

Modelización matemática de la percepción cromática del ojo humano

Alejandro Campo Herbés

Enero 2022

CPR Plurilingüe San Fermín, 3º ESO
Coordinado por: Belén Álvarez Fernández

Índice

1. Introducción	2
2. Antecedentes	4
3. Objetivos	5
4. Metodología	5
4.1. Los espacios de excitación y color	5
4.2. El triángulo de Maxwell	6
4.3. Cambio de bases	8
4.4. De LMS a RGB	10
5. Resultados	11
6. Conclusiones	11
7. Agradecimientos	12
8. Anexos	13
8.1. Anexo 1	13
8.2. Anexo 2	14
8.3. Anexo 3	15
8.4. Anexo 4	16
8.5. Anexo 5	17
8.6. Anexo 6	18
8.7. Anexo 7	19

Abstract

The human eye perceives color using its photoreceptors, named cone cells. There are three types of cones, each one being sensitive to different wavelengths. By defining a crude approximation of these sensitivities via gaussian functions, I formulate a transformation that converts the three stimuli of the eye to a specific wavelength into a color defined in the RGB (Red, Green and Blue; light primary colors) space. This kind of function would prove useful in optical simulations and alike, of which I give an example.

Resumen

El ojo humano percibe el color usando un conjunto de fotorreceptores llamados conos. Hay tres tipos de conos, cada uno sensible a distintas longitudes de onda. Definiendo una aproximación de estas sensibilidades mediante funciones gaussianas, formulo una transformación que convierte los tres estímulos del ojo a una longitud de onda específica en un color definido dentro del espacio RGB (Rojo, Verde y Azul; los colores primarios). Este tipo de función tiene utilidades en simulaciones de carácter óptico, de las cuales aporte un ejemplo.

1. Introducción

En el curso 2020-2021, como respuesta a la duda de una compañera (“¿De qué color son los átomos?”[Cid21]), desarrollé una simulación virtual de los espectros de emisión de cada uno de los elementos químicos. Durante el proceso de diseño, me percaté de que precisaba de una forma de convertir la longitud de onda de la luz reemitida a un color que pudiese mostrar en pantalla (al fin y al cabo, esa era la premisa final de la simulación, mostrar el “color” del átomo). Para ello estudié cómo nuestros ojos perciben los colores, usando las células foto-

receptoras que se encuentran en su interior. Hay dos tipos, los conos y los bastones; debido a que los segundos se usan casi únicamente en la visión nocturna, mis sujetos de estudio fueron únicamente los conos. A su vez, estos conos se clasifican en tres grupos: los conos L (Long), conos M (Medium) y conos S (Short). La diferencia entre ellos es que sus picos de respuesta se encuentran en diferentes rangos del espectro luminoso (Fig. 1). Para los L, la respuesta máxima está próxima a los 570 nm (de ahí su nombre, la onda es “larga”). Para los M, este valor rondaría los 540 nm. Para los S, 470 nm.

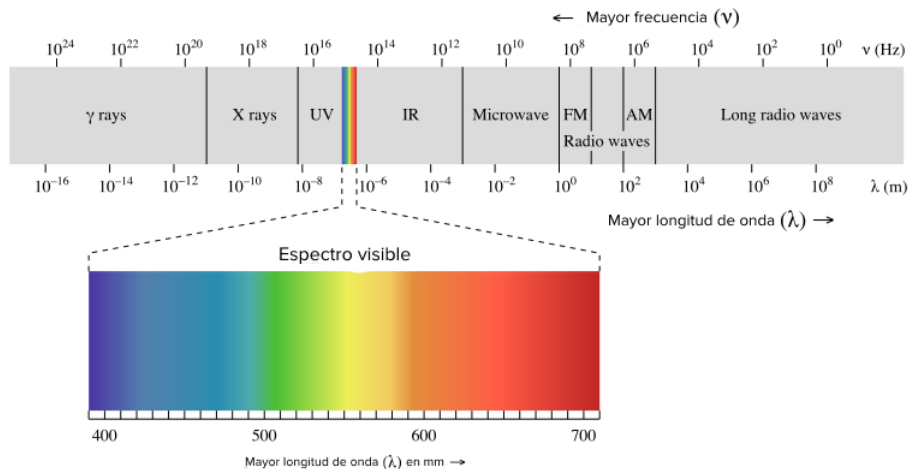


Figura 1: El espectro electromagnético. Imagen tomada de la ChemWiki de UC Davis (Universidad de California en Davis), CC-BY-NC-SA 3.0

Estas respuestas se pueden mostrar gráficamente (Fig. 2), asemejándose a una variación estándar. Gracias a esta característica, puede aproximar las respuestas como funciones gaussianas:

$$L(\lambda) = e^{-\frac{(\lambda-570)^2}{2 \cdot 50^2}} \quad (1)$$

$$M(\lambda) = e^{-\frac{(\lambda-540)^2}{2 \cdot 39^2}} \quad (2)$$

$$S(\lambda) = e^{-\frac{(\lambda-467)^2}{2 \cdot 24,4^2}} \quad (3)$$

Después, le atribuí a cada cono uno de los tres colores primarios (rojo, verde y azul), siendo L = Rojo, M = Verde y S = Azul. Así, por ejemplo, el valor máximo de la respuesta de L equivaldría a un color con la mayor cantidad de rojo posible (255, según la convención informática del RGB).

Lamentablemente (o afortunadamente), las cosas no son tan simples. Es bastante trivial ver que este modelo, además de no tener ninguna prueba empírica de su funcionamiento, se va a romper en múltiples partes del espectro. Esto ocurre en mayor grado en los extremos de la luz visible, el rojo y el violeta: En el rojo, debido a que la función gaussiana del cono L disminuye demasiado rápido como para representar el amplio rango que pertenece al rojo puro (aprox. 650 nm - 720 nm). En el violeta, porque la respuesta del cono L es 0 en ese punto (380 nm - 420 nm), y como sabemos, el violeta es una mezcla de rojo y azul.

Lo que hice para solucionar en parte este problema fue darle unas normas distintas a los rangos anteriormente comentados. De esta forma, en el rango rojo las funciones serían unas constantes equivalentes al rojo puro. En el violeta simplemente fijé el valor del azul y el rojo, haciendo que este último ya no fuese nulo. Esto se puede expresar como unas funciones por partes:

$$L(\lambda) = \begin{cases} e^{-\frac{(\lambda-570)^2}{2 \cdot 50^2}} & \lambda_{min} \leq \lambda < 380 \\ & \& 415 < \lambda < 650 \\ & \& 700 < \lambda \leq \lambda_{max} \\ 126 & 380 \leq \lambda \leq 415 \\ 255 & 650 \leq \lambda \leq 700 \end{cases} \quad (4)$$

$$M(\lambda) = \begin{cases} e^{-\frac{(\lambda-540)^2}{2 \cdot 39^2}} & \lambda_{min} \leq \lambda < 380 \\ & \& 415 < \lambda < 650 \\ & \& 700 < \lambda \leq \lambda_{max} \\ 0 & 380 \leq \lambda \leq 415 \\ & \& 650 \leq \lambda \leq 700 \end{cases} \quad (5)$$

$$S(\lambda) = \begin{cases} e^{-\frac{(\lambda-467)^2}{2 \cdot 24,4^2}} & \lambda_{min} \leq \lambda < 380 \\ & \& 415 < \lambda < 650 \\ & \& 700 < \lambda \leq \lambda_{max} \\ 219 & 380 \leq \lambda \leq 415 \\ 0 & 650 \leq \lambda \leq 700 \end{cases} \quad (6)$$

Siendo λ_{min} la longitud de onda mínima que puede ver el ojo humano y λ_{max} la longitud de onda máxima.

Este “parche”, además de inexacto, existe simplemente por los errores originales del modelo. Es por esto que decidí retomar mi investigación, para obtener una modelización del ojo que sea exacta y fisiológica en origen.

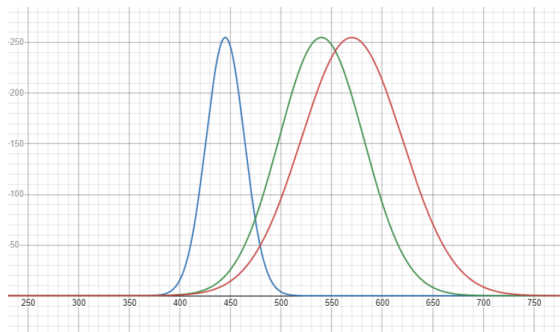


Figura 2: *Funciones gaussianas L , M y S escaladas por 255 para facilitar su visualización. Imagen tomada de una gráfica propia en Desmos (véase [Her22b]).*

2. Antecedentes

La idea de crear colores de forma aditiva, es decir, mezclando en cierta cantidad tres colores primarios fue de James Clerk Maxwell [JSR16]. Durante sus tiempos de estudiante en Cambridge a mediados del siglo XIX, desarrolló un círculo con sectores coloreados, que al girarlo, creaba la ilusión de que el círculo era de un color homogéneo, distinto a los de los sectores. Esto fundó las bases de la colorimetría, ciencia que estudia la medida de los colores y la percepción humana de estos. Maxwell organizó sus resultados en un diagrama de colores, hoy en día denominado Triángulo de Maxwell, con forma de triángulo equilátero (Fig. 3).

A principios del siglo XX, los científicos W.D. Wright y J. Guild llevaron a cabo de forma individual dos experimentos muy similares, que se basaban en las ideas de un espacio de color aditivo de Maxwell [Abr16]. En ellos, un conjunto de sujetos con visión estándar (sin trastornos de la visión, como el amarillamiento de las lentes o el daltonismo) intentaban mezclar tres luces de colores rojo, verde y azul para conseguir un color proporcionado por los científicos. Al contrastar sus resultados con múltiples colores, se dieron cuenta de que las pro-

porciones de los primarios que utilizaban los sujetos en las pruebas coincidían. Fueron tan exactos que el CIE (Commission Internationale de l'éclairage, una organización encargada de desarrollar convenios mundiales para el color y la luz) creó a partir de su trabajo un modelo que relacionaba cada longitud de onda con un color RGB. Esto es lo que llamamos el CIE 1931, que después sería sustituido por una versión mejorada, el CIE 1964, el cual tenía validez para un sujeto simulado con mayor campo de visión (10° en vez de 2°). Este modelo es el estándar mundial, y se usa en el desarrollo de todo tipo de tecnologías, como pantallas o impresoras.

Por último existe una respuesta algorítmica ¹ [DSP10]. En ella, el espectro visible se divide en secciones, y en cada una de ellas se les da a los colores primarios normas distintas (Fig. 4).

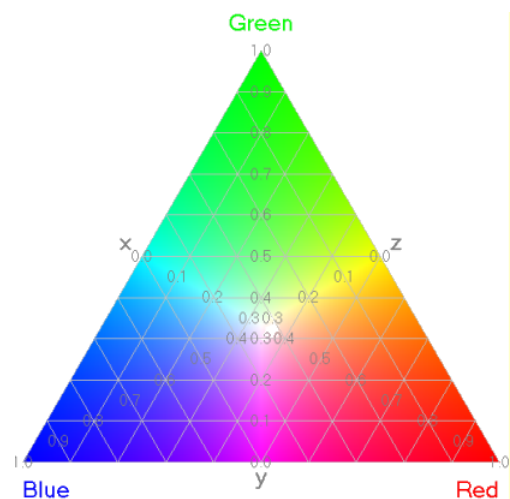


Figura 3: *El triángulo de Maxwell. Imagen tomada de la Fundación de Clerk Maxwell [Rei]*

¹El primer intento para esta investigación estaba completamente basado en este algoritmo, pretendiendo convertir las funciones gaussianas en las gráficas que se encuentran en la Fig. 4. Tras muchas versiones a prueba y error, la falta de bases empíricas sobre las que contrastar mis múltiples hipótesis acabaron haciendo que cambiase las bases del modelo de una forma drástica, para poder sustentarme en los paradigmas de la teoría de colores y biología.

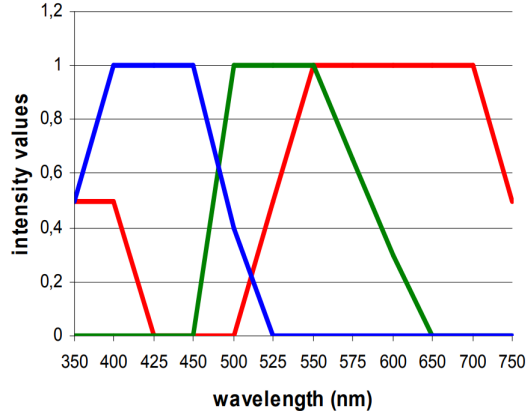


Figura 4: Gráfica de los valores del rojo, verde y azul en un algoritmo de simulación del espectro luminoso (véase [DSP10])

3. Objetivos

El objetivo principal del proyecto es encontrar una función que convierta los estímulos LMS directamente a RGB. Debe tener las siguientes características:

- Debe ser de origen fisiológico, es decir, la entrada de la función debe ser una representación de cómo actúa biológicamente nuestra vista (en este caso, usando las respuestas de los tres tipos de conos). Ésta es la mayor diferencia entre nuestro modelo y las metodologías ya existentes, como el CIE, que usa datos experimentales en vez de tener en consideración los conos directamente.
- No puede ser una función por partes ni tener modificaciones para cada cono, es decir, debemos encontrar una norma general para todo el continuo de colores. Por dos razones:
 - Para diferenciarnos del algoritmo.
 - Como reto matemático.
- No puede hacer ninguna presuposición de la visión que estamos simulando. Gracias a ésto, junto a la procedencia biológica de la respuesta, permitiría simular visiones excepcionales, como por ejemplo la de un sujeto daltónico.

4. Metodología

La referencia principal para el apartado matemático será *Color Vision*, por Bruce MacEvoy (véase [Mac15]).

Para tener una mayor flexibilidad a la hora de hacer comprobaciones gráficas opté por desarrollar mi propio software, en lugar de usar calculadoras diseñadas por terceros. Las tecnologías y el sistema operativo usados fueron los siguientes:

- Python v3.9.5
- Matplotlib v3.4.2
- Visual Studio Code v1.59.0
- Pop!_OS 21.04 (Linux)

4.1. Los espacios de excitación y color

Sean:

- LMS el espacio de excitación (un espacio \mathbb{R}^3 al que cada uno de sus ejes se le atribuye la respuesta de un cono).
- RGB el espacio de color (un espacio \mathbb{R}^3 al que cada uno de sus ejes se le atribuye un color primario).

Podemos definir una función paramétrica $P(\lambda)$ en el espacio LMS, tal que:

$$\vec{P}(\lambda) = \begin{bmatrix} L(\lambda) \\ M(\lambda) \\ S(\lambda) \end{bmatrix} \quad (7)$$

Donde L, M y S son las funciones gaussianas presentadas en (1), (2) y (3).

En la mayoría de los casos naturales, un haz de luz no está formado por una sola onda, sino por varias, cada una con una longitud de onda distinta. En nuestro caso, sólo vamos a tener en consideración un haz de luz formado por una onda, ya que después sería muy simple implementar las ondas compuestas. Esto va a provocar que haya colores imposibles de percibir para nuestro modelo, y por consiguiente,

puntos en el espacio LMS que son inaccesibles. La paramétrica \vec{P} (Fig. 5) es la que nos relacionará la longitud de onda con un punto en el espacio LMS, dándonos así el subconjunto de puntos que sí son accesibles para nuestro modelo.

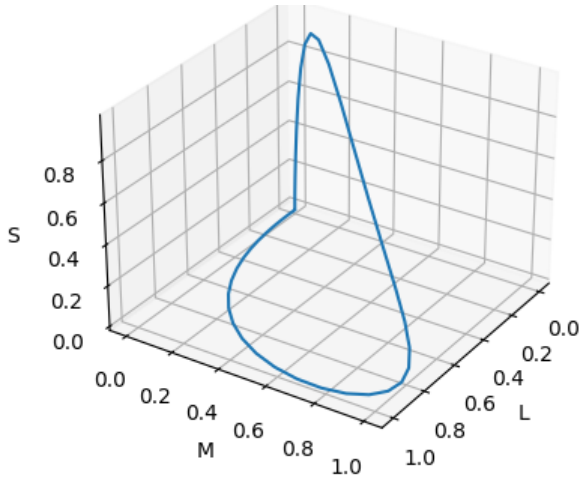


Figura 5: Función paramétrica \vec{P} . Imagen realizada con software propio (código en 8.1)

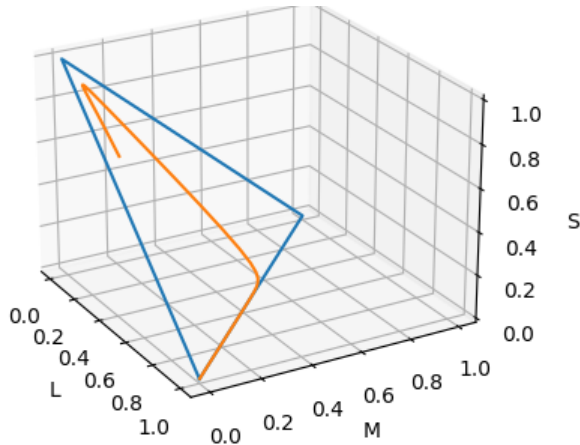


Figura 6: Función paramétrica \vec{P} normalizada (en naranja) dentro del triángulo de Maxwell (en azul). Imagen realizada con software propio (código en 8.2)

Nuestro modelo, además, no tiene en cuenta la densidad de luz que emana de la fuente, lo que se denomina luminosidad de un color y que se expresa como la suma de L, M y S. Para eli-

minarla y quedarnos únicamente con el croma del color, dividimos \vec{P} entre la luminosidad²:

$$\frac{\vec{P}(\lambda)}{B} = \begin{bmatrix} \frac{L(\lambda)}{B} \\ \frac{M(\lambda)}{B} \\ \frac{S(\lambda)}{B} \end{bmatrix} \quad (8)$$

Donde B es la luminosidad (Brightness).

De manera intuitiva, observamos que esto equivale a obtener el peso de cada cono en la respuesta total.

4.2. El triángulo de Maxwell

Además, aparece una propiedad de lo más interesante, y es que la función se encuentra sobre un plano inclinado en el cubo unitario. Si dibujamos un triángulo equilátero que tenga los vértices en los valores máximos de cada gaussiana (es decir, [0, 0, 1], [1, 0, 0] y [0, 1, 0]), veríamos que éste también se encuentra sobre el plano (Fig. 6). Este triángulo es un triángulo de Maxwell, que en vez de tener en sus vértices los primarios RGB, tiene los colores que percibiríamos con la respuesta máxima de un único cono, los cuales se denominan primarios LMS. A estos primarios se les clasifica como imaginarios, ya que no se pueden ver. Esto se debe al simple hecho de que los conos se encuentran muy cerca, y si uno se activa le transmitirá ruido a sus células vecinas, que pueden ser de diferentes tipos, haciendo efectivamente imposible la existencia de una señal pura. Para expresar en qué cantidad se deben mezclar los primarios para cada punto en el triángulo, dibujamos tres líneas opuestas a cada uno de los tres vértices del triángulo y que pasen por ese punto. Al cortar con cada una de las aristas, veremos la proporción entre la longitud del segmento creado y la longitud de la arista (Fig. 3), siendo 1 cuando el punto está en un vértice (un primario al máximo) o 0 cuando está en el vértice contrario (un primario al mínimo).

Para realizar esto seguiremos los siguientes pasos:

²De ahora en adelante, cuando nos refiramos a \vec{P} , usaremos esta versión normalizada (8).

- **Cálculo de la longitud de la arista.** Como todas las aristas son la diagonal de un cuadrado unitario, ésta es $\sqrt{2}$, dado por el teorema de Pitágoras.

- **Hallar la longitud del segmento.** Para esto, definiremos en primer lugar tres funciones lineales equivalentes a las aristas del triángulo (teniendo en consideración que los ángulos interiores de un triángulo equilátero son todos $\frac{\pi}{3}$):

$$A_1 = \tan\left(\frac{\pi}{3}\right)x = \sqrt{3}x \quad (9)$$

$$A_2 = 0 \quad (10)$$

Para la tercera arista usaremos la definición de función lineal a partir de dos puntos (x_1, y_1) y (x_2, y_2) :

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \quad (11)$$

Definiremos el primer punto como $(\sqrt{2}, 0)$, el final de la base del triángulo. El segundo punto será el vértice superior. Su posición en el eje X es la mitad de la base, $\frac{\sqrt{2}}{2}$, y su altura será equivalente a la del triángulo. Esto viene dado otra vez por el teorema de Pitágoras:

$$\begin{aligned} y_2 &= \sqrt{\left(\sqrt{2}\right)^2 - \left(\frac{\sqrt{2}}{2}\right)^2} = \\ &= \sqrt{2 - \frac{1}{2}} = \\ &= \sqrt{\frac{3}{2}} \end{aligned} \quad (12)$$

Por lo que el segundo punto será $(\frac{\sqrt{2}}{2}, \sqrt{\frac{3}{2}})$. Sustituyendo nuestros datos en (11):

$$\frac{x - \sqrt{2}}{\frac{\sqrt{2}}{2} - \sqrt{2}} = \frac{A_3}{\sqrt{\frac{3}{2}}} \quad (13)$$

Desarrollamos el lado derecho:

$$\frac{x - \sqrt{2}}{\frac{\sqrt{2}}{2} - \sqrt{2}} = \frac{x - \sqrt{2}}{-\frac{\sqrt{2}}{2}} = \frac{-2x + 2\sqrt{2}}{\sqrt{2}} \quad (14)$$

De esa forma:

$$\begin{aligned} \frac{A_3}{\sqrt{\frac{3}{2}}} &= \frac{-2x + 2\sqrt{2}}{\sqrt{2}} \\ A_3 &= \left(\sqrt{\frac{3}{2}}\right) \frac{-2x + 2\sqrt{2}}{\sqrt{2}} = \\ &= \frac{-2\sqrt{3}x + 2\sqrt{6}}{2} = \\ &= -\sqrt{3}x + \sqrt{6} \end{aligned} \quad (15)$$

Como las tres líneas que van a pasar por el punto en cuestión siempre van a ser paralelas a estas tres funciones, lo que podemos hacer para definir las matemáticamente es reutilizar los términos dependientes de x y redefinir el independiente, de tal forma que éste sea proporcional al ratio entre el segmento y la arista:

$$L_1 = \sqrt{3}x - a\sqrt{6} \quad (16)$$

$$L_2 = -\sqrt{3}x + (1 - b)\sqrt{6} \quad (17)$$

$$L_3 = \sqrt{\frac{3}{2}}c \quad (18)$$

Donde a , b y c son la proporción entre los segmentos que forman con sus respectivas aristas.

Realmente, lo único que estamos haciendo es desplazar las aristas sin variar su inclinación, construyendo una familia de funciones que son paralelas entre sí.

Como las coordenadas de los puntos del triángulo nos las dará \vec{P} , podemos despejar a , b y c , tal que:

$$a = \frac{\sqrt{3}x - y}{\sqrt{6}} \quad (19)$$

$$b = 1 - \frac{y + \sqrt{3}x}{\sqrt{6}} \quad (20)$$

$$c = \frac{\sqrt{2}y}{\sqrt{3}} \quad (21)$$

Como dicen, una imagen vale más que mil palabras, por lo que he implementado estas fórmulas en un entorno gráfico para demostrar su funcionamiento de una forma más visual (véase [Her22a]).

Ahora nos encontramos con dos problemáticas:

- Las expresiones anteriormente desarrolladas tienen en consideración un sistema de coordenadas que vive en el plano inclinado, con su origen en el vértice izquierdo del triángulo. Nuestro espacio no tiene ninguna de esas características: es tridimensional, no bidimensional, y además su origen de coordenadas se encuentra justo debajo del vértice superior del triángulo (Apartado 4.3).
- Además, este triángulo de Maxwell no es RGB, sino LMS, por lo que debemos efectuar una transformación sobre éste (Apartado 4.4).

4.3. Cambio de bases

Para resolver el primer problema tenemos que efectuar un cambio de bases que sitúe el triángulo en el plano XY. Primero, tenemos que eliminar la rotación horizontal de 45° que tiene el triángulo y \vec{P} , girándolos sobre el eje Z. Sea R_z la matriz de rotación sobre el eje Z:

$$R_z(\theta_1)\vec{P} = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{P} \quad (22)$$

Donde θ_1 es $\frac{\pi}{4}$.

En segundo lugar, debemos eliminar la inclinación del triángulo, haciéndolo girar alrededor del eje X o Y indistintamente. Sea θ_2 el

ángulo por el que lo debemos rotar, y teniendo en consideración que el plano inclinado describe un triángulo rectángulo de catetos iguales a $\frac{\sqrt{2}}{2}$ y 1 con los ejes de coordenadas (Fig. 7):

$$\theta_2 = 2\pi - \arctan \frac{2}{\sqrt{2}} \quad (23)$$

La resta se debe a que las rotaciones se realizan en sentido antihorario, por lo que tenemos que calcular el conjugado del ángulo de inclinación del plano.

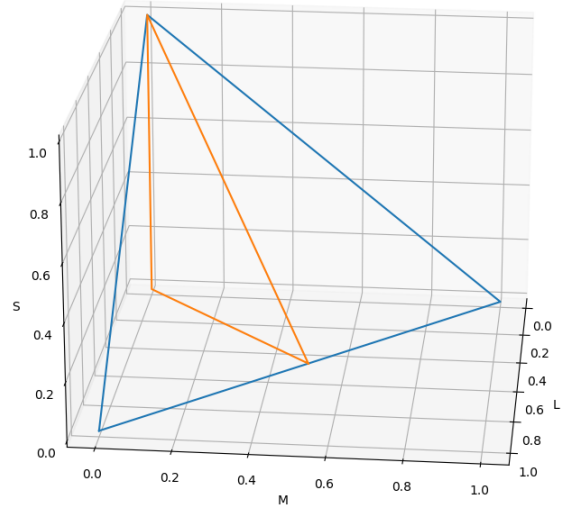


Figura 7: Triángulo rectángulo debajo del plano inclinado. Imagen realizada con software propio (código en 8.3)

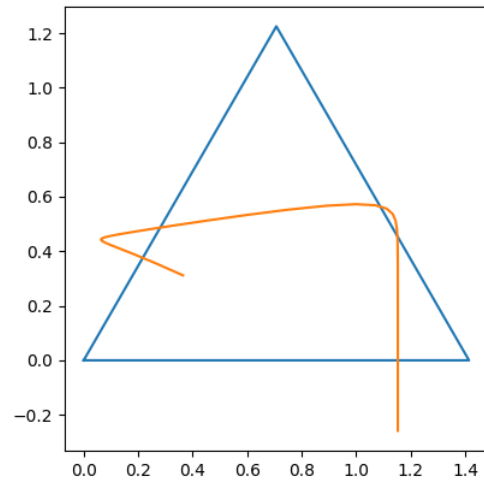


Figura 8: Función \vec{P} rotada 90° . Imagen realizada con software propio (código en 8.4)

Sea R_y la matriz de rotación sobre el eje Y, podemos efectuar la siguiente transformación:

$$R_y(\theta_2)R_z(\theta_1)\vec{P} \quad (24)$$

Finalmente, como \vec{P} ya se encuentra en un plano paralelo al XY, podemos eliminar directamente la coordenada Z:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} R_y(\theta_2)R_z(\theta_1)\vec{P} \quad (25)$$

Consiguiendo así un espacio bidimensional.

Desarrollamos la primera transformación lineal:

$$\begin{aligned} & \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{P} = \\ & = \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \\ 0 \end{bmatrix} L(\lambda) + \begin{bmatrix} -\sin \theta_1 \\ \cos \theta_1 \\ 0 \end{bmatrix} M(\lambda) + \\ & + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} S(\lambda) = \\ & = \begin{bmatrix} L \cos \theta_1 - M \sin \theta_1 \\ L \sin \theta_1 + M \cos \theta_1 \\ S \end{bmatrix} \end{aligned} \quad (26)$$

Y finalmente la segunda:

$$\begin{aligned} & \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 \\ 0 & 1 & 0 \\ -\sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix} \begin{bmatrix} L \cos \theta_1 - M \sin \theta_1 \\ L \sin \theta_1 + M \cos \theta_1 \\ S \end{bmatrix} \\ & = \begin{bmatrix} \cos \theta_2 \\ 0 \\ -\sin \theta_2 \end{bmatrix} (L \cos \theta_1 - M \sin \theta_1) + \\ & + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} (L \sin \theta_1 + M \cos \theta_1) + \\ & + \begin{bmatrix} \sin \theta_2 \\ 0 \\ \cos \theta_2 \end{bmatrix} S \end{aligned} \quad (27)$$

Como vamos a eliminar el valor de la coordenada Z en la siguiente transformación, podemos obviarla directamente y presentar la respuesta como un vector bidimensional:

$$\vec{P}_T = \begin{bmatrix} L \cos \theta_1 \cos \theta_2 - M \sin \theta_1 \cos \theta_2 + S \sin \theta_2 \\ L \sin \theta_1 + M \cos \theta_1 \end{bmatrix} \quad (28)$$

\vec{P}_T se encuentra girada por 90° (Fig. 8). Esto se puede solucionar intercambiando los componentes y volviendo negativo al X, obteniendo:

$$\vec{P}_T = \begin{bmatrix} L \sin \theta_1 + M \cos \theta_1 \\ M \sin \theta_1 \cos \theta_2 - L \cos \theta_1 \cos \theta_2 - S \sin \theta_2 \end{bmatrix} \quad (29)$$

Para finalizar, debemos fijar el origen en el vértice izquierdo del triángulo. Para ello, calcularemos la posición del vértice después de la transformación:

$$\begin{aligned} V &= \begin{bmatrix} 1 \sin \theta_1 + 0 \cos \theta_1 \\ 0 \sin \theta_1 \cos \theta_2 - 1 \cos \theta_1 \cos \theta_2 - 0 \sin \theta_2 \end{bmatrix} = \\ &= \begin{bmatrix} \sin \theta_1 \\ -\cos \theta_1 \cos \theta_2 \end{bmatrix} \end{aligned} \quad (30)$$

Y se lo restamos a nuestra función (Fig. 9):

$$\vec{P}_T = \vec{P}_T - V \quad (31)$$

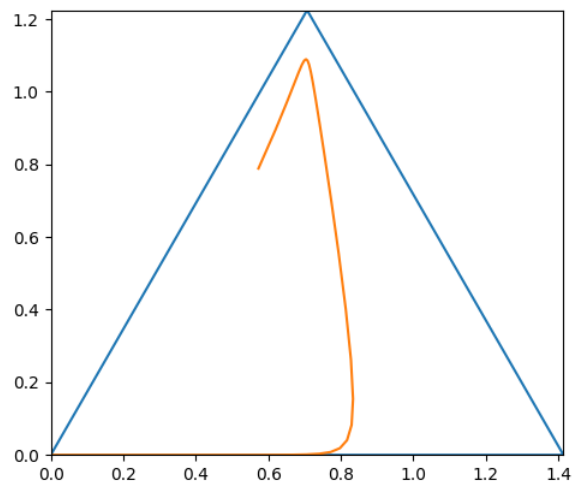


Figura 9: Función \vec{P} transformada. Imagen realizada con software propio (código en 8.5)

4.4. De LMS a RGB

Para poder convertir el triángulo LMS a RGB debemos moverlo a través del espacio de color (Fig. 10) hasta que sus vértices encajen con los primarios apropiados. Para esto, debemos desplazar el triángulo LMS (Fig. 11) hasta que el punto blanco quede en su centro (Fig. 3).

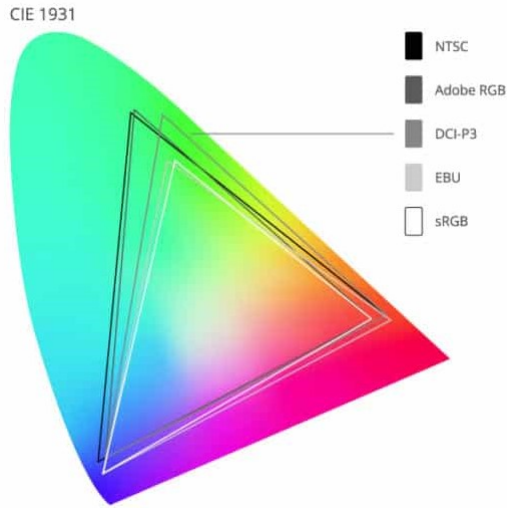


Figura 10: Rango de colores visibles para el ser humano. El triángulo blanco es el espacio RGB. Imagen tomada de: <https://www.viewsonic.com/library/es/trabajo-creativo/fotografia/que-es-la-gama-de-colores/>

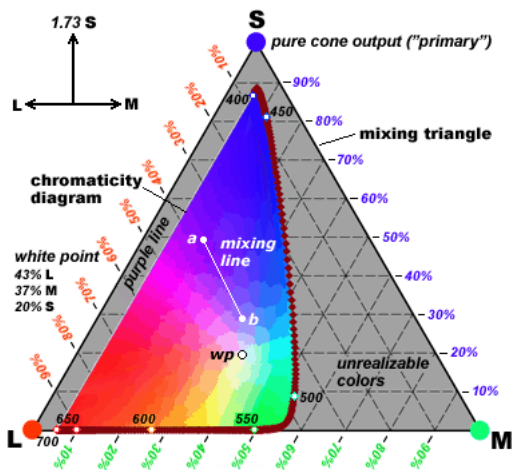


Figura 11: Triángulo LMS (véase [Mac15])

Pero, ¿dónde se encuentra el punto blanco en el espacio LMS exactamente? Depende de

cómo ponderemos cada una de las respuestas de los conos, ya que el blanco se encuentra en el punto donde el peso perceptual de todos los conos es equivalente a sus proporciones en el triángulo. Si nuestro modelo pondera al cono L como un 50 % del total, el blanco se encontrará en el punto donde la proporción del cono L es el 50 %.

El peso perceptual de cada cono se suele definir como la proporción entre el área debajo de su gaussiana y la suma de las áreas de las tres. Sea ρ_L el peso perceptual del cono L:

$$\rho_L = \frac{\int_{-\infty}^{\infty} L(\lambda) d\lambda}{\sigma} \quad (32)$$

Donde:

$$\sigma = \int_{-\infty}^{\infty} (L(\lambda) + M(\lambda) + S(\lambda)) d\lambda \quad (33)$$

Siendo así el peso para el resto de los conos:

$$\rho_M = \frac{\int_{-\infty}^{\infty} M(\lambda) d\lambda}{\sigma} \quad (34)$$

$$\rho_S = \frac{\int_{-\infty}^{\infty} S(\lambda) d\lambda}{\sigma} \quad (35)$$

Usando (19) a (21) podemos convertir estas proporciones a las coordenadas del punto blanco W . Para ello, en primer lugar despejamos x e y :

$$y = \sqrt{\frac{3}{2}}c = \frac{c\sqrt{6}}{2} \quad (36)$$

$$x = \frac{y + a\sqrt{6}}{\sqrt{3}} = \frac{\sqrt{6}\left(\frac{c}{2} + a\right)}{\sqrt{3}} \quad (37)$$

Entonces:

$$W = \begin{bmatrix} \sqrt{6}\left(\frac{\rho_S}{2} + \rho_M\right) \\ \frac{\sqrt{3}}{\rho_S\sqrt{6}} \end{bmatrix} \quad (38)$$

Calculamos la diferencia entre el punto blanco LMS y el RGB:

$$W_{\text{RGB}} - W_{\text{LMS}} = \begin{bmatrix} \frac{\sqrt{6} \left(\frac{1}{3} - \frac{\rho_S}{2} + \frac{1}{3} - \rho_M \right)}{\sqrt{3}} \\ \frac{\sqrt{6} \left(\frac{1}{3} - \rho_S \right)}{2} \end{bmatrix} \quad (39)$$

Ahora, para realizar la conversión a RGB:

$$\vec{P}_{\text{RGB}} = \vec{P}_T + (W_{\text{RGB}} - W_{\text{LMS}}) \quad (40)$$

Esta ecuación es la **modelización matemática generalizada de la percepción cromática del ojo humano**. Para comprobar su funcionamiento creé un simulador web de la visión estándar, que se puede encontrar en <https://alexstatis.github.io/lumen/>.

5. Resultados

Para comprobar la veracidad del simulador desarrollé mediante prueba y error un grupo de nuevas gaussianas (Fig. 12) que, aunque su forma no es tan parecida a la realidad, hacen un mejor trabajo que las anteriores en colocar las longitudes de onda en los puntos LMS adecuados. El mayor problema que había con las anteriores era que, aunque su forma parecía encajar con la referencia (véase [Mac15]), las longitudes de onda están desplazadas por cierto valor, muy posiblemente debido a que las derivadas no concuerdan con el resultado real. Para poder tener una fuente de colores sobre la que contrastar los resultados, usé una versión de la respuesta algorítmica [Bru] que se trató en la *Introducción*.

Como se puede comprobar en la tabla del 8.7, la gama de colores que se encuentra entre 450 nm - 550 nm son muy similares. Sin embargo, cuanto más nos acercamos a los tonos amarillos los resultados empiezan a diferir. Lo que sucede en este rango de colores es que comienza a ser notable el efecto de desplazamiento anteriormente comentado, por lo que los colores amarillentos comienzan un poco antes de

lo debido (aprox. 30 nm), haciendo que en la tabla los colores del modelo fisiológico estén “adelantados” por una celda de distancia de los colores del algoritmo. Los resultados vuelven a aproximarse al entrar en el rojo puro, pero sin duda, el rango 380 nm - 420 nm es el que más se aleja del referente. Esto se debe a que los violetas que residen en esa zona del espectro se encuentran fuera de la gama RGB (Fig. 10).

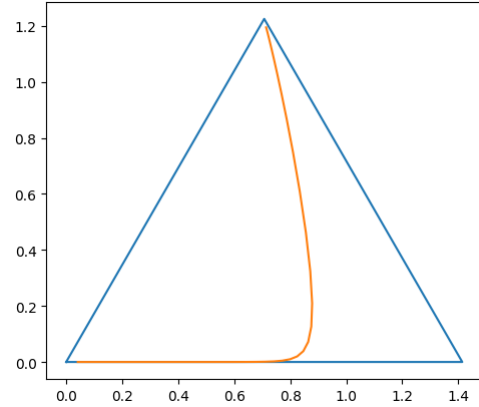


Figura 12: Versión modificada de \vec{P} para representar con mayor exactitud la relación λ -color. Imagen realizada con software propio (código en 8.6)

6. Conclusiones

Se ha comprobado que es posible encontrar una forma de convertir los estímulos oculares a un espacio de color usando únicamente la respuesta de los conos. Esta metodología no tiene como misión ser una sustituta de estándares comerciales como los del CIE, que tienen en consideración únicamente a personas sin trastornos de la visión, si no que nos permite a los investigadores trabajar en situaciones en las cuales las funciones-respuesta de los conos sean dinámicas. Hay una gran variedad de ejemplos en los que se precisa de esta característica variable, como estudios en simulaciones de distintos casos de daltonismo o algoritmos evolutivos (simulaciones virtuales de la evolución biológica) con un enfoque a la visión.

Como se ha tratado en *Resultados*, hay dos factores influyentes en nuestra imprecisión, los cuales deberían ser los principales puntos de

estudio para futuras investigaciones: las gaussianas de entrada y el espacio de salida. Para las primeras, sería posible recopilar en una tabla de datos discretos la sensibilidad de cada cono, sobre la cual se podría ejecutar unos de los múltiples métodos de interpolación polinómica. De esta forma se obtendrían funciones que se asemejan con gran exactitud a la respuesta real del ojo, pero el coste de cálculo sería mucho mayor (llegando a aumentar exponencialmente por cada nodo añadido).

En cuanto al espacio de color sobre el que basar el modelo, se podrían usar versiones más completas del sRGB, como AdobeRGB o ProPhoto RGB. En caso de tener un monitor compatible, también se podrían tener en consideración estándares más modernos como el LCH, el cual comprende toda la gama de colores visibles. En teoría, esto permitiría obtener un modelo mucho más cercano a la perfección, pero la complejidad de la transformación ascendería sustancialmente (dependiendo, por supuesto, del espacio que se elija).

En definitiva, la colorimetría bajo el punto de vista biológico es un campo lleno de potencial, y se podría relacionar con otros estudios florecientes. Este es el caso de las variedades neuronales [JO21], un campo en el cual se intenta resolver el problema de alta dimensionalidad que se encuentra a la hora de estudiar una amplia zona del cerebro con la ayuda de la topología. La idea principal es estudiar el espacio de salida de las múltiples neuronas y, si

este es un subconjunto dentro una variedad de menos dimensiones que el espacio de salida, el problema se puede simplificar a uno de dimensiones equivalentes al de la variedad. Nuestro modelo es, en cierto modo, una versión de esta simplificación. Tenemos un espacio de entrada de tres dimensiones que se encuentra en un triángulo (la variedad) que es bidimensional, y definiendo una transformación que nos define su espacio de coordenadas local en relación al global, simplificamos nuestro problema a una dimensión menor. Este suceso ocurre en múltiples partes del cerebro, no solamente las relacionadas con la visión. Por ejemplo, se han realizado estudios [JO21] sobre cómo el cerebro de pequeños mamíferos guardan la dirección en la que están mirando, basándose únicamente en esta concepción de que el cerebro almacena datos confinándolos en una variedad.

7. Agradecimientos

Gracias a mis profesores Belén Álvarez y Jose Manuel Casal por su maravilloso trabajo coordinando el proyecto y estando ahí para apoyarme; y a Ramiro García, Jefe de Estudios del CPR Plurilingüe San Fermín, por darme la oportunidad de participar en este proyecto.

Dedico este documento a César Tejada, mi antiguo tutor, pues sin él nunca se habría formulado la duda curiosa que dio lugar a este proyecto.

8. Anexos

8.1. Anexo 1

```
1 ###
2 ### Funcion parametrica P
3 ###
4
5 import matplotlib.pyplot as plt
6 from mpl_toolkits import mplot3d
7 from mpl_toolkits.mplot3d import Axes3D
8 import mpl_toolkits.mplot3d.art3d as art3d
9 from mpl_toolkits.mplot3d.art3d import Poly3DCollection
10 import numpy as np
11
12 import matplotlib.pyplot as plt
13 from mpl_toolkits import mplot3d
14 import numpy as np
15
16 # Definicion del rango de longitudes de onda visibles
17 wavelength = np.linspace(380,720,50)
18
19 fig = plt.figure()
20 ax = plt.axes(projection = '3d')
21 fig.add_axes(ax)
22
23 ax.set_xlabel('L')
24 ax.set_ylabel('M')
25 ax.set_zlabel('S')
26
27 # Defincion de las gaussianas L, M y S
28 x = np.exp( ( -(wavelength-570) ** (2) ) / ( 2*(50**2) ) )
29 y = np.exp( ( -(wavelength-540) ** (2) ) / ( 2*(39**2) ) )
30 z = np.exp( ( -(wavelength-467) ** (2) ) / ( 2*(24.4**2) ) )
31
32 ax.plot(x,y,z)
33 plt.show()
```

8.2. Anexo 2

```
1 ###
2 ### Funcion parametrica normalizada
3 ###
4
5 import matplotlib.pyplot as plt
6 from mpl_toolkits import mplot3d
7 from mpl_toolkits.mplot3d import Axes3D
8 import mpl_toolkits.mplot3d.art3d as art3d
9 from mpl_toolkits.mplot3d.art3d import Poly3DCollection
10 from matplotlib.patches import Circle, PathPatch
11 import numpy as np
12
13 import matplotlib.pyplot as plt
14 from mpl_toolkits import mplot3d
15 import numpy as np
16
17 wavelength = np.linspace(380,720,50)
18
19 fig = plt.figure()
20 ax = plt.axes(projection = '3d')
21 fig.add_axes(ax)
22
23 # Deficion de los vertices del triangulo de Maxwell
24 coord = [[1,0,0], [0,0,1], [0,1,0]]
25 coord.append(coord[0])
26
27 xs, ys, zs = zip(*coord)
28 plt.plot(xs,ys, zs)
29
30 ax.set_xlabel('L')
31 ax.set_ylabel('M')
32 ax.set_zlabel('S')
33
34 x = np.exp( ( -(wavelength-570) ** (2) ) / ( 2*(50**2) ) )
35 y = np.exp( ( -(wavelength-540) ** (2) ) / ( 2*(39**2) ) )
36 z = np.exp( ( -(wavelength-467) ** (2) ) / ( 2*(24.4**2) ) )
37
38 # Definicion del brillo
39 B = x + y + z
40
41 # Normalizacion
42 x2 = x / B
43 y2 = y / B
44 z2 = z / B
45 ax.plot(x2,y2,z2)
46
47 plt.show()
```

8.3. Anexo 3

```
1 ###
2 ### Triangulo rectangulo & plano inclinado
3 ###
4
5 import matplotlib.pyplot as plt
6 from mpl_toolkits import mplot3d
7 from mpl_toolkits.mplot3d import Axes3D
8 import mpl_toolkits.mplot3d.art3d as art3d
9 from mpl_toolkits.mplot3d.art3d import Poly3DCollection
10 from matplotlib.patches import Circle, PathPatch
11 import numpy as np
12
13 import matplotlib.pyplot as plt
14 from mpl_toolkits import mplot3d
15 import numpy as np
16
17 wavelength = np.linspace(380,720,50)
18
19 fig = plt.figure()
20 ax = plt.axes(projection = '3d')
21 fig.add_axes(ax)
22
23 # Deficion de los vertices del triangulo de Maxwell
24 coord = [[1,0,0], [0,0,1], [0,1,0]]
25 coord.append(coord[0])
26
27 xs, ys, zs = zip(*coord)
28 plt.plot(xs,ys, zs)
29
30 # Deficion del triangulo rectangulo
31 coord = [[0.5,0.5,0], [0,0,0], [0,0,1]]
32 coord.append(coord[0])
33
34 xs, ys, zs = zip(*coord)
35 plt.plot(xs,ys, zs)
36
37
38 ax.set_xlabel('L')
39 ax.set_ylabel('M')
40 ax.set_zlabel('S')
41
42 plt.show()
```


8.4. Anexo 4

```
1 # Funcion parametrica transformada, pero rotada 90 grados
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import numpy as np
6
7 wavelength = np.linspace(380,720,50)
8 fig, ax = plt.subplots()
9 fig.add_axes(ax)
10
11 coord = [[0,0], [np.sqrt(2),0], [np.sqrt(2)/2, np.sqrt(3)/2]]
12 coord.append(coord[0])
13
14 x = np.exp( ( -(wavelength-570) ** (2) ) / ( 2*(50**2) ) )
15 y = np.exp( ( -(wavelength-540) ** (2) ) / ( 2*(39**2) ) )
16 z = np.exp( ( -(wavelength-467) ** (2) ) / ( 2*(24.4**2) ) )
17
18 l = x + y + z
19 x = x / l
20 y = y / l
21 z = z / l
22 # Definicion de los angulos theta sub-1 y theta sub-2
23 theta1 = 7 * np.pi / 4
24 theta2 = 2 * np.pi - np.tanh(2 / np.sqrt(2))
25
26 # Definicion de la parametrica tras ser transformada
27 vectors = [
28     [(x * np.cos(theta2) * np.cos(theta1) - y * np.cos(theta2) * np.sin(
29         theta1) + z * np.sin(theta2))],
30     [x * np.sin(theta1) + y * np.cos(theta1)],
31 ]
32
33 xs, ys = zip(*coord)
34 plt.plot(xs,ys)
35
36 x2, y2 = zip(vectors)
37
38 x2 = x2[0][0]
39 y2 = y2[0][0]
40
41 print(x2[49], y2[49])
42
43 # Defincion del vertice izquierdo tras ser transformado
44 vertex = [
45     [np.sin(theta1)],
46     [-(np.cos(theta2) * np.cos(theta1))]
47 ]
48 # Transformacion no-lineal
49 for i in range(50):
50     x2[i] = x2[i] - vertex[0]
51     y2[i] = y2[i] - vertex[1]
52
53 plt.gca().set_aspect('equal', adjustable='box')
54 ax.plot(x2,y2)
55 plt.show()
```

8.5. Anexo 5

```
1 # Funcion parametrica transformada
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import numpy as np
6
7 wavelength = np.linspace(380,720,50)
8 fig, ax = plt.subplots()
9 fig.add_axes(ax)
10
11 coord = [[0,0], [np.sqrt(2),0], [np.sqrt(2)/2, np.sqrt(3)/2]]
12 coord.append(coord[0])
13
14 x = np.exp( ( -(wavelength-570) ** (2) ) / ( 2*(50**2) ) )
15 y = np.exp( ( -(wavelength-540) ** (2) ) / ( 2*(39**2) ) )
16 z = np.exp( ( -(wavelength-467) ** (2) ) / ( 2*(24.4**2) ) )
17
18 l = x + y + z
19 x = x / l
20 y = y / l
21 z = z / l
22 # Definicion de los angulos theta sub-1 y theta sub-2
23 theta1 = 7 * np.pi / 4
24 theta2 = 2 * np.pi - np.tanh(2 / np.sqrt(2))
25
26 # Definicion de la parametrica tras ser transformada y rotada por 90 grados
27 vectors = [
28     [x * np.sin(theta1) + y * np.cos(theta1)],
29     [-(x * np.cos(theta2) * np.cos(theta1) - y * np.cos(theta2) * np.sin(
30         theta1) + z * np.sin(theta2))],
31 ]
32
33 xs, ys = zip(*coord)
34 plt.plot(xs,ys)
35
36 x2, y2 = zip(vectors)
37
38 x2 = x2[0][0]
39 y2 = y2[0][0]
40
41 print(x2[49], y2[49])
42
43 # Defincion del vertice izquierdo tras ser transformado
44 vertex = [
45     [np.sin(theta1)],
46     [-(np.cos(theta2) * np.cos(theta1))]
47 ]
48 # Transformacion no-lineal
49 for i in range(50):
50     x2[i] = x2[i] - vertex[0]
51     y2[i] = y2[i] - vertex[1]
52
53 plt.gca().set_aspect('equal', adjustable='box')
54 ax.plot(x2,y2)
55 plt.show()
```

8.6. Anexo 6

```
1 # Funcion parametrica con mejor relacion lambda-color
2 import matplotlib.pyplot as plt
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import numpy as np
6
7 wavelength = np.linspace(380,720,50)
8 fig, ax = plt.subplots()
9 fig.add_axes(ax)
10
11 coord = [[0,0], [np.sqrt(2),0], [np.sqrt(2)/2, np.sqrt(3)/2]]
12 coord.append(coord[0])
13
14 # Redefinicion parametrica para una mejor relacion lambda-color
15 x = np.exp( ( -(wavelength-570) ** (2) ) / ( 2*(50**2) ) )
16 y = np.exp( ( -(wavelength-540) ** (2) ) / ( 2*(45**2) ) )
17 z = np.exp( ( -(wavelength-444) ** (2) ) / ( 2*(30.4**2) ) )
18
19 l = x + y + z
20 x = x / l
21 y = y / l
22 z = z / l
23
24 theta1 = 7 * np.pi / 4
25 theta2 = 2 * np.pi - np.tanh(2 / np.sqrt(2))
26 vectors = [
27     [x * np.sin(theta1) + y * np.cos(theta1)],
28     [-(x * np.cos(theta2) * np.cos(theta1) - y * np.cos(theta2) * np.sin(
29         theta1) + z * np.sin(theta2))],
30 ]
31
32 xs, ys = zip(*coord)
33 plt.plot(xs,ys)
34
35 x2, y2 = zip(*vectors)
36
37 x2 = x2[0][0]
38 y2 = y2[0][0]
39
40 print(x2[49], y2[49])
41
42 # Defincion del vertice izquierdo tras ser transformado
43 vertex = [
44     [np.sin(theta1)],
45     [-(np.cos(theta2) * np.cos(theta1))]
46 ]
47
48 # Transformacion no-lineal
49 for i in range(50):
50     x2[i] = x2[i] - vertex[0]
51     y2[i] = y2[i] - vertex[1]
52
53 plt.gca().set_aspect('equal', adjustable='box')
54 ax.plot(x2,y2)
55 plt.show()
```

8.7. Anexo 7

Longitud de onda (nm)	Algoritmo	Modelo fisiológico con funciones de prueba
400 nm		
450 nm		
485 nm		
495 nm		
510 nm		
540 nm		
570 nm		
600 nm		
630 nm		
660 nm		
720 nm		

Referencias

- [DSP10] Stela Dragulin, Livia Sangeorzan y Mircea Parpalea. “Chromatic patterns of sounds waves in Java Applets coloured sounds”. En: jun. de 2010, págs. 92-97.
- [Mac15] Bruce MacEvoy. “Light and the Eye”. En: *Color Vision*. 2015. URL: <https://www.handprint.com/HP/WCL/color1.html> (visitado 01-08-2015).
- [JSR16] "JSR". *Maxwell Colour Triangle*. 2016. URL: <https://homepages.abdn.ac.uk/j.s.reid/pages/Maxwell/Legacy/MaxTri.html>.
- [Abr16] Chandler Abraham. *A Beginner's Guide to (CIE) Colorimetry*. 2016. URL: <https://medium.com/hipster-color-science/a-beginners-guide-to-colorimetry-401f1830b65a>.
- [Cid21] César Tejada Cid. *¿De qué color son los electrones?* 2021. URL: <http://sanferminciencias.blogspot.com/2021/02/de-que-color-son-los-electrones.html>.
- [JO21] Mehrdad Jazayeri y Srdjan Ostojic. “Interpreting neural computations by examining intrinsic and embedding dimensionality of neural activity”. En: *Current Opinion in Neurobiology* 70 (2021). Computational Neuroscience, págs. 113-120. ISSN: 0959-4388. URL: <https://arxiv.org/ftp/arxiv/papers/2107/2107.04084.pdf>.
- [Her22a] Alejandro Campo Herbés. *Gráfica de un triángulo de Maxwell*. 2022. URL: <https://www.desmos.com/calculator/skzjj04y kz>.
- [Her22b] Alejandro Campo Herbés. *Gráficas de la sensibilidad espectral*. 2022. URL: <https://www.desmos.com/calculator/klnwjkj7dp>.
- [Bru] Dan Bruton. *Wavelength to Colour Relationship*. URL: <https://academo.org/demos/wavelength-to-colour-relationship/>.
- [Rei] Dr. John S. Reid. *Maxwell's creation of colour science at Aberdeen*. URL: https://clerkmaxwellfoundation.org/Colour_science_at_Aberdeen.pdf.