

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**DISEÑO DE UN ALGORITMO PARA EL ANÁLISIS DE CARRETERAS
BASADO EN PROCESAMIENTO DE IMÁGENES**

Tesis para optar el Título de Ingeniero Electrónico

Presentado por:

JOSÉ LUIS VELÁSQUEZ RODRÍGUEZ

Lima - Perú

2009

RESUMEN

El presente trabajo nace de la idea de intentar solucionar la problemática de los accidentes de tránsito en carreteras, especialmente de los buses interprovinciales.

La idea es situar una cámara de video en la parte superior del vehículo y mediante el algoritmo desarrollado, determinar si el bus está en su carril o si se está desviando. Se plantea el reconocimiento y análisis de las líneas de las carreteras mediante el procesamiento de imágenes.

Con el objetivo de implementar un sistema de análisis completo y aprovechando las capacidades del procesamiento de imágenes, se ha planteado reconocer las líneas y distinguirlas de acuerdo a sus características. Considerando las líneas de la pista en zona de carretera, se tienen cuatro tipos:

- Línea blanca continua: indica los límites de la pista.
- Línea amarilla continua: indica pista de doble sentido, lo cual significa que no se puede cambiar de carril.
- Línea blanca segmentada o discontinua: significa división de carriles y tránsito en la misma dirección.
- Línea amarilla segmentada o discontinua: significa que la pista es de doble sentido, sin embargo está permitido el cambio de carril.

Para el presente trabajo de investigación, se consideran tres tipos de líneas: Líneas continuas (blancas o amarillas), líneas discontinuas blancas y líneas discontinuas amarillas. Entonces al reconocer alguna línea, se analizará para saber qué tipo es y finalmente el sistema marcará las líneas para indicar los bordes de los carriles sobre la carretera.

El desarrollo del algoritmo se dividió en tres partes: segmentación de todos los objetos que contrastan con el color del pavimento, identificación de las líneas y clasificación de las líneas encontradas.

En la segmentación de los objetos se utilizó primero Umbralización automática. Este proceso reconocía la tonalidad de gris que marcaba la diferencia entre la tonalidad del pavimento y la de las líneas. Sin embargo, usando este método no se obtuvo los resultados esperados debido a la presencia de líneas con distintas tonalidades. Entonces se usó un filtro detector de contrastes de tal manera que con esto sólo se tiene una idea dónde pueden estar las líneas. Luego se usa esto como máscara y se hace una umbralización automática en cada región independiente hallada en el filtro.

La identificación de las líneas fue un proceso más empírico. Ya que primero se obtuvieron los datos de los objetos: área, longitud, ancho, ángulo y centro de gravedad (centroide). Luego se estableció un rango de valores – hallado experimentalmente- de cada parámetro que mediante comparaciones, se concluía si es que se trataba de una línea. En esta parte se tuvo que hacer muchos ajustes luego de las pruebas. Puesto que en un principio se trabajó con líneas casi ideales y el rango fue muy reducido.

En la clasificación de las líneas, también se usó un método similar al anterior: basado en comparaciones. Aunque en esta parte fue necesario tener mayores consideraciones para el desarrollo del algoritmo. Se tuvo que considerar el tema de la perspectiva para saber obtener mediciones más exactas con relación a la realidad.

Finalmente un sistema de ayuda o asistencia al conductor para buses interprovinciales, tendría como base este algoritmo de análisis desarrollado. Sin embargo faltaría establecer el sistema de control que recibiría las señales del algoritmo de reconocimiento y emitiría las salidas hacia los actuadores ya sea de manera pasiva (algún tipo de sonido o luz) o activa (control del volante o de los frenos). De tal manera que el bus pueda ser más seguro ante errores del chofer.

A MIS PADRES

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1: ACCIDENTES EN CARRETERAS POR INVASIÓN DE CARRIL	2
1.1. Factores influyentes en los accidentes en la carretera	2
1.1.1. Factores generales	2
1.1.2. Factores específicos	2
1.1.3. Factores organizacionales	3
1.2. Actuales desarrollos de soluciones a los accidentes de tránsito en la carretera	3
1.2.1. Descripción	3
1.2.2. Limitaciones	4
CAPÍTULO 2: RECONOCIMIENTO DE LA CARRETERA COMO PARTE DE UN SISTEMA DE ASISTENCIA AL CONDUCTOR	5
2.1. Estado del arte	5
2.1.1. Presentación del asunto de estudio	5
2.1.2. Estado de la Investigación	6
2.1.3. Síntesis sobre el Asunto de Estudio	8
2.2. Conceptos necesarios para entender los sistemas de asistencia al conductor	9
2.2.1. Conducción	9
2.2.2. Sistemas de reconocimiento	10
2.2.3. Líneas laterales de la carretera	10
2.2.4. Sistema de asistencia al conductor	11
2.3. Elementos fundamentales para un sistema de reconocimiento	12
CAPÍTULO 3: ANÁLISIS, DESARROLLO Y RESULTADOS DEL SISTEMA DE RECONOCIMIENTO DE LÍNEAS DE LA CARRETERA	14
3.1. Ubicación y segmentación de las líneas	15
3.1.1. Selección de las imágenes	15
3.1.2. Umbralización automática	18
3.1.3. Segmentación de las líneas	32
3.2. Discriminación de objetos y clasificación de las líneas	36
3.2.1. Análisis del ancho de cada línea	37

3.2.2.	Análisis del ángulo de inclinación de cada línea con respecto a su ubicación	40
3.2.3.	Clasificación de las líneas encontradas	41
3.3.	Agrupación de las líneas discontinuas y reconstrucción de la imagen	43
3.3.1.	Análisis de las líneas discontinuas	43
3.3.2.	Reconstrucción de la imagen	45
CAPÍTULO 4: CORRECCIONES Y EVALUACIÓN FINAL DEL ALGORITMO		48
4.1.	Vista preliminar del algoritmo desarrollado	48
4.2.	Correcciones	49
4.2.1.	Acondicionamiento de las imágenes	49
4.2.2.	Filtro Pasa-altos y Umbralización automática	50
4.2.3.	Modificando el algoritmo	54
4.3.	Evaluación final del algoritmo	64
4.3.1.	Grupo A	66
4.3.2.	Grupo B	71
4.3.3.	Grupo C	75
4.3.4.	Resumen final	79
CONCLUSIONES		80
RECOMENDACIONES		82
FUENTES		83

INTRODUCCIÓN

Los accidentes de tránsito son uno de los mayores problemas a nivel mundial y de especial preocupación en el Perú. El problema radica en un incorrecto manejo de las unidades de transporte terrestre por parte de los conductores, además de la excesiva jornada laboral de los transportistas lo cual genera que el control sobre el vehículo se pierda debido a la fatiga y el sueño. Lamentablemente, los operativos de control de las unidades para asegurarse que se tenga todos los documentos necesarios para poder circular y que el conductor esté en un estado adecuado para conducir, no son suficientes ya que las cifras de accidentes no disminuyen.

Entonces en el caso de un accidente, las vidas de las personas sólo dependen de la efectividad de los cinturones de seguridad y de las bolsas de aire. En el Perú, esto último no es muy reconocido ya que son muy pocos los vehículos que cuentan con este sistema debido a su alto precio. Por otro lado los primeros sí son prácticos para reducir los daños en las personas después de un accidente gracias a cambios en la legislación, la cual obliga el uso tanto en vehículos particulares como en buses interprovinciales. Sin embargo, las altas velocidades en carretera y el gran tamaño de los buses, hacen que estos dispositivos de seguridad sean poco eficientes para reducir un posible resultado fatal ante un accidente.

Es por esto que surge la necesidad de diseñar un sistema capaz de prevenir un accidente y darle una asistencia al conductor de manera que este podría evitarlo. El objetivo principal de la presente tesis radica en el diseño de un sistema de análisis de carreteras como parte de un sistema de asistencia al conductor mediante el reconocimiento y análisis de las líneas de las carreteras. La idea es desarrollar el algoritmo base de un sistema cuyo propósito es orientar en tiempo real al conductor en la carretera en el caso se haya salido de su carril o esté perdiendo el control del vehículo.

CAPÍTULO 1

ACCIDENTES EN CARRETERAS POR INVASIÓN DE CARRIL

1.1. Factores influyentes en los accidentes en la carretera.

1.1.1. Factores generales

En nuestro país, los accidentes de tránsito son un serio problema que cada vez trae más víctimas. Ante este problema las autoridades han ideado diversos mecanismos de control para garantizar un viaje seguro pero lamentablemente no han sido significativos. Y si a eso se le suma la dificultad de manejar en carreteras de la Sierra peruana, por ejemplo, tenemos lo que constantemente se ve en las noticias: accidentes y caídas de buses interprovinciales en los abismos de la Sierra. Actualmente en nuestro país, no existen ningún dispositivo electrónico que apoye al conductor durante el trayecto de manejo, de manera que pueda detectar curvas próximas o cambios involuntarios de carril y prevenir un posible accidente.

1.1.2. Factores específicos

Actualmente, muy pocas empresas de transporte interprovincial en el Perú aplican métodos para la prevención de accidentes. Podemos mencionar la empresa Transportes Cruz del Sur, que cuenta con un radar láser para reconocimiento de objetos (obstáculos) en la carretera, velocidad controlada electrónicamente y ubicación por GPS [11]. Si bien estos sistemas ayudan a tener un viaje más seguro, no es suficiente ya que el mayor índice de accidentes es por pérdida de control en curvas o desvíos de carril y lamentablemente estos sistemas son poco efectivos.

Por otro lado, las compañías de automóviles han estado desarrollando diversos sistemas de asistencia al conductor, que actualmente salen al mercado como parte del equipamiento de sus vehículos de lujo. Como parte de la competencia de mercado entre las compañías, siempre está el

hecho de contar con la más alta tecnología para sus vehículos. Es por esto que en la actualidad los diversos sistemas que brindan son muy efectivos y existe la tendencia de cada vez mejorarlos. Sin embargo, como se mencionó antes, estos sistemas no se venden por separado del vehículo y solo están disponibles en los vehículos de lujo lo cual implica que su precio es alto, ergo hace difícil el acceso para la mayoría de las personas a esta importante tecnología.

Es por esto, que surge la necesidad de investigar en desarrollo de este tipo de sistemas de seguridad que son tan necesarios en el Perú para disminuir los accidentes.

1.1.3. Factores organizacionales

Como se mencionó antes, la Pontificia Universidad Católica del Perú se encuentra capacitada para el desarrollo de sistemas de asistencia al conductor, ya que posee dispositivos y materiales adecuados y cuenta con los docentes con conocimientos necesarios para el mismo. De acuerdo a esto, la infraestructura necesaria dentro de la universidad correspondería a los laboratorios de electrónica de la universidad. Así, las primeras pruebas y la captura de imágenes se harían en las pistas dentro de la universidad, mientras que las simulaciones del sistema y la implementación del mismo se realizarían en laboratorios de electrónica. Asimismo es necesario poseer conocimientos de procesamiento digital de imágenes y video así como también del uso de un software adecuado de análisis digital de imágenes como Matlab.

1.2. Actuales desarrollos de soluciones a los accidentes de tránsito en carretera.

1.2.1. Descripción

Como se mencionó anteriormente, desarrollos para disminuir los accidentes de tránsito en la carretera, en el Perú son prácticamente nulos y en el extranjero, dependen básicamente de los desarrollos tecnológicos

por parte de la industria automotriz. Sin embargo, lo que sí se utiliza son diversos métodos que eventualmente podrían ayudar a reducir los accidentes, como por ejemplo el uso de un radar láser para detectar objetos en la carretera; controlar la velocidad electrónicamente programada a 80 Km./h, según reglamento de tránsito.

1.2.2. Limitaciones

Lamentablemente los sistemas descritos anteriormente si bien son una ayuda a tener mayor control en el manejo, no son muy relevantes para disminuir notablemente la tasa de accidentes. El problema radica en que estos sistemas son de uso general, es decir su funcionamiento es constante y no llegan a asistir al piloto de manera específica.

Por otra parte, en el caso de la industria automotriz, la mayoría de los autos de lujo cuenta con un sistema de reconocimiento de la carretera ya sea por láser o por análisis de video, el cual puede avisar al piloto si involuntariamente éste se está saliendo de su carril o si es que se está aproximando a una curva. Con esto el piloto hará las maniobras que crea conveniente para evitar un accidente. Sin embargo, este sistema no está disponible como producto independiente, sólo se vende como parte del equipamiento en los autos de lujo. Entonces al ser inaccesible comercialmente para el resto del parque automotor y en especial de los buses, este sistema no representa una solución.

CAPÍTULO 2

RECONOCIMIENTO DE LA CARRETERA COMO PARTE DE UN SISTEMA DE ASISTENCIA AL CONDUCTOR.

2.1. Estado del arte.

2.1.1. Presentación del asunto de estudio.

Como ya se ha mencionado la tasa de accidentes de tránsito es muy alta [12], la mayoría de estos accidentes son debido a un problema en la conducción del vehículo, es decir debido a fallas humanas. Las razones pueden ser diversas, exceso de velocidad, fatiga, sueño, distracción, falta de visibilidad, etc. Como un aditamento que evite que estas fallas humanas terminen en tragedia, lo que se propone en este trabajo es un sistema de seguridad frente a posibles accidentes en la carretera tanto para automóviles como para buses.

Hoy en día los sistemas de seguridad que se utilizan son fruto del desarrollo de las décadas de los setentas y ochentas los cuales comprenden básicamente de cinturones de seguridad y bolsas de aire. Estos sistemas son muy eficaces y de gran utilidad, sin embargo su efecto sólo se produce una vez ocurrido el accidente. Por lo que desde la década pasada se está haciendo investigación y desarrollo en Sistemas de Asistencia al Conductor (DAS, por sus siglas en inglés) que tienen por objetivo prevenir y minimizar los accidentes de tránsito por medio de avisos de peligros cercanos al conductor o sistemas mecánicos activos que automáticamente ayudan a tener un mejor control del vehículo. Para lo cual cuentan con avanzadas tecnologías electrónicas como distintos modelos de sensores -tales como láser y ultrasonido-, cámaras de video y procesadores digitales de alto desempeño. Esto último se utiliza para agilizar el proceso e intentar que éste se ejecute en tiempo real.

Cabe resaltar que el sistema de alerta de cambio involuntario de carril (LDWS, *Lane Departure Warning System* [8] o LKAS, *Lane Keeping*

Assistance System [7]) se ha considerado el más interesante debido a sus potenciales usos [5], tanto en seguridad –como antes se ha mencionado– como en los proyectos de vehículos autónomos [13]. La siguiente sección describe algunos estudios realizados sobre reconocimiento del carril de la carretera utilizando diversos métodos.

2.1.2. Estado de la Investigación.

El proceso de reconocimiento de las líneas laterales de la carretera se puede desarrollar de distintas maneras, sin embargo la opción elegida para el presente trabajo es por medio de procesamiento digital de imágenes, debido a que la imagen presenta una digitalización visual muy parecida a lo que percibe el ojo humano. Empero el proceso de reconocimiento tiene sus dificultades, como son: la necesidad de hacerlo en tiempo real, lidiar con diversos factores reales en la imagen (sombras, neblina, oscuridad, líneas con pintura arruinada), y los problemas de costo pensando en un posible uso comercial.

Uno de los trabajos ya presentados sobre este tema, ubica de manera automática las líneas pintadas en forma continua a partir de una fotografía usando el software MatLab en un computador [1]. La técnica es simple y se basa en un enmascaramiento de la imagen a procesar y el empleo de la Transformada de Houg. El proceso es eficiente para pistas libres con líneas continuas y con imperfecciones como rompe muelles o discontinuidad de la línea por garajes presentes. Sin embargo, no funciona de la misma manera para un análisis con otros contextos como veredas o jardines en los bordes de la pista. Por otra parte, el uso de este sistema a nivel comercial no es posible debido al excesivo tiempo de duración del proceso de reconocimiento.

Otro método de reconocimiento de la carretera es mediante el uso de sistemas de medición por láser o LMS (Laser Measurement Systems) [2]. Consiste en analizar y medir el nivel de la superficie del asfalto y poder determinar la distancia entre los bordes asumiendo que la pista va a

estar a un nivel distinto que al resto de vías. Seguidamente, se unen los puntos leídos y se hace un mapeo del posible camino que está a punto de recorrerse. Esto está principalmente diseñado para funcionar en zona urbana donde las pistas por lo general están bordeadas por veredas. Como es de esperarse las limitaciones que presenta son numerosas como por ejemplo: posibles baches, pistas que no estén bien pavimentadas o presencia de materiales o componentes que perjudiquen el análisis (charcos de agua), provocan malas mediciones y hacen que este método no sea la principal fuente de los sistemas de reconocimiento de carretera, sino un apoyo a los procesos con imágenes en situaciones en las cuales éstas no puedan ser procesadas correctamente.

Dentro de los desarrollos de estos sistemas de reconocimiento, la Universidad Nacional de Australia ha ido más allá de una simple detección de la pista [3]. Consiste en un sistema activo de asistencia al conductor basado en visión computarizada que reconoce el carril de la pista y además actúa sobre el vehículo haciéndolo que se mantenga en el carril mediante un torque de apoyo aplicado a la dirección (timón) del móvil. En lo que respecta al reconocimiento de la pista, el algoritmo de detección se basa en las líneas laterales del carril y el color del asfalto en contraste con el ambiente exterior, lo cual nos da no solo una posible estructura de la pista a seguir, sino también una posición relativa del vehículo con respecto al carril y a la pista.

Otra investigación que vale la pena mencionar es sobre el uso de un algoritmo que compara las imágenes obtenidas con modelos de pista preestablecidos a fin de reconocer si es que la imagen se trata de la carretera. También utiliza las líneas de los bordes como guía [4]. Con esto se reduce el tiempo de proceso de reconocimiento automático ya que solo consiste en comparaciones sin necesidad de entrar en detalles que, realizándolo sobre una plataforma de hardware especializado, puede lidiar con el difícil problema del tiempo-real. En resumen, lo que se utiliza en esta investigación es un algoritmo de reconocimiento simple basado

en comparaciones, una base de datos previamente establecida y una arquitectura de bajo costo basada en un ordenador.

2.1.3. Síntesis sobre el Asunto de Estudio.

Lo que anteriormente se ha presentado son diversos métodos que funcionan bien dentro de sus restricciones. Sin embargo, lo que se busca es una técnica general y que funcione bien en todo momento y para todo tipo de ambiente dentro de lo que sería la carretera, es decir vía libre, sin cruces, buena definición de la pintura de los bordes, etc. Lo que se presenta a continuación es una profunda recapitulación de las ventajas y desventajas que presentan los sistemas anteriormente descritos.

El primer método descrito [1] es simple en programación y el algoritmo utilizado para el reconocimiento tampoco es complicado, sin embargo como se mencionó antes, el sistema presenta deficiencias en el análisis a contextos más reales, como son pistas con jardines y hierbas en los bordes, ya que le añade ruido a la imagen lo que dificulta su procesamiento. Además, es importante mencionar el problema del tiempo de procesamiento, el cual es excesivo. Esto es muy importante para el diseño de todo sistema de este tipo: el tema del tiempo-real debe ser tan elemental como el mismo reconocimiento de las pistas, ya que si el sistema reconoce a la perfección la carretera sin importar el contexto pero el tiempo de procesamiento es considerablemente alto, el sistema pasaría a ser un método didáctico o una base para futuros trabajos.

Lo que nos propone el segundo estudio descrito [2], es un sistema alternativo para el reconocimiento. Es una técnica buena como soporte de los sistemas basados en imágenes. Como ventajas cabe resaltar su simplicidad, bajo costo y eficiencia en la duración del procesamiento, por lo que sí se puede implementar para que funcione en tiempo-real. Como desventaja, está que sólo es un sistema de apoyo ya que está muy limitado para diversos panoramas.

Otro algoritmo a analizar es el del reconocimiento por comparación. Gracias al uso de un algoritmo basado en comparaciones, se ha conseguido que el tiempo de procesamiento haya disminuido considerablemente logrando un análisis en tiempo real, y eso es una ventaja que cabe resaltar. Sin embargo, el método presenta deficiencias para pistas que no cumplan con los patrones o modelos preestablecidos, ya que en estos casos simplemente no reconocería la pista. Una posible solución podría ser aumentar la cantidad de imágenes almacenadas de manera que esta base de datos tenga todos los posibles modelos que puedan corresponder a la carretera. Aunque, eso desencadenaría mayor datos a comparar, mayor tiempo de procesamiento y lo que es peor, una extensa búsqueda de todos los tipos de panoramas de la carretera.

Como repaso general, se puede concluir que debido a la variedad de formas de la carretera lo mejor es poder procesar las imágenes de entrada y conseguir un eficiente resultado de reconocimiento por parte del computador. Por otra parte, un algoritmo capaz de reconocer cualquier tipo de carretera puede generar un código de programación (software) relativamente complejo, lo cual nos trae como consecuencia un prolongado tiempo de procesamiento. Sin embargo, un uso apropiado de hardware –se dice apropiado ya que un excesivo uso de hardware implica mayor costo- como plataformas de procesamiento paralelas, podría mejorar el desempeño del sistema.

2.2. Conceptos necesarios para entender los sistemas de asistencia al conductor.

2.2.1. Conducción

En este caso, se tratará el término conducción como la habilidad y capacidad de manejar vehículos terrestres. Pueden ser autos, buses o camiones.

2.2.1.1. Conducción en carretera

Se trata de la conducción en la cual el vehículo manejado circula por la carretera. Involucra alta velocidad, mayor concentración para manejar y mayor peligro en caso de accidente.

2.2.2. Sistemas de reconocimiento

Es un sistema automático capaz de buscar de manera activa información sobre algo o alguien que lo relacione con el patrón a identificar para establecer semejanzas. Se pueden emplear por diversos métodos, como son láser, sensores por ultrasonido, imágenes, videos, audio, etc.

2.2.2.1. Sistemas de reconocimiento por imágenes

Este tipo de sistema de reconocimiento se basa en el análisis de imágenes digitales para poder identificar diversas formas de objetos o personas.

2.2.3. Línea lateral de la carretera

También conocida como Línea de borde de pavimento, se utilizará para demarcar el borde del pavimento a fin de facilitar la conducción del vehículo. Deberá ser línea continua de 0.10m. de ancho de color blanco [14].

2.2.3.1. Línea central

Se utiliza para separar carriles de sentido opuesto en una misma calzada. En el caso de ser carretera podría ser línea discontinua (de 0.10m. – 0.15m. de ancho y con segmentos de longitud 4.50m. espaciadas 7.50m. entre cada una) o doble línea continua, dependiendo del tamaño de la pista. En ambos casos la línea debe ser de color amarillo. No está de más resaltar que en el caso de las líneas discontinuas, está permitido el cambio de carril siempre y cuando sea seguro. Mientras que la doble línea continua indica que no se puede pasar a través de esta, es decir no se puede cambiar de carril por ningún motivo [14].

2.2.3.2. Línea de carril

Son utilizadas para separar los carriles de circulación que transitan en la misma dirección. Estas líneas son discontinuas de color blanco. En el caso de carreteras son de 0.10m. – 0.15m. de ancho y con segmentos de longitud 4.50m. espaciadas 7.50m. entre cada una. Significa que se puede pasar a través de esta. Está permitido el cambio de carril siempre y cuando sea seguro. Se utiliza para separar los carriles rápidos y lentos en una pista del mismo sentido [14].

2.2.4. Sistema de asistencia al conductor

Medio electrónico capaz de predecir un posible accidente mediante diversos procedimientos. Se dice que asiste al conductor ya que brinda permanentemente una ayuda (como un aviso sonoro) a éste para mejorar el control del vehículo y evitar una posible colisión o accidente. Normalmente, estos sistemas funcionan en base a sensores por ultrasonidos, leds infrarrojos, rayos láser o procesamiento de imágenes.

2.2.4.1. Sensores por ultrasonidos

Son sensores que se basan en el tiempo que se demora una señal de ultrasonido en ser recibida. De esta manera determinan la presencia de algún objeto sólido y en algunos casos la distancia a la que se encuentra.

2.2.4.2. Leds infrarrojos

Son diodos que emiten una luz infrarroja imperceptible para la vista humana, de manera que los objetos iluminados puedan ser vistos por una cámara. Su principal uso es en lugares con escasa iluminación.

2.2.4.3. Procesamiento digital de señales (DSP)

Es un área de la ingeniería que se dedica al análisis y procesamiento de señales (audio, voz, imágenes, video) que son discretas. Una vez obtenida la entrada digital, mediante un código de programación (algoritmo) se analiza y se logra establecer una conclusión, que finalmente será la salida del sistema.

2.2.4.3.1. Matlab

Es un programa informático de simulación capaz de resolver algoritmos matemáticos. En este trabajo, lo usaremos para poder analizar nuestra imagen de entrada.

2.2.4.3.2. Transformada de Hough

Es una técnica matemática utilizada en procesamiento de imágenes, que permite encontrar líneas y curvas. El método consiste en transformar las coordenadas de los puntos hallados luego de la detección de bordes, de cartesianas a polares. Con lo cual cada punto evaluado de la imagen, se convierte en una función en el plano $\rho\theta$. Una vez obtenidas las funciones, se evalúan en sus extremos y se concluye si los puntos son o no parte de una recta. [15].

2.3. Elementos fundamentales para un sistema de reconocimiento.

Para poder definir el modelo teórico necesitamos evaluar la aplicación que va a tener y el tipo consumidor que va a hacer uso del presente desarrollo. Luego se podrán obtener las características del modelo para su implementación así como también las limitaciones de su funcionamiento.

En todo sistema de asistencia al conductor, lo que se busca es que tenga la mayor eficiencia posible con un margen de error mínimo dentro de las posibilidades. En estos sistemas es crucial el tema del error por sobre todas las cosas, debido a que una falla en la lectura de entrada o una falla en la cámara instalada podría ocasionar que el conductor pierda el control del auto lo cual, agregado a las altas velocidades de movimiento del vehículo, podría generar un grave accidente. Es por eso que este tema debe ser tratado cuidadosamente.

Las pistas presentan diversos ambientes tanto en ciudad como fuera de ella, lo que nos da una gran gama de imágenes relacionadas a procesar.

Sin embargo, la ciudad nos trae muchos casos que complican el desarrollo del sistema de estudio y lo hacen simplemente ineficaz. Como son los cruces, los semáforos, las imperfecciones en las pistas (como baches y rompe muelles), las distintas rutas que existen para conectar lugares –lo que genera realizar muchas viradas-. Entonces como parte de la eficiencia y performance, el desarrollo del DAS de estudio debe ser únicamente utilizado para carreteras. Debido a que es un camino directo, sin semáforos, sin cruces y relativamente más tranquilo para manejar.

Lo que se propone no sólo viene por el tema de seguridad, sino también como un sistema de confort a largo plazo, lo que sería el manejo automático. Sin embargo, el tema del vehículo autónoma es un asunto que no se va a tratar en este trabajo, simplemente es un posible desarrollo a futuro tomando de base este sistema DAS.

Recapitulando, el sistema de asistencia al conductor basado en el reconocimiento de las líneas de la carretera debe tener una eficiencia tal que errores que podrían ocasionar terribles consecuencias sean prácticamente nulos. Es decir si bien la perfección no está garantizada, lo que se quiere es que los errores no sean de base sino de fondo y no lleguen a perjudicar de manera notable al usuario. Aparte, también está el asunto del tiempo de procesamiento que tantas veces se ha mencionado en este trabajo, el cual es muy importante si se quiere que el DAS se desarrolle de manera comercial.

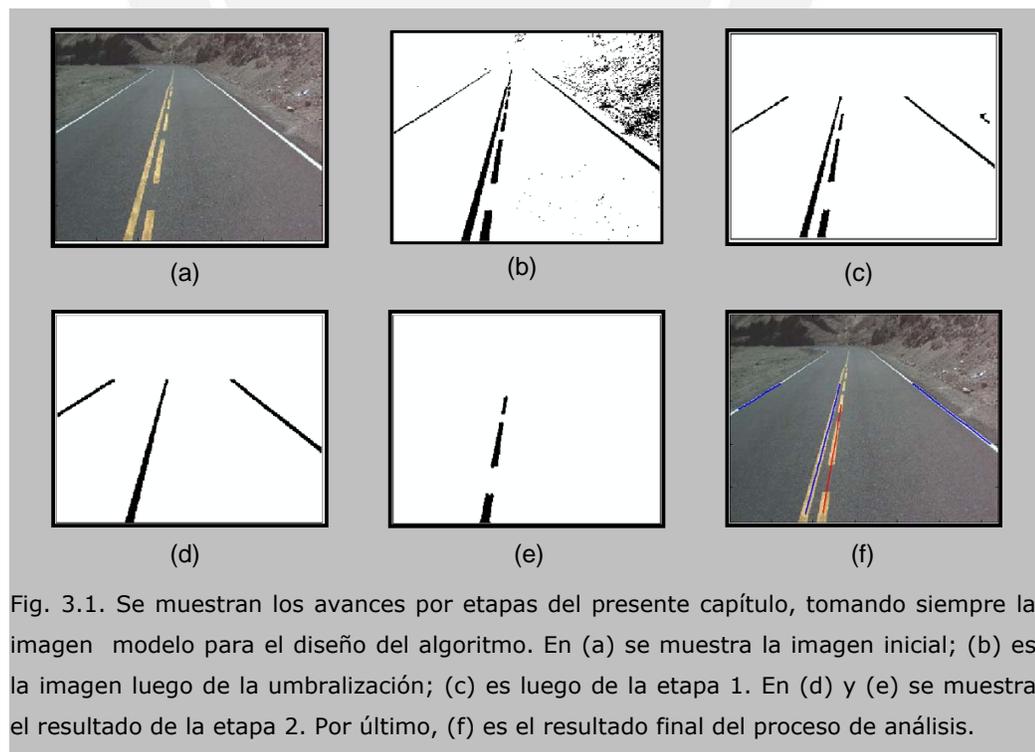
Finalmente, en el Perú un sistema de este tipo podría funcionar convenientemente dentro de lo esperado, ya que se cuenta con algunas carreteras señalizadas y el asfalto bien tratado.

CAPÍTULO 3

**ANÁLISIS, DESARROLLO Y RESULTADOS DEL SISTEMA DE
RECONOCIMIENTO DE LÍNEAS DE LA CARRETERA**

En esta sección se describe los procesamientos utilizados para identificar las líneas encontradas en la imagen de la carretera. La primera parte corresponde a la ubicación y segmentación de las líneas, la segunda parte a la discriminación (tomando sólo los objetos que sean líneas y despreciando el resto) y clasificación de las líneas; por último una tercera parte reordena las líneas discontinuas y reconstruye la imagen procesada.

Es necesario resaltar que se debe empezar analizando una imagen real que simbolice el caso ideal de la carretera para facilitar el diseño del algoritmo. Luego, tomando de base lo constituido y analizando diversas imágenes, se procedería a corregir y mejorar los criterios de análisis (CAP. 4).



Se trabajó con la imagen 'Imagen 1' (Fig. 3.1.a) para el diseño del algoritmo puesto que las líneas de la pista se ven muy bien pintadas y presenta alto contraste con el asfalto, lo cual ayuda al análisis. También se añadió una segunda imagen 'Imagen 2' (Fig. 3.2) para revalidar el proceso de Umbralización Automática.

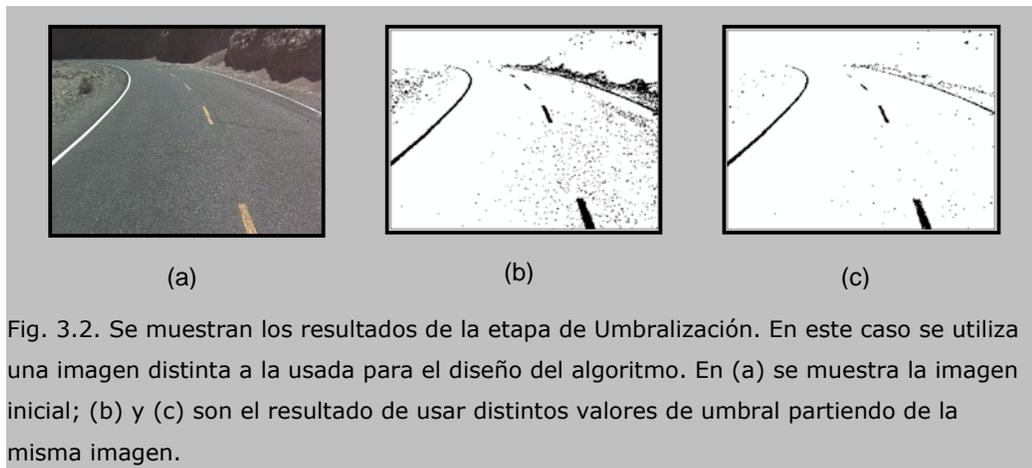


Fig. 3.2. Se muestran los resultados de la etapa de Umbralización. En este caso se utiliza una imagen distinta a la usada para el diseño del algoritmo. En (a) se muestra la imagen inicial; (b) y (c) son el resultado de usar distintos valores de umbral partiendo de la misma imagen.

3.1. Ubicación y segmentación de las líneas

El proceso de ubicación y segmentación de las líneas se considera la primera etapa de todo el desarrollo debido a que es la base para poder separar las líneas una por una de la imagen. Y posteriormente almacenarlas como una tabla de datos para proceder con el análisis de cada línea.

3.1.1. Selección de las imágenes

Se ha elegido imágenes de 320x240 debido a que en procesamiento de imágenes siempre es importante trabajar de preferencia con una cantidad de datos baja, tratando de conocer la lógica del programa y los procesos a seguir. Posteriormente, una vez hecho el algoritmo y conocido el proceso, se podría aumentar la resolución de las imágenes. De esta manera la eficiencia aumentaría sin dejar de lado la practicidad del sistema. Es necesario mencionar que a mayor cantidad de datos

(imágenes más grandes), el costo computacional aumenta lo cual implica mayor tiempo de procesamiento.

En este caso la cámara digital capturó las imágenes con una resolución de 1280x960 y se redujo hasta 320x240 utilizando el método de interpolación bicúbica. Adicionalmente se utilizan las mismas imágenes originales para obtener imágenes de 640x480, reducidas con el mismo algoritmo. Estas nuevas imágenes se utilizan en el Capítulo 4 como parte de pruebas adicionales.

Las imágenes que se muestran en este y el siguiente capítulo forman parte del grupo de imágenes "Grupo A". Más adelante se utilizarán diversas imágenes (de otros grupos) para las correcciones y para mostrar los resultados finales.

Adicionalmente las imágenes que se utilizarán en este capítulo han sido capturadas con una cámara a color con una lente con 12mm de distancia focal, situada en la parte superior de un auto con una inclinación de -20° . Posteriormente el video fue separado en imágenes.



Fig. 3.3. Imagen utilizada para el desarrollo del algoritmo.

Para la selección del plano con el que se va a trabajar, se toman en consideración dos características importantes de estas imágenes:

- Las líneas son blancas o amarillas, no hay otra opción.
- El color del fondo es normalmente oscuro pero se va aclarando con el tiempo debido al polvo, hasta un gris oscuro.

Entonces tomando la primera característica se podría concluir *a priori* que la mejor opción para trabajar sería con el plano rojo, ya que el amarillo representa un mejor contraste en este plano. Por lo tanto la línea estará mejor definida, facilitando su reconocimiento posterior. A continuación se muestra el análisis gráfico:

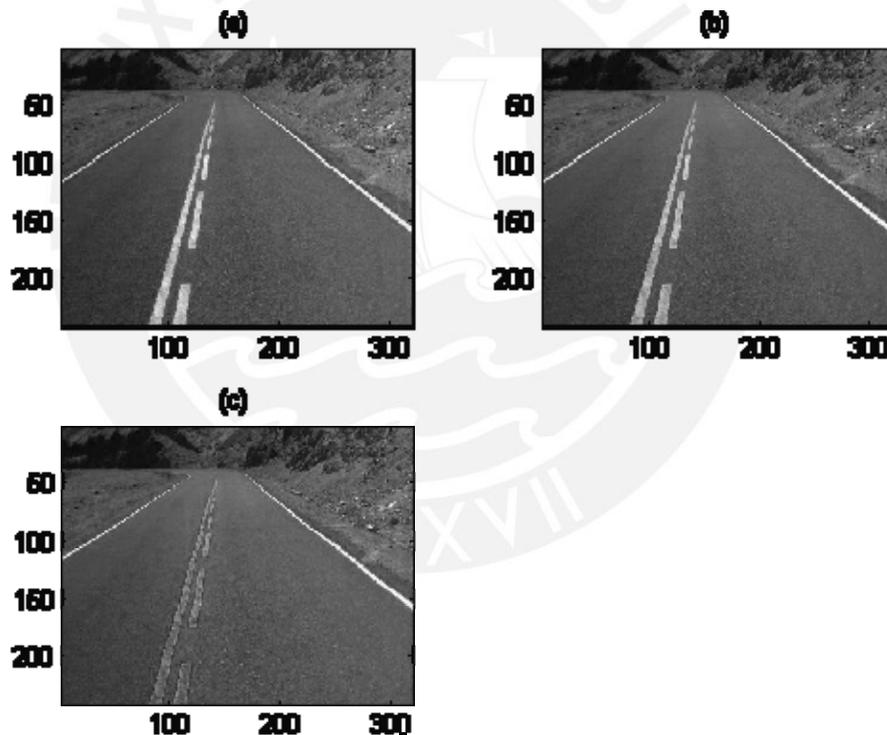


Fig. 3.4. Separación de la imagen en los planos Rojo, Verde y Azul. (a) Corresponde al plano Rojo; (b) corresponde al plano Verde; (c) corresponde al plano Azul.

Tal como se mencionó, el plano rojo presenta más contraste con las líneas, por lo tanto se trabajará con este plano para poder distinguir y separar más fácilmente las líneas.

3.1.2. Umbralización automática

Esta es la parte más importante del desarrollo, ya que de ésta depende el reconocimiento de las líneas. Si en la umbralización hay alguna línea que no ha sido correctamente analizada y no se pudo detectar, no hay forma de recuperarla en el futuro. Es muy importante establecer un correcto valor de umbral.

Como primera solución se tomó un valor umbral fijo, mediante la comparación de histogramas de distintas imágenes. Sin embargo rápidamente se puede concluir que debido a la gran variedad, el método es poco confiable y trae una cantidad significativa de errores.

Por lo tanto se utilizará la umbralización automática como solución a la segmentación de las imágenes. La implementación del mejor método involucró la revisión de distintas técnicas conocidas para la umbralización. Se eligieron las cuatro más aplicables como herramientas para la binarización de las imágenes mediante la umbralización. Cada método se analizó con 15 imágenes seleccionadas al azar. Cada imagen tiene ciertas modificaciones (máscaras), las cuales se explicará posteriormente en otra sección (3.1.3.1. Acondicionamiento de la imagen). A continuación se muestran una breve descripción los métodos y los resultados de cinco imágenes. Los resultados del resto de imágenes se muestran en la sección Anexos.

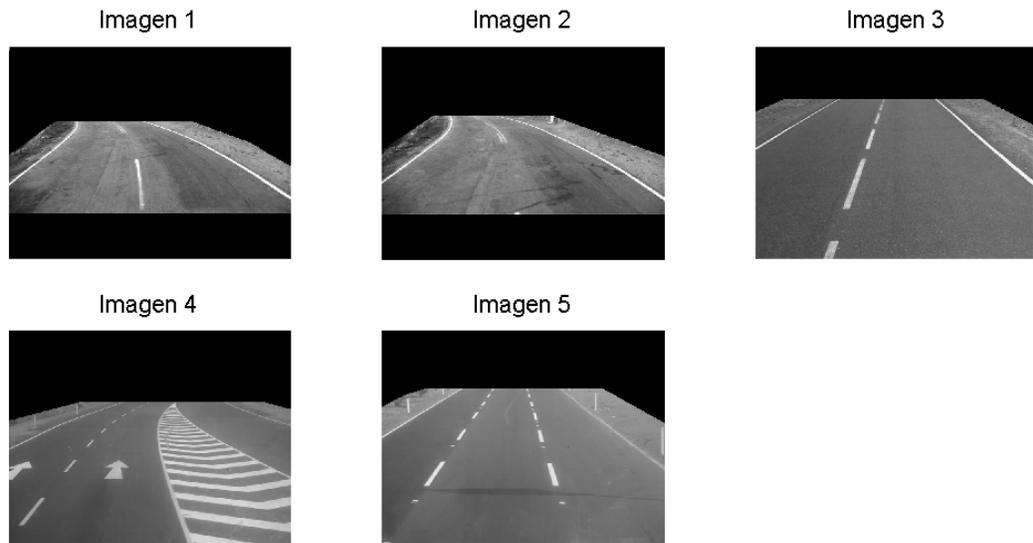


Fig. 3.5. Imágenes originales con una máscara incorporada para disminuir la cantidad de datos innecesarios en el proceso.

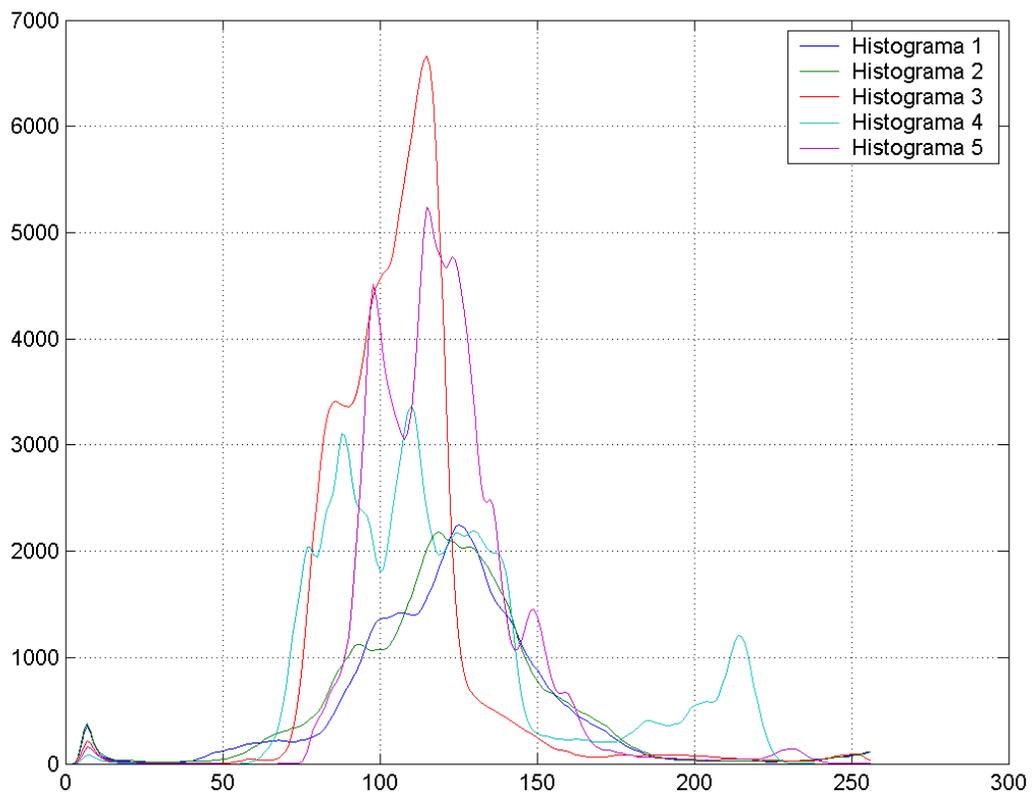


Fig. 3.6. Histogramas de las imágenes presentadas anteriormente. El número de histograma que indica la leyenda, hace referencia al número de imagen.

3.1.2.1. Método de Otsu

El método de Otsu quizás es el método de binarización de imágenes por umbralización más conocido. Consiste en un análisis matemático-estadístico de los valores que se obtienen en el histograma de una imagen en escala de grises. El algoritmo elige el umbral óptimo maximizando la varianza entre clases (modas) mediante una búsqueda exhaustiva [16]. La idea es separar los objetos con mayor grado de notoriedad, del fondo. Por lo que se asume que la imagen está dividida en fondo y objetos presentes. Al establecer un único valor umbral para la separación de los objetos del fondo de la escena, se considera que la imagen es bimodal y la separación de las tonalidades entre los objetos y el fondo es clara.

Para aplicar el método de Otsu en las imágenes del presente trabajo, es necesario analizar la estructura de las imágenes para garantizar *a priori* que el método funcionará de la mejor manera. Las imágenes presentan las siguientes características:

- Región oscura, que sirve para enmascarar partes de la imagen que perjudican el análisis.
- Región del fondo, corresponde al asfalto y ocupa la mayor área de la imagen.
- Región de los objetos, corresponde a las líneas de la pista. Eventualmente podrían adicionarse algunos objetos externos, con la misma tonalidad de gris.

Por consiguiente es necesario anular la parte oscura de la imagen, ya que así como está las áreas de mayor volumen y a partir de las cuales se va a hacer la división con el método de Otsu, serían la región oscura y el resto (asfalto y líneas).

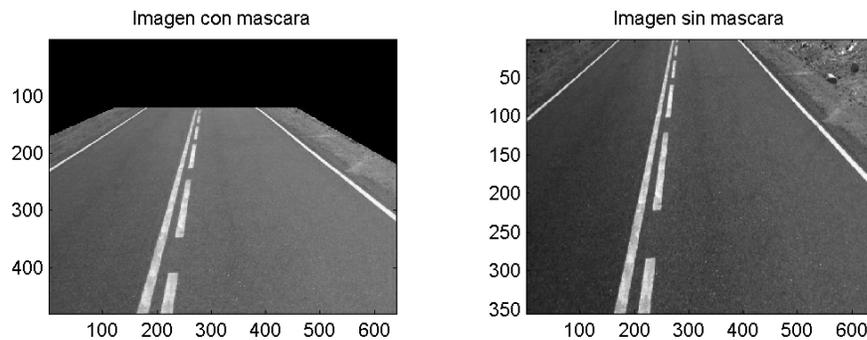


Fig. 3.7. Es necesario tener la imagen únicamente con la pista y con las líneas para obtener una buena umbralización con el método de Otsu.

Una vez anulada la parte oscura, la imagen se reduce hasta el corte de la parte oscura anteriormente marcada. Luego se procede a la binarización de la imagen empleando el método de Otsu, con la ayuda de instrucciones de Matlab previamente establecidas. A continuación se muestran los resultados:

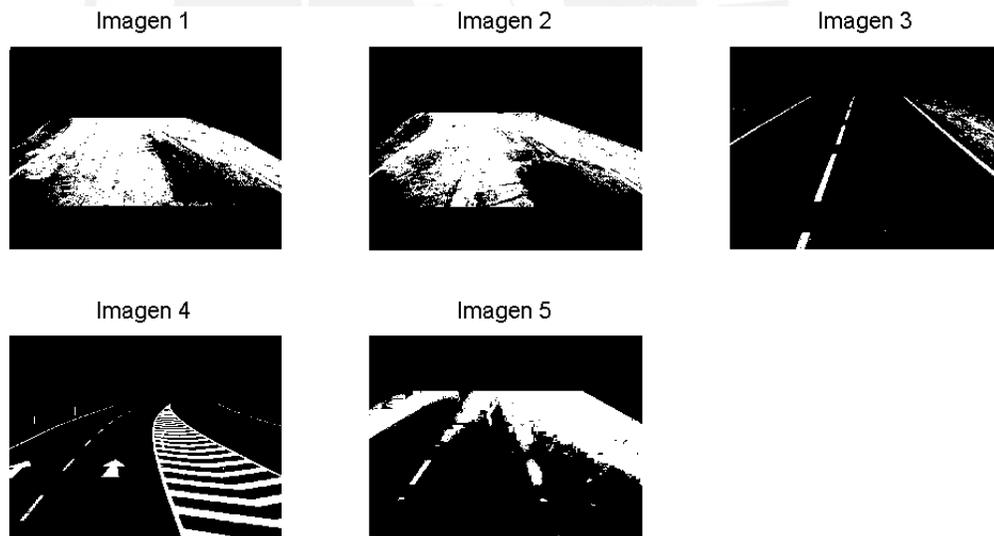


Fig. 3.8. Resultados obtenidos con el método de Otsu. Se puede ver claramente que se incorpora una cantidad de ruido significativa.

Se observó que los resultados no fueron los esperados. Al analizar los histogramas y se llegó a la conclusión que el método no se podía aplicar debido a que las imágenes presentan irregularidades en la distribución de tonalidades de las líneas. Inclusive en algunos casos se puede ver una

distribución bimodal únicamente para la región de las líneas. Lo cual imposibilita el uso del método de Otsu para estas imágenes.

3.1.2.2. Análisis de histogramas en escala logarítmica

Luego de los resultados de umbralización con el método de Otsu, se pudo observar mediante los histogramas que relación entre áreas (región pista y región líneas) es del orden aproximadamente de 1:40. Entonces para observar de mejor manera los cambios, picos y valles del histograma, lo óptimo sería trabajar con una escala logarítmica. Los histogramas muestran de mejor manera la distribución de tonalidades de toda la imagen, tal como se puede ver a continuación:

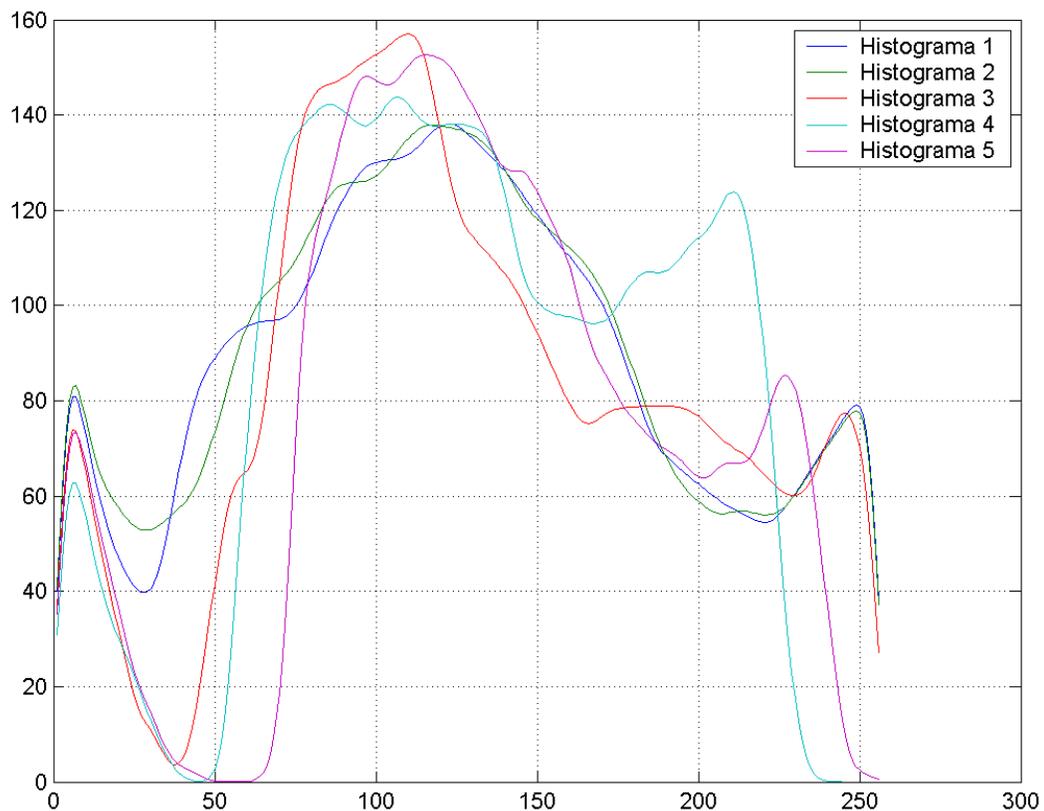


Fig. 3.9. Histogramas en escala logarítmica. Se puede apreciar mejor la sección del histograma correspondiente a las líneas (parte derecha).

Se puede observar que la región del histograma que corresponde a las líneas tiene mayor notoriedad. Entonces al marcarse una mejor diferencia

entre ambas regiones (pista y líneas), va a permitir seleccionar un mejor valor umbral.

Como primera premisa de selección de umbral, se toma el punto más bajo entre ambos picos (pico de la pista, pico de las líneas). Los resultados se muestran a continuación:

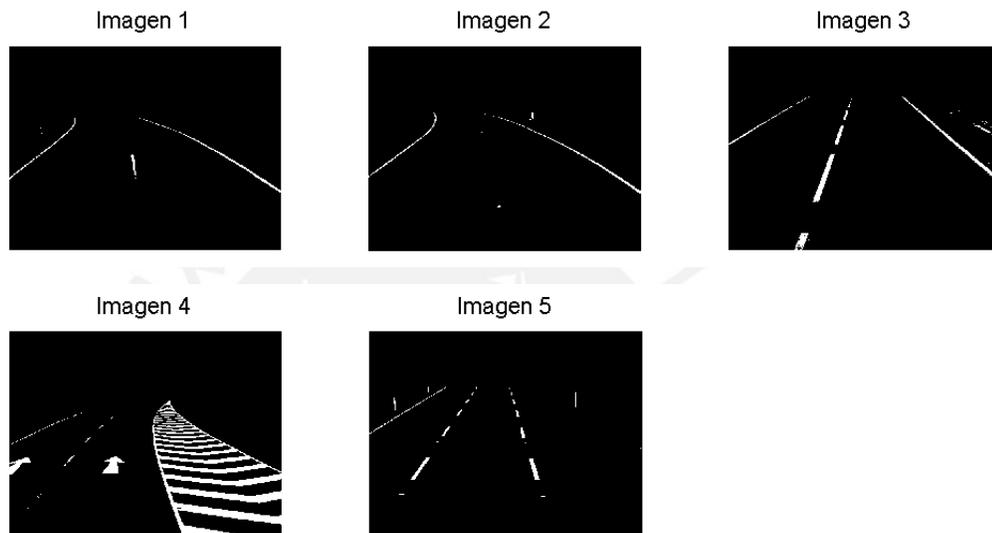


Fig. 3.10. Resultados obtenidos analizando el histograma en escala logarítmica. Manualmente se tomó el valor más bajo entre las campanas correspondientes a la pista y a las líneas.

Al ver que esta premisa no funciona muy bien ya que hay muchas líneas no detectadas, se pasa a tener una segunda consideración. El valor umbral se toma como el punto donde la pendiente de bajada (desde el punto más alto, que normalmente corresponde a la pista) cambia significativamente.

Los resultados se muestran a continuación:

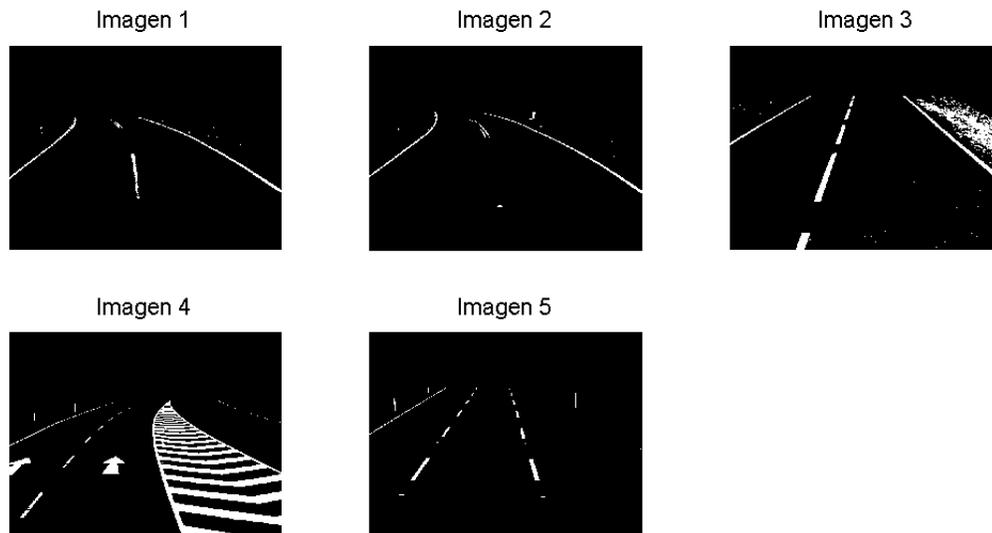


Fig. 3.11. Resultados obtenidos analizando el histograma en escala logarítmica. Manualmente se tomó el valor donde la pendiente de bajada cambia notablemente.

Con estos resultados se puede concluir que seleccionar un valor umbral mediante el análisis logarítmico de los histogramas, tampoco se llega a los resultados óptimos.

3.1.2.3. Detección de bordes

Otra técnica para la segmentación de imágenes es mediante la detección de bordes. En este caso no se utiliza el histograma para seleccionar el valor umbral. El método consiste en detectar el borde a la imagen original, luego tomar valores umbral (posibles) y detectar el borde de esta imagen binarizada. Finalmente ambos bordes se comparan y se toma como umbral, al valor que me da la máxima similitud entre ambos bordes [17].

La implementación del algoritmo es relativamente simple:

- Se utiliza la instrucción "edge" para detectar los bordes de la imagen original y de la imagen binarizada. Es importante que tanto

la imagen original como la imagen binarizada sean procesadas con la misma técnica de detección de bordes.

- Se toma el borde de la imagen original como patrón de comparación.
- Se realiza un barrido con todos los bordes detectados de las imágenes umbralizadas con los valores de gris (del 1 al 256).
- La comparación consiste en la multiplicación de ambos bordes.
- Finalmente se toma como valor umbral aquel valor que produce que la multiplicación tenga la mayor cantidad de píxeles.

Sin embargo fue necesario realizar una modificación a la imagen original. La detección de bordes marca los contrastes que existen en la imagen, por lo que debido a la máscara (región negra) aparece un marco. Para no generar resultados ambiguos, el marco debe eliminarse ya que *a priori* se sabe que este marco corresponde a una región que no es de interés. Entonces se umbraliza la imagen con el valor de 1 (correspondiente a la región negra) y se erosiona con una matriz pequeña de tal manera que una vez que se multiplique con el borde de la imagen original, sea el marco lo único que se elimina.

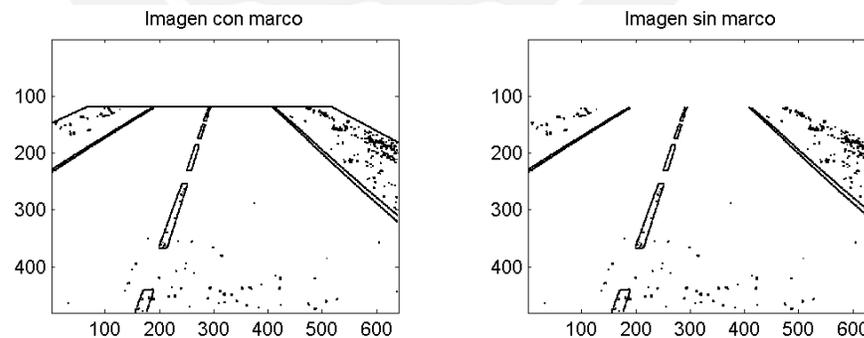


Fig. 3.12. Preparación de las imágenes antes del proceso. Notar la eliminación del marco en la imagen de la derecha.

Posteriormente se analizaron las imágenes. Como primera observación es que el tiempo de procesamiento de este algoritmo, es innecesariamente largo (del orden de los minutos). Es por eso que se hizo una consideración especial:

- Se sabe que el valor umbral se encuentra en la bajada de la campana correspondiente a la región de la pista.
- Se sabe que la región de la pista tiene la mayor área de la imagen, del orden del 80% de la imagen, sin considerar la máscara.
- Se asume que la región más importante del histograma (pista) tiene valores de tonalidad más bajos que la región deseada (líneas).
- Al tener tal presencia en la imagen, la media del histograma va a marcar una notable diferencia entre la región de la pista y el resto (líneas y ruido).
- La intersección del histograma con la media necesariamente va a cruzar la campana de la región de la pista. Sin embargo podría darse el caso que las líneas sean todas de una reducida tonalidad, aumentando la cantidad de píxeles para un tono de gris.
- Los cruces marcaran una región o regiones. En el caso de ser más de una región, se procede a comparar los valores de área de cada una de las regiones encontradas. Como ya mencionó la región de la pista corresponde a la de mayor área.
- Finalmente se toma el valor máximo de esta región como valor inicial de la búsqueda del valor umbral.

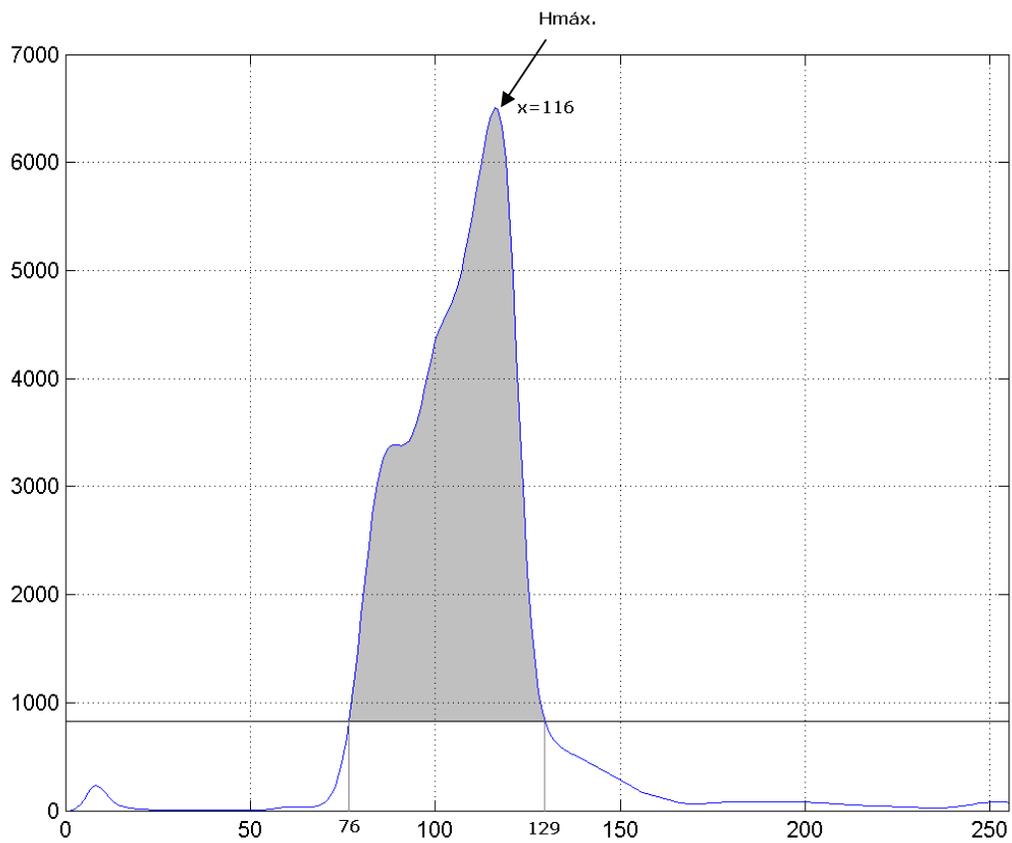


Fig. 3.13. Encontrando un punto de referencia de la región de la pista. El área sombreada corresponde a la interpolación del histograma con la media.

Luego se implementó esta mejora en el algoritmo de umbralización automática y como se esperaba, se notó una reducción importante en el tiempo de procesamiento.

Sin embargo los resultados no mostraron lo que se esperaba:

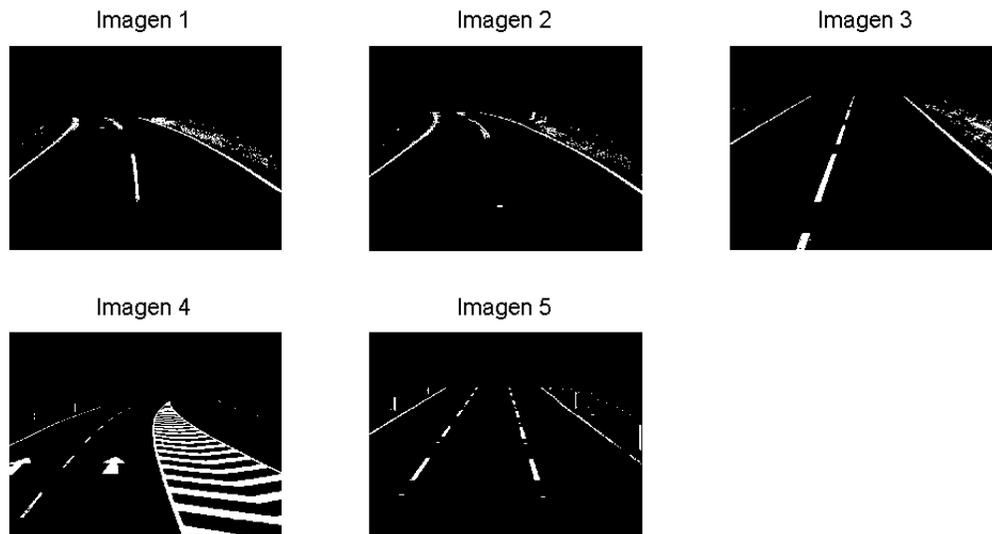


Fig. 3.14. Resultados obtenidos mediante la umbralización por detección de bordes.

Si bien los resultados muestran mejoras con respecto a los métodos anteriores, aún existen algunas líneas no detectadas y la presencia de ruido externo. Por lo lado el algoritmo es poco práctico debido al tiempo de procesamiento, aún con la mejora implementada. Es importante mencionar que aún realizando arreglos y/o métodos de mejora para reducir el tiempo de procesamiento, las imágenes resultantes no van a variar mucho con respecto a lo obtenido. Ya que según [17], las comparaciones son las que van a marcar el valor umbral de tal manera que se maximice la similitud entre las imágenes.

3.1.2.4. Análisis unimodal

Luego de haber probado un método alternativo al análisis del histograma para la umbralización automática, se ha podido comprobar que los resultados y el desempeño del algoritmo no fueron los mejores. Uno de los errores típicos es la falta de detección de algunas líneas que como se mencionó, estas líneas ya no se podrían detectar en el futuro. Es por esto que es necesario seguir revisando técnicas de umbralización, en este caso se realizará nuevamente un análisis de las imágenes mediante el

uso del histograma. Con esto se obtendrá un valor automático para poder umbralizar y poder segmentar la imagen con los objetos deseados.

A partir de las premisas con respecto a las áreas correspondientes para cada región, expuestas en los puntos anteriores, este método tomará los histogramas como unimodales. Se considera como objeto más importante a la región de la pista y la región de las líneas será considerada como objeto secundario. La idea es encontrar la campana correspondiente a la pista y limitarla, estableciendo el valor umbral en el punto donde ésta finaliza. De tal manera que, sabiendo *a priori* que las líneas se encuentran en el lado derecho de la campana, el valor umbral permita segmentar las líneas.

Entonces tal como se desarrolla en [18], se toma el pico de la campana más importante del histograma (que *a priori* se sabe que corresponde a la pista), luego se traza una recta desde este punto hasta el punto $(\max_{H_i=0 \text{ y } H_{i-1} \neq 0}, 0) = (256, 0)$. Finalmente se toma como valor umbral la tonalidad donde la distancia entre la recta trazada y el histograma, es la más alta.

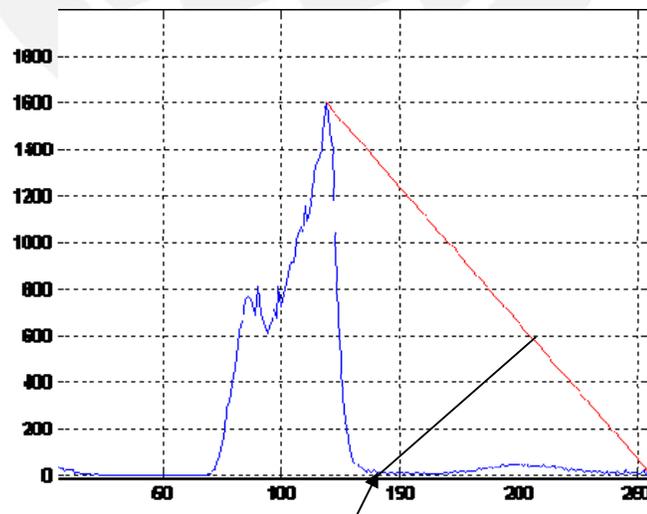


Fig. 3.15. Se traza una línea desde el punto máximo de la región más importante del histograma (correspondiente al pavimento) hasta el valor final (256, 0). Finalmente, se halla el punto más lejano entre la gráfica y la recta.

Se implementó el algoritmo y se obtuvieron los siguientes resultados:

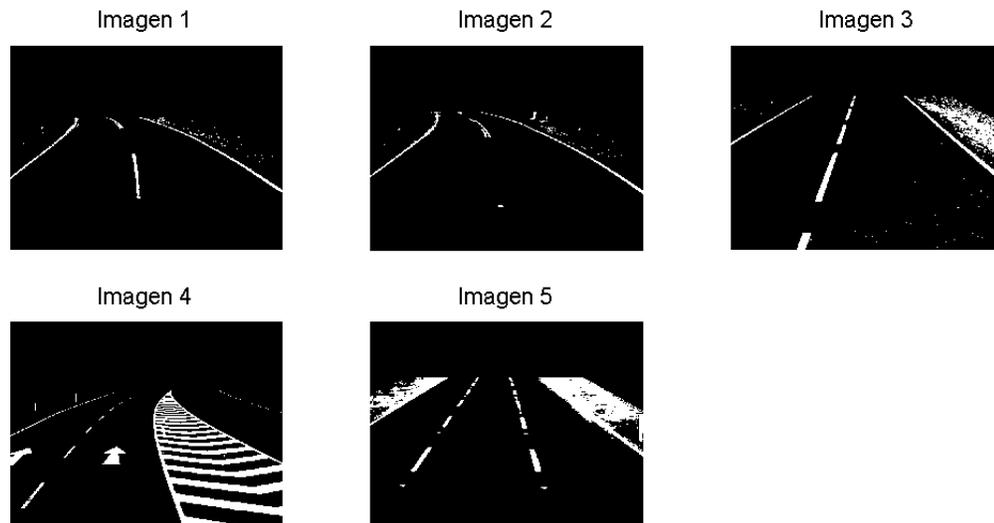


Fig. 3.16. Resultados obtenidos mediante el análisis unimodal.

Como se puede observar, el proceso entrega elementos externos que en algunos casos podrían ser considerables (Imagen 5). Sin embargo las líneas han sido detectadas casi en su totalidad, salvo en la Imagen 4.

Luego de obtener estos resultados y comparándolos con los obtenidos por otras técnicas se llega a la conclusión que dada la variabilidad en la imagen, sea por las distintas tonalidades de las líneas, iluminación, condición de la pista o elementos externos; no es posible trabajar con un valor umbral único para toda la imagen.

Entonces es necesario realizar algunos ajustes en las técnicas de umbralización para poder obtener mejores resultados. Analizándolos se concluye que la técnica "Análisis Unimodal" fue la de que mejor desempeño. Los arreglos necesarios fueron trabajados en conjunto con esta técnica antes implementada en [18].

Finalmente para hacer más preciso el algoritmo de umbralización automática, se utilizó esta técnica de "Análisis Unimodal" dos veces. La primera es para tener una referencia sobre la tonalidad general de las

líneas. Sin embargo luego de esta umbralización aparece bastante ruido. Es por esto que con este valor umbral hallado, se construye una máscara la cual será multiplicada con la imagen en escala de grises. Esta máscara se erosiona para eliminar ruido, luego se dilata de tal manera que se asegure que cada región contenga a las líneas reales. Finalmente cada región se etiqueta y se multiplica con la imagen original.

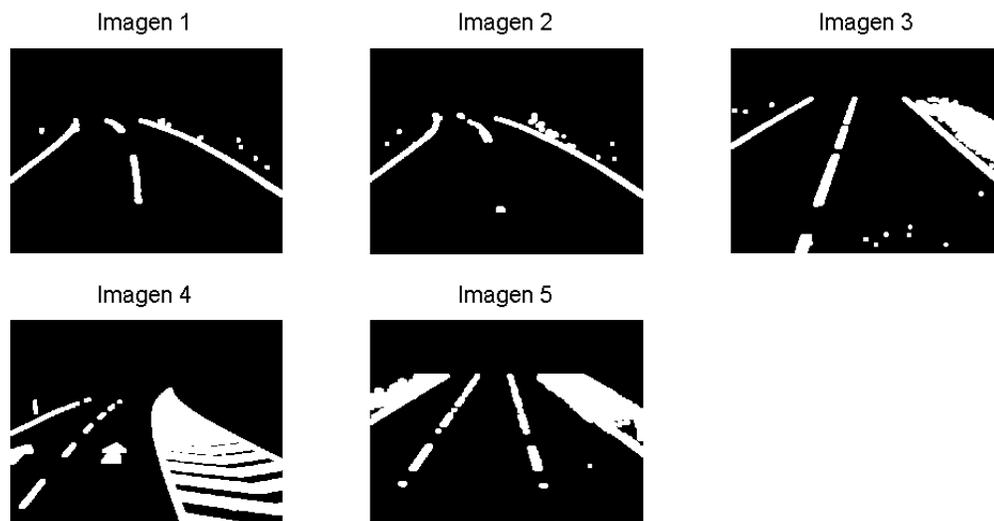


Fig. 3.17. Máscaras obtenidas mediante el análisis unimodal. Estas máscaras se multiplican por la imagen original y se umbralizan nuevamente.

La segunda umbralización es para tener un valor más exacto sobre la tonalidad de las líneas por separado. Primero se etiqueta cada región y luego se umbraliza automáticamente región por región. La idea es obtener un valor umbral por cada línea.

Se implementó el algoritmo y los resultados se muestran a continuación:

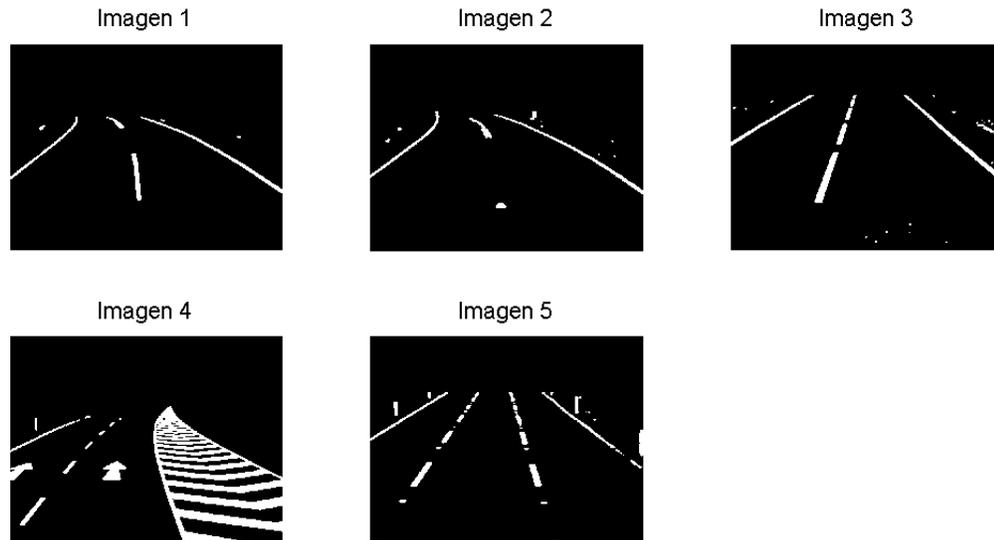


Fig. 3.18. Resultados obtenidos mediante el análisis unimodal mejorado. Notar la disminución de ruido con respecto a la figura 3.14.

Tal como se esperaba, estos resultados han sido notablemente mejores que los obtenidos por los otros métodos de umbralización. Por lo tanto se trabajará en base a esta técnica para la detección de líneas en el presente trabajo.

3.1.3. Segmentación de las líneas.

3.1.3.1. Acondicionamiento de la imagen.

El objetivo de esta parte es separar las líneas del pavimento u otros objetos que no sean relevantes. Luego, almacenar esas líneas mediante tablas.

Es necesario mencionar que el objetivo de la presente tesis está limitado al reconocimiento de la carretera hasta una distancia de 20 a 25 metros del vehículo, ya que es una distancia manejable en el caso de frenadas de emergencia. Por otra parte, un análisis más allá de esta distancia no es relevante en una imagen de la resolución que se está aplicando debido a que muchos datos se pierden por la falta de resolución de píxeles.

Es por esta razón que se aplicará una máscara para considerar únicamente todos los puntos desde $Y=75$ (aproximadamente 22 metros en la imagen desde la posición de la cámara, utilizando las líneas como referencia de medida) hasta $Y=240$.

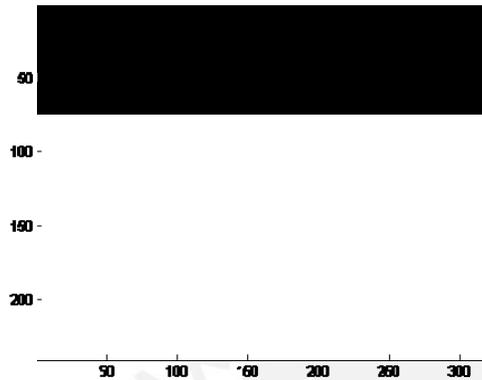


Fig. 3.19. Máscara.

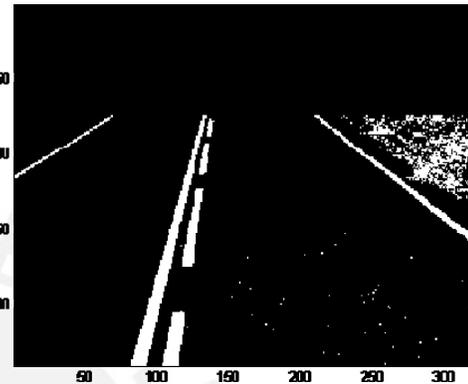


Fig. 3.20. Imagen umbralizada multiplicada por la máscara.

Como se vio en las imágenes anteriores, mucha región del borde se añade a las líneas distorsionándolas. Es por eso que antes de segmentar, necesitamos acondicionar la imagen.

Para acondicionar la imagen y poder obtener una mejor segmentación, se ha creado una máscara para cada línea. Esta máscara es la misma imagen pero dilatada dos veces con una matriz de 5×5 y 3×3 . Luego, esta máscara se segmenta utilizando una función para etiquetamiento [15]. Lo que da como resultado, es que cada línea encontrada de esta máscara tiene aproximadamente el doble de área de la línea original. Entonces la idea es multiplicar estas líneas dilatadas con la imagen original y eliminar el resto de la imagen, de manera que facilite realizar una umbralización independiente para cada línea o grupo de línea que abarque las regiones encontradas en la máscara inicial.

Una vez obtenidas las líneas luego de la umbralización, se acondicionan para darle una forma más parecida a una línea. Se erosiona con una matriz vertical $[0 \ 1 \ 0 ; 0 \ 1 \ 0 ; 0 \ 0 \ 0]$ para eliminar los puntos de tierra

que hayan quedado al borde las líneas que podrían dar mala información, luego se dilata con una matriz suave unitario de 2x2.

La función de detección de bordes de Matlab al detectar los cambios en las tonalidades de gris, se obtiene los bordes de cada objeto. Sin embargo los bordes de los objetos que terminan en los bordes de la imagen no se detectan en su totalidad, ya que el cambio de tonalidad no se puede detectar. Lo cual da como resultado un borde incompleto, es decir un objeto hueco. Es por esto que es necesario diseñar algún algoritmo o función que permita resolver este problema.

En este caso, se diseñó una función que permitía completar el borde de cada objeto. Lo que hace esta función es unir los puntos que se encuentren en los bordes de la imagen.

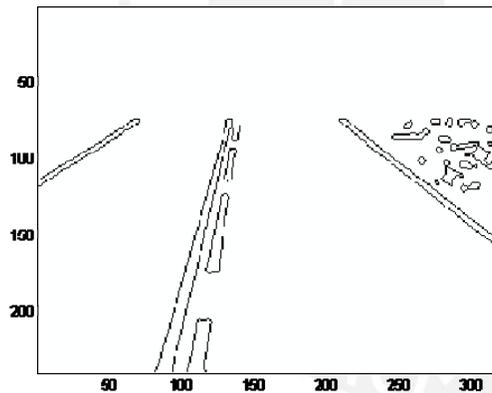


Fig. 3.21. Bordes hallados usando la función de detección de bordes.

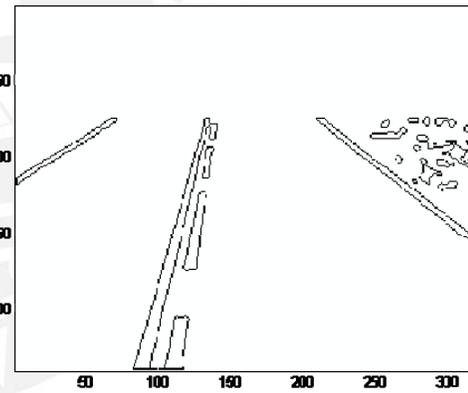


Fig. 3.22. Bordes de los extremos completados con la función para completar el borde.

Notar la diferencia en los bordes en las imágenes Fig. 3.21 y Fig. 3.22.

3.1.3.2. Segmentación de las líneas.

Se utilizó una función de etiquetamiento ya que detecta regiones cerradas y las etiqueta con un tono de gris distinto para cada región encontrada.

Entonces, se obtuvo:

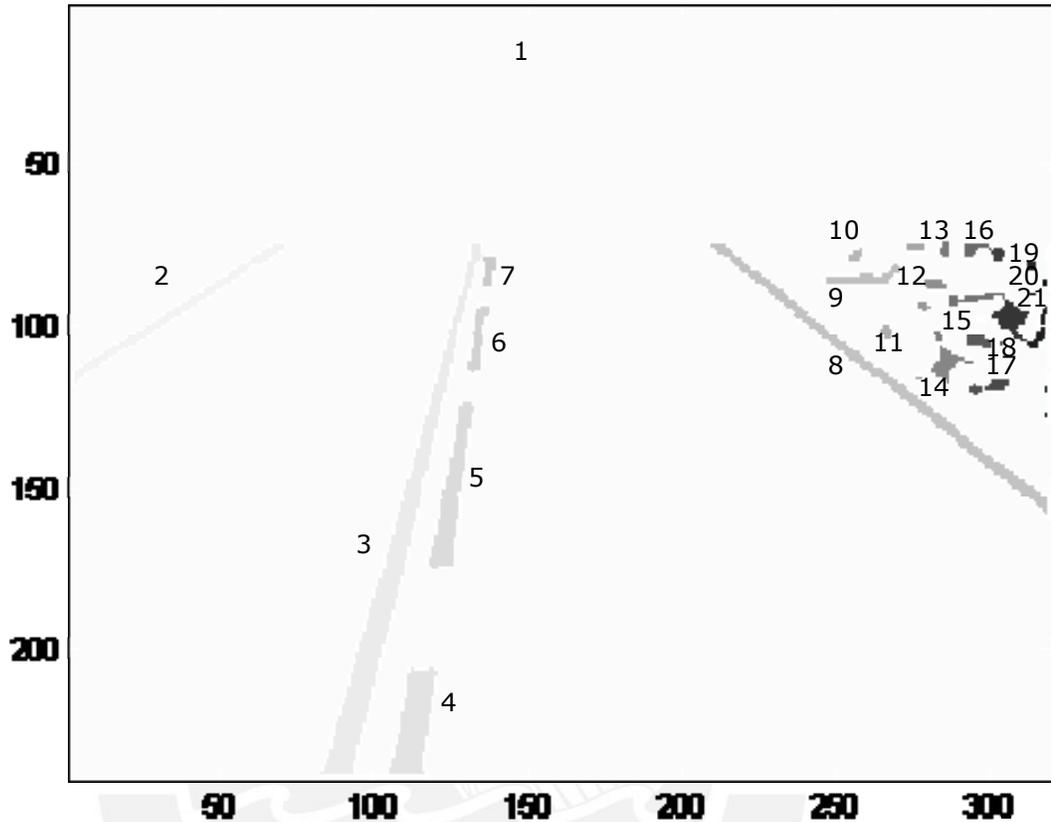


Fig. 3.23. Imagen segmentada debido al etiquetamiento.

Esta función utiliza distintas tonalidades de gris para diferenciar cada objeto encontrado.

Analizando la imagen, se deduce que la función encontró 23 objetos con las siguientes áreas (cantidad de píxeles por objeto).

Área de los objetos encontrados, luego de aplicar BWLABEL					
N° objeto	Área	N° objeto	Área	N° objeto	Área
1	0	9	26	17	6
2	190	10	10	18	43
3	893	11	12	19	1
4	299	12	20	20	17
5	267	13	36	21	5
6	61	14	12	22	3
7	25	15	13	23	2
8	408	16	9		

Se nota que existen muchos objetos con áreas muy pequeñas que complican el procesamiento. Y la diferencia con el área de las líneas es notable. Entonces mediante pruebas con distintas imágenes se tomó el valor de área mayor de 40 para discriminar los objetos que no son líneas.

Entonces de acuerdo a la tabla anterior los objetos de interés son: 2, 3, 4, 5, 6, 8 y 18. El resto de objetos no son relevantes.

Dibujando los objetos, se tiene:

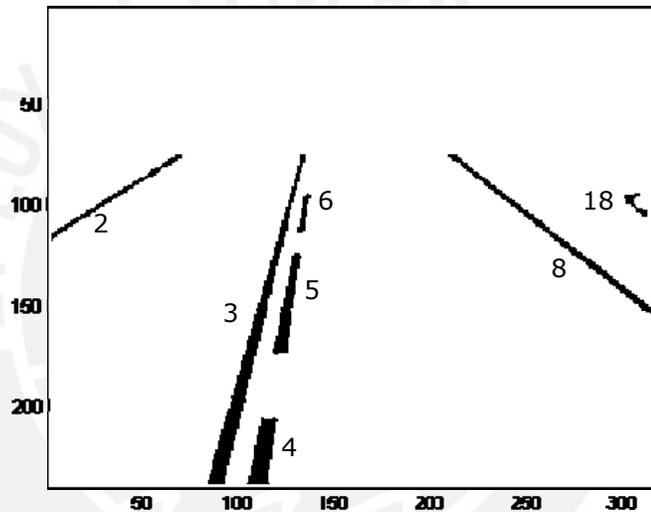


Fig. 3.24. Resultado de los objetos con Área > 40

3.2. Discriminación de objetos y clasificación de las líneas.

En esta parte se analiza cada objeto encontrado por separado mediante sus propiedades (ángulo de inclinación, longitud, ubicación del centroide, ancho de la línea). Luego, mediante una secuencia de comparaciones se concluye si es línea o no. En el caso que sea línea, se clasifica si es continua o discontinua. Lo cual se podrá visualizar en una tabla por separado para cada tipo de línea y en un gráfico.

Para obtener los datos de ángulo de inclinación, longitud y ubicación del centroide se usaron distintas funciones de Matlab.

Para obtener el ancho de línea se implementó una función, la cual proporciona la cantidad de píxeles por cada fila. Esta función da como resultado una tabla del ancho de cada fila y otra tabla indicando a qué fila corresponde ese valor de ancho.

3.2.1. Análisis del ancho de cada línea.

Entonces como primer factor de discriminación en esta parte de desarrollo, se establece que una línea debe tener un ancho constante ya que todas las líneas pintadas en la carretera tienen el mismo ancho. Llevando esto a la imagen, por la perspectiva se nota que mientras Y aumenta, el ancho también. Con lo cual se concluye que los valores obtenidos de ancho por la función antes mencionada, no debe compararse con el promedio, sino con una recta ya que para cada valor de Y es normal que los valores de ancho cambien también. Lo que se debe analizar es cómo es que éstos cambian.

Por lo tanto, los valores de ancho se aproximan a una recta mediante un ajuste polinomial de primer orden usando una función determinada y luego otra función para obtener los resultados de la ecuación, con lo que se obtienen los valores de la recta. En seguida se consigue el error entre el valor de la recta y los puntos medidos. Finalmente se compara y si el ancho medido tiene un error mayor de ± 2 , se concluye que el objeto analizado no es una línea. El valor de ± 2 se encontró empíricamente, probando con diversas muestras.

Para mostrar esta etapa de modo gráfico, se toma de muestra una línea de forma manual y se analiza el ancho de ésta:

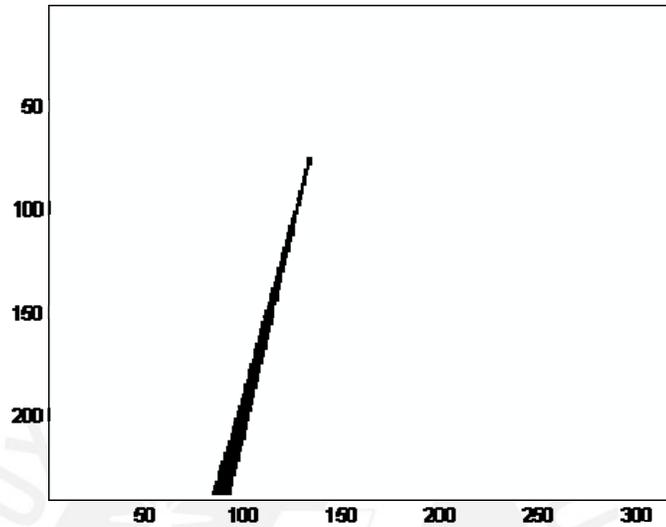


Fig. 3.25. Línea de prueba casi ideal para analizar el ancho.

Se nota que el ancho de la línea es mayor a medida que el eje Y aumenta.

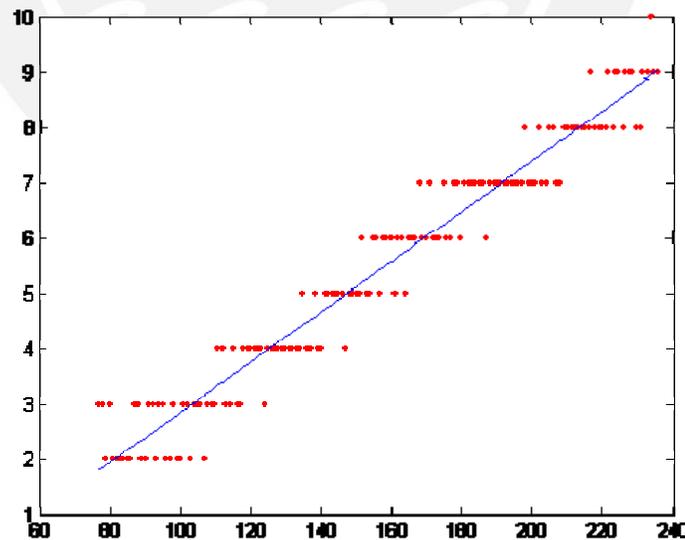


Fig. 3.26. Ajuste lineal del ancho de la línea de prueba.

Utilizando el mismo método en un objeto que se sabe que no es línea, se obtuvieron los siguientes resultados:

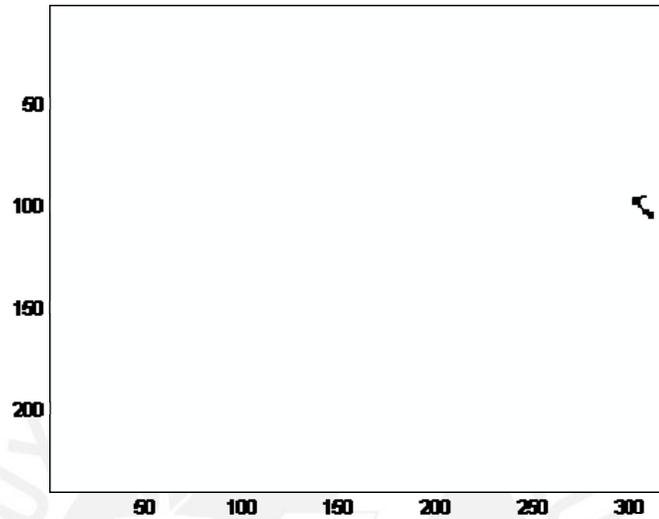


Fig. 3.27. Objeto de prueba para analizar el ancho.

Se analiza el ancho para compararlo con el gráfico de la imagen anterior:

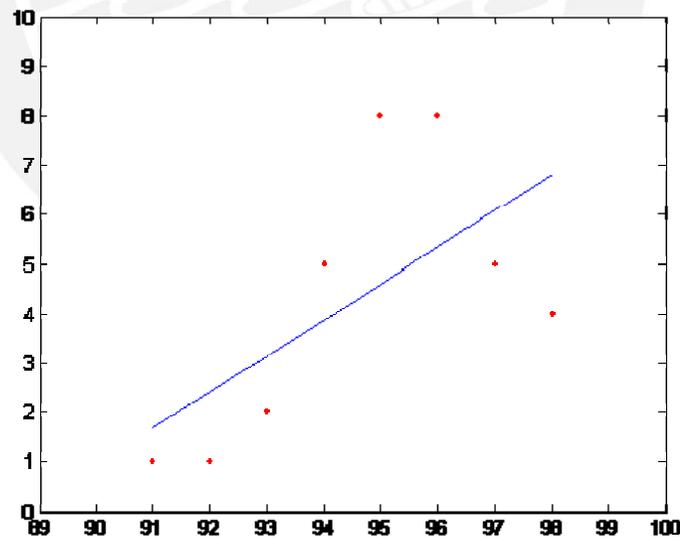


Fig. 3.28. Ajuste lineal del objeto de prueba.

Se observa que el error es muy grande debido que el ancho del objeto no presenta ninguna lógica para analizar por lo tanto el ajuste lineal es uno de los métodos para descartar si el objeto es una línea o no.

3.2.2. Análisis del ángulo de inclinación de cada línea con respecto a su ubicación.

El siguiente método para la discriminación de objetos es analizando el ángulo de inclinación con respecto a su ubicación en el imagen. Es decir, observando la imagen se nota que las líneas tienen su cierto ángulo de inclinación dependiendo donde se encuentren. Entonces se establece un rango de ángulos entre los cuales debe estar la línea para que cumpla con la función de ser parte de la carretera.

Analizando la imagen:

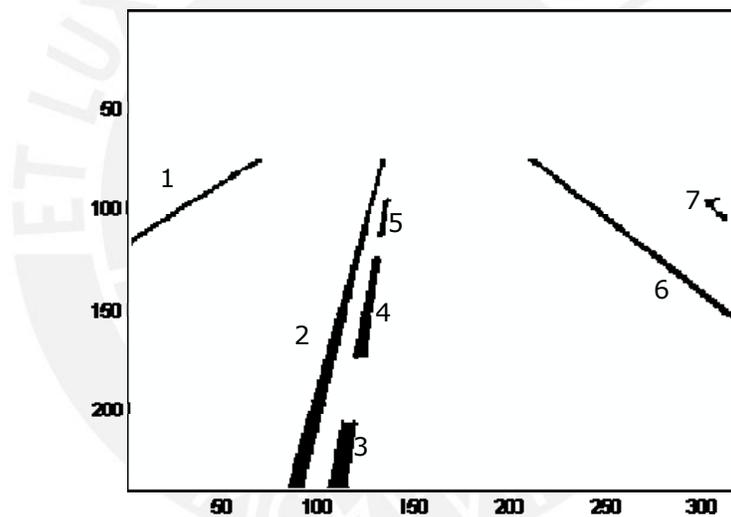


Fig. 3.29. Imagen de prueba para desarrollar este método.

Se obtuvieron los siguientes datos:

Nº objeto	Ángulo de Inclinación	Centroide (x, y)
1	31.23	(36.24, 95.07)
2	74.5	(105.83, 175.19)
3	78.84	(112.97, 222.74)
4	79.7	(125.36, 151.31)
5	79.72	(133.92, 103.87)
6	-36.97	(268.02, 117.78)
7	76.06	(304.88, 95.3)

De esta manera se analizó distintas imágenes y estableció lo siguiente:

- Para objetos con centroide de x entre 0 y 75, el ángulo de inclinación debe estar entre 25° y 90° .
- Para objetos con centroide de x entre 75 y 245, el ángulo de inclinación debe estar entre -90° y -50° ó entre 50° y 90° .
- Para objetos con centroide de x entre 245 y 320, el ángulo de inclinación debe estar entre -90° y -25° .

Utilizando este algoritmo de discriminación de objetos, se logró eliminar el objeto N° 7 de la imagen de prueba.

3.2.3. Clasificación de las líneas encontradas.

Es parte es para clasificar cada línea encontrada en uno de los grupos: líneas continuas o líneas discontinuas.

Para la realización de esta etapa se usó como único factor de selección, la longitud de cada una de las líneas encontradas y comparando este valor con uno previamente establecido. Es necesario resaltar que en esta parte la imagen analizada debe estar compuesta por líneas únicamente.

Para la imagen anterior, se obtiene la longitud para cada línea encontrada.

N° objeto	1	2	3	4	5	6
Longitud	92.1	182	38.4	59.1	24.5	156

Como primera observación se menciona que la tabla tiene solo datos de 6 objetos. Esto es debido a que el objeto N° 7 fue discriminado en la sección 3.2.2. porque no cumplía con las condiciones establecidas.

Tomando estos datos para distintas imágenes se llegó a concluir que la longitud máxima para una línea discontinua es de 77. Entonces para la

clasificación de las líneas se toma un valor de 83 para asegurar que todas las líneas estén bien clasificadas. La condición queda de la siguiente manera.

- Para todas las líneas que tengan longitud mayor de 83 serán registradas como 'Líneas continuas'.
- Para todas las líneas que tengan longitud menor de 83 serán registradas como 'Líneas discontinuas'.

Las tablas 'continuas' y 'discontinuas' del programa elaborado en Matlab entregan los siguientes resultados.

continuas = [1 2 6]

discontinuas = [3 4 5]

Gráficamente se obtuvieron dos imágenes separadas:

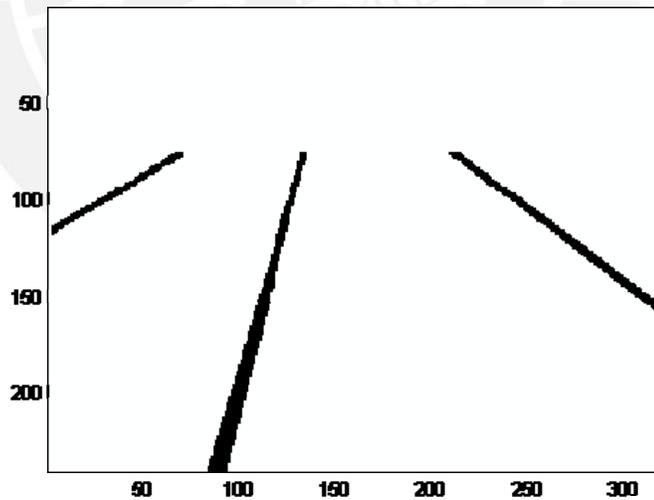


Fig. 3.30. Imagen resultante con las líneas continuas encontradas.

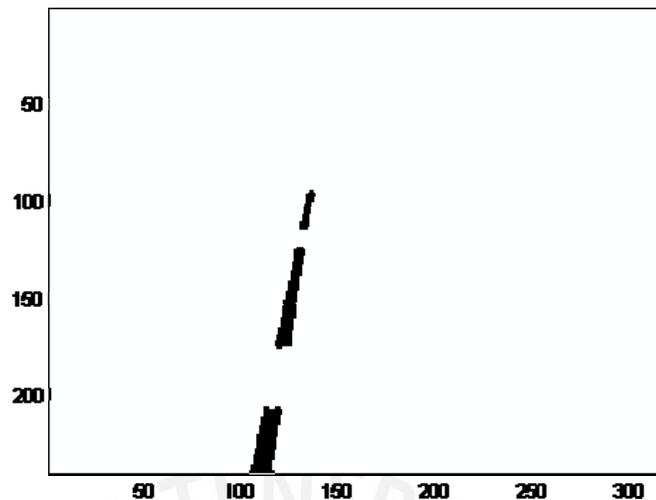


Fig. 3.31. Imagen resultante con las líneas discontinuas encontradas.

3.3. Agrupación de las líneas discontinuas y reconstrucción de la imagen.

3.3.1. Análisis de las líneas discontinuas.

Hasta ahora en lo que se ha visto en el presente trabajo, a partir de una imagen a color de la carretera tomada por una cámara en la parte frontal del vehículo, se ha podido seleccionar las líneas que están presentes, tanto de límite de carril como de límite del pavimento. Luego se han separado en líneas continuas y líneas discontinuas según corresponda. Sin embargo existe un vacío que falta considerar: garantizar que las líneas discontinuas encontradas sean de un solo carril. Es decir, en el caso que la pista sea de tres carriles y el vehículo esté en el carril del centro, habrá líneas discontinuas en ambos lados. El sistema debe ser capaz de diferenciar ambos grupos de líneas. Entonces eso es lo que se verá en esta parte. Lamentablemente la imagen de muestra que se ha utilizado para generar todo el desarrollo no tiene las características antes descritas por lo que el algoritmo de agrupación de líneas discontinuas según el carril no va a ser mostrado con imágenes en esta sección, sólo se mostrará en qué consiste. Sin embargo para certificar que el algoritmo realmente funciona se realizarán diversas pruebas en la sección 4.

3.3.1.1. Obtención de datos.

Tal como se hizo en la discriminación de objetos, en esta parte del análisis también es necesario recopilar datos sobre las características de las líneas discontinuas encontradas. De esta manera se pueden establecer comparaciones entre cada dato y agrupar las líneas según el carril.

Al igual que en la sección 3.2., se utilizaron funciones de Matlab para obtener los datos requeridos. En este caso sólo se va a trabajar con la posición de la línea con respecto a la imagen (centroide) y con el ángulo de inclinación.

Se obtuvieron los siguientes datos:

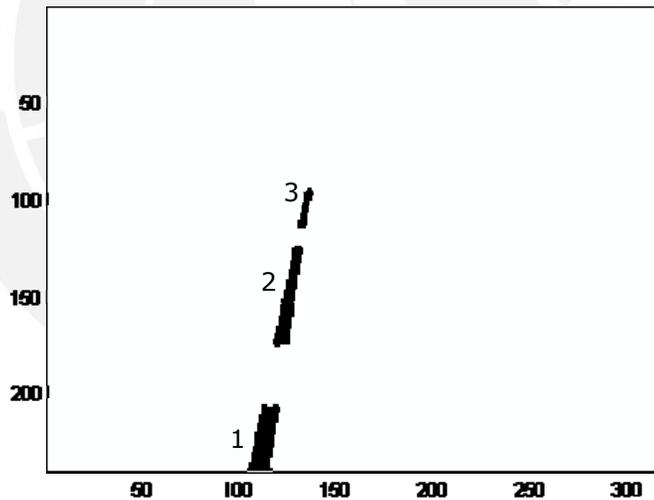


Fig. 3.32. Imagen con las líneas discontinuas que deben ser agrupadas.

Nº Línea discontinua	1	2	3
Ángulo de inclinación	78.82	79.71	79.39
Centroide (x, y)	(112.89, 222.74)	(125.36, 151.31)	(133.94, 103.87)

3.3.1.2. Método a emplear para el agrupamiento de las líneas.

Se nota que los ángulos son muy parecidos, por lo que se establece que la diferencia entre cada ángulo debe ser menor o igual a 15 (este valor se obtuvo mediante varias mediciones a líneas de un solo carril). Luego se agrupan las líneas que cumplen esta comparación y se procede a evaluar el siguiente dato en cada grupo, que es la posición de cada línea con respecto a la imagen.

Se utiliza la siguiente lógica: para que las líneas sean de un carril deben estar alineadas hacia un lado (izquierdo o derecho) por la perspectiva. Entonces se toman los datos hallados del centroide y se hace un ajuste polinomial de grado uno, ya que estas líneas si no son parte de un camino recto, las curvas no son muy pronunciadas y permite la aproximación a la recta. Se ubica el punto que supera una distancia determinada (empíricamente hallada de 5) con respecto al ajuste lineal que se dio y se establece como no perteneciente del grupo de líneas que se había hallado.

3.3.2. Reconstrucción de la imagen.

Luego de haber pasado por todas las etapas de análisis de la imagen, se obtiene finalmente dos tablas de datos (una para líneas continuas y otra para líneas discontinuas) en las que se muestra los puntos medios de cada línea, listos para ser unidos en la imagen final.

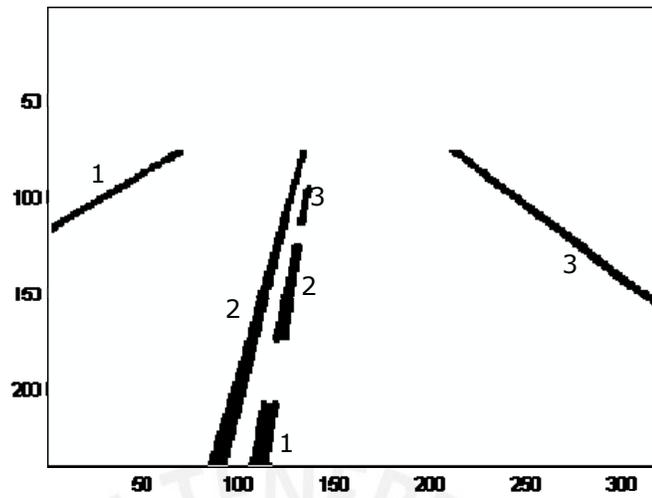


Fig. 3.33. Imagen con las líneas encontradas.

Nº Línea continua	1	2	3
Centroide (x, y)	(36.08, 95.16)	(105.75, 175.39)	(267.74, 117.58)

Nº Línea discontinua	1	2	3
Centroide (x, y)	(112.89, 222.74)	(125.36, 151.31)	(133.94, 103.87)

Por último, se unen las líneas correspondientes y se dibujan en la imagen original.

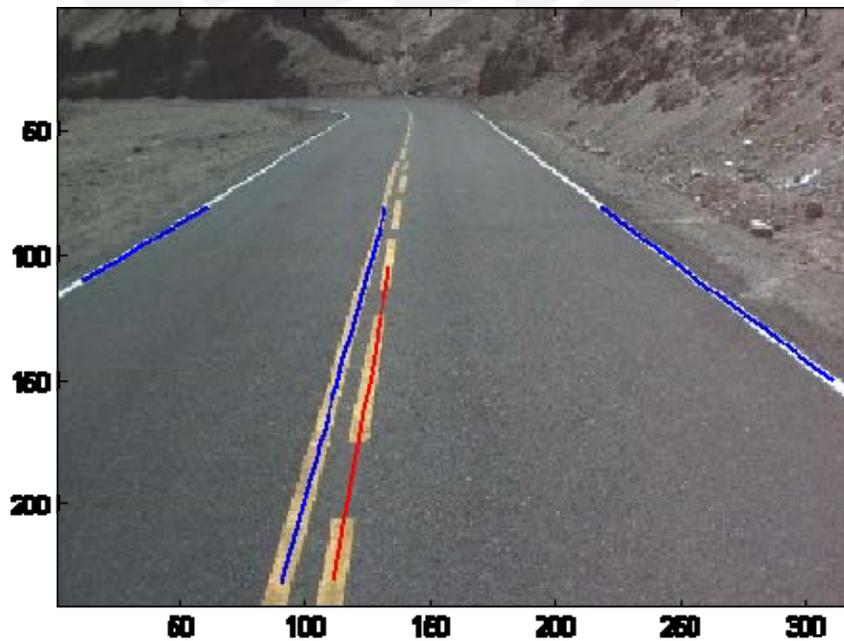


Fig. 3.34. Resultado final.

En la figura Fig. 3.34., la línea roja indica que es línea discontinua y la línea azul indica que la línea es continua. Ambas líneas fueron encontradas mediante el algoritmo de reconocimiento de líneas de la carretera desarrollado en este trabajo.



CAPÍTULO 4

CORRECCIONES Y EVALUACIÓN FINAL DEL ALGORITMO

4.1. Vista preliminar del algoritmo desarrollado

El algoritmo de detección de las líneas desarrollado en el capítulo 3 del presente documento, fue probado con 20 imágenes que cumplieran con el mismo formato de captura de imagen: ángulo de inclinación de la lente, tipo de lente, altura de la lente. Las imágenes correspondían a diversos paisajes de la misma carretera que la que se usó en un principio para desarrollar el algoritmo.

El resultado de las pruebas con estas imágenes fue de una eficiencia de 45%. Lo cual es un valor bajo para un sistema de este tipo.

Luego, se analizaron los errores y se obtuvo lo siguiente:

- 50% debido al algoritmo de detección.
- 30% debido a la umbralización automática.
- 10% debido a la discriminación de figuras fuera del pavimento.
- 10% debido a los formatos de las variables del programa.

Como se puede observar, la mayor cantidad de errores se debe al algoritmo de detección. Sin embargo este problema va a ser el último en analizarse cuidadosamente ya que primero se pueden solucionar problemas más simples y luego probar si es que aún persiste este error.

Las medidas de solución y correcciones que se han considerado para mejorar la eficiencia del sistema, son las siguientes:

- Utilizar imágenes acondicionadas: el análisis se hará únicamente en la región del pavimento. No es necesario procesar toda la imagen.
- Utilizar un filtro pasa-altos con un kernel que permita la detección de cambios bruscos de tonalidad entre píxel y píxel, determinando una

zona o región. Luego aplicar umbralización automática por separado en cada una de las regiones detectadas con el filtro.

- Buscar y reconocer los errores en el algoritmo y buscar formas de solución.

4.2. Correcciones

4.2.1. Acondicionamiento de las imágenes.

Uno de los problemas vistos en el funcionamiento del algoritmo desarrollado en el capítulo 3, fue la presencia de zonas ajenas a la carretera. Por lo que las imágenes que se van a procesar deben mostrar únicamente el pavimento. Los factores externos como tierra, monte, césped, etc. han sido eliminados. En este caso, por no contar con un algoritmo que permita segmentar la carretera, las imágenes fueron manipuladas de forma manual mediante el programa de gráficos Paint.

De forma práctica, esta parte se haría en un proceso separado que permita seleccionar únicamente el pavimento de una imagen eliminando el resto.



Fig. 4.1. Imagen original de la carretera.

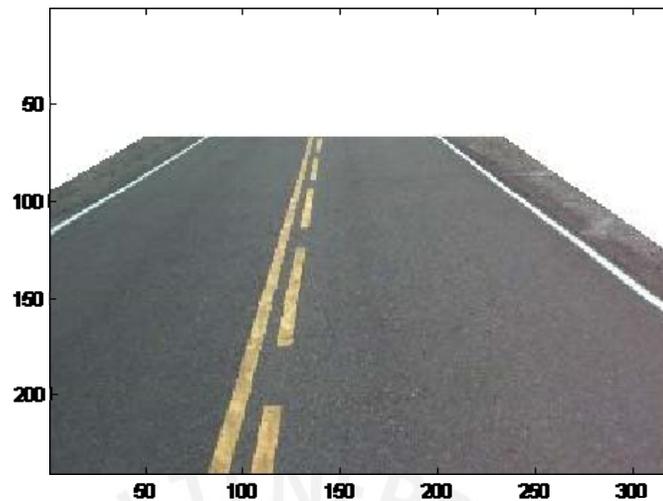


Fig. 4.2. Imagen modificada manualmente usando el programa Paint para eliminar regiones que no son de interés.

4.2.2. Filtro Pasa-altos y Umbralización automática.

Como se vio en el reporte de errores en el punto 4.1., existe una cantidad significativa de errores que se producen debido a la umbralización automática. Esto es debido al uso de un valor fijo de umbral en toda la imagen para hallar la máscara. Lo que sucede es que las líneas no siempre tienen una tonalidad constante en toda la imagen, entonces cuando se umbralice siempre habrán regiones que no se detectan.

Lo que se planteó como solución fue, primero utilizar un filtro pasa-altos con un kernel definido de tal manera que resalte los cambios bruscos de tonalidades comparando píxeles continuos. Esto representa el cambio entre la línea (blanca o amarilla) y el pavimento.

El resultado de esta parte nos da una imagen en la que se definen notoriamente los cambios detectados. Se transforma la imagen a escala de grises y se umbraliza automáticamente de tal manera que la imagen quede binarizada mostrando únicamente las regiones detectadas de cambio de tonalidad. Luego estas regiones se dilatan y se segmentan.

La idea es separar cada una de las regiones y multiplicarla por la imagen original (en escala de grises) por separado. Luego analizar el histograma de esa región obtenida y proceder con la umbralización automática. Y seguir con el proceso para todas las regiones obtenidas. Con esto se obtienen distintos valores de umbralización dependiendo la tonalidad de cada línea, lo cual mejora significativamente la detección. El resultado final es la suma de todos los objetos (líneas) encontrados.

Imagen de Prueba 1:

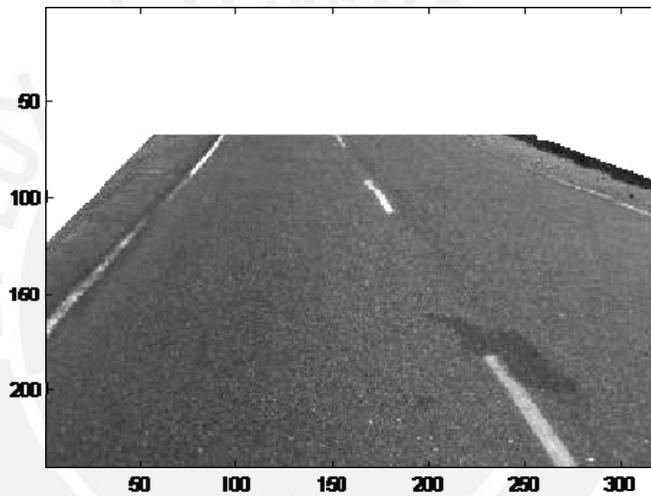


Fig. 4.3. Imagen original en escala de grises usando el plano rojo.

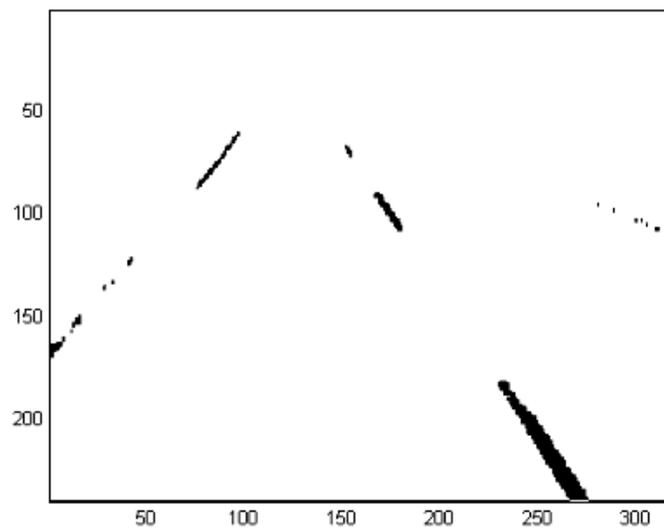


Fig. 4.4. Resultado de la umbralización automática planteada en el Cap. 3

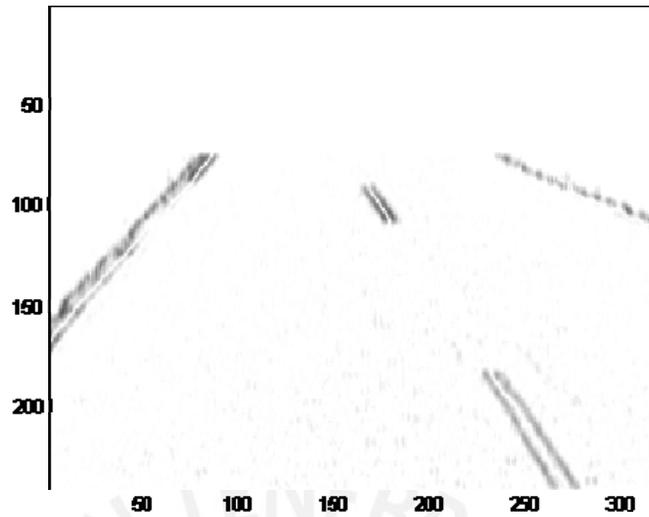


Fig. 4.5. Resultado luego de aplicar un filtro pasa-altos a la Imagen de Prueba 1.

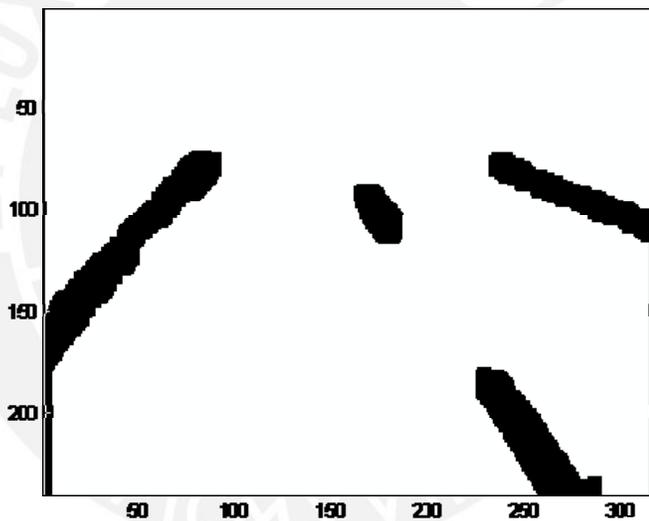


Fig. 4.6. Máscara obtenida de la umbralización y dilatación de la Fig. 4.5. Se utiliza como máscara para seleccionar regiones de la imagen original de manera que se obtiene un valor umbral para cada región.

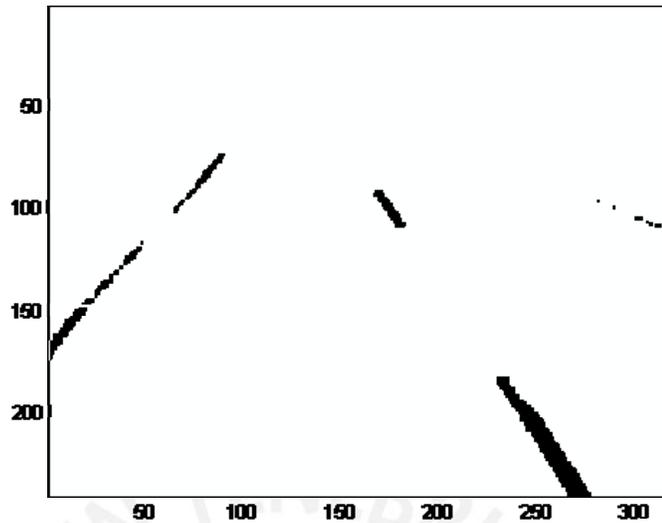


Fig. 4.7. Resultado final de la umbralización usando primero un filtro pasa-altos y luego umbralización automática para cada región por separado. Esta imagen es la suma de cada objeto encontrado en cada región.

Si bien las mejoras se pueden apreciar comparando Fig. 4.4 y Fig. 4.7, realizando diversas pruebas se observó que la matriz utilizada no era muy precisa. Es por esto que se optó por utilizar una matriz que permita detectar todos los cambios bruscos de tonalidad marcando con color oscuro las líneas blancas (Fig. 4.8).

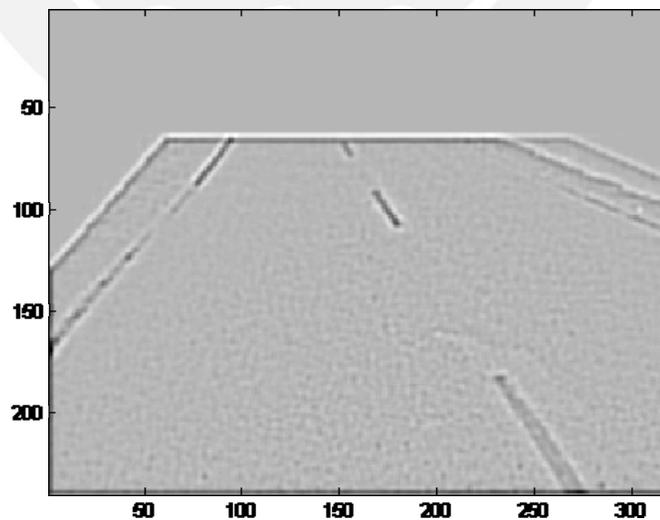


Fig. 4.8. Resultado luego de aplicar filtro pasa-altos con una matriz de detección de cambio de tonalidad.

Se nota que las líneas de tonalidad oscura son las que interesan, sin embargo existen algunas que pueden perjudicar el proceso y son las líneas del borde. Éstas aparecieron debido a la máscara que se hizo manualmente en todas las imágenes simulando un proceso previo. Entonces antes de umbralizar para obtener las líneas de interés es necesario multiplicar la imagen por una máscara que se obtiene de la imagen original. Esta máscara se erosiona para que pueda eliminar las líneas, luego se umbraliza automáticamente (Fig. 4.9).

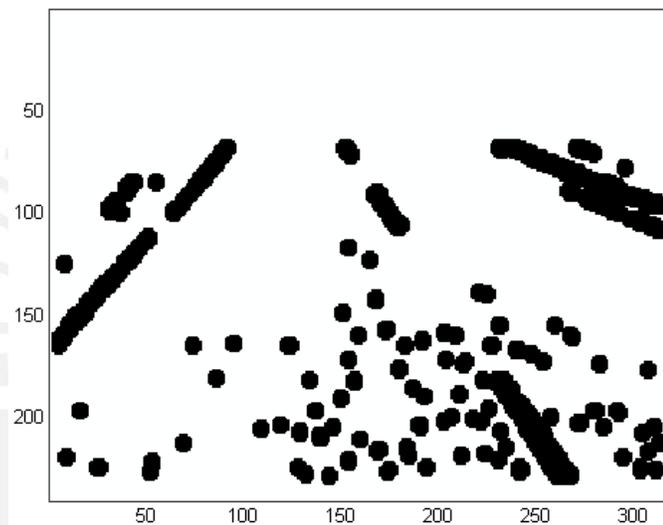


Fig. 4.9. Imagen anterior luego de umbralizar automáticamente y erosionar para eliminar el ruido ocasionado por las distintas tonalidades del pavimento. Con esto se consigue la máscara para umbralizar cada región por separado de la imagen original.

Una vez obtenida esta nueva máscara, previamente dilatada, se segmenta y se hace un recorrido de cada objeto por separado. Este objeto se multiplica por la imagen original, luego se umbraliza. Se suman todos los resultados parciales y se obtienen las líneas buscadas.

4.2.3. Modificando el algoritmo.

Como se mencionó al iniciar este capítulo, la eficiencia lograda en el diseño del algoritmo fue muy baja, es por eso se analizarán los errores para poder tomar las medidas correctivas y aumentar la tasa de eficiencia. Entonces la idea es tomar la base del algoritmo desarrollado y

se darán los ajustes necesarios para reducir significativamente los errores.

Buscando entre las etapas del algoritmo, se encontraron fallas en las siguientes partes:

- **Dilatación:** se utilizaba una dilatación repetitiva, lo cual generaba que algunas líneas que estuviesen muy juntas, se unan. Se redujo la intensidad y la cantidad de las dilataciones, y ya no se vio este problema.

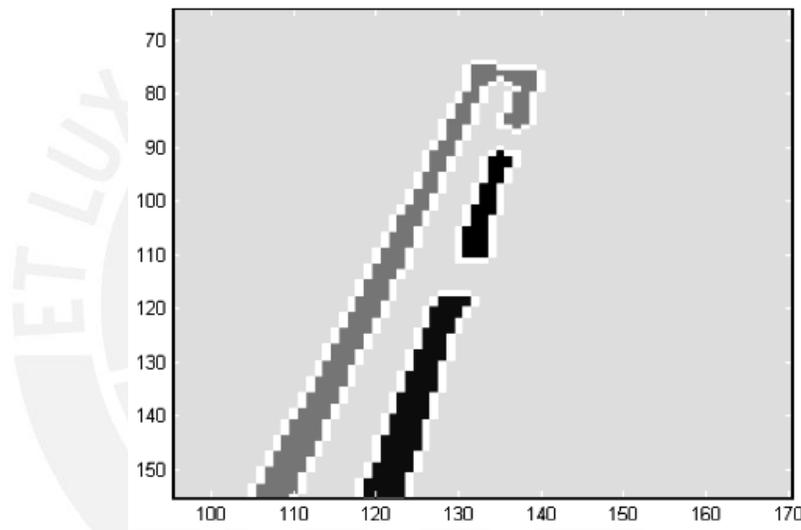


Fig. 4.10. Antes se obtuvo este resultado. Se puede observar que la dilatación hizo que las líneas se junten y luego de la detección de regiones se consideren como una sola región (línea).

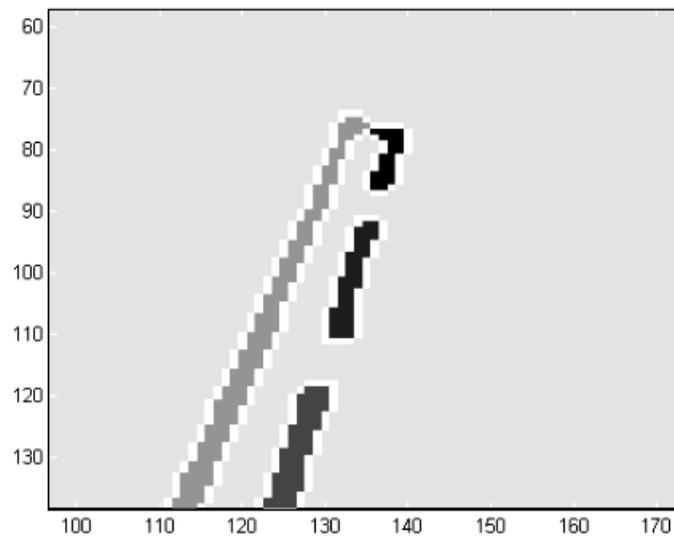


Fig. 4.11. Luego de la corrección, se obtuvo esta gráfica. Esta corrección permitió separar esas líneas que antes estaban juntas.

- **Criterios de discriminación de líneas:** se estaba discriminando líneas cuyos ángulos de inclinación estén en un rango determinado. Se analizaron las líneas y se obtuvieron los datos de los ángulos, luego según estos datos se hicieron los ajustes necesarios del rango de discriminación.

En el desarrollo se planteó:

- Para objetos con centroide de x entre 0 y 75, el ángulo de inclinación debe estar entre 25° y 90° .
- Para objetos con centroide de x entre 75 y 245, el ángulo de inclinación debe estar entre -90° y -50° ó entre 50° y 90° .
- Para objetos con centroide de x entre 245 y 320, el ángulo de inclinación debe estar entre -90° y -25° .

Sin embargo analizando con distintas imágenes, se obtuvieron los siguientes datos:

Imagen de Prueba 2:

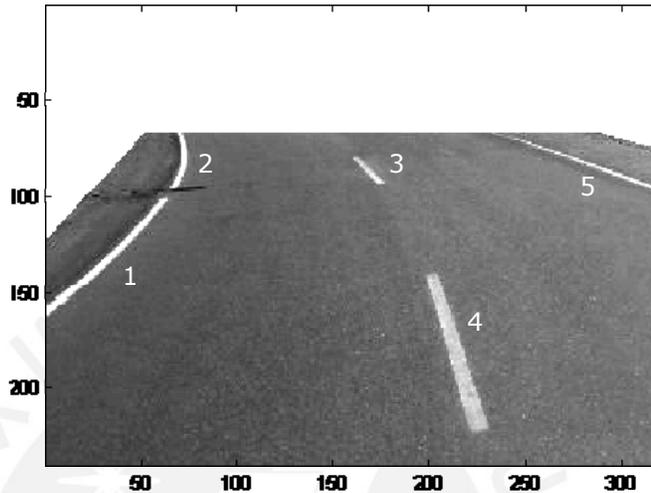


Fig. 4.12. Imagen de Prueba 2. La numeración corresponde a los objetos encontrados.

Datos obtenidos:

N° objeto	Orientación	Centroide (x, y)
1	43.44	33.81 130.17
2	72.95	71.18 85.07
3	-44.6	168.57 85.60
4	-73.19	215.22 183.98
5	-20.06	293.02 85.81

Imagen de Prueba 3:

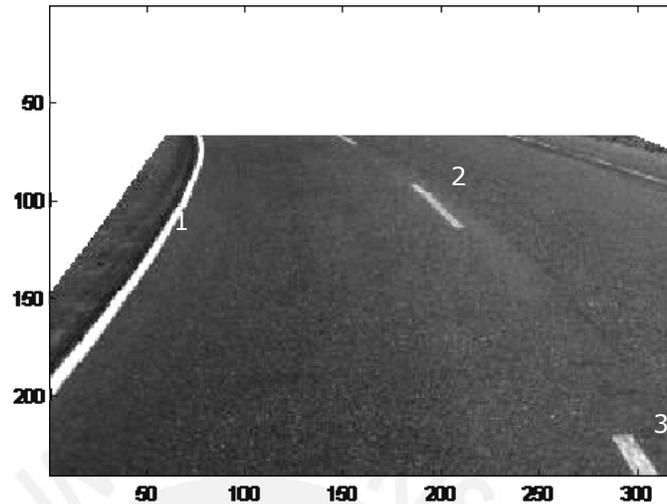


Fig.4.13. Imagen de Prueba 3. La numeración corresponde a los objetos encontrados.

Datos obtenidos:

Nº objeto	Orientación	Centroide (x, y)
1	55.73	38.31 144.39
2	-41.66	197.76 102.07
3	-50.1	298.50 229.34

Se puede observar que según los datos encontrados, el algoritmo no reconocería muchas líneas de las imágenes mostradas anteriormente, sin embargo viendo cada imagen se puede comprobar que sí se trata de líneas y por lo tanto deben poder ser reconocidas. Es por eso que fueron necesarias algunas modificaciones.

El algoritmo cambió a:

- Para objetos con centroide de x entre 0 y 75, el ángulo de inclinación debe estar entre 15° y 90°.
- Para objetos con centroide de x entre 75 y 245, el ángulo de inclinación debe estar entre -90° y -27° ó entre 27° y 90°.
- Para objetos con centroide de x entre 245 y 320, el ángulo de inclinación debe estar entre -90° y -15°.

Además fue necesario agregar dos criterios más, considerando esta vez al ángulo como variable principal y la ubicación en el plano como variable secundaria:

- Para objetos con ángulo de inclinación entre -85° y 85° , el centroide de x debe estar entre 133 y 187.
- Para objetos con ángulo de inclinación entre -78° y 78° , el centroide de x debe estar entre 125 y 195.

- **Ancho de las líneas:** se pudo observar que el algoritmo no reconocía las líneas que si bien estaban ligeramente distorsionadas, se podían distinguir claramente. El problema fue la máxima desviación del ancho permitida, con la aproximación lineal. Se ajustó ese valor establecido previamente, de acuerdo a los datos que se obtenían del análisis de las imágenes.

En el desarrollo se estableció como máximo error permitido entre la recta de ajuste lineal y el ancho medido de cada línea, el valor de 2.5 (hallado empíricamente). A continuación se podrá observar que ese valor se tiene que modificar.

Imagen de Prueba 4:

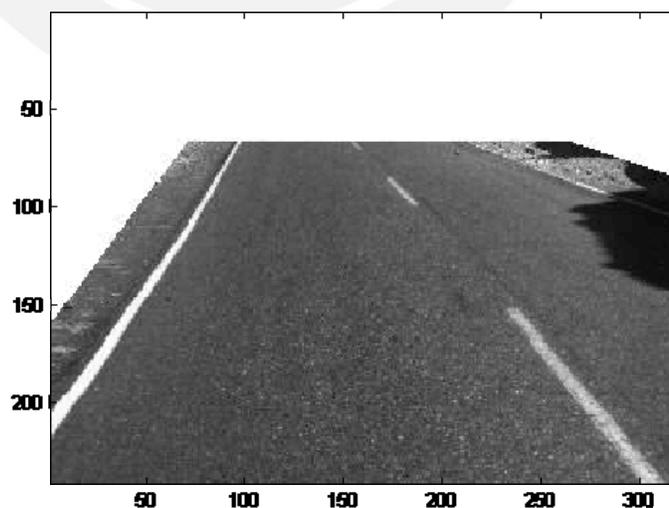


Fig. 4.14. Imagen de Prueba 4 para verificar la variación permitida del ancho de línea.

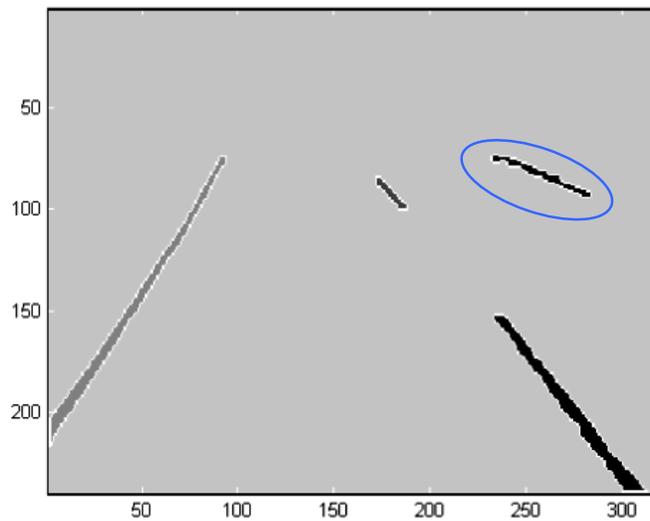


Fig. 4.15. La línea encerrada en la elipse tiene una forma un poco distorsionada. Sin embargo se nota claramente que sí es una línea.

Datos encontrados:

Para la línea mostrada: El error del ancho de línea con la aproximación lineal es de 2.9412.

Sin embargo, es necesario tomar una cierta tolerancia ya que pueden –y van- a existir líneas con la pintura en mal estado. Por lo que el valor máximo permitido de error entre la recta de ajuste lineal y el ancho de cada línea ha sido modificado de 2.5 a 3.5.

- **Longitud de las líneas:** este error si bien no fue muy relevante en lo que respecta al reconocimiento de la línea *per se*, sí lo era para poder entregar un correcto análisis de las líneas en general. Y se ajustó la longitud de línea que clasificaba lo que es línea continua y lo que es línea discontinua.

Realizando las pruebas se pudo comprobar que existían errores para la clasificación de las líneas.

Imagen de Prueba 4:



Fig. 4.16. Imagen en la cual se presentaba un error en el momento de clasificar la línea.

En la imagen anterior se muestra en la elipse a la línea la cual fue confundida con una línea continua. El problema fue el algoritmo clasificaba a todas líneas encontradas mayores de una longitud de 90, como líneas continuas.

Hallando la longitud de la línea, se obtuvo: 127.37

Se nota que la diferencia es significativa así que la modificación de este valor máximo fue inminente. Por lo tanto, se estableció 135 como valor de diferencia entre línea continua y línea discontinua.

Observación:

Es necesario resaltar que una buena clasificación de líneas no se puede dar considerando únicamente esta diferencia (longitud a un valor determinado). Los resultados son muchas veces errados ya que existe ambigüedad entre una línea continua situada en la parte superior de la imagen y una discontinua situada en la parte inferior de la imagen. Esto es lo que se llama perspectiva.

Por ejemplo, se sabe que la línea es continua pero en el momento de la medición por la cantidad de píxeles en la imagen, es fácilmente confundible con una línea discontinua. Es por esto que es necesario establecer un criterio más elaborado para la clasificación.

Lo que se hizo, fue relacionar el tamaño de cada línea con su centro de gravedad. Así, una línea discontinua que se encuentra en la parte superior de la imagen debe tener una longitud corta (valor esperado) a pesar que la longitud de cada línea en la carretera es la misma. Entonces considerando la perspectiva, si se mide la línea por la cantidad de píxeles y supera este valor esperado entonces, la línea no es discontinua.

Para establecer este valor esperado, se hace una fórmula que responde a la ecuación de la recta. Los valores se toman de la medición de una imagen. Tomando los datos obtenidos en la Fig. 4.16. (ya que esta es una imagen casi ideal), relacionando el centroide con la longitud. Luego se aplica la fórmula, reemplazando el valor 'Y' del centroide, puesto que la perspectiva cambia de abajo hacia arriba de la imagen (también cambia del centro a los lados, pero no se está considerando esto). Y la fórmula nos da como resultado el valor de longitud esperado para esa línea.

Luego se compara: si el valor medido es mayor que el valor esperado, entonces la línea no es discontinua.

Datos obtenidos de la Fig. 4.16.

N° objeto	Longitud	Centroide(x, y)
1	184.86	41.90 152.76
2	32.08	179.56 92.21
3	127.37	274.57 199.80

Tomando como referencia los objetos 2 y 3:

Y	Longitud
92.21	32.08
199.80	127.37

Aproximando a valores enteros y tomando una tolerancia:

Y	Longitud
90	37
200	135

La fórmula quedaría: $(Y - 90) = 1.1*(X - 37)$, donde X es el valor esperado.

Volviendo a la imagen de prueba, los valores esperados de longitud para cada línea serían:

Nº Obj.	Longitud	Centroide (x, y)	Longitud esperada	Clasificación
1	184.86	41.90 152.76	92.05	Continua
2	32.08	179.56 92.21	37.00	Discontinua
3	127.37	274.57 199.80	134.81	Discontinua

- **Distinción de colores:** como parte de poder obtener un análisis más completo de las líneas de la carretera, se añadió un algoritmo que distingue y clasifica las líneas discontinuas según sean blancas o amarillas. Se analiza el plano azul de la imagen y se umbraliza con un valor muy alto. Ya que se sabe que en el plano azul el color amarillo no se ve, sin embargo el color blanco sí. Entonces para poder asegurar que no es amarillo se utiliza un valor umbral muy alto ya que además puede pasarse alguna parte de la pintura de la línea que por el uso se volvió gris. Entonces se observa, si no quedó nada de la umbralización, se concluye que la línea es amarilla. En el caso contrario, la línea es blanca. Los resultados de esta parte se podrán observar en el punto 4.3. con los resultados finales de todo el algoritmo.

- **Agrupamiento de las líneas:** se observó en las primeras pruebas realizadas que existían errores reconocer líneas como partes de un grupo cuando no lo son. Es por esto que fue necesario agregar una restricción más al proceso de agrupamiento: al agrupar las líneas, éstas deben tener una distancia máxima de centroide "x" de 90 (valor hallado empíricamente) entre cada una. También se consideró que las líneas dentro de un mismo grupo no deben tener una distancia menos de centroide "y" de 10.
- **Reconstrucción de la imagen:** únicamente a fin de mejorar la presentación de los resultados obtenidos, se agregó un procedimiento para alargar las líneas dibujadas ocupando el espacio de la imagen analizada. Lo que se hizo fue, una vez ordenados los puntos de la línea hallada, aproximarlos a una ecuación polinomial de segundo grado tomando como variable la componente x. Luego se iguala la ecuación con los puntos hasta donde se quiere expandir la línea: arriba 65 y abajo 240. Con lo que se obtiene el valor de x. Se agregan estos nuevos puntos y se dibuja la nueva línea encontrada como resultado final.

4.3. Resultados final del algoritmo.

En esta sección, se mostrará un resumen de los resultados obtenidos luego de analizar diversas imágenes. Estas imágenes se han separado en tres grupos distintos (A, B, C).

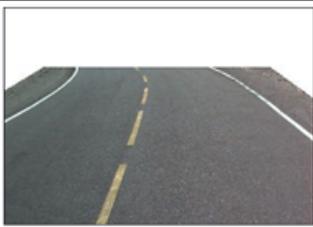
GRUPO	RECTA	CURVA
<p>A</p> <p>Total 35 Imágenes</p>	 <p>26</p>	 <p>8</p>
<p>B</p> <p>Total 15 Imágenes</p>	 <p>8</p>	 <p>7</p>
<p>C</p> <p>Total 18 Imágenes</p>	 <p>14</p>	 <p>4</p>

Fig. 4.17. Imágenes utilizadas para la obtención de resultados finales del sistema desarrollado. Se dividen en tres grupos: A con 35 imágenes en total, B con 15 imágenes y C con 18.

El grupo A muestra las imágenes que se usaron en el diseño del algoritmo, tienen el mismo formato de captura y son de la misma carretera. El grupo B y C son de formatos y carreteras diversas.

Es necesario mencionar que no se usó el mismo código para los tres grupos. La lógica y el orden del proceso es el mismo sin embargo, dado que el total de las imágenes responde a distintos patrones como el ángulo de captura, la altura del lente e incluso la situación del lente en el contexto, es necesario cambiar algunos parámetros para cada grupo.

A continuación se presenta el resultado del algoritmo dividido entre cada grupo. Dado la cantidad de imágenes a mostrar se mostrará una tabla de los resultados obtenidos al procesar las imágenes, luego cuatro imágenes seleccionadas al azar y un resumen del proceso en general para el grupo

de imágenes determinado. Además se realizarán las mismas pruebas utilizando imágenes 640x480 con el fin de buscar mejores resultados cuando el factor tiempo no es un problema para el procesador adecuado. Finalmente a manera de resumen final, se mostrará una tabla indicando las eficiencias obtenidas en cada caso y una imagen representativa.

Los resultados se indican en términos de eficiencia tanto para el algoritmo de detección de líneas como para el de clasificación de éstas. Para calcular estos resultados se hizo un promedio de los resultados de cada imagen por separado (tabla de resultados). Los criterios de evaluación se tomaron considerando la cantidad de líneas presentes y las detectadas, la posición de la línea en la imagen –líneas lejanas-, la condición de la pintura y el estado de la pista. Se está considerando también una eficiencia que responde a la relación de las líneas dibujadas con las reales, sin embargo esta estimación sólo se presentará como resultado para cada grupo de imágenes.

4.3.1. Grupo A

4.3.1.1. Imágenes 320x240

Tabla de resultados:

Imagen	Eficiencia [%]		Problema
	Reconocimiento	Clasificación	
A.1	100	100	-
A.2	95	100	Pintura deteriorada
A.3	100	80	Pintura deteriorada
A.4	100	80	Pintura deteriorada
A.5	100	100	-
A.6	100	80	Pintura deteriorada
A.7	100	100	-
A.8	100	100	-
A.9	80	100	Desprendimiento de tierra
A.10	100	100	-
A.11	75	100	Clara tonalidad de la tierra
A.12	100	100	-
A.13	75	100	Deformación de la línea debido a la clara tonalidad de la tierra
A.14	90	80	Pintura deteriorada

A.15	100	100	-
A.16	95	100	Reducido tamaño de la línea
A.17	100	100	-
A.18	100	100	-
A.19	100	80	Problema debido a sombras
A.20	75	100	Clara tonalidad de la tierra
A.21	80	100	Pintura deteriorada
A.22	80	80	Pintura deteriorada y reducido tamaño de la línea
A.23	100	100	-
A.24	80	100	Pintura deteriorada
A.25	100	100	-
A.26	80	100	Reducido tamaño de la línea
A.27	95	100	Reducido tamaño de la línea
A.28	100	100	-
A.29	100	100	-
A.30	100	100	-
A.31	75	100	Deformación de la línea debido a la clara tonalidad de la tierra
A.32	80	100	Pintura deteriorada
A.33	100	100	-
A.34	100	100	-
A.35	100	100	-

Imágenes obtenidas:

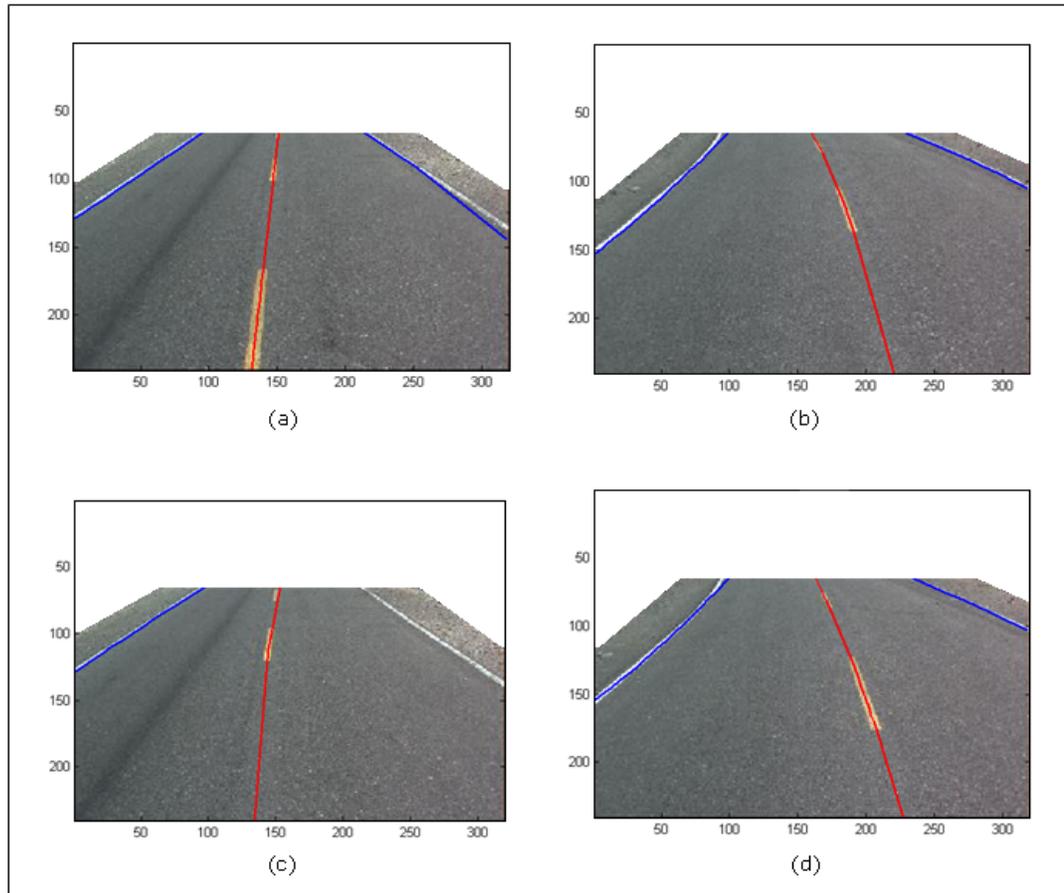


Fig. 4.18. Muestra de imágenes obtenidas en la evaluación del algoritmo. Estas cuatro imágenes fueron seleccionadas al azar.

Resumen:

Se obtuvo una eficiencia total de 93.05% en lo que respecta al algoritmo de detección de líneas. En cuanto al algoritmo de clasificación de las líneas encontradas, se obtuvo 96.57%.

Los principales errores que se encontraron en este grupo de imágenes fueron:

- La condición defectuosa de la pintura no permitía que la línea se vea en su total magnitud.

- La cercanía y el color de la tierra adyacente a la pista, terminaba formando parte de la línea a la cual deformaba y no permitía una detección eficaz.

Adicionalmente se debe mencionar que las líneas dibujadas son una aproximación a las líneas reales. Es por esto que muchas veces se ve una pequeña desviación (especialmente en la parte superior de la imagen) entre la línea real y la dibujada. Por lo que, sin necesidad de entrar en detalles, se considerará un estimado general de 99% de eficiencia obtenido en este aspecto.

Entonces los resultados finales para este grupo de imágenes es de una eficiencia de:

Detección de líneas	:	93.05%
Clasificación de líneas	:	96.57%
Posición de líneas	:	99%

4.3.1.2. Imágenes 640x480

Tabla de resultados:

Imagen	Eficiencia		Problema
	Reconocimiento	Clasificación	
A.1	100	100	-
A.2	100	100	-
A.3	95	90	Pintura deteriorada
A.4	100	90	Pintura deteriorada
A.5	100	100	-
A.6	100	95	Pintura deteriorada
A.7	95	100	Pintura deteriorada
A.8	100	100	-
A.9	80	90	Desprendimiento de tierra
A.10	100	100	-
A.11	95	100	Clara tonalidad de la tierra
A.12	100	100	-
A.13	100	100	-
A.14	90	90	Pintura deteriorada
A.15	100	100	-
A.16	75	95	Reducido tamaño de la línea
A.17	100	100	-

A.18	100	100	-
A.19	100	80	Problema debido a sombras
A.20	70	95	Clara tonalidad de la tierra
A.21	95	95	Pintura deteriorada
A.22	100	95	Pintura deteriorada y reducido tamaño de la línea
A.23	100	100	-
A.24	90	90	Pintura deteriorada
A.25	100	100	-
A.26	65	95	Reducido tamaño de la línea
A.27	90	95	Reducido tamaño de la línea
A.28	100	100	-
A.29	100	100	-
A.30	100	100	-
A.31	70	95	Deformación de la línea debido a la clara tonalidad de la tierra
A.32	95	90	Pintura deteriorada
A.33	100	100	-
A.34	100	100	-
A.35	100	100	-

Resumen:

Se obtuvo una eficiencia total de 94.43% en lo que respecta al algoritmo de detección de líneas. En cuanto al algoritmo de clasificación de las líneas encontradas, se obtuvo 96.57%.

Al igual que usando imágenes de 320x240 los principales errores de detección se debió a las condiciones de la pista. Sin embargo, la incidencia disminuyó y se obtuvieron mejores resultados. Los valores de eficiencia en ambos casos fueron mejores que los anteriores. Entonces los resultados finales para este grupo de imágenes es de una eficiencia de:

Detección de líneas	:	94.43%
Clasificación de líneas	:	96.57%
Posición de líneas	:	99%

4.3.2. Grupo B

4.3.2.1. Imágenes 320x240

Tabla de resultados:

Imagen	Eficiencia [%]		Problema
	Reconocimiento	Clasificación	
B.1	80	100	Pintura deteriorada
B.2	95	80	Pintura deteriorada
B.3	100	75	Pintura deteriorada
B.4	100	100	-
B.5	100	100	-
B.6	80	100	Pintura deteriorada
B.7	90	100	Pintura deteriorada
B.8	95	100	Pintura deteriorada
B.9	100	100	-
B.10	90	100	Reducido tamaño de la línea
B.11	100	80	Pintura deteriorada
B.12	85	80	Pintura deteriorada
B.13	95	75	Pintura deteriorada
B.14	75	100	Deformación de la línea debido a la clara tonalidad de la pista
B.15	95	100	Pintura deteriorada

Imágenes obtenidas:

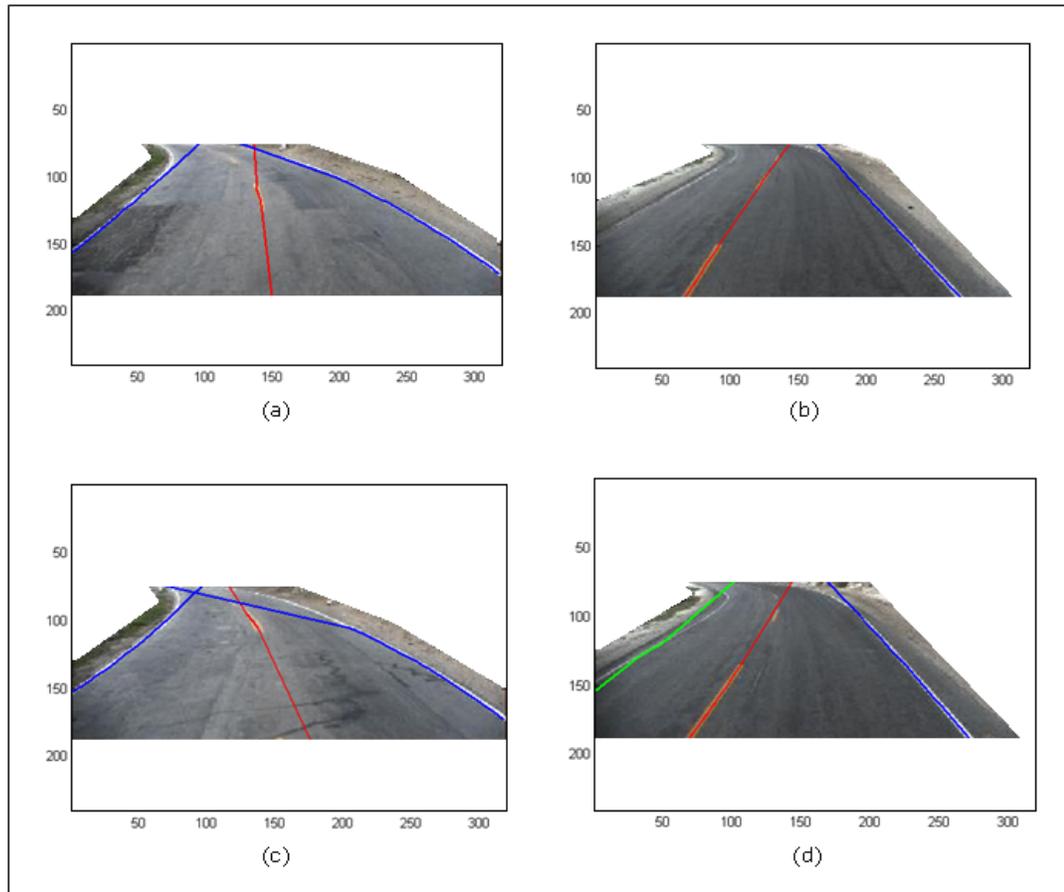


Fig. 4.19. Muestra de imágenes obtenidas en la evaluación del algoritmo. Estas cuatro imágenes fueron seleccionadas al azar.

Resumen:

Se obtuvo una eficiencia total de 92.04% en lo que respecta al algoritmo de detección de líneas y una eficiencia de 92.67% para la parte de clasificación de líneas.

Los principales errores que se encontraron en este grupo de imágenes fueron:

- La mala condición de la línea no sólo impedía observar la línea en su total magnitud, sino que inclusive no se podía distinguir.
- Las manchas oscuras o claras adyacentes a las líneas, dificultan el contraste entre la pista y la misma línea. Dando como resultado en

casos extremos, líneas con terminaciones amorfas las cuales el sistema no las considera.

- La amplitud del paisaje dentro de la imagen original, ocasionaba que muchos detalles se introduzcan en el proceso como sucede con la tierra adyacente. En la parte superior de la imagen, al enfocar el fondo del paisaje, no se puede apreciar con claridad la distancia entre la pista y la tierra, y al ser ésta de tonalidad clara muchas veces se confunde como una parte de la línea.

Tal como sucedió en el grupo de imágenes A, el resultado con las líneas dibujadas es sólo una aproximación de la imagen real. Es por esto que en algunas de ellas se presenta una cierta desviación con la línea original. Sin embargo en este caso la desviación de las líneas dibujadas es más significativa, entonces se ha estimado una eficiencia de 92%.

Por lo que los resultados finales para este grupo de imágenes es de una eficiencia de:

Detección de líneas	:	92.04%
Clasificación de líneas	:	92.67%
Posición de líneas	:	92%

Se puede apreciar que hay una pequeña diferencia con los resultados obtenidos en el Grupo A. Esto se debe a que el algoritmo no fue diseñado para estas imágenes, se cambiaron algunos parámetros y se logró adaptar sin embargo es importante que se tenga en cuenta que un mismo código para ambos grupos de imágenes no funcionaría correctamente debido a la diversidad de la imágenes, considerando especialmente los formatos de captura.

4.3.2.2. Imágenes 640x480

Imagen	Eficiencia [%]		Problema
	Reconocimiento	Clasificación	
B.1	100	95	Pintura deteriorada
B.2	100	95	Pintura deteriorada

B.3	100	100	-
B.4	80	100	Pintura deteriorada
B.5	100	100	-
B.6	100	80	Pintura deteriorada
B.7	100	100	-
B.8	100	100	-
B.9	100	100	-
B.10	70	80	Reducido tamaño de la línea
B.11	90	85	Pintura deteriorada
B.12	75	85	Pintura deteriorada
B.13	100	100	-
B.14	70	85	Deformación de la línea debido a la clara tonalidad de la pista
B.15	100	100	-

Resumen:

Se obtuvo una eficiencia total de 92.33% en lo que respecta al algoritmo de detección de líneas y una eficiencia de 93.67% para la parte de clasificación de líneas.

Tal como sucedió en el grupo de imágenes A, los principales errores se debieron a la condición de la pista. Sin embargo, al utilizarse imágenes con mayor resolución se obtuvo mejores resultados en la eficiencia tanto para la detección de las líneas como para clasificación de éstas. Adicionalmente se obtuvo también una pequeña mejora en la posición de líneas resultantes.

Por lo que los resultados finales para este grupo de imágenes es de una eficiencia de:

Detección de líneas	:	92.33%
Clasificación de líneas	:	93.67%
Posición de líneas	:	93%

4.3.3. Grupo C

4.3.3.1. Imágenes 320x240

Tabla de resultados:

Imagen	Eficiencia [%]		Problema
	Reconocimiento	Clasificación	
C.1	100	100	-
C.2	80	70	Color y forma del vehículo perjudica el procesamiento
C.3	100	100	-
C.4	95	100	Reducido tamaño de la línea
C.5	100	100	-
C.6	100	100	-
C.7	100	100	-
C.8	100	100	-
C.9	50	100	Error en el algoritmo
C.10	80	100	Clara tonalidad de la tierra y pintura deteriorada
C.11	95	100	Reducido tamaño de la línea
C.12	100	100	-
C.13	100	100	-
C.14	100	100	-
C.15	100	100	-
C.16	100	100	-
C.17	100	100	-
C.18	50	100	Error en el algoritmo

Imágenes obtenidas:

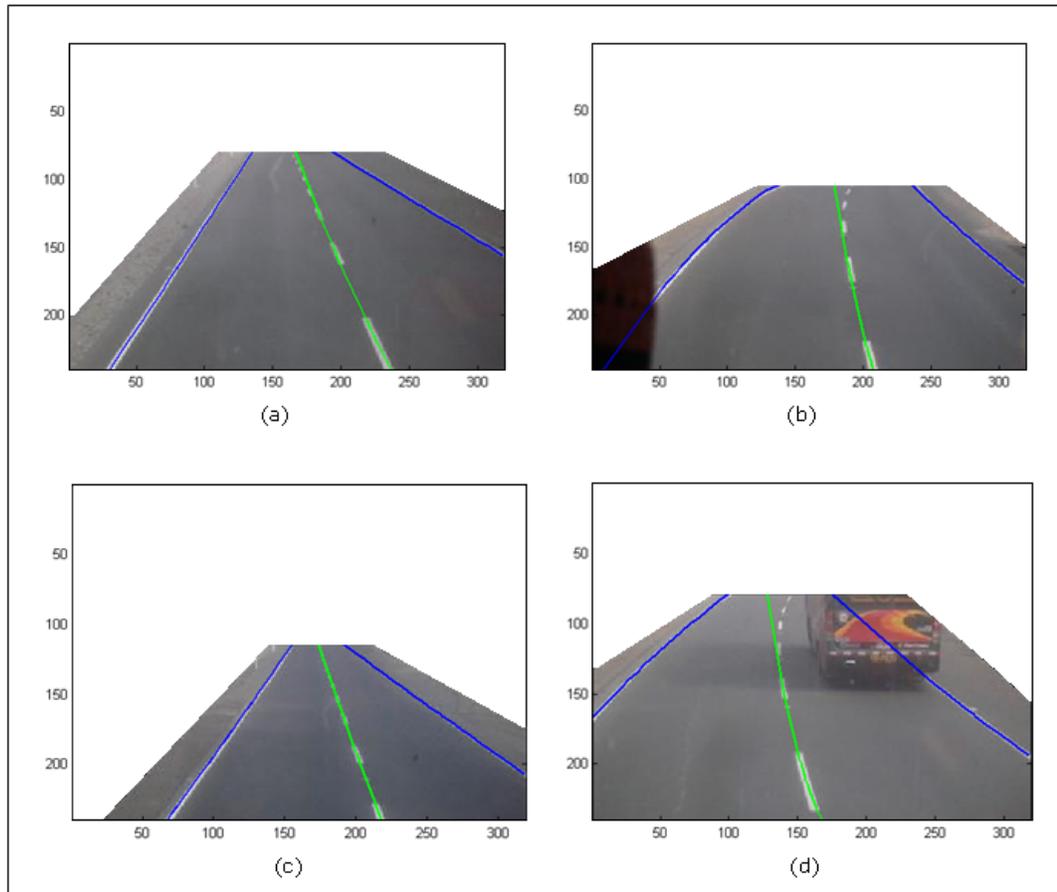


Fig. 4.20. Muestra de imágenes obtenidas en la evaluación del algoritmo. Estas cuatro imágenes fueron seleccionadas al azar.

Resumen:

Se obtuvo una eficiencia total de 91.67% en lo que respecta al algoritmo de detección de líneas y una eficiencia de 98.33% para la parte de clasificación de líneas.

Los principales errores que se encontraron en este grupo de imágenes fueron:

- La presencia de factores externos, como por ejemplo las unidades de transporte. Algunas veces los colores y formas de la carrocería añaden error al proceso.

- La manera cómo se capturaron estas imágenes. El vidrio del parabrisas del vehículo causa que la imagen no sea nítida y si a esto se le añade la claridad de la tierra adyacente, los resultados no son muy buenos.

Es necesario mencionar que a pesar que el algoritmo no fue diseñado para estas imágenes, ya que no cumplen con el formato de captura, se hicieron las modificaciones para la adaptación y el algoritmo funcionó ignorando posibles problemas como los símbolos en la pista: flechas y líneas de incorporación de carril.

Por otro lado, en este grupo de imágenes se observaron errores que ya se sabía que existirían debido a las restricciones del algoritmo. Por ejemplo, como parte de la lógica del algoritmo se tomaron restricciones para ignorar líneas que sean verticales (o casi verticales) y no estén en el centro de la imagen a lo largo del eje X (sección 4.2.3). Para aclarar la razón de esta restricción es que se parte de la hipótesis que el vehículo está alineado en su carril, por lo tanto la visión frontal será la pista con las líneas (continuas o discontinuas) inclinadas hacia el centro de la imagen. Entonces en las imágenes en las cuales se presentó este error, fue debido a que las líneas que sí habían sido detectadas fueron discriminadas por cumplir con la premisa antes descrita.

Tal como sucedió en los grupos de imágenes A y B, el resultado con las líneas dibujadas es sólo una aproximación de la imagen real. Entonces a pesar de ser esta desviación muchas veces imperceptible, se ha considerado una eficiencia de 95% en este aspecto.

Por lo que los resultados finales para este grupo de imágenes es de una eficiencia de:

Detección de líneas	:	91.67%
Clasificación de líneas	:	98.33%
Posición de líneas	:	95%

Al igual que en el Grupo B, la eficiencia es más baja que la obtenida en el Grupo A. Y como ya se ha mencionado antes, esto responde a un problema de diseño. Sin embargo aún considerando estas limitaciones, el sistema se encuentra en un rango de eficiencia aceptable.

4.3.3.2. Imágenes 640x480

Imagen	Eficiencia [%]		Problema
	Reconocimiento	Clasificación	
C.1	100	100	-
C.2	100	100	-
C.3	100	100	-
C.4	75	75	Reducido tamaño de la línea
C.5	95	100	Pintura deteriorada
C.6	75	100	Error en el algoritmo
C.7	100	100	-
C.8	100	100	-
C.9	100	95	-
C.10	90	90	Clara tonalidad de la tierra y pintura deteriorada
C.11	75	100	Reducido tamaño de la línea
C.12	100	100	-
C.13	100	100	-
C.14	100	90	-
C.15	90	90	Error en el algoritmo
C.16	95	100	Problema debido a sombras
C.17	100	100	-
C.18	50	90	Error en el algoritmo

Resumen:

Se obtuvo una eficiencia total de 91.67% en lo que respecta al algoritmo de detección de líneas y una eficiencia de 98.33% para la parte de clasificación de líneas.

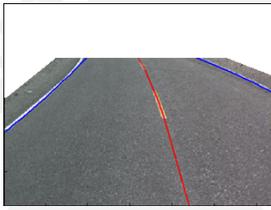
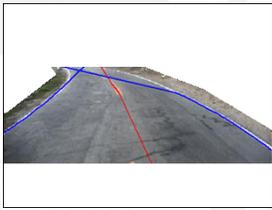
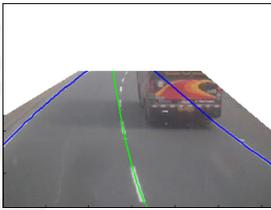
En este caso los errores se dieron por múltiples causas. Los factores externos como la presencia de unidades de transporte, el formato de captura, fueron las principales causas de errores. Sin embargo hubo problemas en el algoritmo, tanto predecibles como errores no forzados. Se tendría que analizar este grupo de imágenes desde el inicio del diseño

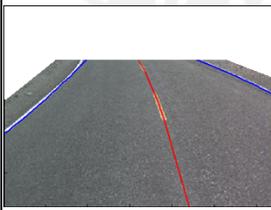
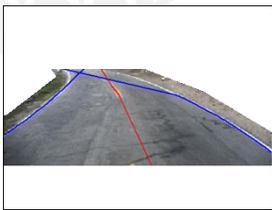
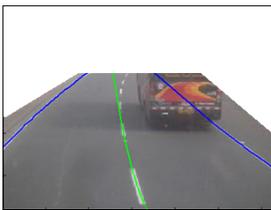
para obtener mejores resultados, algo que no sería práctico ya que el formato de captura de estas imágenes no llega a ser homogéneo.

Por lo que los resultados finales para este grupo de imágenes es de una eficiencia de:

Detección de líneas : 91.39%
 Clasificación de líneas : 96.11%
 Posición de líneas : 95%

4.3.4. Resumen Final

320x240	Grupo A	Grupo B	Grupo C
Eficiencia [%]			
Detección	93.05	92.04	91.67
Clasificación	96.57	92.67	98.33
Posición	99	92	95

640x480	Grupo A	Grupo B	Grupo C
Eficiencia [%]			
Detección	94.43	92.33	91.39
Clasificación	96.57	93.67	96.11
Posición	99	93	95

CONCLUSIONES

- Según lo planteado como hipótesis y objetivo principal (el reconocimiento de las líneas mediante procesamiento de imágenes), se puede certificar que realmente se pudieron obtener los resultados esperados. Tanto el uso de imágenes para reconocer las líneas de la carretera como el desarrollo de un algoritmo capaz de procesar estas imágenes para el análisis y reconocimiento, han sido demostrados en el presente trabajo de manera satisfactoria.
- Se utilizó un algoritmo que permitió seleccionar sólo las líneas de la carretera entre los demás objetos. Si bien la base del desarrollo se realizó asentado en dos imágenes, fue necesario realizar distintas pruebas del algoritmo mediante diversas imágenes para saber su eficiencia en distintos contextos, llegando a la conclusión que usar el mismo algoritmo para todas las imágenes no era lo más recomendable debido a la variedad de formatos de captura de éstas. Para esto, se agruparon las imágenes más parecidas y se adaptó un código para cada grupo, cambiando algunos parámetros de discriminación como longitud de línea, ecuación de perspectiva, ancho de línea, etc.
- Como parte de la evaluación final, es importante mencionar que el algoritmo tiene una eficiencia promedio de 96.24% para el Grupo A. Para el Grupo B y Grupo C es de 92.24% y 95% respectivamente. Estos resultados se han calculado promediando las tres eficiencias mostradas en los puntos anteriores. Aunque la diferencia no es muy pronunciada, se nota que el sistema ha funcionado mejor con las imágenes del Grupo A. Esto se debe (como ya se ha mencionado anteriormente) a una cuestión de diseño del algoritmo.
- Adicionalmente se realizó una evaluación tomando imágenes con mayor resolución. Como se esperaba los resultados mejoraron en la mayoría de casos, no en todos. En algunos casos lo obtenido no fue

tan alto como se esperaba. Esto fue debido a un problema del algoritmo, ya que estaba diseñado para otra tipo de imágenes entonces al usar una imagen con mayor detalle se tendrían que usar otros métodos de procesamiento que aprovechen mejor estos detalles.

- Es necesario mencionar que no se utilizaron estas imágenes con mayor resolución (640x480 px.) desde el inicio del diseño ya que se dio prioridad a un menor tiempo de procesamiento. En las evaluaciones finales, como se esperaba, al utilizar el sistema con estas nuevas imágenes el tiempo de procesamiento aumentaría considerablemente. Utilizando una PC con procesador Intel Celeron D 2.66GHz. la simulación con imágenes 320x240 llegaba a un tiempo de 10 seg. por imagen, sin embargo utilizando las nuevas imágenes de 640x480 se llegó a un tiempo de simulación de 38 seg. por imagen. Claramente más largo.



RECOMENDACIONES

- Una de las principales consideraciones es la eliminación de las regiones que no son de interés para el análisis. Para efectos de las pruebas estas regiones se eliminaron manualmente. Se requiere un proceso previo que realice la función mencionada para complementar y permitir que el sistema se desarrolle automáticamente.
- Con el objetivo de obtener un programa “universal” que funcione eficientemente en cualquier situación, se debe estandarizar el proceso de captura de las imágenes, fijando los objetivos del sistema en lo que respecta al área de detección requerida. Como se apreció en el desarrollo de este trabajo, reconocer líneas que estén a una distancia considerablemente larga complica innecesariamente el proceso y no tiene sentido práctico.
- Dentro de la parte técnica sería recomendable mejorar la parte de acondicionamiento de la imagen y umbralización, añadiendo técnicas de análisis de imágenes más avanzadas de manera que estos procesos sean más dinámicos. Tal como se realizó en la parte de clasificación de líneas, la idea sería ampliar el tema de la perspectiva para estos procesos.
- En el caso que la prioridad sea la mayor efectividad del sistema sin tener problemas con el tema del costo, se podrían utilizar las imágenes con mayor resolución tal que el hardware no tenga problemas con el tiempo de procesamiento. Entonces a partir de allí seleccionar la resolución de las imágenes a utilizar, luego realizar las mejoras convenientes en el código para aprovechar al máximo los detalles y tener los mejores resultados.

FUENTES

- [1] M. Cataño, 'Reconocimiento de las líneas de carretera', *Electro Electrónica*, 2006, vol. 26, pp. 23-28.
- [2] K.R.S. Kodagoda, Chieh-Chih Wang, G. Dissanayake, "Laser Based Sensing on Roads", *Proc. of the Intelligent Vehicles & Road Infrastructure Conference*, Febrero 2005.
- [3] L. Fletcher, L. Petersson, A. Zelinsky, 'Driver Assistance Systems based on Vision In an Out of Vehicles', *Proc. IEEE Symposium on Intelligent Vehicles*, Junio 2003, pp. 322-327.
- [4] A. Broggi, S. Bertè, 'Vision-Based Road Detection in Automotive Systems: A Real-Time Expectation-Driven Approach', *Journal of Artificial Intelligence Research*, 1995, vol. 3, pp. 325-348.
- [5] P. Sharke, 'Smarts Cars', *Mechanical Engineering*, Marzo 2003.
- [6] S. Lee, K. Boo, J. Lee, D. Shin, W. Kwon, K. Roh, D. Kim, 'Vision-Based Vehicle Localization for Automatic Lane Keeping System', *5th Congress on Intelligent Transportation System*, 1998.
- [7] A. Dankers, L. Fletcher, L. Peterson, A. Zelinsky, 'Driver Assistance: Contemporary Road Safety', *Proceedings of the Australasian Conference on Robotics and Automation*, Diciembre 2003.
- [8] T. Schön, A. Eidehall, F. Gustafsson, 'Lane Departure Detection for Improved Road Geometry Estimation', *Intelligent Vehicles Symposium*, 2006, pp. 546-551.

- [9] J. Seo, H. Chul, S. Jeong, Y. Kong, 'Development of Lane Deviation Warning & Preventing System through Vision System and Steering Control', *Seoul ITS Congress*, 1998.
- [10] J. Collado, C. Hilario, A. De la Escalera, J. Armingol, 'Detección y Clasificación de Líneas Viales mediante Análisis Frecuencial', *XXVI Jornadas de Automática*, Setiembre 2005.
- [11] Transportes Cruz del Sur, 'Cruz del Sur, Servicio Cruzero',
<http://www.cruzdelsur.com.pe/cruzero.htm>.
- [12] EMG-PNP/OFITEL Y OGPP - Oficina de Estadística, 'Número de accidentes de tránsito fatales y no fatales por año, según causa 1999-2007'. Ministerio de Transporte y Comunicaciones del Perú.
<http://www.mtc.gob.pe/estadisticas/archivos/xls/3.G.2.xls>.
- [13] A. Polychronopoulos, N. Möhler, S. Ghosh, A. Beutner, 'System design of a situation adaptive lane keeping support system, the SAFELANE system'. *Advanced Microsystems for Automotive Applications*, 2005, pp. 169-183.
- [14] Ministerio de Transportes y Comunicaciones del Perú. 'Marcas en el pavimento: Marcas en el pavimento y bordes de pavimento'.
http://www.mtc.gob.pe/portal/transportes/caminos_ferro/manual/Transito/cap3/seccion32.htm, 2000.
- [15] R. Gonzalez, R. Woods, S. Eddins, 'Digital Image Processing using MATLAB', 2004, pp. 337-404.
- [16] P.S. Liao, T.S. Chen, P.C. Chung, "A Fast Algorithm for Multilevel Thresholding", *Journal of Information science and Engineering*, N° 17, 2001, pp. 713-727.

- [17] S. Belkasim, A. Ghazal, O. Basir, "Edge Enhanced Optimum Automatic Thresholding", *Proceedings of the 2000 international Computer Symposium*, Diciembre 2000, pp. 78-86.
- [18] P.L. Rosin, "Unimodal Thresholding", *Pattern Recognition*, Noviembre 2001, vol. 34, pp. 2083-2096.



En el disco adjunto se almacenan tres carpetas: GRUPO A, GRUPO B Y GRUPO C. Cada una de estas carpetas contiene un archivo de Word y un archivo de Matlab.

En el documento (archivo de Word) se muestran las pruebas con los resultados finales obtenidos para cada una de las imágenes del grupo. Todas las imágenes de prueba fueron manipuladas en Paint para eliminar las regiones que no son de interés.

El archivo de Matlab corresponde al código de Matlab que se usó para el diseño del algoritmo, para cada grupo de imágenes.

Además el disco adjunto contiene una carpeta llamada HERRAMIENTAS. Esta carpeta incluye los distintos archivos de Matlab que se utilizaron como Subrutinas o Funciones para facilitar el proceso.

