



UNIVERSIDAD NACIONAL DEL LITORAL
Facultad de Ingeniería y Ciencias Hídricas

PROYECTO FINAL DE CARRERA
INGENIERÍA INFORMÁTICA

MEDICIÓN DE INTENSIDAD DE TRÁFICO MEDIANTE ANÁLISIS DE VIDEO

Autor: Juan Pablo Ceschi

Director: Dr. Matínez, César

Co-Director: Dr. Albornoz, Enrique M.

Santa Fe - 2016

Agradecimientos

Este proyecto no podría haberse realizado sin el apoyo de un gran número de personas a las que quiero agradecer:

Dr. César Martínez y Dr. Enrique M. Albornoz, quienes brindaron de su tiempo y conocimiento para guiarme a lo largo de todo el proyecto, y tuvieron una excelente predisposición para asistir a reuniones y responder consultas.

Ing. Gastón Martín, por ayudarme con la elección del tema y dedicar tiempo en organizar y asistir a reuniones en la Municipalidad.

Ing. Juan Savoy, quien dedicó parte de su tiempo en escuchar mi propuesta de proyecto y se comprometió en proporcionar un conjunto de videos de tráfico.

A todos los profesores y ayudantes de cátedra, que fueron parte de mi formación a lo largo de estos años.

A todos mis compañeros, que me brindaron su amistad y compartieron experiencias.

A toda mi familia, mis amigos y a mi novia, por acompañarme y bancarme durante esta etapa de mi vida.

Resumen

El presente trabajo consiste en el desarrollo de un sistema automático para medición de intensidad de tráfico, basado en técnicas avanzadas de análisis y procesamiento de imágenes y videos. Para el diseño, implementación y evaluación del sistema se trabajó con una base de datos compuesta por videos de tráfico, los cuales fueron grabados con cámaras estándares que actualmente son utilizadas por la Municipalidad de Santa Fe para seguridad urbana. La medición de intensidad de tráfico consiste en localizar vehículos dentro de la escena del video, clasificarlos según su porte, y utilizar un indicador de tiempo para determinar la intensidad de tráfico.

El sistema desarrollado permite procesar largos períodos de registro en video, analizando y extrayendo información de manera automática. Todo el procesamiento es realizado en tiempo real, trabajando sobre videos que poseen una tasa de 10 fotogramas por segundo. La información producida será de gran utilidad para mejorar las políticas aplicadas a la circulación y para el desarrollo de nuevas herramientas que ayuden a solucionar problemas de congestión vehicular.

Como complemento se desarrolló una interfaz gráfica de usuario, que facilita la interacción del usuario con el sistema y permite la visualización en tiempo real del procesamiento realizado.

Prefacio

En este trabajo se presenta el desarrollo de un sistema para medición de intensidad de tráfico. El mismo no propone el desarrollo de un software orientado a un usuario final, sino el desarrollo de un prototipo funcional que contempla el sistema base de procesamiento más una interfaz gráfica de usuario. El documento se encuentra organizado mediante capítulos de la siguiente forma:

En el capítulo 1 se realiza una breve introducción a la problemática que se busca resolver. Se desarrolla el estado del arte en medición de tráfico y detección, seguimiento y clasificación de objetos mediante procesamiento digital de imágenes y videos. Finalmente se mencionan los objetivos y alcances del Proyecto Final.

En el capítulo 2 se desarrollan conceptos teóricos que fueron utilizados en el proyecto.

En el capítulo 3 se expone el análisis realizado sobre la base de datos para la determinación de parámetros característicos. Se describen casos de prueba y se listan los requerimientos funcionales y no funcionales del sistema. Se presenta el diseño por módulos del sistema, describiendo brevemente las funcionalidades de cada componente. Finalmente se detalla el diagrama de clases diseñado y se mencionan cambios realizados sobre la base de datos.

En el capítulo 4 se detalla la implementación del sistema y se explica el funcionamiento de los componentes de la interfaz gráfica de usuario.

En el capítulo 5 se presentan los casos de prueba diseñados y los resultados obtenidos luego de evaluar el desempeño del sistema.

En el capítulo 6 se desarrollan conclusiones y desarrollos futuros.

Índice general

Resumen	III
Prefacio	IV
Índice de figuras	VII
Índice de tablas	IX
Introducción	10
1.1 Motivación.....	10
1.2 Estado del arte.....	11
1.3 Objetivos del proyecto final.....	13
1.3.1 Objetivo general.....	13
1.3.2 Objetivos específicos.....	14
1.4 Alcances del proyecto final.....	14
Fundamentos teóricos	15
2.1 Representación.....	15
2.2 Modelo de color.....	15
2.2.1 Modelo RGB.....	16
2.2.2 Modelo CIELab.....	16
2.3 Operaciones morfológicas.....	19
2.4 Segmentación.....	20
2.4.1 Umbralización.....	20
2.4.2 Componentes conexas.....	21
2.5 Filtro de mediana.....	22
2.6 Mezcla de gaussianas.....	22
2.7 Ecuación adaptativa del histograma.....	25
Análisis y Diseño del Sistema	30
3.1 Análisis de la base de datos.....	30
3.2 Especificación de requerimientos.....	33
3.3 Casos de uso.....	34
3.4 Diseño del sistema.....	37
3.5 Diagrama de clases.....	40
3.6 Modificación de la base de datos.....	43
Implementación	45
4.1 Tecnologías utilizadas.....	45
4.2 Metodología.....	46
4.2.1 Módulo de pre-procesamiento.....	47

4.2.2 Módulo de detección.....	50
4.2.3 Módulo de seguimiento.....	56
4.2.4 Módulo de conteo.....	60
4.3 Interfaz de usuario.....	61
4.3.1 Consideraciones en el diseño.....	62
4.3.2 Componentes de la interfaz.....	62
Experimentación y resultados	66
5.1 Configuración de parámetros.....	66
5.2 Casos de prueba.....	67
5.3 Análisis de resultados.....	70
5.3.1 Efectividad en la detección.....	71
5.3.2 Velocidad de ejecución.....	73
Conclusiones y desarrollos futuros	75
Bibliografía	77

Índice de figuras

Figura 2.1: Representación de imagen digital por medio de una matriz de $N \times M$ píxeles.....	16
Figura 2.2: Modelo de color RGB.....	17
Figura 2.3: Modelo de color CIE Lab.....	17
Figura 2.4: Ejemplo de erosión.....	19
Figura 2.5: Ejemplo de dilatación.....	20
Figura 2.6: Ejemplos de umbralizado.....	21
Figura 2.7: Ejemplo de detección y etiquetado de regiones.....	22
Figura 2.8: Gráfica de una mezcla de 4 gaussianas con pesos iguales.....	23
Figura 2.9: Ejemplo de histograma.....	26
Figura 2.10: Ejemplo de ecualización del histograma.....	28
Figura 2.11: Imagen dividida en bloques.....	28
Figura 2.12: Ejemplo de interpolación en la ecualización del histograma adaptativo.....	28
Figura 2.13: Ejemplo de ecualización adaptativa del histograma con contraste limitado.....	29
Figura 3.1: Ejemplos de obstáculos dentro de la escena.....	31
Figura 3.2: Ejemplos de escenas con diferentes niveles de iluminación.....	32
Figura 3.3: Ejemplos de sombras que generan falsas detecciones.....	32
Figura 3.4: Ejemplos de oclusiones parciales entre vehículos.....	33
Figura 3.5: Diagrama de casos de uso.....	35
Figura 3.6: Diagrama general del sistema.....	38
Figura 3.7: Diseño detallado del sistema.....	40
Figura 3.8: Diagrama de clases.....	41
Figura 3.9: Capturas de videos descartados por posición inadecuada de la cámara.....	43
Figura 3.10: Ejemplos de videos descartados por grabarse durante la noche.....	44
Figura 4.1: Video representado como secuencia de fotogramas.....	46
Figura 4.2: Ejemplo de definición de una ROI.(a): fotograma original, (b): máscara binaria definida por el usuario, (c): máscara binaria aplicada sobre el fotograma.....	49
Figura 4.3: Ejemplos de normalización de iluminación desfavorable.....	50
Figura 4.4: Ejemplo de modelado del fondo. (a): fotograma, (b): máscara de segmentación.....	51
Figura 4.5: Ejemplo de segmentación.(a):fotograma, (b):máscara de segmentación.....	52

Figura 4.6: Ejemplo de presencia de sombras en la escena del video.(a):fotograma, (b): máscara de segmentación.....	53
Figura 4.7: Captura del sistema para eliminación de sombras.....	54
Figura 4.8: Ejemplo de eliminación de sombras. (a):máscara de segmentación, (b)máscara con sombras detectadas:, (c):máscara resultante de la diferencia.....	54
Figura 4.9: Ejemplo de correspondencia entre los arreglos.....	57
Figura 4.10: Ejemplo de posición inadecuada de la cámara.....	58
Figura 4.11: Ejemplo del primer caso de oclusión.....	58
Figura 4.12: Ejemplo del segundo caso de oclusión.....	59
Figura 4.13: Ejemplo de resolución de oclusión.....	60
Figura 4.14: Interfaz gráfica de usuario.....	63
Figura 5.1: Capturas de los videos utilizados para cada caso de prueba.....	67

Índice de tablas

Tabla 4.1: Parámetros iniciales del sistema.....	47
Tabla 5.1: Valores de los parámetros del sistema para cada caso de prueba.....	70
Tabla 5.2: Resultados correspondientes a los fragmentos del primer caso de prueba.....	71
Tabla 5.3: Resultados correspondientes a los fragmentos del segundo caso de prueba.....	72
Tabla 5.4: Resultados correspondientes a los fragmentos del tercer caso de prueba.....	72
Tabla 5.5: Resultados correspondientes a los fragmentos del cuarto caso de prueba.....	73
Tabla 5.6: Tiempos de ejecución para instancia única del sistema.....	74
Tabla 5.7: Tiempos de ejecución para instancias múltiples del sistema.....	74

Introducción

En las últimas décadas se ha visto un aumento de la cantidad de vehículos motorizados en los países en vías de desarrollo. La creciente disponibilidad de automóviles ha permitido una mayor movilidad individual, la cual sumada al crecimiento poblacional de las ciudades y a la escasa aplicación de políticas estructuradas de transporte urbano, han potenciado el congestionamiento [1]. Citando a dos renombrados especialistas en el campo del modelado de sistemas de transporte: “surge la congestión en condiciones en que la demanda se acerca a la capacidad de la infraestructura transitada y el tiempo de tránsito aumenta a un valor muy superior al que rige en condiciones de baja demanda” [2].

Existe una serie de circunstancias específicas que causan o agravan la congestión, la mayoría de ellas reducen la capacidad de una carretera en un punto determinado o durante un determinado periodo. En las ciudades altamente pobladas, la congestión vehicular es recurrente y se atribuye a la gran demanda del tráfico; el resto de las causas se atribuyen a incidentes de tránsito, obras viales y eventos climáticos. La consecuencia más evidente del congestionamiento es el incremento de los tiempos de viaje, especialmente en las horas pico, que alcanza en algunas ciudades niveles bastante superiores a los considerados aceptables. Además, la lentitud de desplazamiento irrita los ánimos y fomenta el comportamiento agresivo de los conductores. Otros efectos negativos producidos por congestión son: la incapacidad de predecir con exactitud el tiempo de viaje, accidentes causados por automovilistas que pierden la calma al encontrarse estáticos durante mucho tiempo en un mismo lugar, el desperdicio de combustible que provoca un aumento de la contaminación en el aire, y el desgaste de los vehículos producido por la constante aceleración y frenado [3].

1.1 Motivación

En la actualidad, existen aplicaciones¹ que permiten a las personas lidiar a diario con la congestión vehicular, brindando información acerca de las condiciones actuales del tráfico en las calles [4,5]. Esta información es obtenida mediante una combinación de datos proporcionados por instituciones oficiales y por los propios usuarios que interactúan con la aplicación. Por lo tanto, es posible decir que la clave para el éxito de estas aplicaciones yace en la colaboración, el compromiso y el aporte de la comunidad. Otro aspecto importante a mencionar, es el hecho de que estas aplicaciones no realizan conteo de vehículos, y la medición de tráfico realizada es a fines ilustrativos, por lo que la información brindada no es tan precisa como para servir a los propósitos de planificación urbana, toma de decisiones en los municipios, entre otros.

¹ BA Móvil, Waze, Inrix Traffic, entre otros.

Por otra parte, dentro del campo de Visión Computacional el alcance de las técnicas de procesamiento de video permite la detección y clasificación de vehículos a partir de cámaras de video. Un sistema basado en estas técnicas será capaz de proporcionar información precisa, solo requiriendo datos proporcionados por cámaras de video que capturen el tráfico. En la mayoría de las ciudades medianas y grandes se utilizan cámaras de video con fines de seguridad urbana, las cuales se encuentran ubicadas sobre las principales calles y pueden ser empleadas para capturar el tráfico. De esta manera, se estaría reutilizando la infraestructura de cámaras existentes en cada ciudad, evitando así invertir grandes cantidades de dinero en la compra e instalación de cámaras y equipamiento especial.

Surge entonces, la idea de desarrollar un sistema capaz de brindar información precisa y en tiempo real, acerca de la intensidad de tráfico que hay en las calles sin depender del aporte de información por parte de un agente externo. Se propuso diseñar y desarrollar un sistema para medición de tráfico que utilice técnicas avanzadas de procesamiento de imagen y video, el cual trabaja con información brindada por cámaras de video que capturan las calles.

1.2 Estado del arte

Existe una gran variedad de mecanismos utilizados para la detección y el conteo de vehículos, cuyos funcionamientos se basan en la utilización de sensores, tanto a nivel de calzada como introducidos en el pavimento. Cada tecnología tiene un campo de aplicación en función de las características de la medición (temporal o permanente), del tipo de tránsito vehicular (flujo libre, a vuelta de rueda, arranque-parada), y de la precisión requerida [6].

Con respecto a los sensores en calzadas, las tecnologías más utilizadas son:

- *Manguera neumática*: detectan el paso del vehículo en base al impulso de presión que se genera. Pueden contar y clasificar, pero solo para utilización temporal y en tránsitos fluidos. Hay que cuidar la fijación de la manguera para evitar que los vehículos la arrastren o la rompan.
- *Lazos inductivos*: se pueden utilizar en forma temporal o permanente. Detectan el paso del vehículo por variación de la masa magnética sobre el lazo.
- *Sensores piezoeléctricos*: un par de sensores piezoeléctricos colocados a una distancia conocida permiten calcular la velocidad del vehículo y clasificarlo con precisión. Detectan el paso del vehículo en base a la carga eléctrica que se genera en el material piezoeléctrico cuando es pisado por una rueda.
- *Sensores en base a cables de fibra óptica*: son sensores que detectan el paso del vehículo por la variación de la conductividad óptica de un cable de fibra óptica pisado por la rueda del vehículo.

Para sistemas elevados sobre la calzada [7], se utilizan las siguientes tecnologías:

- *Sensor láser infrarrojo activo*: su principio de funcionamiento se basa en detectar la reflexión de un pulso de láser, identificando cuando la misma es producida por la calle o por un vehículo.
- *Radar a hiperfrecuencia*: este tipo de radar utiliza el efecto doppler para determinar la velocidad de un objeto en movimiento comparando la frecuencia de la onda emitida con la frecuencia de la onda reflejada.
- *Detección por video*: se basa en el procesamiento de imágenes en tiempo real, muy adecuada para la detección de vehículos en cruces señalizados donde es difícil o costoso instalar otro tipo de sensor.

Habitualmente, estos sistemas son de uso urbano, y permiten determinar magnitudes tales como número de vehículos, velocidades medias o de cada vehículo, clasificación en función de la longitud o altura del vehículo y determinar las dimensiones (alto, largo, ancho) de los vehículos en circulación, sin necesidad de ranurar las calzadas [7].

Particularmente dentro del campo de Visión Computacional (CV), la temática de seguimiento y clasificación de objetos, se encuentra bastante estudiada y existe una gran variedad de enfoques y métodos utilizados para este fin, dependiendo del caso particular que se esté intentando resolver. En general, los bloques de procesamiento utilizados sobre videos de tráfico son: pre-procesamiento, modelado de fondo y seguimiento de vehículos.

Pre-procesamiento:

El pre-procesamiento cumple la función de parametrizar el sistema y preparar el video de tráfico, para poder utilizarlo en las etapas posteriores de trabajo. Dentro de este bloque de trabajo, se encuentran métodos para modificar el formato de los datos, filtros para reducción de ruido generado por la cámara o por el ambiente, métodos para modificar niveles de iluminación de la escena, definición de regiones de interés, entre otros [8,9].

Modelado del fondo:

El modelado del fondo consiste en utilizar una determinada técnica para modelar el fondo del video, y luego restar ese modelo de fondo obtenido a cada fotograma del video para obtener los objetos que se encuentran en movimiento. Existen técnicas de modelado no recursivas, las cuales guardan en un buffer una cierta cantidad de fotogramas previos del video, y estiman la imagen de fondo basándose en la variación temporal de cada pixel de esos fotogramas. Algunas de estas técnicas son: diferencia de fotogramas, filtro de la mediana, filtro predictivo lineal, modelo no-paramétrico y extractor visual de fondo (ViBe) [8,10,11]. A su vez, existen técnicas de modelado recursivas que actualizan recursivamente un simple modelo de fondo basado en cada fotograma de entrada al sistema. Algunas de las técnicas son: filtro de la mediana aproximada, filtro de Kalman y mezcla de gaussianas (MoG) [8,9,12,13,14].

Seguimiento de vehículos:

La operación de seguimiento cumple la función de mantener actualizada la posición de los vehículos en un video. Para ello, primero debe detectarse el vehículo en la escena para determinar su posición inicial y luego se va actualizando esa posición hasta que el vehículo deja de ser visible o sale de la región de interés. Los diferentes enfoques utilizados para seguimiento se pueden clasificar de la siguiente manera:

- *Seguimiento basado en modelo:* se generan un modelo tridimensional del vehículo y se lo utiliza para seguir la trayectoria del mismo. La debilidad de este enfoque es la dependencia de los modelos geométricos, ya que no se puede tener modelos detallados para todos los vehículos que se podrían encontrar en el camino [9,12,15,16].
- *Seguimiento basado en región:* se identifica una región conectada en la imagen, un blob, asociado con cada vehículo y luego se siguen las pistas del mismo a lo largo del tiempo utilizando una medida de correlación cruzada. Este enfoque funciona bastante bien en el tráfico que fluye libremente, pero en condiciones de tráfico congestionado, algunos vehículos se ocluyen parcialmente entre sí, lo que hace difícil la tarea de segmentación de vehículos individuales [8,9,11,12,14,17,18].
- *Seguimiento basado en contorno activo:* se utiliza una representación del contorno del vehículo y se mantiene dinámicamente su actualización. La ventaja de tener una representación basada en el contorno es que se reduce la complejidad computacional, sin embargo, si dos o más vehículos entran en la región de detección parcialmente ocluidos, el sistema los representará como uno solo [9,12,16,19].
- *Seguimiento basado en características:* este enfoque abandona la idea de seguimiento de objetos en su conjunto y en su lugar, realiza un seguimiento de sub-características tales como puntos o líneas distinguibles en el vehículo. La ventaja de este enfoque es que incluso en presencia de una oclusión parcial, algunas de las características del vehículo en movimiento permanecen visibles y permiten que el mismo sea distinguido [9,12,20].

1.3 Objetivos del proyecto final

Se exponen a continuación los objetivos generales y específicos del Proyecto Final.

1.3.1 Objetivo general

El objetivo general de este Proyecto Final es desarrollar un sistema automático que mediante técnicas de análisis y procesamiento de imagen y video, sea capaz de medir en tiempo real intensidad de tráfico sobre videos.

1.3.2 Objetivos específicos

Los objetivos específicos de este Proyecto Final son:

- Diseñar e implementar un sistema de detección, seguimiento y conteo de vehículos en videos de tráfico, basado en técnicas para detección de movimiento estudiadas y adaptadas a los datos disponibles.
- Experimentar y analizar el desempeño del sistema mediante parámetros objetivos.
- Utilizar la información relevada por el sistema junto con un indicador de tiempo, para determinar la intensidad de tráfico.

1.4 Alcances del proyecto final

Se listan a continuación los alcances del presente Proyecto Final.

Alcances Funcionales:

- Detección, clasificación y conteo de vehículos sobre videos de tráfico.
- Estimación de la intensidad de tráfico en tiempo real.

Alcances No Funcionales:

- Desarrollo de un sistema robusto, que funcione para diferentes posiciones de la cámara y para diferentes momentos del día (excepto nocturnos).
- Desarrollo de un sistema efectivo, capaz de detectar e identificar correctamente los vehículos.
- Desarrollo de un sistema eficiente, que logre obtener resultados en tiempo real.
- El proyecto involucra el desarrollo de un prototipo para ser utilizado en una computadora con videos previamente grabados por cámaras de seguridad. No se pretende realizar una aplicación final específica y orientada a un usuario final.

Limitaciones:

- Se limitará el sistema a posiciones específicas de las cámaras de video.
- Se dejará como un trabajo futuro la optimización de velocidad y/o implementación por hardware (procesamiento con GPU).
- El sistema no trabajará con videos grabados sin iluminación solar.

Supuestos:

- La calidad de los videos será media (420p: 720 x 480) o alta (720p: 1280 x 720).
- Se contabilizarán los vehículos que puedan ser diferenciados a simple vista.

Fundamentos teóricos

El manejo de imágenes digitales se ha convertido en las últimas décadas en un tema de interés extendido dentro de las diferentes áreas de la ciencia. Podemos definir al *procesamiento digital de imágenes (PDI)* como el uso de técnicas y algoritmos de computadora para la manipulación de imágenes digitales. Dentro de las operaciones más comunes de procesamiento encontramos transformaciones geométricas (reducción, rotación, alargamiento), correcciones de color (ajustes de brillo y contraste), alineación de imágenes, segmentación, entre otros. El crecimiento en el poder de cómputo, la capacidad de almacenamiento y los nuevos sistemas de despliegado, captura e impresión de bajo costo, han facilitado el desarrollo de esta disciplina.

2.1 Representación

Una imagen en escala grises puede ser definida como una función bidimensional, $f(x,y)$, donde x e y son coordenadas espaciales, y la amplitud de la función f en cualquier par de coordenadas (x,y) es llamada intensidad o nivel de gris de la imagen en ese punto. Cuando tanto x , y , como la amplitud de f toman valores discretos, la imagen se denomina imagen digital, y puede ser representada como una matriz de $N \times M$ elementos llamados píxeles (Figura 2.1) [21]. Para imágenes digitales con más de un valor por píxel (como imágenes a color por ejemplo), se incorpora una nueva dimensión y se representa como una matriz tridimensional.

$$f(x,y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{pmatrix} \quad (2.1)$$

2.2 Modelo de color

Un modelo de color es la especificación de un sistema de coordenadas y un subespacio dentro del mismo, en el que cada color esta representado por un único punto. Gran parte de los modelos de color están orientados hacia el hardware o hacia aplicaciones donde la manipulación del color es el objetivo.

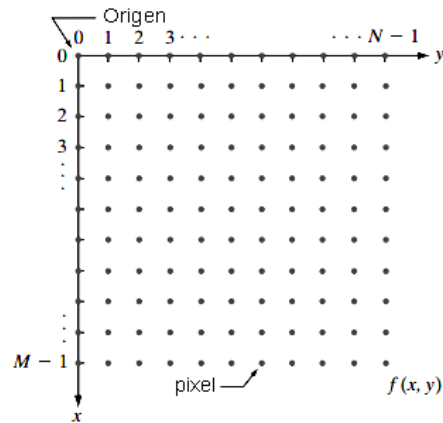


Figura 2.1: Representación de imagen digital por medio de una matriz de $N \times M$ píxeles.

Los modelos orientados a hardware más comúnmente utilizados en la práctica son: el modelo RGB (Red, Green, Blue) en monitores y cámaras digitales; los modelos CMY (Cyan, Magenta, Yellow) o CMYK (CYM + black) para la impresión a color. Modelos como HSV (Hue, Saturation, Value) o HSI (Hue, Saturation, Intensity), son muy utilizados para la manipulación del color [21].

2.2.1 Modelo RGB

Este modelo está basado en un sistema de coordenadas cartesianas, donde el subespacio en el cual se define el color consiste en un cubo normalizado entre $[0,1]$ (Fig. 2.2). El valor rojo (R) se encuentra sobre el vértice $(1,0,0)$, el valor verde (G) se encuentra sobre el vértice $(0,1,0)$, y el valor azul (B) se encuentra sobre el vértice $(0,0,1)$. El origen corresponde al color negro $(0,0,0)$, el punto más lejano a éste último se corresponde con el blanco $(1,1,1)$, y el resto de los vértices corresponden a los colores secundarios cian, magenta y amarillo. Los diferentes colores quedan representados en este modelo como puntos dentro del cubo.

El modelo RGB se basa en la síntesis aditiva, con lo que es posible representar un color como la combinación por adición de los tres colores primarios. Las imágenes digitales están formadas por tres matrices, una por cada color primario, y es frecuente que cada uno de los mismos se codifique con un byte (8 bits), por lo tanto, el rango de valores de intensidad que pueden tomar va de 0 a 255.

2.2.2 Modelo CIELab

El propósito de este modelo es producir un espacio de color que sea más *perceptivamente lineal* (un cambio en un valor de color debe producir un cambio aproximado de importancia visual) que otros espacios de color. Es utilizado normalmente para describir todos los colores que puede percibir el ojo humano.

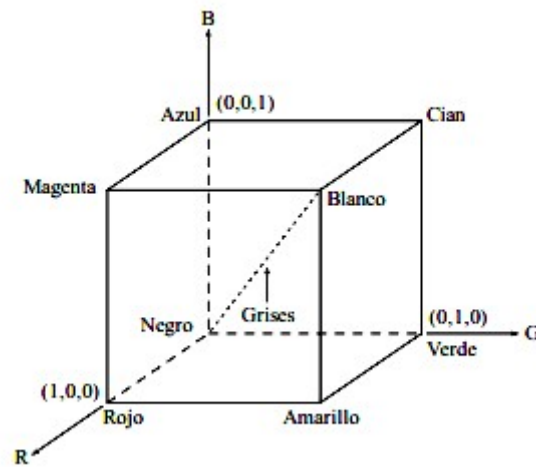


Figura 2.2: Modelo de color RGB

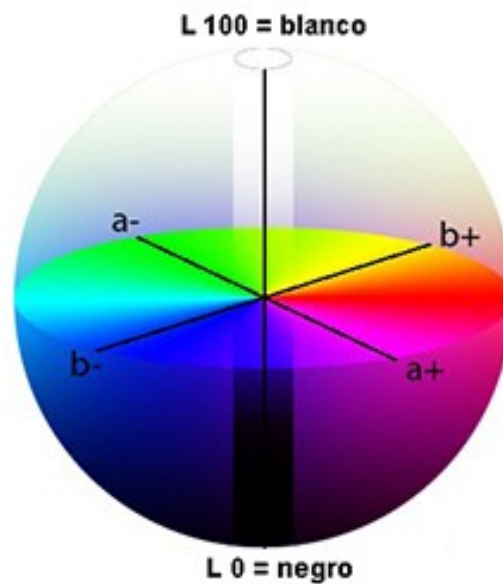


Figura 2.3: Modelo de color CIELab

El modelo CIELab consiste en un sistema coordenado cartesiano definido por tres coordenadas colorimétricas: L^* , a^* y b^* (Fig. 2.3). Los tres parámetros en el modelo representan la luminosidad de color ($L^*=0$ indica negro y $L^*=100$ indica blanco), su posición entre magenta y verde (valores negativos de a^* indican verde, mientras valores positivos indican magenta), y su posición entre amarillo y azul (valores negativos de b^* indican azul, y valores positivos indican amarillo).

Estos parámetros (L^* , a^* y b^*) están relacionados con el *punto-blanco* (punto que se toma como blanco de referencia: X_n, Y_n, Z_n) de los datos XYZ desde donde fueron convertidos. En la práctica, muchas veces se asume que el *punto-blanco* sigue un estándar y no se establece explícitamente.

Transformación RGB a CIELab

Se detalla a continuación la transformación del modelo de color RGB al modelo CIELab utilizada por la librería OpenCV.

Utilizando como ejemplo una imagen a color de 8 bits, se convierten los valores de los tres canales R,G y B, a puntos flotantes y se escalan para ajustarlos dentro del rango [0,1]:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \end{pmatrix} \quad (2.2)$$

Una vez obtenidos estos valores, se llevan a cabo las siguientes operaciones:

$$X = X/X_n \quad \text{donde } X_n = 0.950456$$

$$Z = Z/Z_n \quad \text{donde } Z_n = 1.088754$$

$$L = \begin{cases} 116 * Y^{1/3} - 16 & \text{para } Y > 0.008856 \\ 903.3 * Y & \text{para } Y \leq 0.008856 \end{cases} \quad (2.3)$$

$$a = 500 * (f(X) - f(Y)) + \text{delta}$$

$$b = 200 * (f(Y) - f(Z)) + \text{delta}$$

donde:

$$f(t) = \begin{cases} t^{1/3} & \text{para } t > 0.008856 \\ 7.787 * t + 16/116 & \text{para } t \leq 0.008856 \end{cases} \quad (2.4)$$

$$\text{delta} = \begin{cases} 128 & \text{para imágenes de 8 bits} \\ 0 & \text{para imágenes de punto flotante} \end{cases}$$

Los valores que pueden tomar cada uno de los parámetros son: $0 \leq L \leq 100$, $-127 \leq a \leq 127$, $-127 \leq b \leq 127$. Finalmente se realiza un ajuste sobre los valores para que queden entre 0 y 255:

$$L = L*255/100, \quad a = a + 128, \quad b = b + 128 \quad (2.5)$$

2.3 Operaciones morfológicas

La morfología matemática se basa en la teoría de conjuntos para definir operaciones que modifican de forma espacial los componentes de una imagen. Dentro del campo de *PDI*, la morfología matemática es utilizada por ejemplo para: realce de imágenes, análisis de formas, segmentación, compresión, restauración, detección de ejes, análisis de texturas, reducción de ruido, entre otros.

Este tipo de procesamiento, fundamenta todas sus operaciones en dos operadores: la erosión y la dilatación. Ambas están asociadas a un elemento estructurante, un objeto de referencia con dimensiones pequeñas (algunos píxeles) que presenta una estructura geométrica sencilla (punto, línea, cuadrado, círculo, etc.), asociado a la estructura o topología predominante en los elementos presentes en la imagen [22].

Erosión

Sean A y E conjuntos pertenecientes a Z^2 , la erosión de A por E , denotada $A \ominus E$, es definida como:

$$A \ominus E = \{z \mid (E)_z \subseteq A\} \tag{2.6}$$

Esta ecuación indica que la erosión de A por E es el conjunto de todos los puntos z , tal que E centrado en z está contenido dentro de A . El conjunto E es asumido como elemento estructurante. En la Figura 2.4 se muestra un ejemplo de erosión entre dos conjuntos.

Dilatación

Dados los conjuntos A y E pertenecientes a Z^2 , la dilatación de A por E , denotada $A \oplus E$, es definida como:

$$A \oplus E = \{z \mid (\hat{E})_z \cap A \neq \emptyset\} \tag{2.7}$$

Esta ecuación se basa en la reflexión de E (el elemento estructurante) sobre su origen, y en el desplazamiento de la misma por z . La dilatación de A por E entonces es el conjunto de todos los desplazamientos z , de manera que E y A se superpongan en por lo menos un elemento. En la Figura 2.5 se muestra un ejemplo del resultado de la dilatación entre dos conjuntos.

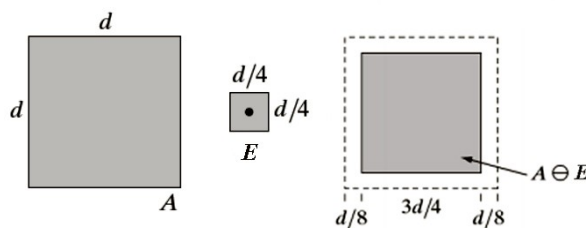


Figura 2.4: Ejemplo de erosión.

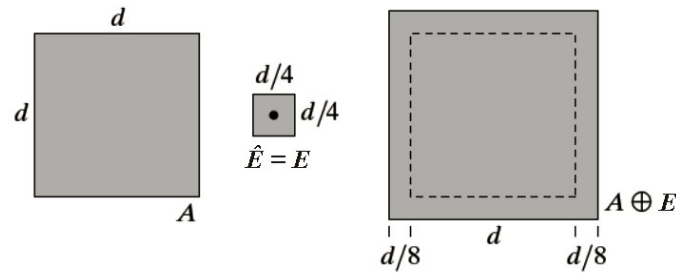


Figura 2.5: Ejemplo de dilatación.

2.4 Segmentación

La segmentación consiste en subdividir una imagen en regiones u objetos. El nivel de detalle con el cual se lleva a cabo la subdivisión dependerá del problema que se intente resolver. Muchos de los algoritmos de segmentación se basan en dos propiedades de valores de intensidad: discontinuidad y semejanza. En la primera categoría, el enfoque consiste en dividir una imagen basado en los cambios abruptos de intensidad, mayormente asociados con bordes o límites de objetos. La segunda categoría se basa en dividir una imagen en regiones que son similares de acuerdo a un conjunto de criterios predefinidos [21].

2.4.1 Umbralización

La umbralización es una técnica de segmentación ampliamente utilizada sobre imágenes en escala de grises, cuando existe una clara diferencia entre los niveles de intensidad de los objetos de interés y el fondo. Al aplicar un umbral T , la imagen en escala de grises $f(x,y)$ quedará binarizada; etiquetando con '1' los píxeles correspondientes al objeto y con '0' aquellos que forman parte del fondo:

$$g(x,y) = \begin{cases} 1 & \Leftrightarrow f(x,y) > T \\ 0 & \Leftrightarrow f(x,y) \leq T \end{cases} \quad (2.8)$$

El umbral puede depender de $f(x,y)$, de alguna propiedad local del píxel, $p(x,y)$, y de su propia posición:

$$T = T(f(x,y), p(x,y), x, y) \quad (2.9)$$

Si el umbral solo depende de $f(x,y)$ se denomina umbral global; en el caso de que además dependa de $p(x,y)$ el umbral es denominado local; y si depende también de la posición (x,y) del píxel, se denomina umbral dinámico.

La umbralización global es la más simple de sus tipos, y su correcto desempeño dependerá de cuán bien pueda partitionarse el histograma de la imagen. En la Figura 2.6 se muestra un ejemplo de umbralización, utilizando distintos valores de umbral: (a) imagen con 256 niveles de gris; (b) resultado con umbral bajo; (c) resultado con umbral alto; (d) resultado con umbral medio.

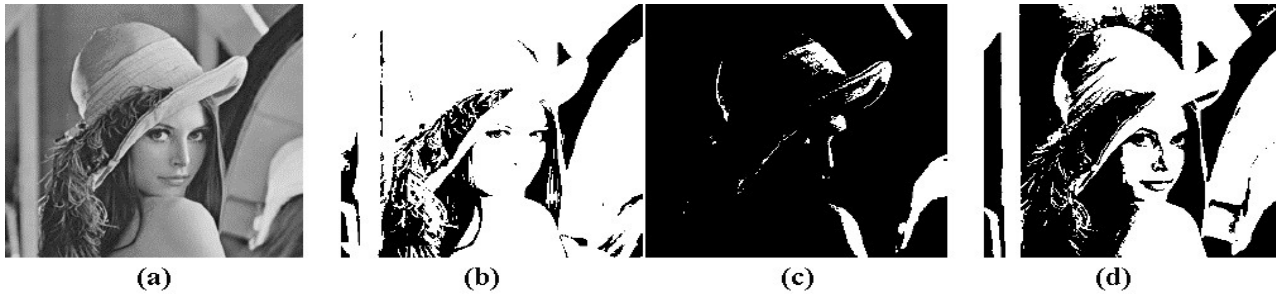


Figura 2.6: Ejemplos de umbralizado.

2.4.2 Componentes conexas

En una imagen binaria podemos encontrar regiones correspondientes a objetos de interés, sin más que atender a un criterio de proximidad espacial. Para poder identificar los píxeles que pertenecen a cada una de las regiones y entonces, asociar estas regiones a los objetos, es necesario obtener las *componentes conexas* presentes en la imagen.

Para definir correctamente una *componente conexa* es necesario definir previamente los siguientes conceptos:

Vecindad: Dado un píxel $P = (x,y)$ de una imagen, se define como δ -vecindad de P a los siguientes píxeles: $(x-1,y)$, $(x+1,y)$, $(x,y-1)$, $(x,y+1)$, $(x-1,y-1)$, $(x+1,y+1)$, $(x-1,y+1)$, $(x+1,y-1)$.

Adyacencia: Dos píxeles son adyacentes si son vecinos y sus valores satisfacen algún criterio específico de similitud, típicamente la igualdad.

Camino: Un camino de longitud n desde el píxel P_0 al píxel P_n es una secuencia de píxeles distintos P_0, P_1, \dots, P_n donde P_i es adyacente a P_{i-1} , para $1 \leq i \leq n$. Dos píxeles P y Q están *conectados* si existe un camino entre ellos.

Dado un conjunto de S píxeles, y un píxel $P \in S$, una *componente conexa* de S es el subconjunto de todos los píxeles $Q \in S$ conectados a P . Como en una imagen puede haber muchas regiones que son de interés, es necesario en una primera instancia identificar cada región de forma individual, para luego poder extraer las características necesarias, como por ejemplo su área o centro de gravedad.

Este proceso se conoce como *etiquetado* y consiste en asignar una misma etiqueta a cada uno de los píxeles que pertenecen a una *componente conexa*. De esta forma, la imagen de etiquetas obtenida permitirá la extracción individualizada de características de cada una de las regiones de la imagen. En la Figura 2.7 se muestra un ejemplo del proceso de detección de regiones, donde: (a) imagen de grises; (b) resultado de umbralizado; (c) resultado etiquetado; (d) detección de centros.

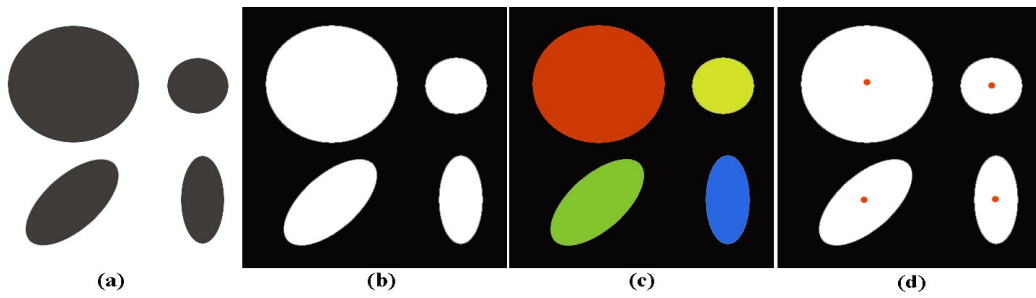


Figura 2.7: Ejemplo de detección y etiquetado de regiones.

2.5 Filtro de mediana

Dentro del área de captura y procesamiento de imágenes digitales, es común encontrarse con imágenes que posean ruido. Podemos definir al ruido como información no deseada que contamina una imagen, y sus principales fuentes surgen durante las etapas de adquisición y/o transmisión de imágenes, como por ejemplo, condiciones climáticas que afectan el desempeño de un sensor de imágenes, o la presencia de interferencias en el canal utilizado para transmitir una imagen. Existen diferentes tipos de ruidos, y se clasifican según la manera en que afectan a una imagen: gaussiano, impulsivo, uniforme, exponencial, gamma, etc.

Una de las principales herramientas utilizadas para la restauración y realce de imágenes son los filtros espaciales. Estos filtros procesan todos los píxeles de la imagen, trabajando sobre pequeñas vecindades y aplicando operaciones predefinidas. Para ello, se utiliza una máscara de $n \times n$ puntos y se la desplaza por toda la imagen aplicando la operación predefinida. Dependiendo del tipo de operación que se utilice, los filtros espaciales se pueden clasificar en lineales y no lineales.

Un filtro de mediana es un filtro espacial no lineal que reemplaza el valor del pixel central de la máscara por el valor de la mediana de todas las intensidades pertenecientes a la vecindad de ese pixel incluyendo su propio valor:

$$f(x, y) = \text{mediana}_{(s,t) \in S_{x,y}} \{g(s, t)\} \quad (2.10)$$

donde S_{xy} representa el conjunto de coordenadas (s, t) de la máscara g centrada en (x, y) .

La función principal de los filtros de mediana es forzar a los píxeles con valores de intensidad muy distintos a sus vecinos, a tomar valores más próximos a los mismos. De este modo se corrigen los píxeles con valores extremos de intensidad que aparecen en áreas aisladas de una imagen.

2.6 Mezcla de gaussianas

En este trabajo, se emplea Mezcla de Gaussianas (MoG, del inglés *Mixture of Gaussians*) para modelar el fondo del video. Este método representa los píxeles de la escena mediante un modelo de mezclas, permitiendo obtener una representación multimodal del fondo.

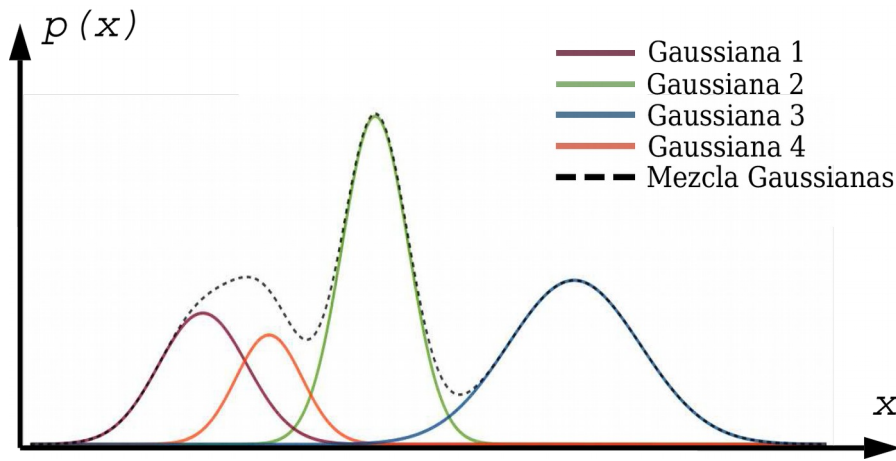


Figura 2.8: Gráfica de una mezcla de 4 gaussianas con pesos iguales.

Este modelado puede usarse como modelo del fondo de la escena y de esta manera es útil para remover movimientos repetitivos de objetos en una escena como lo son: el reluciente del agua, el movimiento de las ramas de los árboles, la oscilación de una bandera, etc. A continuación se explicará el método propuesto por Stauffer y Grimson [14].

Se consideran los valores de un píxel como un proceso aleatorio, el cual es una serie de tiempo de los valores del píxel, por ejemplo, escalares para imágenes en escala de gris o vectores para imágenes a color. En el tiempo t , lo que se conoce de un píxel particular (x_0, y_0) es su historia:

$$\{X^1, \dots, X^t\} = \{I_{(x_0, y_0)}^i : 1 \leq i \leq t\} \quad (2.11)$$

donde I es la secuencia de imágenes.

Cuando se tiene un conjunto de datos con una gran dispersión de valores, es difícil representar su estructura de forma apropiada mediante una distribución Gaussiana simple. Sin embargo, la superposición de dos o más distribuciones puede ajustarse mejor a la dispersión de los datos. Esta superposición se forma mediante una combinación lineal finita de distribuciones simples y es llamado modelo de mezcla de Gaussianas.

La historia reciente de cada píxel es modelada mediante una mezcla de K distribuciones gaussianas. En la Figura 2.8 se muestra un ejemplo de un modelo de mezcla unidimensional compuesto por 4 Gaussianas simples, donde en línea de puntos se representa la mezcla de las mismas. La probabilidad de observar el valor del píxel actual es:

$$P(X_t) = \sum_{i=1}^K w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (2.12)$$

donde

K : número de distribuciones gaussianas a utilizar en la mezcla

$w_{i,t}$: estimación del peso de la gaussiana i de la mezcla en el tiempo t , esto es, qué porción de los datos han sido generados por la gaussiana i . Además $\sum_k w_k = 1$ y $0 \leq w_k \leq 1$

$\mu_{i,t}$: media de la gaussiana i en el tiempo t
 $\Sigma_{i,t}$: matriz de covarianza de la gaussiana i en el tiempo t
 $\eta()$: función de densidad de probabilidad de la distribución gaussiana

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} * |\Sigma^{1/2}|} * e^{-1/2(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (2.13)$$

K se determina de acuerdo a la memoria disponible y el poder computacional, generalmente de 3 a 5 distribuciones son utilizadas. Además por razones computacionales, la matriz de covarianza se asume de la forma:

$$\Sigma_{k,t} = \sigma_k^2 \mathbf{I} \quad (2.14)$$

Existe un modelo de mezcla para cada píxel de la imagen. Dado que el procesamiento de un píxel varía con el tiempo (proceso no estacionario), y teniendo en cuenta el costo computacional, se utiliza el método *on-line K-means* para comprobar si un nuevo valor de un píxel X_t es generado por alguna de las K distribuciones gaussianas existentes. Para ello se verifica si el valor del píxel se encuentra a menos de 2.5σ de la media μ .

Si ninguna de las K distribuciones genera el valor del píxel, la distribución menos probable se sustituye por una distribución con el valor actual del píxel como valor medio, una variación inicialmente alta y un bajo peso anterior. Los pesos anteriores de las K distribuciones al tiempo t , w_t , son ajustados de la siguiente forma:

$$w_{k,t} = (1 - \alpha) w_{k,t-1} + \alpha (M_{k,t}) \quad (2.15)$$

donde α es la tasa de aprendizaje y $M_{k,t}$ es 1 para el modelo emparejado y 0 para el resto de los modelos.

Los parámetros μ y σ de las distribuciones que no generaron el valor del píxel permanecen iguales. Los parámetros de la distribución que emparejó con el nuevo valor son actualizados de la siguiente forma:

$$\mu_t = (1 - \rho) \mu_{t-1} + \rho X_t \quad (2.16)$$

$$\sigma_t^2 = (1 - \rho) \sigma_{t-1}^2 + \rho (X_t - \mu_t)^T (X_t - \mu_t) \quad (2.17)$$

donde

$$\rho = \alpha \eta(X_t | \mu_k, \sigma_k) \quad (2.18)$$

Una de las ventajas de este método es que cuando se permite que algo forme parte del fondo, no se destruye el modelo existente del fondo. El fondo de color original permanece en la mezcla hasta que se convierte en la k^{th} menos probable y se observa un color nuevo. Por lo tanto, si un objeto permanece estacionario el tiempo suficiente para convertirse en fondo y luego se mueve, la distribución del fondo previo continúa existiendo con los mismos μ y σ , pero con bajo w y es rápidamente reincorporada al fondo.

Para determinar cuales de las gaussianas de la mezcla son seleccionadas como modelo de fondo, primero deben ser ordenadas por el valor de w/σ . Así, las distribuciones de fondo más probables permanecen en la parte superior y las distribuciones de fondos transitorios menos probables gravitan hacia la parte inferior y pueden ser sustituidas por nuevas distribuciones.

Las primeras B distribuciones son elegidas como modelo de fondo, donde:

$$B = \operatorname{argmin}_b \left(\sum_{k=1}^b w_k > T \right) \quad (2.19)$$

Esto selecciona las mejores distribuciones mediante un umbral T . Si un valor pequeño para T es elegido, el modelo de fondo es usualmente unimodal. Si este es el caso, utilizando solo la distribución más probable se ahorrará procesamiento. Si el valor de T es grande, el modelo de fondo es multi-modal.

Finalmente para determinar si un píxel forma parte o no del fondo, se compara el conjunto de distribuciones B de la ecuación 2.19 con la distribución con la cual concordó el píxel, llámese k' a esa distribución, entonces: si $k' \in B$ el píxel es parte del fondo, de lo contrario el píxel no forma parte del fondo.

2.7 Ecuación adaptativa del histograma

El *histograma* es una función discreta que representa el número de píxeles en la imagen en función de los niveles de intensidad r . La probabilidad $P(r)$ de ocurrencia de un determinado nivel r , se define como:

$$P(r) = \frac{A(r)}{B} \quad (2.20)$$

donde B es el número de píxeles en la imagen y $A(r)$ es el número de píxeles que poseen el nivel de intensidad r . En la Figura 2.9 se puede ver un ejemplo del histograma correspondiente a una imagen en niveles de gris.

La forma del histograma permite evidenciar ciertas particularidades de la imagen, como lo son el tipo de fondo, el contraste, y en general que tan oscura o brillante es. Una imagen cuyo histograma posee una distribución de los niveles de intensidad concentrados en una zona, presenta un contraste muy bajo, mientras que una imagen cuyo histograma tiene una amplia distribución de los niveles de intensidad, presenta un alto contraste.

En algunos casos, las imágenes poseen histogramas con distribuciones de los niveles de gris que están lejos de ser ideales. Ejemplos típicos son imágenes adquiridas bajo malas condiciones de iluminación, donde existe un exceso de oscuridad, o al contrario, demasiados niveles de blanco. Uno de los métodos más empleados para mejorar o ampliar el contraste en imágenes es la ecualización del histograma.

Ecuación del histograma

La operación de *ecuación* consiste en la redistribución de los niveles de gris de la imagen original sobre todos los niveles disponibles en el sistema, siguiendo una ley particular que determina el tipo de modificación.

Sea r la variable que representa los niveles de gris de la imagen a mejorar, y suponiendo que la intensidad de los píxeles son cantidades continuas que han sido normalizadas en el rango $[0,1]$, donde $r = 0$ es negro y $r = 1$ es blanco, se define la transformación $s = T(r)$:

1. $T(r)$ es de valor único y monótonamente creciente en el intervalo $0 \leq r \leq 1$.
2. $0 \leq T(r) \leq 1$ para $0 \leq r \leq 1$.

La primer condición preserva el orden entre el negro y el blanco de la escala de grises. La segunda condición garantiza una aplicación que es coherente con el rango de valores permitidos para la intensidad de los píxeles.

Los niveles de gris originales y su transformada se pueden caracterizar por sus funciones de densidad de probabilidad $p_r(r)$ y $p_s(s)$. De la teoría de probabilidades, si $p_r(r)$ y $T(r)$ son conocidas y $T^{-1}(s)$ verifica la condición 1, entonces la función de distribución de probabilidad de los niveles de gris transformados es:

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_{r=T^{-1}(s)} \quad (2.21)$$

Considérese la función de transformación:

$$s = T(r) = \int_0^r [p_r(w)] dw \quad (2.22)$$

denominada *función de distribución acumulada r*. De esta ecuación, la derivada de s respecto de r es:

$$\frac{ds}{dr} = p_r(r) \quad (2.23)$$

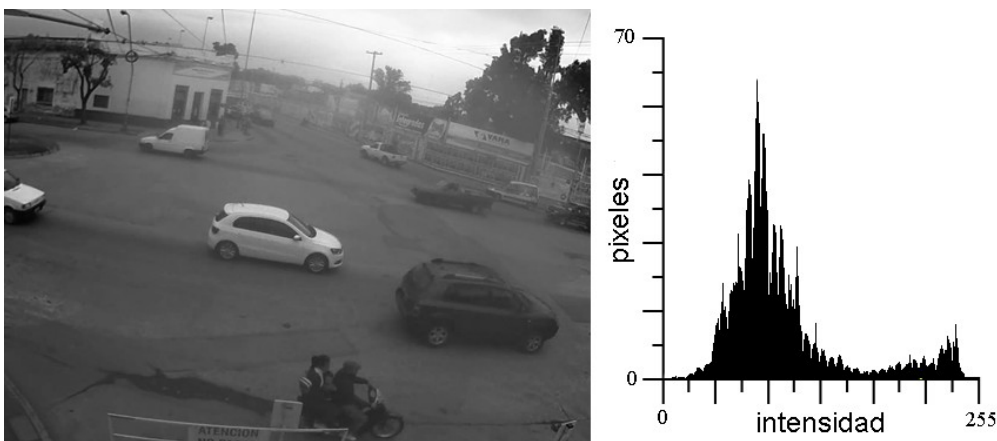


Figura 2.9: Ejemplo de histograma.

Sustituyendo en (2.19) se obtiene:

$$p_s(s) = \left[p_r(r) \frac{dr}{ds} \right]_r = T^{-1}(s) = 1 \quad (2.24)$$

que da una densidad uniforme en el intervalo de definición de la variable s . Esto significa que cuando se emplee una función de transformación igual a la función de distribución acumulada se produce una imagen con niveles de gris uniformemente distribuidos.

Para poder ser aplicado en el procesamiento digital de imágenes, los conceptos antes vistos deben ser expresados en forma discreta. Por lo tanto, la ecuación 2.22 se expresa como:

$$s_k = T(r_k) = \sum_{j=0}^k [p_r(r_j)] \quad (2.25)$$

En la aproximación discreta se puede observar que existe una redistribución de los niveles de gris. En la Figura 2.10 se presenta un ejemplo de ecualización del histograma, donde en la parte superior (a) se puede ver la imagen original con su correspondiente histograma, y en la parte inferior (b) se puede ver la imagen ecualizada y el histograma resultante.

Ecualización del histograma adaptativa con contraste limitado

La ecualización del histograma descrita anteriormente, considera el contraste global de la imagen. En muchos casos, este no es el mejor de los caminos a seguir. Un ejemplo se ve en la Figura 2.10(b), en la cual la imagen resultante de la ecualización pierde detalles debido a que una zona concentrada de píxeles toma valores muy altos de intensidad.

Se puede mejorar el resultado anterior utilizando una ecualización del histograma adaptativa. Este método divide la imagen en pequeños bloques, y ecualiza el histograma de cada uno de ellos. Un bloque afectará con su ecualización del histograma a una zona con el doble de su tamaño. En la Figura 2.11 se presenta un ejemplo donde los puntos negros representan los píxeles centrales de cada bloque, la región coloreada de amarillo representa un bloque (R_{ij}), y el recuadro de color rojo (S_{ij}) representa la zona afectada por la ecualización del histograma del bloque pintado de amarillo.

La Figura 2.12 muestra un ejemplo de como se determinan los valores de cada pixel de la imagen al aplicar una ecualización del histograma adaptativa. Las funciones de transformación son apropiadas para los píxeles del centro de los bloques, coloreados de negro en la parte izquierda de la figura. Todos los otros píxeles se transforman con hasta cuatro funciones de transformación de los bloques con centros más cercanos a ellos, y se les asigna valores interpolados.

Los píxeles de la mayor parte de la imagen (sombreado azul) se calculan mediante una interpolación bilineal, los píxeles cercanos al límite (sombreado verde) se calculan mediante una interpolación lineal, y los píxeles cerca de las esquinas (sombreados de color rojo) tienen que ser tratados de manera especial, debido a que su vecindad no se encuentra completa dentro de la imagen. Esto se puede resolver ampliando la imagen mediante la duplicación de filas y columnas de píxeles en los límites de la imagen.

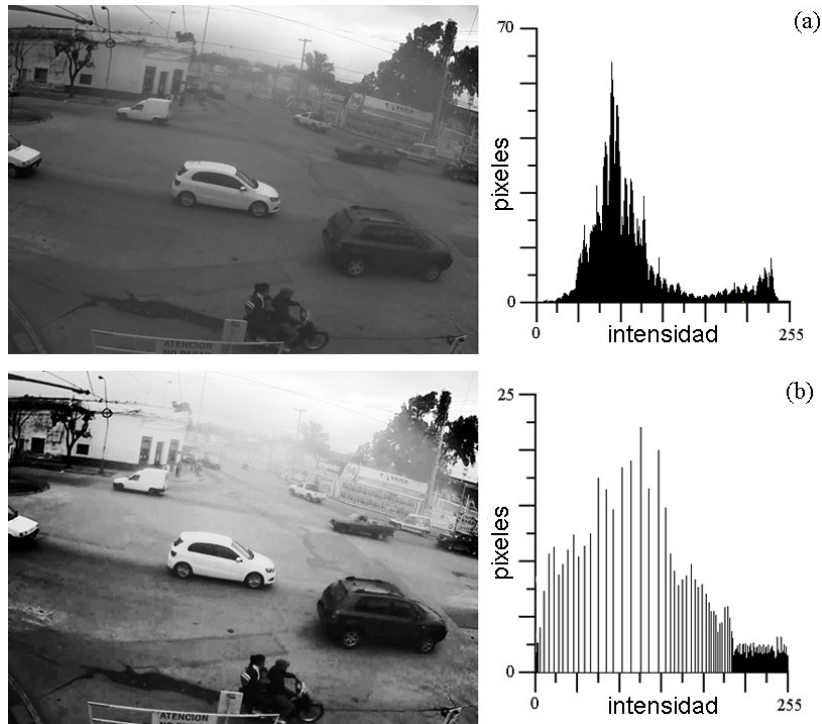


Figura 2.10: Ejemplo de ecualización del histograma

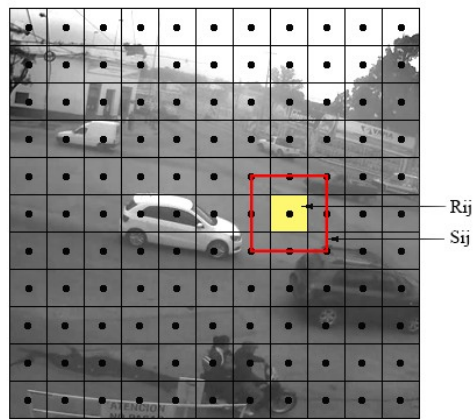


Figura 2.11: Imagen dividida en bloques.

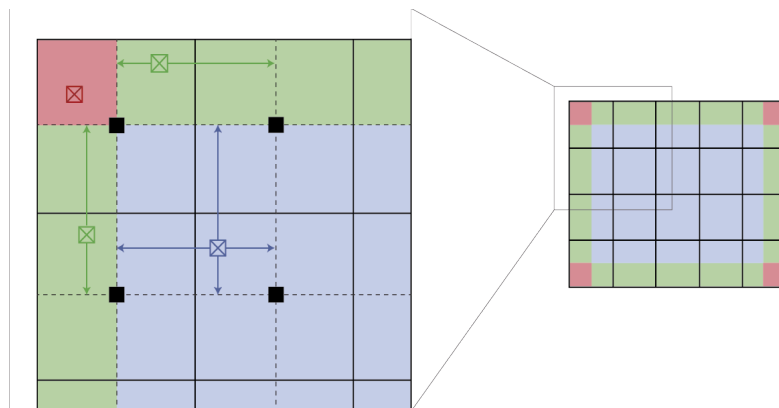


Figura 2.12: Ejemplo de interpolación en la ecualización del histograma adaptativo.

En el caso de existir la presencia de ruido dentro de los bloques, el mismo será amplificado. Para evitar esto, se utiliza el método de ecualización del histograma adaptativa con contraste limitado (CLAHE).

En este método, se establece un límite de contraste donde los pixeles del histograma que posean valores de intensidad mayores al valor impuesto, serán recortados y distribuidos uniformemente en el resto del histograma antes de aplicar la ecualización. En la Figura 2.13 se puede ver un ejemplo del resultado de aplicar una ecualización del histograma adaptativa con contraste limitado. Se puede apreciar a simple vista como se mejora el contraste de la imagen sin perder detalles.

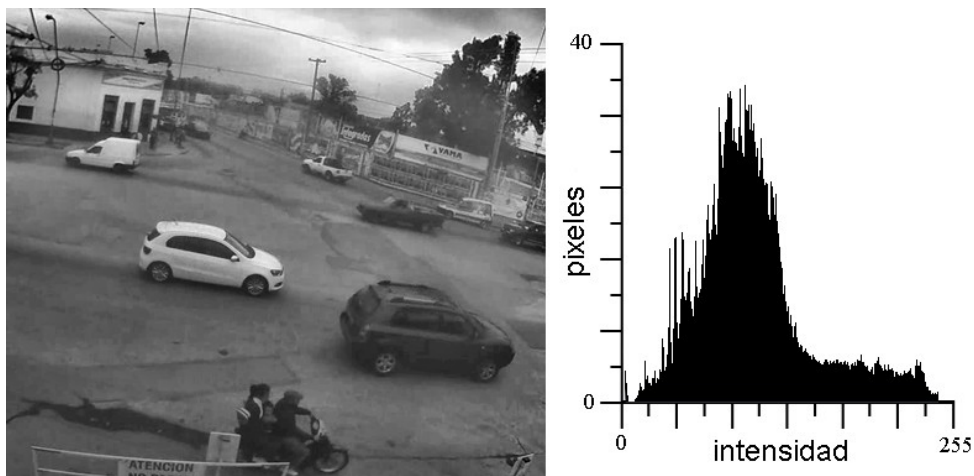


Figura 2.13: Ejemplo de ecualización adaptativa del histograma con contraste limitado.

Análisis y Diseño del Sistema

Como se mencionó en el primer capítulo, el objetivo de este proyecto es desarrollar un sistema que utilice técnicas de procesamiento de video para medir intensidad de tráfico. Este sistema, requiere como dato de entrada una señal de video proveniente de una cámara que sea capaz de captar el tránsito. Por este motivo, se contactó al Municipio de la Ciudad de Santa Fe, para comentarles de que se trataba este proyecto y solicitar el acceso a alguna de las cámaras que actualmente utiliza para seguridad urbana. Dado que el acceso a las cámaras no fue posible, el Municipio nos otorgó un conjunto de videos grabados por las mismas.

Por lo tanto, el diseño del sistema propuesto se elaboró basándose en el estudio del estado del arte realizado previamente, y en un análisis de características de los videos obtenidos. Si bien se trabajó con videos, el sistema resultante podrá ser adaptado en un futuro para funcionar en tiempo real utilizando el flujo de video de una cámara en vivo.

3.1 Análisis de la base de datos

La base de datos que se recopiló esta compuesta por 15 videos grabados por cámaras de seguridad pertenecientes al Municipio de la Ciudad de Santa Fe, en formato MP4 y ASF, con velocidades entre 15-25 fps, y con resoluciones que varían entre 704x576 y 1280x720. Estos videos poseen aproximadamente una hora de duración, y captan varias zonas de la ciudad a diferentes horarios del día.

Previo a comenzar con la etapa de diseño, es necesario identificar las características principales que poseen los videos de tráfico. Para ello, se llevo a cabo un relevamiento de parámetros característicos, detectados mediante observación y que afectarán el funcionamiento del sistema. Este proceso de análisis demandó una importante cantidad de tiempo, pero fue imprescindible para comprender el sistema que debe desarrollarse en su totalidad, tanto en lo que respecta a las funcionalidades como a potenciales problemas que pueden presentarse y deben ser previstos. Se mencionan a continuación los parámetros detectados explicando como podrían influir en el funcionamiento del sistema:

Presencia de obstáculos

La presencia de obstáculos en la escena del video puede afectar la detección de vehículos, provocando detecciones falsas, incorrectas o no deseadas. Además, un gran número de obstáculos generará una complejidad extra para el sistema.

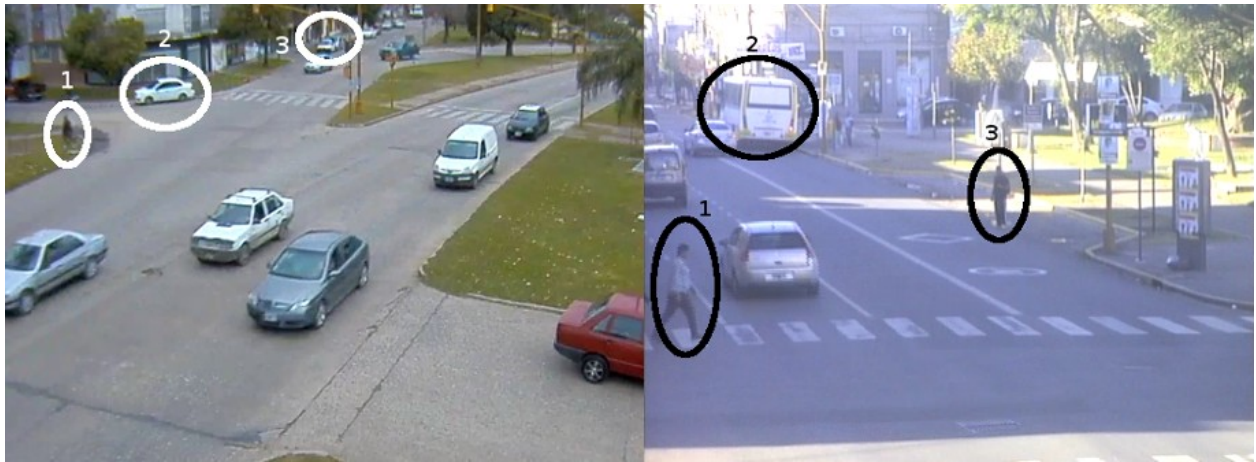


Figura 3.1: Ejemplos de obstáculos dentro de la escena.

Algunos ejemplos de obstáculos son: peatones circulando por la vereda o cruzando una calle, vehículos circulando por zonas donde no es de interés que sean detectados, la presencia de carteles, árboles y/o vehículos estacionados, etc.

En la Figura 3.1 se presentan casos de escenas con obstáculos. En la imagen de la izquierda se observa un peatón circulando por la vereda (1), un automóvil circulando por una zona que no es de interés analizar (2), y un par de automóviles estacionados (3). En la imagen de la derecha se observan dos peatones a punto de cruzar la calle (1 y 3) y un colectivo que se detiene (2).

Condiciones climáticas

Las condiciones climáticas pueden afectar el funcionamiento del sistema ya que alteran la escena y provocan movimientos de objetos que perjudican la detección de vehículos. Por ejemplo, la lluvia aporta una gran cantidad de ruido a la escena del video, los vientos mueven ramas y carteles que generan falsas detecciones, los días nublados alteran la iluminación de la escena, y en contraparte, los días soleados generan la presencia de sombras.

Cambios de iluminación en la escena

Diferentes niveles de iluminación en una escena pueden afectar el funcionamiento de ciertos métodos de modelado de fondo, segmentación y detección de vehículos. Por ejemplo, un método de modelado de fondo que funciona bien durante el día pero deja de funcionar correctamente durante la noche, las luces de los vehículos afectan el proceso de segmentación, las reflexiones de luces generan falsas detecciones, entre otros (Figura 3.2).



Figura 3.2: Ejemplos de escenas con diferentes niveles de iluminación.

Posición de la cámara

La posición de la cámara encargada de capturar el tráfico es fundamental dentro el diseño del sistema, ya que determinará cómo se visualizarán los vehículos en la escena y en qué sentido circularán los mismos. Un correcto posicionamiento de la cámara evitará posibles oclusiones parciales entre vehículos, las interferencias de carteles o árboles en la escena, y además, facilitará el seguimiento y conteo de vehículos.

Presencia de sombras

Las sombras en una escena pueden generar falsas detecciones y problemas durante el seguimiento de vehículos. Por ejemplo, sombras que se generan por variaciones rápidas de iluminación pueden ser detectadas como un objeto en movimiento, o la sombra de un vehículo que se superpone con otro vehículo de la escena haciendo que el sistema interprete ambos vehículos como uno solo. En la Figura 3.3 se pueden apreciar algunos ejemplos de lo mencionado.



Figura 3.3: Ejemplos de sombras que generan falsas detecciones.

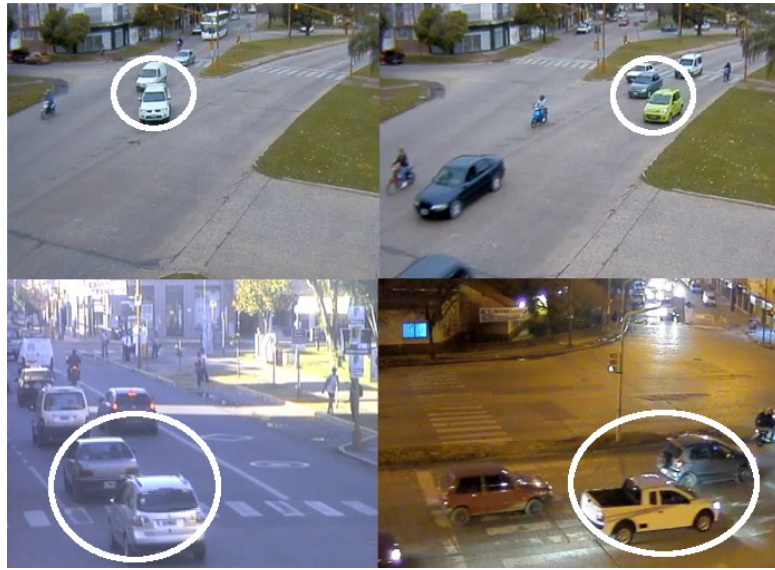


Figura 3.4: Ejemplos de oclusiones parciales entre vehículos

Oclusiones parciales

Las oclusiones parciales se dan cuando un vehículo es parcialmente ocluido por otro provocando que el sistema interprete ambos vehículos como uno solo. Para que este efecto se produzca, tanto las velocidades de los vehículos como la posición de la cámara cumplen un papel fundamental (Figura 3.4).

3.2 Especificación de requerimientos

El término *requerimiento* se utiliza de varias formas en la industria del software: en algunos casos, un requerimiento es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste; en otros casos, es una definición detallada y formal de una función del sistema [23].

A menudo, los requerimientos pueden ser clasificados en funcionales y no funcionales. Los *requerimientos funcionales* son declaraciones de los servicios que debe proporcionar el sistema y de la manera en que éste debe reaccionar a entradas particulares. En algunos casos, los requerimientos funcionales también pueden declarar explícitamente lo que el sistema no debe hacer. Los *requerimientos no funcionales*, son restricciones de los servicios o funciones ofrecidos por el sistema y rara vez se aplican al sistema en su totalidad [23].

A los fines de identificar funcionalidades y limitaciones del sistema propuesto, se realizó un análisis y relevamiento de requerimientos. En una primera instancia se llevaron a cabo dos reuniones con representantes de la municipalidad de Santa Fe, donde se definieron requerimientos del usuario. Posteriormente, luego de realizar un estudio del estado del arte en medición de tráfico e

investigar dentro del campo de Visión Computacional sobre métodos y procedimientos para detección y seguimiento de vehículos, se realizó un relevamiento de requerimientos del sistema:

Requerimientos Funcionales:

1. Carga de archivos: el sistema deberá permitir la carga de archivos de video con formato AVI, MPEG-1 y MP4.
2. Definición de región de interés: el usuario del sistema podrá definir la región del video sobre la cual desea realizar detección de vehículos.
3. Elaborar un modelo de la escena del video.
4. Generar una máscara binaria con los vehículos detectados.
5. Clasificar vehículos por tamaño.
6. Registrar vehículos detectados.

Requerimientos No Funcionales:

1. El sistema deberá realizar todo el procesamiento en tiempo real.
2. Los parámetros del sistema serán ajustables para cada caso particular.
3. El sistema deberá ser capaz de detectar como mínimo un 70% de los vehículos que circulen por la región definida.

3.3 Casos de uso

Los casos de uso son una técnica basada en escenarios, que reflejan la interacción entre el sistema y los actores involucrados. Sus principales objetivos son definir el límite entre el sistema y los elementos externos al mismo, y capturar el conjunto de funcionalidades y comportamientos del sistema a desarrollar [23].

Un actor es una entidad externa al sistema que se modela y que puede interactuar con el mismo. Para el sistema desarrollado, se modela una entidad “usuario” como actor: un usuario podrá ser cualquier persona que posea conocimientos básicos acerca del funcionamiento del sistema. Como posibles usuarios a futuro, se puede mencionar al personal a cargo de las cámaras de video de la municipalidad, personal perteneciente al área responsable de la movilidad urbana de una ciudad, un operario dedicado específicamente a medir el tráfico, etc.

A los fines de representar el comportamiento del sistema propuesto, se elaboraron un conjunto de casos de uso. Los mismos son representados en la Figura 3.5 y detallados a continuación:

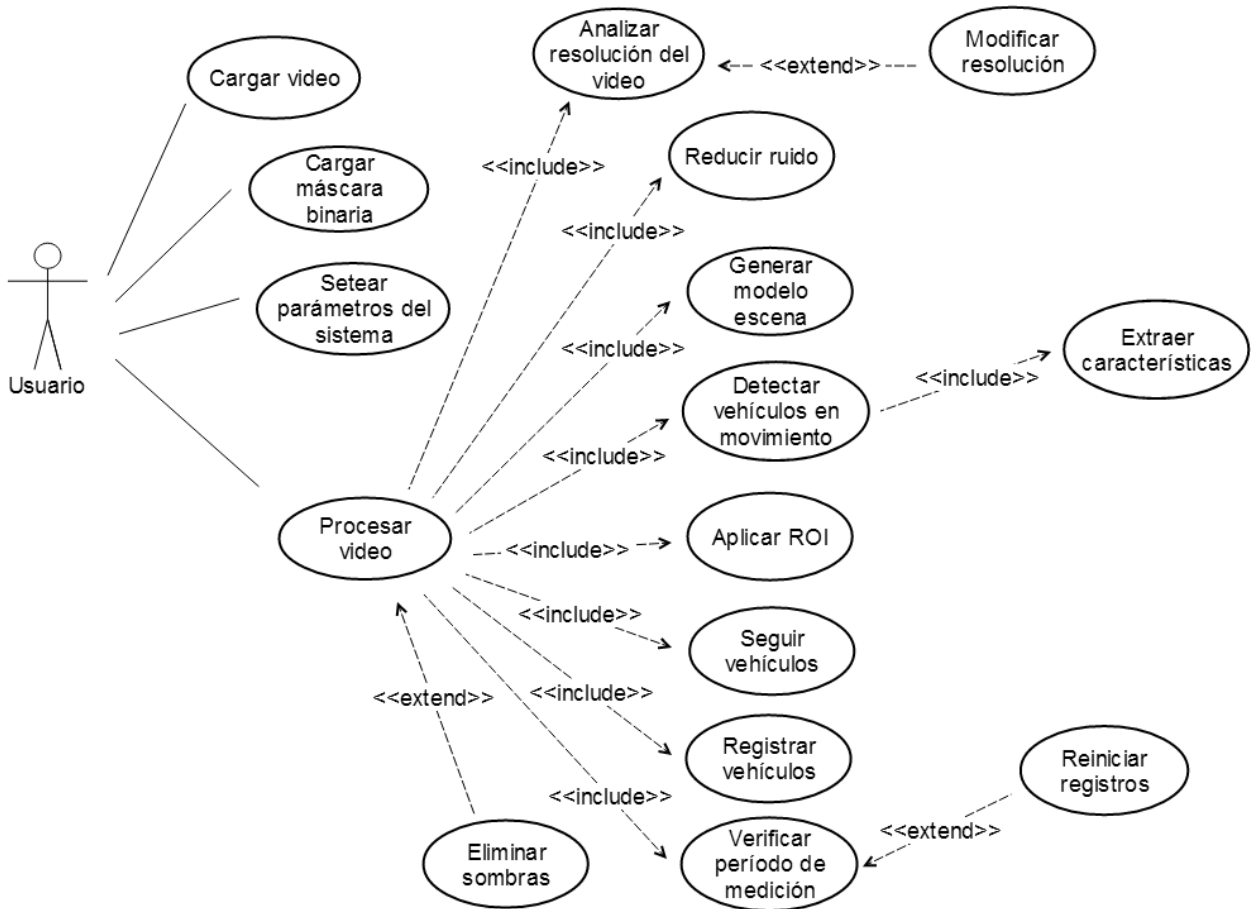


Figura 3.5: Diagrama de casos de uso.

Caso de uso: Cargar video	
Actor: Usuario	
Curso Normal:	Curso Alternativo:
1. El caso de uso comienza cuando el usuario presiona el botón “Cargar video”.	
2. El sistema abre una ventana emergente y el usuario selecciona el video a cargar.	
3. El sistema verifica que el archivo seleccionado sea válido.	
4. El sistema muestra un icono verde y habilita el botón “Cargar máscara”.	4.1 El sistema detecta que el archivo no es válido. 4.1.1 El sistema muestra un ícono rojo. 4.1.2 El sistema vuelve al paso 1.
5. Finaliza el caso de uso.	

Caso de uso: Cargar máscara binaria	
Actor: Usuario	
Curso Normal:	Curso Alternativo:
1. El caso de uso comienza cuando el usuario presiona el botón “Cargar máscara”.	
2. El sistema abre una ventana emergente y el usuario selecciona la máscara binaria a cargar.	
3. El sistema verifica que el archivo seleccionado sea válido.	
4. El sistema muestra un icono verde y habilita el botón “Comenzar”.	4.1 El sistema detecta que el archivo no es válido. 4.1.1 El sistema muestra un ícono rojo. 4.1.2 El sistema vuelve al paso 1.
5. Finaliza el caso de uso.	

Caso de uso: Setear parámetros del sistema	
Actor: Usuario	
Curso Normal:	Curso Alternativo:
1. El caso de uso comienza cuando el usuario modifica alguno de los parámetros por defecto del sistema.	
2. El sistema limita los valores modificados a un cierto rango definido.	
3. Finaliza el caso de uso.	

Caso de uso: Procesar video	
Actor: Usuario	
Curso Normal:	Curso Alternativo:
1. El caso de uso comienza cuando el usuario presiona el botón “Comenzar”.	
2. Se invoca al CU: Analizar resolución del video.	
3. Se invoca al CU: Reducir ruido.	
4. Se invoca al CU: Generar modelo escena.	
5. Se invoca al CU: Detectar vehículos en movimiento.	
6. El sistema verifica si el usuario realizó la configuración de parámetros para eliminación de sombras.	
7. Se invoca al CU: Eliminar sombras.	7.1 El usuario no configuró los parámetros necesarios para eliminación de sombras. 7.1.1 No se invoca al CU.

8. Se invoca al CU: Aplicar ROI.	
9. Se invoca al CU: Seguir vehículos.	9.1 El sistema no detectó vehículos. 9.1.1 No se invoca al CU.
10. Se invoca al CU: Registrar vehículos.	
11. Se invoca al CU: Verificar período de medición.	
12. El sistema verifica si quedan fotogramas restantes por procesar.	
13. El sistema detecta fotogramas y vuelve al paso 2.	13.1 El sistema no detecta fotogramas y da por finalizado el procesamiento.
14. Finaliza el caso de uso.	

3.4 Diseño del sistema

Un diseño de software, es una descripción de la estructura del software que se va a implementar, los datos que son parte del sistema, las interfaces entre los componentes del sistema, y algunas veces, los algoritmos utilizados. No se obtiene directamente, sino que se desarrolla de manera iterativa a través de diferentes versiones.

El sistema propuesto en este proyecto, es un sistema de tiempo real para medición de tráfico. Un *sistema de tiempo real* es un sistema informático que interacciona con su entorno físico y responde a los estímulos del entorno dentro de un plazo de tiempo determinado. Particularmente, el sistema propuesto es un *sistema de tiempo real suave*, cuyo funcionamiento se degrada si los resultados no se producen de acuerdo con los requerimientos temporales especificados [23].

El diseño del sistema de medición de tráfico, se elaboró basándose en el estudio del estado del arte en conjunto con la identificación de parámetros característicos en los videos. Estos dos estudios previos, se utilizaron principalmente para seleccionar los métodos de procesamiento de imagen y video que mejor se adaptaban a los requerimientos del sistema, tanto en funcionalidad como en rendimiento. En la Figura 3.6 se presenta el diagrama general del sistema, identificando los principales módulos que conforman el mismo.

Módulo de Pre-procesamiento

El pre-procesamiento es el primer módulo de trabajo y tiene por objetivo preparar el video de entrada para su posterior procesamiento, realizando mejoras que permitan suprimir distorsiones no deseadas, y realzando las propiedades importantes.

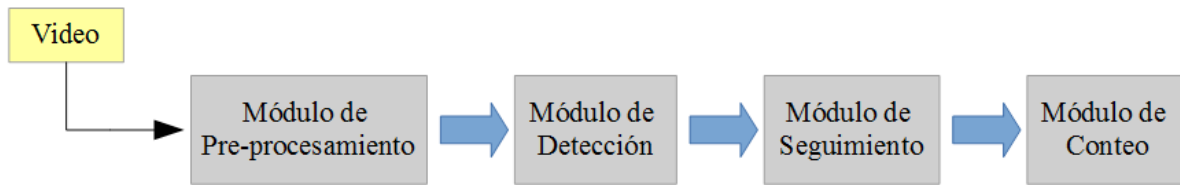


Figura 3.6: Diagrama general del sistema.

Dentro de este módulo se aplican las siguientes técnicas:

- *Modificación de la resolución*: se analizará la resolución del video de entrada, y en base a los requerimientos de las técnicas de procesamientos futuros, será optimizada para reducir costos computacionales.
- *Técnicas de suavizado*: se utilizarán técnicas de suavizado para disminuir el nivel de posibles ruidos presentes en el video.
- *Normalización del nivel de iluminación*: se analizará el nivel de iluminación de la escena del video, y en caso de ser necesario se aplicará una normalización para mejorar el funcionamiento de posteriores algoritmos. Se explorarán diversos modelos de colores para este fin.
- *Determinación de una Región de Interés (ROI)*: se delimitarán ROIs dentro de la escena del video para reducir el costo computacional de procesamiento y excluir las zonas que contengan información que no es relevante para el sistema.
- *Parametrización inicial*: se realizarán una serie de mediciones para definir rangos de tamaños que poseen los distintos tipos de vehículos en la escena, y distancias entre los mismos, que serán necesarias para procesamientos futuros.

Módulo de Detección

El módulo de detección está compuesto por dos bloques principales: el modelado del fondo y la detección de vehículos en movimiento. El modelado del fondo tiene por objetivo construir una imagen de referencia que represente la escena del video. La detección de vehículos en movimiento, consiste en realizar una segmentación de los fotogramas del video, seguido de una clasificación de píxeles correspondientes a background (fondo) o foreground (frente). Las técnicas que se aplican en este módulo son:

- *Modelado del fondo*: se utilizará el método de Mezcla de Gaussianas (MoG) para generar un modelo de la escena del video, el cual se irá actualizando al transcurrir un determinado período.

- *Segmentación*: para segmentar los vehículos en movimiento, se realizará la diferencia entre el modelo de la escena y los fotogramas del video. Posteriormente, se utilizarán operaciones de morfología matemática para obtener un resultado final con menos distorsiones.
- *Eliminación de sombras*: dentro de la escena del video, es muy probable que aparezcan sombras que generen inconvenientes a la hora de detectar y clasificar vehículos. Por lo tanto, se eliminará la mayor cantidad posible de éstas utilizando un método basado en un cambio de modelos de color (de RGB a Lab).
- *Representación*: la representación de cada uno de los vehículos detectados será expresada como una región particular de píxeles, a partir de la cual se podrán estimar diferentes características, como por ejemplo el área y el centro geométrico, que serán de utilidad para procesamientos posteriores.
- *Validación de centros geométricos*: se realizará una validación de centros geométricos, donde se utilizará un límite de área mínima y se evaluará la posición de los centros respecto a la ROI previamente definida.

Módulo de Seguimiento

Este módulo tiene por objetivo seguir la trayectoria de los vehículos detectados en el módulo previo. Para ello, se deben identificar las correspondencias entre los vehículos detectados en los diferentes fotogramas del video. A continuación se describen las técnicas que se aplican en este módulo:

- *Seguimiento*: se realizará el seguimiento de los vehículos detectados dentro de la ROI definida. Para ello se determinará la correspondencia entre los centros geométricos detectados en dos fotogramas consecutivos, utilizando una medida de mínima distancia entre los mismos.
- *Resolución de oclusiones parciales*: se resolverán las oclusiones parciales entre vehículos que se produzcan de manera momentánea dentro de la ROI definida. Para ello, se analizarán las correspondencias entre las posiciones de los centros detectados en un determinado fotograma y el inmediato anterior.

Módulo de Conteo

Dentro de este módulo se lleva un registro de la cantidad de automóviles que circularon dentro de la ROI. Para ello se utilizaron las siguientes técnicas:

- *Límite de conteo*: los vehículos detectados serán registrados una vez que abandonen la ROI previamente definida. Esto se determina verificando la posición actual y previa de los centros detectados para cada vehículo.
- *Actualización de datos*: se llevara un registro constante de los vehículos que abandonen la ROI, clasificándolos según su tamaño. Una vez cumplido el período de tiempo establecido por el usuario para medir la intensidad de tráfico, estos registros serán reiniciados.

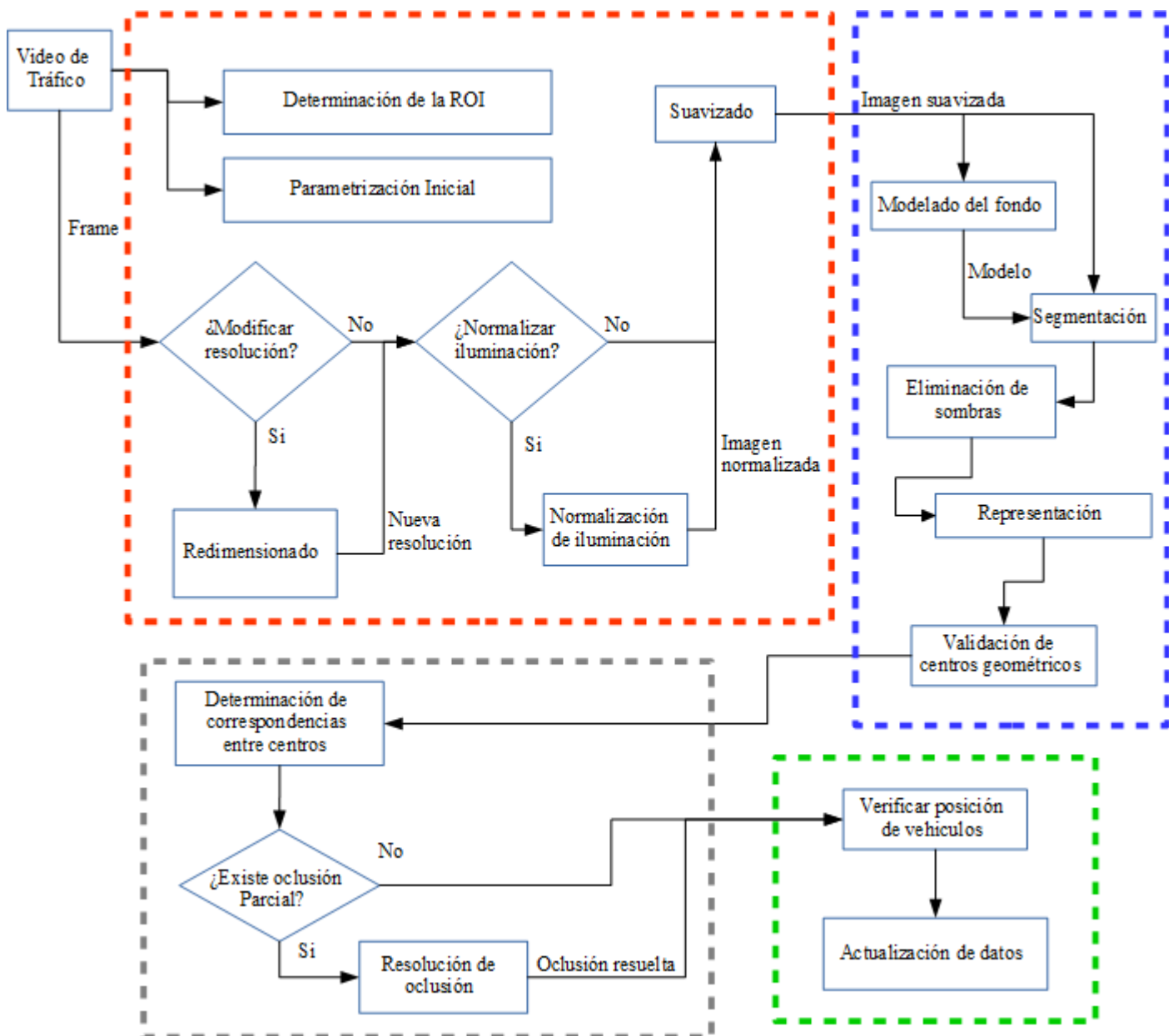


Figura 3.7: Diseño detallado del sistema.

En la Figura 3.7 se presenta un diagrama detallado del diseño, donde se puede apreciar el flujo de trabajo del sistema. Mediante recuadros de color, se representan los módulos especificados en el diagrama general: Módulo de Pre-procesamiento (rojo); Módulo de Detección (azul); Módulo de Seguimiento (gris); Módulo de Conteo (verde).

3.5 Diagrama de clases

Un diagrama de clases es un tipo de diagrama de estructura estática, que modela los conceptos del dominio de la aplicación: clases, atributos, operaciones (o métodos), e interrelaciones. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables aplicando ingeniería directa e inversa [24].

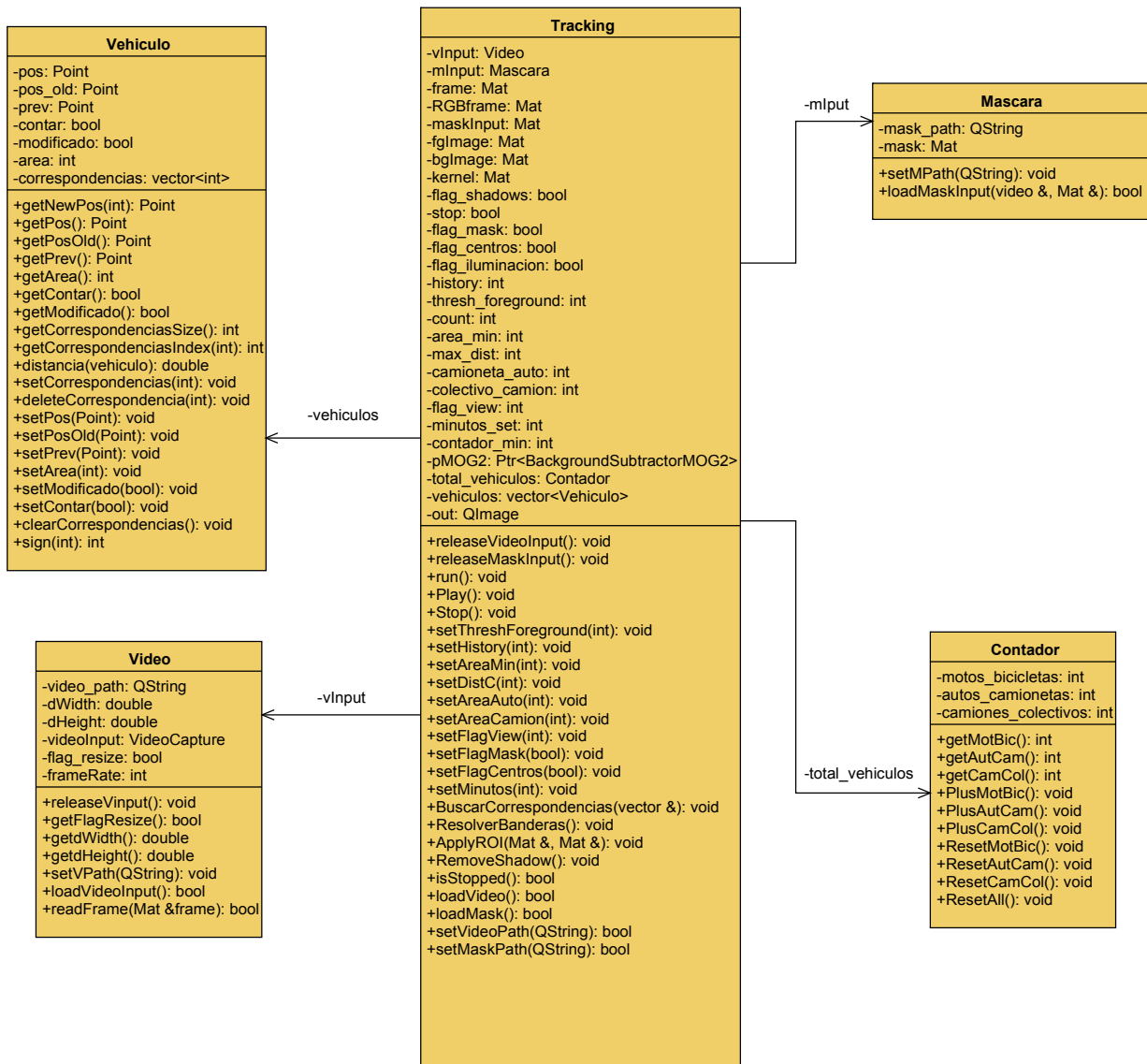


Figura 3.8: Diagrama de clases.

En la Figura 3.8 se observa el diagrama de clases diseñado para el sistema de medición de intensidad de tráfico, en el cual se definieron 5 clases: Video, Mascara, Vehiculo, Contador y Tracking.

La clase *Video* contiene todas las propiedades del video utilizado como entrada al sistema: directorio en el que se encuentra almacenado, dimensiones y cantidad de imágenes por segundo. Además contiene un atributo de tipo *VideoCapture*, una clase de la librería OpenCV que puede ser utilizada tanto para la captura de video de una cámara como para la lectura de archivos de video o secuencias de imágenes. Las operaciones que se encuentran dentro de esta clase tienen el propósito de proporcionar información del video cuando se lo requiera.

La clase *Máscara* se utilizará para representar las ROIs definidas dentro de la escena de video. Para definir una o varias ROIs se utilizará una imagen binaria como máscara sobre el video, la cual tendrá pintada en color blanco las zonas de interés y en color negro el resto. Esta imagen será definida y cargada en el sistema por el usuario.

La clase *Contador* será utilizada para llevar un registro de la cantidad y tipo de vehículos detectados por el sistema. Posee tres acumuladores como atributos, uno por cada tipo de clasificación de vehículo, y sus operaciones están destinadas a la actualización y reseteo de los mismos.

La clase *Vehiculo* se utiliza para representar a cada vehículo detectado por el sistema. Dentro de sus atributos se encuentra la posición del vehículo en diferentes instantes, banderas que indican cuando la posición del vehículo fue actualizada y cuando el vehículo debe ser registrado, el área del vehículo, y un vector utilizado para realizar la correspondencia del vehículo en dos fotogramas consecutivos. La mayoría de las operaciones de esta clase, están destinadas a la carga y obtención de los valores de cada uno de los atributos recién mencionados.

Finalmente, la clase *Tracking* es la clase principal y realiza todas las tareas necesarias para llevar a cabo la detección de vehículos. Alguno de sus principales atributos son:

- Un objeto de tipo Video que contiene el video cargado por el usuario.
- Un objeto de tipo Máscara que contiene la imagen cargada por el usuario.
- Un modelo de la escena del video.
- Banderas para indicar estados de procesamiento.
- Umbrales utilizados para segmentación.
- Tamaños de áreas para clasificación de vehículos.
- Indicador de tiempo para medir la intensidad de tráfico.
- Período de actualización del modelo de la escena.
- Un objeto de tipo Contador para llevar un registro de vehículos.
- Un vector de objetos de tipo Vehiculo, que almacenará los vehículos que se encuentran dentro de la ROI definida en la escena del video.

Las operaciones realizadas dentro de esta clase, ejecutan todas las técnicas y métodos mencionados en la sección anterior donde se describe el diseño del sistema. Por lo tanto esta clase realiza tareas pertenecientes a los módulos de Pre-procesamiento, Detección, Seguimiento y Conteo.



Figura 3.9: Capturas de videos descartados por posición inadecuada de la cámara.

3.6 Modificación de la base de datos

Finalizado el diseño del sistema, se llevó a cabo una revisión de los videos que conforman la base de datos con el objetivo de determinar cuáles de los mismos eran adecuados para trabajar con el sistema y cuáles debían ser descartados. Inicialmente, se optó por no descartar ningún video y utilizar todos en la etapa de análisis de parámetros característicos.

De los 15 videos que conformaban la base de datos en un principio, fueron descartados 11 por los siguientes motivos:

- Un video fue descartado ya que la cámara que lo grabó no se encontraba fija en una posición, sino que giraba hacia los costados. Al encontrarse la cámara en movimiento, no es posible generar un modelo de la escena ya que la misma cambia constantemente. Además no es posible definir una zona para detección de vehículos.
- Se descartaron dos videos debido a que las cámaras que los grabaron estaban ubicadas en una posición inadecuada para capturar el tráfico. En la Figura 3.9 se presentan capturas de los videos, en donde se puede ver que las cámaras graban una zona en la que los vehículos se detienen debido a la presencia de un semáforo. Esto impide definir una zona para detección donde los vehículos se encuentren en constante circulación.
- Ocho videos se descartaron debido a que fueron grabados durante la noche y el sistema se limita a trabajar únicamente con videos grabados durante el día. En la Figura 3.10 se presentan algunas capturas de los videos descartados.

Además, por cuestiones de rendimiento se redujo la tasa de fotogramas de cada video a 10 fotogramas por segundo. Finalmente la base de datos queda conformada por cuatro videos de una hora de duración cada uno, grabados durante el día por una cámara fija posicionada adecuadamente.



Figura 3.10: Ejemplos de videos descartados por grabarse durante la noche.

Implementación

La investigación y desarrollo de los métodos que conforman el sistema de medición de intensidad de tráfico, se llevaron a cabo mediante la elaboración de prototipos.

Un prototipo, “es una versión inicial de un sistema de software que se utiliza para demostrar los conceptos, probar las opciones de diseño y, de forma general, enterarse más acerca del problema y sus posibles soluciones” [23]. En general, se pueden encontrar dos tipos principales de prototipos: desechables y evolutivos.

Un prototipo *desechable* se construye lo más rápido posible, implementándose únicamente aquellos requerimientos que estén pobremente comprendidos. Se utilizan experimentalmente, y son desechados una vez que se ha obtenido la información deseada. Funcionan muy bien en forma aislada para verificar partes relativamente pequeñas de problemas complejos.

Por otro lado, un prototipo *evolutivo* se basa en la idea de desarrollar una implementación inicial relativamente simple, la cual se va re-diseñando y haciendo más compleja a través de varias etapas. Al final de este proceso, el prototipo se convierte en el sistema requerido.

Durante la etapa de desarrollo, se trabajó con ambos tipos de prototipos. Se utilizaron prototipos desechables para implementar y evaluar, tanto el funcionamiento como el rendimiento de los métodos de procesamiento que forman parte de los módulos definidos en la etapa de diseño. Adicionalmente, se utilizaron prototipos desechables para probar métodos alternativos a los propuestos en un principio, y mediante comparación de resultados, alguno de ellos fueron reemplazados. Para la implementación del sistema se trabajó con un prototipo evolutivo, que luego de varios ciclos de ajustes, se convirtió en la versión final del sistema.

4.1 Tecnologías utilizadas

Previo a comenzar con la etapa de desarrollo, es de suma importancia realizar una adecuada selección de las tecnologías con las que se va a trabajar, para que el sistema resultante tenga un buen desempeño. En primer lugar, se realizó un relevamiento de librerías de visión artificial, para determinar cuál de ellas era conveniente utilizar. Se analizaron tres librerías: CImg, Image Processing Toolbox (Matlab) y OpenCV; optando finalmente por utilizar esta última.



Figura 4.1: Video representado como secuencia de fotogramas.

OpenCV es una librería de visión por computador (CV) de código abierto, que puede ser utilizada para desarrollar algoritmos en C++, C, Python y Java, y es compatible con Linux, Windows, Mac OS, iOS y Android. Ha sido diseñada para ser eficiente en cuanto a gasto de recursos computacionales y con un enfoque hacia las aplicaciones de tiempo real. Cuenta con más de 2500 algoritmos optimizados, que abarcan muchas de las áreas de CV [25].

Un video es una sucesión de fotogramas, que ejecutados en secuencia simulan movimiento (Figura 4.1). Esta librería se utilizó para la captura individual de cada fotograma del video de entrada, y puede utilizarse de igual manera para la captura de fotogramas de una cámara de video. Además, se utilizaron varias de sus funciones avanzadas dentro de cada uno de los bloques generales mencionados en el diseño del sistema.

El desarrollo se llevo a cabo utilizando C++ como lenguaje de programación. El principal motivo por el cual se optó por este lenguaje, fue que el mismo se integra con OpenCV y su velocidad de ejecución alta. Además, C++ se puede utilizar de manera gratuita en cualquier entorno de desarrollo, y dado que se tenía conocimiento y experiencia previa trabajando con este lenguaje, se agilizó el desarrollo del sistema.

Para la elaboración del código fuente se trabajó con el software Zinjai¹: un IDE libre y gratuito para programar en C/C++, que presenta una interfaz sencilla e incluye funcionalidades avanzadas.

4.2 Metodología

En un principio, se trabajó sobre un único video de la base de datos, el cual poseía características deseables para el funcionamiento del sistema, tales como una buena posición de la cámara, ausencia de sombras y obstáculos en la escena, y un nivel de iluminación bueno. Una vez implementadas todas las funcionalidades del sistema, se utilizaron videos con características diferentes para evaluar el comportamiento del mismo, y realizar los ajustes y modificaciones necesarios.

¹ <http://zinjai.sourceforge.net/>

4.2.1 Módulo de pre-procesamiento

Este módulo recibe como entrada un video de tráfico y aplica sobre el mismo una serie de procesamientos que lo preparan para ser utilizado en los módulos posteriores.

Definición de parámetros iniciales

El sistema posee un conjunto de parámetros iniciales que deben ser definidos antes de comenzar con el procesamiento. Para conocer el valor de cada uno, es necesario realizar un análisis previo del video. En un principio, cada parámetro se inicializa con un determinado valor, que mediante pruebas se irá ajustando hasta encontrar el valor indicado. En la Tabla 4.1 se listan los parámetros iniciales que posee sistema.

Tabla 4.1: Parámetros iniciales del sistema.

Parámetro	Tipo	Descripción
flag_iluminacion	bool	Bandera que indica si se debe normalizar la iluminación de la escena.
history	int	Cantidad de frames previos almacenados para modelar la escena.
thresh_foreground	int	Umbral definido para la segmentación de objetos en movimiento.
area_min	int	Área mínima que debe poseer un objeto para ser detectado.
max_dist	int	Distancia de correspondencia máxima entre centros detectados en frames consecutivos.
camioneta_auto	int	Área mínima que debe poseer un objeto para ser clasificado como camioneta / auto.
colectivo_camion	int	Área mínima que debe poseer un objeto para ser clasificado como colectivo / camión.
minutos_set	int	Período de medición definido en minutos.
video_path	QString	Directorio del video.
mask_path	QString	Directorio de la máscara binaria.

Las pruebas que se deben llevar a cabo para determinar el valor de cada parámetro inicial son:

- Analizar las áreas de los objetos que se mueven dentro de la escena del video, para definir rangos de valores que permitan clasificar los vehículos según su tamaño y descartar objetos pequeños que no sean de interés. Se definen los parámetros: *area_min*, *camioneta_auto*, *colectivo_camion*.
- Analizar la escena del video e identificar si la misma requiere una normalización de iluminación. En caso de aplicar una normalización, verificar si los resultados son positivos o desfavorables. Se define el parámetro *flag_iluminacion*.
- Visualizar la distancia aproximada que hay entre los centros detectados para cada vehículo en fotogramas consecutivos, y definir una distancia máxima utilizada para establecer la correspondencia entre vehículos de distintos fotogramas. Se define el parámetro: *max_dist*.

- Establecer la duración del período de medición en función de las necesidades propias. Se define el parámetro: *minutos_set*.
- Analizar la velocidad con la que cambia la escena del video, y en función a ello definir la cantidad de fotogramas que se almacenarán para generar el modelo de la misma. Se define el parámetro: *history*.
- Utilizar diferentes valores de umbrales para la segmentación de los objetos en movimiento dentro de la escena. En base a los resultados obtenidos, seleccionar el valor de umbral que se considere indicado. Se define el parámetro: *thresh_foreground*.

Adquisición del fotograma y análisis de resolución

Para el tratamiento de los videos, es necesario utilizar operaciones que reconozcan la secuencia de fotogramas y extraigan información de cada uno.

La carga de un video en el sistema se lleva a cabo mediante funciones de la librería OpenCV, que capturan, decodifican y devuelven los fotogramas en formato Mat: una clase utilizada para representar una densidad numérica n-dimensional en un arreglo de canal simple o multicanal. En este caso, el fotograma representado consiste en una imagen en formato RGB, por lo que el contenedor Mat será una matriz con el ancho y alto de la imagen, y con tres niveles de profundidad.

La adquisición de los fotogramas se realiza dentro de un bucle general, en el que se lleva a cabo la mayor parte del procesamiento del sistema, por lo que se trabaja con un fotograma por cada iteración del mismo. Una vez obtenido un fotograma, se verifica que su resolución no sea mayor a 800x600, aplicando una re-dimensión en caso de superarlo. Con esto se evita que el sistema trabaje con fotogramas de gran resolución, que implicarían un costo computacional mayor, y además, permite generalizar procesamientos posteriores.

Definición de la ROI

Se puede definir una Región de Interés como un subconjunto seleccionado de muestras dentro de un conjunto de datos identificado para un determinado propósito. Particularmente para el sistema desarrollado, el conjunto de datos identificado son los píxeles pertenecientes a cada fotograma del video de entrada. Por lo tanto, la ROI utilizada será un subconjunto de píxeles pertenecientes a cada fotograma.

Para definir una ROI se utiliza una imagen binaria (en tonos de blanco y negro), donde la zona de color blanco representa la ROI y la zona de color negro representa los datos a ser descartados. Esta imagen es elaborada por un usuario del sistema mediante un editor de imágenes, tomando un fotograma del video a analizar como imagen inicial y coloreando de color blanco la ROI y de color negro el resto de la imagen. La imagen binaria debe ser cargada en el sistema antes de comenzar con el procesamiento del video de tráfico. En la Figura 4.2 se muestra un ejemplo de definición de una ROI.

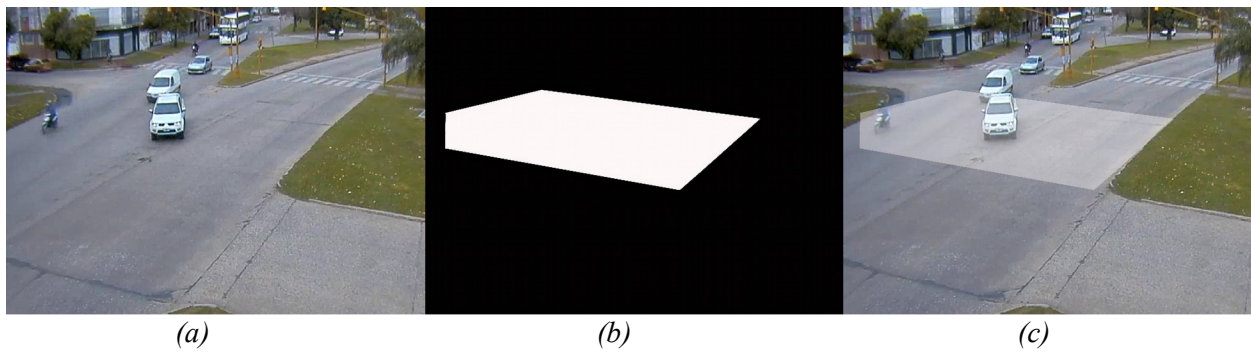


Figura 4.2: Ejemplo de definición de una ROI. (a): fotograma original, (b): máscara binaria definida por el usuario, (c): máscara binaria aplicada sobre el fotograma.

El objetivo de definir una ROI es delimitar una región en la cual el sistema analizará el tráfico. De esta forma, se excluyen zonas que contengan información irrelevante y que puedan dificultar el procesamiento posterior. Adicionalmente, al reducir la zona de análisis se reduce el procesamiento del sistema.

El principal aspecto a tener en cuenta a la hora de definir una ROI, es que dentro de la misma los vehículos deberán estar en constante circulación. Además, se debe realizar un análisis previo de la escena para poder identificar que zona es la más adecuada para evaluar el tráfico, intentando contemplar posibles interferencias como por ejemplo la circulación de peatones, la presencia de semáforos, los cruces de calle, etc.

Normalización del nivel de iluminación

La principal fuente de iluminación presente en los videos de tráfico es la luz ambiental, la cual varía a lo largo del día y se ve afectada por las condiciones climáticas. Un factor importante a tener en cuenta es la variación en la intensidad de la iluminación, ya que afecta la apariencia de los vehículos en la escena produciendo representaciones desiguales de los mismos. Estas variaciones son indeseables ya que dificultan el posterior procesamiento de segmentación y seguimiento de vehículos.

Para normalizar los niveles de iluminación de la escena en un fotograma, se realizan tres procesamientos: Primero se aplica sobre el fotograma una transformación de espacios de color de RGB a CIE Lab, para obtener una representación de la luminosidad de los colores (canal L). Posteriormente, se aplica sobre el canal L una Ecuación del Histograma Adaptativa con Contraste Limitado (CLAHE), para normalizar los niveles de luminosidad de cada color presente. Finalmente, se aplica una transformación de espacios de color de CIE Lab a RGB, para obtener nuevamente el fotograma representado en el espacio RGB.

La normalización de los niveles de iluminación es opcional dentro del sistema. El usuario es quien decide si es necesario aplicarla, basándose en un análisis previo del video a procesar. Si bien una normalización de la iluminación mejora el contraste del video, en algunos casos puede resultar desfavorable.

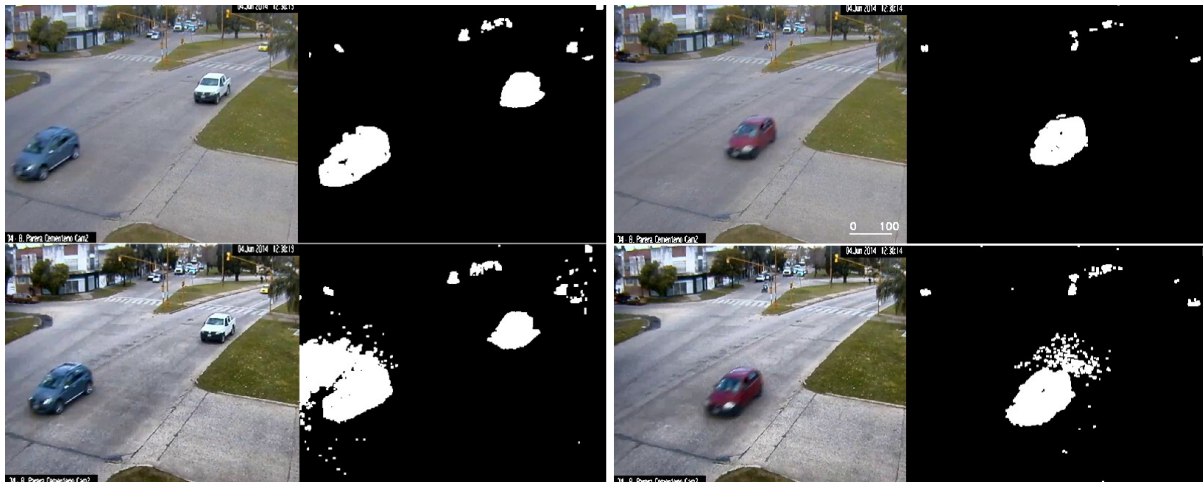


Figura 4.3: Ejemplos de normalización de iluminación desfavorable.

En la Figura 4.3 se muestran dos ejemplos de un video cuyo nivel de iluminación es bueno, y aún así se aplica una normalización de iluminación. El aumento en el contraste del video provoca que nuevas zonas superen el umbral de segmentación, generando ruido y falsas detecciones. En la parte superior se presentan capturas pertenecientes al video original, y en la parte inferior las imágenes pertenecientes al video con los niveles de iluminación normalizados.

Suavizado del fotograma

Mediante observación, se detectó que en la mayoría de los videos se producían pequeños parpadeos en determinados píxeles. Esto es interpretado como un ruido sal-pimienta para el sistema, ya que el mismo detecta algunos de estos parpadeos durante el proceso de segmentación.

Para disminuir este ruido, se aplica sobre cada fotograma del video un filtro de la mediana trabajando sobre una vecindad de 3×3 píxeles. Utilizando este filtro, se fuerza a los píxeles que poseen un valor de intensidad muy diferente a los de sus vecinos, a tomar valores más próximos a los mismos.

4.2.2 Módulo de detección

El objetivo principal del módulo de detección es identificar los vehículos que se encuentran en movimiento dentro de la escena del video. Este módulo recibe como entrada los fotogramas pre-procesados en el módulo anterior, y genera como salida un conjunto de parámetros que representan a cada vehículo detectado.

Modelado del fondo

El modelado del fondo consiste en generar una imagen que represente la escena de video sin vehículos (Figura 4.4). Para ello, se utiliza de forma periódica el método de Mezcla de Gaussianas (MoG), donde se crea un modelo estadístico del fondo computado a partir de una secuencia de fotogramas.



Figura 4.4: Ejemplo de modelado del fondo. (a): fotograma, (b): máscara de segmentación.

Para aplicar el método MoG, se trabaja con un historial de fotogramas previos cuyo tamaño es uno de los parámetros iniciales del sistema y debe ser configurado por el usuario. Un factor importante a tener en cuenta a la hora de definir la cantidad de fotogramas previos que conformarán el historial, es la velocidad con la que se producen cambios en la escena del video. Para una escena que cambia rápidamente se utiliza un historial pequeño de fotogramas, mientras que para escenas que no presentan grandes cambios en períodos cortos de tiempo se utiliza un historial mayor.

La actualización del historial de fotogramas se realiza una vez por segundo mediante una función de la librería OpenCV, la cual carga un fotograma dentro del historial de fotogramas previos y elimina el más antiguo almacenado. La actualización del modelo del fondo se realiza cada dos segundos mediante la función que aplica el método MoG y retorna una imagen con el modelo de la escena. Algunos aspectos a tener en cuenta para este modelo son:

- Un vehículo que se encuentre en movimiento y se detenga por un cierto tiempo, será interpretado por el método como parte del fondo, y por lo tanto será incluido en la imagen del modelo.
- Si la cantidad de fotogramas que conforman el historial es pequeña, en ocasiones pueden aparecer dentro del modelo “fantasmas” de los vehículos que se encuentran en movimiento, como se puede ver en la parte superior derecha de la Figura 4.4(b).

Segmentación

La segmentación tiene por objetivo separar los objetos que se encuentran en movimiento del resto de la escena del video. Se mencionarán a continuación una serie de métodos que se encuentran implementados como funciones dentro de la librería OpenCV.

El primer paso es comparar el valor de los píxeles del fotograma actual con los del modelo de la escena para detectar que píxeles pertenecen al fondo. Para ello, se realiza una diferencia absoluta entre el fotograma y el modelo. Posteriormente, se aplica sobre el resultado de la diferencia una transformación de espacios de color de RGB a escala de grises.

El siguiente paso es crear una máscara binaria con los vehículos en movimiento. Se aplica un filtro Gaussiano sobre la imagen en escala de grises, para debilitar las impurezas resultantes de la diferencia absoluta. Luego, se realiza un umbralizado de la imagen obteniendo así su versión binaria. Un parámetro importante en este proceso es el valor del umbral a utilizar, el cual forma parte del conjunto de parámetros iniciales del sistema y debe ser configurado por el usuario.

Finalmente, para refinar el resultado y eliminar pequeñas impurezas se aplica sobre la imagen binaria una erosión. En la Figura 4.5 se muestra un ejemplo de la máscara binaria resultante de la segmentación.

Eliminación de sombras

La eliminación de sombras consiste en detectar las sombras producidas por los vehículos en movimiento, que son interpretadas por el sistema como parte del vehículo o como un objeto independiente, y eliminar los píxeles blancos de la máscara binaria que se corresponden con las mismas.

La eliminación de sombras se realiza mediante segmentación de color en el espacio CIE-Lab. El proceso de detección es el siguiente:

1. El sistema le muestra al usuario en una ventana, una copia del fotograma actual del video.
2. El usuario selecciona con el mouse un pixel que considere como sombra.
3. El sistema guarda la coordenada del pixel seleccionado.
4. El sistema aplica sobre la copia una conversión de espacios de color de RGB a CIE-Lab.
5. El sistema guarda los valores de cada uno de los canales (L , a y b) que hay en la coordenada del pixel anteriormente seleccionado.



Figura 4.5: Ejemplo de segmentación. (a):fotograma, (b):máscara de segmentación.

6. El usuario define para cada canal un radio de amplitud, que es utilizado para definir rangos máximos y mínimos de valores que serán segmentados.
7. El sistema aplica una segmentación sobre los tres canales utilizando los rangos definidos previamente y se genera una máscara binaria con las zonas segmentadas.
8. Para eliminar las sombras detectadas, se realiza una diferencia entre la máscara binaria resultante de la segmentación de vehículos y la máscara binaria resultante de la detección de sombras.

La eliminación de sombras es opcional dentro del sistema, y es el usuario quien decide si es necesario aplicarla o no. Generalmente, este procesamiento se aplica sobre videos cuyas escenas contienen una iluminación con un nivel alto de intensidad. Sin embargo, también puede utilizarse para refinar la máscara binaria de segmentación eliminando las sombras que generan los vehículos en la parte inferior. Un ejemplo de esto último se puede ver en la Figura 4.5, donde la escena no presenta sombras marcadas, pero al mirar la máscara binaria se puede observar que en la parte inferior de cada vehículo aparece una pequeña sombra, la cual podría causar problemas en la detección de vehículos que circulan cerca.

En la Figura 4.6 se muestra un ejemplo de un video cuya escena se ve afectada por la luz del sol, provocando que los vehículos que circulan por la misma produzcan sombras. El sistema interpreta estas sombras como parte de los vehículos, incrementando notablemente el área de píxeles que representan a cada uno dentro de la máscara binaria de segmentación.

Para reducir estos efectos negativos, se aplica la eliminación de sombras utilizando los valores definidos por el usuario. En la Figura 4.7 se presenta una captura del sistema para eliminación de sombras, en el cual se observa en la parte izquierda una copia del fotograma que contiene en color negro las zonas detectadas como sombra.



Figura 4.6: Ejemplo de presencia de sombras en la escena del video. (a): fotograma, (b): máscara de segmentación.

En la Figura 4.8 se presenta la máscara binaria que contiene todo lo que es interpretado por el sistema como sombra, y el resultado de la diferencia con la máscara de segmentación. Se puede ver que la sombra no se elimina completamente de esta última, pero se reduce considerablemente su presencia en la misma.

Representación de vehículos

Una vez obtenida la máscara binaria con los vehículos en movimiento detectados, se lleva a cabo la identificación y caracterización de los mismos. Cada vehículo es representado como una región de píxeles conectados, por lo tanto, el sistema debe identificar estas regiones y extraer sus características.

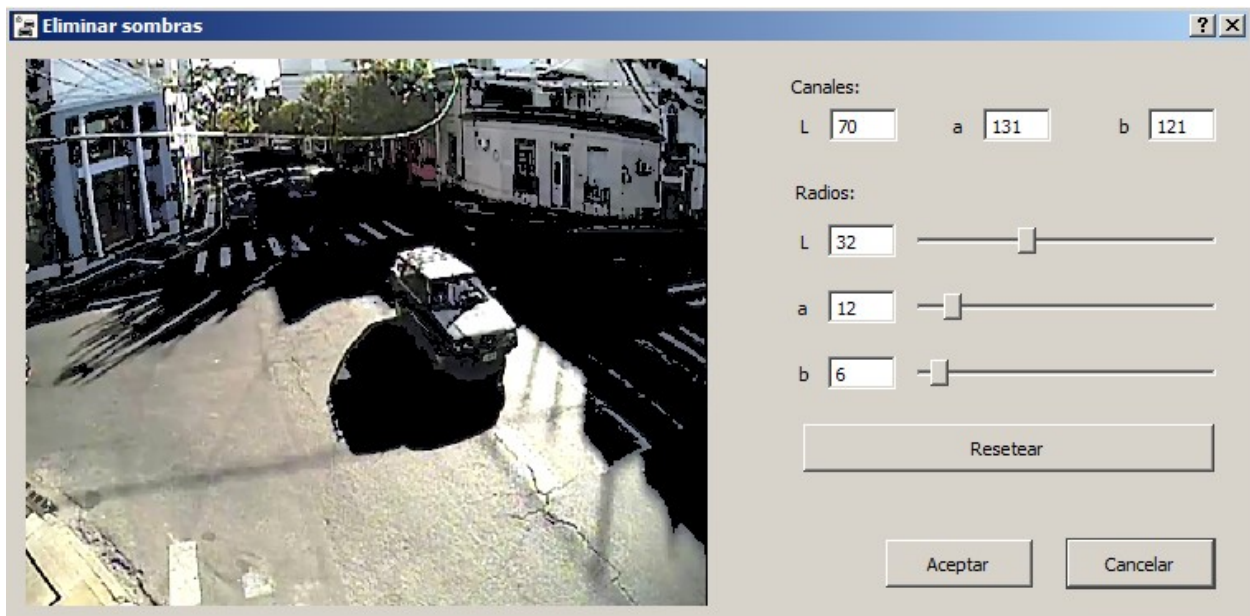


Figura 4.7: Captura del sistema para eliminación de sombras.

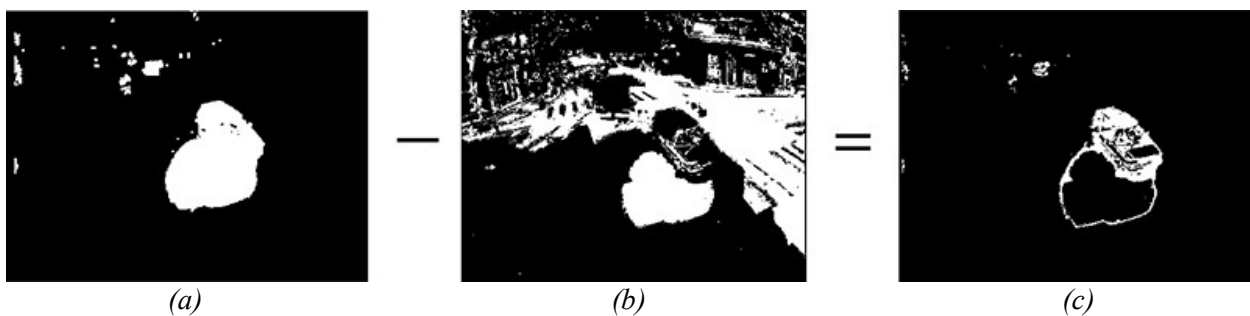


Figura 4.8: Ejemplo de eliminación de sombras. (a): máscara de segmentación, (b) máscara con sombras detectadas., (c): máscara resultante de la diferencia.

Para ello, se utiliza una función de la librería OpenCV que recibe como parámetro una imagen binaria y detecta todas las regiones de píxeles conexos presentes en la misma. Además, se extrae un conjunto de características de cada región, de las cuales solo se utilizarán el área (expresada en cantidad de píxeles que conforman la misma), y el centro geométrico.

Es importante que la máscara binaria que recibe como entrada el bloque de representación, no presente demasiado ruido. El motivo principal, es que la función utilizada aquí detecta cualquier región de píxeles conexos, sin importar el tamaño de la misma. Por lo tanto, detectará cada pixel de ruido como una región independiente y calculará las características del mismo, generando un costo computacional mayor al esperado.

Validación de centros geométricos

La validación de centros geométricos consiste en identificar cuales de los objetos detectados se corresponden con vehículos, y cuales no. Además se verifica que los objetos identificados como vehículos tengan una posición válida para el sistema.

Como se mencionó anteriormente, durante la detección se pueden capturar en algunos casos objetos que no son vehículos. Estos objetos poseen un tamaño menor al establecido para los vehículos, y se corresponden en su mayoría a ruido en la máscara binaria. Para eliminar estos objetos, se analiza el conjunto de objetos detectados y se verifica que el área de los mismos supere el valor de área mínimo establecido por el usuario en los parámetros iniciales. Los objetos que no superan el límite impuesto, son descartados por el sistema.

Paralelo a la validación de áreas, el sistema verifica que la posición del centro geométrico de los objetos se encuentre dentro de los límites definidos por la ROI. Los objetos que se encuentren fuera de los límites, son descartados.

Posteriormente, el sistema verifica la proximidad entre los centros geométricos detectados. En el caso de que dos centros se encuentren muy cercanos, el sistema interpretará que ambos centros pertenecen a un mismo vehículo, uniendo ambos objetos en uno solo cuyo centro geométrico será el promedio de ambos centros y el área será la suma de las dos áreas. Este caso se produce cuando un vehículo queda dividido en la máscara binaria, producto de la segmentación.

Finalmente, los vehículos que superaron las validaciones anteriores son cargados en el sistema como un objeto de tipo *Vehiculo*, el cual contiene los siguientes parámetros:

- Point *pos*: posición actual del vehículo.
- Point *pos_old*: posición del vehículo en el fotograma anterior.
- Point *prev*: posición de la oclusión ocurrida con otro vehículo en el fotograma anterior.
- bool *contar*: bandera que indica si el vehículo debe ser registrado en los contadores del sistema.
- bool *modificado*: bandera que indica si el objeto tuvo una correspondencia.

- `int area`: área del vehículo en píxeles.
- `vector<int> correspondencias`: vector que almacena las correspondencias detectadas.

Solo se cargan los parámetros *pos* y *area*, con la posición del centro geométrico y el área del vehículo respectivamente.

4.2.3 Módulo de seguimiento

El módulo de seguimiento recibe como entrada un conjunto de características correspondientes a cada vehículo detectado en el módulo anterior, y se encarga de establecer las correspondencias con los vehículos detectados en el fotograma previo.

Seguimiento de vehículos

Dado que un video es una sucesión de fotogramas, un vehículo que circule por la escena del mismo aparecerá en un conjunto de fotogramas consecutivos, en posiciones diferentes, hasta que abandone la misma. Por lo tanto, para determinar la trayectoria de un vehículo se deben detectar las posiciones en las que el mismo aparece en cada fotograma.

En esta instancia, el sistema trabaja con dos arreglos que contienen objetos de tipo *Vehículo*: *v_old* y *v_new*. El arreglo *v_new* contiene todos los vehículos detectados en un determinado fotograma, que superaron las validaciones de centros. El arreglo *v_old* contiene los vehículos detectados en el fotograma previo al de *v_new*.

El seguimiento de vehículos se realiza mediante la búsqueda de correspondencias entre los objetos de ambos arreglos. El sistema calcula las distancias euclídeas entre los centros de los objetos de ambos arreglos, y guarda las correspondencias de los centros más cercanos. Se utiliza un límite máximo de distancia, definido por el usuario en los parámetros iniciales, para evitar generar correspondencias erróneas.

El sistema actualiza los objetos de *v_new* que tuvieron una sola correspondencia, cargando en *pos_old* la posición del objeto con el cual se corresponde, y en *area* el promedio ponderado de las áreas de ambos objetos, dándole mayor peso al área del objeto de *v_new*. Además, se modifica el valor de la bandera *modificado* del objeto de *v_old*, para indicar que el mismo tuvo una correspondencia.

Todos los objetos de *v_new* que no tuvieron correspondencias, son considerados por el sistema como vehículos nuevos dentro de la ROI y por lo tanto no se actualiza ningún parámetro. Los objetos de *v_old* que no tuvieron correspondencias, son considerados por el sistema como vehículos que abandonaron la ROI, y se analiza posteriormente si los mismos deben o no ser registrados por los contadores. Finalmente, para los objetos que tengan más de una correspondencia, el sistema interpreta que existe una oclusión entre vehículos.

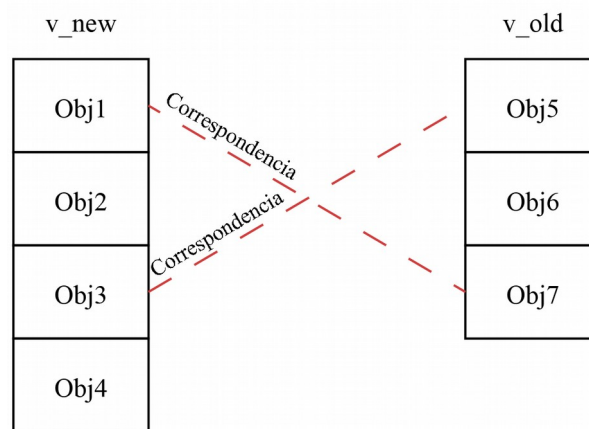


Figura 4.9: Ejemplo de correspondencia entre los arreglos.

En la Figura 4.9 se muestra un ejemplo de correspondencias entre los dos arreglos. Como se puede ver, los objetos *Obj1* y *Obj3* se corresponden con los objetos *Obj7* y *Obj5* respectivamente, por lo que se actualizan de la siguiente manera:

$$\text{Obj1} \rightarrow \text{pos_old} = \text{Obj7} \rightarrow \text{pos}$$

$$\text{Obj1} \rightarrow \text{area} = ((2 * \text{Obj1} \rightarrow \text{area}) + \text{Obj7} \rightarrow \text{area}) / 3$$

$$\text{Obj3} \rightarrow \text{pos_old} = \text{Obj5} \rightarrow \text{pos}$$

$$\text{Obj3} \rightarrow \text{area} = ((2 * \text{Obj3} \rightarrow \text{area}) + \text{Obj5} \rightarrow \text{area}) / 3$$

Los objetos *Obj2* y *Obj4* son considerados como vehículos nuevos dentro de la ROI y no sufren actualizaciones. Por último, el objeto *Obj6* no tuvo ninguna correspondencia y posteriormente el sistema analizará si debe ser registrado o no.

Resolución de oclusiones parciales

Una oclusión se produce cuando dos o más vehículos circulan muy cercanos dentro de la escena de un video, provocando que el sistema los detecte como un único vehículo. Se considera que la oclusión es parcial, cuando la misma se produce durante un trayecto corto dentro de la ROI establecida. La posición de la cámara juega un papel fundamental, ya que determina como se verán los vehículos en la escena. En la Figura 4.10 se muestra un ejemplo en el que la cámara no posee una buena posición, provocando que los vehículos *v1* y *v2* se ocluyan.

Se detectaron tres casos de oclusiones que pueden darse dentro de la escena de un video, donde las causas principales son la inadecuada posición de la cámara, y la similitud en la velocidad, dirección y sentido en la que circulan los vehículos. El primer caso se produce cuando dos o más vehículos ingresan a la ROI ocluidos entre sí, siendo detectados por el sistema como un único vehículo, y luego de recorrer un corto trayecto el sistema los detecta como vehículos independientes. En este caso, el sistema asigna a cada vehículo independiente el centro geométrico de la oclusión como posición en el fotograma previo.



Figura 4.10: Ejemplo de posición inadecuada de la cámara.

En la Figura 4.11 se presenta un ejemplo del primer caso de oclusión parcial. En la parte superior de la imagen, se puede ver un automóvil y un camión que ingresan a la ROI ocluidos, provocando que luego del proceso de segmentación ambos sean detectados por el sistema como una única región de píxeles con centro geométrico $P1$. En la parte inferior de la imagen, se presenta el fotograma inmediato siguiente en el cual los vehículos ya no se ocluyen, por lo que luego del proceso de segmentación, el sistema los detecta como regiones independientes con centros $P2$ y $P3$. Al aplicar el proceso de seguimiento, donde los vehículos detectados en el primer ($obj1$) y segundo ($obj2$ y $obj3$) fotograma se encuentran en v_old y v_new respectivamente, ambos objetos $obj2$ y $obj3$ quedarán correspondidos con $obj1$. Cuando el sistema actualice el arreglo v_new , ambos objetos tendrán como posición previa la posición del objeto $obj1$.

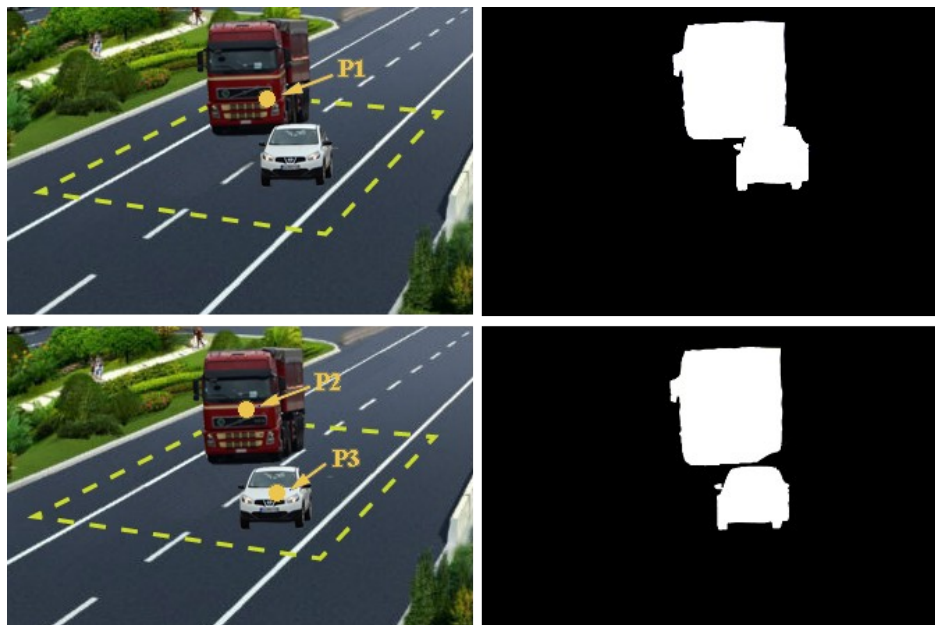


Figura 4.11: Ejemplo del primer caso de oclusión.

El segundo caso de oclusión parcial se produce cuando dos o más vehículos ingresan a la ROI sin ocluirse entre ellos, y luego de recorrer un corto trayecto, se ocluyen provocando que el sistema los detecte como un único vehículo. Para resolver este caso, el sistema reemplazará el centro geométrico correspondiente a la oclusión por centros resultantes de aplicar proyecciones lineales sobre los centros geométricos, correspondientes a los vehículos ocluidos, que fueron detectados en el fotograma previo.

En la Figura 4.12 se presenta un ejemplo del segundo caso. En la parte superior de la imagen, se pueden ver dos automóviles circulando dentro de la ROI sin ocluirse entre ellos, por lo que luego del proceso de segmentación, el sistema detecta ambos vehículos como regiones independientes con centros $P1$ y $P2$. En la parte inferior de la imagen, se presenta el fotograma inmediato siguiente en el cual los vehículos se ocluyen, siendo detectados por el sistema como una única región con centro $P3$. Al aplicar el proceso de seguimiento, donde los vehículos detectados en el primer ($obj1$ y $obj2$) y segundo ($obj3$) fotograma se encuentran en v_old y v_new respectivamente, el objeto $obj3$ quedará correspondido con los objetos $obj1$ y $obj2$.

Posteriormente el sistema chequea la cantidad de correspondencias obtenidas para cada objeto de v_new , detectando que el objeto $obj3$ posee mas de una correspondencia. Esto es interpretado por el sistema como una oclusión, y para resolverla aplica sobre cada objeto correspondido el siguiente procesamiento:

- Se verifica si el parámetro $prev$ contiene una posición válida, para determinar si el objeto formó parte de una oclusión en el fotograma anterior.
- En caso positivo, se calcula la diferencia entre las coordenadas de los puntos $prev$ y $P3$.
- En caso negativo, se calcula la diferencia entre las coordenadas de los puntos pos y pos_old del objeto correspondido.

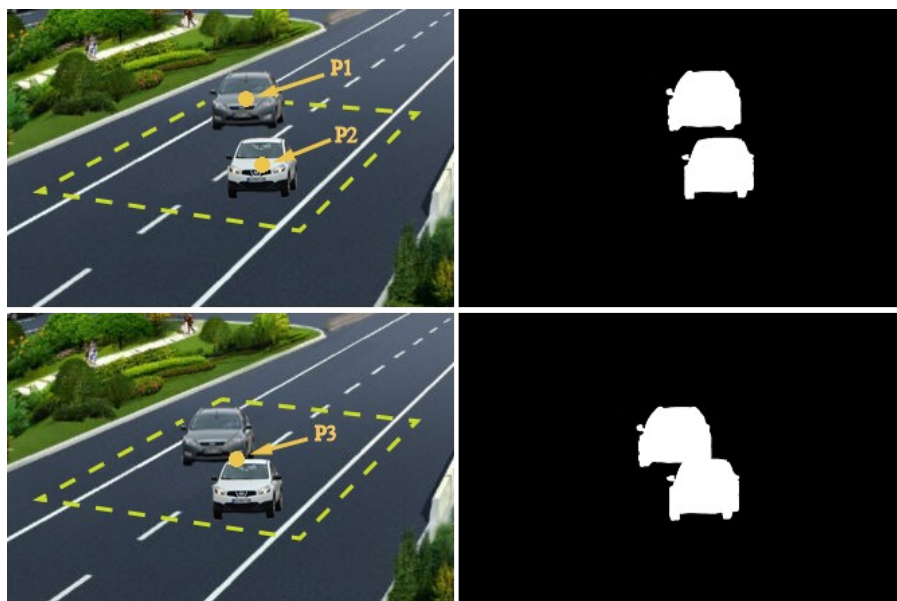


Figura 4.12: Ejemplo del segundo caso de oclusión.

- Se calcula una proyección lineal del punto pos del objeto correspondido, sumándole a este último la diferencia calculada.
- Se crea un nuevo objeto de tipo *Vehiculo* al cual se le asigna la proyección calculada como posición actual, la posición del objeto correspondido como posición previa, el área del objeto correspondido, y en el parámetro $prev$ se le asigna la posición de la oclusión ($obj3.pos$).
- El objeto $obj4$ es cargado en el arreglo v_new como un nuevo vehículo en la escena.
- Al finalizar el procesamiento de los objetos correspondidos, el objeto $obj3$ es eliminado del arreglo v_new .

En la Figura 4.13 se muestra un ejemplo de como se realizan las proyecciones de centros para resolver la oclusión.

Finalmente, el tercer caso de oclusión se produce cuando dos o más vehículos ingresan a la ROI ocluidos, y durante todo el trayecto recorrido dentro de la misma permanecen ocluidos. Para este caso particular, el sistema no logra identificar la existencia de una oclusión, y por lo tanto los vehículos ocluidos serán interpretados como un único vehículo dentro de la escena.

4.2.4 Módulo de conteo

El módulo de conteo recibe como entrada el arreglo v_old , y se encarga de determinar que vehículos del mismo deben ser clasificados y registrados.

Límite de conteo

Para determinar que vehículos deben ser registrados el sistema recorre el arreglo v_old , y para cada objeto analiza el estado del parámetro $modificado$. Si $modificado$ posee un valor “verdadero”, significa que ese vehículo tuvo una correspondencia con un vehículo del arreglo v_new , y por lo tanto es un vehículo que aún continúa dentro de la ROI y no debe ser registrado.

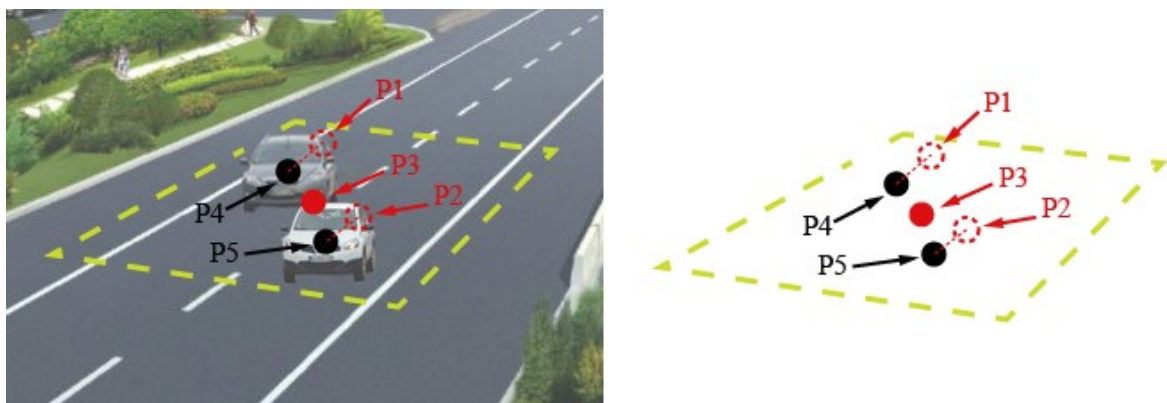


Figura 4.13: Ejemplo de resolución de oclusión.

Si *modificado* posee un valor “falso”, significa que ese vehículo no tuvo correspondencia alguna con los vehículos de *v_new*. Para este caso, el sistema realiza una proyección lineal del centro geométrico correspondiente al vehículo, y verifica si la misma abandona la ROI definida. En caso positivo, el vehículo es marcado para ser registrado modificando el valor de su parámetro *contar* a “verdadero”.

Actualización de datos

Para llevar a cabo el registro de vehículos que abandonan la ROI, se utiliza un conjunto de contadores. Los vehículos son clasificados según el tamaño de sus áreas en: Bicicletas/Motocicletas, Autos/Camionetas y Camiones/Colectivos. Los tamaños de clasificación son definidos por el usuario en los parámetros iniciales.

El sistema recorre el arreglo *v_old* y para cada objeto analiza el estado del parámetro *contar*. Si *contar* posee un valor “verdadero”, se realiza la clasificación del vehículo comparando el tamaño de su área con los valores preestablecidos. Una vez clasificado el vehículo, se actualiza el contador correspondiente a la clase. Al finalizar el procesamiento, el arreglo *v_old* es eliminado y el arreglo *v_new* es utilizado como arreglo *v_old* en la siguiente iteración.

El usuario define inicialmente un período de medición, en minutos, durante el cual se registran los vehículos y se actualizan los contadores. Una vez transcurrido el tiempo definido, el sistema reinicia los contadores y comienza nuevamente a registrar vehículos. En un desarrollo futuro, los datos almacenados en los contadores serán registrados en una base de datos junto con otro tipo de información como por ejemplo fecha, hora, lugar de medición, etc.

4.3 Interfaz de usuario

Como complemento y mas allá del alcance del presente Proyecto Final de Carrera, se llevo a cabo el desarrollo de una interfaz gráfica de usuario. Los principales objetivos buscados fueron facilitar la interacción del usuario con el sistema y permitir la visualización en tiempo real del procesamiento realizado.

Para el desarrollo de la interfaz se utilizaron las siguientes herramientas de software:

- **Qt Creator**: es un IDE (entorno de desarrollo integrado) multiplataforma para el desarrollo de aplicaciones en C++ con las bibliotecas Qt. Posee una licencia LGPL y esta ampliamente documentado. Se pueden encontrar muchos ejemplos en la web y existe una gran comunidad online donde se puede encontrar información útil y consultar cualquier problema o duda.
- **Bibliotecas Qt**: es un framework multiplataforma orientado a objetos, desarrollado como software libre y de código abierto a través de Qt Project, donde participa tanto la comunidad, como desarrolladores de Nokia, Digia y otras empresas. Posee licencias GNU LGPL 2.1 y GNU GPL 2 y 3.

4.3.1 Consideraciones en el diseño

Inicialmente el sistema fue desarrollado para ser utilizado como una aplicación de consola. Los parámetros eran configurados previo a la ejecución, y el sistema realizaba todo el procesamiento dentro de un bucle general, donde por cada iteración se extraía un fotograma del video.

Para el desarrollo de una aplicación con GUI, la cual posee un mecanismo de gestión de eventos diferente al de una aplicación de consola, fue necesario cambiar la estructura de código utilizada. El bucle general de procesamiento no puede implementarse directamente ya que dificultaría la gestión de los eventos de la interfaz, provocando que los componentes de esta última no funcionen correctamente. En este contexto, se diseñó una solución que presenta las siguientes características:

- Se utiliza una clase *mainwindow* que se encarga de gestionar todos los eventos correspondientes a los componentes de la interfaz. Dentro de esta clase se crea un objeto de tipo *tracking*, el cual se encarga de todo el procesamiento de video.
- En Qt se utilizan señales y slots para la gestión de eventos: una señal es emitida cuando un evento ocurre; un slot es una función que se llama en respuesta a una señal particular. Los eventos producidos por los componentes de la interfaz son gestionados mediante señales y slots definidos por defecto. Sin embargo, se modificaron algunos slots para que realicen funcionalidades específicas.
- Para gestionar la extracción de fotogramas se crea una señal y un slot particular. Cuando la interfaz genera el evento correspondiente a iniciar el procesamiento del video, la clase *tracking* comienza con el ciclo de extracción de fotogramas en un hilo de ejecución secundario. Se extrae un fotograma, se aplica el conjunto de procesamientos correspondiente, y al finalizar se emite una señal que llama a un slot dentro de la clase *mainwindow*, el cual indica que el procesamiento del fotograma ha finalizado. Tanto la señal como el slot contienen el fotograma procesado como parámetro.
- El procesamiento del video puede ser interrumpido y reanudado por el usuario. Esto se lleva a cabo a través de eventos que modifican banderas de control dentro del bucle de procesamiento de la clase *tracking*.

4.3.2 Componentes de la interfaz

En la Figura 4.14 se presenta una captura de la interfaz gráfica de usuario desarrollada. Se distinguen en cuadros de colores los diferentes paneles que presenta la interfaz:

- Panel de video (azul).
- Panel de configuración (rojo).
- Panel de visualización (amarillo).
- Panel de resultados (verde).

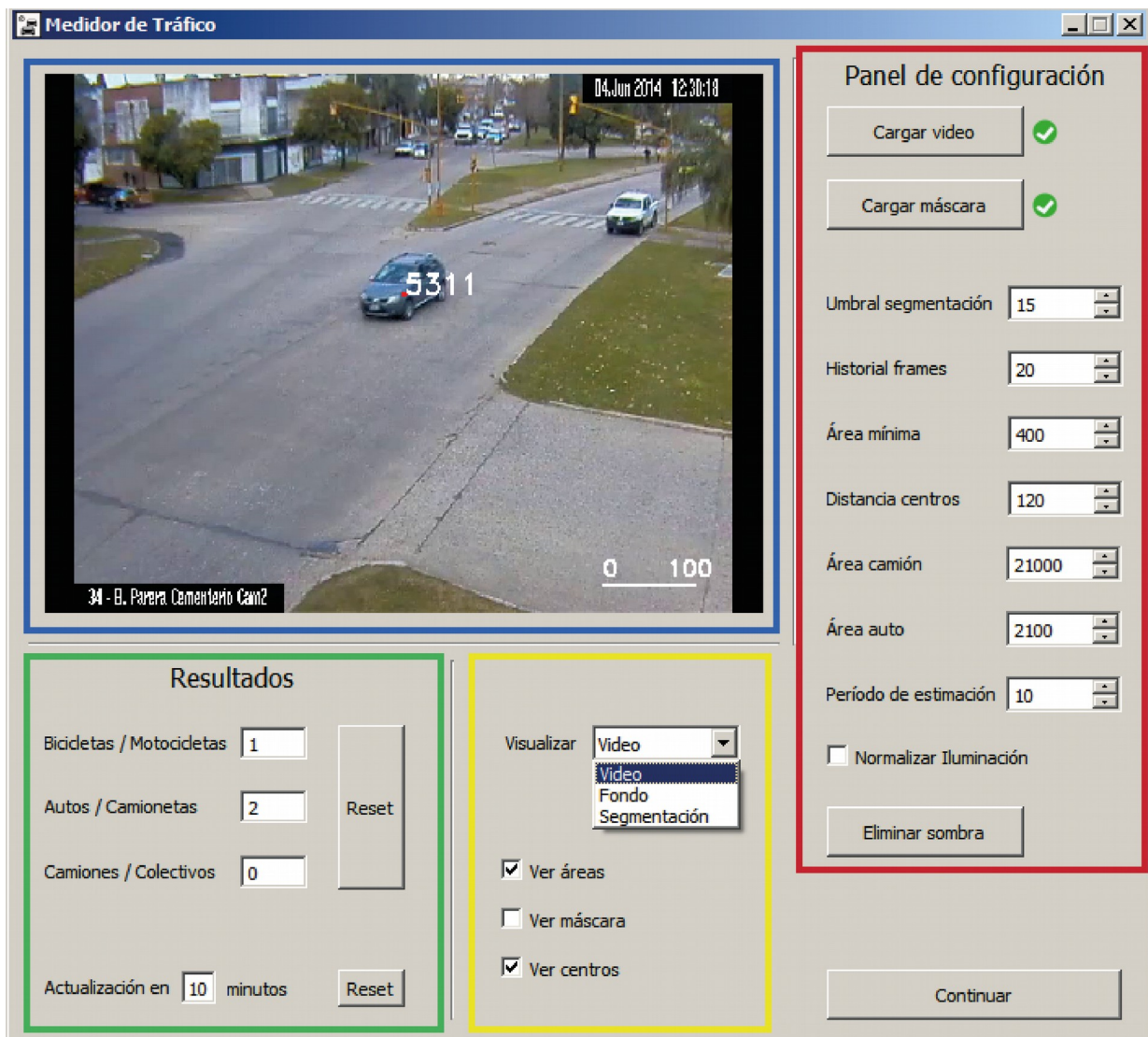


Figura 4.14: Interfaz gráfica de usuario

El panel de video cumple la función de mostrar los fotogramas del video una vez finalizado el procesamiento aplicado a los mismos. Adicionalmente se puede mostrar en este panel el modelo del fondo generado o la máscara resultante de la segmentación, y ver en tiempo real como los mismos van cambiando. En la parte inferior derecha se muestra una pequeña medida de referencia que indica el tamaño de 100 píxeles, la cual puede ser de utilidad para el usuario a la hora de definir los parámetros del sistema.

El panel de configuración le permite al usuario definir parámetros del sistema:

- El botón “Cargar video” es utilizado para cargar en el sistema el video de tráfico a procesar. Si la carga fue exitosa, un icono verde aparecerá a la derecha del botón. De lo contrario aparecerá un icono rojo y el usuario deberá realizar la carga nuevamente.

- El botón “Cargar máscara” es utilizado para cargar la máscara binaria, generada previamente por el usuario, que define la ROI de detección. Si la carga fue exitosa, un icono verde aparecerá a la derecha del botón. De lo contrario aparecerá un icono rojo y el usuario deberá realizar la carga nuevamente.
- El parámetro “Umbral segmentación” define el umbral utilizado para la segmentación de vehículos.
- El parámetro “Historial frames” define la cantidad de fotogramas previos que serán utilizados para generar el modelo del fondo.
- El parámetro “Área mínima” define el menor área que debe tener un vehículo para ser detectado por el sistema.
- El parámetro “Distancia centros” define la distancia máxima que puede haber entre centros geométricos de vehículos, para que una correspondencia sea válida.
- El parámetro “Área camión” define el área mínima que debe poseer un vehículo para ser registrado como camión/colectivo.
- El parámetro “Área auto” define el área mínima que debe poseer un vehículo para ser registrado como auto/camioneta.
- El parámetro “Período de estimación” define la cantidad de minutos que durará el registro de vehículos. Pasado ese tiempo, los contadores se reiniciarán y se continuará con el registro de vehículos.
- El checkbox “Normalizar iluminación” indica si debe aplicarse o no, la normalización de iluminación sobre la escena del video.
- El botón “Eliminar sombra” puede ser utilizado opcionalmente para eliminar las sobras de los vehículos. Al presionar el botón se abre una ventana emergente (ver Figura 4.7), dentro de la cual se puede ver a la izquierda una captura del fotograma que esta siendo procesado, y a la derecha un pequeño panel donde el usuario define valores para la segmentación de color.

El panel de visualización es utilizado por el usuario para seleccionar la información que quiere ver en el panel de video:

- Se puede alternar entre ver el video original, la actualización del modelo de fondo o la máscara resultante de la segmentación de vehículos.
- Visualización del área de cada vehículo detectado.
- Visualización del centro geométrico de cada vehículo detectado.
- Visualización de la ROI definida.

El panel de resultados muestra las mediciones parciales realizadas por el sistema, para cada clasificación de vehículo. Además, en la parte inferior se muestra la cantidad de minutos restantes para la finalización del período de medición. Se agregaron dos botones “reset”, para reiniciar los contadores y/o el período de medición manualmente.

Finalmente el botón “Continuar”, el cual cambia de estado a “Pausar”, es utilizado por el usuario para detener/continuar el procesamiento del video. Esta funcionalidad es de suma utilidad, ya que en cualquier momento el usuario puede pausar el procesamiento y visualizar los resultados parciales de sus diferentes etapas.

Experimentación y resultados

Para evaluar el desempeño del sistema desarrollado, se diseñaron casos de prueba a partir de los videos que conforman la base de datos. Cada caso de prueba consiste en un fragmento de video con una duración aproximada de diez minutos, para el cual:

- Se define una ROI dentro de la escena.
- Se realiza una configuración adecuada de los parámetros del sistema.
- Se lleva a cabo el conteo y clasificación de vehículos mediante observación.

Para obtener una mejor estimación de los resultados, se utilizaron dos fragmentos de diez minutos por cada video de la base de datos, dando un total de ocho mediciones realizadas. Se optó por trabajar con fragmentos de 10 minutos de duración, para que las tareas de control y verificación de resultados no sea tediosa y requiera demasiado tiempo. Para seleccionar los fragmentos, se realizó previamente un análisis de cada video para determinar los instantes en los que circulaba una mayor cantidad de vehículos, y así poder realizar un número mayor de registros. En la Figura 5.1 se muestran las capturas de los videos utilizados para cada caso de prueba.

5.1 Configuración de parámetros

La configuración de los parámetros del sistema se llevó a cabo mediante un proceso iterativo de prueba y corrección. Un factor importante a tener en cuenta a la hora de definir valores de parámetros, es la posición en la que se encuentra la cámara de video:

- La posición define la dirección y sentido en la que circulan los vehículos.
- La distancia entre la cámara y los vehículos altera el tamaño con el que se capturan estos últimos.

En primer lugar, se configura el parámetro correspondiente a la dirección y sentido con la que circulan los vehículos en la escena, el cual consiste en un par de valores que son utilizados para realizar las proyecciones lineales de los centros geométricos de cada vehículo. Este es un parámetro interno del sistema y debe ser configurado dentro del código fuente, previo a ejecutar la aplicación.

Luego, se realiza la configuración del umbral utilizado para la segmentación de vehículos y el historial de fotogramas utilizados para generar el modelo del fondo. Ambos parámetros se configuran en simultaneo debido a que están involucrados dentro del proceso de segmentación.



Figura 5.1: Capturas de los videos utilizados para cada caso de prueba.

El umbral se inicializa con un valor de 30, y el historial se inicializa con una cantidad de 100 fotogramas. Posteriormente, luego de analizar la máscara resultante de la segmentación, ambos parámetros van siendo ajustados según corresponda, utilizando como ayuda las opciones “Visualizar: video, fondo o segmentación” en la interfaz.

Los siguientes parámetros a configurar son los tamaños de las áreas: área mínima, área para autos/camionetas y área para camiones/colectivos. Para configurar estos parámetros se activa la opción “ver áreas” en la interfaz, se corre el video y se realiza un registro de las áreas correspondientes a cada clasificación de vehículo. Finalmente se analizan los rangos de áreas obtenidos, y se definen valores para cada parámetro.

Finalmente, se define la distancia máxima que puede haber entre centros de vehículos y la duración del período de medición. El parámetro de distancia se inicializa con un valor de 100, tomando como referencia la regla ubicada en la parte inferior del panel de video, y luego de realizar algunas pruebas puede ser ajustado utilizando como ayuda la opción “ver centros” en la interfaz. El período de medición se establece en 10 minutos, debido a que ese tiempo corresponde a la duración total de los videos de cada caso de prueba.

En el caso de que sea necesario aplicar una normalización sobre los niveles de iluminación de la escena del video, se habilita la opción “Normalizar Iluminación” en la interfaz. Luego se observan los resultados obtenidos de la segmentación, y se analiza si es conveniente o no habilitar esta opción.

Si el proceso de segmentación se ve afectado por la presencia de sombras, se utiliza el botón “Eliminar sombra” en la interfaz, y mediante observación se seleccionan los valores adecuados para cada parámetro.

5.2 Casos de prueba

Se describen a continuación las características correspondientes a cada caso de prueba.

Caso de prueba 1:

- *Presencia de obstáculos*: la escena de video no presenta obstáculos que pudieran dificultar la detección de vehículos.

- *Condiciones climáticas*: las condiciones climáticas son favorables y no generan dificultades durante el procesamiento.
- *Cambios de iluminación*: la escena presenta un nivel de iluminación adecuado y homogéneo. No se detectan grandes cambios de iluminación.
- *Posición de la cámara*: la cámara se encuentra en una posición elevada y captura el tráfico de manera diagonal-frontal.
- *Presencia de sombras*: se detectan sombras en la parte inferior de los vehículos que circulan. Se remueven utilizando eliminación de sombras.
- *Oclusiones parciales*: se producen oclusiones parciales debido a la similitud en la velocidad y dirección de los vehículos, y a la posición en diagonal de la cámara.
- *Horario aproximado de grabación*: 12:30hs.

Caso de prueba 2:

- *Presencia de obstáculos*: la escena presenta vehículos que circulan en diferentes direcciones y sentidos. Para este caso de prueba es de interés registrar los vehículos que circulan por la zona inferior de la escena, en dirección horizontal. Por lo tanto, se definió estratégicamente la ROI para que el resto de los vehículos no afecte el proceso de detección.
- *Condiciones climáticas*: el cielo se encuentra medianamente nublado, por lo tanto la iluminación general de la escena no es buena. Para mejorar el contraste se aplica una normalización en los niveles de iluminación.
- *Cambios de iluminación*: la escena presenta una iluminación homogénea y de bajo nivel. No se detectan grandes cambios de iluminación.
- *Posición de la cámara*: la cámara se encuentra medianamente elevada y captura el tráfico de manera perpendicular a la dirección de circulación.
- *Presencia de sombras*: se detectan pequeñas sombras en la parte inferior de los vehículos. Dado que las mismas no afectan el procesamiento de segmentación, no son eliminadas.
- *Oclusiones parciales*: se producen oclusiones parciales entre vehículos. Las causas principales son la posición de la cámara y la similitud en la velocidad de los vehículos.
- *Horario aproximado de grabación*: 12:30hs.

Caso de prueba 3:

- *Presencia de obstáculos*: la escena presenta vehículos que circulan en diferentes direcciones. Para este caso de prueba es de interés registrar los vehículos que circulan por la zona central de la escena, en dirección transversal. Si bien se definió una ROI para detectar estos vehículos, el resto de los vehículos que circulan en otra dirección también pasan por la

misma. Por lo tanto, si bien el sistema detectará todos los vehículos que circulen dentro de la ROI, posteriormente se descartan de los registros los vehículos que no son de interés contabilizar.

- *Condiciones climáticas*: las condiciones climáticas son favorables y no generan dificultades durante el procesamiento.
- *Cambios de iluminación*: la intensa luz solar provoca que la escena tenga zonas con altas variaciones en los niveles de iluminación. Debido a que estas variaciones cambian en un lapso corto de tiempo, se redujo el tamaño de historial de fotogramas utilizados para generar el modelo del fondo, para que este último se adapte rápidamente a los cambios.
- *Posición de la cámara*: la cámara se encuentra medianamente elevada y captura el tráfico de manera diagonal-frontal.
- *Presencia de sombras*: se detectan sombras de gran tamaño, producidas por el ángulo con el que incide la luz solar sobre la escena. Se remueven parcialmente utilizando eliminación de sombras.
- *Oclusiones parciales*: se producen oclusiones parciales entre vehículos debido a la posición de la cámara y la similitud en la velocidad de los vehículos.
- *Horario aproximado de grabación*: 15:00hs.

Caso de prueba 4:

- *Presencia de obstáculos*: la escena presenta vehículos que circulan en diferentes direcciones y peatones que circulan por la senda peatonal. Para este caso de prueba es de interés registrar los vehículos que circulan por la zona central de la escena, en dirección transversal. Si bien se definió una ROI para detectar estos vehículos, algunos de los peatones que se encuentra circulando también pasan por la misma. Por lo tanto, los peatones que sean detectados por el sistema, serán considerados como falsos positivos registrados.
- *Condiciones climáticas*: las condiciones climáticas son favorables y no generan dificultades durante el procesamiento.
- *Cambios de iluminación*: la escena presenta una iluminación homogénea y de nivel medio. No se detectan grandes cambios de iluminación.
- *Posición de la cámara*: la cámara se encuentra medianamente elevada y captura el tráfico de manera diagonal-posterior.
- *Presencia de sombras*: se detectan sombras en la parte inferior de los vehículos que circulan. Se remueven utilizando eliminación de sombras.
- *Oclusiones parciales*: se producen oclusiones parciales entre vehículos debido a la posición de la cámara y la similitud en la velocidad de los vehículos.
- *Horario aproximado de grabación*: 10:00hs.

5.3 Análisis de resultados

Previo a comenzar con las pruebas del sistema, se llevo a cabo un registro de la cantidad y tipo de vehículos que circularon por la ROI definida en cada caso de prueba. Para obtener esta información, se recurrió a la observación de los videos y al registro manual de vehículos. Estos valores serán utilizados para determinar la efectividad de las mediciones realizadas por el sistema.

Adicionalmente fue necesario realizar un control, nuevamente mediante observación, de las mediciones realizadas por el sistema para determinar los errores cometidos:

- Falso positivo: para cada categoría de clasificación, un falso positivo se produce cuando el sistema registra dentro de la misma un vehículo que corresponde a otra categoría.
- Falso negativo: para cada categoría de clasificación, un falso negativo se produce cuando el sistema no registra un vehículo correspondiente a la misma, o registra el vehículo dentro de otra categoría.

Por ejemplo, una motocicleta que circuló dentro de la ROI y fue registrada como un automóvil, será un falso negativo para la categoría “motocicletas” y será un falso positivo para la categoría “autos”. En el caso de que la motocicleta no fuese registrada por el sistema, directamente se convertiría en un falso negativo para la categoría “motocicletas”.

Las mediciones y controles realizados mediante observación, se llevaron a cabo dos y hasta tres veces para cada caso, con la finalidad de minimizar la cantidad de errores cometidos. En la Tabla 5.1 se detallan los valores configurados para cada parámetro del sistema en los cuatro casos de prueba.

Tabla 5.1: Valores de los parámetros del sistema para cada caso de prueba

Parámetro	Caso 1	Caso 2	Caso 3	Caso 4
Umbral segmentación	15	25	20	30
Historial frames	20	200	20	100
Área mínima	400	400	1500	1500
Distancia centros	120	100	100	100
Área camión	21000	17500	75000	30000
Área auto	2100	4000	7000	2500
Período estimación	10	10	10	10
Eliminar sombras	si	no	si	si
Normalizar iluminación	no	si	no	no

5.3.1 Efectividad en la detección

En esta sección se exponen los resultados producidos por el sistema luego de procesar los videos correspondientes a los casos de prueba. Para la presentación de resultados, se unieron las mediciones realizadas sobre cada par de fragmentos pertenecientes al mismo video. Los vehículos fueron clasificados según su tamaño en tres categorías: bicicletas/motocicletas, autos/camionetas y camiones/colectivos.

Para cada categoría de clasificación se detallará la siguiente información:

- Cantidad de vehículos medida mediante observación (Cm).
- Cantidad de vehículos detectada por el sistema (Cd).
- Cantidad de falsos positivos (Fp).
- Cantidad de falsos negativos (Fn).
- Porcentaje de falsos positivos ($\%Fp$), el cual se calcula como $\%Fp = Fp / Cd$.
- Porcentaje de falsos negativos ($\%Fn$), el cual se calcula como $\%Fn = Fn / Cm$.
- Porcentaje de detección ($\%D$), el cual se calcula como $\%D = (Cd - Fp) / Cm$.

En la Tabla 5.2 se presentan los resultados obtenidos para los fragmentos de video correspondientes al primer caso de prueba. Para vehículos de pequeño y mediano porte, el porcentaje de detección supera el 85% y los porcentajes de falsos positivos y falsos negativos no superan el 15%. Dado que el porcentaje de detección es elevados y los porcentajes de falsos positivos y falsos negativos son bajos, se puede concluir que la medición realizada por el sistema fue buena.

Por otro lado, para vehículos de gran porte el porcentaje de detección apenas logra superar el 60% y el porcentaje de falsos negativos supera el 35%. La reducida cantidad de vehículos correspondientes a esta categoría que circularon durante el video, provoca que ante pocos errores los porcentajes resultantes crezcan rápidamente. Las principal causa de errores fue la diferencia de tamaño entre los vehículos detectados en cada carril de la calle. Por ejemplo, el tamaño de una camioneta o trafico detectada en el carril derecho, era similar o superior al tamaño de un camión pequeño o sin acoplado detectado en el carril izquierdo. Por lo tanto se concluye que la detección no fue buena.

Tabla 5.2: Resultados correspondientes a los fragmentos del primer caso de prueba.

Clasificación	Cm	Cd	Fp	Fn	$\%Fp$	$\%Fn$	$\%D$
Motos/bicis	108	102	6	12	5,88	11,11	88,89
Autos/camionetas	191	186	17	22	9,14	11,52	88,48
Camiones/colectivos	19	13	1	7	7,69	38,84	63,16
Total vehículos	318	301	24	41	7,97	12,89	87,11

En la Tabla 5.3 se presentan los resultados obtenidos para los fragmentos de video correspondientes al segundo caso de prueba. Para vehículos de pequeño porte, el porcentaje de detección supera el 80% y los porcentajes de falsos positivos y falsos negativos no superan el 25%. Las principales causas de errores fueron la oclusión entre vehículos y la incorrecta segmentación de algunos vehículos de mediano y gran porte. Se concluye que la medición realizada por el sistema fue aceptable. Para vehículos de mediano y gran porte, los porcentajes de detección superan el 90% y los porcentajes de falsos positivos y falsos negativos no superan el 20%. Por lo tanto se concluye que la medición realizada por el sistema fue buena.

En la Tabla 5.4 se presentan los resultados obtenidos para los fragmentos de video correspondientes al tercer caso de prueba. Para vehículos de pequeño porte, el porcentaje de detección no supera el 50% y los porcentajes de falsos positivos y falsos negativos rondan entre el 40% y 50%. Las principales causas de errores fueron la oclusión entre vehículos y la eliminación parcial de las sombras producidas por los mismos. Se concluye que la medición realizada por el sistema no fue buena.

Para vehículos de mediano y gran porte, los porcentajes de detección superan el 90% y los porcentajes de falsos positivos y falsos negativos no superan el 10%. Por lo tanto se concluye que la medición realizada por el sistema fue buena.

En las Tabla 5.5 se presentan los resultados obtenidos para los fragmentos de video correspondientes al cuarto caso de prueba. Para vehículos de pequeño porte, los porcentajes de detección no superan el 70% y los porcentajes de falsos positivos y falsos negativos rondan entre el 30% y 42%. Las principales causas de errores fueron la oclusión entre vehículos y la presencia de peatones en la escena. Se concluye que la medición realizada por el sistema no fue buena.

Tabla 5.3: Resultados correspondientes a los fragmentos del segundo caso de prueba.

Clasificación	Cm	Cd	Fp	Fn	%Fp	%Fn	%D
Motos/bicis	52	56	13	9	23,21	17,31	82,69
Autos/camionetas	88	86	4	6	4,65	6,82	93,18
Camiones/colectivos	5	6	1	0	16,67	0,00	100,00
Total vehículos	145	148	18	15	12,16	10,34	89,66

Tabla 5.4: Resultados correspondientes a los fragmentos del tercer caso de prueba.

Clasificación	Cm	Cd	Fp	Fn	%Fp	%Fn	%D
Motos/bicis	64	57	25	32	43,86	50,00	50,00
Autos/camionetas	253	251	15	17	5,98	6,72	93,28
Camiones/colectivos	4	4	0	0	0,00	0,00	100,00
Total vehículos	321	312	40	49	12,82	15,26	84,74

Tabla 5.5: Resultados correspondientes a los fragmentos del cuarto caso de prueba.

Clasificación	Cm	Cd	Fp	Fn	%Fp	%Fn	%D
Motos/bicis	32	36	15	11	41,67	34,38	65,63
Autos/camionetas	201	192	12	21	6,25	10,45	89,55
Camiones/colectivos	8	8	1	1	12,50	12,50	87,50
Total vehículos	241	236	28	33	11,86	11,86	86,31

Para vehículos de mediano y gran porte, los porcentajes de detección superan el 80% y los porcentajes de falsos positivos y falsos negativos no superan el 15%. Por lo tanto se concluye que la medición realizada por el sistema fue aceptable.

5.3.2 Velocidad de ejecución

Un requerimiento fundamental para que el sistema pueda funcionar en tiempo real, es que el procesamiento aplicado sobre los fotogramas se lleve a cabo en un tiempo inferior a la tasa de fotogramas que posea el video a procesar. Dado que este proyecto se llevó a cabo utilizando el hardware disponible, y no uno especializado para tal fin, se estableció con anterioridad que los videos a procesar tendrán una tasa de 10 fotogramas por segundo. Por lo tanto, el procesamiento aplicado sobre cada fotograma deberá realizarse en un tiempo inferior a $1/10 = 0.1$ segundos.

Como se mencionó en capítulos anteriores, todo el procesamiento aplicado sobre los fotogramas se lleva a cabo dentro de un bucle general, donde se realiza la lectura y procesamiento de un fotograma por cada iteración del mismo. Para determinar el tiempo de procesamiento, se mide la duración de cada iteración del bucle general, utilizando variables y funciones del archivo de cabecera *ctime* perteneciente a la librería estándar de C++.

El hardware utilizado posee las siguientes características:

- Microprocesador: Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz (4 CPUs), ~2.1GHz
- Memoria RAM: 4096MB
- Sistema operativo: Windows 7 Ultimate

Para poder realizar una buena estimación del tiempo que conlleva procesar un fotograma, se llevaron a cabo 108 mediciones para cada uno de los cuatro casos de prueba. Posteriormente se realizó un promedio de los tiempos para cada caso individual y un promedio general de todos los tiempos. En la Tabla 5.10 se presentan los resultados obtenidos, donde Pm (*s*) es el promedio de las 108 mediciones en segundos, y $Ptot$ (*s*) es el promedio total de las 432 mediciones en segundos.

Dado que los tiempos de ejecución medidos no superan el valor de la tasa de fotogramas impuesto inicialmente, se puede concluir que el sistema trabaja con normalidad en tiempo real sobre videos de tráfico.

Tabla 5.6: Tiempos de ejecución para instancia única del sistema.

Casos de prueba	Pm (s)	Ptot (s)
Primer caso	0,0285	
Segundo caso	0,0370	0,0314
Tercer caso	0,0292	
Cuarto caso	0,0310	

Una vez realizada la medición de tiempos para una única instancia de ejecución del sistema, se llevó a cabo una medición de tiempos para varias instancias del sistema funcionando en paralelo. En las Tabla 5.7 se presentan los resultados de mediciones realizadas para 2, 4 y 10 instancias del sistema funcionando en paralelo. En este caso, se realizó el promedio de tiempos sobre 60 mediciones realizadas para cada instancia de ejecución. Se comprobó que a medida que el número de instancias en ejecución crece, también lo hace el tiempo de procesamiento necesario para cada fotograma, dificultando que el procesamiento aplicado sobre los videos se realice en tiempo real.

Tabla 5.7: Tiempos de ejecución para instancias múltiples del sistema.

Instancias en ejec.	Pm (s)
2	0,0298
4	0,0564
10	0,1607

Conclusiones y desarrollos futuros

En este proyecto se presenta el desarrollo de un sistema para medición de tráfico, basado en técnicas de análisis y procesamiento de imagen y video. Para llevar a cabo este trabajo, fue necesario aplicar muchos de los conocimientos adquiridos a lo largo de la carrera, realizar actividades de investigación, diseño, desarrollo y prueba.

El sistema desarrollado permite analizar y extraer información de manera automática de largos períodos de registro en video: permite la carga de videos e imágenes, aplica técnicas de preprocesamiento y segmentación para la detección de vehículos, y lleva a cabo la representación, seguimiento y clasificación por tamaño de los vehículos detectados.

En un principio se intentó trabajar adicionalmente con videos de tráfico grabados durante la noche, pero dada la baja calidad de los mismos, sumado a las grandes distorsiones generadas por las luces de los vehículos y sus reflexiones, los métodos implementados no generaban resultados positivos. Por ello, luego de varias pruebas sin resultados alentadores se optó por limitar el sistema a trabajar con videos grabados durante el día.

Se observó que los principales factores que influyeron sobre el desempeño del sistema fueron la posición de la cámara que captura el tráfico, la presencia de sombras producidas por vehículos y la iluminación general de la escena. Debido a que se trabajó sobre videos de tráfico, donde estos factores no pueden alterarse, se intentó que el sistema se adapte lo mejor posible a través de una simple configuración de parámetros.

Para videos de tráfico que presentaron características aceptables, el desempeño del sistema fue bueno, superando el 80% de acierto en las detecciones. Sin embargo, para videos que poseían ciertas características poco favorables el desempeño de sistema fue inferior, principalmente para vehículos de pequeño porte.

En cuanto al rendimiento del sistema, inicialmente se tuvo problemas debido a que algunos métodos demandaban demasiado tiempo de procesamiento, principalmente en el modelado del fondo y la eliminación de sombras. Para solucionar este inconveniente se redujo la tasa de fotogramas de los videos. Los resultados obtenidos de las mediciones de tiempos de procesamiento, indican que la velocidad de ejecución del sistema supera a la tasa de fotogramas que poseen los videos. Por lo tanto el sistema opera con total normalidad en tiempo real sobre videos que posean una tasa de 10 fotogramas por segundo.

Se buscó facilitar la interacción sistema-usuario a través del desarrollo de una GUI, donde el usuario además de realizar configuraciones de parámetros, puede visualizar diferentes etapas del

procesamiento. Esto le permite tener una idea de como esta funcionando cada parte del sistema, y así poder realizar modificaciones y ajustes donde sea necesario.

El sistema desarrollado trabaja actualmente sobre videos de tráfico. Sin embargo, se espera que a futuro el mismo pueda ser adaptado para trabajar con cámaras de video en tiempo real. Esto permitirá tener una mayor libertad en la configuración de algunos factores que son fundamentales para el correcto funcionamiento del sistema, como lo son la posición de la cámara, los niveles de iluminación de la escena o la presencia de obstáculos que dificulten el proceso de medición.

A partir de la utilización de esta herramienta, se podrá disponer de datos en forma rápida y confiable. Estos pueden ser utilizados por entidades gubernamentales para mejorar las políticas aplicadas a la circulación, o para el desarrollo de nuevas herramientas que ayuden a solucionar problemas de congestión vehicular. Además, dado que el sistema fue desarrollado utilizando herramientas de código abierto y sobre plataformas estándares, el mismo podrá ser reproducido fácilmente por cualquier entidad que requiera obtener este tipo de información.

Como trabajos futuros se pueden mencionar:

- Establecer un protocolo para la configuración de las cámaras de video.
- Mejorar el procesamiento utilizado para eliminación de sombras.
- Automatizar la configuración de parámetros.
- Implementar la detección de vehículos utilizando más de una ROI.
- Implementar la medición de velocidad para vehículos.
- Elaborar un manual de usuario, para que el sistema pueda ser utilizado por personas que posean conocimientos mínimos en procesamiento de imágenes y video.

Bibliografía

- [1] Bull, Alberto. Congestión de tránsito: el problema y cómo enfrentarlo. No. 87. United Nations Publications, 2003.
- [2] de Dios Ortuzar, Juan, and Luis G. Willumsen. Modelling transport. New Jersey: Wiley, 1994.
- [3] Armando Josué Angel Ramírez (6/05/2005): "Disposición a pagar por reducir la congestión vehicular en las vías primarias de la ciudad de México", disponible en http://catarina.udlap.mx/u_dl_a/tales/documentos/lec/angel_r_aj/ (fecha de consulta: 18/04/2015).
- [4] Jordi Sabaté: "Seis aplicaciones de móvil para evitar los atascos en carretera", disponible en <http://www.consumer.es/web/es/tecnologia/software/2014/11/18/220941.php> (fecha de consulta: 18/04/2015).
- [5] Diario Clarín (25/06/2014): "La aplicación BA Móvil suma el flujo de tránsito en tiempo real", disponible en http://www.clarin.com/capital_federal/TransitosmartphonesBA_Movil_0_1163284124.html (fecha de consulta: 18/04/2015).
- [6] "Aforo y clasificación vehicular con sensores en pavimento", disponible en <http://www.sistemasdesaje.com/sensores-en-pavimento.html> (fecha de consulta: 21/09/2015).
- [7] "Aforo y clasificación vehicular mediante sistemas elevados", disponible en <http://www.sistemasdesaje.com/sistemas-elevados.html> (fecha de consulta: 22/09/2015).
- [8] Sen-Ching, S. Cheung, and Chandrika Kamath. "Robust techniques for background subtraction in urban traffic video." *Electronic Imaging, International Society for Optics and Photonics* (2004).
- [9] Coifman, Benjamin, et al. "A real-time computer vision system for vehicle tracking and traffic surveillance." *Transportation Research Part C: Emerging Technologies* 6.4 (1998): 271-288.
- [10] Gervasonia, L., et al. "Un método eficiente para la sustracción de fondo en videos usando GPU." *Mecánica Computacional Vol XXXIII* (2014): 1721-1731.
- [11] Hsieh, Jun-Wei, et al. "Automatic traffic surveillance system for vehicle tracking and classification." *Intelligent Transportation Systems, IEEE Transactions on* 7.2 (2006): 175- 187.
- [12] Hadi, Raad Ahmed, Ghazali Sulong, and Loay Edwar George. "Vehicle Detection and Tracking Techniques: A Concise Review." *arXiv preprint arXiv:1410.5894* (2014).
- [13] Bouwmans, Thierry, Fida El Baf, and Bertrand Vachon. "Background modeling using mixture of gaussians for foreground detection-a survey." *Recent Patents on Computer Science* 1.3 (2008): 219-237.

[14] Stauffer, Chris, and W. Eric L. Grimson. "Adaptive background mixture models for realtime tracking." *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on..* Vol. 2. IEEE (1999).

[15] Pang, Clement Chun Cheong, William Wai Leung Lam, and Nelson Hon Ching Yung. "A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images." *Intelligent Transportation Systems, IEEE Transactions on* 8.3 (2007): 441-459.

[16] Ambardekar, Amol, Mircea Nicolescu, and George Bebis. "Efficient vehicle tracking and classification for an automated traffic surveillance system." *Proceedings of the 10th IASTED International Conference.* Vol. 623.

[17] Gupte, Surendra, et al. "Detection and classification of vehicles." *Intelligent Transportation Systems, IEEE Transactions on* 3.1 (2002): 37-47.

[18] Lai, Andrew HS, George SK Fung, and Nelson HC Yung. "Vehicle type classification from visual-based dimension estimation." *Intelligent Transportation Systems, Proceedings. IEEE. IEEE* (2001).

[19] Rad, Roya, and Mansour Jamzad. "Real time classification and tracking of multiple vehicles in highways." *Pattern Recognition Letters* 26.10 (2005): 1597-1607.

[20] Kanhere, Neeraj K., and Stanley T. Birchfield. "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features." *Intelligent Transportation Systems, IEEE Transactions on* 9.1 (2008): 148-160.

[21] Gonzalez, Rafael C., and Richard E. Woods, *Digital image processing 3rd edition*, 2007.

[22] TORRES, Wuilian J., and Roger J. BELLO. "Procesamiento de imágenes a color utilizando morfología matemática."

[23] Sommerville, Ian, and María Isabel Alfonso Galipienso. *Ingeniería del software*. Pearson Educación, 2005.

[24] Rumbaugh, James, and Ivar Jacobson. *El lenguaje unificado de modelado*. 1999.

[25] "About OpenCV", disponible en <http://opencv.org/about.html> (fecha de consulta: 6/10/2015).