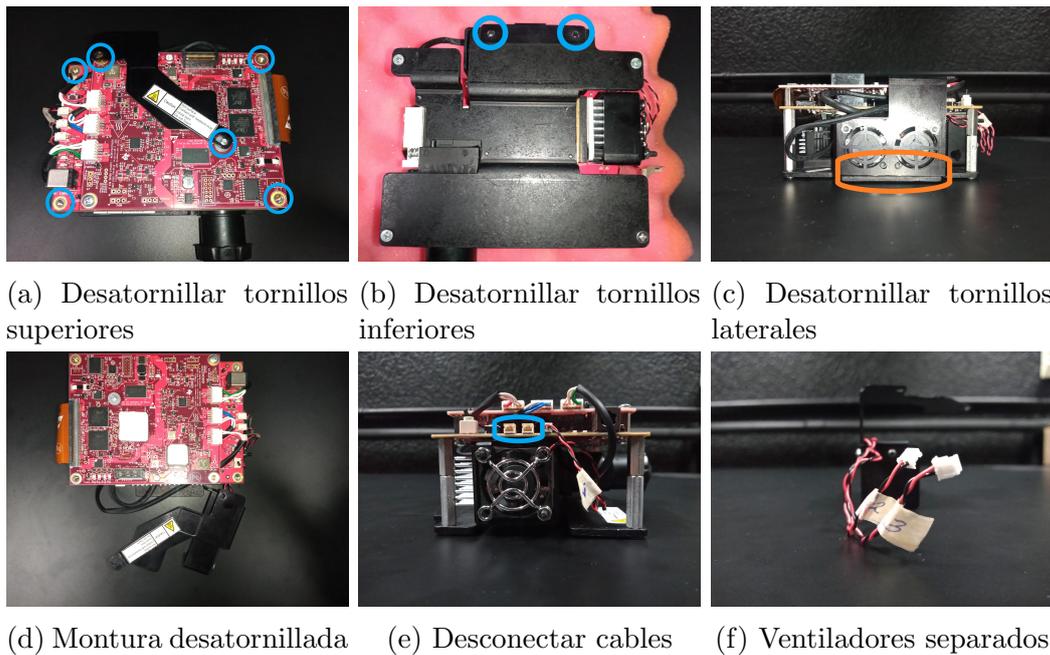
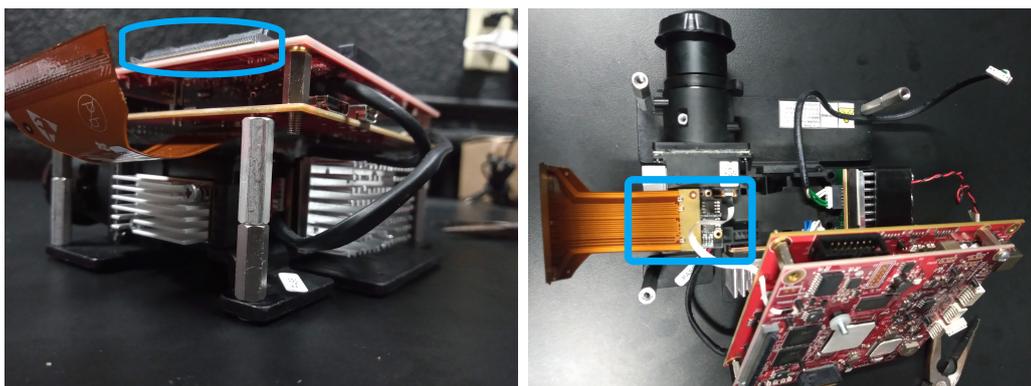


Figura B.2: Desmontaje de los ventiladores del DLP LightCrafter Display 3310 EVM. Se señalan los tornillos que hay que remover en las figuras B.2a, B.2b y B.2c para liberar la montura de los ventiladores tal y como se muestra en la figura B.2d. Se muestran en B.2e los cables que hay que desconectar para separar totalmente la montura de los ventiladores y quede por separado como en la figura B.2f.

organización durante el desmontaje, primero, se etiquetaron los cables de los ventiladores del uno al tres como se muestra en la figura B.1d para evitar confusiones al conectarlos y desconectarlos y, segundo, los tornillos y tuercas se fueron guardando en contenedores según el orden en que se iban removiendo. El desmontaje comenzó desatornillando las tuercas y tornillos de la parte superior del módulo de evaluación que se señalan en la figura B.2a, algunos sostenían a la *System Board* y otros a la montura de los ventiladores laterales (ver figura B.1b). Se desatornillaron después los tornillos hexagonales debajo y de lado del módulo de evaluación señalados en las figuras B.2b y B.2c, respectivamente; dichos tornillos sostenían una placa pequeña que a su vez sostenía la montura de los dos ventiladores laterales. Una vez removida la placa que sostenía a la montura se podía remover fácilmente ésta como se muestra en la figura B.2d excepto por los cables de los ventiladores conectados a la *System Board* etiquetados como 3 y 2 al principio del proceso. Luego de desconectar estos últimos cables, se separaron los ventiladores laterales del módulo de evaluación como se muestra en las figuras B.2e y B.2f, respectivamente.

Una vez separados los ventiladores, el resto del desmontaje era sencillo. Se tenía que desconectar el cable plano flexible del subsistema *Display Board* como se muestra en la figura B.3a para poder desmontar tanto este subsistema como el del *System Board* del resto del módulo de evaluación. Los tornillos que sujetaban a los subsistemas en cuestión ya habían sido retirados en pasos anteriores, por lo cual solo se levantaron suavemente las tarjetas rojas hasta dejar al descubierto el subsistema *Light Engine*. La figura B.3b señala dentro de un recuadro azul el área donde se encontraba el DMD que estaba fijado con pegamento y tornillos. Cuidadosamente se quitaron los tornillos y se hizo palanca con un destornillador plano para despegar



(a) Se desconectó el cable plano flexible que conectaba con el DMD (b) Acceso a la *Light Engine* donde se encontraba el DMD

Figura B.3: Pasos finales del desmontaje del módulo de evaluación. La imagen de la izquierda muestra cómo se desconectó el cable plano flexible y la imagen del lado derecho muestra el subsistema *Light Engine* con el lugar donde se encontraba el DMD encerrado en un recuadro azul.

el DMD junto con el disipador. Finalmente, se limpió la pasta térmica del disipador y el reverso del DMD con alcohol isopropílico y pañuelos para lentes marca THOR-LABS cuidando siempre que la manipulación de los componentes fuera con unas pinzas de plástico como las que se muestra en la figura B.4c evitando así rayaduras y suciedad.

B.2. Impresión de las monturas auxiliares

La *Display Board* conectaba con un cable plano flexible o FFC (por sus siglas en inglés, **F**lexible **F**lat **C**able) que en uno de sus extremos tenía una tarjeta. La tarjeta en el extremo del cable plano flexible estaba en contacto tanto con el DMD como con el disipador térmico, pero con el disipador además estaba atornillada. Por otro lado, entre el DMD y el disipador no existía ninguna conexión en principio y, para ello, se diseñó en AutoCAD un soporte que unía a los tres elementos: el DMD, la tarjeta al final del cable plano flexible y el disipador térmico. Una vista de dicho soporte está en la figura B.5a. La montura se diseñó en base a las dimensiones de los elementos que sostendría y las dimensiones del mismo módulo de evaluación específicamente en la sección donde los elementos se ensamblaban. Las medidas de los largos, anchos, profundidades, posiciones de los agujeros de los tornillos y demás se hicieron con un vernier digital que se muestra en la figura B.4a que tenía una resolución de 0.1 mm y una precisión de ± 0.1 mm. Se imprimió la montura en una impresora 3D marca FLASHFORGE y modelo GUIDER II usando filamento de plástico. Sin embargo, se realizaron varias impresiones por prueba y error porque algunas medidas no se pudieron hacer con el Vernier y se aproximaron poco a poco hasta que fueron funcionales y porque otras veces la precisión de la impresora cambiaba dependiendo la velocidad de impresión.

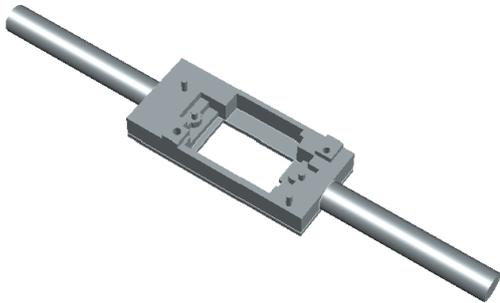
Además de la impresión de la montura para el DMD, se imprimieron otras tres monturas auxiliares como se explica en la sección 3.5. La plataforma con la que se fijaba la base del módulo de evaluación a la mesa óptica se muestra en las figuras



(a) Vernier electrónico digital (b) Impresora 3D FLASHFORGE (c) DMD sostenido por pinzas de plástico

Figura B.4: Herramientas usadas para el diseño e impresión de las monturas auxiliares. Del lado derecho se muestra el vernier digital que se usó para medir las dimensiones, en medio se muestra la impresora 3D donde se imprimieron las monturas que se muestran en la figura y del lado izquierdo se muestra un ejemplo de la manipulación del DMD con pinzas de plástico. Todas las herramientas fueron necesarias para la creación de las monturas mostradas en la figura B.5.

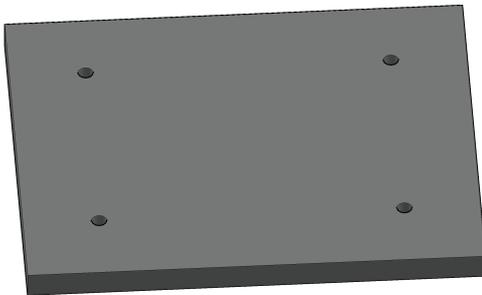
B.5c y B.5d. Uno de los bloques donde se insertaban los extremos de la montura del DMD para poder fijar este a un poste de cada lado se muestran en las figuras B.5e y B.5f.



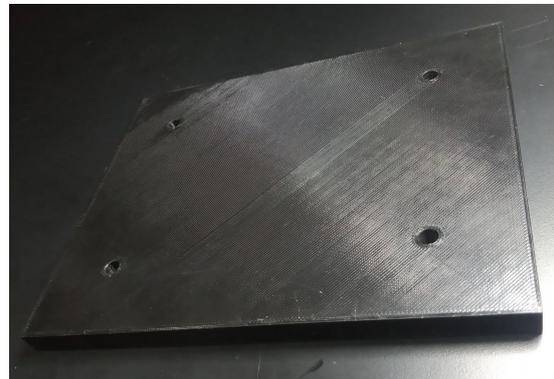
(a) Diseño en AutoCAD de la montura para el DMD



(b) Diseño impreso de la montura para el DMD



(c) Diseño en AutoCAD de la plataforma del módulo de evaluación



(d) Diseño impreso de la plataforma del módulo de evaluación



(e) Diseño en AutoCAD de los bloques auxiliares que complementan con la montura del DMD



(f) Diseño impreso de los bloques auxiliares que complementan con la montura del DMD

Figura B.5: Diseño de las monturas en AutoCAD del lado izquierdo y las monturas ya impresas del lado derecho. Se muestran la montura que unía a los componentes del DMD, la plataforma con la cual se fijaba el módulo de evaluación a la mesa óptica y uno de los bloques que afianzaban la montura del DMD a los postes, en ese orden.

Apéndice C

Modos Laguerre-Gauss experimentales y computacionales

A continuación, se muestran todos los tipos de haces Laguerre-Gauss obtenidos en el experimento para índices azimutales desde $l = 0$ hasta $l = 3$ y para índices radiales desde $p = 1$ hasta $p = 3$. Se muestran primero las imágenes experimentales seguidas de las imágenes de la simulación. Dentro de todas estas imágenes de intensidades están las presentados en la sección 3.6 como resultados del experimento y su simulación.

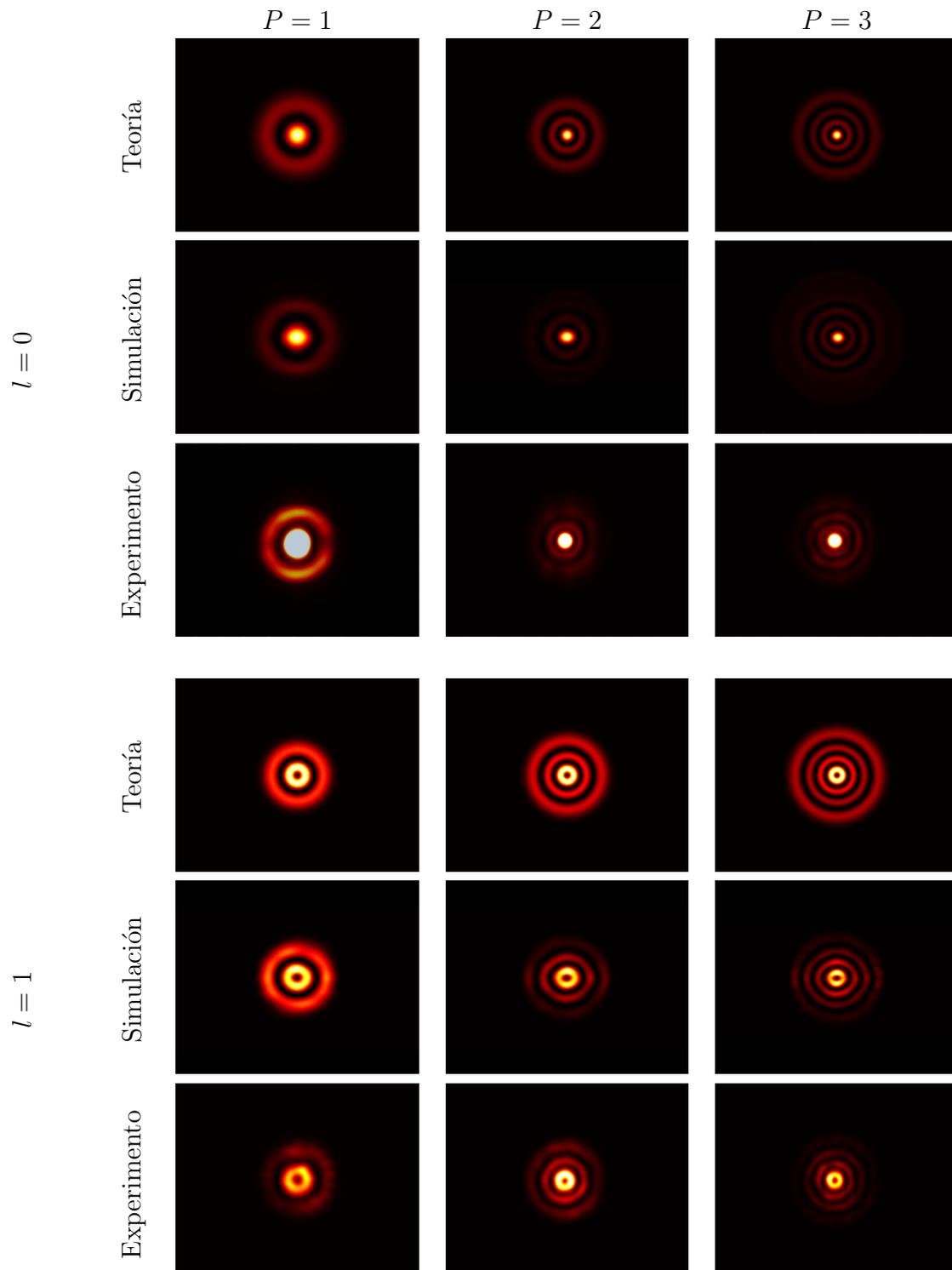


Tabla C.1: Comparación de haces Laguerre-Gauss experimentales y simulados para $l = 0$ y $l = 1$

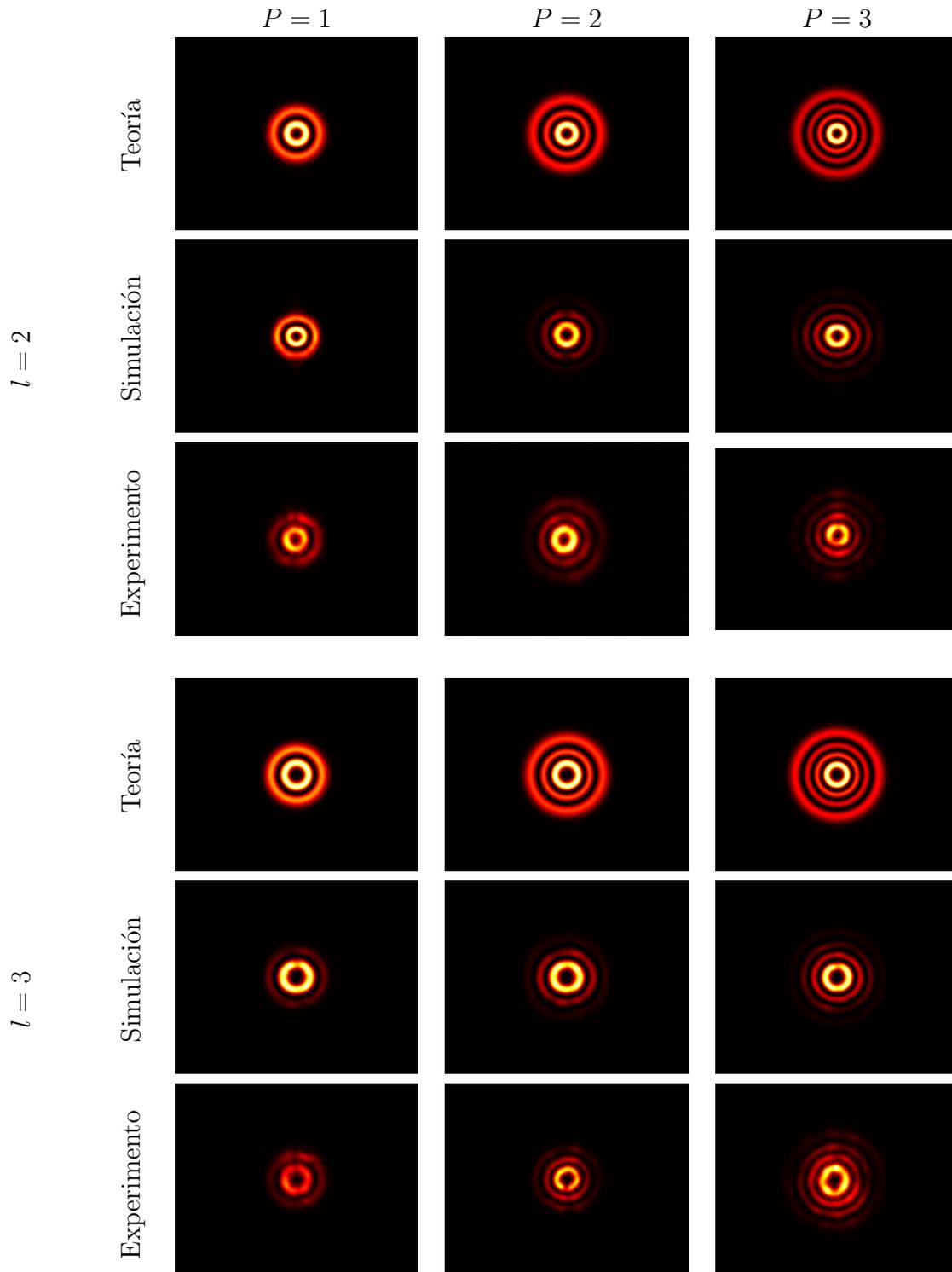


Tabla C.2: Comparación de haces Laguerre-Gauss experimentales y simulados para $l = 2$ y $l = 3$