

Uso de Segmentación para detección de señales de tráfico en vídeo

Ricardo Díaz-Arroyo, Eduardo Font
 rickydiaz05@estudiantec.cr, eduardofontc@hotmail.com
 Área de Ingeniería Mecatrónica
 Tecnológico de Costa Rica

Resumen

El presente documento detalla el planeo y solución de la tercera tarea del curso de Sistemas de Visión. Utilizando los conocimientos de segmentación por medio de k-means, además de otros tipos de segmentación adquiridos en el curso, se implementó un sistema que encuentra y señala señales de tránsito azules en un vídeo suministrado del recorrido de un auto en la calle de una ciudad. Para esto, se utilizó el lenguaje de programación Python, junto a bibliotecas para uso científico y matemático como Numpy y OpenCV 2.

Palabras clave

Segmentación, K-Means, máscara, HSV.

I. INTRODUCCIÓN

La Visión Artificial es un campo interdisciplinario que "tiene como finalidad la extracción de información del mundo físico a partir de imágenes, utilizando para ello un computador" [1]. En el caso de la tarea sobre la cual se elabora este documento, se pedía diseñar una solución al problema de identificar y señalar señales de tránsito azules en un vídeo del recorrido de un auto en una calle de una ciudad. Para extraer las señales, se requería encontrar los tonos de azul característicos de estas señales en el cuadro deseado del vídeo, así como sus geometrías específicas. Por medio de las técnicas de segmentación (por contenido y avanzada) desarrolladas a lo largo del curso de Sistemas de Visión, se planteó una solución y un prototipo para el problema dado.

II. REQUERIMIENTOS

II-A. Entorno

El entorno de trabajo es el video dado para analizar y extraer las señales de tránsito azules en un recorrido de auto sobre una calle en la ciudad.

II-B. Identificación de las señales

"Deben usar ustedes fundamentalmente el método de segmentación por medio de "K-means" (adicionalmente pueden usar cualquier otra técnica vista en el curso, si les sirve de ayuda, pero sólo para refinar los resultados provistos por el k-means) para llevar a cabo la detección, en el fragmento de video proporcionado en la carpeta OneDrive del curso, de las señales de tráfico con fondo azul". [2].

II-C. Desarrollo y método

- "Tienen ustedes que desarrollar una estrategia de segmentación de forma tal que hagan un uso de la técnica "k-means" que demuestre un equilibrio entre coste computacional, la trivialidad de la clasificación y el resultado solicitado. Deben ustedes desarrollar un criterio para establecer el número adecuado de centroides a utilizar, demostrando por medio de las pruebas correspondientes la validez del mismo". [2].
- "La estrategia ha de estar adecuadamente explicada en términos de la teoría propia del curso, y se deben documentar y explicar las diversas pruebas realizadas, las cuales deben responder a los criterios que se deducen del texto de este enunciado, analizando y evidenciando también las pruebas que no hayan sido exitosas o con el nivel deseado de rendimiento". [2].

II-D. Aclaración de implementación

"Su solución puede funcionar sobre el stream de video en tiempo real, o bien puede analizar un único cuadro (frame) del mismo, siempre y cuando dicho frame pueda elegirse en cualquier momento del tiempo por medio de una pulsación de teclado". [2].

III. METODOLOGÍA

III-A. Descripción de la solución

La solución a este problema se basa en el uso de la técnica de agrupamiento de información conocida como K-Means. Esta técnica se utiliza para juntar la información en grupos y aplicarles una etiqueta (normalmente el centroide o el valor central entre los elementos del grupo), lo que permite analizar con mayor facilidad los elementos que muestran un cierto parentesco en un criterio deseado. En el caso de las imágenes, esta agrupación se realiza por color, agrupando en una cantidad de grupos establecidos a estos y devolviendo una imagen donde todos los píxeles que pertenecen a un cluster o grupo se les asigna de color el valor del centroide. A partir de esto, se aplica una máscara de un rango de colores, binarizando la imagen en segmentos en cuyo k-menas está dentro del rango de color y en los segmentos que esto no ocurre. Una vez obtenida la binarización, se detectan contornos y de filtran en base al área y geometría para posteriormente señalarlos en la imagen original. En la Figura 1 se muestra el diagrama de bloques de la solución para el problema.

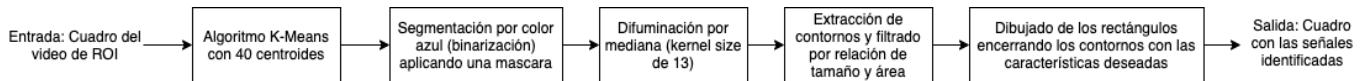


Figura 1: Diagrama de bloques de la solución planteada. Elaboración propia.

III-B. Selección del frame del vídeo

La aplicación corre el vídeo (de la ROI) continuamente al abrirse. Para encontrar las señales de tránsito en un cuadro, se presiona la barra espaciadora y en ese instante se pausa el vídeo, se realizan los cálculos respectivos y se muestran estos sobre el cuadro en el que el vídeo está pausado. Para seguir con la reproducción del vídeo, se vuelve a presionar la barra espaciadora y para cerrar el programa se presiona la tecla “q”.

IV. RESULTADOS Y ANÁLISIS

IV-A. ROI

Para realizar el análisis se optó por reducir el tamaño de la imagen original del vídeo que se puede apreciar en la figura 2. Esto debido a que las señales en fondo azul que se desean analizar están en una región del vídeo específica. Esto se cumple para todo el largo del vídeo e incluso para cualquier otro vídeo tomado en el mismo auto, ya que lo que se le retira a la imagen es la parte superior donde se observa el cielo y las partes altas de edificios.



Figura 2: Imagen del vídeo original y la ROI a utilizar. Cuadros extraídos de [2]

A partir de esta imagen de video completo se generó una región de interés(**ROI**) que se puede observar en la figura 2 en la parte inferior

IV-B. Implementación del K-Means

Para la implementación del método de K-Means se tuvieron que tomar ciertas cosas en consideración. Primeramente sabiendo que el método K-means es un método iterativo y dependiendo de la cantidad de iteraciones y clusters que se requieran puede llegar a ser un método computacionalmente pesado y lento para implementar en todo un vídeo. Por esto y basándose en la aclaración de la implementación de los requerimientos se realizó la aplicación del método K-Means a un solo frame del vídeo para poder liberar un poco de capacidad computacional. Esto se implementó de manera que cada vez que se corra el programa se reproduce el vídeo y al presionar la barra espaciadora del teclado el vídeo es pausado y se analiza el frame en el que se está durante la pausa, según la aclaración de los requerimientos. Otra de las consideraciones que se debió tener fue una selección de clusters adecuada ya que si la cantidad de clusters es muy pequeña no se observan las señales de color azul y si la cantidad de clusters es muy alta se requiere de mayor poder computacional y mayor tiempo para la entrega de la respuesta, además de tener mayor cantidad de todos de un color lo cual dificulta una implementación de segmentación por color. Por ejemplo en la Figura 3 se puede observar que al usar 10 clusters no se identifican los tonos de azul en las señales mientras que con 40 sí. El valor de 40 clusters fue escogido para poder obtener las señales azules con el menor valor posible para no generar un tiempo de respuesta mayor. Además de haber limitado el K-Means a 10 iteraciones. En la figura 4 se puede observar la salida obtenida para una implementación del Método K-Means con 40 clusters, para un frame del vídeo.



Figura 3: Comparación 40 clusters (arriba) y 10 clusters (abajo) K-Means. Elaboración propia, basado en vídeo extraído de [2].



Figura 4: Implementación K-means 40 clusters. Elaboración propia, basado en vídeo extraído de [2].

IV-C. Segmentación por color y binarización en máscara

A partir de la imagen obtenida luego del método de K-Means se realiza una segmentación por color en la cual se establece un umbral de distintos tonos de azul desde un celeste claro con valores (100,110,30) HSV (usa el formato de HSV de OpenCV 2) hasta un azul oscuro con valores (170,255,180) HSV. Esta segmentación por color permite filtrar la mayoría de todos los de azul cercanos a los de las señales. Luego de esto se realiza una máscara con una habilitación muy fuerte en la que se representa solamente los valores obtenidos luego de la segmentación por color. En la figura 5 se puede observar la máscara obtenida luego de la finalización por color para los valores del umbral en HSV antes mencionados.

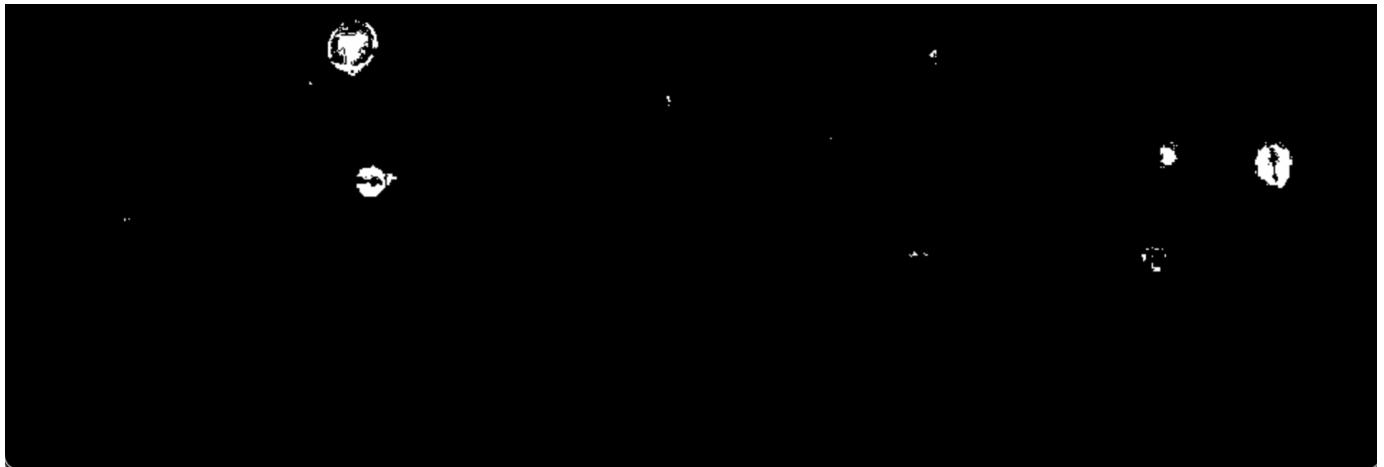


Figura 5: Máscara binarizada de segmentación por color. Elaboración propia, basado en video extraido de [2].

Se obtienen algunos errores en algunos frames del vídeo ya que hay otros objetos con tonos de azul similares a los de las señales que por ende están dentro del umbral utilizado. Por ejemplo en la figura 6 se observa en el recuadro de la izquierda la detección de la parte trasera de un autobús que contiene un tono dentro del umbral seleccionado, que es cercano al color de las señales.



Figura 6: Error de similitud de color con bus. Elaboración propia, basado en vídeo extraído de [2].

IV-D. Contornos e Identificación de las señales

Se procede luego de obtener los objetos que se observaron en la figura 5 a tomar los contornos encontrados y a generarles una caja a su alrededor para denotar las señales encontradas. Se realiza un filtrado por relación de tamaño y por el área del contorno encontrado en la máscara, de esta forma el ruido de objetos pequeños o muy grandes no se tomarán para la identificación de señales. La caja es denotada por las coordenadas x y y dentro de la imagen general del la ROI y se denota con un color verde como se puede observar en las figuras 7, 8, 9. Para distintas imágenes.



Figura 7: Ejemplo identificador de señales encontradas. Elaboración propia, basado en vídeo extraído de [2].

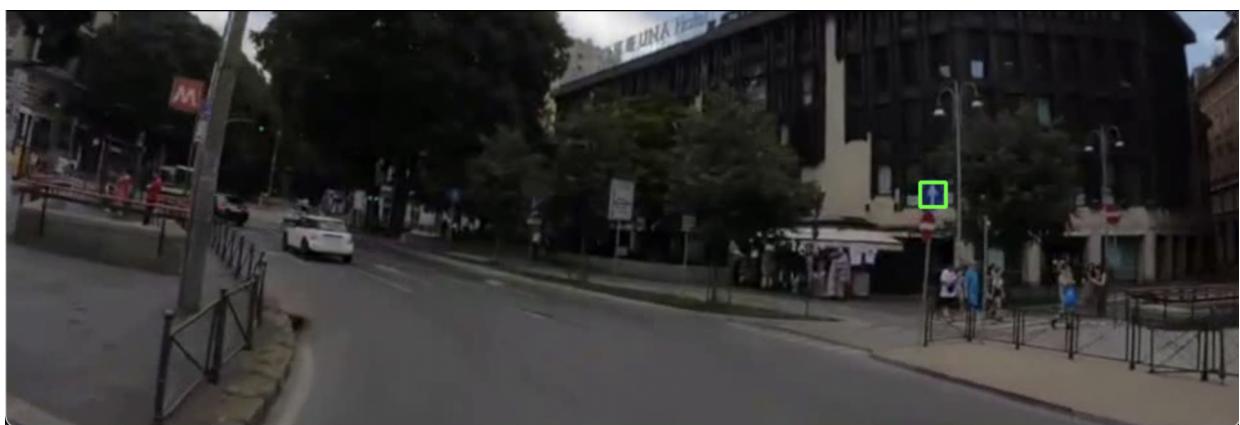


Figura 8: Ejemplo identificador de señales encontradas. Elaboración propia, basado en vídeo extraído de [2].



Figura 9: Ejemplo identificador de señales encontradas. Elaboración propia, basado en vídeo extraído de [2].

Al utilizar el filtrado por relación de tamaño y por el area en la máscara se mejora el problema antes mencionado acerca del bus. Se aclara que es una mejora ya que como se puede observar en la Figura 10 si se realiza el análisis a un frame en el que el bus aún se vea lo suficientemente pequeño por la distancia a la que está, será tomado como si fuera una señal azul. Pero en la figura 11 se puede observar que al tener un frame en el que el bus se vea más grande, este será filtrado por el tamaño y no se tomará como un a señal azul.



Figura 10: Ejemplo de un falso verdadero para Bus. Elaboración propia, basado en vídeo extraído de [2].



Figura 11: Ejemplo Bus filtrado por tamaño de área. Elaboración propia, basado en vídeo extraído de [2].

La solución planteada presenta ciertas fallas. Estas fallas ocurren usualmente por 2 motivos. El primero es que depende de cuando sea tomado cierto frame, por cómo se ve la imagen y como están compuestas las señales, se puede tener una señal que en la máscara sea tomada como 2 señales azules. Por ejemplo en la Figura 12 se puede observar cómo la señal azul es tomada como 2 señales aparte debido a las flechas blancas que tiene dentro de ella, sin embargo no ocurre para todos los frames en

los que se tenga la misma señal, varía de uno a otro por la cercanía de las flechas a los bordes si es tomada como 2 señales azules o una.



Figura 12: Ejemplo de fallo tomado como 2 señales distintas. Elaboración propia, basado en vídeo extraído de [2].

El segundo motivo de falla se debe a la poca claridad que posee alguna señal azul. Por ejemplo, en la Figura 13 se tiene una señal que está cercana a unos arboles, lo que hace que no tenga suficiente claridad proveniente del sol para ser detectada como una señal azul.



Figura 13: Ejemplo de fallo por objeto poco iluminado. Elaboración propia, basado en vídeo extraído de [2].

V. CONCLUSIONES Y RECOMENDACIONES

- Al utilizar el Método de K-Means se debe llegar a un balance entre el coste computacional, un agrupamiento utilizable y no redundante. Para obtener lo requerido
- Para mejorar la detección de las señales de tránsito, se pueden utilizar técnicas como redes neuronales o template matching sobre los cuadros señalados (como ROIs) para filtrar con mayor exactitud cuales regiones realmente contienen señales de tránsito y hasta poder clasificarlas.
- Es importante revisar correctamente la documentación de las funciones que se desean utilizar de una biblioteca, pues existen variaciones entre diferentes nomenclaturas y formatos (por nombrar algunos). Por ejemplo, OpenCV 2 utiliza los valores HSV con valores entre 0 y 179 para el H y entre 0 y 255 para el S y el V, por lo que se pueden llegar a errores que no se comprenden debido a ingresar valores sin tomar estas consideraciones.

REFERENCIAS

- [1] J. Vélez et al, Visión Por Computador. 2003.
 [2] J. Crespo, TAREA PARCIAL EVALUABLE NÚMERO 3: SEGMENTACIÓN AVANZADA. 2021.