Trabajo Fin de Grado
Ingeniería de Tecnologías Industriales

Uso de métodos de visión artificial para la detección de la luz de semáforos

Autor: Carmen Palmero Jaén
Tutor: Antonio Javier Gallego Len

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2022

Proyecto Fin de Carrera
Ingeniería de Tecnologías Industriales

Uso de métodos de visión artificial para la detección de la luz de semáforos

Autor:
Carmen Palmero Jaén

Tutor:
Antonio Javier Gallego Len
Profesor Sustituto Interino

Dpto. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2022

Proyecto Fin de Carrera: Uso de métodos de visión artificial para la detección de la luz de semáforos

Autor: Carmen Palmero Jaén

Tutor: Antonio Javier Gallego Len

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

A mis padres, que han hecho posible mi felicidad y formación con sus valores y amor.

A mi hermano y Alejandro.

Agradecimientos

Tras cuatro años de esfuerzo y felicidad plena al escribir esto, mis primeras palabras de agradecimiento sin duda alguna son a mis padres, a mi hermano y a Alejandro, que creyeron en mí desde el momento que hice la primera maleta y salí de casa para mudarme a Sevilla a cumplir el sueño de ser Ingeniera. Siempre han estado seguros y me han transmitido la confianza de que lo conseguiría, además de darme una fuerza incalculable cuando más lo he necesitado.

Gracias a Antonio, mi tutor. Por su motivación constante y por guiarme y ayudarme cada vez que lo he necesitado.

Gracias a vosotros, mis amigos. Mercedes, Manolo, Sergio, Lucía y Paco. Sin duda el fin de una carrera no es el hecho de llegar hasta aquí, sino de qué forma vives el proceso. Juntos hemos caído y nos hemos empujado para llegar a donde estamos hoy con innumerables momentos de felicidad plena vividos juntos.

A Laura y Mar, que, sin duda, son el día a día que cualquier persona desearía tener. Por estar tan presentes durante estos dos años, por celebrar los logros de unas y otras como si de propios se tratasen y, sobre todo, por ser cómplices, apoyo y hogar. Gracias de verdad.

Gracias, por tanto, a todo aquel que ha creído en mi.

Carmen Palmero Jaén

Estudiante de Grado en Ingeniería de Tecnologías Industriales

Sevilla, 2022

Resumen

El objetivo del presente trabajo de fin de grado es el desarrollo de un método de detección de luces de semáforos usando Visión Artificial.

Para llevar a cabo dicho procedimiento, en primer lugar, se realizará un tratamiento previo de la imagen necesario para descartar zonas que no sean de interés en la misma. Una vez realizado dicho sistema, se obtendrá una imagen binaria de bordes que será la imagen con la que se trabajará para la detección de las zonas de interés.

Las luces de semáforos son círculos que se detectarán mediante la aplicación de la Transformada de Hough, dando lugar a nuevas imágenes que almacenarán solo la zona de interés, es decir, la luz del semáforo, y que serán usadas para su posterior clasificación.

El último paso consistirá en desarrollar mediante el cálculo de indicadores de bondad, distintos métodos de clasificación hasta la obtención de aquel que proporcione resultados concluyentes.

Abstract

The objective of this final degree project is to develop a method for detecting traffic lights using Artificial Vision.

To carry out this procedure, first of all, the image will be pre-treated in order to discard areas that are not of interest in the image. Once this system has been carried out, a binary image of edges will be obtained, which will be the image that will be used to detect the areas of interest.

The traffic lights are circles that will be detected by applying the Hough Transform, giving rise to new images that will store only the area of interest, that is, the traffic light, which will be used to classify them.

The last step will consist of developing, by calculating goodness-of-fit indicators, different classification methods until the one that provides conclusive results is obtained.

Índice

Agradecimientos **ix**

Resumen **xi**

Abstract **xiii**

Índice **xv**

Índice de Tablas **xvii**

Índice de Figuras **xix**

1 Introducción **1**

2 Preprocesamiento de las imágenes **7**

2.1. Introducción al procesamiento de imágenes y su aplicación a semáforos *7*

2.1.1 Separación de la imagen en canales RGB 7

2.1.2 Umbralización por color 12

2.1.3 Métodos morfológicos 17

2.1.4 Representación de la segmentación 20

2.1.5 Conclusiones del capítulo 22

3 Detección de Círculos **23**

3.1 Adecuación de la imagen *23*

3.1.1 Imagen en escala de grises 23

3.1.2 Imagen binaria de bordes y reducción a un píxel 25

3.2 Transformada de Hough *27*

3.2.1 Introducción a la transformada de Hough 27

3.2.2 Detección de líneas rectas 27

3.2.3 Detección de círculos 29

3.3 Resumen del capítulo *34*

4 Extracción de la zona de interés **35**

4.1 Procedimiento de extracción *35*

4.2 Zonas recortadas *36*

5 Métodos de clasificación **37**

5.1 Clasificación por correlación *37*

5.1.1 Elección de patrones 37

5.1.2 Obtención del coeficiente de correlación 40

5.2 Método Perceptrón *44*

5.2.1 Introducción al clasificador no-paramétrico entrenable 44

5.2.2 Características discriminantes 45

5.2.3 Desarrollo del método 47

5.2.4 Resultados obtenidos 48

5.3 Resumen del capítulo *53*

6 Conclusiones y líneas futuras **54**

Referencias **57**

ÍNDICE DE TABLAS

Tabla 2-1. Umbrales óptimos para segmentación 13
Tabla 2-2. Valores normalizados de los canales RGB para cada color del semáforo 14
Tabla 4-1. Recopilación de las zonas de interés de cada semáforo estudiado 36
Tabla 5-1. Clasificación por correlación: Sombreado claro: Entrenamiento; Sombreado oscuro: Pruebas 42
Tabla 5-2. Valores medios de cada canal. Sombreado: Claro: Entrenamiento; Oscuro: Comprobación 46
Tabla 5-3. Resultados correspondientes al Método Perceptrón 52

ÍNDICE DE FIGURAS

Figura 1-1. Sección transversal del ojo humano [3]. 2
Figura 1-2. Etapas de un proceso de Visión Artificial [2]. 3
Figura 1-3. Imágenes de semáforos 4
Figura 1-4. Ejemplo de vía de tránsito caracterizada por la presencia de numerosos semáforos. 5
Figura 1-5. Escena en la que el círculo del semáforo no es el de mayor tamaño. 5
Figura 2-1. Modelo RGB 8
Figura 2-2. Representación de los canales RGB para un semáforo verde. 8
Figura 2-3. Imagen obtenida con la herramienta *imtool* de MATLAB para inspección. 9
Figura 2-4. Canales RGB (rojo, verde y azul de izquierda a derecha) de un semáforo rojo. 10
Figura 2-5. *imtool* de un semáforo rojo. 10
Figura 2-6. Dos imágenes de semáforos ámbar, tomadas en condiciones distintas de iluminación 11
Figura 2-7. Separación en canales RGB de un semáforo ámbar. 11
Figura 2-8. Representación de histogramas del canal rojo para dos semáforos distintos. 12
Figura 2-9. Demostración de píxeles blancos en semáforos 14
Figura 2-10. Diferentes tonalidades de semáforos verdes extraídas de imágenes reales. 14
Figura 2-11. Semáforo rojo segmentado mediante umbralización por color 15
Figura 2-12. Semáforo verde segmentado mediante umbralización por color 16
Figura 2-13. Semáforo ámbar segmentado mediante umbralización por color 16
Figura 2-14. Representación del proceso de erosión 17
Figura 2-15. Representación del proceso de dilatación 17
Figura 2-16. Ejemplo de aplicación de apertura a una imagen binaria. 18
Figura 2-17. Aplicación de métodos morfológicos a semáforo verde 18
Figura 2-18. Aplicación de métodos morfológicos a semáforo ámbar 19
Figura 2-19. Aplicación de métodos morfológicos a semáforo rojo 19
Figura 2-20. Segmentación completa de semáforo rojo 20
Figura 2-21. Segmentación completa de semáforo ámbar 21
Figura 2-22. Segmentación completa de semáforo verde 21
Figura 2-23. A la izquierda imagen real y su correspondiente histograma. A la derecha imagen realzada 22
Figura 3-1. Imagen RGB segmentada 24
Figura 3-2. Imagen segmentada en escala de grises 24
Figura 3-3. Procedimiento para la extracción de bordes de un píxel 25
Figura 3-4. Obtención de imagen binaria de bordes de un píxel para semáforo rojo 26
Figura 3-5. Direcciones de código cadena 26
Figura 3-6. a) Plano x-y; b) Espacio de parámetros 27

Figura 3-7. (a) Rectas en el plano x-y; (b) Curvas en el plano θ-ρ 28
Figura 3-8. A la izquierda imagen binaria de bordes; a la derecha detección del círculo 30
Figura 3-9. Detección de círculos poco definidos de semáforos en ámbar 30
Figura 3-10. Detección de círculos en semáforo lejano verde 31
Figura 3-11. Detección de semáforo rojo con señales de tráfico presentes 32
Figura 3-12. Ejemplo de detección de luz verde 33
Figura 3-13. Detección de círculo en escena con señal de tráfico 33
Figura 4-1. Círculo inscrito en cuadrado 35
Figura 5-1. Colores sintéticos 37
Figura 5-2. De izquierda a derecha: Tonalidad naranja, tonalidad amarilla e imagen sintética 38
Figura 5-3. De izquierda a derecha: Tonalidad verde, tonalidad turquesa e imagen sintética 38
Figura 5-4. Explicación visual de la formación de matrices 39
Figura 5-5. Ilustración del cálculo del valor medio por píxel 39
Figura 5-6. Imágenes patrón para el cálculo del coeficiente de correlación 40
Figura 5-7. Para cada color de semáforo, a la izquierda imagen real y a la derecha imagen filtrada 40
Figura 5-8. Modelo del perceptrón 44
Figura 5-9. Procedimiento de clasificación de imágenes 45
Figura 5-10. Representación en tres dimensiones de los patrones usados para entrenamiento 48
Figura 5-11. Representación en dos dimensiones (canales rojo y verde) de los semáforos patrón 49
Figura 5-12. Representación en dos dimensiones (canales azul y verde) de los semáforos patrón 49
Figura 5-13. Representación en dos dimensiones (canales azul y rojo) de los semáforos patrón 50
Figura 5-14. Representación de las superficies de separación 50
Figura 5-15. Canal rojo frente a canal verde con superficies de separación 51
Figura 5-16. Ventana de comandos en MATLAB 51
Figura 5-17. Comprobación de resultados 52

1 INTRODUCCIÓN

El campo científico-técnico se desarrolla y camina actualmente hacia la creación de sistemas cada vez más autónomos, más rápidos y más especializados. El interés de la visión artificial radica en reconstruir el ojo humano mediante un computador, dotando a este último de una capacidad sensorial que supone una tarea difícil ya que entran en juego elementos accesorios como sensores o algoritmos complejos. No obstante, a pesar de no ser una tarea simple, actualmente existe un gran interés por imitar de forma artificial los cinco sentidos del ser humano, destacando entre ellos el de más atractivo interés: la habilidad de ver.

La visión artificial se puede definir como una disciplina capaz de englobar todos los procesos que proporcionan ojos a un computador con el fin de extraer información del mundo físico a partir de una o varias imágenes e interpretar el contenido de sus escenas. De esta forma, se dota al computador de un sofisticado mecanismo de percepción, que permite responder a su entorno de una forma inteligente. [1]

Por tanto, se podría resumir que la visión consta de dos fases primordiales: la recepción y la interpretación de imágenes. La primera de ellas es un problema ya resuelto, en el que, en el caso de un sistema de visión inteligente, el *ojo* sería la cámara, escáner, telescopio, satélite, etc. que toma la imagen, ya sea instalado en el computador o externo al mismo y la *retina* un sensor que es sensible a la intensidad luminosa. Así que, en la visión artificial, el proceso aún no resuelto completamente es la interpretación de las imágenes que incluye el entendimiento de la imagen con la posterior actuación en consecuencia de lo que se entienda. [2]

En un principio se pensaba que la interpretación de las imágenes era tarea fácil, ya que cuando este problema fue planteado, se hacía con la idea de que los computadores eran muy inteligentes y potentes. Por ejemplo, un computador es capaz de resolver sistemas de ecuaciones diferenciales de manera más sencilla que los seres humanos. Esto es debido a que el computador es entrenado y enseñado a responder de una determinada forma ante casos como este. La base de esto está en que el ser humano conoce a la perfección cómo realizar estos procedimientos. Sin embargo, la visión es un sentido humano que se podría caracterizar como "automático", es decir, el ser humano no hace nada de forma activa para conseguir ver, por tanto, indagar en el "funcionamiento" del ojo humano no es tarea fácil. Más bien se trata de un objetivo ambicioso que se encuentra en una etapa primitiva.

Antes de proceder a la ejemplificación de los campos de aplicación a los que hace referencia este proyecto, es necesario partir del conocimiento básico del sistema de visión artificial:

En primer lugar, la información que llega al ojo humano se trata de intensidad luminosa que procede de la imagen captada del entorno. Esta luz es captada a través de la córnea y se enfoca en la retina. Posteriormente se trata en los dos fotorreceptores que presenta la retina: los conos y los bastones, que constituyen dos sistemas que operan a diferentes niveles de brillo. Los conos son los responsables de la percepción del color y los

bastones captan las diferentes intesidades luminosas, que desde ahí es trasladada al cerebro, donde se ejecuta la visión.

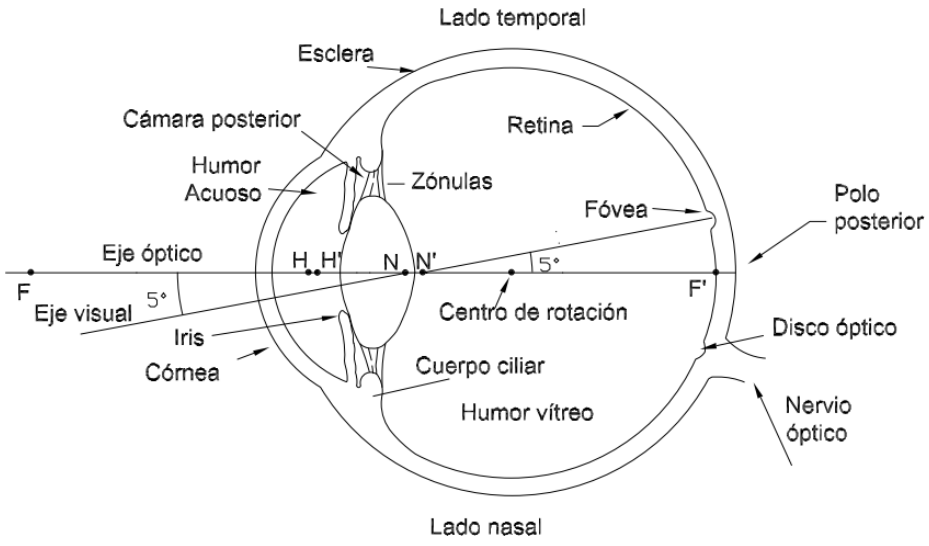


Figura 1-1. Sección transversal del ojo humano [3].

Todo lo que ocurre en el ojo humano es un proceso análogo a lo que ocurriría en el computador con la cámara de vídeo y el conjunto de sensores instalados. Para ello, aunque depende del proceso y la imagen que se esté tratando se conocen varias etapas:

- La primera fase consiste en la *adquisición de la imagen* mediante algún tipo de sensor. Para crear un buen sistema de visión, el paso más importante consiste en obtener una buena captura que pueda ser tratada posteriormente. Para ello es necesario una cámara de video o digital. Si dicha imagen no es apropiada, no se obtendrán resultados satisfactorios.

  Una vez capturada la imagen, esta queda constituida por una matriz de *píxeles* a partir del *muestreo* y la *cuantización*. Cada uno de estos *píxeles* ofrece información sobre una región de la imagen.

  El *muestreo* se realiza normalmente sobre una rejilla rectangular obteniendo la matriz de *píxeles.* Estos componentes de la matriz se corresponden con los niveles de gris de la imagen, o con diferentes colores si se da el caso.

Los siguientes pasos necesitan de un computador que almacene las imágenes y ejecute los algoritmos que requiere cada fase.

- La segunda etapa consiste en el *preprocesamiento* de las imágenes. Esto se realiza con el objetivo de facilitar las etapas posteriores, mediante filtrados y transformaciones geométricas de las figuras.
- A continuación, se realiza la *detección de bordes*, que en ocasiones es útil para extraer de la imagen bordes de los objetos que se estén buscando.
- La *segmentación* sirve para aislar los elementos que interesen de una escena con el fin de comprenderlas. Se trata de regiones que tienen píxeles que comparten algún patrón.
- De la segmentación extraída se *obtienen las características* necesarias para proceder a la posterior clasificación. Esto es necesario para diferenciar los objetos segmentados.

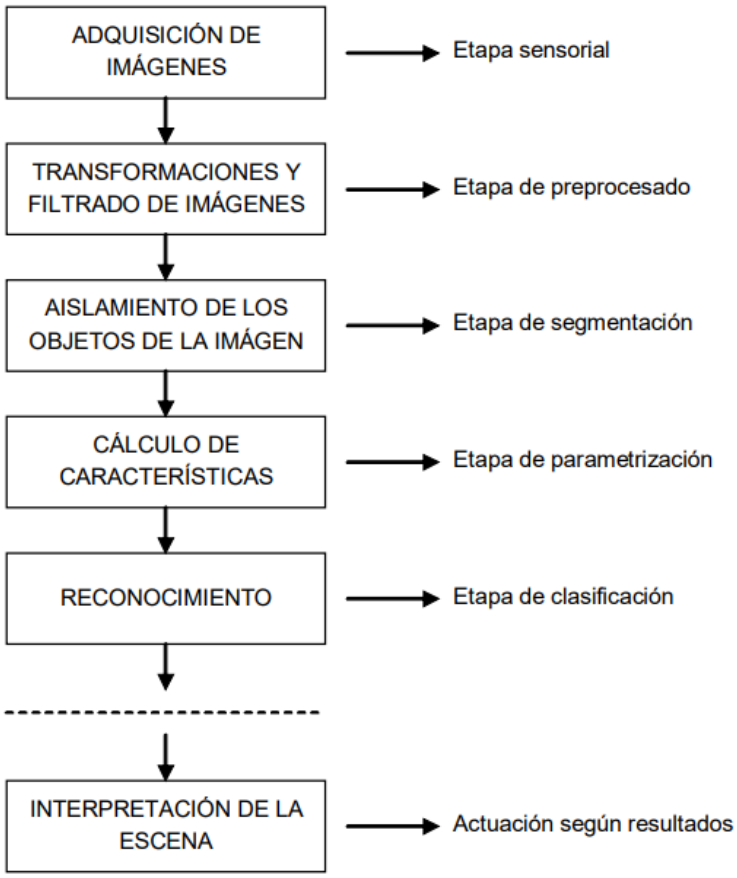


Figura 1-2. Etapas de un proceso de Visión Artificial [2].

En conclusión, se trata de un campo que está en continuo crecimiento y evolución. Esto es consecuencia del aumento de interés que supone la cantidad de aplicaciones que tiene y problemas que puede resolver, en función de las limitaciones que se presenten, entre las que se encuentra robótica, sistemas de inspección automático en industrias, análisis de imágenes en el campo de la medicina, biología, seguridad, meteorología, geología, cartografía, agricultura y el principal objeto de interés de este proyecto de fin de carrera: la navegación autónoma de vehículos.

En la actualidad, el automóvil es uno de los vehículos más usados por el ser humano, que se encuentra en la constante búsqueda de sistemas que hagan que estos vehículos sean cada vez más autónomos. Prueba de ello ha sido la evolución de sistemas de gestión de aparcamiento, teniendo sus orígenes en un "simple" sistema de asistencia al aparcamiento que usaban como aviso un pitido a medida que el vehículo se iba aproximando al obstáculo hasta llegar al completo estacionamiento a partir de visión periférica. Todo ello se implementa con el fin de facilitar al ser humano dicha tarea y para gestionar los espacios disponibles.

Este sistema consta de cámaras que proporcionan una visión de 360º para localizar los objetos y un conjunto de sensores que se han desarrollado hasta lograr un estacionamiento de vehículos completamente automatizado.

Sin embargo, a pesar de que la conducción autónoma es casi posible debido a dichos sensores instalados en los vehículos o software de asistencia a la conducción, visión nocturna, cambio de carril, adaptación de velocidad inteligente y sistemas anticolisión, etc. la tecnología todavía se encuentra lejos de encontrar una conducción completamente automatizada.

El objetivo que se presenta en este proyecto consiste en desarrollar un algoritmo capaz de detectar luces de semáforos a partir de visión artificial y, tras detectar, reconocer y clasificar los semáforos presentes.

Figura 1-3. Imágenes de semáforos

No solo se busca una correcta detección, sino la posterior clasificación para obtener qué porcentaje de acierto se obtiene de dicha clasificación dado que se trata de temas de seguridad vial, por tanto, se busca una detección eficaz a la vez que segura.

Por tanto, se desarrollará un sistema de detección, que utilizará métodos de umbralización y detección de formas mediante la Transformada de Hough, acompañado de dos mecanismos de clasificación con el fin de obtener los mejores resultados posibles. Estos clasificadores serán el Método Perceptrón y un clasificador por correlación.

El algoritmo desarrollado para la detección de semáforos será probado con 40 muestras que se caracterizan por ser muy distintas entre ellas, variadas en tamaño, en color, en situación y en cercanía.

Dicho esto, se han presentado diversas dificultades a la hora de construir el algoritmo, y es que al ser imágenes reales y no sintéticas intervienen un gran número de factores, como son fundamentalmente:

- La iluminación de cada una de las escenas que se presenta en el entorno del semáforo.
- La geometría del semáforo.
- El color y la textura de las superficies existentes.
- Distorsiones de la cámara.

La mayor parte de las técnicas empleadas en visión artificial para la detección de formas concretas se basan en identificar una o varias características comunes a todos los semáforos en este caso.

Lo realmente complicado de esto es que cada uno de los factores anteriormente mencionados, aportan en mayor o menor grado “deformaciones en la imagen” que hacen imposible la creación de un programa genérico que presente una solución común a todos los casos que puedan ser planteados, generando, por tanto, un problema en la extracción de características de cada imagen.

Aplicándolo a la detección de luces de semáforos, la idea inicial sería localizar los círculos, que es una característica que comparten todos, y, posteriormente realizar su análisis de color para obtener la clasificación completa.

Dicho círculo debería cumplir unas condiciones concretas. Aplicándolo al caso de que el sistema de visión está implementado en un vehículo, el círculo a localizar debería ser el más grande, ya que el sistema debe localizar el semáforo más cercano al coche en cuestión.

Por ejemplo, puede ocurrir que dicho sistema se encuentre en un caso como el que puede ser una calle de gran longitud, que presenta muchos semáforos y que se puede observar en la *Figura 1-4.*

Figura 1-4. Ejemplo de vía de tránsito caracterizada por la presencia de numerosos semáforos.

Para resolver este caso se empleará la transformada de Hough, que obtiene el círculo de mayor tamaño en la imagen. Pero ¿qué ocurre por ejemplo si encontramos una imagen en la que una señal de tráfico presente un círculo de tamaño superior al del semáforo de interés?

Figura 1-5. Escena en la que el círculo del semáforo no es el de mayor tamaño.

En este caso, el uso de la transformada de Hough como única solución no haría el efecto deseado, ya que en la escena se presenta un círculo de mayor tamaño, y como se ha indicado previamente, dicha transformada detectaría el círculo más grande, siendo este el de la señal de tráfico.

Es por eso, que se aplicarán otras técnicas de manera adicional para poner solución a este problema y conseguir que la única detección posible sea la del círculo de iluminación del semáforo.

La memoria se ha dividido en los siguientes capítulos:

- En el capítulo 2, se tratará el preprocesamiento de las imágenes de las que se extraerán las luces de los semáforos. Para ello se emplearán varios métodos:
    - Umbralización por color, con el fin de descartar zonas que no sean de interés.
    - Métodos morfológicos para obtener círculos completos.
- En el capítulo 3, se desarrollará la obtención de cada escena como una imagen binaria de bordes, donde la zona de interés será la luz del semáforo y se detectará dicha luz mediante la Transformada de Hough.
- En el capítulo 4, una vez que sean detectados los círculos, se crearán nuevas imágenes extrayendo de las anteriores solo la zona de interés detectada.
- Por último, en el capítulo 5 se realizará una clasificación completa de todos los semáforos detectados, proponiendo varios métodos para dicha discriminación.

2 PREPROCESAMIENTO DE LAS IMÁGENES

En este capítulo se desarrollará el modelo para el tratamiento previo que se realizará a las imágenes captadas de las luces de semáforos mediante simulaciones en MATLAB. Para ello se realizarán pruebas con 40 semáforos distintos, y repartidos en doce con iluminación verde, diecisiete con encendido en ámbar y once rojos.

El preprocesamiento constará de varias etapas, entre las que destacarán la separación de canales para cada una de las imágenes tratadas, una umbralización previa por colores y la aplicación de métodos morfológicos, más concretamente, dilatación y erosión.

2.1. Introducción al procesamiento de imágenes y su aplicación a semáforos

La finalidad del procesamiento de las imágenes consiste en generar una imagen de mayor calidad, reducida en ruido y, sobre todo, que simplifique las tareas posteriores que serán realizadas con dicha imagen. Por tanto, el procesamiento no pretende realizar o extraer características concretas de las imágenes de interés, sino preparar dicha imagen para facilitar la mencionada extracción de características.

2.1.1 Separación de la imagen en canales RGB

Existen numerosas razones por las que los sistemas basados en imágenes en escala de grises son mucho más utilizados. Estos motivos son que en las imágenes en color se produce un mayor volumen de datos, un aumento del costo del sistema, y que en el supuesto de que se quieran extraer solo características geométricas de la imagen, el color no aporta información relevante.

No obstante, en el actual proyecto se tratarán escenas 3D de imágenes reales y a color. Esto es debido a que el objetivo del problema planteado es poner solución a la detección de semáforos, y no solo semáforos, sino proceder a la posterior clasificación del color de estos, por tanto, la forma de representar esta realidad consiste en suponer las imágenes como matrices de dimensiones (X, Y) píxeles, además de una tercera dimensión que serán las tres capas de color R, G, B, siendo estas capas la roja, verde y azul de la imagen.

El modelo RGB es un modelo aditivo. Con este proceso es posible descomponer la imagen color en tres imágenes correspondientes a las capas mencionadas anteriormente. Esta representación se corresponde con una percepción física del color por parte del ojo humano.

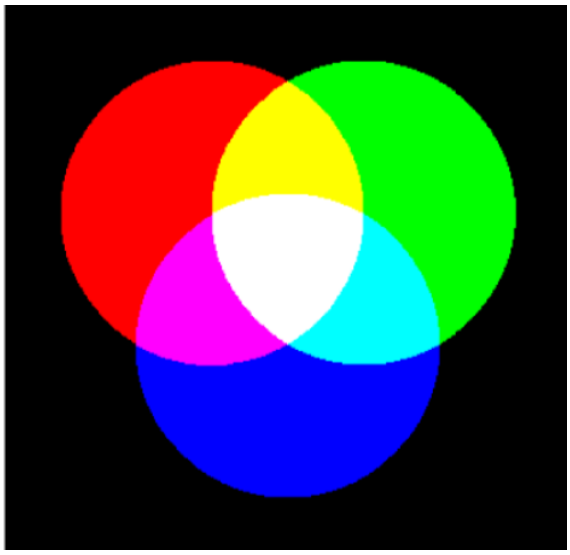


Figura 2-1. Modelo RGB

Será la intensidad luminosa la que determine el color propio de la imagen. Esta intensidad luminosa viene dada por la suma de las componentes roja, verde y azul.

En las capturas obtenidas, se representarán tres imágenes distintas para una sola toma de escena, siendo cada una de las imágenes el canal rojo, el canal verde y el canal azul de la misma.

El tratamiento de cada una de estas imágenes será necesario para la posterior umbralización por canales, en el que se procederá a la segmentación basada en el valor de los píxeles de la matriz que conforman la imagen. Al tratarse la imagen de una "matriz de matrices" es necesario conocer el valor de cada píxel en cada una de las capas.

En la separación de estas capas será posible detectar a simple vista los colores que caracterizan un semáforo. Esto es debido a que, por ejemplo, en un semáforo verde, la capa más representativa será la verde, donde se podrá observar el alto valor de los píxeles en la zona donde el semáforo está encendido con luz verde.

Figura 2-2. Representación de los canales RGB para un semáforo verde.

Tal y como se ha comentado anteriormente, es fácilmente observable en la imagen anterior que, en el caso de un semáforo verde, la capa verde es la que más información aporta, ya que los píxeles que se corresponden con la zona verde de la imagen poseen los valores más altos posibles.

En un caso ideal, el canal rojo y el canal azul estarían compuestos por matrices de tamaño el mismo que la imagen de interés y con todos sus píxeles con valor cero y que, por tanto, no aportarán ningún tipo de información, además de una capa verde que tuviera la zona del círculo con valores muy altos, idealmente a 255.

El canal rojo apenas es representativo, y, aunque no es completamente una matriz de ceros debido a las alteraciones que provoca el estar trabajando con imágenes reales, apenas aporta información.

En cuanto al canal azul, es destacable que los píxeles que se corresponden con la zona verde poseen valores muy altos. Esto es debido a que idealmente el color verde tiene unos valores de rojo y azul muy bajos y verdes muy altos. Sin embargo, tal y como se ha mencionado anteriormente, al tratarse de imágenes reales, estos resultados se alteran, provocando situaciones como la que se observan en la *Figura 2-2,* en la que el canal azul es muy elevado, pero no al nivel del canal verde.

Esto es observable haciendo uso de la herramienta *imtool* de MATLAB, donde podemos ver la información individual de cada píxel en cada canal.

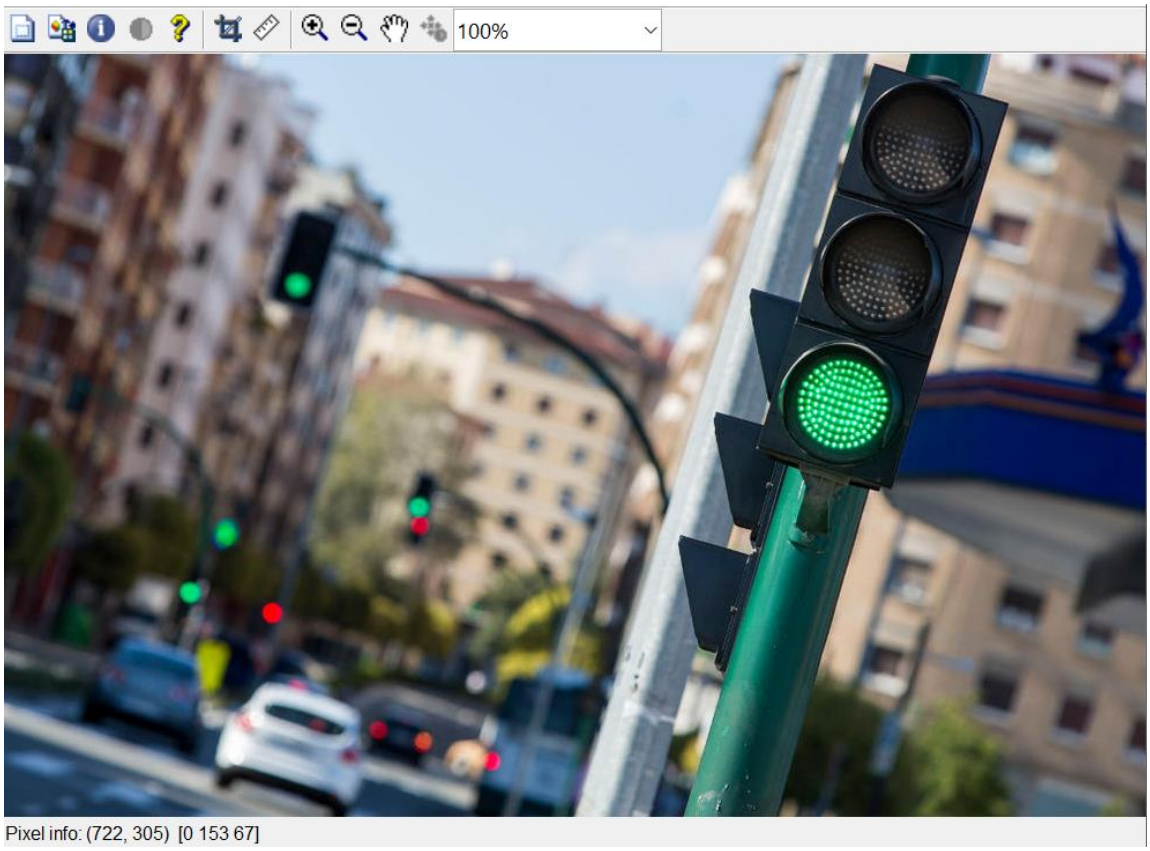


Figura 2-3. Imagen obtenida con la herramienta *imtool* de MATLAB para inspección.

En la esquina inferior izquierda se puede observar que, situando el cursor encima de la zona de interés del semáforo, se obtiene un valor muy bajo en la capa roja, un valor muy alto en la capa verde y un valor medio en la capa azul.

Llegados a este punto es necesario destacar la problemática que supone el brillo que presentan las luces del semáforo. Esto hace que se altere considerablemente la toma de muestras, ya que, si se observa con claridad la figura anterior, se percibe que, debido a la alta intensidad luminosa de los píxeles, pueden llegar a ser casi blancos, ocasionando de esta forma un incremento considerable del valor de cada canal.

En el caso del resto de semáforos ocurre lo mismo. A continuación, se mostrarán los canales correspondientes a una imagen donde la zona de interés es el encendido rojo de un semáforo y otro ámbar.

Figura 2-4. Canales RGB (rojo, verde y azul de izquierda a derecha) de un semáforo rojo.

En la imagen anterior se observa que en este caso la luz del semáforo tiene muy altos valores de intensidad en el canal rojo, y tal y como debería ser en el caso óptimo, valores casi nulos en los canales verde y azul, ya que se observa que, en estas capas, los círculos del semáforo son prácticamente negros, es decir, píxeles con valor cero. No obstante, se siguen dando valores medios de cada canal en la escena ocasionados por tratarse de imágenes reales, pero es observable a simple vista que la parte más representativa de la imagen es el círculo iluminado del semáforo en el canal rojo.

Usando de nuevo la herramienta *imtool,* se verifican estos resultados:

Figura 2-5. *imtool* de un semáforo rojo.

En la imagen se puede observar que en la esquina inferior izquierda aparece la información del píxel donde esté situado el cursor, siendo, en este caso, en un píxel correspondiente a la luz roja del semáforo.

Se aprecia que el canal rojo presenta un valor de intensidad de 233, y 2 y 8 para los canales verde y azul respectivamente. Estos datos no suelen ser comunes, ya que, como se ha explicado en varias ocasiones, con las condiciones de iluminación que aporta el entorno de la imagen, no se producen resultados tan concluyentes.

Además, también se ha especificado que, al tratarse de luces, suelen darse brillos en dichos píxeles que incrementan de forma considerada el valor de cada canal, obteniéndose valores de intensidad en cada canal por encima de 255.

Por último, es importante destacar qué ocurre con las imágenes ambar. En condiciones ideales, el color ámbar presenta unos valores normalizados RGB de (255, 191, 0) para cada canal respectivamente. Sin embargo, este color del semáforo es el que más problemas puede plantear cuando se desee extraer información de cada una de sus capas. Esto es debido a que las condiciones de iluminación afectan mucho a los niveles de intensidad que se establecen en las imágenes. Pudiendo encontrarse semáforos ámbar con unos niveles de intensidad luminosa en cada canal que hacen que se asemeje a un color amarillo "chillón", y otros casos en los que se corresponden más con tonalidades anaranjadas.

Figura 2-6. Dos imágenes de semáforos ámbar, tomadas en condiciones distintas de iluminación

Si analizamos por capas una imagen ámbar, se observa que, tal y como se expresa en los valores normalizados, se trata de valores de intensidad muy altos en el canal rojo, valores medios en el canal verde y muy bajos en el azul. Pero el hecho de que el valor de rojo sea muy alto, puede traer problemas consigo a la hora de distinguir entre imágenes de semáforos rojos y semáforos ámbar.

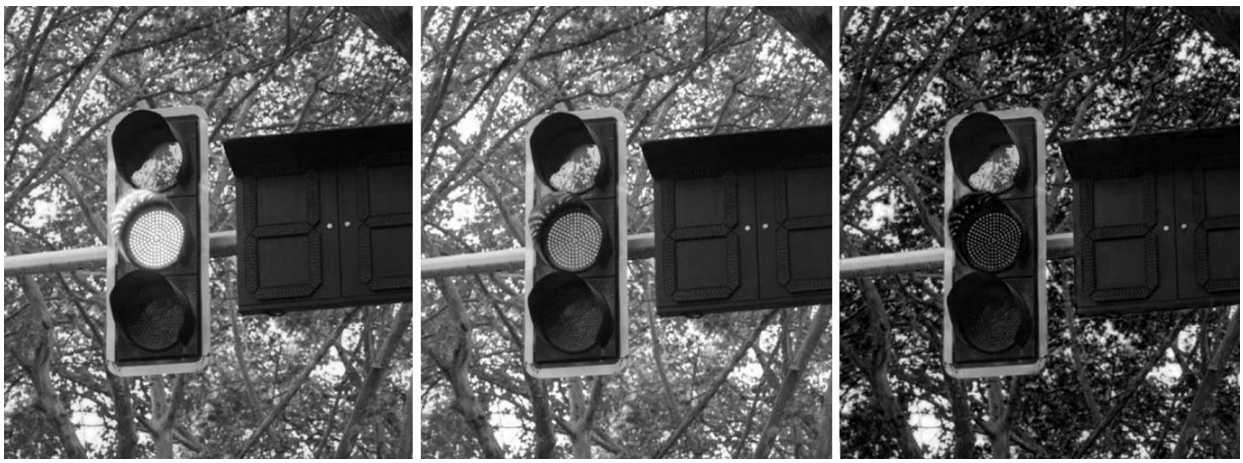


Figura 2-7. Separación en canales RGB de un semáforo ámbar.

Es por eso por lo que se concluye que la segmentación de las luces de semáforo a partir de umbralización por color, no dará resultados beneficiosos a la hora de la detección que se busca, por eso será necesario recurrir, de forma adicional a modelos geométricos que cumplan con un gran porcentaje de beneficio el modelo buscado.

2.1.2 Umbralización por color

La umbralización de imágenes es uno de los métodos que abarca la segmentación de imágenes, y es el proceso por el cual se busca un umbral óptimo que permita distinguir en una imagen los objetos del fondo. [4]

La selección de un umbral óptimo es crucial para poder realizar una buena segmentación.

De manera general, la umbralización por color a partir de umbrales globales suele ser usada en imágenes sintéticas, cuando la definición entre los objetos y el fondo es clara y cuando las condiciones de iluminación son uniformes y por tanto adecuadas a la aplicación de este método.

Para ello se suele usar un histograma, abarcando diferentes propiedades de este, como valles, picos y curvaturas.

En el presente proyecto, y tal y cómo se ha comentado con anterioridad este método no sería óptimo para calcular de forma exacta la localización del círculo en la imagen del semáforo en escena. Esto es debido a que se trabajará con imágenes reales, con todas las propiedades contrarias a las que se han indicado en el primer párrafo. No se trata de imágenes en las que se distinga a la perfección el objeto en cuestión del fondo, y, además, en casos reales no se tratará nunca de imágenes con una iluminación de escena regular.

A estos inconvenientes propuestos es necesario añadir el hecho de que se tratarán imágenes RGB, por lo que es necesario tratar cada una de las capas por separado. Muestra de ello es el caso que se muestra a continuación:

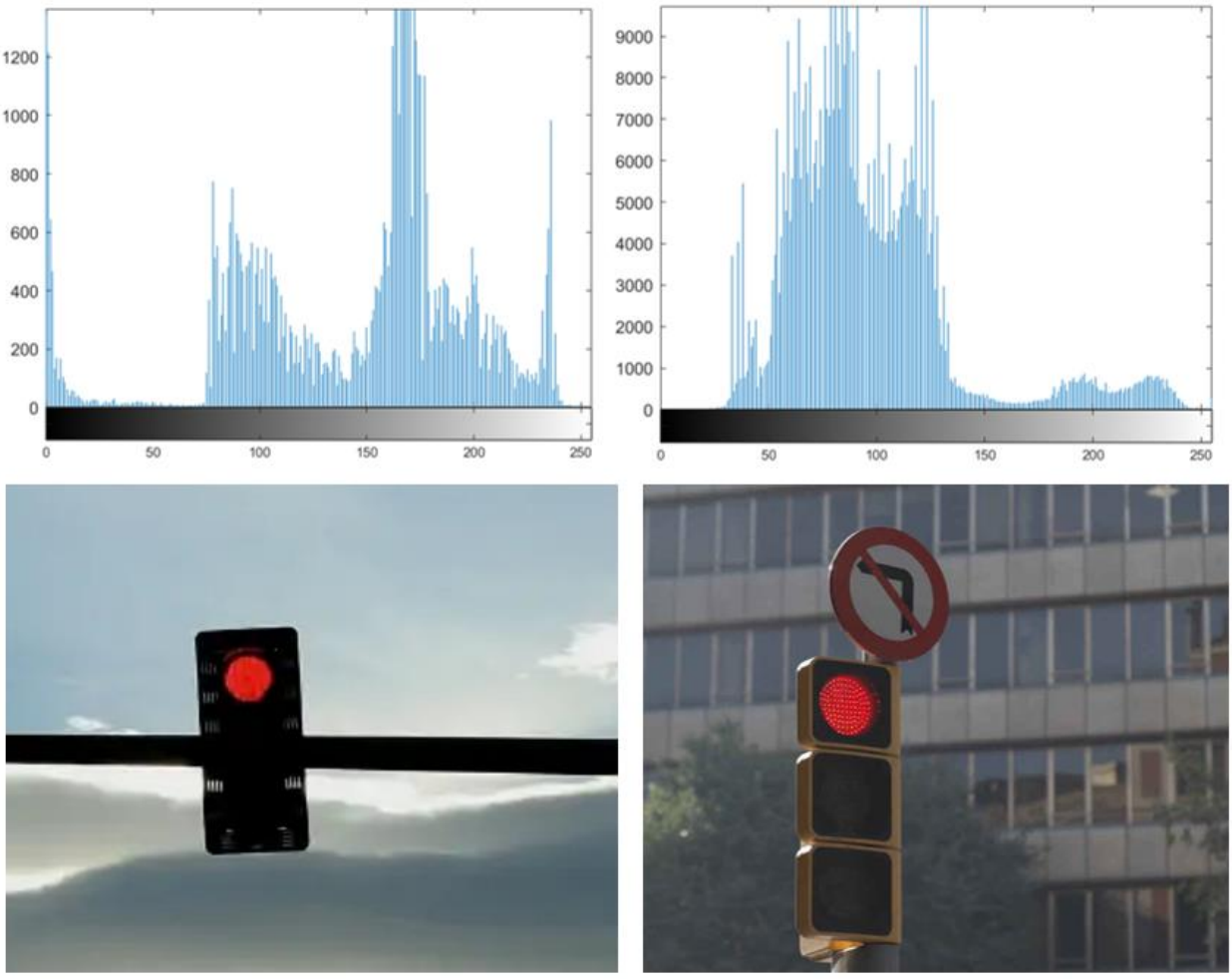


Figura 2-8. Representación de histogramas del canal rojo para dos semáforos distintos.

En las figuras adjuntas se puede observar que, para dos semáforos en condiciones de iluminación aparentemente similares, los histogramas obtenidos para el canal rojo del conjunto RGB son muy diferentes, es decir, en el primero de ellos se producen dos valles, uno en torno al valor del nivel de gris 130 y otro en torno a 230. Sin embargo, en el semáforo de la derecha, solo se produce un valle significativo, pero también en torno a 130, luego a pesar de ser histogramas notablemente diferentes, tienen sus valores más importantes extraíbles en el mismo punto aproximadamente. Se podría hacer un análisis más exhaustivo usando para cada capa un nivel de intensidad obtenido con el comando *graythresh*.

Haciendo esto solo para la capa roja, que se ha considerado la más representativa como parece lógico para el semáforo rojo, para el caso del semáforo mostrado en la izquierda, el umbral óptimo propuesto por este método es 120, y para el semáforo de la derecha 144, pudiendo concluir que, a pesar de que se trata de valores relativamente cercanos, no son óptimos en cuanto a la segmentación, puesto que lo que se busca es la máxima exactitud posible y la máxima precisión en el momento de obtener resultados.

Aunque en imágenes sintéticas no sería necesario, al tratarse de imágenes reales es necesario también marcar umbrales del resto de canales, aunque el semáforo en cuestión sea el rojo. Esto es debido a muchos factores. El más destacable de ellos es que los semáforos tratados y, en general la mayoría de ellos mostrarán como fondo un cielo que por lo general será de tonalidades azules, o árboles con componentes elevadas del canal verde, entre otros muchos.

Si realizamos el análisis de estas dos imágenes en los canales verde y azul, podemos observar que, para la primera imagen, el umbral óptimo de la capa verde se trata de un nivel de intensidad de 134 y para la segunda imagen (la de la derecha) en torno a 150.

Por último, en cuanto al canal azul, se puede concluir que para la imagen de la izquierda el umbral óptimo es un nivel de intensidad de 139 y para la imagen de la derecha de 156.

Estos resultados permiten determinar que será necesario para la umbralización establecer valores mínimos y máximos, para cada canal en cada una de las imágenes, debido a la gran variedad de condiciones y elementos en escena.

Esto, como se ha comentado con anterioridad, es necesario porque hay que discriminar y segmentar la imagen lo máximo posible a fin de conseguir una segmentación lo más exacta alcanzable, por tanto, tras discriminar solo zonas que cumplan con los valores establecidos de umbralización por color, se eliminarán muchos objetos de las imágenes que no resultan de interés, como las ya mencionadas en otros apartados señales de tráfico, que suponen un problema en cuanto a la segmentación con métodos geométricos.

Es por eso por lo que se han definido los siguientes valores para la umbralización:

Tabla 2-1. Umbrales óptimos para segmentación

	Canal rojo	**Canal verde**	**Canal azul**
Semáforo rojo	(185 - 255)	(30 - 100)	(50 - 105)
Semáforo ambar	(180 - 255)	(55 - 200)	(0 - 100)
Semáforo verde	(4 - 150)	(180 - 255)	(60 - 255)

Estos valores máximos y mínimos para cada canal han sido obtenidos mediante el estudio y la extracción del umbral óptimo de varias imágenes en diferentes condiciones de iluminación para obtener unos umbrales relativamente variados y restrictivos.

Para ello se ha partido de valores normalizados de los colores rojo, ambar y verde, teniendo en cuenta que se tratan de colores brillantes, debido a que proceden de luces que iluminan artificialmente a los círculos del semáforo, luego se aplican valores con intensidad luminosa muy alta.

Estos valores son los que se muestran a continuación:

Tabla 2-2. Valores normalizados de los canales RGB para cada color del semáforo

	Canal rojo	**Canal verde**	**Canal azul**
Semáforo rojo	255	0	0
Semáforo ambar	255	191	0
Semáforo verde	0	255	0

Haciendo analogía entre unos datos y otros, es destacable que en ambas tablas se establecen valores muy altos del canal rojo para los semáforos rojo y ámbar. Además, se observa que el canal verde para el semáforo ámbar se establece en valores intermedios, y que para el semáforo verde el canal verde presenta valores muy altos.

Sin embargo, es fácilmente observable que, a pesar de que en valores normalizados el canal azul es nulo para los colores de semáforos, la realidad es que esta componente nunca es nula, y no solo se trata de que no es nula, sino que además para semáforos verdes esta componente es muy alta y para el resto de los semáforos presenta valores medios.

Esto es debido a que todas las luces de semáforos en condiciones reales presentan mucho brillo, lo cual proporciona que la mayoría de ellas presenten un gran porcentaje de píxeles blancos.

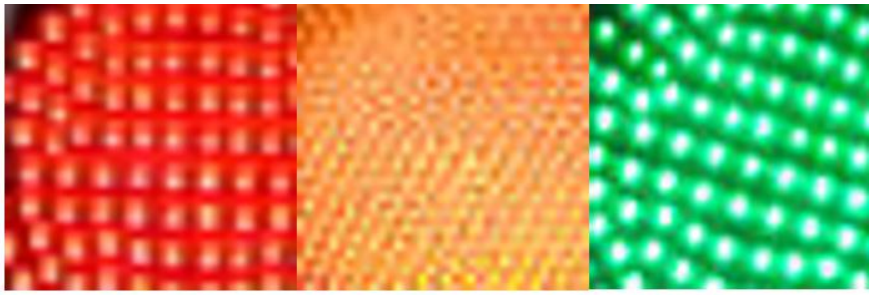


Figura 2-9. Demostración de píxeles blancos en semáforos

Además, en el caso del semáforo verde, se hace más pronunciada la presencia de componentes del canal azul debido a la variedad de tonalidades que se presentan según las condiciones de iluminación de la escena.

Figura 2-10. Diferentes tonalidades de semáforos verdes extraídas de imágenes reales.

En la segunda imagen se puede observar que el nivel de intensidad en la capa azul será elevado, ya que no se trata de verde como tal, sino verde-turquesa, que eleva notablemente la componente azul.

Además, también será determinante la cercanía con la que se haya capturado la imagen, por ejemplo, en el semáforo de la derecha de la *Figura 2-10*, se pueden observar con claridad los píxeles del semáforo en cuestión. Sin embargo, en las dos primeras imágenes se observan luces difuminadas, muy probablemente ocasionado por la distancia a la que ha sido tomada la foto.

Una vez realizada la segmentación a partir de los umbrales óptimos definidos con la intención de localizar solo las luces de los semáforos, el procedimiento restante se realizará con el tratamiento de las imágenes obtenidas de dicha segmentación tratando imágenes que solo tengan como píxeles significativos aquellos que cumplan los valores establecidos.

En resumen, las imágenes resultantes presentarán solo como píxeles no nulos aquellos que presenten para cada canal las componentes de intensidad luminosa que se correspondan con los valores establecidos para los semáforos rojo, ámbar y verde.

Para la obtención de estos resultados se ejecutará una función que devolverá como respuesta una imagen que representará aquellos píxeles que se correspondan con los colores del semáforo rojo, otra imagen para el semáforo verde y una última para la obtención del semáforo ámbar. Estas imágenes se denominarán *luz_roja, luz_verde* y *luz_ambar* respectivamente.

Por ejemplo, para un semáforo rojo, la situación óptima sería que en *luz_roja* aparecieran los píxeles blancos que representen la luz roja del semáforo, y las otras dos imágenes como matrices de píxeles con valor cero.

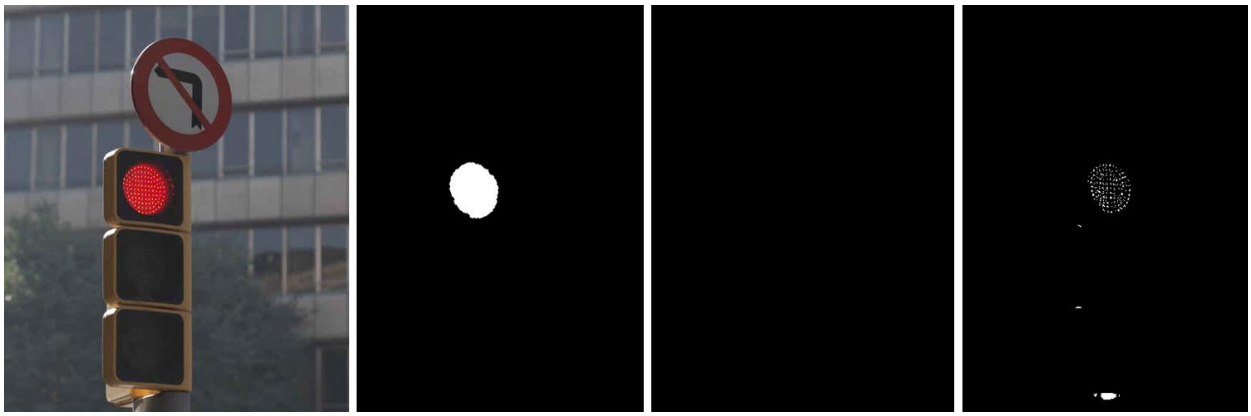


Figura 2-11. Semáforo rojo segmentado mediante umbralización por color

En la imagen anterior se puede observar que, de izquierda a derecha, en primer lugar, se muestra el semáforo real, con encendido en rojo, y las tres imágenes restantes representan la umbralización para cada uno de los semáforos establecidos, rojo, verde y ambar.

Se observa un problema que estará presente para todos los semáforos ámbar y rojo, y es que las componentes de cada uno son muy parecidas, dando lugar a confusión en la detección. En la imagen que se muestra no se trata de un problema muy significativo puesto que la segmentación para ámbar a pesar de tener que presentar una imagen completamente negra, muestra muy pocos píxeles blancos, por lo que no se considera una imagen mal tratada.

Además, se puede observar cómo se elimina todo el fondo, lo que puede hacer concluir que se trata de una buena umbralización.

En la mayoría de los casos no va a ocurrir que se vaya a eliminar el fondo solo con la umbralización por color. Esto es debido a que el entorno puede presentar una escena que muestre píxeles que tengan componentes que cumplan con los valores de umbralización establecidas.

A continuación, se mostrará una imagen donde la zona de interés es el encendido de un semáforo verde. Además de eso, se tiene de fondo un cielo azul cuya intensidad luminosa se corresponde con la segmentación que se ha establecido para los semáforos verdes.

Se muestra, de izquierda a derecha: la imagen real, la umbralización con valores para semáforos rojos, semáforos verdes, y por último semáforos ámbar.

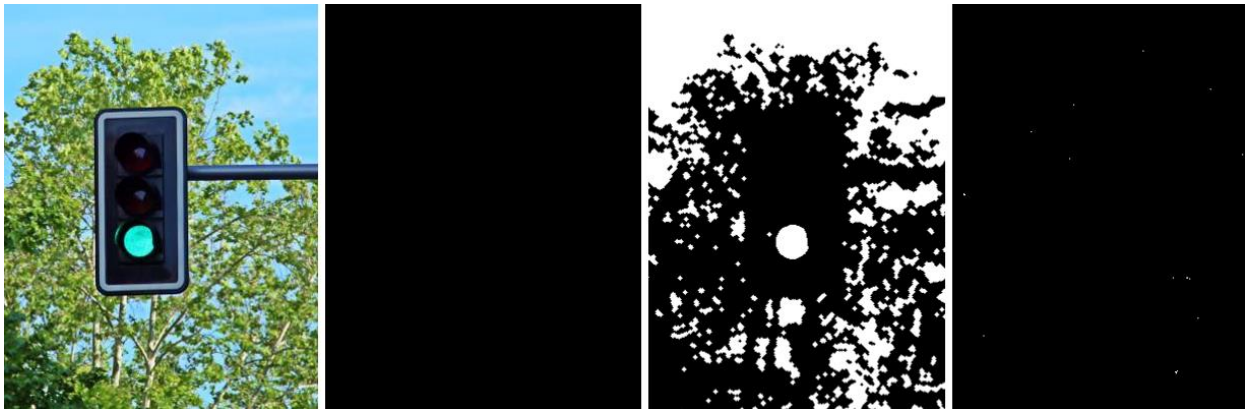


Figura 2-12. Semáforo verde segmentado mediante umbralización por color

Tal y como se ha indicado, como el cielo presenta componentes que cumplen los umbrales establecidos para segmentar el semáforo verde, aparecen por tanto más píxeles blancos de los que sería óptimo. Sin embargo, es observable que el círculo del semáforo, que es la zona de interés, es fácilmente distinguible.

Por último, en el semáforo ambar se obtienen resultados similares:

Figura 2-13. Semáforo ámbar segmentado mediante umbralización por color

Una vez llegados a este punto, lo siguiente será eliminar los restos que aparecen en las matrices de imágenes que deberían ser completamente negras. Para ello se utilizarán métodos morfológicos. En concreto se aplicará acreción seguida de erosión.

2.1.3 Métodos morfológicos

Los métodos morfológicos se usan para simplificar imágenes y conservan las principales características de forma de los objetos. En este caso será usada como pre-procesamiento de las imágenes con el fin de suprimir los ruidos que aparecen a causa de que no se produce una perfecta umbralización por color.

Para ello, en primer lugar, se realizará una dilatación, con el fin de rellenar y preparar mejor los círculos de los semáforos para su posterior detección con la transformada de Hough, y, a continuación, será aplicada una erosión con el propósito de eliminar los puntos sobrantes que no son de interés en la imagen.

La erosión consiste en la eliminación de todos los píxeles del objeto en cuyo vecindario haya al menos un píxel que pertenezca al fondo.

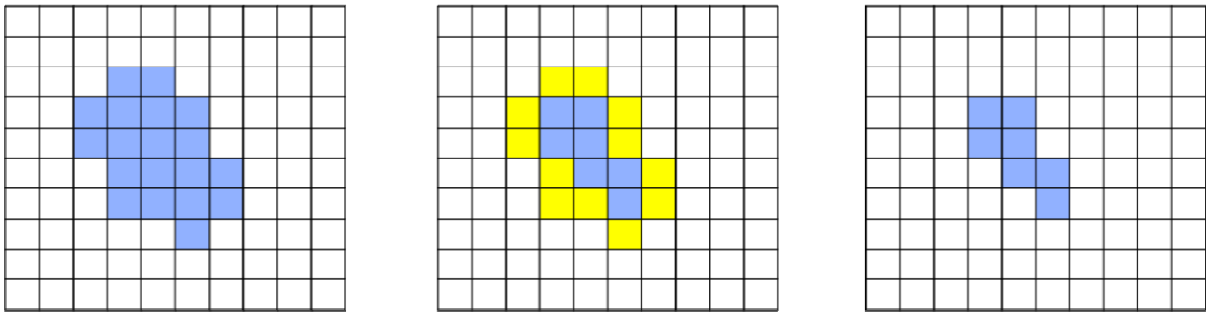


Figura 2-14. Representación del proceso de erosión

La dilatación consiste en añadir todos los píxeles de fondo en cuyo vecindario exista al menos un píxel perteneciente al objeto. Su efecto es aumentar el tamaño de los objetos o eliminar círculos muy pequeños.

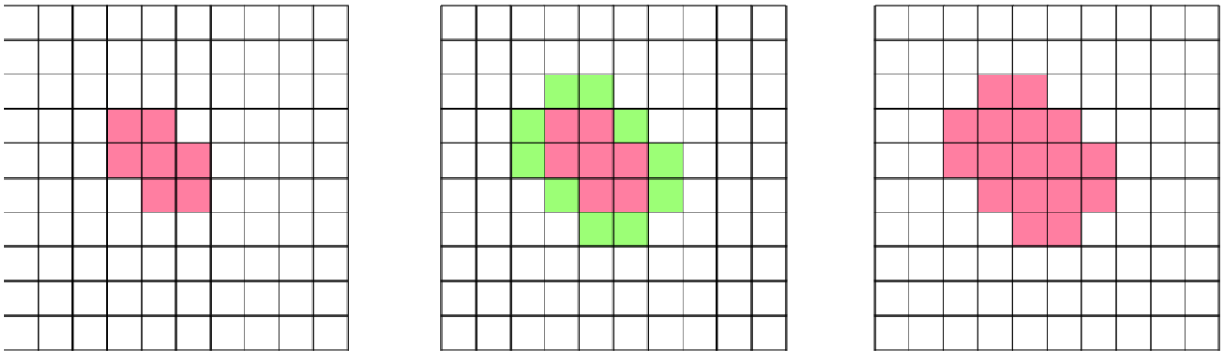


Figura 2-15. Representación del proceso de dilatación

La acreción en MATLAB se realizará con la orden *imdilate* y la erosión con *imerode.* Ambas necesitan recibir como entrada, además de la imagen que se desea dilatar, un *cincel*, que, para todos los casos será establecido como una esfera. El parámetro que varía será el tamaño del cincel tras varias pruebas.

El *cincel* anteriormente mencionado (en MATLAB *strel*) representa un elemento estructurante morfológicamente plano que se usa para la dilatación y la erosión.

"Un elemento estructurante plano es un entorno que asume valores binarios, que puede ser 2D o multidimensional y en el que los píxeles con valor *true* se incluyen en el cómputo morfológico, en tanto que los píxeles denominados con el valor *false*, no. El píxel central del elemento estructurante, llamado *origen,* identifica el píxel de la imagen que se está procesando". [5]

La combinación de dilatación seguida de erosión da lugar a un proceso de apertura cuyo fin es eliminar elementos salientes, conexiones finas o suavizar los contornos de un objeto. [6]

En el caso de tratamiento de semáforos, interesa la aplicación de una apertura porque lo que se busca es eliminar restos que aportan ruido a la imagen además de suavizar el contorno del círculo que va a ser tratado posteriormente, es decir, tratar lo máximo posible de conseguir una superficie totalmente circular con el fin de facilitar el posterior proceso de detección de círculos a partir de la transformada de Hough. Un ejemplo de aplicación de apertura en una imagen binaria es el que se puede observar en la *Figura 2-16*, tal y como se muestra a continuación:

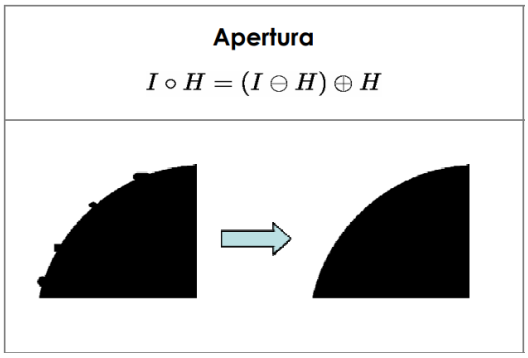


Figura 2-16. Ejemplo de aplicación de apertura a una imagen binaria.

Tras la aplicación de la apertura, se puede observar en la imagen anterior cómo se eliminan los restos sobrantes del borde.

Del apartado anterior, los resultados que se han obtenido son adicionalmente a la imagen original, una imagen segmentada según cada color del semáforo, es decir, una imagen donde se segmenta solo el color verde del semáforo, otra imagen donde se segmenta el color rojo del semáforo y por último una donde se segmenta el color ámbar del semáforo.

Como ya se ha explicado esto se realiza para ser lo más exactos posibles en cuanto a la segmentación definitiva de la luz del semáforo. Podría simplificarse el proceso, pero lo idóneo para conseguir la máxima precisión es ser concretos y extensos en cuanto a la segmentación.

Por tanto, los métodos morfológicos serán aplicados a cada una de las imágenes anteriormente descritas. No obstante, realmente lo que interesa será la aplicación del método a la imagen de segmentación verde, cuando el semáforo sea verde, a la roja cuando el semáforo sea rojo y a la imagen segmentada en ámbar cuando el semáforo sea ámbar, pero como uno de los requisitos buscados en el código es hacerlo completamente general, se aplicará en todos.

Para demostrar cómo se comporta este método en el tratamiento de los semáforos usados en el presente proyecto, a continuación, se muestran algunos ejemplos donde se comprueba el funcionamiento de este:

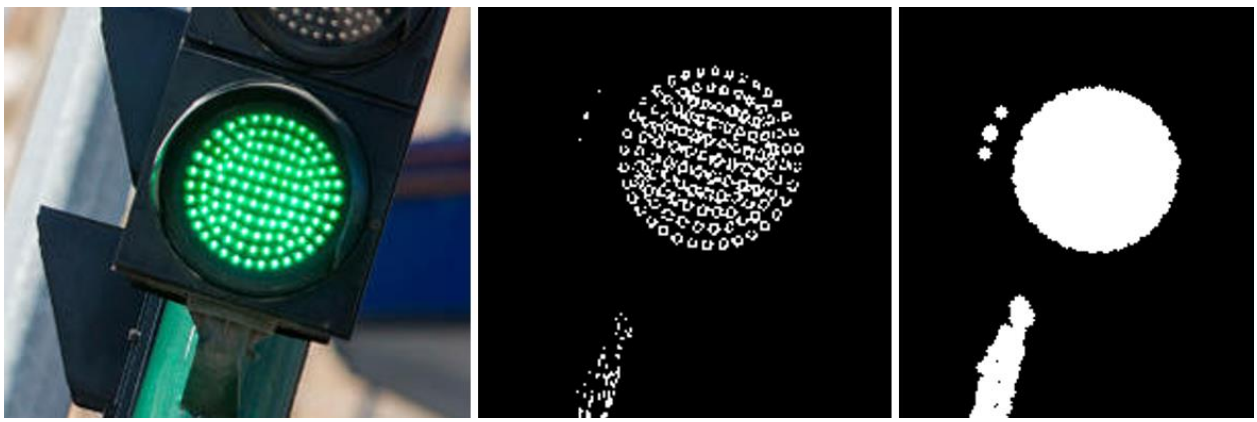


Figura 2-17. Aplicación de métodos morfológicos a semáforo verde

La imagen anterior es un claro ejemplo del objetivo que presenta la aplicación de métodos morfológicos en imágenes. En la primera imagen empezando por la izquierda, se representa la imagen original, seguida de la imagen binaria en la que la umbralización no detecta completamente el círculo, sino píxeles concretos del mismo y, por último, la aplicación de la apertura que, a pesar de que aumenta suavemente el tamaño del ruido, cumple su función, que es rellenar el círculo de la manera más perfecta posible.

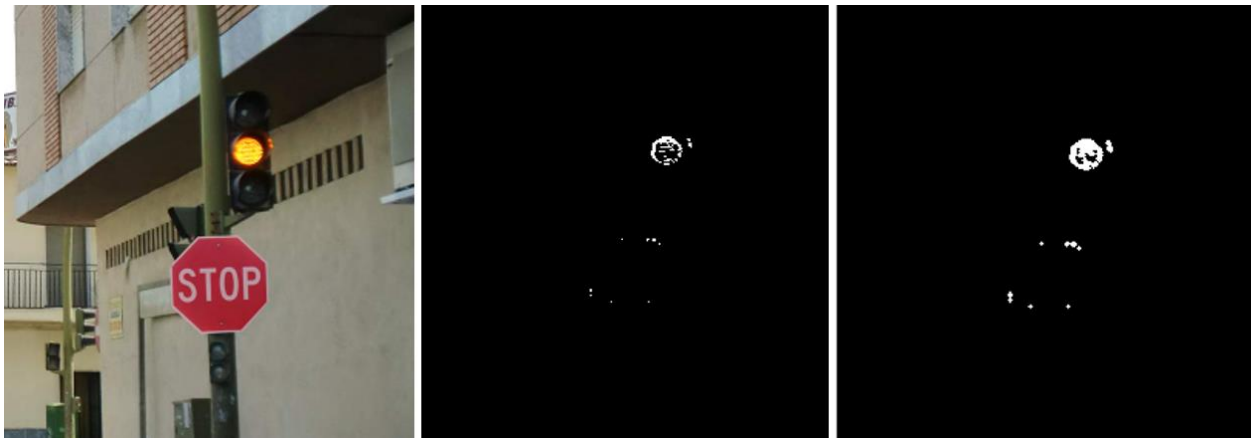


Figura 2-18. Aplicación de métodos morfológicos a semáforo ámbar

En el caso mostrado en la *Figura 2-18* se puede observar que el método no cumple a la perfección su función, puesto que se pueden apreciar huecos en el interior de la luz del semáforo.

Al tratarse de imágenes reales este es un caso esperado, es decir, no se busca encontrar en cada caso la solución mostrando un círculo completamente perfecto, sin huecos ni ruido en sus bordes.

No obstante, se puede apreciar que el contorno del círculo está bien definido, que es lo que realmente se pretende para una buena localización posterior de dicha forma.

En apartados anteriores se trataba la posibilidad de que una señal de tráfico de color rojo, "falseara" los resultados ocasionando que el semáforo rojo no fuera detectado debido a que el mayor círculo se diera en dicha señal. Puesto que la segmentación actual que se está realizando se corresponde con valores de intensidad luminosa muy elevados debido al brillo que presentan las luces de semáforos, este problema desaparece por el motivo que se explica a continuación:

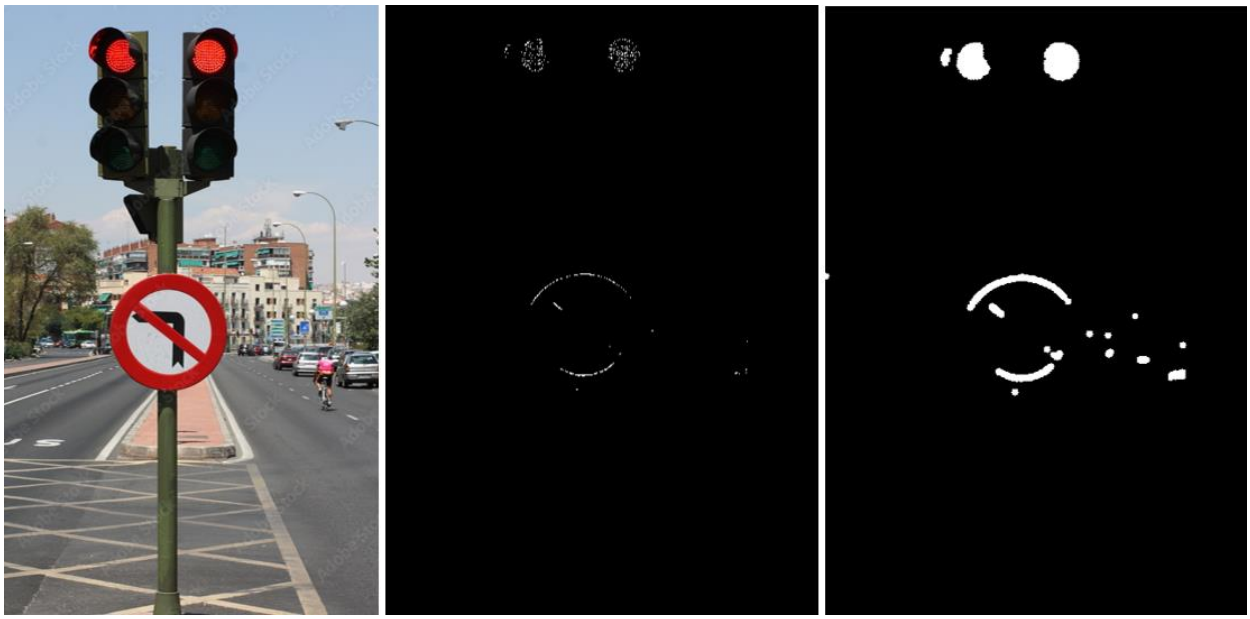


Figura 2-19. Aplicación de métodos morfológicos a semáforo rojo

Se puede observar que en la segmentación por umbralización se elimina gran parte del círculo de la señal de tráfico. Esto es, volviendo a recalcar, debido a que se trata de imágenes reales, y la componente de cada canal no es constante a lo largo de toda la zona roja de la señal de tráfico, por eso aparece una zona representada que se incrementa al aplicar la apertura.

Se concluye que, de la *Figura 2-19* también se obtienen resultados exitosos gracias a la combinación de segmentación por umbralización y aplicación de métodos morfológicos, pudiéndose observar en la imagen de la derecha, la representación completa del círculo del semáforo encendido.

2.1.4 Representación de la segmentación

El siguiente paso, una vez han sido descartadas todas las zonas de la imagen que no cumplan los umbrales establecidos para cada color del semáforo, será representar en una imagen RGB la unión de las tres segmentaciones previas realizadas, es decir, como se tendrán tres imágenes con los colores de los semáforos segmentados, se unirán los tres en una sola imagen con el fin de realizar los pasos siguientes en función de dicha imagen.

Al tratarse de imágenes binarias, que son imágenes en las que las zonas de interés están a 1 y las de fondo a 0, el procedimiento para extraer solo las zonas de interés será multiplicar la suma de esas imágenes por la imagen original.

Para demostrar qué se persigue con este procedimiento se muestran a continuación figuras de algunos resultados obtenidos.

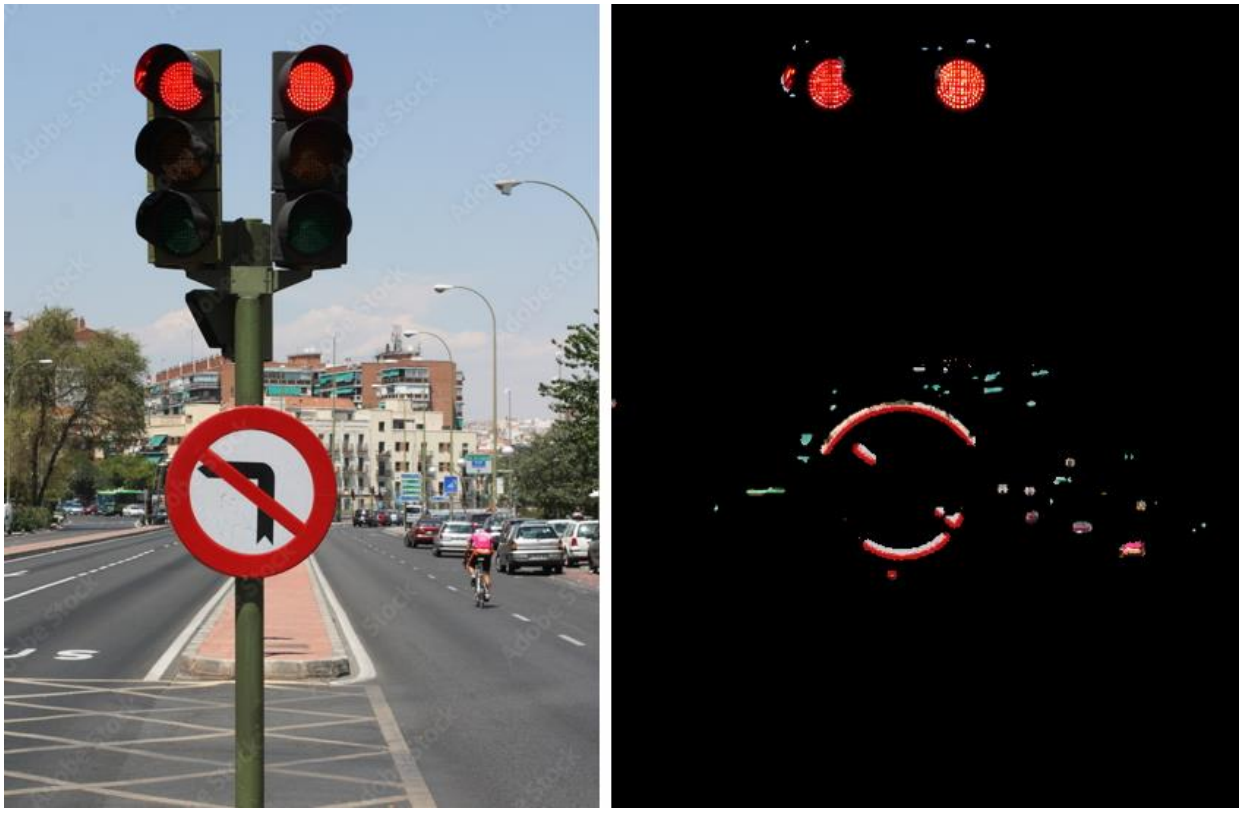


Figura 2-20. Segmentación completa de semáforo rojo

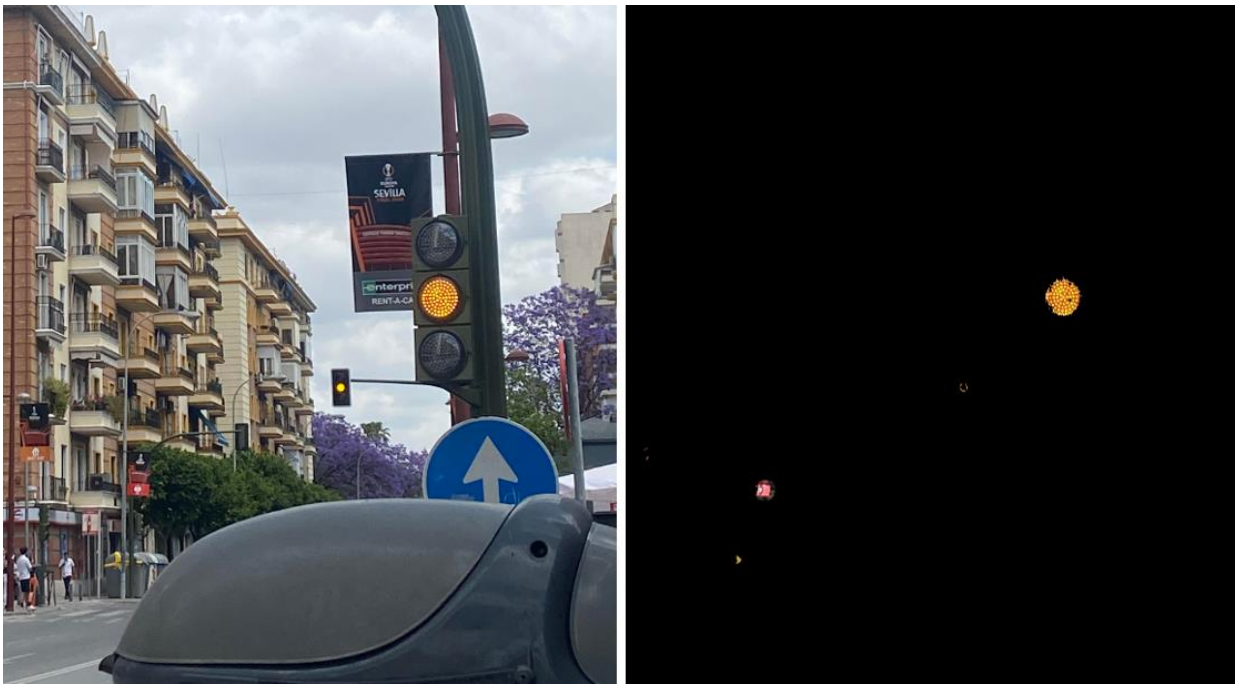


Figura 2-21. Segmentación completa de semáforo ámbar

Figura 2-22. Segmentación completa de semáforo verde

En la *Figura 2-22* se puede observar cómo, en la escena además de una luz de semáforo verde, aparece también una flecha que cumple con los umbrales establecidos para semáforo ámbar. Esto hace que en la segmentación sea visible dicha zona, la cual no ocasionará múltiples problemas debido a que no se trata de un objeto de imagen con forma circular.

En el resto de las imágenes adjuntadas, aunque aparece cierto ruido difícil de eliminar, se puede apreciar que la zona de interés, que es el círculo del semáforo, se encuentra en un primer plano, como el círculo más grande de la escena.

2.1.5 Conclusiones del capítulo

El procedimiento realizado en el actual capítulo ha permitido la obtención de una versión simplificada en RGB de las imágenes originales aportadas para la comprobación del buen funcionamiento del método.

Con esto se llega a una matriz de ceros, es decir, una imagen negra, donde los únicos píxeles con valor distinto de cero son aquellos que se corresponden con las zonas que presentan unos niveles de intensidad de acuerdo con los semáforos reales.

Es por todo ello que el procedimiento descrito se considera un preprocesamiento de la imagen, puesto que se prepara para el posterior tratamiento geométrico.

Al tratarse de imágenes reales, tomadas en diferentes circunstancias y condiciones de escena, se ha descartado un posible tratamiento previo como podría ser el suavizado. Esto es debido a que las operaciones de suavizado tienen por objeto reducir el ruido a costa de difuminar levemente la imagen, y, en cualquier caso, lo que se pretende para la detección posterior del círculo, es unos bordes concretos y bien diferenciados del fondo de la imagen.

En algunas circunstancias suele ser conveniente plantear un realce sobre el entorno de vecindad de cada píxel, que consiste en redistribuir los niveles de grises de la imagen. A pesar de que la finalidad de este procedimiento se considera adecuar la imagen original para una determinada aplicación, este método también ha sido descartado en el pre-procesamiento debido a que lo que se consigue con este es dar una medida de brillo medio y una estimación del contraste. Al tratarse, como ya se ha explicado, de imágenes que se caracterizan por los elevados niveles de intensidad que se dan en las luces de los semáforos, que son las zonas de interés, no sería óptimo ni regular aplicar este realce.

En la *Figura 2-23,* se muestra un ejemplo de imagen a la que le ha sido aplicada un proceso de realce por igualación del histograma, pudiendo observar en él, cómo se produce una alteración completa del fondo, debido a que, aunque a simple vista no es claramente observable, en la imagen real el fondo presenta gran variedad de niveles de intensidad, y, por tanto, al aplicar un método de ecualización del histograma, esta variedad queda más reflejada provocando una distorsión de la imagen.

La ecualización del histograma tiene como objetivo obtener un nuevo histograma a partir del histograma original, con una distribución uniforme de los diferentes niveles de intensidad, normalizando el número de intensidades a valores entre 0 y 1, y en dividir cada elemento del histograma por el número de píxeles de la imagen, obteniendo una distribución uniforme.

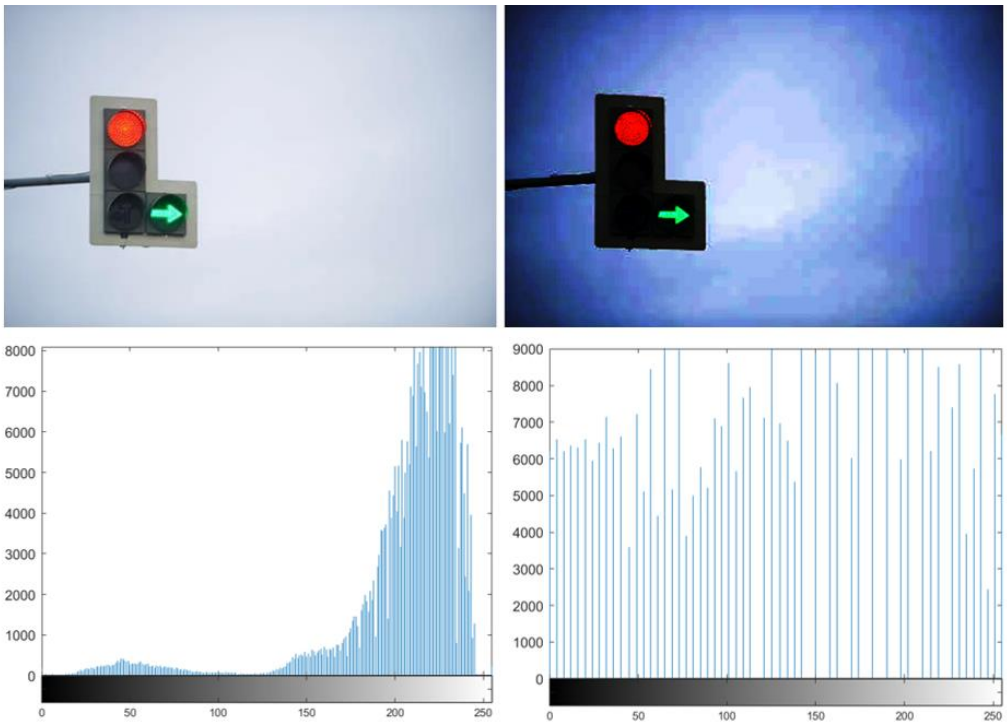


Figura 2-23. A la izquierda imagen real y su correspondiente histograma. A la derecha imagen realzada

3 DETECCIÓN DE CÍRCULOS

Una vez realizada la simplificación de las imágenes y tras haber eliminado de todas ellas las zonas de no interés para el algoritmo a desarrollar, a partir de umbralización por color, lo que sigue es, entre las formas que aparezcan en la imagen, hacer una nueva selección mediante la detección de círculos, que es la forma que presentan los objetos que se desean discriminar en la imagen.

Esto será realizado a partir de la Transformada de Hough, que tiene como objetivo localizar el círculo de mayor tamaño en las imágenes que se le adjunten.

Para aplicar esta transformada serán necesarios varios pasos previos puesto que la imagen a la que se le aplicará dicha transformada debe cumplir unos requisitos que se desarrollarán a lo largo del capítulo.

3.1 Adecuación de la imagen

Tal y como se ha concretado anteriormente, para la aplicación de la Transformada de Hough, la imagen debe cumplir unas especificaciones concretas.

En primer lugar, es necesario obtener la imagen a tratar en escala de grises debido a que posteriormente se obtendrá una imagen binaria de bordes y, una vez obtenida la imagen binaria de bordes, dichos bordes se reducirán a un píxel con el propósito de lograr una correcta aplicación de la Transformada.

3.1.1 Imagen en escala de grises

En visión artificial es muy común la conversión de imágenes a escala de grises para su posterior binarización con el fin de mejorar sus propiedades y proceder después a la extracción de características. Una imagen en escala de grises consiste en una matriz donde cada componente se refiere al nivel de intensidad de cada píxel. [7]

Para dicha conversión en MATLAB se usará la orden *rgb2gray*, que convierte la imagen real RGB a escala de grises eliminando la información de tono y saturación a la vez que conserva la luminancia.

De esta forma se consigue una imagen de una sola capa que puede ser tratada de forma conjunta, a diferencia de la imagen en RGB, cuyo tratamiento requiere la división de la imagen en los tres canales existentes (rojo, verde y azul, tal y como su propio nombre indica).

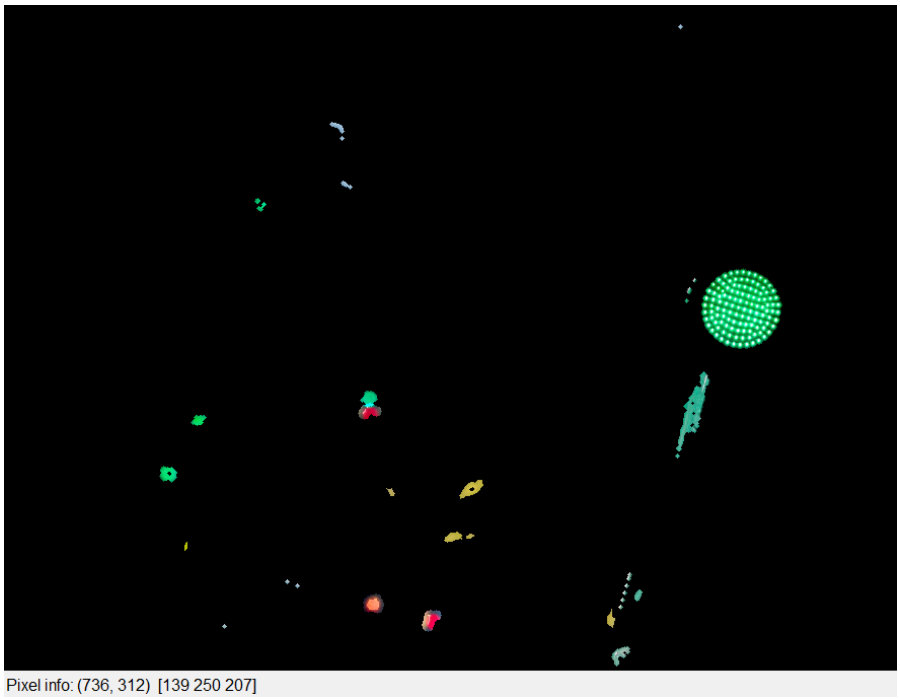


Figura 3-1. Imagen RGB segmentada

Haciendo un análisis con la herramienta *imtool* en Matlab, se puede observar que, en la *Figura 3-1*, el valor de uno de los píxeles, concretamente perteneciente al círculo del semáforo es especificado como un vector de tres componentes, concretamente [139 250 207], cada uno equivalente a su correspondiente canal RGB.

Además, es apreciable que posee un nivel de intensidad muy elevado en el canal verde, y algo menores, aunque no bajos en los canales rojo y azul, debido al brillo presente en el objeto de interés.

Figura 3-2. Imagen segmentada en escala de grises

Sin embargo, en la *Figura 3-2* se ha aplicado la conversión a escala de grises.

En ella se puede observar, con el cursor en las mismas coordenadas aproximadamente que en la imagen anterior, que el valor del píxel se expresa con un solo escalar, en concreto 255, debido a la orden *rgb2gray* aplicada, y adecuándola para el siguiente paso que consistirá en la obtención de una imagen binaria de bordes.

3.1.2 Imagen binaria de bordes y reducción a un píxel

El contorno de un objeto viene dado por los píxeles que integran la frontera de este. Estos bordes son puntos donde se producen discontinuidades en el conjunto de píxeles vecinos.

La representación del contorno no es sencilla por varios motivos, entre los que destaca su sensibilidad al ruido. Este ruido suele aparecer derivado de la naturaleza de la imagen, que hace que se pueda producir la aparición de un borde que realmente no existe. También pueden darse defectos en la segmentación, como por ejemplo bordes de elevado grosor, cuando lo deseable es que dichos bordes sean de un solo píxel. Suele ser un proceso muy poco eficiente desde el punto de vista computacional y de almacenamiento de la información.

Una imagen binaria de contornos se trata de un contorno en blanco con el fondo negro. El concepto de imagen binaria es importante puesto que de esta forma los objetos de interés se etiquetarán con un valor distinto de los píxeles que conforman el fondo, por eso se trata del paso previo para la separación de regiones. Para ello se definirá un umbral para dejar fuera de la imagen de contornos gradientes débiles.

El procedimiento consiste en utilizar un elemento estructurante para erosionar la imagen original buscando los valores mínimos de la imagen original en una vecindad 2x2, que es lo que se considera en elemento estructurante.

Estos valores mínimos de la vecindad se van almacenando en una nueva matriz, que posteriormente será la que se le reste a la imagen original, que será la imagen en escala de grises de la segmentación realizada previamente.

Para un resultado más preciso en el que los bordes sean de grosor un píxel, la imagen original se transforma en una imagen donde se discriminen todos los píxeles o elementos de la matriz que conforma la imagen que tengan un valor inferior a 100.

Para explicarlo mejor se adjunta un ejemplo visual del procedimiento:

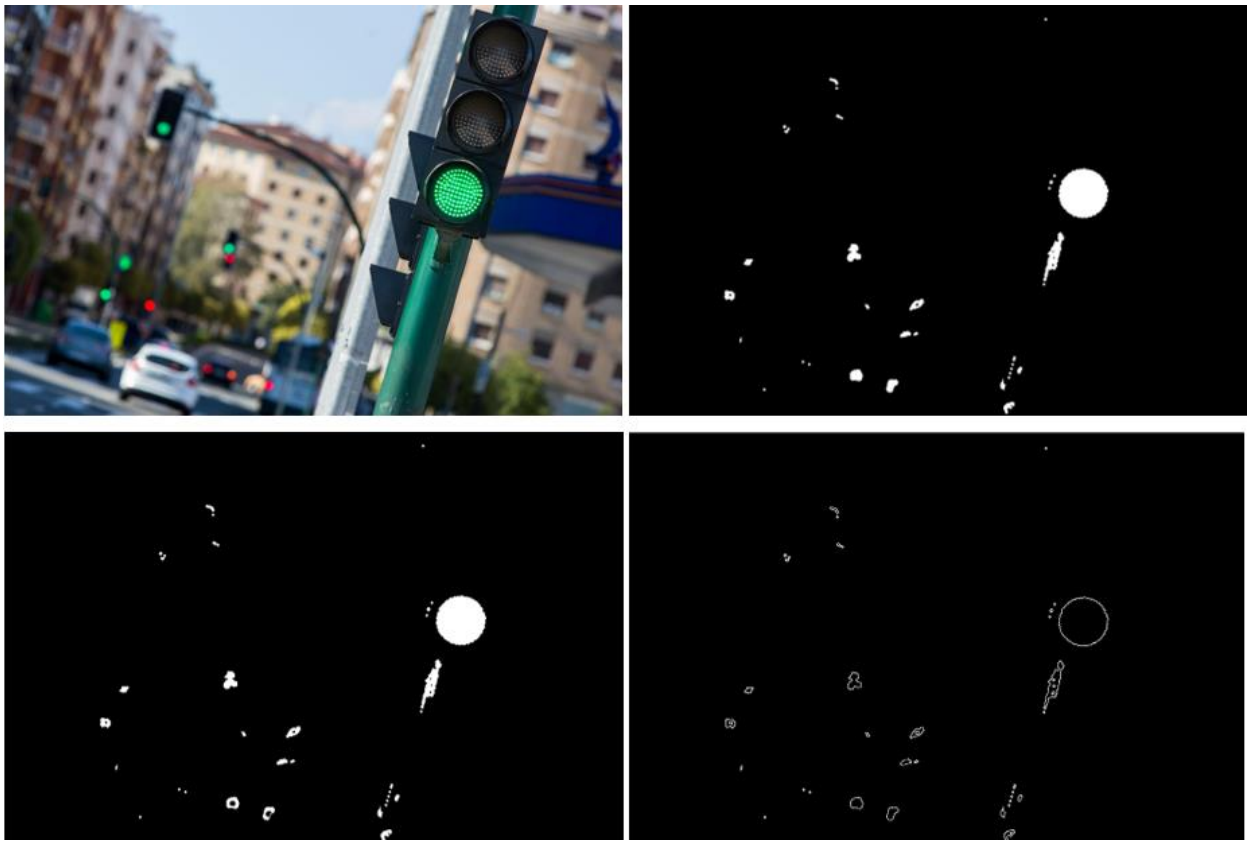


Figura 3-3. Procedimiento para la extracción de bordes de un píxel

En la secuencia de imágenes se puede observar en la imagen superior izquierda la imagen original donde se puede observar una imagen de semáforo bien definida. A continuación, en la imagen superior derecha se obtiene la segmentación en escala de grises del semáforo en cuestión.

El siguiente paso realizado en la imagen inferior izquierda ha sido aplicar la detección de bordes con un umbral de intensidad de valor 25. Tal y como se puede observar no se llega a resultados concluyentes puesto que no se obtienen bordes del semáforo, siendo este la zona de interés.

Es por los motivos anteriores por lo que para la obtención de la última figura (inferior derecha), se ha establecido un valor umbral de intensidad 100. De esta forma se consiguen los resultados esperados, donde se observa con claridad un círculo bien enlazado y de un píxel de grosor, que define el círculo del semáforo que se tratará con la Transformada de Hough.

Figura 3-4. Obtención de imagen binaria de bordes de un píxel para semáforo rojo

Existen otros métodos de detección de bordes, como pueden ser los códigos de cadena, donde se representan las fronteras en base a un conjunto de segmentos con longitud y dirección específica. El contorno se recorre en sentido horario mientras se le van asignando direcciones concretas. Esta opción ha sido descartada por su resultado sensible al ruido y de gran longitud.

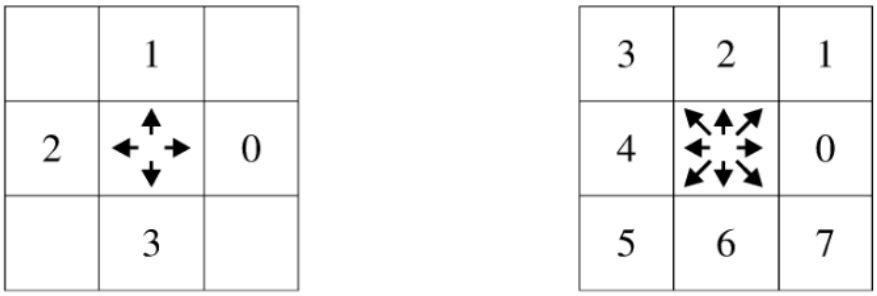


Figura 3-5. Direcciones de código cadena

Otras formas existentes son la aproximación poligonal, ajuste de rectas, representación polar (que consiste en calcular el centro de masas del contorno y representar la distancia de cada punto a dicho centro), descriptores de Fourier, etc.

También se puede aplicar la detección de bordes usando filtros de gradiente. Sin embargo, al tratarse de imágenes con ruido, el filtrado aplicado no será suficiente para la obtención concreta y precisa de bordes de un píxel y, uno de los principales requerimientos de la imagen que se introduzca para ser tratada con la transfomada de Hough es que deben ser binarias y de un píxel de grosor.

Es por ello por lo que, una vez obtenidas todas las imágenes de bordes de grosor un píxel, se procederá a la detección de círculos mediante la Transformada de Hough, procedimiento que se describirá en la siguiente sección.

3.2 Transformada de Hough

3.2.1 Introducción a la transformada de Hough

Tal y como se ha ido especificando a lo largo del proyecto, el objetivo de la visión artificial consiste en poder asemejar el sentido de la visión a un computador.

El ser humano es capaz de diferenciar con claridad patrones y formas que tiene aprendidos. Esta capacidad es la que se intenta trasladar al computador con la Transformada de Hough haciendo a este método uno de los más importantes en el tratamiento de imágenes.

La Transformada de Hough se trata de un método usado para aislar formas concretas dentro de una imagen y que ha sido diseñado con el fin de detectar curvas que puedan ser parametrizadas como líneas rectas o círculos.

Los resultados obtenidos de la aplicación de este método se caracterizan por la elevada robustez y buenos resultados que se obtienen.

En adición a dicho beneficio, es necesario añadir también que las imágenes que se le adjuntarán al método son imágenes que han sido procesadas con una gran variedad de aplicaciones a fin de hacer aún más robusto el algoritmo.

Sin embargo, este método presenta un coste computacional muy elevado, hecho que hace que se busquen en la actualidad cada vez más métodos, paralelos a este que disminuyan dicho coste.

Se ha indicado en apartados anteriores que a la hora de aplicar la Transformada de Hough a una imagen es necesaria la obtención previa de una imagen binaria de los píxeles que forman los bordes. [8]

3.2.2 Detección de líneas rectas

El algoritmo creado por Hough en 1962, permite determinar el conjunto de rectas que forman una nube de puntos, cumpliendo la siguiente ecuación:

$$y = a * x + b$$

Siendo a y b los parámetros que determinan la pendiente (a) y la ordenada en el origen (b) que hacen existentes infinitas rectas que pasan por un punto (x, y). [9]

El objetivo es encontrar los conjuntos de puntos en la imagen que satisfacen la ecuación de la recta anteriormente planteada.

Este caso es planteado suponiendo que se conocen las dos constantes que determinan la recta, pero ¿qué ocurriría en el caso en el que lo que se conocen son los puntos x e y, pero no los parámetros de la recta?

En este caso los parámetros pasarán a ser variables y las coordenadas serán constantes. De esta forma se obtendrá una nueva recta, denominando a dicha transformación *Transformada de Hough*.

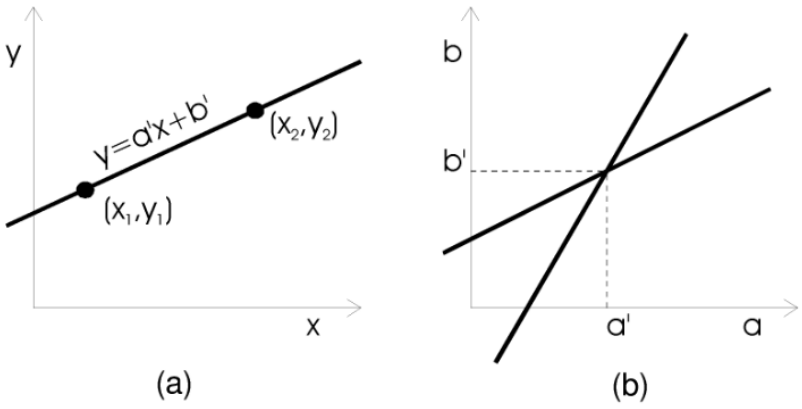


Figura 3-6. a) Plano x-y; b) Espacio de parámetros

Hacer este espacio de parámetros o también llamado espacio de Hough es lo mismo que representar una recta de pendiente "-x" y ordenada en el origen "y". Por tanto:

"El espacio de Hough representa el lugar geométrico de todas las rectas que en el plano imagen pasan por el punto (x, y)".

En la *Figura 3-6* se puede observar que el punto donde se cortan las dos rectas en el espacio de parámetros (a',b') se corresponde a la recta que pasa por los puntos (x_1,y_1) y (x_2,y_2). Así que puntos alineados en el plano imagen se transforman en el espacio de parámetros en rectas que se cortan en un solo punto.

Sin embargo, este método plantea un problema, y es que en rectas verticales el parámetro *a* tiende a infinito, por eso se plantea la representación de una recta en coordenadas polares:

$$x * \cos(\theta) + y * sen(\theta) = \rho$$

Siendo θ y ρ el ángulo formado con el eje x y la distancia recta al centro respectivamente y que se explicará posteriormente.

Para aplicar la transformada de Hough es necesaria la discretización del espacio de parámetros, es decir, se implementan unas celdas acumuladoras que dividen el espacio de parámetros y que permite evaluar todos los valores por punto, implementando un intervalo concreto que incluya un valor mínimo y uno máximo elegidos.

La ecuación evaluada para una línea recta es la siguiente:

$$b = -a * x + y$$

Es importante establecer que el valor de *b* se redondeará al valor discreto más cercano. Cada vez que se obtenga un valor concreto, esa posición en la *matriz de celdas de acumulación* se incrementará en una unidad.

Una vez que se finalice el método, el número de las celdas que contengan la mayor "puntuación" o número de votos, determinará el conjunto de rectas detectadas.

Lo mismo ocurre si el procedimiento se realiza con coordenadas paramétricas, cuya diferencia reside en que, en este caso, en lugar de obtener rectas, por cada punto estará asociado una sinusoide en el plano θ y ρ, como se muestra en la siguiente figura [10].

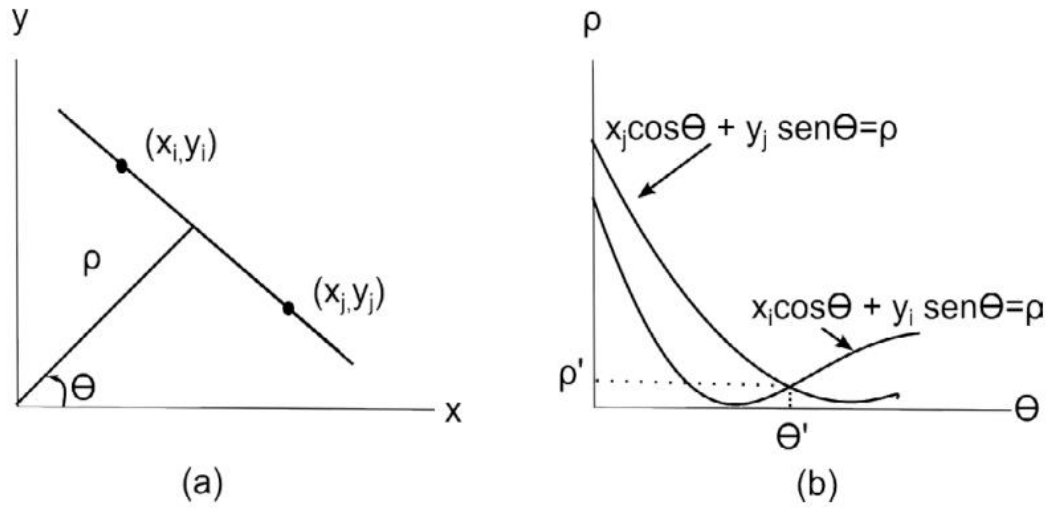


Figura 3-7. (a) Rectas en el plano x-y; (b) Curvas en el plano θ-ρ

A lo largo de todo el proyecto se ha ido recalcando la importancia de tener presente que se trabaja con imágenes reales. Por consiguiente, en este caso también es necesario tener en cuenta dicha consideración debido a que el tratamiento previo debe ser muy preciso si se quieren obtener buenos resultados, ya que si esto no ocurre la detección de las formas requeridas no dará resultados satisfactorios.

Una vez explicado de manera concisa el proceso para la detección de rectas se procede a la explicación de detección de círculos que es análogo al presente método descrito.

3.2.3 Detección de círculos

En este caso se realiza el mismo procedimiento que para la detección de rectas, pero ahora la ecuación paramétrica que tendremos será la correspondiente a un círculo, por tanto, los parámetros serán las coordenadas que establecen el centro de este y el radio. Es por eso por lo que el acumulador ahora tendrá tres dimensiones debido a que de primera mano no se conocerán los radios de los círculos que forman las luces de los semáforos. Si este radio fuera conocido, el acumulador tendría dos dimensiones como en el caso de la recta [8].

La ecuación de una circunferencia es la siguiente:

$$r^2 = (x - a)^2 + (y - b)^2$$

Donde *r* es el radio de la circunferencia y (*a*, *b)* son las coordenadas de su centro.

El siguiente paso es parametrizar dicha ecuación de forma que se obtienen las siguientes ecuaciones:

$$x - r * cos(\Theta) = \text{a}$$

$$y - r * sen(\Theta) = \text{b}$$

El procedimiento desarrollado en la función creada para la detección de círculos en semáforos, en primer lugar, ha sido crear una matriz donde se guardarán los valores acumulados. Esta matriz será *Acu,* que tendrá tres dimensiones como ya se ha explicado anteriormente, tendrá un tamaño en las dos primeras dimensiones de acuerdo con las dimensiones de la imagen que se esté tratando, y una tercera dimensión que en el presente caso se ha establecido en 100.

El paso principal será establecer un valor de radio. Para este valor y con el fin de llegar a todas las zonas, se recorrerá la matriz que guarda los píxeles de la imagen binaria de bordes por filas y columnas, y en el caso de que se encuentre un píxel de valor uno, es decir, en el caso en el que encontremos una zona de interés, se aplicará el procedimiento de detección.

Este procedimiento consiste en, para el punto encontrado recorrer un primer bucle cuyo índice será el valor del ángulo *theta* que comprende ángulos entre cero y 360 grados.

Para cada ángulo se calcularán las coordenadas de los centros *a* y *b* según las ecuaciones paramétricas anteriormente propuestas.

El siguiente paso es comprobar que los valores propuestos por dichas ecuaciones se encuentran dentro de la zona delimitada por la imagen. Si esto ocurre, se redondean estos valores al entero más cercano y se vota en la matriz de acumulación, es decir, en la celda posicionada en las coordenadas *(b, a, radio)* se añade una unidad.

Es destacable que se coloca en la primera dimensión la coordenada *b* del centro propuesto ya que las imágenes en MATLAB se recorren en filas – columnas.

Una vez realizado esto, se incementará el valor del ángulo *theta* hasta haber recorrido los 360 valores. Recorridos todos ellos, se sigue recorriendo la imagen binaria de bordes y se repite todo el procedimiento para todos los píxeles de la imagen binaria bordes calculada en apartados anteriores.

Una vez que la imagen se ha recorrido entera se incrementa el valor del radio y se vuelve a desarrollar todo el procedimiento hasta finalizar todos los bucles.

Para localizar la circunferencia de mayor tamaño, se sigue con detectar aquellas celdas de la matriz de acumulación que tienen más votos. Esto se comprueba recorriendo un bucle que tenga como índice el tamaño de la tercera dimensión de la matriz de votos acumulados, es decir, los cien radios probados.

Entre todos los valores que conforman la matriz de acumulación para un radio concreto se detecta el máximo y se guarda en un vector (*votos*) y se procede a la escritura de las coordenadas de la fila y la columna en las que se encuentra dicho máximo en las variables *rx* y *ry* y el valor del radio en *radmax*.

Los siguientes pasos consistirán en seguir iterando de forma que el valor de las coordenadas *rx, ry* y *radmax* se modificarán solo en el caso de que el máximo encontrado sea mayor que los anteriores detectados.

De esta forma, una vez que se ha iterado para todos los valores de radio existentes y en beneficio de discriminar solo el valor máximo entre todos los posibles encontrados, se concluirá la detección del círculo, obteniendo como parámetros de salida de la función el valor del radio, y las coordenadas de su centro.

El siguiente paso, una vez han sido detectados los valores del radio y las coordenadas del centro del círculo, ha sido su representación mediante un círculo rojo junto con la imagen original para facilitar la visualización de los resultados.

Un caso sencillo es el que se muestra a continuación:

Figura 3-8. A la izquierda imagen binaria de bordes; a la derecha detección del círculo

En el caso presentado se puede observar a primera vista que la imagen binaria de bordes presenta un círculo significativo, que en este caso se corresponde con la luz del semáforo y además es el más grande, por tanto, la detección se hace de forma rápida y sencilla.

Además, un aspecto a destacar de la Transformada de Hough es que no solo detecta la curva en la imagen, sino que también obtiene los parámetros de ésta que mejor se ajustan a los datos generados. Por ejemplo, en el caso de que no se detecten círculos completos, se generará un ajuste de los datos existentes.

Figura 3-9. Detección de círculos poco definidos de semáforos en ámbar

En el ejemplo anterior se puede observar que, en la imagen binaria representada a la izquierda no se aprecia con claridad un círculo como el que se podía visualizar en el primer ejemplo. Sin embargo, es en este caso donde se puede hacer notar que la Transformada de Hough implementada no solo busca el centro y el radio de círculos completos, sino que es capaz de localizar aquellos puntos que mejor se adapten a la forma buscada, aunque en la imagen binaria no se trate de una forma completa.

En la imagen de la derecha se ilustra el resultado definitivo donde se demuestra todo lo dicho anteriormente.

Se puede ver que, a pesar de ser un círculo pequeño debido a la lejanía de la toma de la foto con respecto al semáforo, ha sido bien detectado gracias al conjunto del tratamiento previo de las imágenes realizado y la aplicación de la Transformada de Hough para la detección.

En la siguiente figura se puede apreciar que en la escena se dan unas condiciones de iluminación distintas a la imagen anterior, por tanto, el círculo del semáforo en la imagen binaria de bordes está más definido, pero la detección se produce de la misma forma, con la única diferencia de que en este caso el encendido del semáforo es verde y anteriormente era ámbar.

Figura 3-10. Detección de círculos en semáforo lejano verde

A continuación, se mostrará la misma escena, pero para el semáforo con la luz encendida en rojo y capturado desde otro ángulo. En este caso se observará que con la umbralización y todo el tratamiento previo se ha descartado completamente cualquier objeto de la imagen que no se encuentre en el intervalo de color establecido para la detección a pesar de que existen numerosos objetos en escena y una gran variedad de combinaciones posibles en los tres canales RGB.

Esto ha resultado una amplia simplificación y aporta gran robustez al algoritmo desarrollado, demostrando que la umbralización realizada se ha realizado de manera precisa y que, además, responde bien ante cambios en los valores de intensidad luminosa presentes en las distintas imágenes.

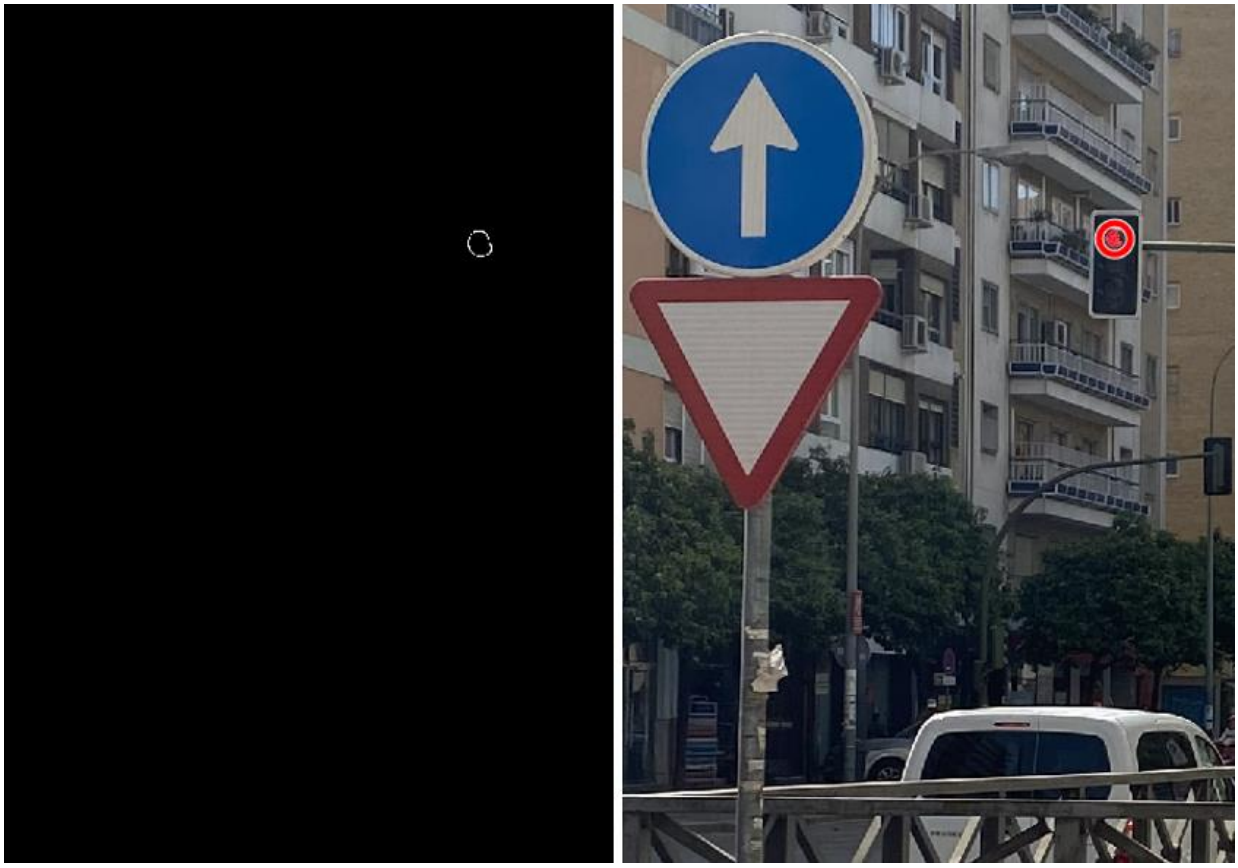


Figura 3-11. Detección de semáforo rojo con señales de tráfico presentes

En la imagen binaria de bordes se puede observar que no solo se ha descartado todo el "*background"* sino que además es posible apreciar cómo la señal de ceda el paso, a pesar de ser roja se discrimina completamente gracias a que no cumple con el umbral de una luz roja brillante.

Además, también se nota la existencia de una señal circular, que también ha sido descartada gracias a la umbralización. En el caso de que esta señal no hubiera sido discriminada, la situación más probable es que el resultado final no tendría éxito ya que el círculo más grande detectado sería esta señal.

Una vez representadas las detecciones de los tres colores de un mismo semáforo en diversas condiciones y tomando las capturas desde zonas distintas se procederá a la representación de otros resultados que resultan de interés y que muestran casos más variados para los tres colores presentes en los semáforos.

Figura 3-12. Ejemplo de detección de luz verde

A continuación, se mostrará un ejemplo que ha sido tratado y mencionado durante todo el proyecto debido a que se trata de una imagen en la que la zona de interés y que se desea detectar es la luz roja del semáforo, pero sin embargo aparece una señal de tráfico, circular y además de color rojo. El tratamiento previo que se realizó para obtener éxito en la detección hace que los resultados sean los siguientes:

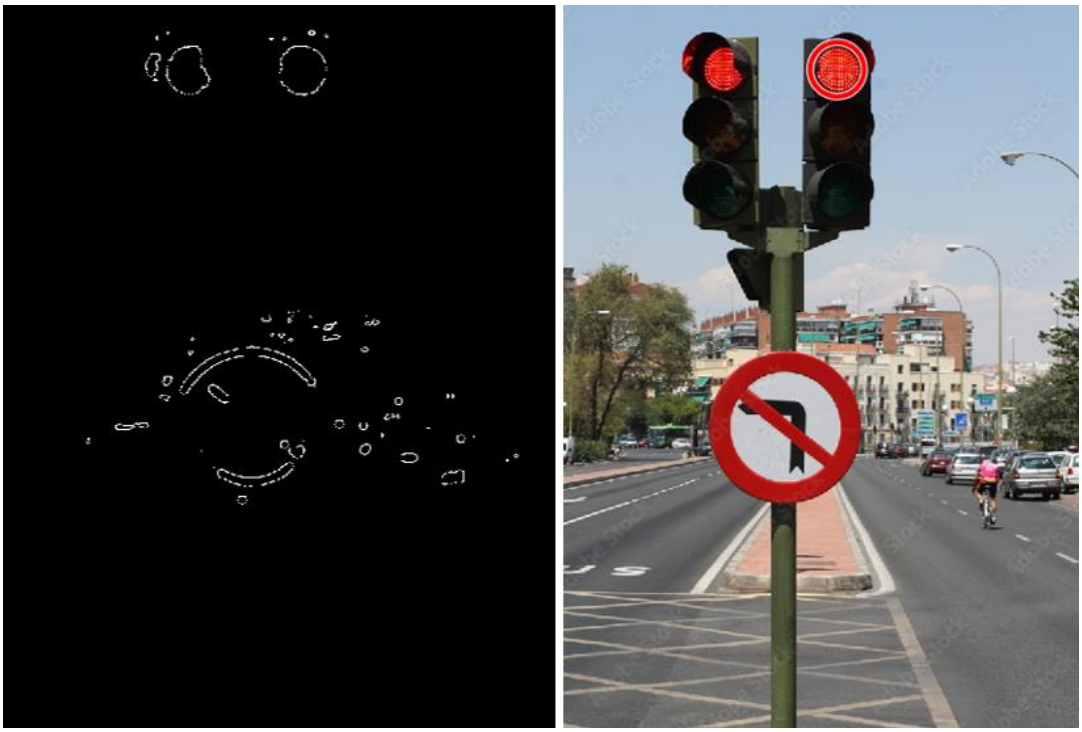


Figura 3-13. Detección de círculo en escena con señal de tráfico

3.3 Resumen del capítulo

Una vez comprobado que el conjunto de umbralización y Transformada de Hough produce resultados concluyentes y que, además, la comprobación de las 40 imágenes es prueba de este éxito, el paso siguiente será extraer información de esa zona de interés con el fin de comprobar qué fiabilidad tiene el método desarrollado.

Dicha información será crear un método que sea capaz de clasificar con la máxima fiabilidad posible el color que presenta la zona de interés, ya que, una vez que ha sido detectado que existe un semáforo en una cercanía concreta, lo que sigue, y de vital importancia, es conocer qué información contiene dicha zona para una actuación posterior.

Para ello el primer paso que sigue será recortar la zona de la imagen que encuadra la luz del semáforo y hecho esto los pasos siguientes serán el tratamiento y clasificación de esta selección.

Se busca que el proyecto sea lo más automatizado posible, pues se pretende asemejar y adaptar dicho trabajo a sistemas autónomos, por tanto, la selección de la imagen se realizará de forma automática gracias a los parámetros de salida de la función que aplica la transformada de Hough.

4 EXTRACCIÓN DE LA ZONA DE INTERÉS

En este capítulo se describirá el procedimiento seguido para localizar y extraer de las imágenes originales las zonas de éstas que sean de interés para su posterior tratamiento y clasificación. Se usarán herramientas de MATLAB con el fin de conseguir una extracción automática a la vez que efectiva y regular para todas las imágenes, puesto que en los apartados siguientes dichas secciones deberán ser comparadas entre ellas y deben presentar las mismas características.

4.1 Procedimiento de extracción

Los pasos que seguir para la separación de la luz del semáforo que posteriormente se analizará se trata, en primer lugar, de recortar la imagen original.

Para ello se usarán los valores de las coordenadas del centro del círculo y del radio obtenidos en el capítulo anterior a partir de la Transformada de Hough. Con estos valores es posible aplicar la orden *imcrop* de MATLAB, que recibirá como parámetros de entrada el rectángulo de corte que se desea segmentar a partir de sus valores de esquina inferior derecha (dado por sus coordenadas *x* e *y*), su anchura y su altura.

Para conocer la localización de la esquina inferior derecha, se supone un círculo encerrado en un rectángulo:

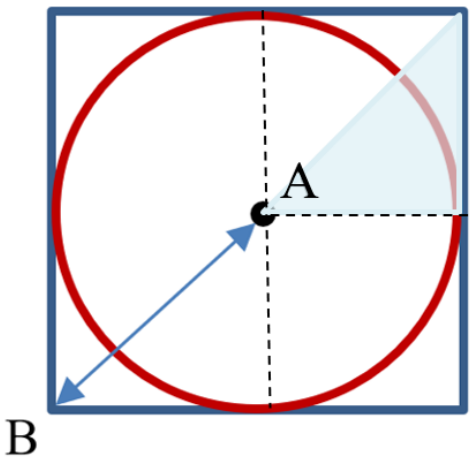
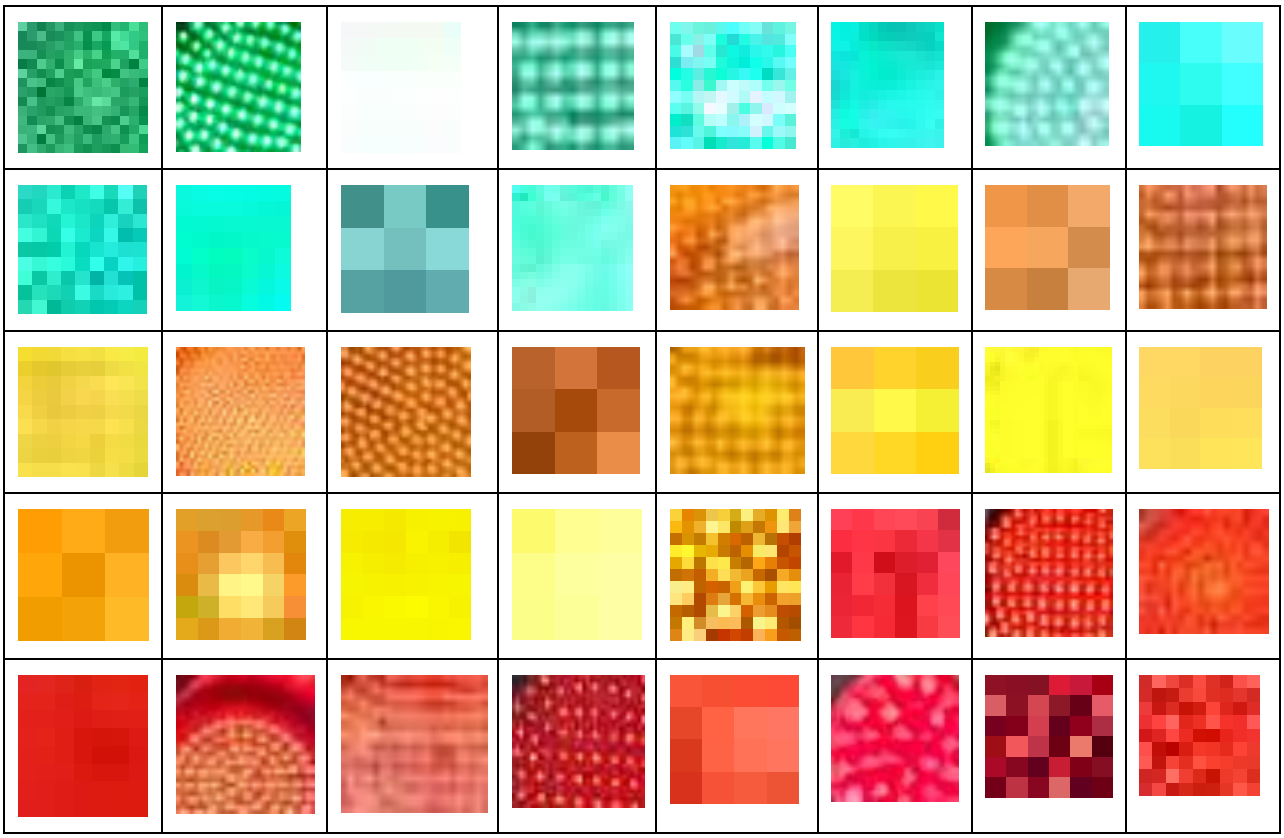


Figura 4-1. Círculo inscrito en cuadrado

En la *Figura 4-1* se puede observar que la distancia entre el punto A y B se puede calcular conociendo que se forma un triángulo (como el sombreado en la figura) en el que los dos catetos tienen la longitud del radio, y por tanto la hipotenusa es el radio por la raíz cuadrada de 2. De esta forma se localizan todos los círculos.

4.2 Zonas recortadas

Tabla 4-1. Recopilación de las zonas de interés de cada semáforo estudiado



Una vez han sido extraídas todas las regiones en las imágenes estudiadas, se puede añadir a modo de conclusión que su caracterización y clasificación necesitará un método muy exhaustivo debido a que, los colores que se muestran en cada una de las segmentaciones son muy variados.

Se puede observar cómo para los semáforos verdes existen situaciones en las que el color es verde casi al nivel normalizado, pero también se dan casos en los que el color tiene más componente de azul que de verde, dificultando de esta forma su clasificación, y provocando que, en consecuencia, el procedimiento que se realice tenga en cuenta todos estos aspectos.

Otro detalle que destacar es que, en la tercera imagen se puede observar con claridad que se trata de una segmentación prácticamente de tonalidades blancas, es decir, al realizar la media de los valores de intensidad luminosa presentes en cada canal, será fácilmente observable que las componentes de cada uno de ellos (RGB) rondarán valores superiores a 250.

En cuanto a los semáforos ámbar y rojo, también surge una problemática, y es que tendrán componentes muy similares además de variadas, debido a que en el semáforo ámbar se puede apreciar que se representan imágenes naranjas a la vez que se dan componentes que forman un color más amarillo. Dichas componentes naranjas son fácilmente asemejables al color rojo del semáforo, luego la discriminación entre la barrera que separa ambos colores será muy estrecha.

Los valores de intensidad luminosa de cada canal para el color rojo normalizado son (255, 0, 0), para el amarillo (255, 233, 0) y para el naranja (255, 127, 0), por tanto, será el nivel de intensidad luminosa del canal verde el que determine la separación correcta de estos colores.

5 MÉTODOS DE CLASIFICACIÓN

En este capítulo se describirá el procedimiento final seguido para obtener una clasificación precisa de las luces de semáforos detectados. Para ello se desarrollarán dos clasificadores distintos con el fin de buscar aquel que presente unos resultados más exactos.

Por un lado, se desarrollará un método de clasificación de imágenes mediante correlación, de forma que se compararán las imágenes con un patrón concreto y aquel que presente mayor coeficiente de correlación discriminará la imagen.

Por otro lado, se creará una función de decisión que será entrenada con un porcentaje concreto de semáforos y probada con el porcentaje restante.

5.1 Clasificación por correlación

5.1.1 Elección de patrones

La correlación es una medida estadística que expresa la relación entre dos variables que están relacionadas linealmente.

El procedimiento que hay que realizar cuando se clasifican imágenes por correlación empieza por buscar un buen "patrón" que presente características similares a las imágenes que van a ser clasificadas.

Para ello, el primer paso será decidir qué patrones se usarán para la clasificación.

Es necesario establecer un patron que se corresponderá lo máximo posible con el color rojo de un semáforo, otro para el color ámbar y, por último, para la gama verde.

En una primera prueba, se buscan resultados aplicando patrones formados por imágenes sintéticas roja, ámbar y verde:

Figura 5-1. Colores sintéticos

Sin embargo, si a simple vista se produce la comparación de los colores sintéticos de la *Figura 5-1* con los de la *Tabla 4-1* del capítulo anterior, se podrá observar que, a pesar de tratarse del mismo color, presentan valores de intensidad luminosa muy diferentes.

El ejemplo más representativo de ello sería comparar la imagen sintética de color ámbar, con uno de los semáforos que presentan intensidades luminosas típicas de un color amarillo. En este caso la correlación sería muy baja.

Figura 5-2. De izquierda a derecha: Tonalidad naranja, tonalidad amarilla e imagen sintética

En el caso de semáforos verdes ocurrirían situaciones similares a causa de que, como se ha expuesto en capítulos anteriores, dicho color presenta casos en los que el brillo del semáforo ocasiona que su representación presente un valor de intensidad luminosa muy elevada en el canal azul.

Figura 5-3. De izquierda a derecha: Tonalidad verde, tonalidad turquesa e imagen sintética

Es por estos motivos por los que este procedimiento con patrones sintéticos devolvería resultados muy poco concluyentes.

La segunda opción planteada, será comprobar la correlación de los semáforos aplicando patrones que se traten de semáforos reales, es decir, tomar una de las imágenes segmentadas de un semáforo rojo, otro verde y otro ámbar, y compararlas con el resto de las imágenes segmentadas de cada uno de los 40 semáforos tratados en capítulos anteriores, pero ¿cómo solucionaría esta decisión la correlación entre aquellos casos que se han expuesto anteriormente?

Como es esperable, la correcta clasificación de semáforos naranjas y semáforos amarillos no se soluciona aplicando como patrones imágenes reales. En cualquier caso, mejoraría el índice de bondad del método ya que, por ejemplo, para el semáforo verde, si en la mayor parte de cada uno de los semáforos el nivel de intensidad luminosa en el canal azul es elevado, lo ideal sería tomar un patron con dichas componentes elevadas.

No obstante, no sería un buen clasificador en aquellos semáforos que presenten niveles más adecuados a los normalizados.

Lo mismo ocurre con el semáforo ámbar. Si se toma como patrón una imagen real que se corresponda con el color amarillo, porque la mayoría de los semáforos ámbar son amarillos, aumentará ligeramente el porcentaje de acierto en el método.

Pero estos hechos mencionados no solucionan el problema de la clasificación ya que de esta forma se están tomando plantillas en beneficio del acierto, y suponiendo que se conoce la probabilidad de que se den unas

tonalidades u otras.

No obstante, los resultados obtenidos de aplicar este procedimiento no proporcionan hechos concluyentes, puesto que el índice de bondad oscilaría entre el 40% y el 50%, y se busca un porcentaje de acierto superior al 70% aproximadamente.

Por tanto, la solución óptima será entrenar un patrón con varias de las imágenes segmentadas como se explica a continuación:

Se desarrollará una función en MATLAB que, en primer lugar, lea todas las imágenes segmentadas (las que se muestran en la *Tabla 4-1*). Para cada una de estas imágenes:

- Guardar en una matriz *plantilla* la imagen leída.
- Crear nuevas imágenes donde se guarden los canales RGB de la imagen. Estas matrices serán *capa_roja_plantilla, capa_verde_plantilla* y *capa_azul_plantilla.*
- Las capas anteriores deberán ser redimensionadas puesto que para desarrollar la correlación se tratarán conjuntamente y, por tanto, es necesario que se trate de matrices de iguales dimensiones.

Una vez realizado este procedimiento, se almacenarán las capas de cada canal perteneciente a cada una de las imágenes en matrices de matrices.

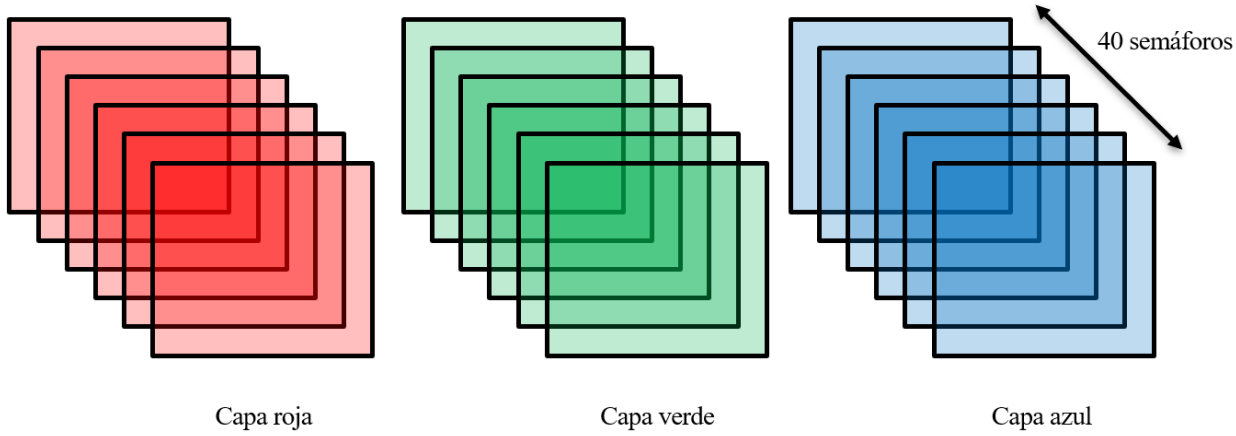


Figura 5-4. Explicación visual de la formación de matrices

Una vez calculadas las 40 capas para cada canal RGB, el siguiente paso es calcular el valor medio de una muestra representativa de semáforos verdes, semáforos rojos y semáforos ámbar, para cada uno de ellos y para cada píxel, es decir, por ejemplo, en las muestras de capas rojas hallar la media de los valores del píxel (1,1) a lo largo de todas las muestras y capas, después del (2,1), (3,1), etc.

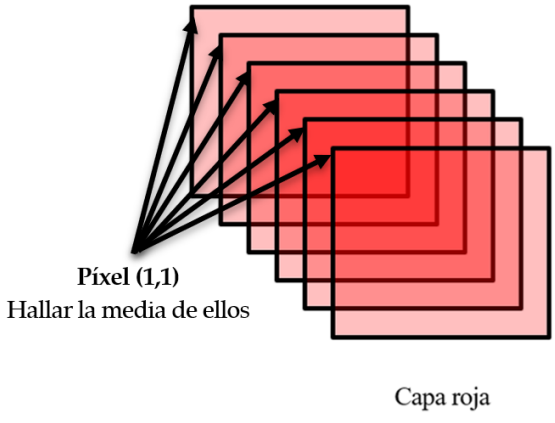


Figura 5-5. Ilustración del cálculo del valor medio por píxel

De esta forma se obtienen para cada color de semáforo (rojo, verde y ámbar) tres capas, una por cada canal RGB, es decir, nueve capas en total, que se compararán para calcular el coeficiente de correlación con las capas de los semáforos reales a probar.

Una vez obtenidas estas muestras, antes de usarlas como patrón serán filtradas, con el fin de eliminar todo el ruido que pueda aparecer presente en la imagen a causa de la acumulación de este debido a tratarse de imágenes reales.

Dicho filtrado se realizará con el comando de MATLAB *medfilt2*, que realiza un filtrado de mediana de la imagen en dos dimensiones. Este es el motivo por el que el filtrado se realizará por capas, porque el filtro no aplica para tres dimensiones. "Cada píxel que sale de dicha función contiene el valor de la mediana de una vecindad 3 por 3 alrededor del píxel correspondiente de la imagen de entrada". [11]

En definitiva, las imágenes patrón obtenidas para cada uno de los semáforos serán:

Figura 5-6. Imágenes patrón para el cálculo del coeficiente de correlación

Calculados los patrones, lo que sigue será el cálculo del coeficiente de correlación usando dichas plantillas. Es importante conocer de antemano que no se obtendrán resultados exactos debido a que cada imagen está presentada con unas condiciones y un ruido concreto, y la comparación con tres patrones concretos dificulatrá la obtención de resultados óptimos.

5.1.2 Obtención del coeficiente de correlación

En el apartado anterior se han obtenido los patrones necesarios para calcular con precisión el coeficiente de correlación. Los puntos que seguir para una comparación lo más optima posible empezarán por aplicar un filtrado a las imágenes de semáforos reales que se someterán a comparación.

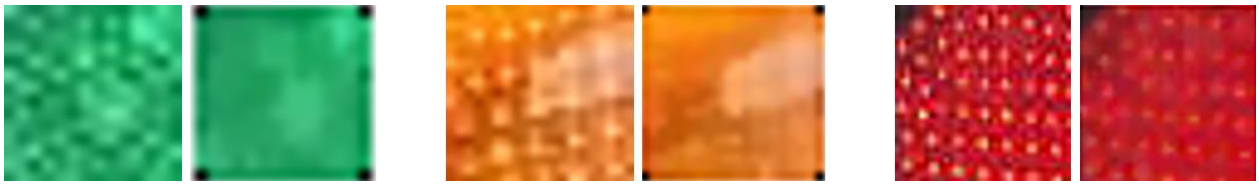


Figura 5-7. Para cada color de semáforo, a la izquierda imagen real y a la derecha imagen filtrada

Una vez que la imagen de interés ha sido tratada, se puede observar que el resultado del filtrado se asemeja en mayor grado a la imagen que servirá como patrón y que está mostrada en la *Figura 5-6* que si la comparáramos directamente sin filtrar. En este último caso el resultado sería deficiente debido a que se distinguen bien píxeles de color más brillantes. El caso más significativo es el mostrado en la *Figura 5-7* en el caso del semáforo rojo. En su representación real, se pueden observar algunos píxeles blancos debido, como se ha explicado, al brillo de la luz del semáforo. Esto se soluciona con el filtrado.

Lo siguiente será reajustar el tamaño de las imágenes al del patrón para poder compararlas. Esta operación se realizará con la orden *imresize*, obteniendo de esta forma las 40 imágenes de luces de semáforos segmentadas filtradas y con las mismas direcciones que los patrones.

Para proceder al cálculo del coeficiente de correlación se usará otra función en MATLAB: *normxcorr2*. Esta función también necesita recibir imágenes en dos dimensiones. Por eso el procedimiento se realizará capa por capa, es decir, para cada semáforo introducido que se desee clasificar, será necesaria su descomposición en los tres canales RGB, y, cada uno de esos canales será comparado mediante el coeficiente de correlación con las capas RGB del semáforo patrón rojo, después del semáforo patrón ámbar, y, por último, con el semáforo patrón verde.

En resumen, se desarrollarán nueve comparaciones.

La orden usada para el cálculo del coeficiente de correlación devuelve una matriz que contiene los coeficientes de correlación entre todos los píxeles existentes en las imágenes. Sus argumentos de entrada serán la imagen "plantilla", especificada como matriz numérica. Los valores de dicha matriz no pueden ser todos iguales. El segundo argumento de entrada será la imagen real.

La fórmula que utiliza esta orden es [12]:

$$\gamma(u,v) = \frac{\sum_{x,y}[f(x,y)-fu,v][t(x-u,y-v)-t]}{\{\sum_{x,y}[f(x,y)-fu,v]^2 \sum_{x,y}[t(x-u,y-v)-t]^2\}^{0.5}}$$

donde

- *f* es la imagen.
- *t* es la media de la plantilla.
- *f u,v* es la media de *f(x,y)* en la región debajo de la plantilla.

De todos los valores presentes en las matrices que contienen los coeficientes de correlación es necesario buscar el pico. Por tanto, después de comparar la imagen de interés con los tres colores de semáforo se buscarán para cada capa el máximo y se guardarán en tres vectores distintos: *correlaciones_con_verde, correlaciones_con_roja* y *correlaciones_con_ambar.* Estos vectores tendrán tres componentes, que incluyen el coeficiente de correlación máximo para cada canal.

Para discriminar la imagen se sumarán las tres componentes de cada vector, y se compararán entre ellas. De esta forma, aquel sumatorio que proporcione un valor más alto determinará el color del semáforo.

Una vez obtenido el coeficiente de correlación más alto, el siguiente paso será comprobar qué índice de bondad presenta este clasificador.

El indicador de bondad se trata de un índice simple que proporciona el porcentaje de clasificaciones incorrectas.

Su implementación consiste en dividir el conjunto de objetos mal clasificados entre el conjunto de todos los objetos usados en la prueba y obtener el valor en porcentaje.

El conjunto total de objetos usados en la prueba debe ser una buena muestra de los casos de uso futuros, por tanto, con una buena muestra y, por consiguiente, si los datos usados son suficientes, el indicador de bondad (PCI) obtenido de la prueba es indicativo para pruebas que se realizarán en usos futuros.

$$\text{PCI} = \frac{\text{card (CI)}}{\text{card (CT)}} * 100$$

Tabla 5-1. Clasificación por correlación: Sombreado claro: Entrenamiento; Sombreado oscuro: Pruebas

Semáforo	Color real	Clasificación por correlación
1	Verde	Verde
2	Verde	Rojo
3	Verde	Verde
4	Verde	Verde
5	Verde	Verde
6	Verde	Verde
7	Verde	Verde
8	Verde	Ámbar
9	Verde	Verde
10	Verde	Verde
11	Verde	Verde
12	Verde	Verde
13	Ámbar	Ámbar
14	Ámbar	Ámbar
15	Ámbar	Ámbar
16	Ámbar	Rojo
17	Ámbar	Ámbar
18	Ámbar	Rojo
19	Ámbar	Rojo
20	Ámbar	Ámbar
21	Ámbar	Rojo
22	Ámbar	Ámbar
23	Ámbar	Rojo
24	Ámbar	Rojo
25	Ámbar	Ámbar
26	Ámbar	Ámbar
27	Ámbar	Ámbar
28	Ámbar	Ámbar
29	Ámbar	Rojo
30	Rojo	Rojo
31	Rojo	Rojo
32	Rojo	Rojo
33	Rojo	Rojo
34	Rojo	Rojo
35	Rojo	Rojo
36	Rojo	Rojo
37	Rojo	Ámbar
38	Rojo	Rojo
39	Rojo	Rojo
40	Rojo	Ámbar

De entre los semáforos que no pertenecen a aquellos que se han empleado para obtener la plantilla para calcular el coeficiente de correlación se obtienen los siguientes resultados:

- El número total de semáforos mal clasificados es siete.
- El conjunto total de semáforos sometidos a clasificación es 19.
- Se obtiene un indicador de bondad del 36.84% de semáforos mal clasificados, lo que se corresponde con un 63.16% de acierto.

Por tanto, realizando una clasificación por este procedimiento, se obtienen unos resultados muy poco concluyentes.

Realmente, este número podría ser considerado elevado, teniendo en cuenta que se ha realizado un tratamiento a imágenes reales, que presentan ruido, brillos, escenas con gran cantidad de objetos, etc.

Sin embargo, es necesario tener presente que esta clasificación se busca con el fin de añadir un método más a la automatización de vehículos, así pues, se necesita un porcentaje de aciertos que ronde el 100%, considerando un 63.16% muy pequeño.

Es por los motivos mencionados anteriormente que se desarrollará un nuevo clasificador con el fin de aumentar los beneficios en cuanto a resultados obtenidos. Existen varios clasificadores que podrían ser empleados, entre los que destacan clasificadores lineales, clasificadores por distancia y clasificador de Bayes.

Las diferencias más sustanciales entre unos y otros son [13]:

- Los clasificadores no-paramétricos entrenables, se caracterizan porque se divide el espacio en regiones correspondientes a cada una de las clases. Estas regiones son separadas por unas funciones de decisión que suelen ser lineales, aunque también se presentan en modo lineales a trozos o no lineales.
- Los clasificadores por distancia son aquellos en los que cada clase se agrupa en torno a un prototipo, que suele ser el centroide. Se asigna a cada objeto la clase del prototipo más próximo.
- En el Clasificador de Bayes los semáforos en este caso son elementos de una población y se clasifican en la clase que tenga con una probabilidad a posteriori mayor.

Se implementará a continuación un Clasificador no-paramétrico entrenable o también llamado Método Perceptrón. Para ello en primer lugar será necesario hacer una selección de características, posteriormente se entrenará el clasificador para obtener la función discriminante y, por último, se obtendrán resultados.

5.2 Método Perceptrón

5.2.1 Introducción al clasificador no-paramétrico entrenable

Tal y como se ha indicado en apartados anteriores, la función del presente clasificador consiste en formar regiones separadas, cada una de ellas por una función de decisión que puede ser lineal o no.

El primer paso para el diseño del clasificador será realizar un entrenamiento con el fin de obtener un vector de parámetros a partir de un conjunto de muestras conocidas denominadas "patrones de entrenamiento".

Estos patrones se usan a modo de modelo para encontrar la función discriminante que clasificará correctamente los patrones del universo de trabajo.

Para ello se tomarán los 40 semáforos segmentados en capítulos anteriores y un porcentaje de ellos (60% aproximadamente) será usado para entrenar el clasificador, de forma que se obtengan funciones discriminantes lo más representativas posibles.

El 40% restante se empleará para comprobar la exactitud y la fiabilidad del método implementado. Para obtener buenos resultados los patrones entrenables deben adquirir la propiedad de generalización, que asegure que un sistema clasifica correctamente patrones que no ha visto durante el proceso de cálculo de funciones discriminantes.

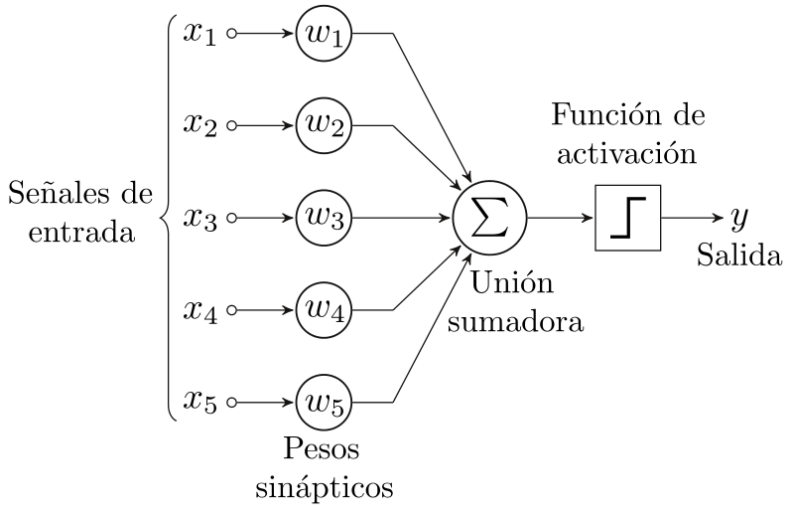


Figura 5-8. Modelo del perceptrón

En la figura anterior se puede observar que se genera un vector *w* de tantas componentes como señales de entrada existan. Estos se unen en un sumatorio, dando lugar a una salida *y* que será la que discrimine los vectores de características que se sometan a prueba, es decir, *y* será 1 si la función probada para una clase es mayor que cero y -1 en caso contrario. De esta forma, para el caso en el que se esté entrenando el clasificador para dos clases distintas, se comprueba que si está "por encima" de la función pertenece a una de las clases planteadas y si está "por debajo" de la función pertenece a la otra.

Se trata, por tanto, de un algoritmo iterativo, y parte de un vector inicial *w* arbitrario, que, por lo general se inicializará a cero.

El procedimiento de obtención de las características discriminantes y del entrenamiento iterativo se explicará en secciones posteriores.

5.2.2 Características discriminantes

El primer paso para proceder a la clasficiación de objetos consiste en extraer un *vector de características* que será usado para diferenciar unos objetos de otros. Estas características permitirán discriminar a qué clases puede pertenecer cualquier imagen segmentada.

Las características discriminantes deben cumplir una serie de objetivos que incluyen: estar ligadas a las clases de forma teórica o experimental (homogeneidad intraclase), deben tener capacidad discriminante (dispersión interclase) y, por último, no pueden ser redundantes.

Teóricamente, cada vez que se realice un procedimiento de clasificación, será necesario obtener una imagen donde los objetos segmentados sean aquellos que se deseen clasificar, ya que, en el caso de añadir imágenes reales que presenten mucho ruido y elementos presentes en la escena que no sean de interés, se producirán errores en la clasficiación.

En el presente proyecto las características discriminantes serán los valores medios de intensidad luminosa de cada canal RGB para cada una de las imágenes que se desean clasificar. En otros casos, la elección de aquellas características de los objetos que van a permitir su reconocimiento suele ser más complicada, pero en este caso la clasificación que interesará será aquella que discrimina las imágenes según su color.

Para ello, el procedimiento será el que se muestra a continuación:

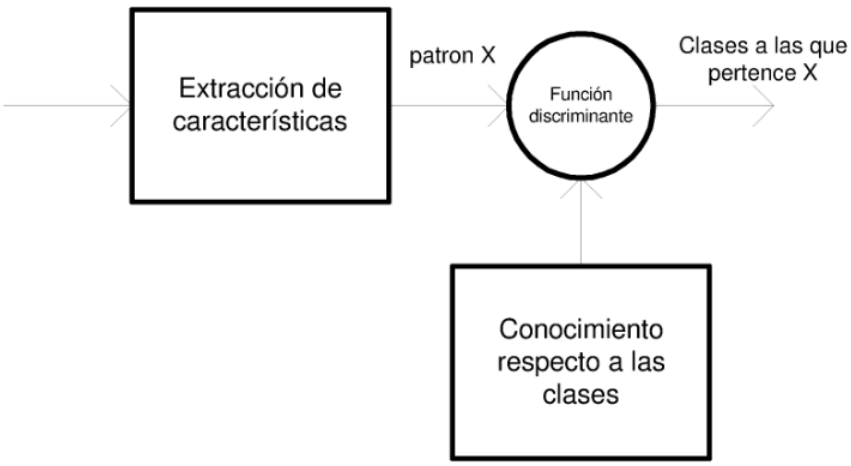


Figura 5-9. Procedimiento de clasificación de imágenes

Por tanto, para la extracción de características se realizará primero la obtención de la media de cada canal para cada imagen.

Es por ello por lo que se ha creado una función que devuelve los valores medios y cuyo procedimiento es el siguiente:

Dentro de un bucle que leerá todas las imágenes que ya tienen segmentado el círculo del semáforo, se produce dicha lectura de las imágenes. Posteriormente se divide la imagen en cada uno de sus tres canales con el fin de calcular el valor de los píxeles rojos, de los píxeles verdes y por último de los azules.

Una vez que la imagen ha sido dividida en sus tres canales RGB el siguiente paso es guardar en una variable *suma_rojo, suma_verde* y *suma_azul,* la suma de todos los píxeles de cada canal, y su posterior división para obtener el valor medio.

El último paso será guardar estos valores medios en una matriz que será usada para realizar la posterior clasificación. [8] [9]

Tabla 5-2. Valores medios de cada canal. Sombreado: Claro: Entrenamiento; Oscuro: Comprobación

	Semáforo	Canal rojo	Canal verde	Canal azul
SEMÁFOROS VERDES	1	41.6429	170.3214	107.4388
	2	44.0421	202.2645	122.4145
	3	246.9286	253.5357	250.1964
	4	91.3250	200.9000	171.3750
	5	104.5289	245.6694	234.2066
	6	26.0769	236.0128	219.9231
	7	123.0907	221.2608	198.0476
	8	48.6667	250.0000	243.3333
	9	28.8025	223.0370	196.8519
	10	6.3750	249.2857	211.8393
	11	99.2222	177.7778	175.8889
	12	110.3500	252.0625	226.9625
SEMÁFOROS ÁMBAR	13	229.4711	132.0368	52.0447
	14	247.1111	241.1111	76.2222
	15	229.2222	152.1111	83.1111
	16	204.7000	111.9368	58.2053
	17	241.4198	216.0000	67.4321
	18	247.9005	160.9198	83.1119
	19	209.8966	117.7276	37.1798
	20	188.0000	97.5556	37.5556
	21	229.5257	157.5435	15.0850
	22	252.5556	219.8889	50.3333
	23	251.9734	251.6701	41.2293
	24	253.1875	218.4375	93.2500
	25	248.5556	164.7778	15.2222
	26	234.0278	171.6944	54.9722
	27	247.0278	241.3889	0
	28	253.4444	253.4444	146.8889
	29	223.0744	152.9421	37.3554
SEMÁFOROS ROJOS	30	241.2500	48.5278	63.2778
	31	227.6522	57.1540	40.9291
	32	242.8842	69.4605	36.3395
	33	222.7143	29.3214	20.3214
	34	210.6348	74.6982	60.7378
	35	227.4111	108.7095	89.5929
	36	188.6991	38.6759	45.9691
	37	245.1875	88.3750	61.6250
	38	241.8166	41.0610	86.8450
	39	157.5833	27.7778	47.1389
	40	229.1605	51.0123	41.3580

5.2.3 Desarrollo del método

Para desarrollar la clasificación mediante el Método Perceptrón se ha creado una función cuyo desarrollo se explicará en la presente sección:

En primer lugar, de los valores de la *Tabla 5-2* se extraen aquellos valores que servirán para entrenar el clasificador. Estos valores son los que están sombreados de manera más suave.

Para la clasificación, y, observando los valores medios que se obtienen, es evidente que la zona crítica será aquella que discrimine las clases en función del canal rojo y el canal verde, puesto que el azul es de escaso interés en este caso.

Una vez que se extraen los valores para el entrenamiento se produce el desarrollo del método, que seguirá los siguientes pasos (todos estos pasos serán realizados dos veces con el fin de crear dos funciones de decisión: una para la discriminación de semáforos rojos y ámbar y otra para ámbar y verde):

- En primer lugar, se guardarán en una matriz *datos* los valores de los canales rojo y verde para el semáforo rojo y para el semáforo verde. Además, se inicializará el vector de parámetros *omega* con ceros.
- A continuación, empieza el proceso iterativo de entrenamiento:
    - Los semáforos rojos tienen un nivel de intensidad luminosa mayor en el canal rojo que el semáforo ámbar, y, sin embargo, tienen menor nivel de intensidad luminosa en el canal verde que el semáforo ámbar, por tanto, para una función discriminante, los semáforos rojos deberían situarse en la zona menor que cero ("por debajo" de la función) y los semáforos ámbar, al contrario.
    - La función de decisión está dada por el producto del vector de características y el vector de parámetros.
    - Si se está entrenando para el semáforo rojo, si la función es positiva (error), se penaliza el vector de parámetros *omega* haciéndolo más pequeño mediante el producto de un factor de corrección por el patrón que se esté evaluando.
        - El factor de corrección es un número fijo y positivo, que en el caso actual se ha establecido con valor 0.1, que se trata de un valor orientativo puesto que solo servirá para la corrección del vector de parámetros.
    - En el caso de que al probar un semáforo rojo el valor de la función de negativo, que sería lo correcto, no se penaliza, sino que se mantiene con el mismo valor.
    - En el caso de que se entrene para un semáforo ámbar, la función deberá ser positiva. Por tanto, si al evaluarla se produce un valor negativo habrá que penalizar el vector de parámetros aumentando su valor.
    - En cada iteración, el valor del vector de parámetros actualizado se compara con el anterior, es decir, se produce la diferencia entre el actual y el anterior, se eleva al cuadrado y se realiza el sumatorio de las componentes de dicha diferencia. La tolerancia será la raíz cuadrada de ese valor.
    - Si la tolerancia es menor que 0.001, se sale del bucle y se obtendrá un vector de parámetros definitivo.
- Una vez obtenido el vector de parámetros definitivo, la superficie de decisión se obtendrá con la ecuación que sigue:

$$z = \omega(1) * x(1)^2 + \omega(2) * x(2)^2 + \omega(3) * x(3)^2 + \omega(4) * x(4)^2 + \omega(5) * x(5)^2$$

5.2.4 Resultados obtenidos

Una vez se han creado las superficies de decisión para la separación de las tres clases (semáforos rojos, semáforos verdes y semáforos ámbar), se representarán a continuación las demostraciones gráficas de dicho procedimiento.

En primer lugar, se representará la distribución de las muestras seleccionadas para entrenar el clasificador.

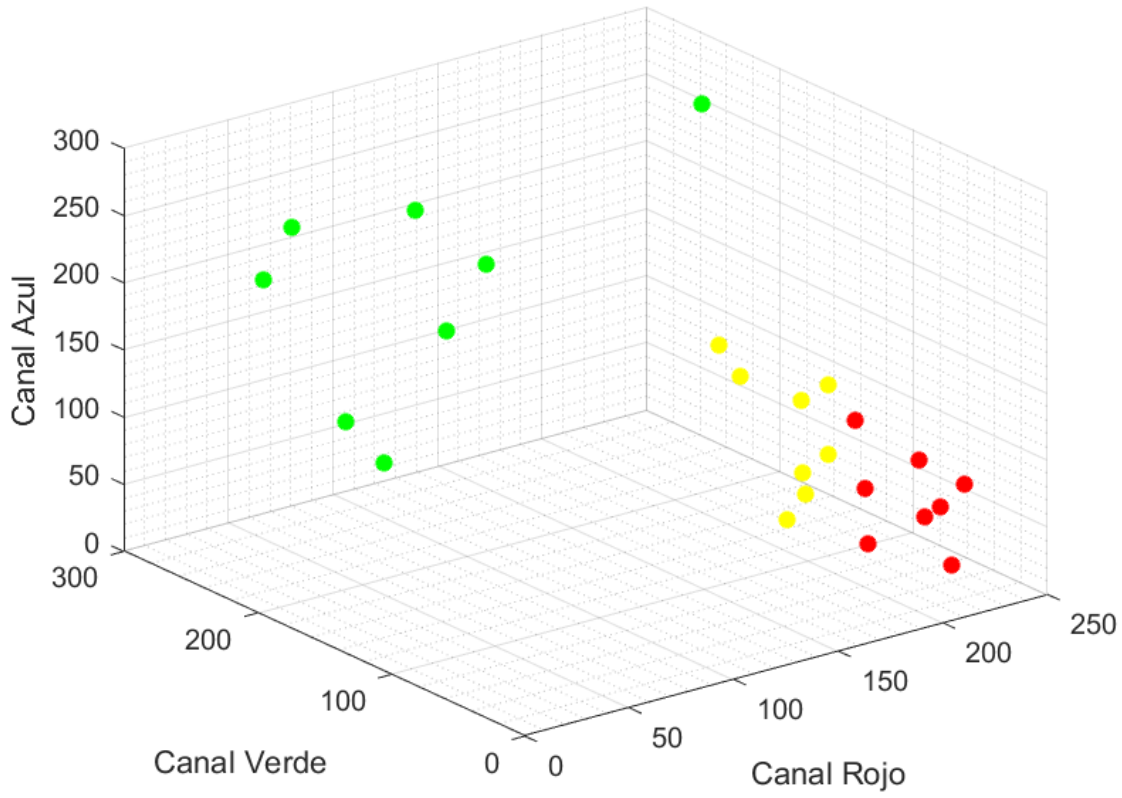


Figura 5-10. Representación en tres dimensiones de los patrones usados para entrenamiento

Se puede observar que, en la representación en tres dimensiones de dichos patrones, los semáforos verdes se separan a simple vista por cualquier función lineal del resto de los semáforos.

En capítulos anteriores se mostró que uno de los semáforos tenía todas sus componentes muy elevadas a causa de ser una segmentación de un semáforo verde muy brillante. De su extracción de la imagen real, se podía observar que se trataba de una imagen completamente blanca. Este semáforo se distingue claramente del resto gracias a sus elevadas componentes en cada canal, situando su representación en las zonas más elevadas del canal rojo, el canal verde y el canal azul.

A pesar de que el semáforo verde se pueda discriminar muy bien del resto, la clasificación de semáforos ámbar y rojos no se aprecia tan bien a simple vista. Es por ello por lo que para clasificarlas se ha hecho hincapié en los canales rojo y verde.

El canal azul es muy útil para la separación del semáforo verde respecto a los otros dos y, para la separación de los semáforos rojos y ámbar, a pesar de que se han usado los canales rojo y verde, el más significativo es el verde. Esto se puede apreciar en la *Figura 5-10*, en la que, aunque los dos colores de semáforos se encuentran muy cerca, existe cierta diferencia gracias a la variación en el canal verde, siendo los niveles de intensidad de dicho canal más elevado en los semáforos de color ámbar que en los de color rojo.

Este hecho se ejemplifica mejor en la imagen que aparece a continuación:

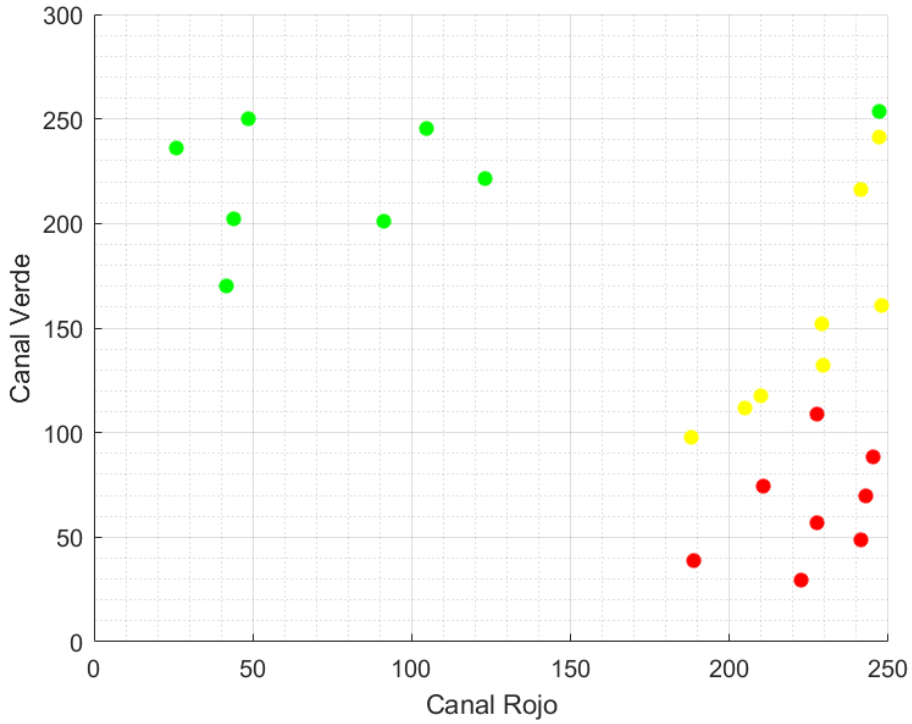


Figura 5-11. Representación en dos dimensiones (canales rojo y verde) de los semáforos patrón

En la figura anterior se puede observar que todas las clases son separables a partir de los canales rojo y verde. Esto se corresponde en cierto modo con los conceptos normalizados, en los que se conoce que la componente azul de los colores de interés es nula.

En este caso dicha componente no es nula, pero no aporta muy importante información.

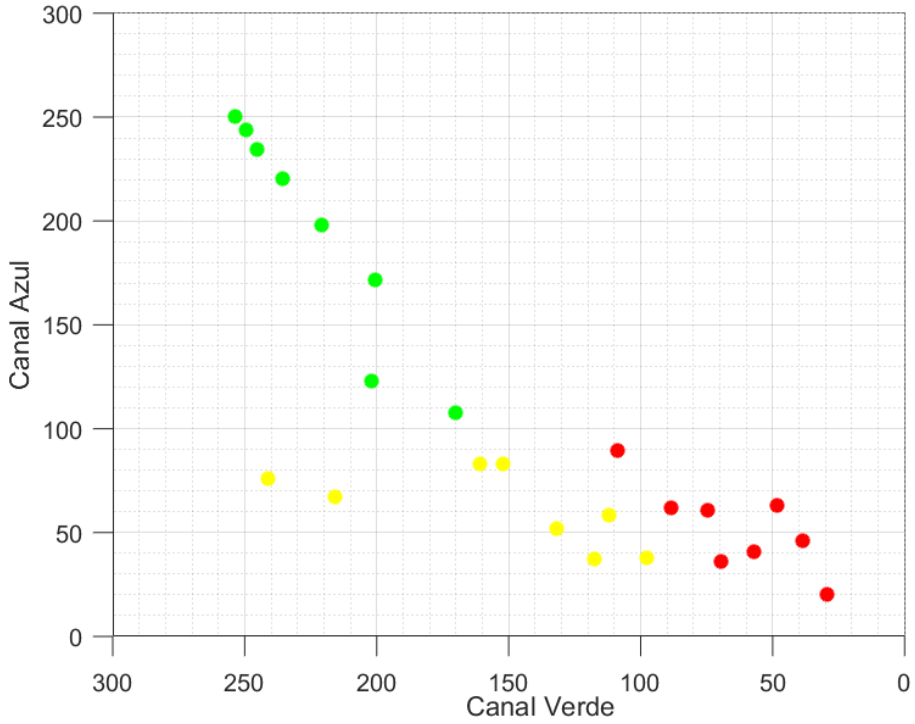


Figura 5-12. Representación en dos dimensiones (canales azul y verde) de los semáforos patrón

Además, como se puede observar en la imagen anterior, el canal azul no solo no aporta demasiada información, sino que además no es una buena característica para discriminar ni entrenar la función de decisión. Lo mismo ocurre si en lugar de representar el canal verde frente al canal azul se representa el canal rojo frente al azul. El resultado es que, de nuevo, las clases no son separables, pero en este caso con más hincapié aún, pues las clases que se corresponden con semáforos ámbar y rojo se solapan completamente:

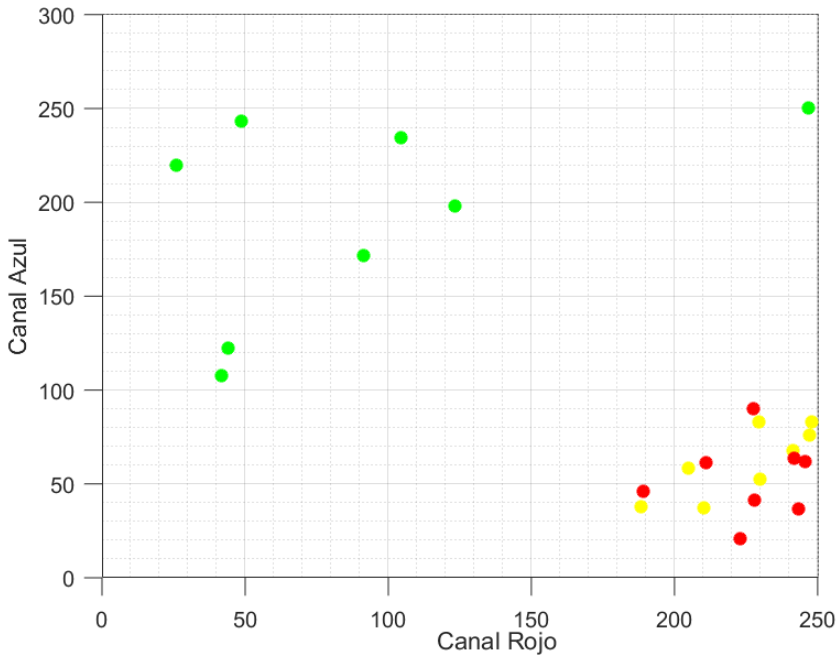


Figura 5-13. Representación en dos dimensiones (canales azul y rojo) de los semáforos patrón

El siguiente paso ha sido representar las superficies de separación en tres dimensiones con el fin de visualizar los resultados obtenidos con este método

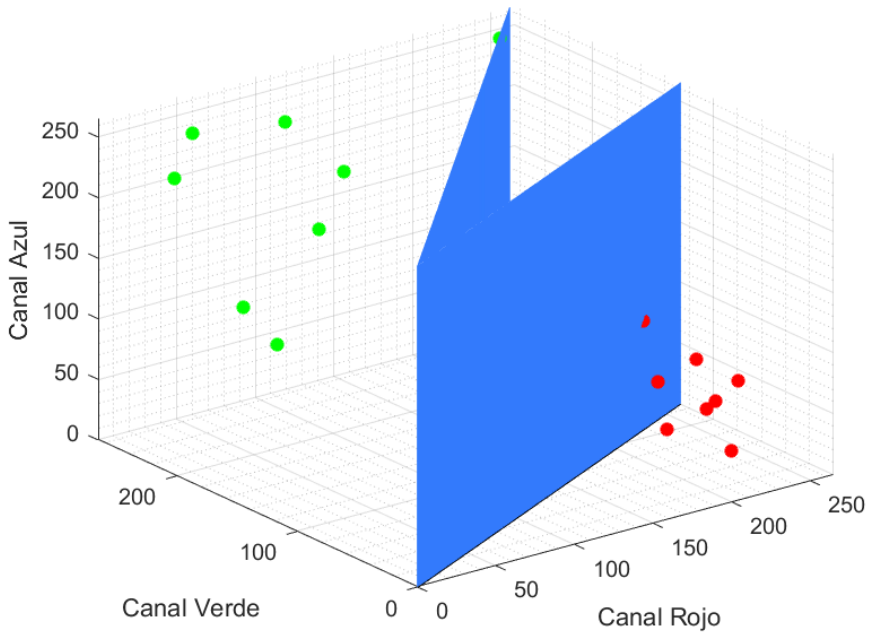


Figura 5-14. Representación de las superficies de separación

Estas superficies se representan como planos que a simple vista son lineales, pero, la realidad es que se trata de superficies no lineales, puesto que así se ha programado el Método Perceptrón, que presentan un radio tan grande que en la zona de interés parecen lineales.

Para una mejor observación de los resultados es conveniente una representación en dos dimensiones:

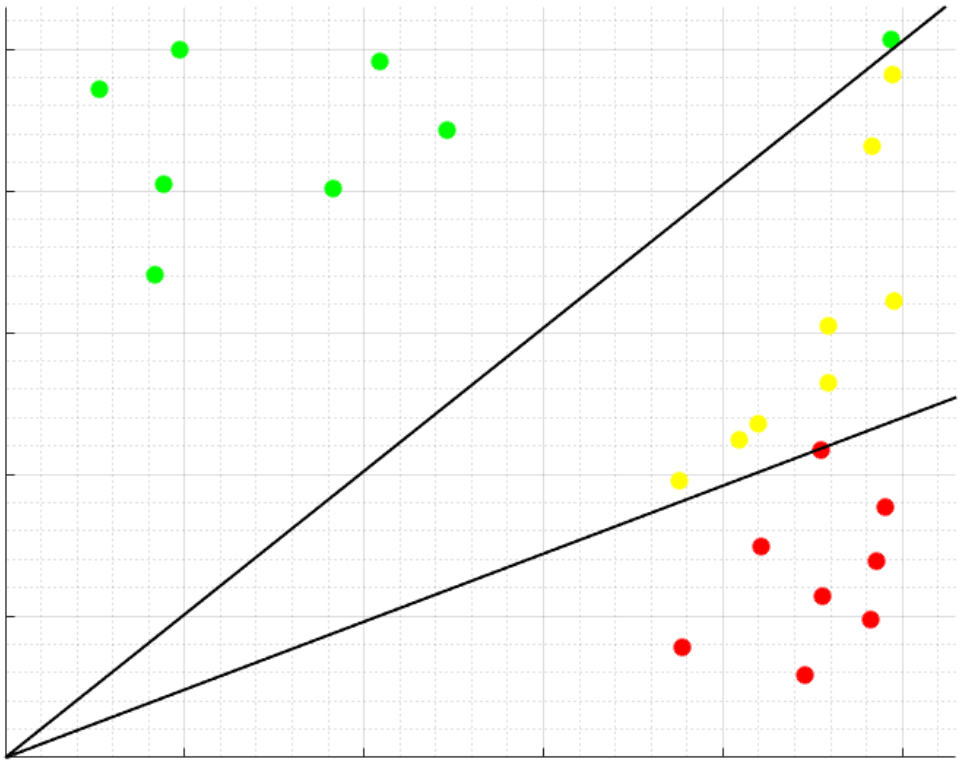


Figura 5-15. Canal rojo frente a canal verde con superficies de separación

En ella se puede observar como las tres clases quedan completamente separadas mediante dos planos. Existe uno de los semáforos que puede llevar a confusión puesto que se encuentra en la frontera de la función de decisión que separa las clases correspondientes a los semáforos ámbar y rojo. Sin embargo, es posible comprobar que se sitúa en la franja correcta evaluando la función para ese semáforo.

Para ello se ha creado una interacción entre el usuario y MATLAB a partir de la ventana de comandos, de forma que, por pantalla se le solicita al usuario el semáforo que se quiera comprobar y evalúa la función según sus niveles de intensidad en cada canal, de la forma que sigue:

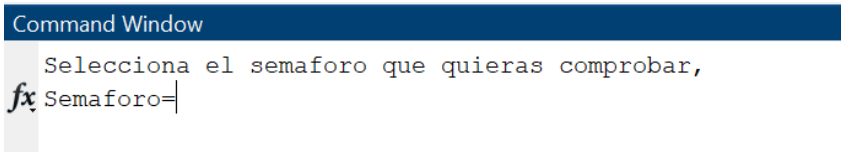


Figura 5-16. Ventana de comandos en MATLAB

Una vez solicitado el semáforo que se desea comprobar, se introduce aquel que sea de interés. En este caso con la herramienta *Data Tips* en MATLAB se observa que el semáforo que se quiere comprobar tiene unas componentes RGB de (227.411, 108.709, 89.5929), que se corresponde con el semáforo 35, por tanto:

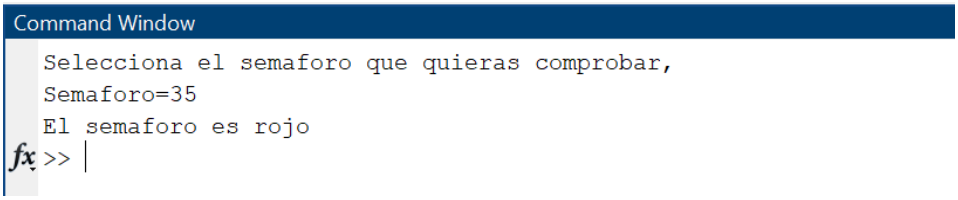


Figura 5-17. Comprobación de resultados

Hecho esto, se comprueba que dicho semáforo se encuentra dentro de la zona correcta, puesto que, evaluando la función en los niveles de cada canal que se corresponden con él, se obtienen los resultados apropiados.

Por consiguiente, para comprobar la fiabilidad del método planteado se evaluarán todos los semáforos de la misma forma, obteniendo los siguientes resultados:

Tabla 5-3. Resultados correspondientes al Método Perceptrón

Semáforo	Color real	Clasificación por correlación
9	Verde	Verde
10	Verde	Verde
11	Verde	Verde
12	Verde	Verde
21	Ámbar	Ámbar
22	Ámbar	Ámbar
23	Ámbar	Ámbar
24	Ámbar	Ámbar
25	Ámbar	Ámbar
26	Ámbar	Ámbar
27	Ámbar	Ámbar
28	Ámbar	Ámbar
29	Ámbar	Ámbar
38	Rojo	Rojo
39	Rojo	Rojo
40	Rojo	Rojo

De los datos obtenidos se puede extraer como conclusión que el método cumple de forma óptima con los resultados buscados, puesto que, aplicando el Indicador de Bondad, se obtiene un porcentaje de error nulo en las comprobaciones, o lo que es lo mismo, un porcentaje de acierto del 100% en la clasificación de todos los semáforos.

5.3 Resumen del capítulo

A modo de conclusion, se destacará que, puesto a la existencia de numerosos métodos de clasificación, ha sido necesaria la implementación de dos de ellos para llegar a resultados óptimos.

Como primer paso se ha implementado un método de clasificación por correlación. El problema principal planteado ha sido la elección de un patrón correcto que se adecúe bien a las muestras de ruido que aportan la realidad de las imágenes tomadas. Tras probar el método con plantillas sintéticas y de valores normalizados, con la posterior llamada de atención de que no era un método fiable por sus malos resultados, se ha procedido al cálculo de coeficientes de correlación empleando como plantilla una de las segmentaciones tomadas directamente de las imágenes reales. El resultado de esta decisión ha sido, como cabe esperar, de resultados que no se corresponden en gran medida a los esperados, así que, la última opción ha sido la elección de un patrón usando para su creación varias imágenes de semáforos segmentados, se ha calculado la media de dichos semáforos en cada uno de los canales y para cada píxel, y para conseguir que sea aún más robusto, se ha aplicado un filtrado en la imagen obtenida.

Aplicando el coeficiente de correlación usando como patrón el último método especificado, los resultados obtenidos han sido muy poco concluyentes, puesto que se ha obtenido un índice de acierto en la clasificación de un 63.16%.

Esta ha sido la justificación para implementar un nuevo método con el fin de obtener mejores resultados. Dicho procedimiento efectuado ha sido el Método del Perceptrón, que basa su desarrollo en crear funciones de decisión que distinga entre regiones que se corresponden a las diferentes clases posibles.

Para ello se ha entrenado el método con un 60% de los semáforos segmentados, obteniendo un vector de parámetros para cada una de las separaciones (semáforo ámbar de rojo y semáforo ámbar de verde).

Una vez obtenidas las funciones de decisión con el vector de parámetros anteriormente mencionado, se ha procedido a la comprobación del 40% restante de semáforos que se ha reservado para realizar evidenciar el método, obteniendo unos resultados de acierto del 100% y, descartando por tanto la posibilidad de que se encuentre un mejor método.

6 CONCLUSIONES Y LÍNEAS FUTURAS

A lo largo de todos los capítulos que conforman el proyecto, se ha desarrollado un proceso capaz de detectar en cualquier tipo de imagen, con diversas condiciones de iluminación en la escena, distancia al objetivo, ruido, etc. el círculo que conforma la luz de un semáforo, es decir, su zona de interés. Además, se ha conseguido la clasificación de todas las muestras obteniendo unos resultados muy satisfactorios.

A modo de conclusión se puede extraer que, la correcta segmentación de la zona de interés de la imagen se convierte en un factor de vital importancia el tratamiento previo de la imagen.

Se ha especificado a lo largo de todo el trabajo, la necesidad de tener en cuenta que en todo momento se ha realizado un trabajo con imágenes reales, la mayoría capturadas mediante un teléfono móvil, en las que el ruido existente y la puesta en escena de elementos aleatorios dificultan mucho la detección de dichos semáforos.

Para ello los pasos previos han sido realizar una segmentación por umbralización. De este modo ha sido posible descartar de la imagen aquellos objetos que, por sus características de niveles de intensidad luminosa en cada canal, no eran significativos en la imagen tratada. Cabe destacar que, esta umbralización no ha sido suficiente para la detección de las luces del semáforo puesto que no se considera un método con una fiabilidad tan elevada como para basar todo el proyecto en él.

Una vez que se han descartado aquellas zonas no representativas de la imagen, el siguiente paso ha sido aplicar métodos morfológicos en la imagen segmentada. Este procedimiento se realiza con el fin de mejorar el aspecto de los posibles círculos, disminuyendo el ruido que pueda estar presente en sus bordes, o los huecos formados en el mismo. Concretamente se ha aplicado una apertura, siendo esta una dilatación seguida de una erosión.

El último paso de tratamiento de la imagen se ha tratado de la obtención de una imagen binaria de bordes. Este proceso es necesario puesto que para aplicar la detección de círculos mediante la Transformada de Hough es necesario que la imagen a tratar sea una imagen binaria, y que, además, tenga el borde del círculo bien diferenciado.

Tal y como se ha indicado anteriormente, el ultimo paso ha sido la detección del círculo objetivo mediante la Transformada de Hough. Esta forma de detección se caracteriza por su robustez en la localización de dichos círculos.

Obteniendo resultados gratos en los 40 semáforos que se han probado, el paso final ha sido la clasificación de estos según su color. Esto es debido a que, en un sistema implementado en vehículos en el que su propósito sea la detección de semáforos, no solo los tendría que localizar, sino también comprobar cuál es la acción que debe realizar el vehículo ante su detección. Para ello se discriminarán los semáforos con respecto a algún método de clasificación existente.

El primer método probado ha sido mediante correlación, a lo que se ha añadido el problema de cómo elegir un patrón correcto que garantice unos resultados beneficiosos debido a la disparidad entre imágenes. Tras probar

diversos patrones la conclusión más lógica ha sido que la plantilla que proporciona mejores resultados consiste en aplicar una media de una muestra de semáforos reales con posterior filtrado, con el fin de suavizar su contenido. Tras probar el método se concluye que los resultados obtenidos no proporcionan una clasificación adecuada que cumpla con lo que se busca. Es por ello por lo que se ejecuta un método de clasificación mediante discriminantes lineales o también llamado Método Perceptrón, que consiste en crear funciones discriminantes y agrupar las clases en regiones. Una vez obtenido un vector de parámetros mediante un proceso iterativo, se han obtenido las superficies de decisión, comprobando que, con las muestras existentes se obtienen unos resultados con un índice de acierto del 100%.

En conclusión, con los métodos anteriormente planteados, se puede recapitular y afirmar que se han llegado a los resultados buscados con la correcta detección y clasificación de los semáforos.

En cuanto a líneas futuras de investigación, se podría desarrollar un método de detección en tiempo real, es decir, mediante una cámara de vídeo instalada en un vehículo y configurada en MATLAB, se podría ver el tratamiento de las imágenes en directo, sustituyendo el tratamiento de una sola captura por un sistema que use un conjunto de *frames* del vídeo.

Además, en el caso de resultados satisfactorios en el método, se podría implementar el sistema en un vehículo y buscar las acciones correctas del mismo según se detecte un semáforo u otro, es decir, plantear el caso de un vehículo circulando y que disminuya su velocidad o frene en función de si encuentra una luz de semáforo u otra.

REFERENCIAS

[1] Á. Romero-Acero, A. Marín-Cano y J. A. Jiménez-Builes, «Sistema de clasificación por visión artificial de mangos tipo Tommy,» *Revista UIS Ingenierías,* vol. 14, nº 1, pp. 21-31, 2015.

[2] A. González Marcos, F. J. Martínez de Pisón Ascacíbar, A. V. Pernía Espinoza, F. Alba Elías, M. Castejón Limas, J. Ordieres Merés y E. Vergara González, «Técnicas y Algoritmos básicos de Visión Artificial,» Universidad de La Rioja, La Rioja, 2006.

[3] M. C. Puell Marín, «Óptica Fisiológica: El sistema óptico del ojo y la visión binocular,» Universidad Complutense de Madrid, Madrid.

[4] C. Cattaneo, L. Larcher, A. Ruggeri, A. Herrera y E. Biasoni, «Métodos de Umbralización de Imágenes Digitales Basados en Entropía de Shannon y Otros,» *Asociación Argentina de Mecánica Computacional,* vol. XXX, pp. 2785-2805, 2011.

[5] «MathWorks,» [En línea]. Available: https://es.mathworks.com/help/images/ref/strel.html. [Último acceso: 21 Junio 2022].

[6] L. Á. Ruiz Fernández, «Aplicación de Filtros Morfológicos en Imágenes,» Universitat Politècnica de València, Valencia.

[7] J. A. Cortés Osorio, W. Ardila Ureña y J. A. Mendoza Vargas, «Técnicas alternativas para la conversión de imágenes a color a escala de grises en el tratamiento digital de imágenes,» *Scientia et Technica,* vol. XVII, nº 47, pp. 207-212.

[8] J. G. Jiménez, Visión por Computador, Málaga: ITP-Paraninfo, 1999.

[9] J. Vélez Serrano, A. B. Moreno Díaz, Á. Sánchez Calle y J. L. Esteban Sánchez-Marín, Visión por Computador, Madrid: Dykinson, 2003.

[10] L. Canul-Arceo, J. López-Martínez y L. Narváez-Díaz, «Algoritmo rápido de la transformada de Hough para detección de líneas rectas en una imagen,» *Programación Matemática y Software,* vol. 7, nº 2, pp. 8-13, 2015.

[11] «MathWorks,» [En línea]. Available: https://es.mathworks.com/help/images/ref/medfilt2.html. [Último acceso: 25 6 2022].

[12] «MathWorks,» [En línea]. Available: https://es.mathworks.com/help/images/ref/normxcorr2.html#bvh5vek-7. [Último acceso: 25 6 2022].

[13] Apuntes de la asignatura Sistemas de Percepción, Universidad de Sevilla.