
Técnicas de visión computacional para la segmentación de carriles

TECNICAS DE VISION COMPUTACIONAL PARA LA SEGMENTACIÓN DE CARRILES

Erick Rodríguez Hernández

erickcarman822@hotmail.com

Leslie Janet Carboney Palafox

leslie.cidetec@gmail.com

Maestría en Tecnología de Cómputo

Centro de Innovación y Desarrollo Tecnológico en Cómputo

Resumen

El presente trabajo muestra diferentes técnicas de procesamiento de imágenes para poder segmentar las líneas presentes dentro de una carretera que sirven para delimitar los carriles por los cuales debe circular un automóvil. Entre las técnicas se encuentra la aplicación de un filtro gaussiano, aplicación de detección de bordes con Canny, la utilización del método de Hough entre otros.

Introducción

Con el rápido incremento del tránsito en la ciudad, la seguridad al conducir se convierte en un punto importante. El salir de los carriles por los que se transita causa alrededor del 30% de los accidentes en carretera, la mayoría de estos accidentes son causados por la distracción y fatiga del conductor.

Los sistemas de detección de carriles son muy útiles para evitar esta clase de accidentes, siendo la seguridad su propósito general. Están hechos para que el sistema detecte cuando el vehículo está saliendo de la carretera y advertirle al conductor y es una tarea crítica que deben realizar los vehículos autónomos [1].

En este artículo se utilizarán algunas técnicas conocidas de visión por computadora para reconocer las marcas de carril en las carreteras.

Desarrollo

Para realizar el procesamiento de imágenes se realizaron los programas en Python usando la librería OpenCV [2], las técnicas que se utilizaron se describen a continuación:

- La imagen original, utilizada para realizar las pruebas para el procesamiento, mostrada en la Figura 1, se convierte al espacio de color HLS.
- La imagen se convierte a escala de grises.
- Se crea una máscara de píxeles amarillos y blancos.
- Para reducir el ruido en la imagen se aplica un filtro Gausiano.
- Se realiza la detección de bordes con Canny.
- Se crea una máscara que nos muestre solo la región de interés en la imagen.
- Y finalmente se dibujan líneas sobre la imagen que son encontradas por el método de Hough.



Figura 1. Imagen de prueba para aplicar las técnicas de procesamiento.

La entrada en cada paso es la línea de salida del paso anterior, por ejemplo, se aplica la detección de los bordes con Canny a la imagen que previamente se le aplicó un filtrado Gaussiano.

A. Conversión a diferentes espacios de color

Teniendo la imagen original se procedió a explorar la visualización de la imagen en otro espacio de color, como lo es el Hue, Saturation, Lightness (Matiz, Saturación, Luminosidad) o HSL por sus siglas en inglés, que permite realizar una mejor segmentación de las líneas de color blanco y amarillo. La Figura 2 muestra la visualización de la imagen una vez que se convirtió de RGB a HSL.

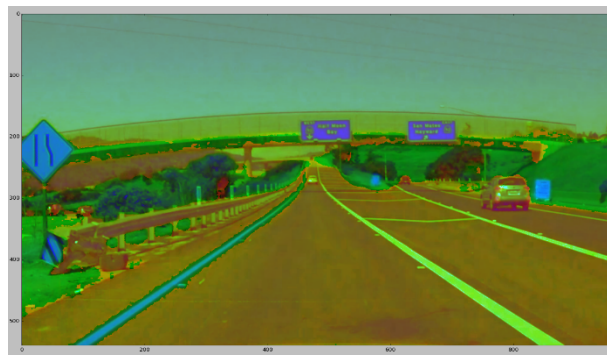


Figura 2. Imagen procesada en el espacio de color HSL.

B. Escala de grises

Es de interés la detección de líneas blancas o amarillas en una imagen de una carretera, éstas van a mostrar un contraste particularmente grande cuando la imagen se encuentre en escala de grises, lo cual ayudará a realizar un tratamiento más sencillo de la imagen.

C. Máscara de píxeles amarillos y blancos

Para poder obtener un segmentado de las líneas más eficiente, se realizó un enmascaramiento de la imagen procesada en HSL, tomando en cuenta los valores superiores e inferiores para poder detectar píxeles en colores amarillos y blancos. Una vez procesada la máscara, se aplicó a la imagen en escala de grises, como se muestra en la Figura 3, ayudando aún más a reducir el ruido dentro de la imagen.

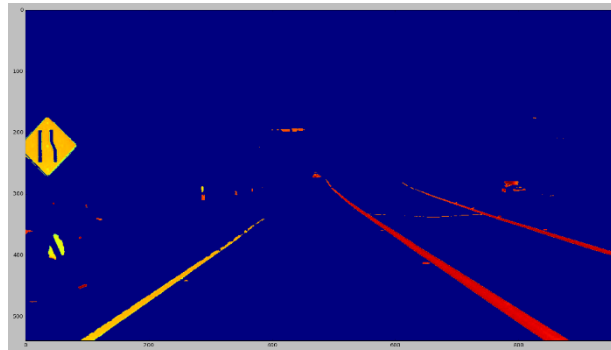


Figura 3. Imagen con el enmascaramiento de pixeles blancos y amarillos.

D. Filtrado Gaussiano

El filtrado Gaussiano es una técnica de pre-procesamiento utilizado para suavizar los bordes de una imagen tratando de reducir el ruido. Dicho filtro se aplica intentando reducir el número de líneas que se han detectado, teniendo en cuenta que no es adecuado aplicar un valor muy grande en el valor de su Kernel, si no será más difícil poder detectar los bordes que delimitan la carretera de la imagen.

E. Detección de Bordes con Canny

Ahora que se ha procesado lo suficiente la imagen, se puede aplicar un Detector de Bordes de Canny [3], cuya función es identificar los bordes de una imagen y descartar todos los demás datos. Es posible encontrar el gradiente de borde y dirección para cada pixel dada la siguiente ecuación:

$$Edge_{Gradient}(G) = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$Angle(\theta) = \tan^{-1} \left(\frac{G_y}{G_x} \right)$$

La implementación requiere de dos parámetros de umbral, uno alto y uno bajo que determinan si se incluye o no al borde en cuestión. Un umbral captura la intensidad de cambio de un punto dado. Cualquier punto más allá del umbral alto se incluirá en la imagen resultante, mientras

que los puntos intermedios entre los valores de umbral dados, sólo se incluirán si están próximos al umbral alto. Los bordes que están por debajo del umbral bajo se descartan tal como lo muestra la Figura 4.

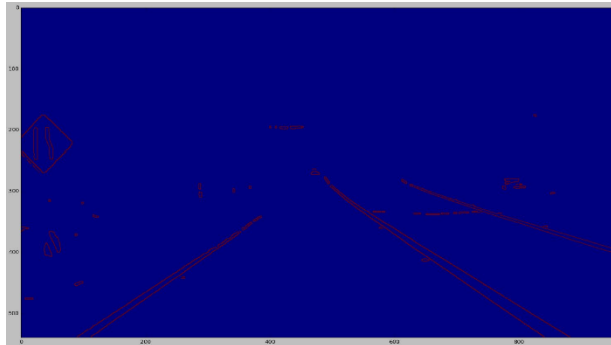


Figura 4. Imagen donde se muestra la detección de bordes utilizando Canny.

F. Región de interés

El siguiente paso es determinar una región de interés y descartar cualquier línea que se encuentre fuera de este polígono. Es decir, dadas las imágenes procesadas se determina una región donde se desea que los carriles permanezcan y que lo demás sea descartado, tal como se muestra en la Figura 5.

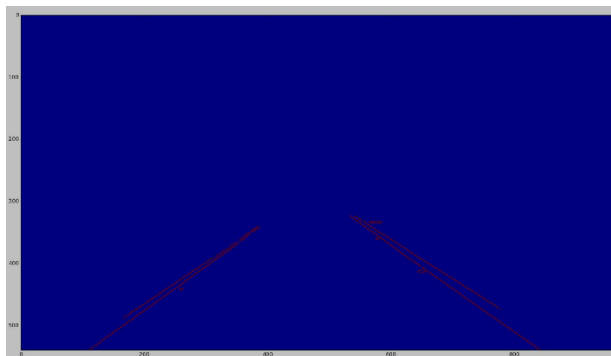


Figura 5. Imagen dónde sólo queda nuestra región de interés.

G. Transformada de Hough

A continuación se aplica la técnica de la Transformada de Hough para extraer líneas y dibujarlas en la imagen. El objetivo de la Transformada de Hough es encontrar líneas identificando todos los puntos que se encuentran en ellas. Esto se hace mediante la conversión del sistema actual denotado por el eje (x, y) a uno paramétrico donde los ejes son (m, b) . En este plano las líneas son representadas como puntos, los puntos se presentan como líneas y las líneas de intersección significa que el mismo punto está en varias líneas.

H. Procesamiento final

Una vez que se realizó la transformada de Hough para encontrar todas las líneas que componen al carril de la carretera, se continúa ahora haciendo un promediado de todas las líneas para poder así manejar solo una línea que defina el carril, esto se determina primero definiendo si las líneas corresponden al carril izquierdo o derecho verificando la pendiente que manejan. Así ahora podemos sumar y promediar las líneas izquierdas con todas las izquierdas y las derechas con todas las derechas. Dando como resultado final la Figura 6.

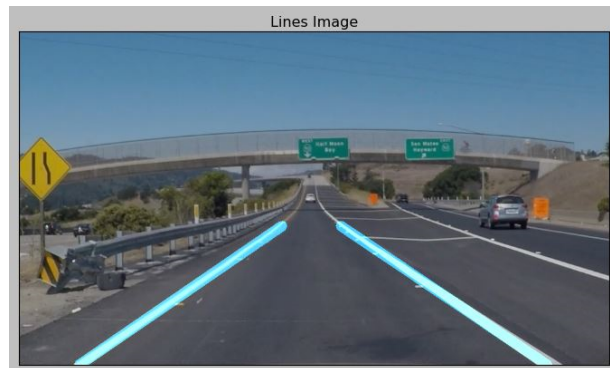


Figura 6. Imagen final una vez aplicado las técnicas de procesamiento.

Conclusiones

El desarrollo de nuevas computadoras con mayores capacidades de procesamiento, ha dado lugar al desarrollo de aplicaciones para el procesamiento de imágenes en sistemas en tiempo real, hoy en día vemos que se encuentra en la cúspide de desarrollo de vehículos autónomos. Este trabajo da lugar a tratar de crear nuevas soluciones a segmentación de carriles durante una curva.

Durante el promediado de la suma de las líneas cuando se trató de procesar un video se presentaron fallas al aparecer una división entre cero; como trabajo a corto plazo se procederá a la mejora del algoritmo.

Referencias

[1] Kaur, G. & Kumar, D. (2015, Febrero). Lane Detection Techniques: A Review. Recuperado el 16 de mayo del 2018, de <https://pdfs.semanticscholar.org/008e/d93ff68a833b8203d7b93383f4822aab5840.pdf>

[2] OpenCV, Recuperado el 18 de mayo del 2018, de <http://docs.opencv.org/2.4/index.html>.

[3] Canny Edge Detection, Recuperado el 18 de mayo del 2018, de http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html