

FACULTAD DE CIENCIAS
UNIVERSIDAD DE ZARAGOZA
GRADO EN FÍSICA

Trabajo de fin de grado

DESARROLLO E IMPLEMENTACIÓN DE
UNA CÁMARA HIPERESPECTRAL

Autora:

Leyre Larraya Sancho

Directores:

Enrique Carretero Chamarro

Carlos David Heras Vila

Curso 2021-2022

Índice

1. Introducción	2
2. Fundamentos teóricos	3
2.1. Curvas de reflectancia	3
2.2. Coordenadas de color	4
3. Montaje experimental, calibración y procesado de imágenes	7
3.1. Descripción del montaje	7
3.2. Fuentes de luz	8
3.3. Caracterización del filtro	9
3.4. Caracterización de las muestras de color y del blanco con el colorímetro	11
3.5. Caracterización de la cámara	12
3.5.1. Ruido en la imagen	12
3.5.2. Linealidad de la intensidad con el tiempo de exposición	14
3.6. Toma de imágenes y procesado	15
3.6.1. Toma de imágenes	15
3.6.2. Obtención de la reflectancia y correcciones aplicadas	16
3.6.3. Obtención de las coordenadas de color	17
4. Resultados	18
4.1. Muestras de color	18
4.2. Cuadro	21
5. Conclusiones y trabajo futuro	23
Bibliografía	25
Apéndices	
I. Programa de control de la cámara y el motor	I
II. Programa de procesado de las imágenes, para el cálculo de la reflectancia y las coordenadas de color	III

Resumen

El objetivo de este trabajo es desarrollar y construir una cámara hiperspectral, que pueda aplicarse en diferentes proyectos del Grupo de Tecnologías Fotónicas relacionados con el análisis de superficies.

Para ello, se ha caracterizado un filtro pasa-banda variable, cuya banda de transmisión cambia con la posición, que ha permitido recorrer y analizar el espectro visible desde 400 nm hasta 700 nm. También se ha motorizado el sistema, de forma que se ha podido variar la longitud de onda que capta la cámara monocromática que se ha usado para capturar la imagen. Una vez caracterizados los componentes del montaje experimental, se ha medido el espectro reflejado por 6 muestras de color diferentes y se han comparado con el espectro medido con un colorímetro. Para finalizar, hemos tomado la imagen de una obra pictórica para analizarla y hemos obtenido las curvas de reflectancia y las coordenadas de color de cada píxel.

Esta cámara se usará principalmente para el análisis de obras de arte y para el estudio de los pigmentos y barnices utilizados en estas.

1. Introducción

Las cámaras convencionales miden la intensidad de tres bandas amplias de longitudes de onda, que son las correspondientes al rojo, al verde y al azul. Esto permite reconocer la forma y el color del objeto fotografiado, pero no proporciona más información. Por otro lado, las cámaras hiperspectrales toman lo que se conoce como imagen cúbica [1], en la que dos dimensiones son las espaciales y la tercera corresponde a la longitud de onda. Este tipo de cámaras proporcionan un espectro de reflectancia mucho más amplio, y esto nos puede permitir obtener información del material, y su comportamiento frente a la radiación electromagnética. Dado que el comportamiento frente a la luz es diferente para cada material, estudiando un espectro lo suficientemente ancho se pueden identificar los materiales fotografiados, suponiendo que se cuenta con una base de datos con la que comparar.

La fotografía hiperspectral surgió de una investigación de la NASA, en la que se buscaba observar la Tierra remotamente desde el espacio. Esta técnica revolucionó el sensado a distancia, y se extendió a muchos campos diferentes. En la observación de la superficie terrestre, esta herramienta se puede usar para crear mapas de la distribución de minerales o vegetación.

Las cámaras hiperspectrales tienen usos muy diversos, desde la agricultura hasta la visión automática y el área de la biomedicina. Nosotros nos hemos centrado en el análisis de obras pictóricas [2].

Una de estas aplicaciones es revelar imágenes debajo de la pintura, como los dibujos preparatorios del artista. Para esto es necesario poder analizar la reflectancia en el infrarrojo cercano, ya que es conocido por los investigadores en este campo que los pigmentos usados en la pintura son, en general, transparentes en el infrarrojo (en el rango de $1\ \mu m$ a $2\ \mu m$), lo que permite observar estos trazos. También permite observar daños en la obra, como grietas, que han sido reparadas posteriormente.

La fotografía hiperspectral también puede usarse para mejorar la legibilidad de documentos antiguos, que pueden estar corruptos por el deterioro de la tinta, para lo que se observa la imagen en el infrarrojo cercano.

Otro uso muy común es el renderizado del color de la imagen en distintas condiciones de iluminación. Esto no se puede conseguir con las cámaras RGB, ya que éstas capturan la imagen con una iluminación determinada. Sin embargo, como las cámaras hiperspectrales lo que hacen es obtener las curvas de reflectancia en cada píxel de la imagen, se puede calcular el color correspondiente según diferentes condiciones de iluminación, que vienen descritas por los iluminantes, como se explica en la sección 2.2. En este caso, un análisis del espectro visible, y tal vez en el infrarrojo muy cercano, es suficiente.

Otra aplicación muy importante es la monitorización del estado de las obras de arte, observando el cambio en el espectro de las obras antes y después del transporte y del tratamiento que se les realiza para su conservación, y también para controlar

la degradación natural de los pigmentos y barnices. Además, permite controlar las limpiezas que se realizan con láseres, monitorizando la reflectancia en el rango del ultravioleta, ya que este rango es el que mejor permite investigar los posibles daños producidos por este tipo de limpieza.

Dado que los diferentes materiales tienen distintas respuestas espectrales ante la radiación electromagnética, llamadas “huellas espectrales”, se pueden usar las cámaras hiperespectrales para la identificación de los pigmentos usados en una obra. La forma más común en óleos de caracterizar la pintura es tomar pequeñas muestras y analizarlas químicamente, pero esto no es posible en manuscritos, ya que no es posible coger muestras. Para esto sí es necesario un método de identificación no invasivo, como las cámaras hiperespectrales. Es importante saber que, cuando tenemos mezclas de pigmentos, la identificación de estos es mucho más costosa, pero se han desarrollado métodos para hacerlo.

Una ventaja del uso de la fotografía hiperespectral en la identificación de materiales [3], es que permite estudiar la distribución espacial de estos. También permite estudiar las diferentes capas de pigmentos y recubrimientos, como barnices, y permite diferenciar qué partes han sido restauradas, y cómo lo han sido. Sin embargo, un reto que presenta esta técnica es reducir la relación señal-ruido cuando estamos en condiciones de poca luz.

Otro uso que empieza a darse a estas cámaras es la autenticación de obras de arte [4]. Para esto, se realiza un análisis de los pigmentos, y se compara con datos conocidos del tipo de pintura del artista. Esto resalta la necesidad de construir una base de datos con los comportamientos espectrales de los diferentes pigmentos, barnices y materiales usados en la pintura.

El objetivo de este trabajo es obtener una representación correcta de los colores con un número escaso de muestras y comprobar que se pueden diferenciar colores similares. Para hacerlo, nos limitamos al rango visible en el análisis de la reflectancia de las muestras.

2. Fundamentos teóricos

El funcionamiento de una cámara hiperespectral se basa en la obtención de la curva de reflectancia de cada píxel, de forma que se pueden calcular las coordenadas de color y reconstruir la imagen.

2.1. Curvas de reflectancia

Cuando un material es iluminado con una fuente de luz, parte de esta luz es absorbida por la materia, otra parte se transmite (atraviesa el material), y el resto se refleja. El color con el que percibimos la materia opaca que nos rodea se debe a esta luz reflejada, que es la que llega a nuestros ojos.

Para obtener la reflectancia de una muestra para una longitud de onda, la forma más sencilla es medir la intensidad de la luz que refleja en relación a un blanco patrón. El blanco que suele utilizarse es un material que refleja aproximadamente toda la luz de manera isótropa, conocido como espectralón. Si el blanco patrón que se usa no refleja el 100 % de la luz, hay que tener en cuenta su reflectancia a la hora de calcular la de la muestra. En este caso, la reflectancia viene dada por la ecuación 1, en la que la reflectancia del blanco, R_B , está dada en tanto por uno.

$$R(\lambda) = \frac{I_{muestra}(\lambda)}{I_{espectralon}(\lambda)/R_B(\lambda)} \quad (1)$$

La curva de reflectancia de una muestra se construye calculando la reflectancia para diferentes longitudes de onda en el rango de interés, en nuestro caso este rango es el visible, desde 400 nm hasta 700 nm.

2.2. Coordenadas de color

Para poder representar una imagen por pantalla, necesitamos calcular las coordenadas RGB, que podemos obtener a partir de las coordenadas XYZ, para las cuales necesitamos la curva de reflectancia de la muestra, que obtenemos con nuestra cámara.

Las coordenadas XYZ vienen dadas por las ecuaciones 2, 3 y 4, donde $R(\lambda)$ es la curva de reflectancia que hemos obtenido, $S(\lambda)$ es el iluminante utilizado para observar la muestra y \bar{x} , \bar{y} y \bar{z} son los triestímulos definidos por la Comisión Internacional sobre la Iluminación (Comission Internationale de l'Eclairage, CIE) para un observador patrón. En este caso, se han utilizado dos iluminantes diferentes, el iluminante D65 y un iluminante definido a partir de nuestro LED (es decir, su espectro, en muchos casos normalizado a 100 en 560 nm), y los triestímulos definidos para el 10^o observador de color, también llamado observador de color 1964.

$$X = 100 \frac{\int_{380}^{780} R(\lambda) S(\lambda) \bar{x}(\lambda) d\lambda}{\int_{380}^{780} S(\lambda) \bar{y}(\lambda) d\lambda} \quad (2)$$

$$Y = 100 \frac{\int_{380}^{780} R(\lambda) S(\lambda) \bar{y}(\lambda) d\lambda}{\int_{380}^{780} S(\lambda) \bar{y}(\lambda) d\lambda} \quad (3)$$

$$Z = 100 \frac{\int_{380}^{780} R(\lambda) S(\lambda) \bar{z}(\lambda) d\lambda}{\int_{380}^{780} S(\lambda) \bar{y}(\lambda) d\lambda} \quad (4)$$

Dado que las medidas realizadas son discretas y no contamos con medidas en todo el rango de longitudes de onda necesarias, hemos sustituido las integrales por sumatorios en nuestro rango de longitudes de onda, que comprende desde 410 nm hasta 700 nm. Hemos elgido este rango porque, para las longitudes de onda más cortas, la medida se realiza con mucho error, tanto por el filtro como por la poca intensidad de luz en esa zona. Además, los valores triestímulo en este rango son pequeños comparados con su valor en otras zonas del espectro visible, por lo que podemos despreciar su contribución

a las integrales de las ecuaciones 2, 3, 4. Por esta misma razón podemos despreciar la contribución de las longitudes de onda por encima de 700 nm.

Una vez obtenidas las coordenadas XYZ, podemos emplear las matrices de conversión XYZ-RGB que hemos podido encontrar en [5]. Estas matrices están definidas para los blancos de referencia de los iluminantes patrón, por lo que no hay problema en usarlas directamente con el iluminante D65, pero cuando usamos el definido con nuestro LED tendremos que implementar un algoritmo de adaptación cromática, ya que las coordenadas del blanco de nuestra fuente no coinciden con las del blanco del D65.

Para esta adaptación hemos usado el método de Bradford, que podemos ver en [6]. El método utilizado en el algoritmo nos indica la matriz $[M_A]$ que utilizaremos para obtener la matriz de transformación $[M]$ que tendremos que aplicar a las coordenadas XYZ que hemos calculado para poder usar la matriz XYZ-RGB.

La transformación que tenemos que realizar es la dada por la ecuación 5, donde X_D , Y_D y Z_D son las coordenadas de destino, las que podemos usar con la matriz de transformación a RGB, y X_S , Y_S y Z_S son las coordenadas que hemos calculado con las curvas de reflectancia y el iluminante correspondiente.

$$\begin{bmatrix} X_D \\ Y_D \\ Z_D \end{bmatrix} = [M] \begin{bmatrix} X_S \\ Y_S \\ Z_S \end{bmatrix} \quad (5)$$

La matriz $[M]$ la calculamos como se indica en la ecuación 6, donde (ρ, γ, β) son las coordenadas en el dominio de respuesta cónica, que se calculan según la ecuación 7, donde $i = S, D$. En esta ecuación, X_W , Y_W y Z_W son las coordenadas del blanco, que se calculan como se explica más adelante.

$$[M] = [M_A]^{-1} \begin{bmatrix} \rho_D/\rho_S & 0 & 0 \\ 0 & \gamma_D/\gamma_S & 0 \\ 0 & 0 & \beta_D/\beta_S \end{bmatrix} [M_A] \quad (6)$$

$$\begin{bmatrix} \rho_i \\ \gamma_i \\ \beta_i \end{bmatrix} = [M_A] \begin{bmatrix} X_{Wi} \\ Y_{Wi} \\ Z_{Wi} \end{bmatrix} \quad (7)$$

Una vez tenemos las coordenadas XYZ transformadas y expresadas en relación al blanco D65, podemos aplicar la transformación a RGB. Para ello tenemos que elegir el espacio de color, que nos determina la matriz a utilizar. Los espacios más comunes son sRGB y adobe, cuyo rango de color podemos observar en el diagrama de color mostrado en la Figura 1. En este diagrama podemos ver también que podemos obtener el mismo color usando curvas de reflectancia diferentes, o combinandolas, pero hay que destacar que a cada curva de reflectancia le corresponde un único color, una vez fijado el iluminante.

Esta transformación es lineal, pero la codificación RGB no lo es, por lo que tenemos que aplicar una corrección a los valores obtenidos, conocida como corrección “Gamma”

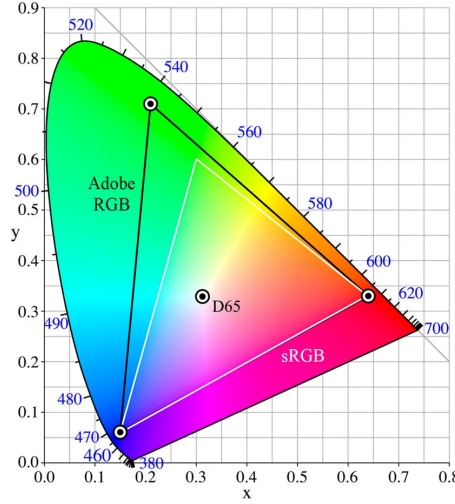


Figura 1: Comparación del rango de color de los espacios sRGB y Adobe

[7]. Para aplicar esta corrección necesitamos que los valores (r,g,b) obtenidos estén en el rango (0,1). La corrección aplicada viene dada por la ecuación 8, donde V es el valor de la coordenada (R,G,B) final, y v es el valor lineal que hemos obtenido con la matriz de transformación, (r,g,b).

$$V = \begin{cases} 12,92v & \text{si } v \leq 0,0031308 \\ 1,055v^{1/2,4} - 0,055 & \text{en otro caso} \end{cases} \quad (8)$$

Aplicando esta corrección, obtenemos los valores finales de las coordenadas RGB, en el rango (0,1). Si lo necesitamos, podemos reescalarlos en el rango (0,255), sencillamente multiplicando por 255.

A continuación se pueden ver los valores de las matrices utilizadas, tanto $[M_A]$, la matriz del método Bradford, como las matrices de transformación a RGB, tanto para el espacio sRGB como para adobe, que llamamos $[M_{sRGB}]$ y $[M_{adobe}]$.

$$[M_A] = \begin{bmatrix} 0,8951 & 0,2664 & -0,1614 \\ -0,7502 & 1,7135 & 0,0367 \\ 0,0389 & -0,0685 & 1,0296 \end{bmatrix}$$

$$[M_{sRGB}] = \begin{bmatrix} 3,2404542 & -1,5371385 & -0,4985314 \\ -0,9692660 & 1,8760108 & 0,0415560 \\ 0,0556434 & -0,2040259 & 1,0572252 \end{bmatrix}$$

$$[M_{adobe}] = \begin{bmatrix} 2,0413690 & -0,5649464 & -0,3446944 \\ -0,9692660 & 1,8760108 & 0,0415560 \\ 0,0134474 & -0,1183897 & 1,0154096 \end{bmatrix}$$

El cálculo de las coordenadas del blanco para diferentes iluminantes se realiza según las ecuaciones 2, 3 y 4, pero tomando $R(\lambda) = 1 \forall \lambda$.

3. Montaje experimental, calibración y procesamiento de imágenes

3.1. Descripción del montaje

Para construir y calibrar la cámara hiperespectral se han usado los siguientes componentes, que se ven en la Figura 2:

- Cámara CMOS USB3.0 monocromática de la marca MIGHTEX.
- Filtro variable lineal LVF-HL de Ocean Insight.
- Motor paso a paso de 60 mm de recorrido de ThorLabs.
- Dos LEDs Cree blancos.
- Dos lentes de focal corta.
- Rendija de anchura 1.2 mm.
- Espectroradiómetro USB4000-UV-VIS de OceanView.
- Colorímetro CR-400 de Konica Minolta.

El espectroradiómetro se ha utilizado para medir las bandas de transmisión del filtro y el espectro de los LEDs, y con el colorímetro se han medido las curvas de reflectancia de las muestras de color, para poder comparar las curvas obtenidas con la cámara. Este espectroradiómetro permite medir en todo el rango del visible, y puede medir también en parte del ultravioleta e infrarrojo cercano. La calibración de las fuentes de luz se ha hecho usando un espectralón, y midiendo la luz que refleja.

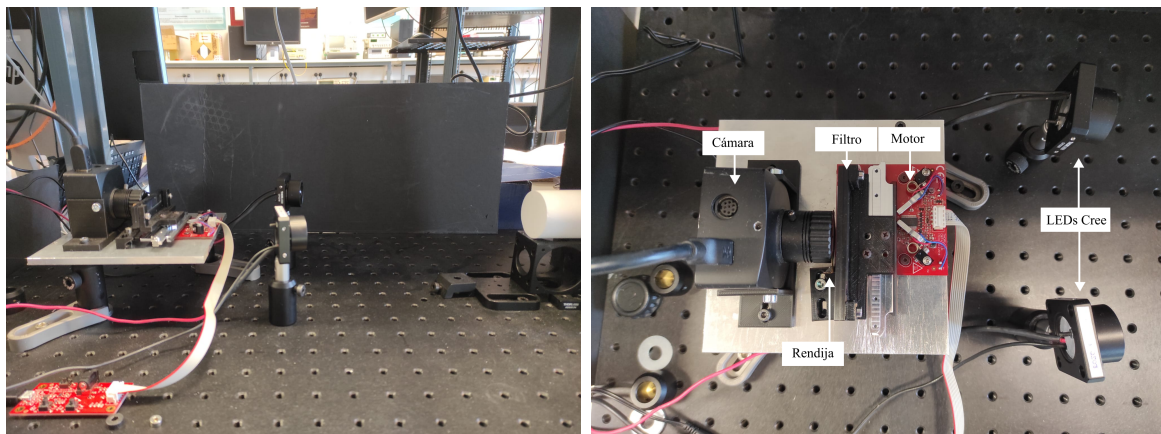


Figura 2: Montaje de laboratorio de la cámara hiperespectral

Al principio contábamos con un motor de 25 mm de recorrido, pero vimos que no podíamos recorrer todo el rango del espectro que queríamos, por lo que cambiamos al motor de 60 mm que usamos en el montaje final.

La incorporación de la rendija nos ha permitido limitar la cantidad de luz que llega a la cámara, y con ello la anchura de la banda de longitudes de onda que medimos. Así, al mover el motor, y por lo tanto el filtro, varían las longitudes de onda que llegan a la cámara. Por esta razón, en adelante hablaremos de forma indistinta de la longitud de onda que llega a la cámara y de la posición del filtro.

El filtro lineal que hemos utilizado está formado por dos filtros superpuestos, uno pasa-alta y otro pasa-baja. Al variar la posición relativa de los dos filtros, obtenemos un filtro pasa-banda, en el que podemos seleccionar la anchura de la banda.

3.2. Fuentes de luz

Al plantearnos la iluminación en el sistema, encontramos dos opciones, usar lámparas halógenas o usar LEDs. Empezamos el proceso utilizando una lámpara halógena, pero vimos que no proporcionaba suficiente luz en el rango de los azules, por lo que pasamos a usar LEDs blancos de 4000 K. Estos LEDs tenían un pico de emisión en torno a los 420 nm, pero había una bajada importante de luz en la zona del verde, por lo que los cambiamos por los LEDs Cree blancos que se usaron finalmente para las medidas. Estos LEDs tienen el pico de emisión azul en 460 nm, y diferentes fósforos que absorben esta luz azul y la reemiten en diferentes bandas de luz. Esto nos da un espectro con picos a varias longitudes de onda diferentes y una intensidad comparable en el resto del espectro emitido.

Medimos los espectros tanto de la lámpara halógena como de los LEDs Cree blancos usando el espectroradiómetro, obteniendo los espectros que se muestran en la Figura 3 y en la Figura 4. El espectro de la lámpara halógena se tomó con un tiempo de integración de 5 segundos, mientras que el de los LEDs se hizo para 85 milisegundos. Esto resalta la diferencia de intensidades entre ambas fuentes, e indica que los LEDs podrían ser una mejor opción, pero el rango en longitud de onda es más corto, apenas llegando a 400 nm. Además, como la intensidad de la luz es más baja y se necesita un tiempo de integración mucho mayor, el espectro de la lámpara halógena tiene más ruido que el del LED. Sin embargo, los Cree tienen menor contribución en el infrarrojo. Esta menor contribución en el IR resulta beneficiosa, ya que no contamos con un filtro de infrarrojo en el montaje.

Así, la lámpara halógena sería beneficiosa para medir la zona de los azules, entre 390 y 420 nm, mientras que los LEDs Cree son más ventajosos en el resto del espectro, entre 420 y 700 nm.

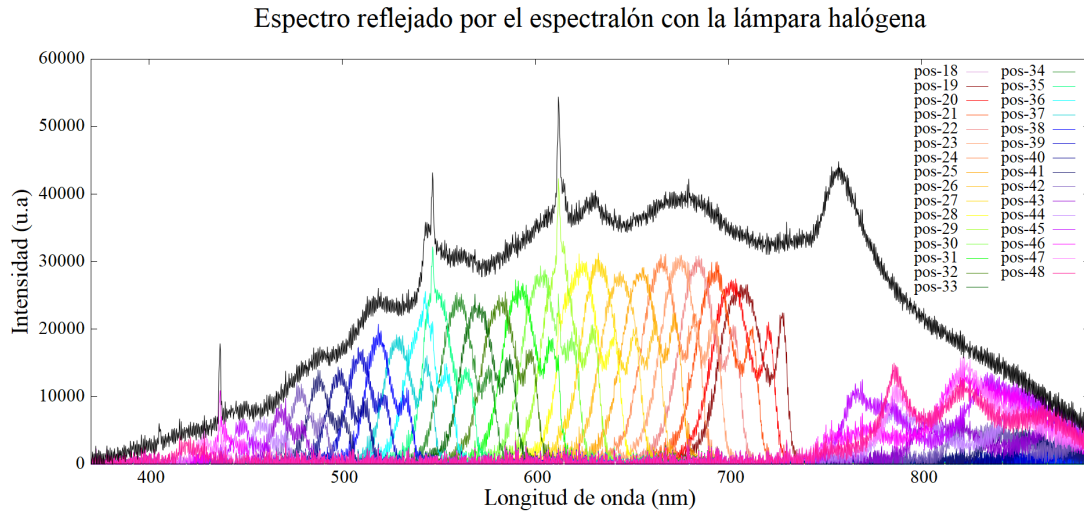


Figura 3: *Espectro de la fuente halógena*

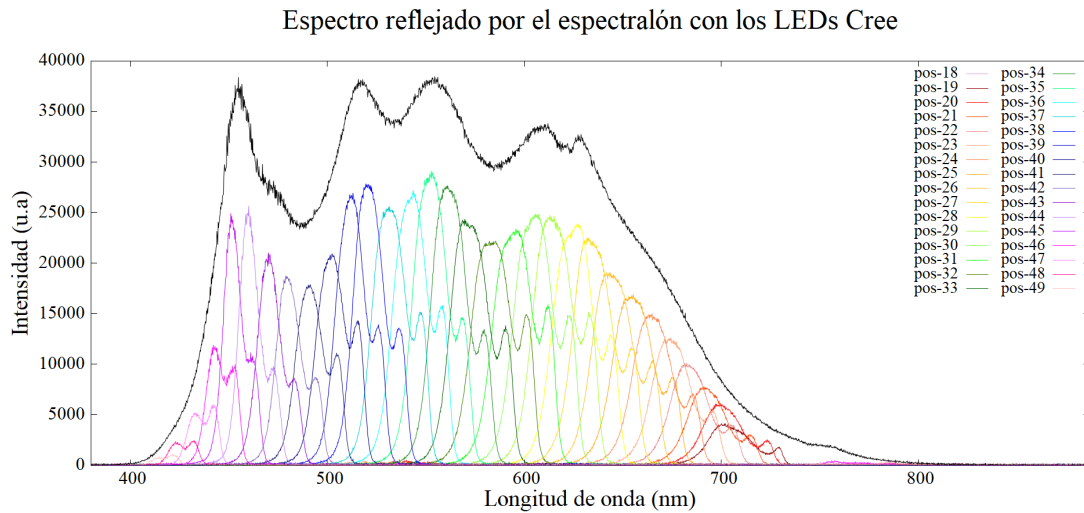


Figura 4: *Espectro del diodo LED*

Inicialmente el montaje solo contaba con una fuente de luz, pero la iluminación en el plano del objetivo era muy poco uniforme y asimétrica, así que decidimos colocar dos fuentes de luz, como puede verse en la Figura 2. Con este montaje obtuvimos una mayor homogeneidad en la iluminación, y un aumento tanto en la simetría como en la intensidad de la iluminación.

3.3. Caracterización del filtro

Para caracterizar el filtro, usamos el espectroradiómetro para medir la banda de luz que el filtro deja pasar según la parte del mismo que atraviesa la luz medida. Esta posición la medimos usando la posición en la que se encuentra el motor paso a paso.

En la Figura 3 y la Figura 4 podemos ver las diferentes bandas de transmisión del filtro para cada posición del mismo. Vemos que la anchura de las bandas varía según

la posición entre 20 y 50 nm, aproximadamente. Podemos ver que las bandas del azul también dejan pasar cierta luz en el IR, entre 780 y 870 nm, que es en proporción bastante mayor a la luz azul que pasa, en el caso de la lámpara halógena, o comparable pero menor a la luz azul, en el caso de los LEDs. Esto introduce una sobreestimación de la luz que pasa en el rango de los azules, pero el error es menor para los LEDs, ya que proporciona menor cantidad de luz en el IR.

Si en un futuro contáramos con un filtro IR en el rango adecuado, podría solucionarse el problema que nos causan las bandas en el IR, y podría plantearse usar la lámpara halógena, cuyo espectro nos permitiría medir en las longitudes de onda cercanas al ultravioleta.

Además, hemos comprobado que el filtro es lineal, calculando la posición del máximo de la banda de transmisión y realizando un ajuste lineal de la longitud de onda en función de la posición del motor. Este ajuste se puede ver en la Figura 5, y la ecuación 9 nos describe la longitud de onda que deja pasar el filtro en cada posición.

$$\lambda(nm) = (-9,50 \pm 0,10) \cdot Pos(mm) + (886 \pm 5) \quad (9)$$

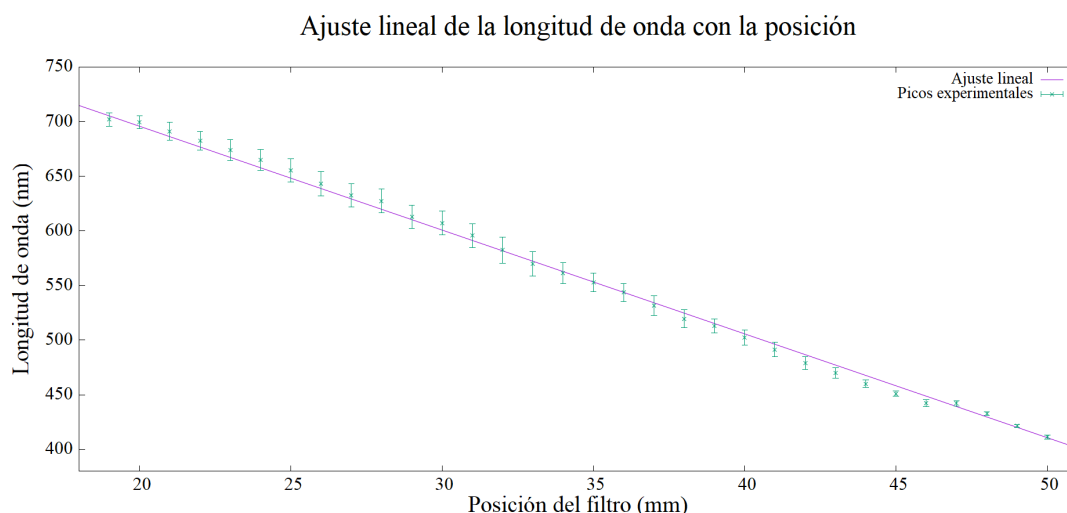


Figura 5: Ajuste lineal de la posición del pico en longitud de onda y la posición del filtro

Usando el espectro y las bandas de transmisión del LED, hemos calculado la transmitancia del filtro en función de su posición, mostrada en la Figura 6. Podemos ver que la transmitancia es aproximadamente constante en el valor máximo de la banda, en torno al 75 % en el rango de 450 nm a 700 nm. Sin embargo, las bandas de transmisión se van estrechando y la transmitancia va disminuyendo conforme nos vamos acercando al azul por lo que en esa zona tendremos menos luz. También podemos ver que al acercarnos a los 400 nm empieza a haber mucho más ruido, lo que se debe a que los LEDs no nos dan mucha luz en ese rango del espectro. Lo mismo pasa por encima de los 750 nm, salvo porque podemos ver en el espectro de la lámpara halógena una banda de transmisión de infrarrojo que aparece cuando el filtro se encuentra en la zona de los

azules. También aparece en el del LED, pero es mucho menos intensa. Esto afectará a las curvas de reflectancia, ya que hace que tengan un valor mayor en la zona del azul del que deberían tener, como hemos explicado anteriormente.

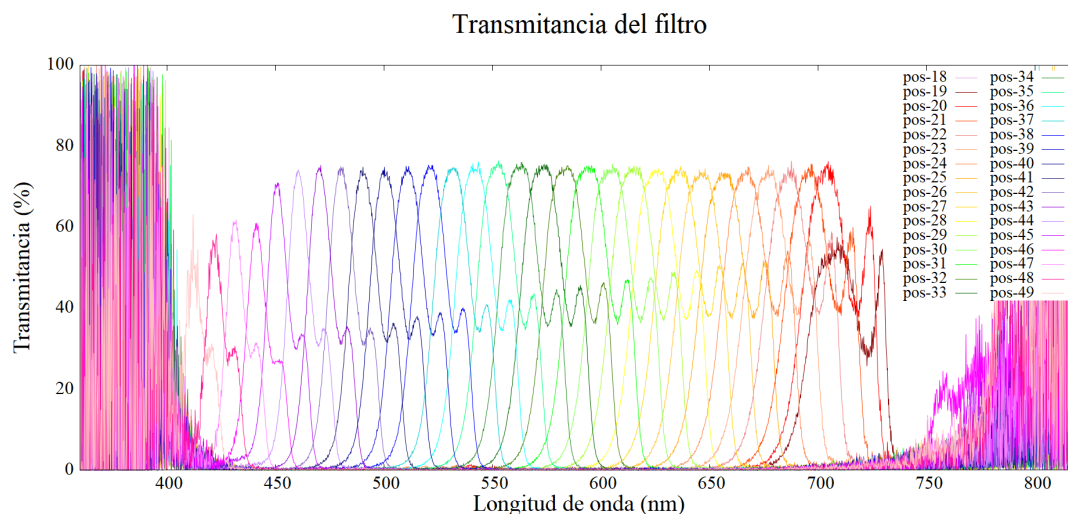


Figura 6: *Transmitancia del filtro en función de la longitud de onda*

Uno de los problemas más limitantes que encontramos en el desarrollo del proyecto, aparte del espectro de los LEDs, fue en el filtro. Por el propio diseño de este dispositivo, este tiene una rendija en el centro, en la zona en la que se unen los dos filtros que forman el conjunto. Esta rendija produce reflejos en la imagen tomada por la cámara cuando la luz pasa por esta posición o por las adyacentes. Esta zona coincide con los azules, en torno a 400 nm, por lo que no podemos medirlos con precisión. Es posible que este defecto también influya en las medidas de reflectancia de esta zona, haciendo que su valor sea mayor que el real.

Existe un filtro del mismo fabricante sin esta rendija, por lo que para la futura mejora del dispositivo podría ser conveniente reemplazar el filtro utilizado. No utilizamos este filtro porque era más caro, y ya teníamos el que hemos estudiado.

También es necesario comentar que la posición del objeto influye en la posición del filtro que atraviesa la luz reflejada, y esto produce cierta variación en la longitud de onda que llega a la cámara. El resultado de este fenómeno es que las curvas de reflectancia se desplazan, pero el cambio es despreciable para el campo de visión de nuestra cámara.

3.4. Caracterización de las muestras de color y del blanco con el colorímetro

Hemos medido la reflectancia de las muestras usando un colorímetro, que promedia en una zona de la muestra la luz que ésta refleja. Más adelante compararemos estas medidas con las curvas de reflectancia que obtenemos con la cámara, que es la reflec-

tancia de cada píxel. Para comparar, tomaremos la curva de un píxel por el centro de la muestra, y veremos si mantiene la forma y el valor de la reflectancia en cada punto.

Para calcular las curvas, necesitamos medir también un blanco con el que comparar. Empezamos usando un espectralón, que es un blanco patrón de reflectancia isótropa aproximadamente 1 en todas las longitudes de onda del visible. Sin embargo, el espectralón con el que contamos era del tamaño de las muestras, y esto nos dio problemas cuando quisimos calcular la reflectancia de toda la imagen, que incluye la muestra de color y el fondo. Para solucionar este problema, cambiamos el blanco, y cogimos un folio blanco. Para poder usarlo, necesitamos conocer su curva de reflectancia, para poder utilizarla en el cálculo de la reflectancia con la ecuación 1.

Para medir la curva de reflectancia del blanco usamos el mismo colorímetro con el que medimos las muestras. De esta manera se minimiza un poco el error, ya que tendrá la misma incertidumbre que las curvas de las muestras.

En la Figura 7 podemos ver las curvas que obtuvimos con el colorímetro.

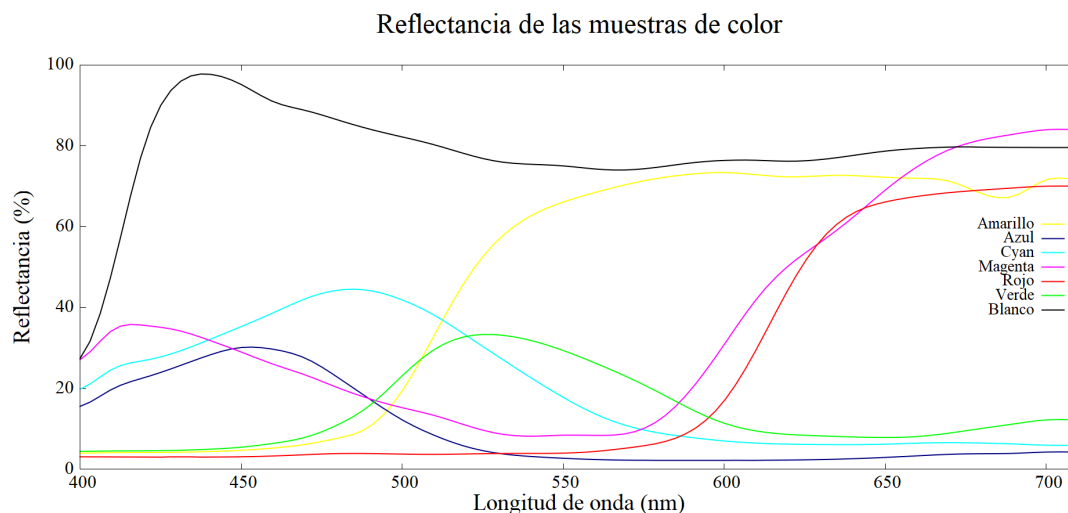


Figura 7: Curvas de reflectancia de las muestras y del folio medidas con el colorímetro

3.5. Caracterización de la cámara

El siguiente paso fue determinar el ruido que introducía la cámara en las imágenes, y estudiar cómo variaba la intensidad en un píxel según el tiempo de exposición con el que se tomaba la fotografía.

3.5.1. Ruido en la imagen

Para estudiar el error de la cámara, que caracterizamos como ruido térmico, tomamos 20 imágenes en la misma posición y con el mismo tiempo de exposición, y comparamos el valor de cada píxel en cada imagen. Si no hubiera ruido, todas las imágenes tendrían el mismo valor para un píxel determinado. Como hemos podido

comprobar en las imágenes capturadas, esto no es así, por lo que para calcular la reflectancia tomaremos el promedio de varias imágenes, y calcularemos el error como la desviación estándar de las intensidades. En la Figura 8 podemos ver cómo varía el ruido de la cámara en función de la intensidad normalizada del píxel, en el intervalo de 68 % de confianza .

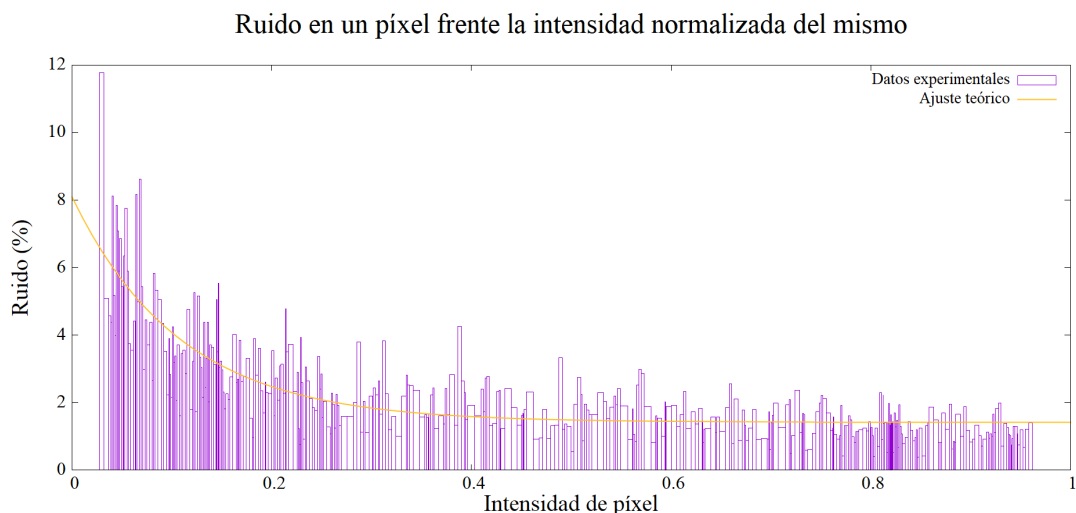


Figura 8: *Ajuste del ruido en un píxel según la intensidad en el mismo*

Podemos expresar el ruido observado en un píxel según la intensidad en dicho píxel usando la ecuación 10, como podemos ver en la Figura 8.

$$Ruido(I) = a \exp\{-bI\} + c \quad (10)$$

Usando Gnuplot, podemos obtener los valores de las constantes a , b y c .

$$a = 6,7 \pm 0,4$$

$$b = 9,2 \pm 0,7$$

$$c = 1,41 \pm 0,06$$

Como podíamos esperar, se observa que el ruido es más significativo en las zonas oscuras, por lo que decidimos tomar imágenes de la misma posición del filtro usando diferentes tiempos de exposición, lo que aumenta la intensidad en estas zonas. Esto puede saturar la imagen en zonas de la muestra, por lo que en cada píxel elegiremos la imagen con mayor intensidad en dicho píxel sin que esté saturada.

Entre los diferentes tiempos de exposición, y la toma de varias imágenes con cada uno de ellos para realizar un promedio, el tiempo que se tarda en tomar las medidas aumentó considerablemente, por lo que tuvimos que limitar el número de fotografías que tomamos. Al final, para la obtención una imagen final se toman 900 fotografías de la muestra, 25 para cada posición, y tardaba 30 minutos.

3.5.2. Linealidad de la intensidad con el tiempo de exposición

Para estudiar la dependencia de la intensidad en un píxel con el tiempo de exposición utilizado, tomamos cuatro fotografías con diferentes tiempos de exposición. Después dividimos cada imagen por la exposición correspondiente, y comparamos los diferentes valores de los píxeles. Podemos ver que los valores coinciden, lo que indica que la intensidad en un píxel es lineal con el tiempo de exposición utilizado.

Tomamos las imágenes de las muestras de color, del blanco y del cuadro que analizamos al final en diferentes momentos, y usando diferentes tiempos de exposición, mostrados en la Tabla 1. Con esta variedad de tiempos de exposición, nos aseguramos de tener los máximos valores posibles en los píxeles para reducir el error en la medida.

	T_1 (ms)	T_2 (ms)	T_3 (ms)	T_4 (ms)	T_5 (ms)
Blanco	1	3	5	10	50
Muestras de color	1	3	5	10	17
Cuadro	1	4	20	50	100

Tabla 1: *Tiempos de exposición utilizados para la toma de medidas*

Hemos medido la intensidad del píxel (250, 250) usando el folio blanco, y hemos comprobado que el tiempo de exposición es lineal. La cámara mide en un rango de intensidad de $[0, 255]$, y los errores se han calculado usando la Ecuación 10.

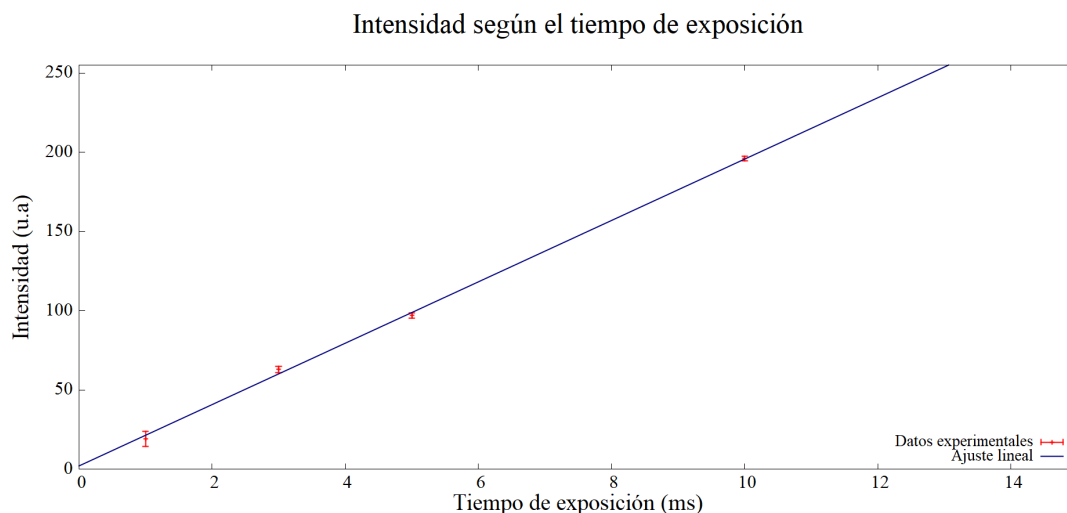


Figura 9: *Dependencia de la intensidad en el píxel (250, 250) del blanco extenso (folio) según el tiempo de exposición.*

Cada píxel saturará para un tiempo de exposición diferente, y este tiempo también dependerá del objeto que se fotografía. Sin embargo, mientras no sature, la intensidad es lineal con el tiempo de exposición de la cámara. En la Figura 9 podemos ver que el tiempo de exposición $T_5 = 50 \text{ ms}$ del folio estará saturado, ya que el valor que le

correspondería es mucho mayor que 255, que es el valor máximo de la intensidad en un píxel.

3.6. Toma de imágenes y procesamiento

Una vez tenemos caracterizados los diferentes elementos del sistema, podemos empezar a tomar las fotografías y a centrarnos en el procesamiento de las mismas. Este proceso tiene dos partes más o menos diferenciadas. La primera es la obtención de las curvas de reflectancia a partir de las imágenes tomadas, y la segunda es el cálculo de las coordenadas de color de cada píxel. Contamos con programas diferentes para cada parte del proceso.

3.6.1. Toma de imágenes

El proceso de la toma de fotografías se ha automatizado usando Matlab y Visual Studio, de forma que contamos con un programa que puede controlar el motor y la cámara al mismo tiempo. La base de este programa la realizó un compañero del Grupo de Tecnologías Fotónicas, y yo lo modifiqué posteriormente conforme necesitaba tomar más imágenes. El programa se presenta en el Apéndice I. En la Figura 10 se muestra un ejemplo de las imágenes que capta la cámara en cada posición para el blanco y para una de las muestras de color.

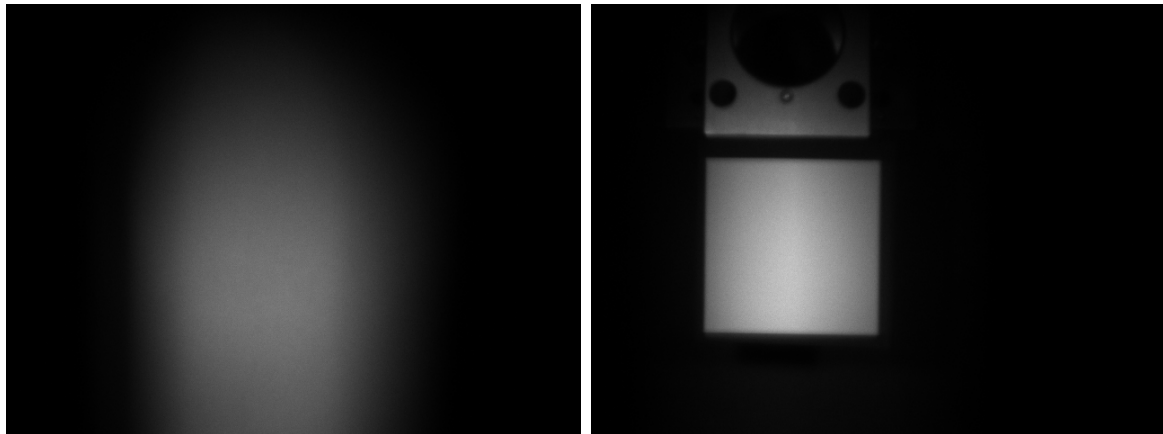


Figura 10: *Imagen tomada por la cámara para el blanco extenso (izquierda) y el amarillo (derecha) en la posición 18 mm del filtro y para un tiempo de exposición de 3 milisegundos*

El programa de Visual Studio configura la cámara, y los tiempos de exposición a utilizar para la toma de las fotografías, mientras que el programa de Matlab controla el movimiento del motor y el momento en el que se toman las imágenes. También se encarga de renombrar las imágenes, para facilitar su identificación, y de guardarlas en una carpeta diferente.

3.6.2. Obtención de la reflectancia y correcciones aplicadas

Para el procesamiento de las imágenes obtenidas se ha utilizado el código que he implementado en Matlab para este proyecto y que se presenta en el Apéndice II. Este programa lo ejecutamos una vez para cada muestra, ya que trabaja con un solo objetivo en cada iteración. Lo primero que hacemos es, para cada posición, abrir las imágenes de dicha posición y promediarlas según su tiempo de exposición. Luego seleccionamos en cada píxel el que tenga un mayor valor sin llegar a saturarse, y lo dividimos entre su exposición. De esta manera, obtenemos una hipermatriz con la intensidad de cada píxel para cada posición (o equivalentemente, para cada longitud de onda), en la que se ha minimizado todo lo posible el ruido de cada píxel.

Este proceso lo repetimos para el blanco, de forma que obtuvimos dos hipermatrices de la misma dimensión y en la que tenemos almacenadas las intensidades normalizadas de cada píxel en función de su longitud de onda. Una vez tenemos estas matrices, podemos aplicar la Ecuación 1 en cada píxel y así obtenemos la reflectancia de cada píxel y para cada una de las longitudes de onda que medimos.

Pese a que hemos intentado obtener la máxima intensidad en cada píxel, hemos colocado delante de la cámara una rendija que limite la anchura de la banda de longitudes de onda que llega a la CCD. El problema es que esto también limita el tamaño de la imagen que tomamos, y encontramos que tenemos parte de la imagen que sigue siendo negra. Esto implica que el valor de los píxeles en estas zonas es cero, tanto para el blanco como para la muestra, y al calcular las curvas de reflectancia esto nos da problemas, al dividir por cero. Para solucionarlo, hemos añadido una condición en el cálculo de la reflectancia, de forma que si la intensidad en el blanco es nula, el valor de la reflectancia de ese píxel se fija a cero.

Por otro lado, el ruido de la cámara en esas zonas negras es muy grande, lo que hacía que la reflectancia fuera mayor que 1. Esto no puede pasar, por lo que añadimos una función que tuviera en cuenta el ruido térmico de la cámara calculado en la sección 3.5.1.

Siguiendo este proceso, obtenemos las curvas de reflectancia que se muestran en la Figura 11.

Podemos observar que las curvas se ajustan bastante a las que hemos medido con el colorímetro, salvo para el amarillo y las zonas de longitudes de onda cortas. Esto puede deberse a que las curvas son las de un solo píxel en la muestra, y promediando probablemente mejoraría la curva, y a que los píxeles tienen cierto error que se propaga y afecta al error en la curva de reflectancia.

También podemos ver que las muestras cuya reflectancia es grande en el infrarrojo cercano tienen un valor más grande en la zona del azul de lo que deberían. Esto se debe a que el filtro deja pasar una cantidad de luz infrarroja comparable a la luz azul en esa zona, y eso afecta a las medidas que tomamos. Para quitar esta luz se necesitaría un filtro de luz IR del rango adecuado, pero no contábamos con él, así que queda como trabajo futuro en caso de querer mejorar el dispositivo.

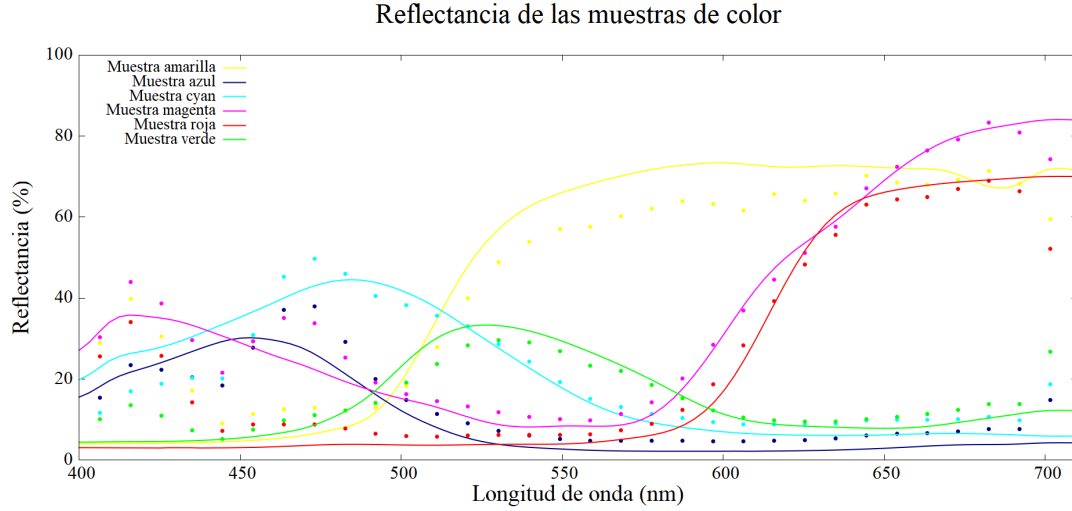


Figura 11: *Curvas de reflectancia de las muestras de color. La reflectancia medida con el colorímetro está representada con la línea sólida, y la curva de reflectancia calculada con la cámara se presenta con puntos*

3.6.3. Obtención de las coordenadas de color

Una vez obtenida la curva de reflectancia de cada píxel, pasamos a calcular las coordenadas de color de cada uno de ellos, siguiendo el procedimiento descrito en la sección 2.2. Para esto hemos usado dos iluminantes para compararlos, el D65 y el LED, el 10º observador de color, y calculado las coordenadas RGB en el sistema sRGB, ya que este es el más generalizado en las pantallas actuales.

Lo primero que tenemos que hacer es calcular las coordenadas de los puntos triestímulo de nuestros iluminantes, es decir, las coordenadas del blanco para el D65 y para el LED. Para ello se utilizan las ecuaciones 2, 3 y 4 y tomando la reflectancia igual a 1. Las coordenadas del blanco para cada iluminante, normalizadas en Y, se presentan en la Tabla 2.

	X_B	Y_B	Z_B
D65	0.9482	1	1.0738
LED	0.9175	1	0.6908

Tabla 2: *Coordenadas XYZ del blanco para cada iluminante*

Aplicamos estas mismas ecuaciones a las curvas de reflectancia que tenemos calculadas, de manera que obtenemos una hipermatriz formada por tres matrices (cada una con una de las coordenadas, XYZ) de la misma dimensión que las imágenes originales.

Para calcular las coordenadas RGB a partir de las coordenadas XYZ, aplicamos el algoritmo de adaptación cromática para poder usar las matrices ya definidas de conversión a RGB, ya que se calculan para un blanco determinado. En el caso de las matrices que hemos usado, necesitamos pasar del blanco del LED al del D65. Una vez tenemos esto, aplicamos la matriz de conversión de XYZ a sRGB, y obtenemos las

coordenadas RGB lineales. Las pantallas trabajan con una codificación no lineal, por lo que aplicamos la corrección “gamma”.

Por último, implementamos una función para representar la imagen final en pantalla y que nos permita seleccionar un píxel para representar la reflectancia en el mismo.

Como hemos explicado, para el cálculo de las coordenadas de color es muy importante la elección del iluminante, ya que éste nos modifica el resultado final, debido a la diferencia en los espectros. En la Figura 12 podemos ver la curva espectral de cada uno de los iluminantes utilizados.

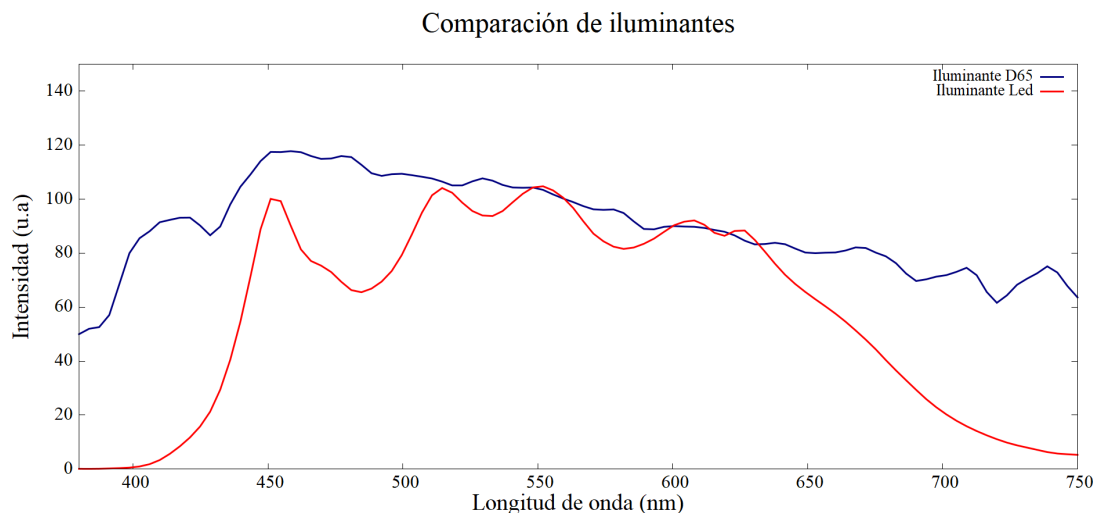


Figura 12: Comparación de los espectros de los dos iluminantes utilizados para el cálculo de las coordenadas de color

4. Resultados

En esta sección se presentan las imágenes finales que se obtuvieron con nuestra cámara, tanto para las muestras de color como para el cuadro que medimos como paso final del proyecto. Con este cuadro, pudimos ver el resultado de la cámara cuando se miden varios colores en la misma muestra.

4.1. Muestras de color

Primero calculamos las coordenadas de color con las curvas de reflectancia que medimos con el colorímetro para poder usarlas de referencia, ya que la manera en la que vemos los colores en pantalla depende de la configuración de la propia pantalla, y sería difícil saber si el resultado obtenido con la cámara es correcto si no pudiéramos comparar con el color obtenido con la curva de reflectancia medida por el colorímetro.

Los resultados obtenidos para las curvas de reflectancia del colorímetro con el iluminante D65 se muestran en la Figura 13, y con el iluminante LED se obtienen los colores de la Figura 14.

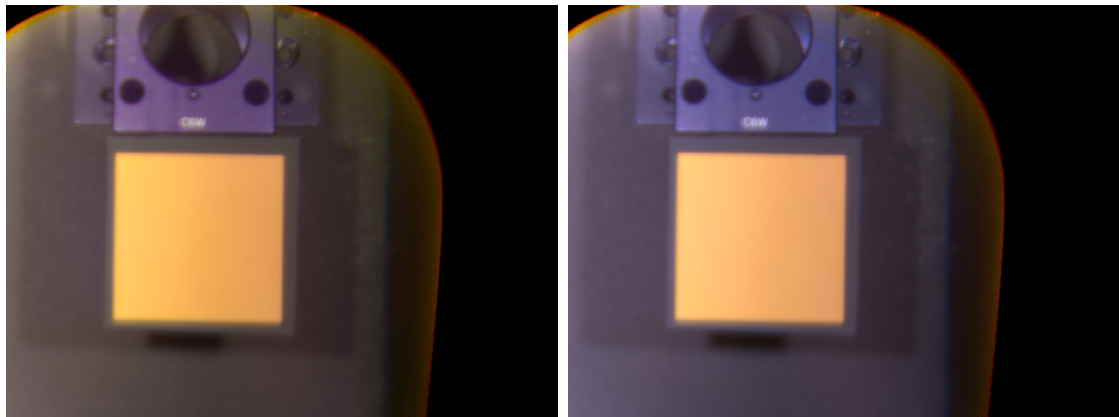


Figura 13: *Illuminante D65*



Figura 14: *Illuminante LED*

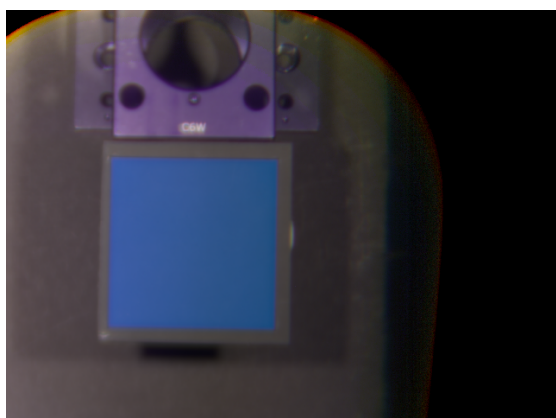
Podemos ver que los colores obtenidos con ambos iluminantes son muy parecidos, pero no son iguales. Esta diferencia se hace más apreciable al observar las imágenes obtenidas con las fotografías de la cámara, especialmente en el fondo negro. Estas imágenes se muestran en las figuras 15, 16, 17, 18, 19 y 20.



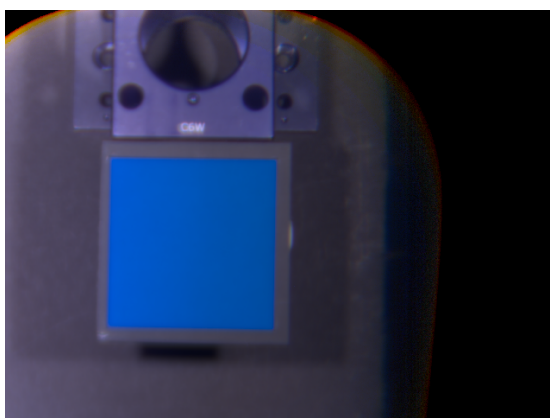
(a) *D65*

(b) *LED*

Figura 15: *Muestra amarilla*

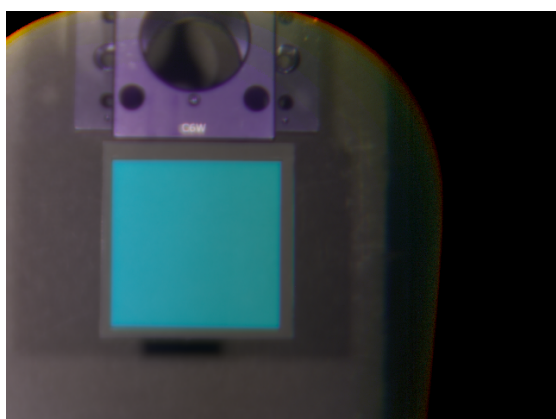


(a) *D65*

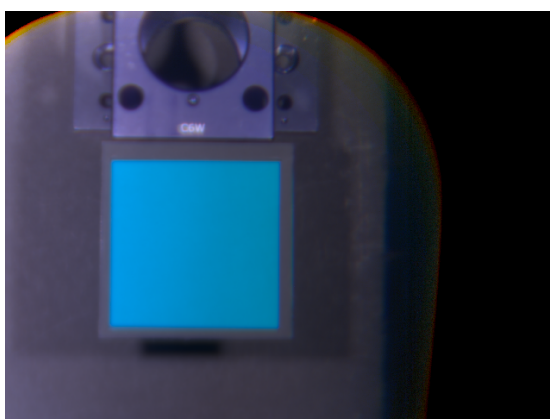


(b) *LED*

Figura 16: *Muestra azul*

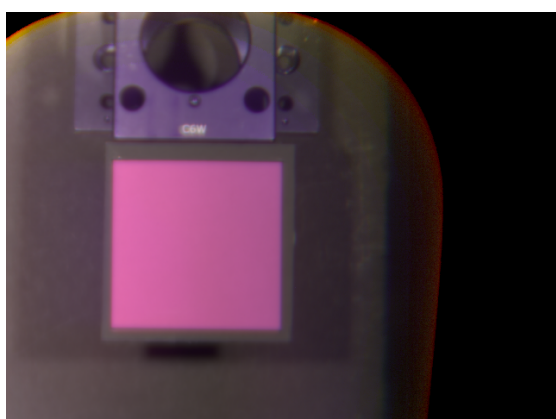


(a) *D65*

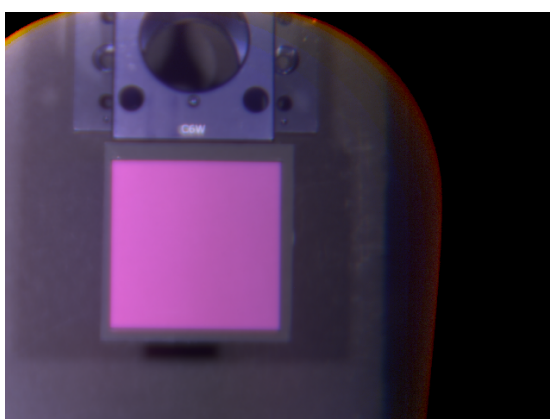


(b) *LED*

Figura 17: *Muestra cyan*

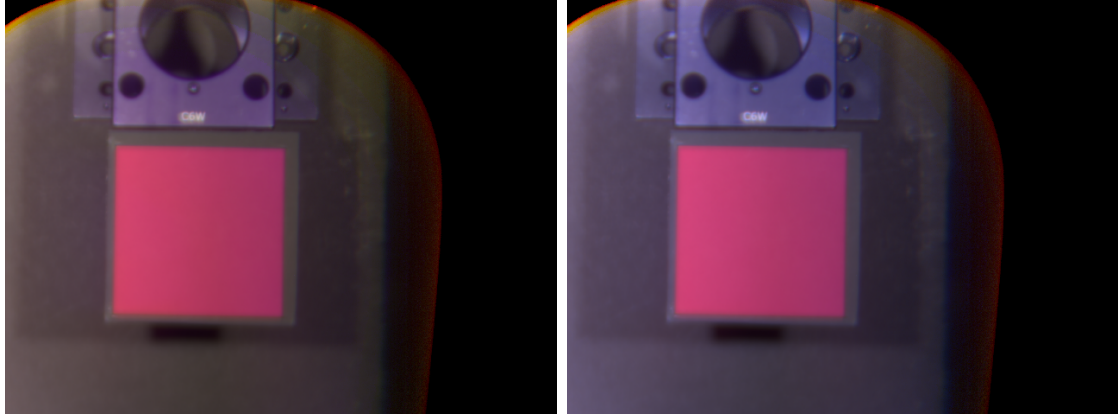


(a) *D65*



(b) *LED*

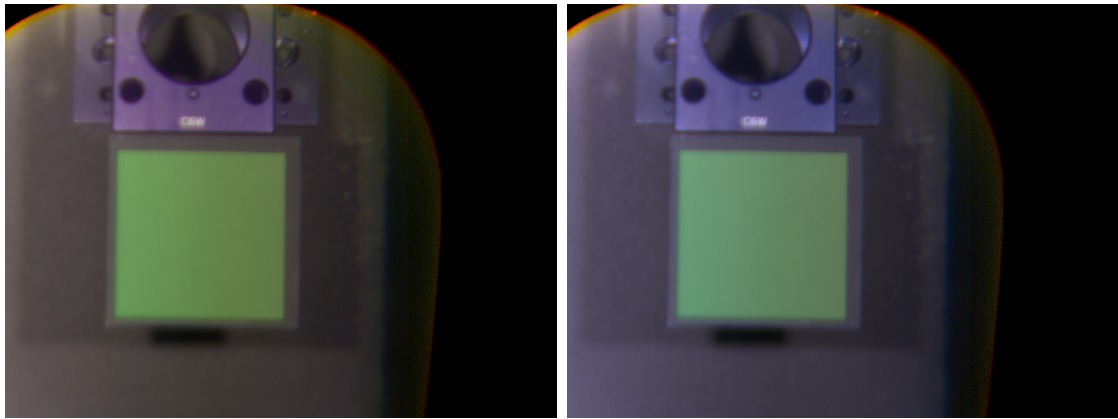
Figura 18: *Muestra magenta*



(a) *D65*

(b) *LED*

Figura 19: *Muestra roja*



(a) *D65*

(b) *LED*

Figura 20: *Muestra verde*

Los colores obtenidos con las curvas del colorímetro son los correspondientes a una zona promediada de la muestra, mientras que los colores obtenidos con nuestra cámara son los colores de cada punto de la muestra. La distribución de luz no es uniforme al iluminar las muestras de color, por lo que no se obtiene un color completamente uniforme. Podemos ver que los colores de las muestras son bastante similares a los obtenidos con el colorímetro, aunque no iguales. Esto se debe a que las curvas de reflectancia obtenidas con la cámara no son exactamente iguales a las del colorímetro, como se observaba en la Figura 11.

4.2. Cuadro

Por último hemos tomado las imágenes de un cuadro, hemos calculado sus coordenadas de color y lo hemos mostrado por pantalla. En la Figura 21 se puede ver una imagen del cuadro tomada con una cámara RGB, específicamente una cámara de móvil. En esta imagen, la obra está iluminada con luz natural. El resultado obtenido con la

cámara hiperespectral se puede ver en la Figura 22, para el que se han usado también los dos iluminantes, el D65 y el LED.

Podemos ver que las imágenes de la cámara hiperespectral son más oscuras, lo que se debe a la diferencia en la iluminación y al procesado de la imagen que realiza el móvil de manera automática, pero se siguen pudiendo distinguir la imagen y los distintos colores.

Además, visualmente el cuadro se asemeja más a las imágenes obtenidas con la cámara hiperespectral.

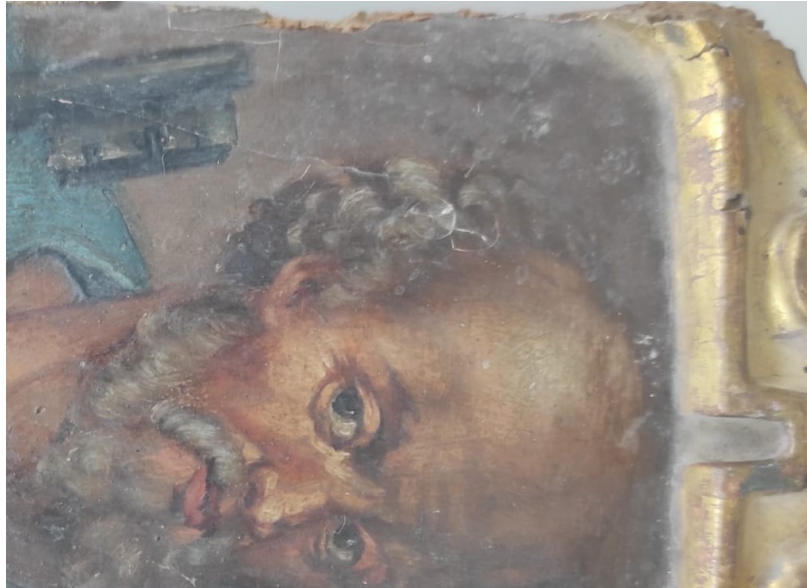
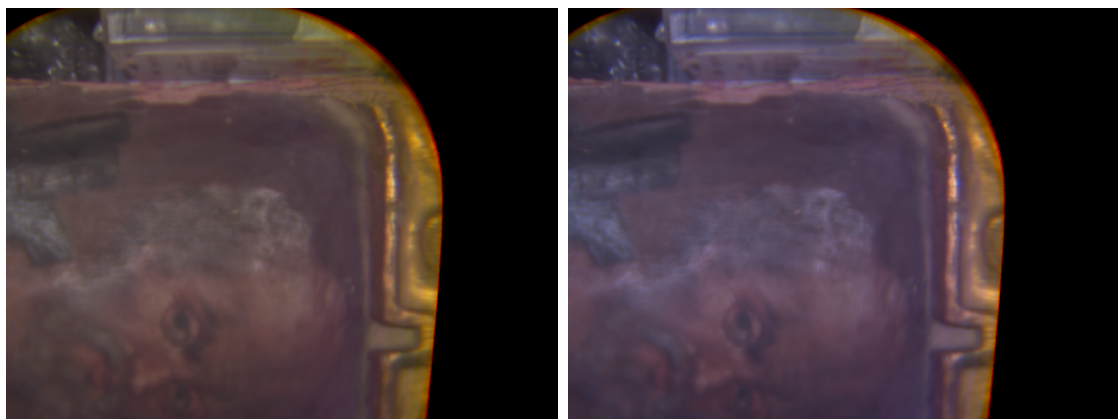


Figura 21: *Fotografía del cuadro tomada con una cámara RGB*



(a) *D65*

(b) *LED*

Figura 22: *Imágenes del cuadro obtenidas con la cámara hiperespectral*

En la Figura 23 podemos ver un ejemplo de cómo representa nuestro programa la reflectancia del píxel que escogemos. En el cuadro predominan los colores oscuros, por lo que la reflectancia no es muy grande, pero hay zonas del marco, como la que se muestra, que sí reflejan más. La imagen que se muestra es la del iluminante D65, pero

la reflectancia en ese píxel no depende del iluminante, usando el LED el resultado sería el mismo.

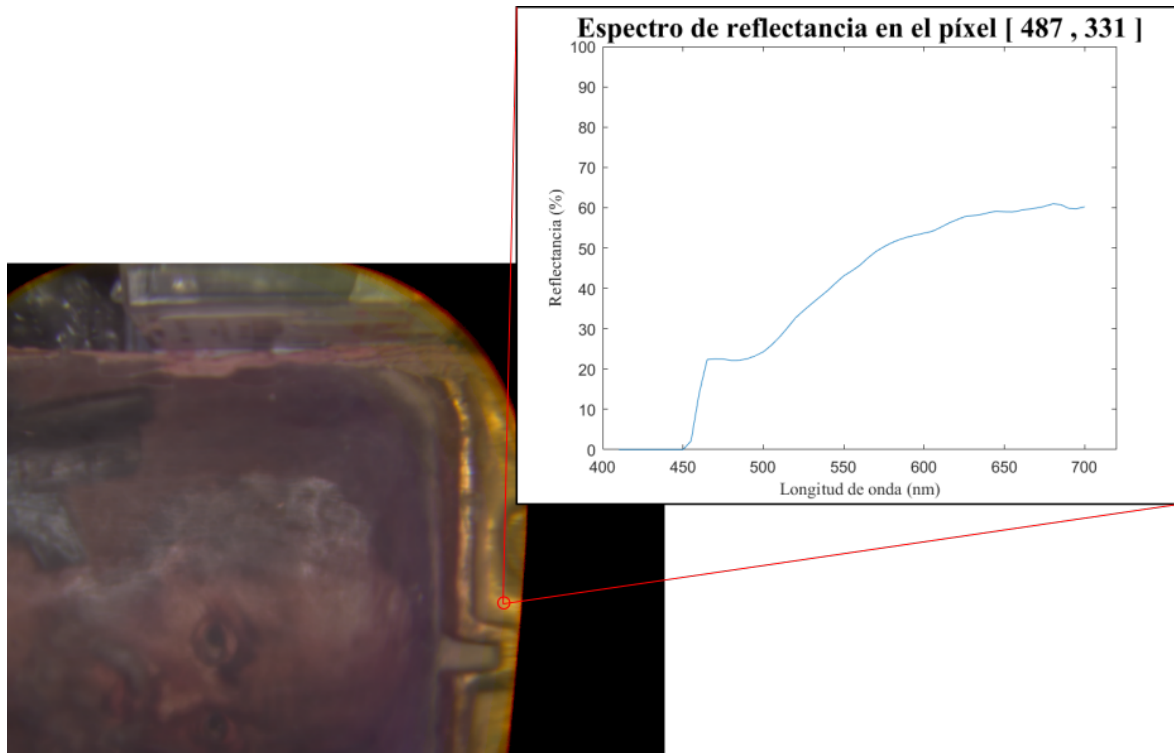


Figura 23: *Ejemplo del resultado obtenido con el programa para la reflectancia en un píxel*

5. Conclusiones y trabajo futuro

Hemos conseguido construir una cámara hiperespectral completamente funcional usando componentes discretos. Hemos calibrado individualmente los distintos componentes del montaje, que son las fuentes de luz, el filtro y la cámara.

Las curvas de reflectancia que hemos obtenido con la cámara son muy similares a las obtenidas con un colorímetro comercial, y los colores que hemos calculado con dichas curvas son muy parecidos a los que obtenemos con las curvas medidas con el colorímetro.

Esta cámara podrá usarse para el estudio de obras de arte y permitirá caracterizar distintos barnices que se usaban en éstas, permitiendo la creación de una base de datos que contenga las características espectrales de los barnices más utilizados en cuadros y obras pictóricas. También se puede usar para obtener el color que tendrían de dichas obras si no tuvieran el barniz. Esto se haría midiendo el espectro del cuadro, y restando la curva espectral del barniz, que o bien habría que medir o se podría obtener de una base de datos.

Es importante destacar que los colores representados pueden variar de una pantalla a otra, ya que los espacios de color y las ganancias no son iguales en todos los monitores, y en muchos casos no se pueden ajustar. Sin embargo, se puede obtener la curva de

reflectancia en cada posición de la imagen, y es esta curva la que determina el color que se está observando.

En un futuro se podría mejorar este dispositivo utilizando fuentes de luz con un mayor espectro en el azul, e incorporando un filtro de infrarrojo en el rango adecuado para eliminar la contribución de esta parte del espectro en la zona cercana al ultravioleta. También sería un buen paso reemplazar el filtro usado por uno que no introduzca reflejos en la zona de 400 nm, es decir, que no tenga una rendija en el centro del filtro.

Además, se podría minimizar el error realizando una calibración más profunda del motor y la transmitancia del filtro, ya que hay un pequeño error en la posición del motor que podría afectar a la posición del pico de la banda de transmitancia del filtro. Sin embargo, hemos visto que este error no es significativo y lo hemos despreciado.

Bibliografía

- [1] Costanza Cucci, John K. Delaney y Marcello Picollo. “Reflectance Hyperspectral Imaging for Investigation of Works of Art: Old Master Paintings and Illuminated Manuscripts”. En: *Accounts of Chemical Research* 49 (10 oct. de 2016), págs. 2070-2079. ISSN: 15204898. DOI: 10.1021/acs.accounts.6b00048.
- [2] Haida Liang. “Advances in multispectral and hyperspectral imaging for archaeology and art conservation”. En: *Applied Physics A: Materials Science and Processing* 106 (2 feb. de 2012), págs. 309-323. ISSN: 09478396. DOI: 10.1007/s00339-011-6689-1.
- [3] Jakub Sandak y col. “Nondestructive evaluation of heritage object coatings with four hyperspectral imaging systems”. En: *Coatings* 11 (2 feb. de 2021), págs. 1-15. ISSN: 20796412. DOI: 10.3390/coatings11020244.
- [4] Adam Polak y col. “Hyperspectral imaging combined with data classification techniques as an aid for artwork authentication”. En: *Journal of Cultural Heritage* 26 (jul. de 2017), págs. 1-11. ISSN: 12962074. DOI: 10.1016/j.culher.2017.01.013.
- [5] Bruce Justin Lindbloom. *RGB/XYZ Matrices*. URL: http://www.brucelindbloom.com/index.html?Eqn_RGB_XYZ_Matrix.html.
- [6] Bruce Justin Lindbloom. *Chromatic Adaptation*. URL: http://www.brucelindbloom.com/index.html?Eqn_ChromAdapt.html.
- [7] Bruce Justin Lindbloom. *Gamma correction*. URL: http://www.brucelindbloom.com/index.html?Eqn_XYZ_to_RGB.html.

I. Programa de control de la cámara y el motor

```
%% Iniciar workspace
clear
clc

%% Parametros de barrido
N = 36;% Numero de pasos
NFrames = 5;% Numero de frames que captura en cada posicion
pasomm = 1;% Tamaño del paso en mm
carpetaResultados = './FotosCuadro/';% Carpeta donde guardar las imagenes

if (N*pasomm >60)
    warning('¡Los parametros de barrido exceden el recorrido del motor!');
end

%% Iniciar motor
motor = TLMotor(13);% Se crea el motor
motor.moveToPosition(18,1);% Se lleva a la posicion inicial
motor.setJogStepSize(pasomm);% Se ajusta el paso

%% Bucle de medida
for n=1:N
    [~,posicionString] = motor.getCurrentPosition();
    for x=1:NFrames
        if(x<10)
            if(n<10)
                nombreImagen = [carpetaResultados 'n0' num2str(n) '_' ...
                    strrep(posicionString, '.', '_') 'fr0' num2str(x) '_'];
            else
                nombreImagen = [carpetaResultados 'n' num2str(n) '_' ...
                    strrep(posicionString, '.', '_') 'fr0' num2str(x) '_'];
            end
        else
            if(n<10)
                nombreImagen = [carpetaResultados 'n0' num2str(n) '_' ...
                    strrep(posicionString, '.', '_') 'fr' num2str(x) '_'];
            else
                nombreImagen = [carpetaResultados 'n' num2str(n) '_' ...
                    strrep(posicionString, '.', '_') 'fr' num2str(x) '_'];
            end
        end
    end
end
```

```

    end
    system('./Camara/USBCam.exe');
    pause(8)
    movefile('0.bmp',[nombreImagen,'exp0.bmp']);
    movefile('1.bmp',[nombreImagen,'exp1.bmp']);
    movefile('2.bmp',[nombreImagen,'exp2.bmp']);
    movefile('3.bmp',[nombreImagen,'exp3.bmp']);
    movefile('4.bmp',[nombreImagen,'exp4.bmp']);
end
display([posicionString(1:end-2) '(mm)']);
if (n<N)
    motor.moveForward();
    pause(2)
end
end

```

II. Programa de procesamiento de las imágenes, para el cálculo de la reflectancia y las coordenadas de color

```
%% Inicializamos workspace
clear
clc

%% Establecemos las carpetas de lectura de las imagenes
listE = dir('./ Blanco/');
FolderE = './Blanco/ ';
NombreOut = 'Cuadro_D65.png';
listC = dir('./FotosCuadro/');
FolderC = './FotosCuadro/';
out = fopen('Cuadro.txt','w');

%% Definimos las dimensiones de las matrices de datos
M4D = imread(listE(3).name);
dim = size(M4D);
NRow = dim(1,1);
NCol = dim(1,2);

%% Definimos constantes del programa
NMedias = 5;
NFr = 5;
NPos = uint16(length(listE)/(5*NFr));
NPos = 32;
TExpB = [1,3,5,10,50];
TExpC = [1,4,20,50,100];
amp = 6.68783;
fact = 9.24943;
base = 1.41903;

%% Definimos ficheros utilizados
ReflectanciaBlanco = fopen('RefBlanco.txt','r');
Datos = fopen('datosParaXYZ1964.txt','r');
Iluminante = fopen('iluminante.txt','r');
D65 = fopen('iluminante.txt','r');
Matriz = fopen('xyzArgb.txt','r');
B_Mat = fopen('Bradford.txt','r');
```

```

B_MatInv = fopen('Bradford_Inv.txt','r');

%% Definimos las matrices que utilizamos, salvo las que alguna de sus
dimensiones depende de parámetros calculados más tarde
EMatrix = zeros(NRow,NCol);
CMatrix = zeros(NRow,NCol);
lambda=zeros(NPos,1);
D = zeros(4,81);
I = zeros(2,81);
ID65 = zeros(2,81);
lambdaT = zeros(1,81);
ErrorE = zeros(NRow,NCol);
ErrorC = zeros(NRow,NCol);
M = zeros(3,3);
Brad = zeros(3,3);
BradInv = zeros(3,3);
XYZ = zeros(NRow,NCol,3);
XYZB = zeros(3,1);
XYZB_D65 = zeros(3,1);
rgb = zeros(NRow,NCol,3);
rgb2 = zeros(NRow,NCol,3);
Blanco = zeros(2,39);

%% Cálculo de longitud de onda según la posición
for i=1:32% NPos
    lambda(i,1) = 872.807 - 9.51668 * (17+i);
end

%% Leemos ficheros
Blanco2 = fscanf(ReflectanciaBlanco,' %f %f \ n',[2,39]);
D2 = fscanf(Datos,' %f %f %f %f \ n',[4,81]);
I2 = fscanf(Iluminante,' %f %f \ n',[2,81]);
M1 = fscanf(Matriz,' %f %f %f \ n',[3,3]);
Brad2 = fscanf(B_Mat,' %f %f %f \ n',[3,3]);
BradInv2 = fscanf(B_MatInv,' %f %f %f \ n',[3,3]);
MD65 = fscanf(D65,' %f %f \ n',[2,81]);
for i=1:3
    for j=1:3
        M(i,j) = M1(j,i);
    end
end
end

```

```

for i=1:3
    for j=1:3
        Brad(i,j) = Brad2(j,i);
    end
end
for i=1:3
    for j=1:3
        BradInv(i,j) = BradInv2(j,i);
    end
end
for i = 1:39
    Blanco(:,i) = Blanco2(:,40-i);
end

%% Comparamos Nlambda de blanco y calculada para ver cuantas Pos usamos
y terminamos de definir matrices
dim = size(Blanco);
NP = dim(1,2);
if(NP<NPos)
    NPos = NP;
end
Reflectancia = zeros(NRow,NCol,NPos);
RefBlanco=zeros(39,2);
p250250 =zeros(2,NPos);

%% Extrapolamos los valores de la reflectancia del blanco, en caso de
que no este en las mismas lambdas que nuestros datos
RefBlanco(:,1) = Blanco(1,:);
for j = 1:NPos
    Imin = 2;
    while (lambda(j)<Blanco(1,Imin) && lambda(j) =Blanco(1,Imin))
        Imin = Imin+1;
    end
    Imax = Imin-1;
    if(lambda(j)==Blanco(1,Imin))
        RefBlanco(j,2) = Blanco(2,Imin);
    else
        RefBlanco(j,2) = ((Blanco(2,Imax)-Blanco(2,Imin))/(Blanco(1,Imax)
        -Blanco(1,Imin)))*(lambda(j)-Blanco(1,Imin))+Blanco(2,Imin);
    end
    RefBlanco(j,2) = RefBlanco(j,2)/100;
end

```

```

end

%% Calculo la curva de reflectancia
for p = 1:NPos
    Et = zeros(NRow,NCol,5);
    Ct = zeros(NRow,NCol,5);
    % Calculo los promedios
    for ExpE = 0:4
        for fr = 1:5
            ind=ExpE+3+NFr*5*(p-1)+5*(fr-1);
            nombreE = [FolderE listE(ind).name];
            Mat = imread(nombreE);
            Mat = im2double(Mat);
            Mat = im2gray(Mat);
            Et(:,:,ExpE+1) = Et(:,:,ExpE+1) + Mat;
        end
        Et(:,:,ExpE+1) = Et(:,:,ExpE+1)/5;
    end
    for ExpC = 0:4
        for fr = 1:5
            ind=ExpC+3+NFr*5*(p-1)+5*(fr-1);
            nombreC = [FolderC listC(ind).name];
            Mat = imread(nombreC);
            Mat = im2double(Mat);
            Mat = im2gray(Mat);
            Ct(:,:,ExpC+1) = Ct(:,:,ExpC+1) + Mat;
        end
        Ct(:,:,ExpC+1) = Ct(:,:,ExpC+1)/5;
    end
end

% Miro si cada píxel está saturado, y lo cogemos si no
for r = 1:NRow
    for c = 1:NCol
        ExpE = 5;
        while (Et(r,c,ExpE)>0.95)
            ExpE = ExpE-1;
        end
        EMatrix(r,c) = Et(r,c,ExpE)/(TExpB(ExpE));
        ErrorE(r,c) = amp*exp(-EMatrix(r,c)*fact)+base;
        ErrorE(r,c) = ErrorE(r,c)/100;
    end
end

```

```

        ExpC = 5;
        while (Ct(r,c,ExpC)>0.95)
            ExpC = ExpC-1;
        end
        CMatrix(r,c) = Ct(r,c,ExpC)/TExpC(ExpC);
        ErrorC(r,c) = amp*exp(-CMatrix(r,c)*fact)+base;
        ErrorC(r,c) = ErrorC(r,c)/100;
    end
end

Reflectancia(:, :, p) = CMatrix*RefBlanco(p,2)./EMatrix;
for r=1:NRow
    for c=1:NCol
        if (EMatrix(r,c) <0.003)
            Reflectancia(r,c,p) = 0;
        end
        if (Reflectancia(r,c,p) >= 1)
            if (EMatrix(r,c) <0.004 || CMatrix(r,c) <0.004)
                Reflectancia(r,c,p) = 0;
            else
                Reflectancia(r,c,p) = (CMatrix(r,c)-ErrorC(r,c))*
                    *RefBlanco(p,2)./(EMatrix(r,c)+ErrorE(r,c));
                CMatrix(r,c) = CMatrix(r,c)-ErrorC(r,c);
                EMatrix(r,c) = EMatrix(r,c)+ErrorE(r,c);
            end
        end
    end
end
end
p250250(1,p) = lambda(p);
p250250(2,p) = Reflectancia(250,250,p);
end
p250250(2,:) = p250250(2,:)*100;
fprintf(out, ' %f %f\n', p250250);

%% Calculamos las coordenadas de color
lambdaT2 = D2(1,:);
% Reordenamos
if (lambdaT2(2)>lambdaT2(1))
    for i = 1:80
        lambdaT(i) = lambdaT2(81-i);
        D(:,i) = D2(:,81-i);
    end
end

```



```

        I(:,i) = I2(:,81-i);
        ID65(:,i) = MD65(:,81-i);
    end
end
IndMinD = 81;
IndMaxD = 1;
mini = 0;
maxi = 0;
for i = 1:81
    if(D(1,i)<=lambda(1)  mini == 0)
        IndMinD = i;
        mini = 1;
    end
    if(D(1,82-i)>=lambda(NPos)  maxi == 0)
        IndMaxD = 82-i+1;
        maxi = 1;
    end
end
end
global DeltaD;
DeltaD = IndMaxD -IndMinD;
%Definimos matrices D e I con los elementos que vamos a utilizar
DatosAcotados = zeros (4,DeltaD);
IAcotado = zeros (2,DeltaD);
ID65Acotado = zeros (2,DeltaD);
global RefCalculada;
RefCalculada = zeros(NRow,NCol,DeltaD);
global lambdaCalc;
lambdaCalc = zeros(DeltaD,1);
Blanco2 = zeros (2,DeltaD);
for i = 1:DeltaD
    DatosAcotados(:,i) = D(:,IndMinD+i-1);
    IAcotado(:,i) = I(:,IndMinD+i-1);
    lambdaCalc(i) = DatosAcotados(1,i);
    Blanco2(1,i) = DatosAcotados(1,i);
    ID65Acotado(:,i) = ID65(:,IndMinD+i-1);
end
DeltaL = lambdaCalc(1)-lambdaCalc(2);
% Extrapolamos del blanco
for j = 1:(DeltaD)
    Imin = 2;
    while (Blanco2(1,j)<Blanco(1,Imin)  Blanco2(1,j) =Blanco(1,Imin))

```

```

        Imin = Imin+1;
    end
    Imax = Imin-1;
    Blanco2(2,j) = ((Blanco(2,Imax)-Blanco(2,Imin))/(Blanco(1,Imax)-
        Blanco(1,Imin)))*(Blanco2(1,j)-Blanco(1,Imin))+Blanco(2,Imin);
end

% Calculamos para cada pixel
for j = 1:(DeltaD)
    Imin = 2;
    while (lambda(Imin)>lambdaCalc(j) && lambda(Imin) =lambdaCalc(j))
        Imin = Imin+1;
    end
    Imax = Imin-1;
    RefCalculada(:,j) = ((Reflectancia(:,Imax)-Reflectancia(:,Imin))/
        /(lambda(Imax)-lambda(Imin)))*(lambdaCalc(j)-lambda(Imin))+Reflectancia(:,Imin)
end

RxB = 0;
RyB = 0;
RzB = 0;
for i = 1:DeltaD
    RxB = RxB + ID65Acotado(2,i)*DatosAcotados(2,i);
    RyB = RyB + ID65Acotado(2,i)*DatosAcotados(3,i);
    RzB = RzB + ID65Acotado(2,i)*DatosAcotados(4,i);
end
XYZB_D65(1,1) = RxB/RyB;
XYZB_D65(2,1) = RyB/RyB;
XYZB_D65(3,1) = RzB/RyB;

RxB = 0;
RyB = 0;
RzB = 0;
Rx = zeros(NRow,NCol);
Ry = zeros(NRow,NCol);
Rz = zeros(NRow,NCol);
DenY = zeros(NRow,NCol);
for i = 1:DeltaD
    RxB = RxB + IAcotado(2,i)*DatosAcotados(2,i);
    RyB = RyB + IAcotado(2,i)*DatosAcotados(3,i);
    RzB = RzB + IAcotado(2,i)*DatosAcotados(4,i);

```

```

end
XYZB(1,1) = RxB/RyB;
XYZB(2,1) = RyB/RyB;
XYZB(3,1) = RzB/RyB;
for i = 1:DeltaD
    Rx(:, :) = Rx(:, :) + RefCalculada(:, :, i)*IAcotado(2,i)*DatosAcotados(2,i);
    Ry(:, :) = Ry(:, :) + RefCalculada(:, :, i)*IAcotado(2,i)*DatosAcotados(3,i);
    Rz(:, :) = Rz(:, :) + RefCalculada(:, :, i)*IAcotado(2,i)*DatosAcotados(4,i);
    DenY(:, :) = DenY(:, :) + IAcotado(2,i)*DatosAcotados(3,i);
end
XYZ(:, :, 1) = Rx(:, :)./DenY(:, :);
XYZ(:, :, 2) = Ry(:, :)./DenY(:, :);
XYZ(:, :, 3) = Rz(:, :)./DenY(:, :);

    % Adaptación cromática
CRD_B = Brad*XYZB;
CRD_D65 = Brad*XYZB_D65;
A = zeros(3,3);
for i = 1:3
    A(i,i) = CRD_D65(i,1)/CRD_B(i,1);
end
MConv = BradInv*A*Brad;

    % Aplicamos conversion a XYZ
XYZ(:, :, 1) = XYZ(:, :, 1)/XYZB(1);
XYZ(:, :, 2) = XYZ(:, :, 2)/XYZB(2);
XYZ(:, :, 3) = XYZ(:, :, 3)/XYZB(3);
vec = zeros(3,1);
for r = 1:NRow
    for c = 1:NCol
        vec(1) = XYZ(r,c,1);
        vec(2) = XYZ(r,c,2);
        vec(3) = XYZ(r,c,3);
        XYZ(r,c,:) = MConv*vec;
    end
end

    % Conversión a rgb
for r = 1:NRow
    for c = 1:NCol
        vec(1) = XYZ(r,c,1);

```

```

        vec(2) = XYZ(r,c,2);
        vec(3) = XYZ(r,c,3);
        rgb2(r,c,:) = M*vec;
    end
end

gamma = 2.4;
for r = 1:NRow
    for c = 1:NCol
        for x = 1:3
            if(rgb2(r,c,x) <= 0.0031308)
                rgb(r,c,x) = 12.92*rgb2(r,c,x);
            else
                rgb(r,c,x) = 1.055*(rgb2(r,c,x)^(1/gamma))-0.055;
            end
        end
    end
end

f = figure(1);
image(rgb);
imwrite(rgb,NombreOut);
while(f == 1)
    figure(1);
    [x,y] = ginput(1);
    x = uint16(x);
    y = uint16(y);
    plotReflectancia(x,y);
    pause
end
close(f);

%% Definimos la función para dibujar la reflectancia
function [] = plotReflectancia(coordX,coordY)

global RefCalculada
global lambdaCalc
global DeltaD
y = zeros(DeltaD,1);
for i = 1:DeltaD
    y(i) = RefCalculada(coordY,coordX,i)*100;
end

```

```

end

figure;
plot(lambdaCalc,y);
axis([400 720 0 100]);
Titulo = ['Espectro de reflectancia en el píxel [ ' num2str(coordX) ' , '
num2str(coordY) ' ]'];
ylabel('Reflectancia (%)','FontName','Times');
xlabel('Longitud de onda (nm)','FontName','Times');
title (Titulo,'FontName','Times','FontSize',18);
end

```