



Universidad
Nacional
de Rosario



TECNICATURA UNIVERSITARIA EN INTELIGENCIA ARTIFICIAL (TUIA)

PROCESAMIENTO DE IMÁGENES (PDI)

Docentes de la cátedra:

Dr., Ing. Gonzalo Sad

Ing. Julian Álvarez

Trabajo Práctico 3 (TP3)

Estudiantes (GRUPO 7):

María Celi

Francisco J. Alomar

Bruno Longo

AÑO 2024 (1er CUATRIMESTRE), Rosario.

INTRODUCCIÓN

En el presente escrito presentamos el procesamiento automático de dos videos. Las consignas están divididas en dos puntos que optamos por resolver conjuntamente en un solo *script*. Los videos están grabados desde un auto que transita por una autovía. La consigna es diseñar un algoritmo que detecte las líneas del carril por el que transita el auto en ambos videos. En el desarrollo del informe, explicamos la implementación del código generado. Al final, exponemos algunas conclusiones.

DETECCIÓN DE AUTOMÁTICA DE LÍNEAS DE CARRIL EN AUTOVÍA.

El código procesa ambos videos de la consigna. La idea general del *script* es que el procesamiento de los videos puede observarse en tiempo real, mientras el video original se reproduce, puede observarse el remarcado de las líneas del carril. Tomamos el primero de los videos para explicar el código, dado que para el segundo se sigue la misma secuencia de pasos:

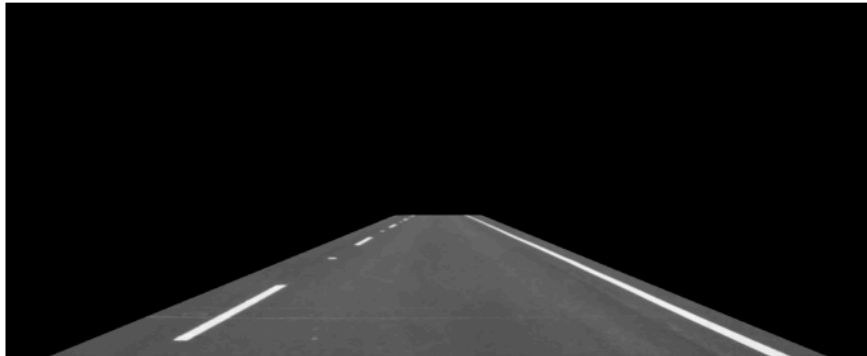
- Se llama a la función `grabar_video_entrada()` que recibe como parámetros la ruta del video a procesar y la ruta que tendrá el video ya procesado. Dentro de la misma, se definen parámetros de tamaño de imagen y se inicia un bucle `While()` que procesa los frames mientras el video se reproduce.

Dentro, se utilizarán fundamentalmente las 2 funciones siguientes a fin de proveer la solución.

- Primero se llama a la función `detectar_lineas_carril()`. Esta función recibe como parámetro cada frame del video. De `frame.shape` obtenemos el alto y ancho del frame para delinear un trapecio como región de interés para el procesamiento. El frame se ve del siguiente modo:



- Allí se capta la imagen del frame normalmente.
- Luego se recorta el trapecio con la región de interés (ROI):

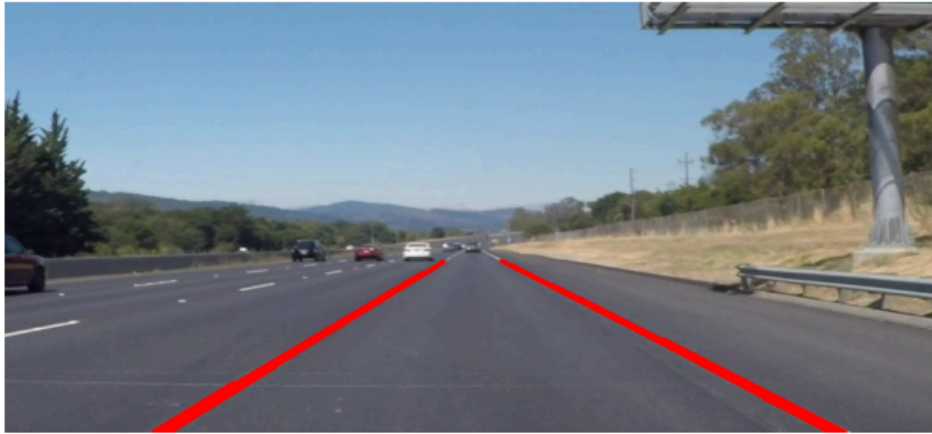


- Después se aplican filtros y finalmente se binariza, a fin de distinguir claramente las rectas propias del carril:



- A la imagen binarizada es el input para que las líneas de carril sean reconocidas a través de la función `HoughLinesP()`:

Frame con Líneas



- La segunda parte, una vez que tenemos las líneas, es realizada por la función DibujarLineas(), la cual recibe "Frame" y "Lines" para devolver "Combined_Image". Estas imágenes se ven como en el ejemplo superior.
- Por último obtenemos los videos y sus rutas de acceso, llamados "Ruta_n_output.mp4", en los cuales puede observarse la continuidad de los frames obtenidos por la combinación de líneas y frames originales:

Frame con Líneas



Al resolver el problema planteado, nos preguntamos por la importancia de detectar las líneas en tiempo real dado que la imagen resulta de un auto en movimiento. En la primera implementación que hicimos el video que mostraba el procesamiento sucedía en “cámara lenta”, debido a que la detección mediante la transformada de Houghs y su presentación en tiempo real consumían recursos de *hardware*, obteniendo este resultado. Solucionamos el inconveniente agregando la función DibujarLineas(), es decir separando la detección de líneas de su “ploteo” en el frame, diseñando así un código más modular. Sin embargo, otras resoluciones hubieran sido factibles, como por ejemplo, aplicar el procesamiento cada un determinado número de frames.

CONCLUSIÓN

Nos dimos cuenta de que las técnicas aprendidas durante la cursada son útiles para resolver diversos problemas. Este aprendizaje nos ha permitido apreciar la potencia de estas herramientas y la variedad de enfoques posibles para abordar un mismo desafío.

En este trabajo, no hemos tenido mayores dificultades y logramos resolver el problema gracias a la experiencia adquirida en entregas previas. Aunque el código funciona bien para los dos videos procesados, sabemos que podría mejorarse para ser más generalizado. Sería interesante evaluar su desempeño en diferentes condiciones, como días nublados, nocturnos, o con distintas señalizaciones de carretera, ajustando los parámetros de la región de interés (ROI) en consecuencia. El enfoque de las mejoras debería estar en el recorte del ROI y en su adaptación a diferentes situaciones para generalizar la detección de líneas de carril.

Nos ha agradado ver cómo con algunos conocimientos hemos podido resolver cuestiones variadas en el área de procesamiento de imágenes. También destacamos el valor de entender el funcionamiento de cada herramienta para aplicarla de manera adecuada según la situación.

En conclusión, con este último trabajo práctico y la cursada en su totalidad, nos llevamos una gran caja de herramientas y una amplia visión para la resolución, entendimiento y comprensión del procesamiento de imágenes.