

**DESARROLLO UN SISTEMA DE VISIÓN POR COMPUTADOR PARA LA  
IDENTIFICACIÓN DE LAS INFRACCIONES DE TRÁNSITO ORIENTADO A LA  
GENERACIÓN DE ESTUDIOS DE MOVILIDAD.**

Trabajo de Grado

Autor:

Zuly Marcela Vargas Pinilla

Director:

Diego M. Rivera.

UNIVERSIDAD PEDAGÓGICA NACIONAL  
FACULTAD CIENCIA Y TECNOLOGÍA  
LICENCIATURA EN ELECTRONICA  
BOGOTÁ D.C 2017


## AGRADECIMIENTOS

*Agradezco a mis padres y hermanos por su apoyo durante este ciclo de pregrado.*

*A la Universidad Pedagógica Nacional por haberme formado como licenciada y como persona*

*A mi compañero de vida por su apoyo incondicional durante la realización de este proyecto.*

*Al profesor Diego Rivera quien hizo posible que se realizara este proyecto.*

	<b>FORMATO</b>	
	<b>RESUMEN ANALÍTICO EN EDUCACIÓN - RAE</b>	
<b>Código:</b> FOR020GIB	<b>Versión:</b> 01	
<b>Fecha de Aprobación:</b> 21-02-2017	<b>Página</b> 1 de 5	

1. Información General	
<b>Tipo de documento</b>	Trabajo de Grado
<b>Acceso al documento</b>	Universidad Pedagógica Nacional. Biblioteca Central
<b>Título del documento</b>	Desarrollo un sistema de visión por computador para la identificación de las infracciones de tránsito orientado a la generación de estudios de movilidad.
<b>Autor(es)</b>	Vargas Pinilla, Zuly Marcela
<b>Director</b>	Rivera, Diego Mauricio
<b>Publicación</b>	Bogotá. Universidad Pedagógica Nacional, 2017. 75 p.
<b>Unidad Patrocinante</b>	Universidad Pedagógica Nacional
<b>Palabras Claves</b>	SISTEMA DE VISIÓN POR COMPUTADOR; INFRACCIONES DE TRÁNSITO; TRAYECTORIA.

2. Descripción
<p>Trabajo de grado que se propone desarrollar un sistemas de visión por computador para la identificación de las infracciones de tránsito en un cruce vehicular, por medio de un algoritmo que detecta la trayectoria de los vehículos y así determinar los infractores. Este sistema muestra datos cuantitativos detallados sobre algunas de las infracciones cometidas por los conductores las cuales son: cruce indebido, obstrucción y contravía en una zona determinada, teniendo en cuenta el código de tránsito.</p> <p>El algoritmo consta de una interfaz de usuario donde se ingresan el número de entradas y salidas que serán objeto de análisis de trayectoria y así se adquiere la cantidad de vehículos infractores en las secciones que el usuario ingreso, que son visualizadas en la interfaz de forma estadística.</p>

### 3. Fuentes

Gonzales P, Vision por computador imagenes digitales y aplicaciones, Mexico: Alfaomega, 2002.

MathWorks, *Image Morphology*, 2009.

SURA, «SURA,» Cuidado a conducir por las intersecciones viales, [En línea]. Available: <https://www.sura.com/blogs/autos/cuidado-intersecciones-viales.aspx>.

Nilakorn Seenouvong y Ukrit Watchareeruetai, «A Computer Vision Based Vehicle Detection and Counting System,» [En línea]. Available: <http://ieeexplore.ieee.org/document/7440510/>.

S. Atev, , H. Arumugam y O. Masoud, «A Vision-Based Approach to Collision Prediction at Traffic Intersections. IEEE International Conference on Information Acquisition,» 2005. [En línea]. Available: <http://ieeexplore.ieee.org/document/1549845/>.

L. Liwei , X. Junliang y Haizhou , «IEEE International Conference on Information Acquisition,» Multi-view Vehicle Detection and Tracking in Crossroads, [En línea]. Available: <http://ieeexplore.ieee.org/document/6166688/>.

M. Abril y Mauricio Valencia, «redalyc.org,» Sistema De Visión Artificial Para El Registro De Densidad Peatonal En Tiempo Real, [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=84903515>.

University, Department of Computer Science and Information Engineering National Taipei, «IEEE International Conference on Information Acquisition,» Crossroad Traffic Surveillance Using Superpixel Tracking and Vehicle Trajectory, 2014. [En línea]. Available: <http://ieeexplore.ieee.org/document/6977103/>.

Shung-Tsang Tseng, Canción Kai-Tai y Jen-Chao Tai, «Departamento de Ingeniería Eléctrica y de control,» Real-Time Image Tracking For Automatic Traffic Monitoring And Enforcement Applications, [En línea]. Available: [www.elsevier.com/locate/imavis](http://www.elsevier.com/locate/imavis).

Kamijo, Matsushita, Ikeuchi y Sakauchi, «IEEE International Conference on Information Acquisition,» Traffic Monitoring and Accident Detection at Intersections, 2000. [En línea]. Available: <http://ieeexplore.ieee.org/document/880968/>.

W. Xiaoling, M. Li-Min y Z. Biaobiao , «A Video-based Traffic Violation Detection System. IEEE International Conference on Information Acquisition,» 2013- 2014, [En línea]. Available: <http://ieeexplore.ieee.org/document/6885246/>.

H. M. Muhammad y H. A. Muhammad , «IEEE International Conference on Information Acquisition,» Vision Based Intelligent Traffic Management System. , [En línea]. Available: <http://ieeexplore.ieee.org/document/6137134/>.

C. F. Q. Urrego, «Adquisición de variables de tráfico vehicular usando visión artificial. Revista de ingeniería.,» (2009). [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=43021583005>.

- Quiroga, Romero, García y Parra, «Revista Facultad de Ingeniería Universidad de Antioquia.,» Adquisición de variables de tráfico peatonal utilizando vision por computador, (2010- 2011). [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=43021583005>.
- FRAV, «FRAV,» Estudio de intersecciones semaforicas regulas por vision artificial, [En línea]. Available: <http://frav.es/index.php/trabajos-fin-de-grado/107-leidos/resumenes/158-estudio-de-intersecciones-semaforicas-reguladas-por-medio-de-vision-artificial>.
- F. GRANADOS y J. I. MARIN , «redalyc.org,» Detección De Flujo Vehicular Basado En Visión Artificial, [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=84903530>.
- Katanyoo , Wittaya y Surasak , «Traffic Violation Detection Using Multiple Trajectories Evaluation of Vehicles. IEEE International Conference on Information Acquisition,» 2013. [En línea]. Available: <http://ieeexplore.ieee.org/document/64>.
- Unad, «Ingeniería de Software,» El modelo DRA (Desarrollo Rápido de Aplicaciones), [En línea]. Available: [http://datateca.unad.edu.co/contenidos/301404/301404\\_ContentadoEnLinea/leccin\\_12\\_el\\_modelo\\_de\\_construccion\\_de\\_prototipos.html](http://datateca.unad.edu.co/contenidos/301404/301404_ContentadoEnLinea/leccin_12_el_modelo_de_construccion_de_prototipos.html).
- J. Portillo , «Instituto politecnico nacional,» Detección de movimiento de objetos mediante secuencias de video., [En línea]. Available: <http://www.sepi.esimez.ipn.mx/electronica/archivos/991.pdf>.
- G. Santillán y I. D. Ibarra, «Visión Artificial y Procesamiento Digital de Imágenes usando Matlab,» García Santillán, Iván Danilo, Visión Artificial y Procesamiento Digital de Imágenes usando Matlab. Ibarra – Ecuador,2008.pag 73}, 2008. [En línea].
- Universidad valladoid, «Vision artificial industrial,» Morfologia, [En línea]. Available: [www.librovision.eii.uva.es/ppt/5\\_Morfologia.pptx](http://www.librovision.eii.uva.es/ppt/5_Morfologia.pptx).
- L. Hincapie, Desarrollo De Una Herramienta Sistémica Basada En procesamiento digital, [En línea]. Available: <https://red.uao.edu.co/bitstream/10614/6472/1/T04528.pdf>.
- MathWorks, *Correcting Nonuniform Illumination*.
- MathWorks, *Motion-Based Multiple Object Tracking*.
- The MathWorks, Inc., «The MathWorks, Inc.,» Image Processing Toolbox User's Guide, 1994-2005. [En línea]. Available: <http://matlab.izmiran.ru/help/toolbox/images/enhanc17.html>.
- «Rotonde cam,» [En línea]. Available: <http://rotondecam.nl/>,. [Último acceso: 01 2017].
- J. A. Somolinos Sánchez, Avances en robótica y visión por computadorEscrito por, Ediciones universidad de castilla, 2002.
- José F. Valencia, Evaluación del impacto del preprocesamiento de imagenes en la segmentacion del gris, 2014. [En línea]. Available:

<http://www.scielo.org.co/pdf/teclo/v17n33/v17n33a04.pdf>.

Conde, Pérez, González, Silva, Vaquero y Rodríguez, Estudio De Intersecciones Semafóricas Reguladas Por Visión Artificial, (2003). [En línea]. Available: <http://www.frav.es/pdf/2003/colima2003.pdf>.

#### 4. Contenidos

##### Antecedentes

Desarrollo del sistema se plantean las técnicas implementadas para ; Determinar la trayectoria, detectar y seguir los vehículos, además de esto se describen las estrategias para identificar las infracciones de tránsito y la trayectoria.

Descripción del algoritmo se presentan en diagramas de flujo los códigos implementados.

Resultados son la evaluación de desempeño del algoritmo con videos simulados y reales.

#### 5. Metodología

Para el desarrollo del algoritmo debe ser de modo sistemático, donde se divide el proyecto en varias etapas para idear, implementar y obtener resultados, en este caso se usa la metodología (RAD) desarrollo rápido de aplicaciones las fases que la componen son:

##### 1. Planificación:

Esta etapa se determina las funciones que debe cumplir la aplicación, las cuales son:

- Identificar las infracciones de tránsito contravía, cruce indebido y obstrucción.
- Determinar la trayectoria de los vehículos.
- Interfaz de visualización de los resultados.

##### 2. Diseño:

En esta etapa se realizan las siguientes actividades:

- Diseño de la aplicación para el usuario.

- Diseño de estrategias para hallar la trayectoria de los vehículos.
- Diseño de estrategias para identificar vehículos infractores.
- Programación y análisis de los algoritmos implementados para determinar cuál es estos se adapta mejor a nuestras necesidades.
- Realizar un video que cumpla con las características que el algoritmo requiere para funcionar correctamente.

### 3. Construcción:

En esta se implementa el algoritmo escogido y se realiza una evaluación de los resultados obtenidos con el fin de implementar mejoras a este.

## 6. Conclusiones

- El sistema de visión por computador para la identificación de las infracciones de tránsito permite al usuario interactuar con las variables del algoritmo e identificar las infracciones de tránsito ya descritas como lo son; cruce indebido, obstrucción y contravía en videos ya sea real o simulado cumple con los requerimientos.
- Para este sistema de visión de artificial se aplicó una técnica de corrección de iluminación no uniforme para mejorar el proceso de segmentación en este caso mejorando el contraste de la imagen RGB.
- Se ha desarrollado una estrategia para determinar la trayectoria de los vehículo por medio de zonas marcadas en cada fotograma como entradas y salidas teniendo en cuenta que el usuario determina estas marcas y en caso de que un vehiculo no atraviese algunas de estas zonas no se podrá determinar dicha trayectoria, ya sea porque no existe zona marcada o el vehículo se encuentra dentro del fotograma al inicio del video, esta estrategia resultó ser eficiente cumpliendo con el objetivo de detectar la trayectoria.

- La evaluación de desempeño del algoritmo depende en cierta medida de las características del video y de la posición de la cámara que en condiciones ideales con videos simulados arroja un informe confiable.
- Para visualizar los resultados del algoritmo se diseñó una interfaz de usuario que le permite a este realizar calibración del umbral y las variables de entrada y salida en el video así mismo observar la reproducción del video y en simultánea a este la segmentación correspondiente en el video.

<b>Elaborado por:</b>	Vargas Pinilla, Zuly Marcela
<b>Revisado por:</b>	Rivera, Diego Mauricio

<b>Fecha de elaboración del Resumen:</b>	21	02	2017
--	----	----	------



## Contenido

1	INTRODUCCIÓN .....	14
1.1	Planteamiento <i>del problema</i> .....	15
1.2	Objetivos .....	15
1.2.1	Objetivo general .....	15
1.2.2	Objetivos específicos .....	15
1.3	Antecedentes .....	16
1.3.1	A Computer Vision Based Vehicle Detection and Counting System .....	16
1.3.2	A Vision-Based Approach to Collision Prediction at Traffic Intersections.....	16
1.3.3	Multi-view Vehicle Detection and Tracking in Crossroads.....	16
1.3.4	Sistema De Visión Artificial Para El Registro De Densidad Peatonal En Tiempo Real	17
1.3.5	Crossroad Traffic Surveillance Using Superpixel Tracking and Vehicle Trajectory Analysis	17
1.3.6	Real-Time Image Tracking For Automatic Traffic Monitoring And Enforcement Applications.....	17
1.3.7	Traffic Monitoring and Accident Detection at Intersections.....	18
1.3.8	A Video-based Traffic Violation Detection System .....	18
1.3.9	Vision Based Intelligent Traffic Management System.....	18
1.3.10	Adquisición de variables de tráfico vehicular usando visión por computador .....	18
1.3.11	Adquisición de variables de tráfico peatonal utilizando visión por computado.....	19
1.3.12	Estudio de intersecciones semafóricas reguladas por visión artificial.....	19
1.3.13	. Detección De Flujo Vehicular Basado En Visión Artificial.....	19
1.3.14	Traffic Violation Detection Using Multiple Trajectories Evaluation of Vehicles .....	19
2	MARCO TEÓRICO Y METODOLOGÍA.....	20
2.1	Metodología .....	20
2.1.1	Planificación: .....	20
2.1.2	Diseño: .....	20
2.1.3	Construcción:.....	20
2.2	Marco Teórico .....	21

2.2.1	Espacio RGB y escala de Grises.....	21
2.2.2	Seguimiento De Vehículos.....	21
2.2.3	Filtro De Kalman .....	22
2.2.4	Procesamiento De Imágenes.....	23
2.3	Iluminación.....	24
2.3.1	Posición de la cámara.....	25
2.4	Algoritmos Implementados.....	26
3	DESARROLLO .....	28
3.1	Descripción general del sistema.....	28
3.2	Captura de video .....	29
3.3	Segmentación.....	29
3.4	Determinar Trayectoria e Identificación de Infracciones de tránsito .....	32
3.4.1	Elementos de Entrada – Salida .....	32
3.5	Estrategia para determinar trayectoria.....	33
3.6	Estrategias Para Identificar Infracciones De Transito .....	34
3.6.1	Cruce indebido .....	35
3.6.2	Obstrucción .....	36
3.6.3	Contravía .....	36
3.7	Procesamiento De Imágenes.....	36
3.7.1	Pre-Procesamiento .....	36
3.8	Seguimiento de Objetos.....	36
3.9	Interfaz Grafica.....	37
3.9.1	Panel de control: .....	38
3.9.2	Informe:.....	40
3.10	Descripción Del Algoritmo.....	41
3.10.1	Cargar Video .....	44
3.10.2	Entradas / Salidas .....	45
3.10.3	Detección y Predicción .....	45
3.10.4	Llenar tabla.....	46
3.10.5	Identificar la trayectoria del vehículo .....	47
3.10.6	Generación Informe .....	48
3.10.7	Iluminación.....	48

3.10.8	Identificación de Infracciones .....	51
4	RESULTADOS .....	53
4.1	Extracción del fondo.....	53
4.1.1	Video simulado UNITY.....	53
4.2	Video Real extraído cámara IP .....	57
4.3	Video simulado en PTV VISSIM .....	60
4.4	Análisis de video elaboración propia .....	63
5	CONCLUSIONES .....	64
6	Anexos .....	65
7	Bibliografía .....	73

## Tabla de Imágenes

<i>Figura 1 Espacio RGB [19].....</i>	<i>21</i>
<i>Figura 2: Imagen binaria .....</i>	<i>22</i>
<i>Figura 3 Obtención vehículo .....</i>	<i>22</i>
<i>Figura 4: Imagen dilatada [21] .....</i>	<i>23</i>
<i>Figura 5 Imagen erosionada [21] .....</i>	<i>24</i>
<i>Figura 6 Imagen morfología cierre [21].....</i>	<i>24</i>
<i>Figura 7 Esquema general .....</i>	<i>29</i>
<i>Figura 8 Fotograma original.....</i>	<i>30</i>
<i>Figura 9 Segmentación de un fotograma antes de aplicar las operaciones morfológicas .....</i>	<i>30</i>
<i>Figura 10 Dilatación .....</i>	<i>31</i>
<i>Figura 11 Apertura .....</i>	<i>31</i>
<i>Figura 12 Cierre .....</i>	<i>31</i>
<i>Figura 13 Selección área.....</i>	<i>32</i>
<i>Figura 14 Entradas (Verde)/ Salidas (Amarillo). .....</i>	<i>33</i>
<i>Figura 15 Calcular la intersección entre dos cajas.....</i>	<i>34</i>
<i>Figura 16 Interacción entre un vehículo y una salida .....</i>	<i>34</i>
<i>Figura 17 Entrada y salidas .....</i>	<i>35</i>
<i>Figura 18 Interfaz Gráfica.....</i>	<i>37</i>
<i>Figura 19 visualización de los botones del panel de control.....</i>	<i>38</i>
<i>Figura 20 Edición de entradas y salidas.....</i>	<i>38</i>
<i>Figura 21 Tabla comportamiento .....</i>	<i>39</i>
<i>Figura 22 Display de visualización .....</i>	<i>39</i>
<i>Figura 23 Información en tiempo de procesamiento.....</i>	<i>40</i>
<i>Figura 24 Botones de navegación .....</i>	<i>40</i>
<i>Figura 25 Reporte .....</i>	<i>41</i>
<i>Figura 26 Algoritmo de detección de vehículos. ....</i>	<i>43</i>
<i>Figura 27 Tabla Información general .....</i>	<i>47</i>
<i>Figura 28 Ajuste de Iluminación .....</i>	<i>49</i>
<i>Figura 29 Imagen Original.....</i>	<i>49</i>
<i>Figura 31 Histograma imagen original.....</i>	<i>50</i>
<i>Figura 30 Histograma ajuste iluminación.....</i>	<i>50</i>
<i>Figura 32 Algoritmo identificación de infracciones .....</i>	<i>52</i>
<i>Figura 33 superior fondo original, inferior fondo Extracción de fondo simulado UNITY .....</i>	<i>54</i>
<i>Figura 34 Resultado de la segmentación simulación n1.....</i>	<i>55</i>
<i>Figura 35 Grafica comportamiento vehicular video simulado 1.....</i>	<i>56</i>
<i>Figura 36 original y corrección de iluminación .....</i>	<i>57</i>
<i>Figura 37 Segmentación video real .....</i>	<i>58</i>
<i>Figura 38 Extracción de fondo video real .....</i>	<i>58</i>
<i>Figura 39 comportamiento vehicular video real.....</i>	<i>60</i>

<i>Figura 40 Resultado de la segmentación simulación n3.....</i>	<i>61</i>
<i>Figura 41 Extracción de fondo simulación n3.....</i>	<i>61</i>
<i>Figura 42 comportamiento vehicular simulación n3.....</i>	<i>62</i>
<i>Figura 43 Imagen tomada desde un quinto piso en la ciudad de Bogotá. ....</i>	<i>63</i>
<i>Figura 44: Imagen binaria se superposición de un vehiculo sobre otro .....</i>	<i>64</i>

## Índice de tablas

<i>Tabla 1: Resultados de falso positivos, falsos negativos, verdaderos positivos y verdaderos negativos.....</i>	<i>55</i>
<i>Tabla 2: Reporte de la simulación n1 .....</i>	<i>55</i>
<i>Tabla 3: Resultados de comportadito video simulado.....</i>	<i>56</i>
<i>Tabla 4: Informe generado por el algoritmo video real.....</i>	<i>59</i>
<i>Tabla 5 Resultados de comportamiento video real .....</i>	<i>59</i>
<i>Tabla 6: Resultados de la simulación n3.....</i>	<i>61</i>
<i>Tabla 7: Resultados de comportadito video simulado n3.....</i>	<i>62</i>

## 1 INTRODUCCIÓN

El ser humano carece de habilidades para la realización de algunas tareas, en el caso de los sentidos, la visión está limitada debido a la cantidad de tareas que el hombre puede realizar al mismo tiempo, por este tipo de razones se diseñan sistemas capaces de realizar actividades extras conocidos como visión artificial en estos, las máquinas son dotadas de cámaras para simular el sistema de visión realizando un procesamiento digital de imágenes con capacidad de medir magnitudes físicas.

Un sistema de visión artificial por computador es la capacidad de máquina para ver el mundo que le rodea [1] así extraer características y propiedades de las imágenes utilizando procesos automáticos.

Debido al aumento en la densidad del tráfico vehicular se hace cada vez más difícil monitorear los vehículos que circulan por las vías por tanto, el objetivo de este proyecto es desarrollar un sistema de visión por computador para la identificación de las infracciones de tránsito en un video, del cual se extrae información contenida en las imágenes de éste, para su interpretación y análisis correspondiente. El algoritmo consta de una interfaz de usuario donde se ingresan el número de entradas y salidas que serán objeto de análisis de trayectoria y así se adquiere la cantidad de vehículos infractores en las secciones que el usuario ingreso, que son visualizadas en la interfaz de forma estadística. El procesamiento de imágenes es de suma importancia en el desarrollo de este proyecto debido a los cambios de iluminación que se presentan en las escenas captadas en el video, además de esto permite el mejoramiento de las imágenes y reducción de ruido. Primero, se realiza un algoritmo para extraer los carros del fondo luego de esto se realiza seguimiento y predicción de los vehículos y se determinan las estrategias para identificar las infracciones de tránsito cometidas.

El sistema está desarrollado en MatLab 2015a utilizando la herramienta de GUIDE, en la que se programa la adquisición de un video, para su posterior procesamiento, con el apoyo de otras herramientas como la aplicación image morphology [2], con esta última se logra crear las diferentes operaciones morfológicas para la mejorar la calidad de las imágenes y luego aplicar los algoritmos de reconocimiento de comportamiento y asignación de infracciones de tránsito, se busca que el sistema detecte y evalúe el comportamiento de los vehículos generando reportes, en tiempo de ejecución y que posteriormente sean

guardados para su análisis, esto se alcanza utilizando segmentación fotograma a fotograma de toda la película a evaluar, identificando la trayectoria de cada vehículo, realizando marcas para la detección de la posición en el fotograma , generando una tabla de permisos para la asignación de infracciones.

En el transcurso de este documento se explicará paso a paso cómo se consigue cumplir con estos objetivos y la realización de pruebas con videos reales y simulaciones de laboratorio para obtener un resultado deseado y generar las conclusiones de lo obtenido.

## 1.1 Planteamiento *del problema*

En las intersecciones viales o los cruces vehiculares se debe tener atención al conducir debido a las maniobras que se deben realizar en éstos, un conductor desatento puede incurrir en una infracción de tránsito y con esto un accidente [3] Según el artículo “Cuidado al conducir por las intersecciones viales” de cada 100 accidentes, 20 se presentan en estos espacios de la vía.

En las ciudades con mayor índice de infracciones vehiculares poniendo en riesgo a los demás conductores y peatones. Convirtiéndose para las autoridades encargadas del tránsito en una labor tediosa detectar el número de infracciones vehiculares cometidas debido al desmesurado crecimiento de los vehículos en circulación aumentando que así mismo aumentan las cifras de accidentalidad, aunque existan campañas de concientización de la importancia de una conducción segura, es por esto que es necesario automatizar la información del tráfico en una aplicación que además de mostrar el comportamiento de los vehículos en la zona genere un informe para que estos datos sean analizados.

## 1.2 Objetivos

### 1.2.1 Objetivo general

Desarrollar un sistema de visión por computador para la identificación de las infracciones de tránsito orientado a la generación de estudios de movilidad.

### 1.2.2 Objetivos específicos

- Aplicar técnicas de corrección de iluminación para mejorar el proceso de segmentación.
- Implementar técnicas de segmentación para localizar los vehículos en videos.

- Programar una estrategia apropiada que permita hallar la trayectoria de los vehículos en las vías.
- Realizar la evaluación de desempeño de las técnicas implementadas.
- Diseñar una interfaz gráfica que permita visualizar los resultados del algoritmo.

### 1.3 Antecedentes

Los sistemas de visión artificial se han incrementado empleándose en diferentes disciplinas, por ello se diseñan cada día nuevos algoritmos más eficientes que de una u otra forma complementan al ser humano para la realización de actividades repetitivas, para el caso de la visión artificial realizan prácticas por medio del monitoreo de vehículos con el objetivo de mejorar la movilidad en las vías y disminuir el número de accidentes.

Los sistemas que usan para el seguimiento y conteo de los vehículos la técnica de sustracción de fondo son;

#### 1.3.1 A Computer Vision Based Vehicle Detection and Counting System

Este proyecto consiste en el seguimiento de vehículos y el conteo de los mismos en tiempo real para esto se utilizó la técnica de sustracción de fondo para detectar los objetos en movimiento, para el tratamiento de imágenes usa operaciones morfológicas y llenado de agujero y como estrategia de conteo uso cinco zonas por las cuales ingresaban los vehículos para determinar e identificar cada uno de estos por medio del centroide [4]

#### 1.3.2 A Vision-Based Approach to Collision Prediction at Traffic Intersections

Sistema se ejecuta en tiempo real y que se adapta a los cambios en el medio ambiente durante la duración de la vigilancia. Método de sustracción de fondo, mezcla gaussiano y blob para encontrar las regiones que encuentran conectadas y así detectar los objetos en movimiento [5].

#### 1.3.3 Multi-view Vehicle Detection and Tracking in Crossroads

En este trabajo se presenta un algoritmo de rastreo y detección en un cruce en tiempo real y sistema de procesamiento por medio de sustracción de fondo y filtros para detectar contornos. La estrategia usada para detectar el movimiento del vehículo es por medio de unas regiones determinadas por líneas. Además de esto el algoritmo funciona con diferentes condiciones meteorológicas (nieve, soleado y nublado) demuestran la eficacia y la eficiencia [6].



#### 1.3.4 Sistema De Visión Artificial Para El Registro De Densidad Peatonal En Tiempo Real

Sistema de detección de transeúntes los cuales son contabilizados en video en un margen de tiempo determinado. Para detectar el movimiento de las personas en la vía inicialmente se realiza sustracción de fondo luego de que ya detecto los objetos en movimiento se realiza una binarización con el fin de eliminar los píxeles que no corresponden a un transeúnte y por medio de un umbral realizar el seguimiento de estos, hallar el centroide y etiquetarlos [7].

Los siguientes algoritmos usan el filtro de kalman para predecir la trayectoria del objeto

#### 1.3.5 Crossroad Traffic Surveillance Using Superpixel Tracking and Vehicle Trajectory Analysis

Este documento, tiene como finalidad desarrollar un algoritmo que permita detectar la trayectoria de varios vehículos en una vía, verificar el comportamiento de estos en la vía como; infracciones como el giro indebido y analizar la información de los objetos en movimiento en video. Diseñaron un método que ellos denominaron "Trajectory Map" que descubre las áreas por las cuales los vehículos transitan con mayor frecuencia , usan el filtro de Kalman para el seguimientos de los vehículos y para detectar cruce indebido usa técnica de rejillas identificando con colores la trayectoria del vehículo [8].

#### 1.3.6 Real-Time Image Tracking For Automatic Traffic Monitoring And Enforcement Applications

Un sistema de monitorización del tráfico basado en tiempo real para el seguimiento automático de automóviles y motocicletas en un cruce de carretera. Para detectar la trayectoria de los vehículos mide el borde de cada objeto detectado con respecto a dos zonas determinadas, cuando estos bordes están demasiado cerca se superponen formando una imagen.

Cuando un coche produce una colisión, los vehículos rastreados detienen rápidamente. Las previsiones de los estados del modelo dinámico mover ligeramente el sobre y por lo tanto el contorno, estimada por el modelo dinámico de predicción y técnicas de filtrado de Kalman, superará la actual del vehículo actual [9].

Algoritmos que fueron diseñados para monitorear el tráfico vehicular:

### 1.3.7 Traffic Monitoring and Accident Detection at Intersections

Sistema que permite monitorear el tráfico vehicular en una intersección, este sistema es capaz de detectar el comportamiento de cada uno de los vehículos en la vía. Para esto desarrollaron un algoritmo que denominaron Markov random field (MRF), el cual clasifica cada bloque en vehículos o, de forma equivalente, asigna una etiqueta del vehículo para cada bloque, el cual permite evaluar el comportamiento tanto de los peatones como de los vehículos en una intersección con el mínimo margen de error y así detectar las infracciones de tránsito y los accidentes. Las infracciones que trata este algoritmo son; cruce indebido y obstrucción [10]

### 1.3.8 A Video-based Traffic Violation Detection System

Sistema de detección de infracciones de tráfico, tales como; acelerar cuando el semáforo está en rojo y retroceder el vehículo. Usando Background Difference como método de segmentación además de esto aplicando la transformada de Wavelet permite el reconocimiento de formas como en este caso el contorno de la imagen. Por medio del conteo de píxeles se puede obtener la comparación de parámetros como área, perímetro, compacidad y diámetro de los contornos [11]

### 1.3.9 Vision Based Intelligent Traffic Management System

En este trabajo se propone un algoritmo de detección de vehículos y la estimación de la densidad de tráfico que conduce a un eficiente sistema de gestión de tráfico, la detección de los vehículos se realiza por medio de Dynamic background subtraction y operaciones morfológicas, la precisión del algoritmo es de 92 % en cuanto al conteo de vehículos, realiza por medio de una interfaz gráfica de usuario para verificar mejor los resultados del proceso [12].

### 1.3.10 Adquisición de variables de tráfico vehicular usando visión por computador

Este artículo es el desarrollo de un sistema de detección de vehículos con el fin de realizar un conteo de los mismos en tiempo real y así mismo estimar la velocidad promedio. Desarrollan un algoritmo de detección de primer plano así detectar los vehículos además de esto incorporan un modelo FGD que permite obtener los bordes, para unir los segmentos que pertenecen al mismo objeto además de esto un algoritmo de media aumentada el cual consiste en asignar un color al valor más alto al pixel que es parecido al pavimento, este algoritmo debe adaptarse a los posibles cambios de iluminación [13].

Proyectos realizados con el fin de monitorear a los peatones:

#### 1.3.11 Adquisición de variables de tráfico peatonal utilizando visión por computado.

Este proyecto fue realizado para contar los peatones que se movilizan por una vía, para esto realiza una segmentación por medio de borde y regiones además de esto usan el algoritmo EM (Expectation Maximization) para extraer características que permitan identificar que es una persona. El seguimiento es realizado con la técnica de flujo óptico (Lukas kanade). Para el conteo de peatones usan una línea que permite contar solamente los peatones quienes la atraviesen, para determinar el número de peatones que circulan en un grupo calculan el promedio del área ocupada por cada persona así determinan la cantidad de personas que transitan en un grupo [14].

#### 1.3.12 Estudio de intersecciones semafóricas reguladas por visión artificial.

En este proyecto desarrollaran un algoritmo de alto nivel para determinar la frecuencia con la que transitan los peatones por una cebsa, así mismo los vehículos, para controlar el funcionamiento de un semáforo permitiendo mejorar la movilidad. Mediante el uso de dos cámaras una que detecta personas y otra vehículos [15].

#### 1.3.13 . Detección De Flujo Vehicular Basado En Visión Artificial

Sistema que detecta los vehículos y los peatones con esto realizando un conteo en las intersecciones y avenidas. Para controlar la iluminación del sistema utilizan filtrado homomórfico, para detectar las personas y los vehículos en movimiento para esto realizan una suma de la diferencia entre dos secuencias, posteriormente se aplica un umbral y segmentan cada uno de los objetos en movimiento, seguido de esto se extrae las características de los objetos por medio de los descriptores de Fourier para así clasificarlos ya sea personas o vehículos en una red neuronal [16]

#### 1.3.14 Traffic Violation Detection Using Multiple Trajectories Evaluation of Vehicles

Este documento utiliza el método diferencia fotograma basado en el modelo de color HSV para detectar vehículo. Los procedimientos de detección de vehículos algoritmo son como sigue: transformar la imagen de la región de detección desde el espacio de color RGB al espacio de color HSV. La estrategia usada en este algoritmo es por medio de comparación de regiones establecidas para lograr con mayor facilidad solo el comportamiento de los píxeles en dichas áreas y así compararlos [17].

## 2 MARCO TEÓRICO Y METODOLOGÍA

### 2.1 Metodología

Para el desarrollo del algoritmo debe ser de modo sistemático, donde se divide el proyecto en varias etapas para idear, implementar y obtener resultados, en este caso se usa la metodología (RAD) desarrollo rápido de aplicaciones las fases que la componen son [18]:

#### 2.1.1 Planificación:

Esta etapa se determina las funciones que debe cumplir la aplicación, las cuales son:

- Identificar las infracciones de tránsito contravía, cruce indebido y obstrucción.
- Determinar la trayectoria de los vehículos.
- Interfaz de visualización de los resultados.

#### 2.1.2 Diseño:

En esta etapa se realizan las siguientes actividades:

- Diseño de la aplicación para el usuario.
- Diseño de estrategias para hallar la trayectoria de los vehículos.
- Diseño de estrategias para identificar vehículos infractores.
- Programación y análisis de los algoritmos implementados para determinar cuál es estos se adapta mejor a nuestras necesidades.
- Realizar un video que cumpla con las características que el algoritmo requiere para funcionar correctamente.

#### 2.1.3 Construcción:

En esta se implementa el algoritmo escogido y se realiza una evaluación de los resultados obtenidos con el fin de implementar mejoras a este.

## 2.2 Marco Teórico

Visión artificial según Gózales P. En el libro visión por computador imágenes digitales y aplicaciones [1] es la “ciencia que estudia la interpretación de imágenes mediante computadores digitales” que tiene como fin adquirir los datos o características de una imagen, para la interpretación de escenas en videos o imágenes por medio de diversas técnicas como el procesamiento de imágenes (captura, transformación, codificación de imágenes), reconocimiento de forma entre otras.

### 2.2.1 Espacio RGB y escala de Grises

Una imagen a color es un espacio RGB formada por tres capas o planos de color; rojo (R), verde (G) y azul (B) para el procesamiento de imágenes es necesario transformar cada uno de los fotogramas de una secuencia de video en escala de grises para generar menor gasto computacional. En la Figura 1 Espacio RGB está definida en un espacio tridimensional, en tanto que al convertirla en una escala de grises nos permite definirla en un espacio unidimensional con valores de 0 a 255 [19]

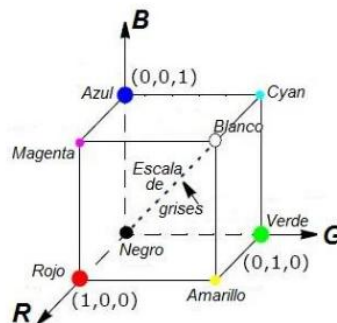


Figura 1 Espacio RGB [19]

### 2.2.2 Seguimiento De Vehículos

#### 2.2.2.1 Umbralización

Esta técnica de binarización permite convertir una imagen en escala de grises a binaria, como se muestra en la Figura 2: Imagen binaria (a), es una imagen a escala de grises, se compara el nivel de gris de cada píxel de la imagen con el umbral establecido por el usuario cuando el valor de este píxel tiene una intensidad menor al umbral es 0 (negro) y cuando la intensidad es mayor es 1 (blanco), en la imagen (b) se muestra el resultado de la imagen binaria.

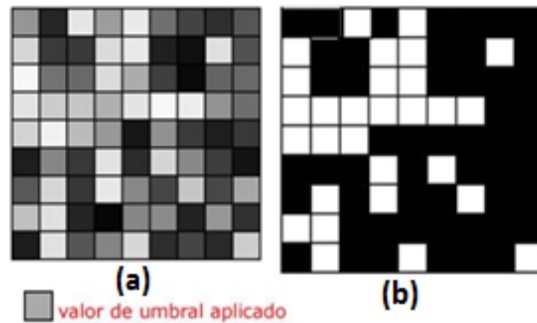


Figura 2: Imagen binaria

### 2.2.2.2 Segmentación

Según G Santillán en el libro visión artificial y procesamiento digital de imágenes usando MatLab es el proceso por el cual se extrae de la imagen cierta información subyacente para su posterior uso [20]. En este caso la segmentación está orientada por similitud, es decir por regiones.

#### 2.2.2.2.1 Substracción del fondo (Detección de primer plano.).

Esta técnica es utilizada para la detección de los vehículos en movimiento, la cual consiste en una diferencia entre el fotograma actual y el modelo de fondo como se muestra en Figura 3 Obtención vehículo. Dando como resultado los objetos que permanezcan sin movimiento aparecen en negro y los objetos que cambian aparecen en blanco. Además de esto con la técnica de binarización se escoge un umbral para cada uno de los videos, dependiendo del umbral asignado se puede eliminar sombras y píxeles que no hacen parte de ningún carro y así extraer solo los carros eliminando de la imagen motos y personas.

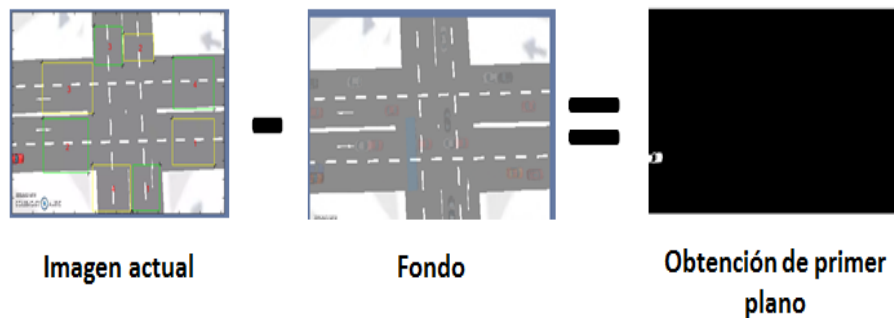


Figura 3 Obtención vehículo

Referencia: Elaboración propia

### 2.2.3 Filtro De Kalman

El filtro de Kalman, es un algoritmo recursivo que procesa datos de manera óptima utilizado en diferentes aplicaciones entre ellas un modelo que permite predecir la

posición de un objeto en este caso los vehículos que realiza una estimación estados pasados, presentes y futuros. Este filtro a diferencia de otros posee una respuesta dinámica mucho mejor comparadas e incluso con el filtro de Luenberger ya que este filtro depende de los términos de corrección y predicción.

Por tanto la programación del algoritmo del filtro de Kalman requiere un lenguaje con más facilidad de programación de operaciones entre matrices como es el caso del lenguaje de programación MatLab.

#### 2.2.3.1 Centroide

Conocido como el centro de masa esta característica es utilizada para conocer la posición y evaluar su comportamiento frente al filtro que se encarga de predecir su posición en este caso el filtro de Kalman y es a partir de este que se generan las cajas que rodean el objeto.

### 2.2.4 Procesamiento De Imágenes

#### 2.2.4.1 Operaciones Morfológicas

Después de la segmentación de las imágenes es importante para la extracción de objetos, representar los píxeles que forman una región, esto con el fin de eliminar píxeles que no hagan parte de un vehículo.

##### 2.2.4.1.1 Dilatación

Permite rellenar agujeros o unir regiones próximas que en la imagen se han podido separar, como se evidencia en la Figura 4: Imagen dilatada en (a) es la imagen original al aplicar un elemento estructurante (b) con su respectivo origen da como resultado (c) las regiones que están aislados del objeto se unen y forman una sola imagen.

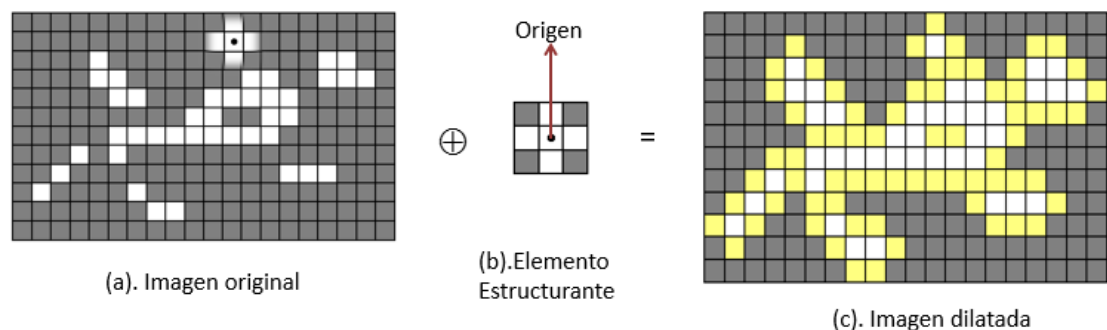


Figura 4: Imagen dilatada [21]

#### 2.2.4.1.2 Erosión

Se emplea para eliminar pequeños detalles quedando solo las formas de mayor tamaño, como se muestra en la Figura 5 Imagen erosionada

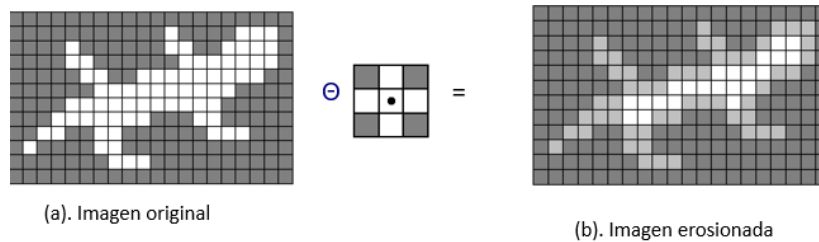


Figura 5 Imagen erosionada [21]

#### 2.2.4.1.3 Cierre

Esta operación morfológica permite rellenar los agujeros que tenga la imagen, como se evidencia en la Figura 6 Imagen morfología cierre.

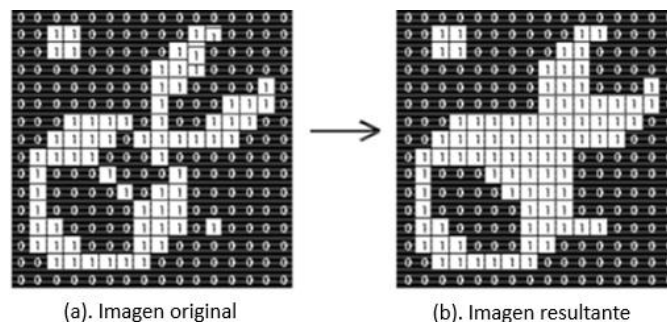


Figura 6 Imagen morfología cierre [21]

#### 2.2.4.2 Extracción de Regiones y Etiquetado

##### *Blob Analysis* (Análisis de manchas)

Detecta grupos de píxeles conectados en una imagen binaria, que suelen corresponder a los objetos en movimiento así mismo etiqueta cada uno de estos, para que los objetos nuevos no sean nuevamente detectados en los siguientes fotogramas del video, así es posible conocer la posición de un objeto por medio del cálculo del centroide.

### 2.3 Iluminación

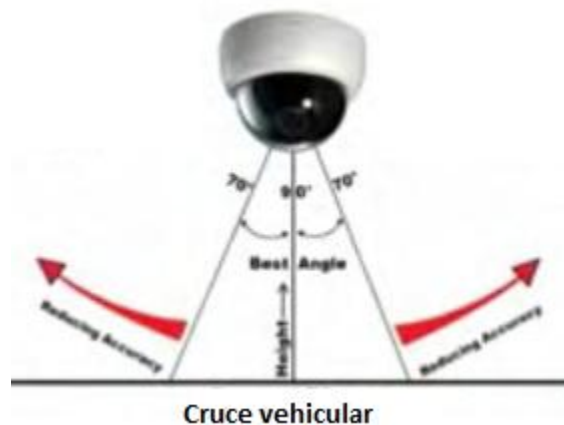
La luz ambiente no proviene de una dirección concreta, es decir incide en todas las direcciones sobre el objeto, es por esto que se generan sombras debido a elementos del ambiente como árboles, postes, los mismos vehículos esto generando cambios en la iluminación de la escena. El primer aspecto que hay que



considerar en lo que respecta a la estabilidad en la iluminación no debe verse afectada por variaciones de la luz ambiente.

### 2.3.1 Posición de la cámara

La ubicación de la cámara debe ser un modelo de reflexión de luz como se observa en la Ilustración 1 Posición de la cámara el rayo de luz incide en todas las direcciones sobre la escena. Para que el algoritmo funcione correctamente se debe grabar un video con parámetros específicos como; la posición de la cámara Ilustración 1 Posición de la , la cual debe estar ubicada de forma frontal al cruce con un ángulo de  $90^\circ$  y no superar uno de  $70^\circ$ , esto se logra ubicando la cámara en ángulo de visión alto con el fin de capturar el mayor número de elementos de la escena y así facilitando el procesamiento de imágenes.



*Ilustración 1 Posición de la cámara/ Iluminación frontal [22]*

## 2.4 Algoritmos Implementados

### 2.4.1.1 Algoritmo Corrección De Iluminación

Debido a los cambios de iluminación que se presentan en el ambiente es necesario implementar un algoritmo que permita modificar estos cambios bruscos en la iluminación del video, debido a que es de suma importancia para así reducir la complejidad del algoritmo de seguimiento de objetos. Para esto se usó un algoritmo *Correcting Nonuniform Illumination* de los ejemplos de MatLab. Secuencia para el proceso de corrección de iluminación:

- *Lectura de la imagen:* Se usa la sintaxis `imread` donde me genera una matriz de 0 a 255.
- *Estimar el fondo:* Utilizan la instrucción `imopen` con un elemento estructurante disco con un diámetro de 15, para estimar la iluminación de fondo.
- *Extracción de objetos:* Se realiza una diferencia entre la imagen original y el fondo
- *Aumentar el contraste:* La instrucción `imadjust` permite ajustar los valores de intensidad de la imagen.
- *Umbral de la Imagen:* Crea una imagen binaria por método de umbralización y elimina las áreas que son menores al especificado por medio de `bwareaopen`.
- *Identificar los objetos:* La función `bwconncomp` busca todos los componentes conectados (objetos) en una imagen binaria [23].

### 2.4.1.2 Motion-Based Multiple Object Tracking

Sistema de seguimiento de múltiples objetos en un video, extraído de los ejemplos de MatLab, este algoritmo utiliza la técnica de sustracción de fondo para la detección de los objetos en movimiento. El fondo es modelado con una mezcla de modelo Gaussiano extrayendo en primer plano (objetos) para eliminar los píxeles que no hacen parte de ningún objeto se aplican las operaciones morfológicas a la máscara de primer plano eliminando el ruido. Una vez son eliminados estos píxeles, se detectan los grupos de píxeles conectados por medio del análisis de manchas o blob donde es posible hallar la posición de los objetos y así verificar si corresponden a los objetos en movimiento.

Para predecir el movimiento de un objeto en una secuencia de video, se utiliza el filtro de Kalman, este es necesario cuando se encuentran obstáculos que no permiten la visibilidad del movimiento del objeto por ejemplo un árbol en el video, el filtro determina la probabilidad de cada detección y se la asigna a cada pista. Cada track lleva la cuenta del número de cuadros consecutivos, donde permaneció sin asignar, si la cuenta excede un umbral especificado se supone que el objeto salió del campo de visión y se borra la pista esto es un problema para el algoritmo de monitoreo de vehículos puesto que si un vehículo permanece

por unos instantes sin movimiento el filtro lo elimina y cuando este se mueva se cuenta como otro carro más.

El algoritmo está desarrollado en diferentes funciones secuencialmente de la siguiente forma:

### 1. *Función Crear Objetos*

Se crean los objetos de lectura del video cargando un archivo usando la sintaxis *vision.VideoFileReader*, para generar los objetos de reproducción del video se usa *vision.VideoPlayer* especificando la posición en pantalla del video.

Posteriormente se crean los objetos para el detector del primer plano y para verificar que píxeles corresponden a un objeto se usa *vision.BlobAnalysis* donde se especifica el mínimo de píxeles conectados en un área, así mismo muestra el centroide y las coordenadas de las cajas que encierran los objetos encontrados.

### 2. *Función InitializeTracks*

Es una estructura que permite mantener el estado de un objeto rastreado, la estructura contiene los siguientes campos:

Id: número del objeto.

Bbox: El cuadro de límite actual del objeto.

KalmanFilter: Un objeto de filtro de Kalman

Age: El número de tracks desde que se detectó la pista primera.

TotalVisibleCount: Número total de tramas.

ConsecutiveInvisibleCount: El número de tramas consecutivas en las que no se detectó movimiento.

### 3. *Funcion Detect Objects*

En esta función se detectan los objetos de primer plano y se aplican las operaciones morfológicas correspondientes para eliminar ruido y llenar huecos.

### 4. *Función predictNewLocationsOfTracks*

Predice la ubicación del objeto en el track por medio del filtro de Kalman y el centroide hallado con anterioridad.

### 5. *Funcion displayTrackingResults*

Esta función es el paso final del algoritmo donde visualiza un cuadro delimitador de cada objeto y el ID de etiqueta para cada pista de un fotograma de video [24].

### 3 DESARROLLO

Este proyecto se realizó con el fin de monitorear los carros que se movilizan por las vías, las características de este algoritmo son: Los objetos de interés son los carros además de esto no realiza reconocimiento de placas, ni características físicas de los carros como; color, tipo de carro entre otros, para evaluarlo se cargan videos propios, no en tiempo real.

Para el desarrollo e implementación de este sistema de visión artificial se hace uso de una herramienta de software matemático MATLAB 2015b, permitiendo así la implementación de un algoritmo para la detección de los vehículos para este fin se usa los toolbox de MatLab; Image Processing, Computer Vision y para la creación de la interfaz de usuario se usa la herramienta GUIDE.

Para el desarrollo práctico del proyecto se trabajó con un video obtenido de un sistema de cámaras de vigilancia de tráfico IP. Para simular el tráfico vehicular y las infracciones de tránsito se empleó Unity, y Ptv Vissim.

#### 3.1 Descripción general del sistema

Los elementos de la aplicación se encuentran distribuidos como muestra en la Figura 7 Esquema general dividido en tres bloques principales: a. Bloque de Multimedia: Se carga previamente el video que será procesado. b. Procesamiento: En este se realiza la obtención, sustracción y detección de fondo, etc. c. Control de usuario: Este bloque complementa al de procesamiento encargado de tomar las decisiones del algoritmo para la generar un informe de los vehículos que transitan en el video.

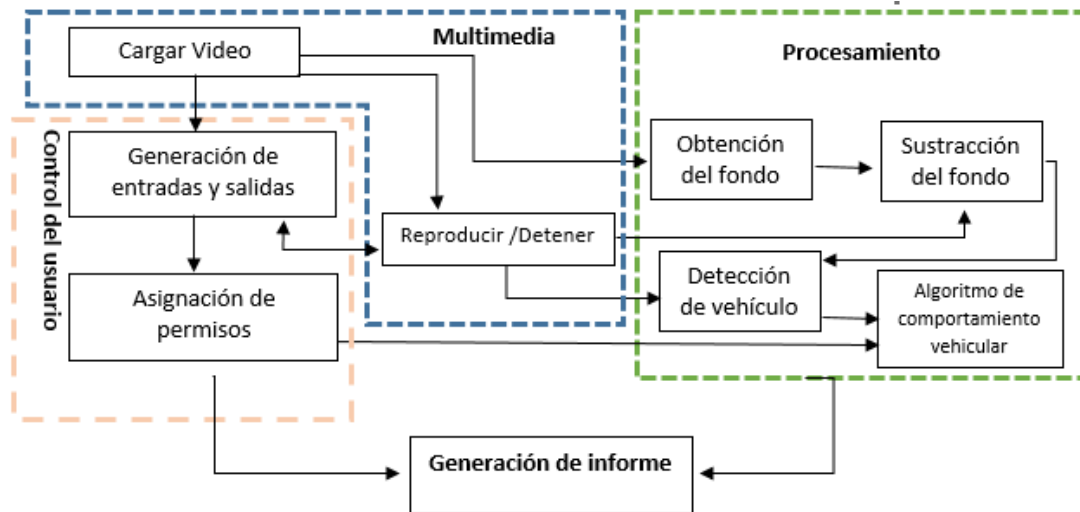


Figura 7 Esquema general

Referencia: Elaboración propia

### 3.2 Captura de video

El video real implementado se adquirió de una cámara IP con una duración de 33 segundos, el cual debía unas características del ambiente especiales como:

- A. Condición climática: Se debía tomar en horas del día, especialmente nublado debido a que un día soleado genera sombras en los objetos que se encuentran en movimiento y dificulta el proceso de segmentación.
- B. Posición de la cámara: La cámara se ubica en la vista superior al cruce vehicular como se muestra en la Ilustración 1 Posición de la cámara.

### 3.3 Segmentación

El objetivo de esta etapa es separar los carros del fondo como se ve en la Figura 8 Fotograma original donde el vehiculo entrante hace parte del fotograma original, para esto se empleó el método de umbralizado dando como resultado extracción de los carros en una imagen binaria Figura 9 Segmentación de un fotograma antes de aplicar las operaciones morfológicas.

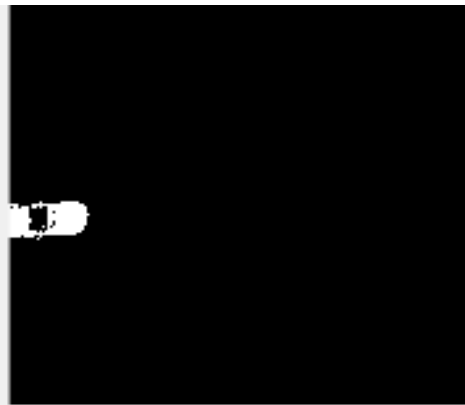
Para que el resultado de la sustracción de fondo sea bueno es importante eliminar los píxeles aislados de la imagen como los que se pueden observar en la Figura 9 Segmentación de un fotograma antes de aplicar las operaciones morfológicas, para esto se utilizan las *operaciones morfológicas*, en el desarrollo de este proyecto se aplica inicialmente Figura 10 Dilatación con un elemento estructurante de 3x3 de forma rectangular como se observa esta operación permite abrir los píxeles

aislados que no se mostraban para luego eliminarlos , luego una Figura 11 Apertura con un elemento de 3x3 rectangular permitiendo disminuir la intensidad de los pixeles a eliminar y finalmente Figura 12 Cierre de 3x3 de forma rectangular que permite conectar los pixeles que se encuentran dentro de la imagen carro, en la Figura 13 Selección área se aplica un filtro para eliminar pixeles cuya área sea menor a 100 pixeles conectados y se puede apreciar una imagen binaria sólida.



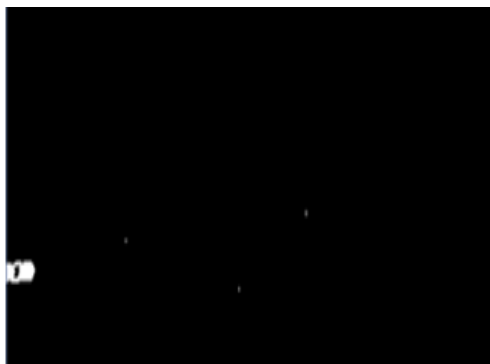
*Figura 8 Fotograma original*

*Referencia: Elaboración propia*



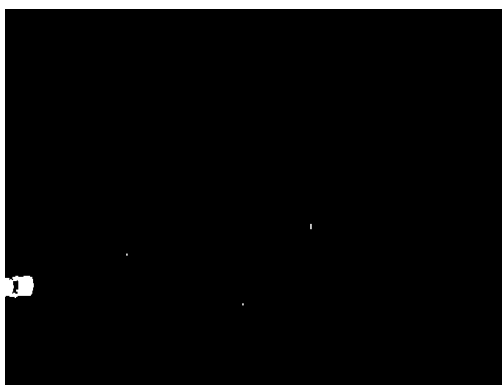
*Figura 9 Segmentación de un fotograma antes de aplicar las operaciones morfológicas*

*Referencia: Elaboración propia*



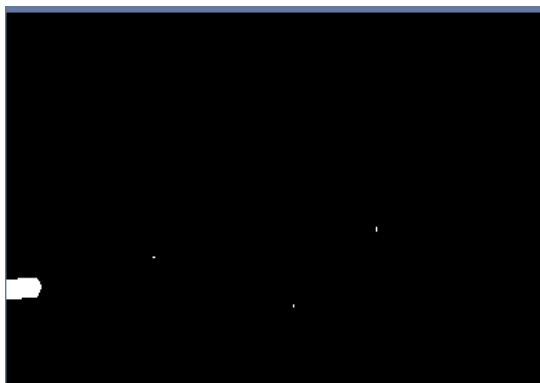
*Figura 10 Dilatación*

*Referencia: Elaboración propia*



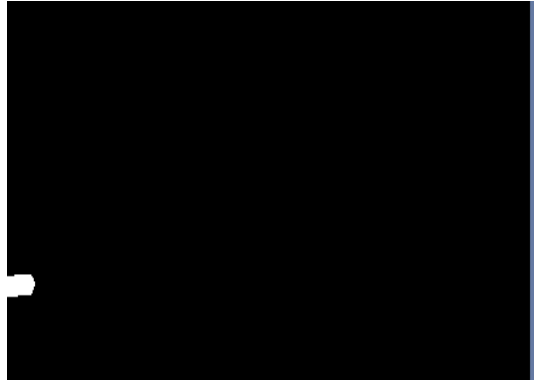
*Figura 11 Apertura*

*Referencia: Elaboración propia*



*Figura 12 Cierre*

*Referencia: Elaboración propia*



*Figura 13 Selección área*

*Referencia: Elaboración propia*

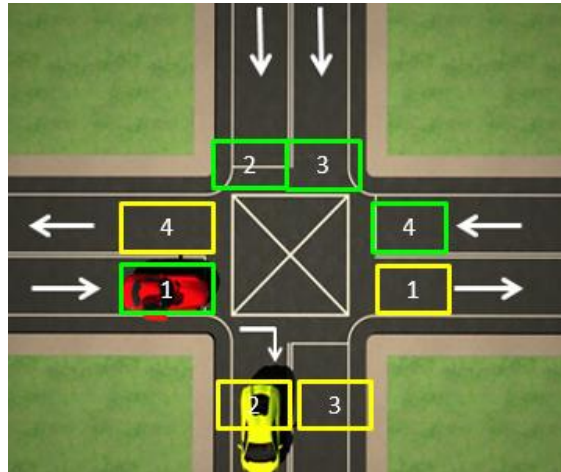
### 3.4 Determinar Trayectoria e Identificación de Infracciones de tránsito

La aplicación consta de dos estrategias para el funcionamiento la primera es como determinar la trayectoria de los vehículos por tanto se seleccionó una estrategia de zonas de interés. La segunda es para determinar las infracciones de tránsito para esto se hace una verificación de los datos ingresados por el usuario que permitirán identificar si el vehículo comete algunas de las infracciones propuestas.

#### 3.4.1 Elementos de Entrada – Salida

Las variables de entrada y salida del algoritmo son los elementos que el usuario determina en la Figura 14 Entradas (Verde)/ , se pueden observar las entradas de color verde y las salidas de color amarillo en cada una de las vías, teniendo en cuenta la dirección en la que transitan los vehículos, las cuales son objeto de análisis para determinar la trayectoria de los carros y las infracciones cometidas, para este fin se crean unos rectángulos de entrada y salida ubicados en un fotograma del video, creados por el usuario debe determinar: alto, ancho y posición en el fotograma. Además de esto se pueden ingresar la cantidad de rectángulos que el usuario desee y de la misma manera puede eliminarlos.





*Figura 14 Entradas (Verde)/ Salidas (Amarillo).*

*Referencia: Elaboración propia*

**Rectángulo de entrada:** Es una región por donde inicialmente empezaría hacer detectado el vehículo, es decir por donde entra el carro en alguna de las vías teniendo en cuenta las características el lugar y los cruces permitidos.

**Rectángulo de salida:** Es la región por donde el vehículo empieza a dejar de ser detectado y termina su desplazamiento.

Para la creación de un rectángulo inicialmente se escoge uno de los píxeles de una esquina de donde se quiere que se inicie el rectángulo luego se escoge la esquina diagonal para obtener el tamaño para finalizar se intersectan los píxeles horizontales y verticales a estas marcas.

### 3.5 Estrategia para determinar trayectoria

La estrategia usada para determinar la trayectoria de un vehículo, se realiza por medio de los rectángulos creados, para verificar la zona por la que el vehículo se desplazó se revisa inicialmente el rectángulo de entrada por donde debería ser identificado desde que empezó hacer detectado y respectivamente se revisan los demás rectángulos creados ya sea una entrada o una salida respectivamente, de esta manera se monitorean cada uno de los vehículos. Estos dos rectángulos son comparados por medio de la intersección entre las dos cajas ya sea la de entrada o salida y la caja que encierra el objeto detectado, con esto es posible determinar la región por la que se desplazó el vehículo. Usando la ecuación presentada en la Figura 15 Calcular la intersección entre dos cajas, se calcula el área de las cajas luego obtiene la intersección entre el área de cada caja ( $bboxA$  y  $bboxB$ ), dividido por el área mínima de las dos cajas, esta intersección debe ser mayor al 75% en este caso para el algoritmo.

En la Figura 16 Interacción entre un vehículo y una salida se capta el momento exacto en el que un vehículo con su respectiva marca es intersectado por una salida.

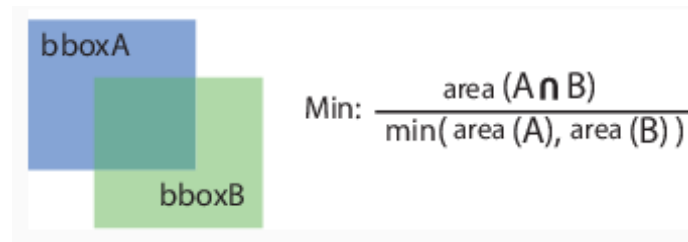


Figura 15 Calcular la intersección entre dos cajas

Referencia: [25]

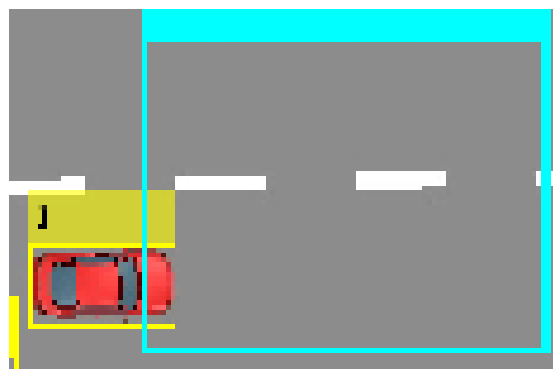


Figura 16 Interacción entre un vehículo y una salida

Referencia: Elaboración propia

La instrucción de MatLab para encontrar la intersección de dos cajas es *overlapRatio* esta función devuelve un valor entre cero y uno, siendo uno la relación perfecta entre la superposición de las cajas.

### 3.6 Estrategias Para Identificar Infracciones De Transito

La estrategia usada para determinar las infracciones de tránsito depende de las reglas de movimiento que el usuario inicialmente determinó es decir, que el algoritmo identifica como movimiento permitido todo aquello que el usuario haya marcado, de esta manera se obtendrán los resultados en el informe de toda la aplicación. Las infracciones que son identificadas en la aplicación son:

### 3.6.1 Cruce indebido

En el Pseudo código 1 se realizan los pasos lógicos para la toma de decisión en cuanto a esta infracción, el vehículo que es detectado realizando un giro indebido o no permitido es marcado como infractor. La  $(Pos\_Entrada) || (Pos\_Salida) * (Permiso) = 0$  Ecuación 1 evalúa si se cometió un cruce indebido teniendo en cuenta la posición de entrada y salida del vehículo además de esto los permisos de movimiento que el usuario determino.

$$(Pos\_Entrada) || (Pos\_Salida) * (Permiso) = 0 \quad \text{Infracción Ecuación 1}$$

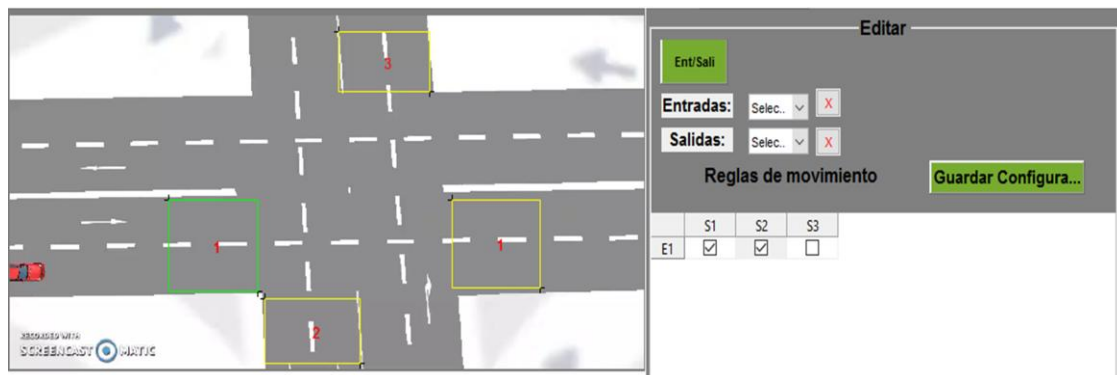
En la imagen de la izquierda se generó una entrada y las posibles salidas para la dirección de este vehículo y en la imagen de la derecha en reglas de movimiento se determinó que el vehículo que cruce por la entrada 1 solo debe circular por la salida 1 y 2, si algún vehículo sale por la salida 3 estaría cometiendo un giro indebido.

SI  $(Posición\ de\ entrada\ o\ la\ posición\ de\ salida)$  tiene permiso >  
Vehículo = "no tiene esta infracción"

SI NO  
Vehículo = "infracción por cruce indebido"

FIN SI

*Pseudo código 1: Estructura de condición para definir la infracción de cruce indebido*  
*Referencia: Elaboración propia*



*Figura 17 Entrada y salidas*  
*Referencia: Elaboración propia*

### 3.6.2 Obstrucción

Son aquellos vehículos que permanecen inmóviles por un tiempo y obstruyen el paso de los demás carros. La

$\#Fotogramas > Promedio Fotogramas por vehículos con Entrada \cong 0$

0 Ecuación

$2\#Fotogramas > Promedio Fotogramas por vehículos con Entrada \cong 0$

determina cuando se ha cometido obstrucción en la vía, cuando el número de fotogramas que ha recorrido cada carro es mayor que promedio de estos fotogramas, sabiendo que el vehículo ya entró por un rectángulo de entrada permitido, es decir diferente de cero este ha cometido una infracción.

$\#Fotogramas > Promedio Fotogramas por vehículos con Entrada \cong 0$  Ecuación 2

### 3.6.3 Contravía

Esta infracción se comete cuando un vehículo inicialmente entra por un rectángulo de salida, es decir que M (momento) de entrada sucede en un tiempo superior al momento M de salida.

$M Entrada > M Salida = Infracción$  Ecuación 3

## 3.7 Procesamiento De Imágenes

### 3.7.1 Pre-Procesamiento

En esta etapa se convierten las imágenes de entrada a escala de grises utilizando la función *im2bw* de MatLab con un porcentaje del 25% que existe en entre el fondo y la imagen procesada.

## 3.8 Seguimiento de Objetos

Para realizar el seguimiento de los carros detectados en las secuencias de los videos se acude a la técnica de componentes conectados la cual encuentra esos grupos de píxeles que probablemente corresponde a un carro y le extrae características como el área, centroide y la caja que envuelve el objeto

La función utilizada para determinar el centroide y el tamaño de la caja que rodea los objetos es:

```
[~, centroids, bboxes] = handles.obj.blobAnalyser.step(mask);
```

La técnica basada en filtro de Kalman permite obtener un modelo de la posición del objeto en cada instante del video, en este proyecto se implementa este filtro

con el fin predecir trayectoria cuando exista con un objeto en la escena que no permita la visibilidad de la trayectoria del vehículo.

### 3.9 Interfaz Grafica

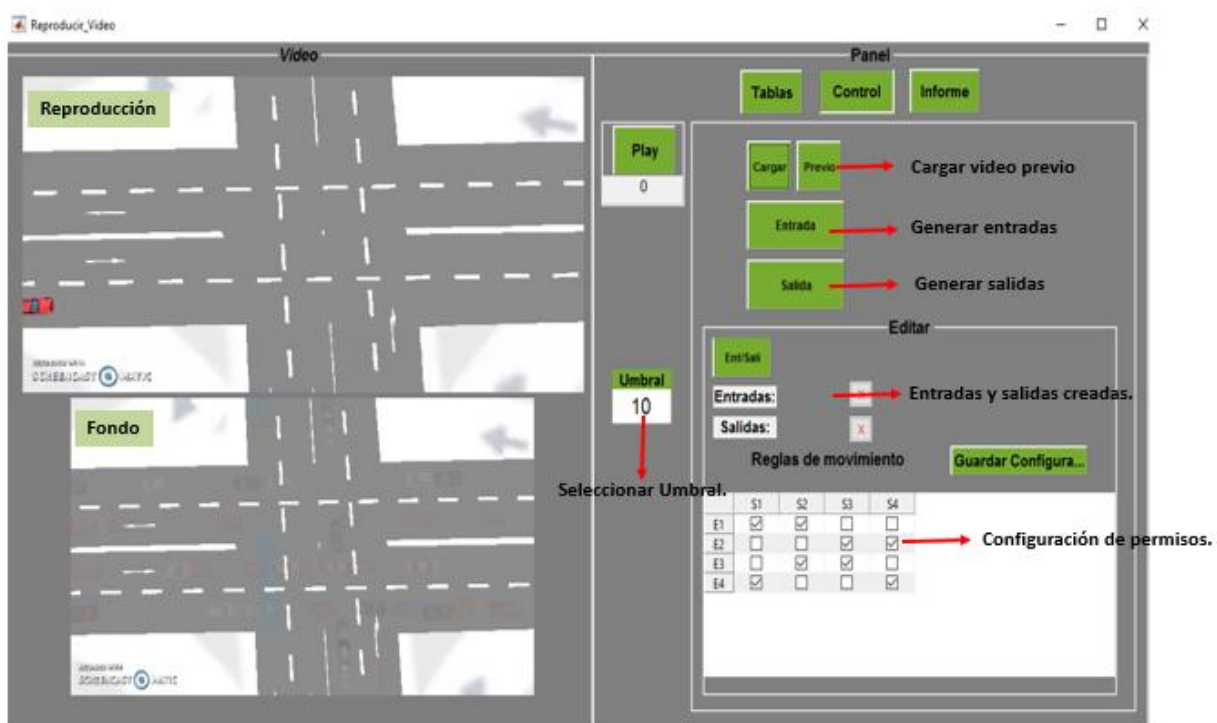


Figura 18 Interfaz Gráfica  
Referencia: Elaboración propia

En la Figura 18 Interfaz Gráfica permite en primer lugar al usuario tener una interacción con los procesos que complementan los algoritmos, debido a que determina las variables de entrada y salida de este obteniendo un informe sobre la cantidad de vehículos, cuántos y cuáles de estos cometieron alguna o más infracciones de tránsito. En segundo lugar visualizar los datos obtenidos por los algoritmos permitiendo que el diseño gráfico del aplicativo sea más amigable para el usuario final.

El diseño de la interfaz está dividido en:

### 3.9.1 Panel de control:

Los controles de multimedia están distribuidos de acuerdo a su función haciendo intuitivo para cargar videos nuevos o pre- establecidos. Esta función se hace por medio de los botones *cargar* y *previo* ubicados en el área superior de la sección de panel de control junto a los botones que se utilizan para generar entradas y salidas en los videos, como se visualizan en la Figura 19 visualización de los botones del panel de control.



Figura 19 visualización de control

los botones del panel de

Referencia: Propia del autor

#### 3.9.1.1 Edición:

Siendo esta sub-área del panel de control de gran importancia para el proceso, puesto que es aquí en donde el usuario toma las decisiones de editar el número de entradas o salidas que requiera el video para su procesamiento en la Figura 20 Edición de entradas y salidas. Se puede observar los objetos desplegables donde se almacenan las entradas y las salidas.



Figura 20 Edición de entradas y salidas.

Referencia: Elaboración propia

#### 3.9.1.2 Tabla de comportamiento:

Al igual a como se ve en un cruce vehicular en la Figura 20 Edición de entradas y salidas. se generan las reglas o permisos con un objeto booleano facilitando consentimientos de movimiento en determinada dirección, esto con el fin de completar las variables del algoritmo que se encarga de generar el

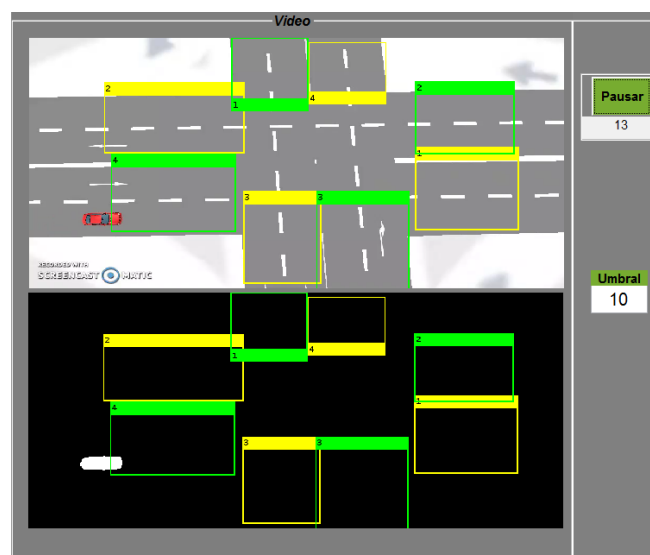
comportamiento que han tenido los vehículos detectados, esta tabla funciona al igual que una matriz booleana en donde las filas son interpretadas como las entradas mientras que las columnas son interpretadas como las salidas, siendo el uno lógico (1) como el permiso concedido y el cero lógico (0), como denegado, por tanto si un vehículo sale por un cero es considerado como un infractor.

Reglas de movimiento					Guardar Configura...
	S1	S2	S3	S4	
E1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
E2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
E3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
E4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

*Figura 21 Tabla comportamiento*  
Referencia: Elaboración propia

### 3.9.1.3 Visualización:

En el sistema en la parte izquierda de la interfaz se encuentran los ejes de visualización de la aplicación en donde se observan los videos previamente y durante todo el proceso del tratamiento de las imágenes. En el axes superior (Ax\_video) de la Figura 22 Display de visualización se observa el video en formato RGB mientras que en el axes inferior (Ax\_Segmentado) de esta misma figura, primero se observa la sustracción de fondo y una vez se le ejecuta la función de reproducir con el botón de play (Bnt\_play/pause) se visualiza la segmentación de cada fotograma en formato monocromático (blanco y negro).

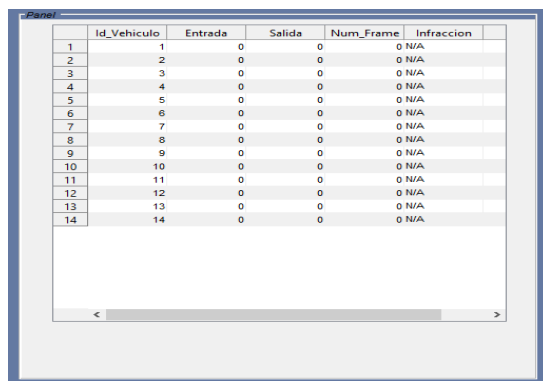


*Figura 22 Display de visualización*

Referencia: Elaboración propia

#### 3.9.1.4 Visualización en tiempo de procesamiento:

La Figura 23 Información en tiempo de procesamiento muestra una tabla de manera dinámica obtiene datos en tiempo de ejecución del video, en la que visualiza la información del número asignado al vehículo, por donde entra, por donde sale, cuantos fotogramas ocupa en el video a partir desde donde es detectado y el tipo de infracción que comete en el caso de que así sea, de lo contrario en esta celda deja los caracteres N/A indicando que no ha cometido ninguna infracción.



	Id_Vehiculo	Entrada	Salida	Num_Frame	Infraccion
1	1	0	0	0	N/A
2	2	0	0	0	N/A
3	3	0	0	0	N/A
4	4	0	0	0	N/A
5	5	0	0	0	N/A
6	6	0	0	0	N/A
7	7	0	0	0	N/A
8	8	0	0	0	N/A
9	9	0	0	0	N/A
10	10	0	0	0	N/A
11	11	0	0	0	N/A
12	12	0	0	0	N/A
13	13	0	0	0	N/A
14	14	0	0	0	N/A

Figura 23 Información en tiempo de procesamiento

Referencia: Elaboración propia

#### 3.9.1.5 Botones de navegación:

Es por medio de estos se ubican cualquiera de las secciones o sub- secciones anteriormente nombradas ya sean las tablas dinámicas, el panel de control o la generación de un informe.

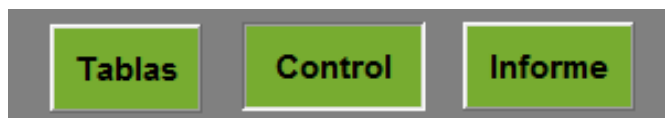


Figura 24 Botones de navegación

Referencia: Elaboración propia

### 3.9.2 Informe:

Es una conexión a un documento en Excel como se ve la Figura 25 Reporte donde quedan registrados todos datos de trayectoria y las infracciones cometidas de los vehículos detectados en el video.



	A	B	C	D	E	F	G
1							
2							
3							
4		Id_Vehiculo	entrada	salida	Nuemo de f	tipo de infrccion	
5		1	0	0	0	N/A	
6		2	0	0	0	N/A	
7		3	0	0	0	N/A	
8		4	0	0	0	N/A	
9		5	0	0	0	N/A	
10		6	0	0	0	N/A	
11		7	0	0	0	N/A	
12		8	0	0	0	N/A	
13		9	0	0	0	N/A	
14		10	0	0	0	N/A	
15		11	0	0	0	N/A	
16		12	0	0	0	N/A	
17		13	0	0	0	N/A	
18		14	0	0	0	N/A	
19		15	0	0	0	N/A	
20		16	0	0	0	N/A	
21		17	0	0	0	N/A	
22		18	0	0	5	N/A	
23		19	0	0	0	N/A	
24		20	0	0	23	N/A	

Figura 25 Reporte

Referencia: Elaboración propia

### 3.10 Descripción Del Algoritmo

En el siguiente diagrama de flujo FFigura 26 Algoritmo de detección de vehículos. se evidencia el proceso general para la detección de los vehículos, pasos basados el Pseudo código 2 en el cual se explica de manera general como se obtiene los resultados en la segmentación de cada fotograma del video, desde el momento en el que se carga el video, hasta la generación del reporte :

Iniciar la aplicación

Leer imagen

*Pseudo código 2.1.: bucle repetitivo para definir entradas y salidas*

```
PARA < i=1 HASTA cantidad_de_fotogramas en pasos de 1 >  
    Escala de grises= fotograma en escala de grises  
    Definir entradas y salidas  
FIN PARA I
```

*Pseudo código 2.2: bucle repetitivo para aplicaciones morfológicas*

```
PARA j=1 HASTA cantidad_de_fotogramas en pasos de 1  
    Imagen= (escala de grises – fondo)  
    Aplicación morfológica.
```

*Pseudo código 2.2.1: Estructura de condición para etiquetar y predecir movimiento*

```
SI (detecta movimiento)  
    SI (existe el objeto identificado)  
        Etiquetar objeto  
    SI NO  
        Crear etiqueta  
    FIN SI  
SI NO  
    Predice movimiento  
FIN SI
```

*Pseudo código 2.2.2: Estructura de condición para eliminar etiquetas y predecir movimiento*

```
SI área<75  
    Etiqueta =0;  
SI NO  
    Prededir  
FIN SI
```

Generar reporte

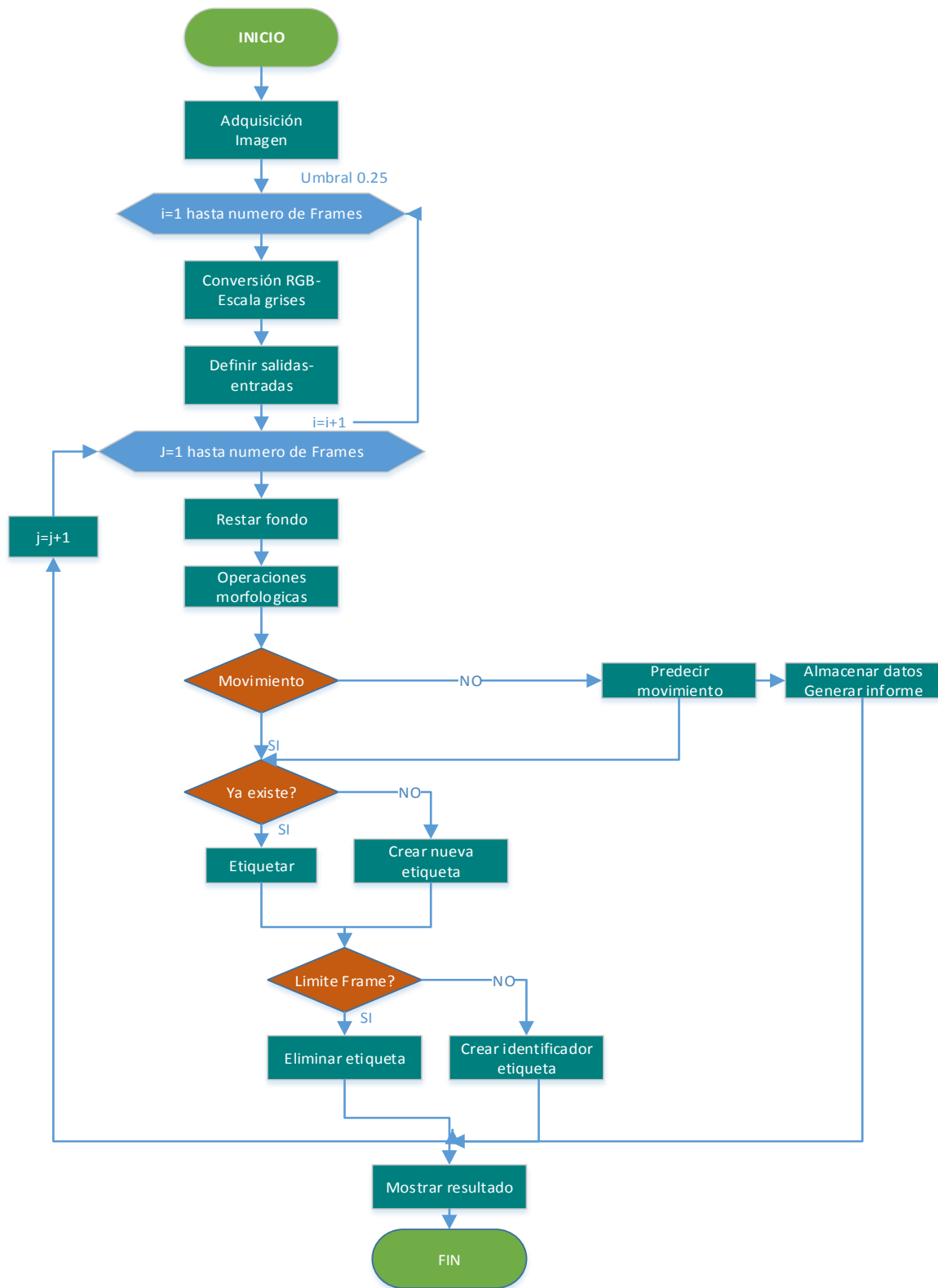


Figura 26 Algoritmo de detección de vehículos.

Referencia: Elaboración propia

### 3.10.1 Cargar Video

En esta sección de la aplicación el usuario puede cargar un video nuevo o uno que esta predeterminado, el formato que acepta MatLab es mp4,avi o mov..

*Cargar Video pre establecido:* Al ejecutar la función “btn\_previo” el sistema busca en la carpeta de multimedia el video con el nombre previo.Se carga el video en el objeto llamado `vob` luego se obtiene la información relacionada con cada fotograma esto se realiza con la función `read(inf) NumberOfFotogramas`

Se crean “2” matrices de ceros “0” con el mismo tamaño de cada fotograma una para la densidad y una donde se guardará la diferencia de los del el fotograma original y el fotograma de la densidad. Código Anexo 2 Obtener la densidad de color del video

Inicialmente creamos un objeto de tipo archivo con la función `uigetfile`, que se encarga de obtener la ruta buscada con el explorador de la ventana emérgete al presionar el botón (`Btn_Cargar`).

Una vez cargado se llama a la función expresada en el Pseudo código 3: Bucle repetitivo para obtener la densidad de los píxeles de un fotograma.que se encargan de extraer el fondo. Para generar manualmente las entradas y las salidas en las funciones de los botones “`Btn_Entrada` y `Btn_Salida`” que activan posteriormente la función que reconoce la posición del ratón dentro del axes “`Ax_Video`”.

En el Pseudo código 3 Se procesa un bucle repetitivo que recorre todo el video con el fin de obtener la densidad de cada pixel en un fotograma quedando como resultado el fondo a extraer de todo el video. Código Anexo 2 Obtener la densidad de

*Pseudo código 3: Bucle repetitivo para obtener la densidad de los píxeles de un fotograma.*

*Referencia: Elaboración propia*

```
PARA < k=1 HASTA números_de_fotogramas en pasos de 1 >
    Diferencia =1- (la conversión a escala de grises con un umbras del 25% )
    Densidad =densidad + diferencia
    Acumulación de fotogramas= Acumulación de fotogramas+ (1- diferencia)
    Acumulación_de_fotogramas=Acumulación_de _fotogramas / (números_de_fotogramasa)
    Fondo =fondo + fotograma(k) del video
```

```
FIN PARA k
```

color del video

Posteriormente se aplica un filtrado a cada capa del video RGB y es guardado en la matriz fondo y mostrado en pantalla.

### 3.10.2 Entradas / Salidas

La asignación del número de entradas y salidas se establece mediante las coordenadas obtenidas del axes "Ax\_Video", los cuales tienen unos valores en x, y para cada punto de intersección en donde se creará un rectángulo, teniendo en cuenta la imagen del fotograma. Para generar dichos rectángulos se hace uso de la función *Rectangle* a la cual se le asigna una posición en x, y una posición y para formar el punto de origen desde donde inicia, se le asigna la magnitud horizontal, la magnitud vertical y color del rectángulo.

Sintaxis:

```
rectangle('position', posicion_en_X, posicion_en_y, magnitud_X, magnitud_Y, 'EdgeColor', color);
```

El usuario debe configurar muy bien los siguientes parámetros: Ubicación de los rectángulos de entrada/ salida y las dimensiones de los rectángulos de tal forma que adapten al ancho y alto de los objetos a detectar. Código Anexo 5 Generador de un rectángulo

### 3.10.3 Detección y Predicción

#### 3.10.3.1 Detección de Vehículos:

Para esta función inicialmente se aplica la diferencia entre el fondo y la matriz del video, luego con un umbral establecido por el usuario se obtiene una imagen en blanco y negro en donde los píxeles negro corresponden al fondo y los blancos a los vehículos detectados, de esta forma se aplican las respectivas operaciones morfológicas para dilatar, abrir y cerrar los píxeles que se desean resaltar o eliminar los que se consideran ruido. Con la función *blobAnalyser* se obtiene los centroides y los recuadros de cada vehículo. Código [Anexo 8](#)

Generalmente en los cruces vehiculares conocidos como esquinas el paisaje se encuentra rodeado ya sea de postes eléctricos o de árboles, que entorpecen el sistema de detección por tanto, se hace necesario implementar una función encargada de predecir la futura posición de un vehículo la función que se utiliza para este proyecto es *KalmanFilter* teniendo en cuenta la configuración de este como una aceleración constante.

El Pseudo código 4 explica cómo se asigna la predicción de los track perdidos en un fotograma. Código Anexo 14 Predicción de vehículos

*Pseudo código 4: Bucle repetitivo para aplicar el filtro de Kalman en un fotograma.  
Referencia: Elaboración propia*

```
PARA < i= 1 HASTA número_tracks >  
    Caja= caja del track (i)  
    Predicción del centroide=Aplicación filtro de Kalman(Caja)  
    Generar caja de 3x4 (Predicción del centroide)  
FIN PARA i
```

A cada vehículo detectado se le asigna una identificación y se envuelve en un recuadro, este id se utiliza para contar los objetos que reconoce el algoritmo y conocer el comportamiento del objeto como; su dirección y trayectoria para posteriormente calificadas en el reporte generado en Excel esto se logra a través de las funciones generar track, asignar track y eliminar track.

Antes de eliminar un track se debe dar de baja el recuadro con el fin de que si aparece nuevamente porque fue obstruido no se genere uno nuevo y mantenga el mismo número de identificación que se le asignó al momento de ser detectado por primera vez.

Para finalmente eliminar un track y sus características se utiliza la siguiente función.

Sintaxis:

```
[assignments, unassignedTracks, unassignedDetections] =  
detectionToTrackAssignment(),
```

Esta se encarga de verificar primero si existen dichos tracks. *Código Anexo 11,12,13,14,15*

#### 3.10.4 Llenar tabla

La tabla permite al usuario observar cómo se comporta el algoritmo para cada uno de los vehículos detectados tiene un id o identificador el cual permite al usuario realizar un seguimiento visual de la trayectoria de cualquiera de estos. Estos datos son tomados de forma dinámica al cambio de cada fotograma, aunque sus datos no son los definitivos para el informe que una vez generado, esta tabla realiza una actualización para cargar los vehículos infractores por obstrucción ya que esta infracción depende del promedio y duración en fotogramas del cada vehículo en el video.

	Id_Vehiculo	Entrada	Salida	Num_Frame	Infraccion	
1	1	2	1	45	N/A	^
2	2	4	3	1	N/A	
3	3	1	3	29	Contravía	
4	4	4	3	56	N/A	
5	5	2	2	205	Contravía	
6	6	1	1	51	N/A	
7	7	2	3	76	Contravía	
8	8	2	1	124	N/A	
9	9	2	1	68	N/A	
10	10	1	1	4	N/A	
11	11	2	1	42	Contravía	
12	12	2	1	57	N/A	
13	13	4	2	3	Contravía	
14	14	2	1	55	N/A	
15	15	2	3	18	Contravía	
16	16	2	1	99	N/A	
17	17	2	1	120	N/A	
18	18	0	0	0	N/A	
19	19	2	0	30	N/A	
20	20	3	0	49	Contravía	v
<div> <div>&lt;</div> <div></div> <div>&gt;</div> </div>						

*Figura 27 Tabla Información general  
Referencia: Elaboración propia*

Esta tabla también es tomada en cuenta para almacenar los acontecimientos en orden, con esto se quiere decir que si un vehículo pasa por una salida antes de que pase por una entrada la tabla lo interpretará como una falta o infracción.

### 3.10.5 Identificar la trayectoria del vehículo

Basados en la tabla de identificación de vehículos cuando se genera una nueva detección las entradas y las salidas son cargadas con cero y al momento de detectar un cruce por algunas de las zonas creadas por el usuario ya sean salidas o entradas se determina su comportamiento verificando que inicialmente debe activar una única entrada y una única salida de lo contrario se genera una infracción.

### 3.10.6 Generación Informe

Para generar el informe en Excel con el fin de obtener los datos registrados en la tabla de la interfaz y ejecutar los algoritmos de las infracciones por medio del botón *Btn\_Informe* y la sentencia *xlswrite*, se crea un documento .xlsx donde se guardan a partir de la columna B5 el id de vehículo, reporte de entrada, reporte de salida, numero de fotogramas de cada vehículo e infracciones cometidas si este es el caso, este documento es guardado en la carpeta predeterminada de MatLab.

### 3.10.7 Iluminación

En un video real se visualizan cambios de iluminación debido a las características del ambiente del lugar esto generando cambios en la intensidad de los píxeles que conlleva a perder el fondo de la imagen dificultando el procesamiento de las imágenes y la identificación de los vehículos.

Sintaxis

$g = imadjust(f, [low\_in \ high\_in], [low\_out \ high\_out], gamma)$  [23]

Con esta función se busca ajustar los valores de intensidad mejorando el contraste de imágenes, ya sea para aumentándolo o reduciéndolo. La función  $J=imadjust(I)$  de MatLab mapea los valores de intensidad de la imagen  $I$  a nuevos valores de intensidad en la imagen  $J$  de tal forma que a partir del 1% de los datos se satura para el límite inferior y superior las intensidades de la imagen  $I$ . El hecho de obviar los límites de la imagen  $I$  y considerar un porcentaje fijo permite mejorar el contraste.

Metodología de uso:

Para usar *imadjust*, se deben realizar dos pasos:

1. Ver el histograma de la imagen para determinar los límites del valor de intensidad.
2. Especificar estos límites como una fracción entre 0.0 y 1.0 para que pueda pasarlos a *imadjust* en el vector  $[low\_in \ high\_in]$ .

Una forma más conveniente para hallar estos límites es por medio de la función *stretchlim*. (La función *imadjust* usa *stretchlim* para su sintaxis más simple, *imadjust(I)*.)

Como se muestra en el Pseudo código 5 se usa un bucle repetitivo para recorrer y mejorar cada fotograma del video esta función calcula el histograma de la imagen



y determina automáticamente los límites de ajuste. La función *stretchlim* devuelve estos valores como fracciones en un vector que se puede pasar como el argumento de [low\_in high\_in] a *imadjust* los resultados de este tratamiento se puede observar en al comprar la figura Figura 29 Imagen Original y La figura Figura 28 Ajuste de Iluminación con sus respectivos histogramas.

*Pseudo código 5: Iluminación de un fotograma*

*Referencia: Elaboración propia*

Iniciar la aplicación

Leer fotograma

PARA  $j=1$  HASTA cantidad\_de\_fotogramas en pasos de 1

    Calculo de Histograma

    Fotograma\_Iluminado = corregir iluminación de fotograma ( $j$ )

FIN PARA J

FIN



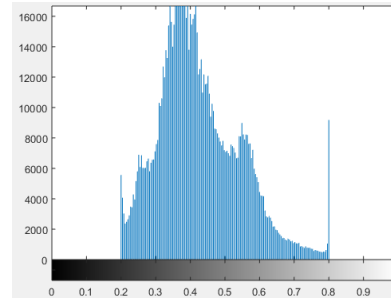
*Figura 29 Imagen Original*

*Referencia: Elaboración propia*

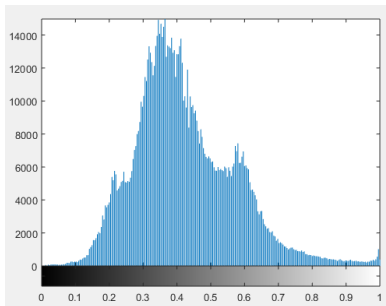


*Figura 28 Ajuste de Iluminación*

*Referencia: Elaboración propia*



*Figura 30 Histograma ajuste iluminación  
Referencia: Elaboración propia*



*Figura 31 Histograma imagen original  
Referencia: Elaboración propia  
Figura 30 Histograma ajuste iluminación*

El eje horizontal representa los niveles de gris en el extremo izquierdo se encuentra el valor más oscuro (negro) y en el extremo derecho el más claro (blanco). El resto de niveles se distribuyen uniformemente. En la Figura 31 Histograma imagen original se obtiene buen contraste debido a que el histograma se extiende ocupando casi todo el rango de tonos. Sin embargo la imagen se observa con una tonalidad oscura para esto se aplica un contraste y el resultado obtenido en la Figura 30 Histograma ajuste iluminación con los tonos oscuros eliminados.

Por defecto, `stretchlim` usa los valores de intensidad que representan el 1% inferior (0,01) y el 1% superior (0,99) del rango como los límites de ajuste. Al ajustar los ambos extremos de la gama de intensidad, `stretchlim` hace más espacio en el rango dinámico ajustado para las intensidades restantes. Pero puede especificar otros límites de rango como un argumento para `stretchlim`.

Existen otras técnicas utilizadas para mejorar la iluminación de las imágenes, entre estas las más utilizadas son: `histeq`, `adaphisteq`, `imsharpen`, `Gaussian3x3`.

### 3.10.8 Identificación de Infracciones

En el siguiente diagrama de flujo se describen las infracciones; cruce indebido y obstrucción vehicular de esta forma es generado el algoritmo para la identificación de las infracciones.

Como complemento a este algoritmo se tiene en cuenta la tabla de permisos asignados por el usuario en donde si la casilla se encuentra marcada en la intersección de una entrada y una salida indica que el vehículo tiene el permiso de circular en esa dirección.

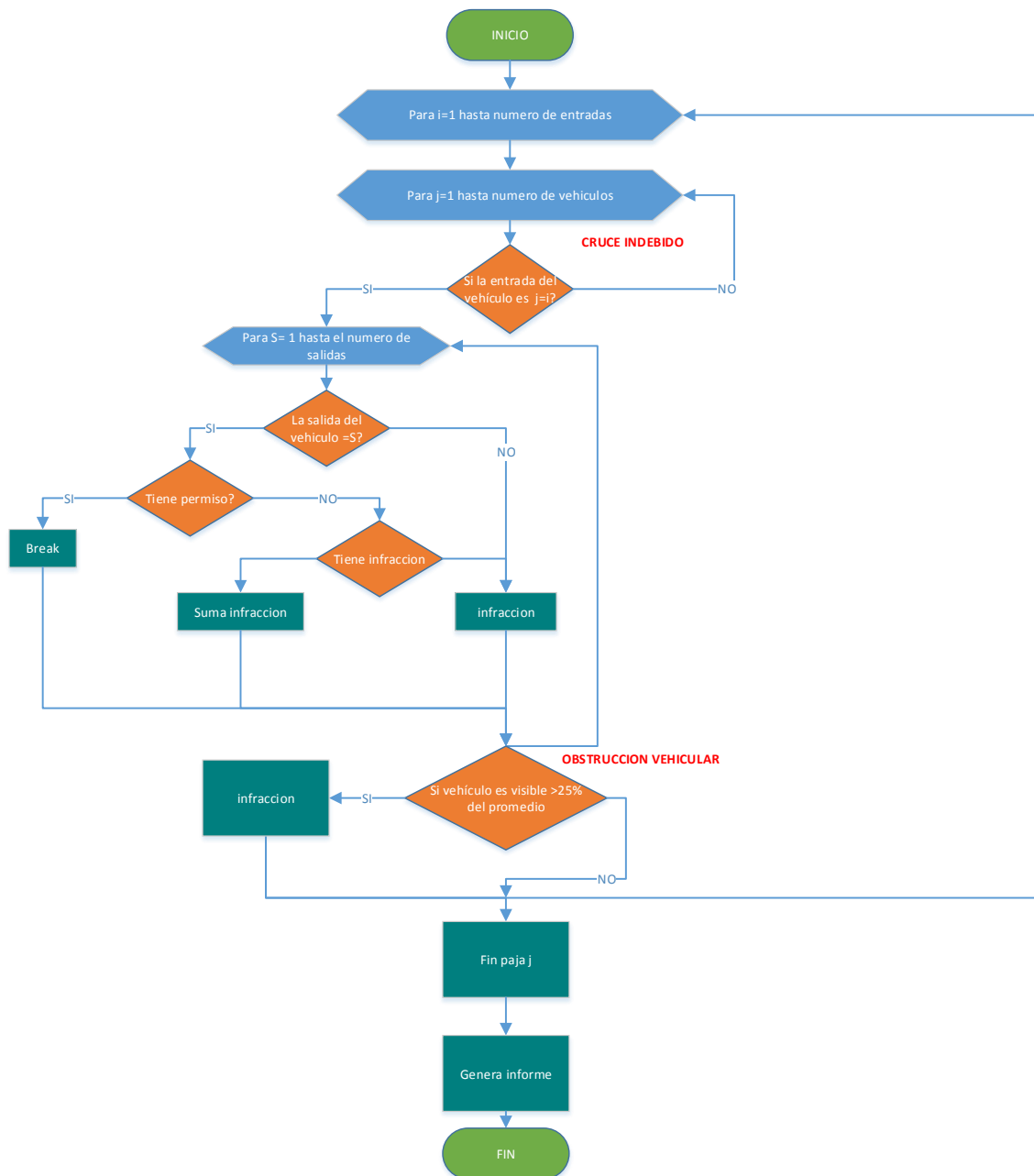


Figura 32 Algoritmo identificación de infracciones

Referencia: Elaboración propia

## 4 RESULTADOS

Para verificar la eficacia de las técnicas implementadas en el algoritmo se realizaron pruebas con videos simulados en Unity obteniendo buenos resultados en cuanto al procesamiento de imágenes y seguimiento de los objetos. Teniendo estos resultados se grabó un video en un cruce vehicular en Bogotá, el lugar de grabación fue un quinto piso pero esta altura no fue suficiente porque los carros se superpusieron y esto generaba problemas al segmentar. Finalmente se optó por videos de cámaras ip en lugares como glorietas y cruces viales en donde se capturo el video real del proyecto.

La evaluación del desempeño del algoritmo se implementó con el siguiente video online en: [26] este fue capturado en horas de la tarde y los videos simulados se realizaron en: PTV VISIM y UNITY, estos con el fin de representar las infracciones de tránsito identificadas por el algoritmo.

Para evaluar la eficacia del algoritmo se realizó un conteo manual de los vehículos infractores que circularon por la rotonda, para obtención de resultados del algoritmo se realizan los siguientes procedimientos:

### *Procedimiento experimental*

1. El uso del sistema inicia y se encuentra en funcionamiento una vez se carga un video en cualquiera de los formatos avi, mp4, wmp.
2. Genera las reglas de movimientos en donde asigna entradas y salidas además de asignar permisos a cada entrada para sus correspondientes salidas.
3. Al dar clic en “play” se retira el fondo e inicia la secuencia de fotograma.
4. Generar reporte.

Para que el sistema funcione correctamente se deben implementar videos, como se muestra en la Ilustración 1 Posición de la cámara que el video sea grabado en un lugar alto. Y el usuario seleccione las entradas y salidas con las dimensiones correctas.

### 4.1 Extracción del fondo

#### 4.1.1 Video simulado UNITY

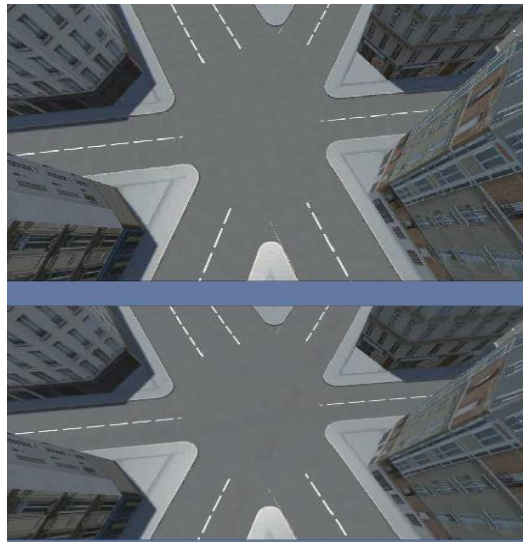
Las características de este video simulado se muestran en la Imagen 1 Características video UNITY, donde la duración del video es de 160 seg y el video procesado por el algoritmo tiene una duración de 720 seg.

Ítem	Sin procesar	Procesado
Duración	160 seg	720 seg
# Frames	4799	
Ancho	854	
Alto	480	

*Imagen 1 Características video UNITY*

Como su puede observar en la Figura 33 superior fondo original, inferior fondo Extracción de fondo simulado UNITY, las condiciones de luz, ubicación de la cámara y la estabilidad de la cámara de esta última son fundamentales para obtener un buen resultado.

En un ambiente ideal la sustracción del fondo como la detección de vehículos es ideal teniendo en cuenta que no se generan sombras y la cámara está completamente fija, aun así la obtención del fondo depende de la cantidad de fotogramas y la taza de muestreo.



*Figura 33 superior fondo original, inferior fondo Extracción de fondo simulado UNITY*

*Referencia: Elaboración propia.*

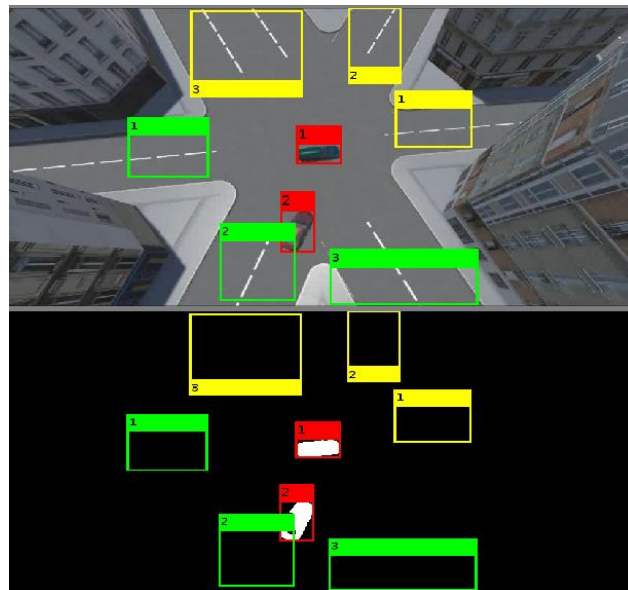
Sin embargo en el eje grafico inferior de la Figura 33 superior fondo original, inferior fondo Extracción de fondo simulado UNITY se observan sombras restantes de la simulación, ruidos aceptables que no generan gran interferencia con el objetivo deseado.

En la Tabla 1 se muestran los resultados obtenidos del reporte de la simulación, la cantidad de *falsos positivos* es nula mientras que en la Figura 34 Resultado de la segmentación simulación n1. se mantiene un fondo completamente negro indicando que la extracción del mismo es exitosa, por tanto los *falsos negativos* también tiene un porcentaje de 0% este video de 30 segundos en la que se simulan las diferentes infracciones de tránsito y en la Tabla 2 se genera el reporte en Excel de donde se obtiene la gráfica del comportamiento vehicular expresados en la Tabla 3 indica que los vehículos detectados no infractores fueron 7, mientras que los infractores por contravía fueron 6, los que realizaron un cruce indebido fueron 0 y los que causaron obstrucción fueron 3, y no se encontró otro tipo de comportamiento .

*Tabla 1: Resultados de falso positivos, falsos negativos, verdaderos positivos y verdaderos negativos*

*Referencia: Elaboración propia*

FP	FN	VP	VN
0	0	12	0



*Figura 34 Resultado de la segmentación simulación n1.*

*Referencia: Elaboración propia*

*Tabla 2: Reporte de la simulación n1*

*Referencia: Elaboración propia*

Id_Vehículo	Entrada	salida	Número de fotograma	Tipo de infracción
-------------	---------	--------	---------------------	--------------------

1	2	1	45	N/A
2	4	3	1	N/A
3	1	3	29	ContravíaCruce Indebido
4	4	3	56	N/A
5	2	2	205	Obstrucción
6	1	1	51	N/A
7	2	3	76	Contravía Cruce Indebido
8	2	1	124	Obstrucción
9	2	1	68	N/A
10	1	1	4	N/A
11	0	0	42	Contravía
12	2	0	37	N/A

Tabla 3: Resultados de comportadito video simulado

Referencia: Elaboración propia

No infractores	Infractores por contravía	por	Infractores por Giro indebido	Infractores por obstrucción	por	Otros
7	6		0	3		0

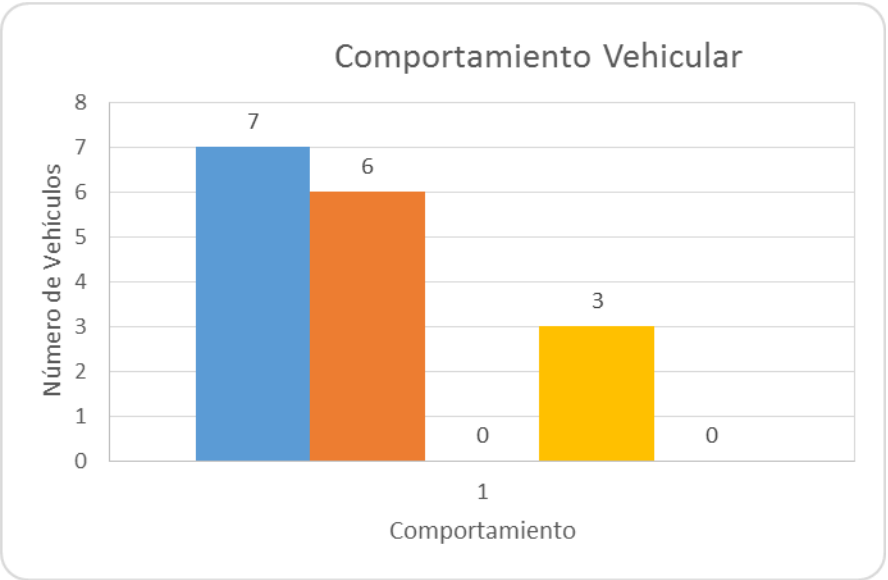


Figura 35 Grafica comportamiento vehicular video simulado 1



## 4.2 Video Real extraído cámara IP

Las Características del video como se muestra en la Imagen 2 Características Video Real se realiza una comparación del video sin procesar y el video procesado con una duración de 33,2804 seg y 278 seg respectivamente, es decir (0,30 seg x Frame).

Ítem	Sin procesar	Procesado
Duración	33,2804 seg	278 seg
# Frames	909	
Ancho	1376	
Alto	768	

Imagen 2 Características Video Real

Para iniciar con la extracción del fondo primero realizamos la corrección de iluminación como se muestra en la Figura 36 original y corrección de iluminación al lado derecho el fotograma corregido.



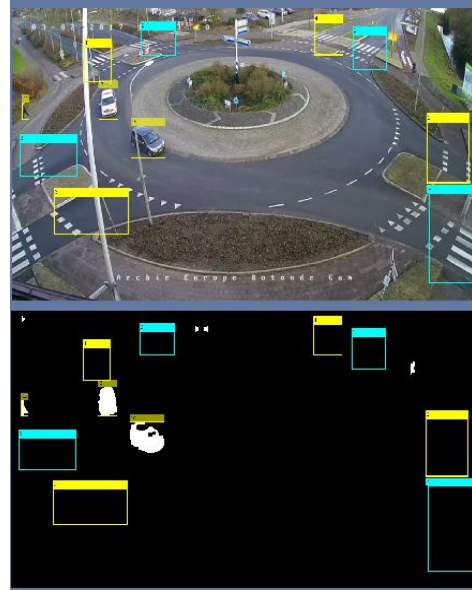
Figura 36 original y corrección de iluminación  
Referencia: Propia

Para la extracción del fondo en el video real se puede observar que en la parte superior de la Figura 38 Extracción de fondo video real corresponde a la original y la inferior al fondo extraído del video, seguido de esto se realiza la segmentación de la imagen dando como resultado la parte inferior de la Figura 36 original y corrección de iluminación En la que se observan píxeles aislados esto debido a la ubicación de la cámara y que por cuestiones de perspectiva algunos transeúntes parecen ser de tamaños muy parecidos a los de un vehículo.

El análisis para las detecciones son las siguientes:

- FP: cantidad de píxeles que generan un cambio tomado como movimiento pero que en realidad no lo es; 6 eventos para un porcentaje del 26%

- b. VP: cantidad de vehículos detectados que generan movimientos; 11.
- c. FN: no existen para la prueba realizada en un video de 33 segundos con 1100 fotogramas aproximadamente.
- d. VN: Este grupo de píxeles no conectados que se alcanzan a ver pero no son detectados como vehículos son 6 que salen al inicio del video de prueba.



*Figura 37 Segmentación video real*

*Referencia: Elaboración propia*



*Figura 38 Extracción de fondo video real*

*Referencia: Elaboración propia*

En la Tabla 4 se identificaron un número 23 vehículos de los cuales, los vehículos con identificación 3 y 17 ingresaron por la vía marcada como entrada

1 mientras que el mayor flujo vehicular se da por la entrada 3, los indicados con entradas y salidas en 0 corresponden a la población de verdaderos negativos expresado anteriormente.

*Tabla 4: Informe generado por el algoritmo video real*

*Referencia: Elaboración propia*

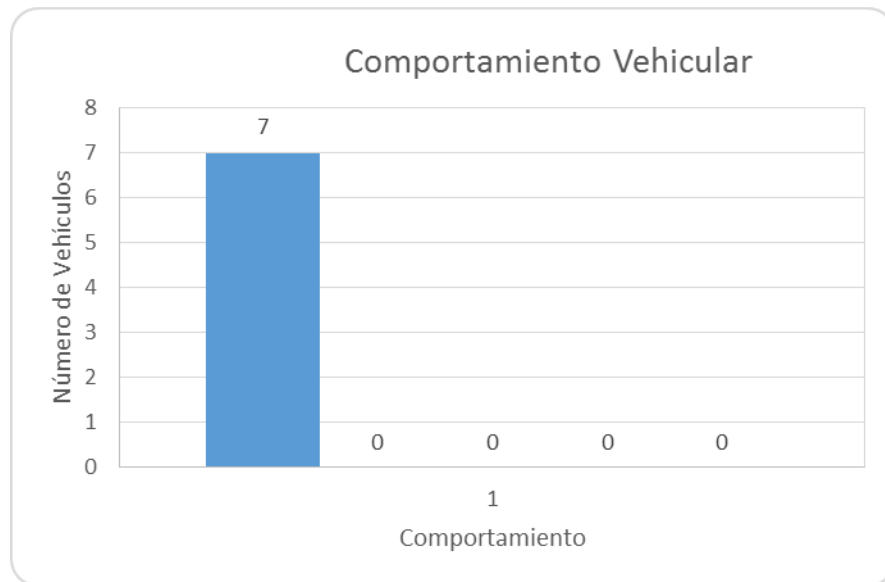
Id Vehículo	Entrada	salida	Número de frame	Tipo de infracción
2	1	3	92	N/A
4	1	3	62	N/A
9	4	4	78	N/A
10	4	4	63	N/A
11	4	4	29	N/A
13	4	4	69	N/A
18	2	0	37	N/A

De la Tabla 5 se deduce que el número de vehículos con un comportamiento no infractores es de 13 en total mientras que, los infractores por contravía al igual a los de obstrucción son 5, no existen infractores por cruces indebidos para un total de 23 vehículos detectados, estos datos son graficados y mostrados en la Figura 39 comportamiento vehicular video real como el comportamiento general de los vehículos en este video.

*Tabla 5 Resultados de comportamiento video real*

*Referencia: Elaboración propia*

No infractores	Infractores por contravía	Infractores por Giro indebido	Infractores por obstrucción	Otros
7	0	0	0	0



*Figura 39 comportamiento vehicular video real*

*Referencia: Elaboración propia*

### 4.3 Video simulado en PTV VISSIM

La siguiente prueba realizada como se muestra en la Imagen 3 Características video PTV-V se realizó en el simulador PTV VISSIM con las siguientes características: Duración del video 30,336 seg y duración del video procesado 132seg, es decir (0,17seg x Frame).

Ítem	Sin procesar	Procesado
Duración	30,336 seg	132 seg
# Frames	756	
Ancho	1016	
Alto	542	

*Imagen 3 Características video PTV-V*

Número de entradas: 4

Número de salidas: 4

Para la extracción del fondo en el video se puede observar que la figura superior de la *Figura 40 Resultado de la segmentación simulación n3*. Corresponde a la original y la inferior al fondo extraído del video, seguido de esto se realiza la segmentación de la imagen dando como resultado la parte inferior de la *Figura 41 Extracción de fondo simulación n3* en la que no se observan píxeles aislados y se reconocen 3 vehículos.

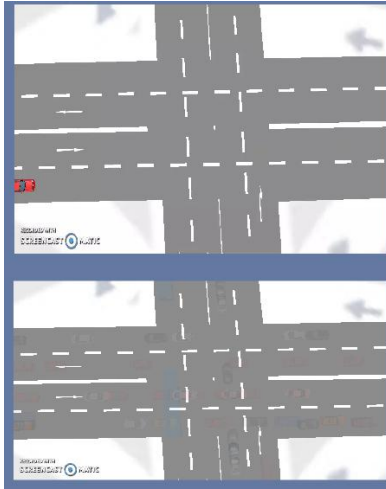


Figura 40 Resultado de la segmentación simulación n3.

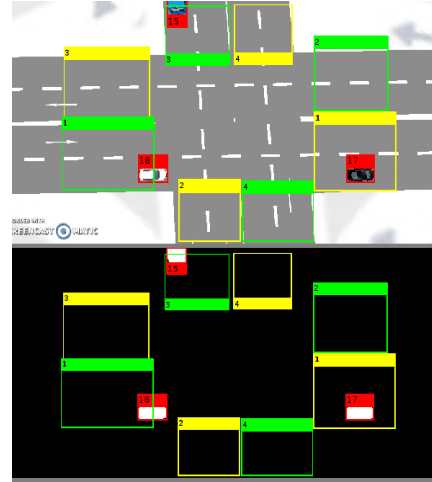


Figura 41 Extracción de fondo simulación n3

Referencia: Elaboración propia

El reporte generado en Excel arroja los resultados de la Tabla 6 en la que indica un numero de 11 vehículos reconocidos de los cuales 9 tienen un comportamiento que no infractores mientras que los infracciones por contravía es uno y uno para infracción por obstrucción respectivamente estos resultado son también expuestos en la Tabla 7 y graficado en la Figura 42 comportamiento vehicular simulación n

Tabla 6: Resultados de la simulación n3

Referencia: Elaboración propia.

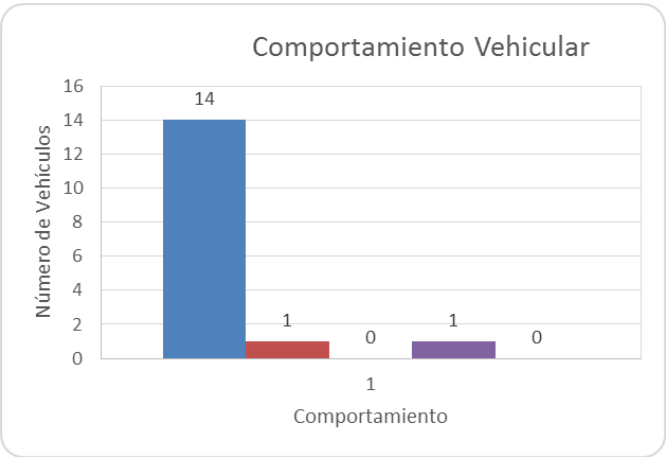
Id_Vehiculo	Entrada	Salida	Número de frame	tipo de infracción
1	1	1	58	N/A
2	2	3	4	N/A
3	4	4	23	N/A
4	2	3	60	N/A
5	1	1	132	Obstrucción
6	2	3	69	N/A
7	4	1	53	N/A
8	1	1	75	N/A

9	1	1	25	N/A
10	1	1	35	N/A
11	2	3	55	N/A
12	4	4	8	N/A
13	1	1	68	N/A
15	3	0	40	Contravía
16	1	0	10	N/A
17	4	1	26	N/A

*Tabla 7: Resultados de comportadito video simulado n3*

*Referencia: Elaboración propia*

No infractores	Infractores por contravía	Infractores por Giro indebido	Infractores por obstrucción	Otros
14	1	0	1	0



*Figura 42 comportamiento vehicular simulación n3*

*Resultados de comportadito video real*

Como se evidencia en los tres videos comparados la gráfica de respuesta para todos es prácticamente la misma por tanto podemos llegar a la conclusión de que el modelo de comportamiento en el tráfico tiende a ser el mismo.

#### 4.4 Análisis de video elaboración propia

Análisis de un video tomado en el norte de Bogotá, como se aprecia en la Figura 43 Imagen tomada desde un quinto piso en la ciudad de Bogotá.ésta imagen tomada desde el quinto piso de un edificio, muestra que no es suficiente la altura para formar una imagen por cada vehiculo. Como se demuestra en la imagen binaria

Figura 44, se muestra un taxi sobreponiendo a otro por tanto, el método de binarización pierde la profundidad de los vehículos y al momento de la segmentación de la imagen se procesa como un solo vehiculo, también por perspectiva los transeúntes parecen ser incluso mucho más grandes que los carros. Diferenciar entre la entrada o salida cruzada por un vehiculo se hace más complicada debido a este mismo ángulo de enfoque, por tanto este video no es considerado para realizar un buen análisis, entre otros muchos tomados en la ciudad, por consiguiente se opta por tomar video de vigilancia de cámaras ip.



Figura 43 Imagen tomada desde un quinto piso en la ciudad de Bogotá.



*Figura 44: Imagen binaria se superposición de un vehículo sobre otro*

## 5 CONCLUSIONES

- El sistema de visión por computador para la identificación de las infracciones de tránsito permite al usuario interactuar con las variables del algoritmo e identificar las infracciones de tránsito ya descritas como lo son; cruce indebido, obstrucción y contravía en videos ya sea real o simulado cumple con los requerimientos.
- Para este sistema de visión de artificial se aplicó una técnica de corrección de iluminación no uniforme para mejorar el proceso de segmentación en este caso mejorando el contraste de la imagen RGB.
- Se ha desarrollado una estrategia para determinar la trayectoria de los vehículo por medio de zonas marcadas en cada fotograma como entradas y salidas teniendo en cuenta que el usuario determina estas marcas y en caso de que un vehículo no atravesase algunas de estas zonas no se podrá determinar dicha trayectoria, ya sea porque no existe zona marcada o el vehículo se encuentra dentro del fotograma al inicio del video, esta estrategia resultó ser eficiente cumpliendo con el objetivo de detectar la trayectoria.



- La evaluación de desempeño del algoritmo depende en cierta medida de las características del video y de la posición de la cámara que en condiciones ideales con videos simulados arroja un informe confiable.
- Para visualizar los resultados del algoritmo se diseñó una interfaz de usuario que le permite a este realizar calibración del umbral y las variables de entrada y salida en el video así mismo observar la reproducción del video y en simultánea a este la segmentación correspondiente en el video.

## 6 Anexos

*Anexo 1 cargar video pre-establecido*

*Anexo 2 Obtener la densidad de color del video*

```
vob = VideoReader('previo.mp4'); se carga el video en el objeto vod
frame = vob.read(inf); obtiene la información relacionada con cada fotograma
nFrames = vob.NumberOfFrames; se obtiene el número de frames de todo el video
```

Se crean matrices de 0 con el mismo tamaño de cada frame una para la densidad y una donde se guardara la diferencia de los del el frame otrignial y el frame de la densidad

```
background_frame2 = double(frame*0);
pixel_sample_density = im2bw(double(frame*0));
diff_frame = double(frame*0);
stream_frame = diff_frame(:,:,1);
bk_downsample = 50;
```

```
for k = 1:bk_downsample:nFrames
    diff_frame = 1-im2bw(uint8(diff_frame),.25);
    pixel_sample_density = pixel_sample_density + diff_frame;
    stream_frame = stream_frame + (1-diff_frame)/(nFrames/bk_downsample);
    nonmoving = double(read(vob, k));
    background_frame2 = background_frame2 + nonmoving;
end
```

```
background_frame2(:,:,1) = background_frame2(:,:,1)./pixel_sample_density;
background_frame2(:,:,2) = background_frame2(:,:,2)./pixel_sample_density;
background_frame2(:,:,3) = background_frame2(:,:,3)./pixel_sample_density;
handles.fondo=uint8(background_frame2);
axes(handles.Ax_Segmentado);
imshow(handles.fondo)
```

*Anexo 3 filtro para cada capa del video para obtener el fondo*

```

handles.CoorX_Entrada=[563.858974358974 720.166666666667 776.611111111111
917.722222222222 783.123931623932 922.064102564103 157.893162393162 294.662393162393
157.893162393162 294.662393162393 414.064102564103 557.346153846154]; % carga x con
los valores anteriores en coorx

handles.CoorY_Entrada=[450.419889502762 519.292817679558 185.408839779005
254.281767955801 118.033149171271 183.911602209945 354.596685082873 426.464088397790
278.237569060773 347.110497237569 519.292817679558 450.419889502762]; % carga x con
los valores anteriores en coory
handles.CoorX_Salida=[783.123931623932 915.551282051282 783.123931623932
911.209401709402 157.893162393162 294.662393162393 160.064102564103 294.662393162393
529.123931623932 674.576923076923 383.670940170940 531.294871794872]; % carga x con
los valores anteriores en coorx
handles.CoorY_Salida=[350.104972375691 424.966850828729 264.762430939226
344.116022099447 127.016574585635 194.392265193370 198.883977900552 278.237569060773
28.1988950276241 116.535911602210 28.1988950276241 115.038674033149]; % carga x con
los valores anteriores en coory

```

#### Anexo 4 Coordenadas del video pre establecido.

```

handles.Rectangulo_Entrada(handles.Num_Rectangulo_Entrada)=rectangle('position',[x(handles.PosCoorX_Entrada-1),y(handles.PosCoorY_Entrada-1),TamaX,TamaY],'EdgeColor','g');

handles.Rectangulo_Salida(handles.Num_Rectangulo_Salida)=rectangle('position',[x(handles.PosCoorX_Salida-1),y(handles.PosCoorY_Salida-1),TamaX,TamaY],'EdgeColor','y');

```

#### Anexo 5 Generador de un rectángulo

```

[handles.ARCH,handles.ruta] = uigetfile({'*.avi'; '*.mp4'; '*.wmv'; '*.*'}...
,'Selecciona un video'); % Selecciona la ruta del archivo .
handles.MatVideo = readFrame(handles.vidObj); % Genera la matriz RGB y la guarda en
Matvideo.
axes(handles.Ax_Video); % posiciona en el axes ax_video.
handles.h = imagesc(handles.MatVideo); % Muestra la imagen y la guarda en h.

```

```

[x(handles.PosCoorX_Entrada),y(handles.PosCoorY_Entrada)]=ginput(1); % obtiene en n
click la posicion x,y.

[x(handles.PosCoorX_Salida),y(handles.PosCoorY_Salida)]=ginput(1);

```

#### Anexo 6 obtener de coordenadas para los rectángulos

#### Anexo 7 Cargar Video

```

movement_old = imabsdiff((frame),im2single(handles.fondo));
umbral=graythresh(movement_old);
mask = im2bw(movement_old,umbral);
mask = imdilate(mask, strel('rectangle', [3,3]));
mask = imopen(mask, strel('rectangle', [3,3]));
mask = imclose(mask, strel('rectangle', [10,10]));
mask=bwareaopen(mask,100);
[~, centroids, bboxes] = handles.obj.blobAnalyser.step(mask);

```

#### Anexo 8 Detección de vehículos

---

```

function predictNewLocationsOfTracks()
for i = 1:length(tracks)
bbox = tracks(i).bbox;
predictedCentroid = predict(tracks(i).kalmanFilter);
predictedCentroid = int32(predictedCentroid) - bbox(3:4) / 2;
tracks(i).bbox = [predictedCentroid, bbox(3:4)];
end

```

```

function predictNewLocationsOfTracks()
for i = 1:length(tracks)
bbox = tracks(i).bbox;
predictedCentroid = predict(tracks(i).kalmanFilter);
predictedCentroid = int32(predictedCentroid) - bbox(3:4) / 2;
tracks(i).bbox = [predictedCentroid, bbox(3:4)];
end

```

e

#### Anexo 9 Asignación de un track

#### Anexo10 Predicción de vehículos

```

function createNewTracks(handles)
    centroids = centroids(unassignedDetections, :);
    bboxes = bboxes(unassignedDetections, :);

    se toman cada uno de los cancroides detectados en la función de detección
    for i = 1:size(centroids, 1)

        centroid = centroids(i,:);
        bbox = bboxes(i, :);

        % Create a Kalman filter object configuración .
        kalmanFilter = configureKalmanFilter('ConstantVelocity', ...
            centroid, [200, 50], [100, 25], 100);

        % Create a new track configuración .
        newTrack = struct(...
            'id', nextId, ...
            'bbox', bbox, ...
            'kalmanFilter', kalmanFilter, ...
            'age', 1, ...
            'totalVisibleCount', 1, ...
            'consecutiveInvisibleCount', 0);
        % Add it to the array of tracks.
        tracks(end + 1) = newTrack;

        asignacion del nuevo track en la tabla

        NewTabla= struct(...
            'id', nextId, ...
            'Entrada', 0, ...
            'Salida', 0, ...
            'Num_Frame', 0, ...
            'Infraccion', 'N/A');
        Tabla(end + 1) = NewTabla;

    Datos_tabla =struct2table(Tabla);
    Datos_tabla=table2cell(Datos_tabla);

```

*Anexo11 funciones generar track, asignar track y eliminar track.*

```

function [assignments, unassignedTracks, unassignedDetections] = ...
    detectionToTrackAssignment()

    nTracks = length(tracks);
    nDetections = size(centroids, 1);

    % Compute the cost of assigning each detection to each track.
    cost = zeros(nTracks, nDetections);
    for i = 1:nTracks
        cost(i, :) = distance(tracks(i).kalmanFilter, centroids);
    end

    % Solve the assignment problem.
    costOfNonAssignment = 20;
    [assignments, unassignedTracks, unassignedDetections] = ...
        assignDetectionsToTracks(cost, costOfNonAssignment);
end

```

*Anexo 12 Asignación de un track*

```

function updateUnassignedTracks()
    for i = 1:length(unassignedTracks)
        ind = unassignedTracks(i);
        tracks(ind).age = tracks(ind).age + 1;
        tracks(ind).consecutiveInvisibleCount = ...
            tracks(ind).consecutiveInvisibleCount + 1;
    end
end

```

*Anexo 13 Reactivador de tracks ya creados*

```

tracks ya creadosfunction updateUnassignedTracks()
    for i = 1:length(unassignedTracks)
        ind = unassignedTracks(i);
        tracks(ind).age = tracks(ind).age + 1;
        tracks(ind).consecutiveInvisibleCount = ...
            tracks(ind).consecutiveInvisibleCount + 1;
    end
end

```

```

function deleteLostTracks()
    if isempty(tracks)
        return;
    end

    invisibleForTooLong = 20;
    ageThreshold = 8;

    % Compute the fraction of the track's age for which it was visible.
    ages = [tracks(:).age];
    totalVisibleCounts = [tracks(:).totalVisibleCount];
    visibility = totalVisibleCounts ./ ages;

    % Find the indices of 'lost' tracks.
    lostInds = (ages < ageThreshold & visibility < 0.6) | ...
        [tracks(:).consecutiveInvisibleCount] >= invisibleForTooLong;

    % Delete lost tracks.
    tracks = tracks(~lostInds);
end

```

*Anexo 14 Eliminar track*

```

function displayTrackingResults()
    if ~isempty(tracks)
        reliableTrackInds = ...
            [tracks(:).totalVisibleCount] > minVisibleCount;
        reliableTracks = tracks(reliableTrackInds);
        if ~isempty(reliableTracks)
            % Get bounding boxes.
            bboxes = cat(1, reliableTracks.bbox);
            % Get ids.
            ids = int32([reliableTracks(:).id]);
            isPredicted(predictedTrackInds) = {' predicted'};
            %
            labels = strcat(labels, isPredicted);
            Id_Vehiculos = cell2mat(labels);
            % Draw the objects on the frame.
            frame = insertObjectAnnotation(frame, 'rectangle',bboxes, labels);
            % Draw the objects on the mask.
            mask = insertObjectAnnotation(mask,'rectangle',bboxes, labels);

            end
        end
    % Display the mask and the frame.
    insertar_Entradas_Salidas(); % se insertan los cadrados de las entradas y
    salidas
    image(mask, 'Parent', handles.Ax_Segmentado);% muestra los frames del video.
    image(frame,'parent',handles.Ax_Video)
    set( handles.Ax_Video,'Visible', 'off');
    set( handles.Ax_Segmentado,'Visible', 'off');
    pause(0.05/handles.vidObj.FrameRate);
    Datos_tabla =struct2table(Tabla);
    Datos_tabla=table2cell(Datos_tabla);
    set(handles.Tbl_Vehiculos,'data',Datos_tabla);
    end

```



## 7 Bibliografía

- [1] Gonzales P, Vision por computador imagenes digitales y aplicaciones, Mexico: Alfaomega, 2002.
- [2] MathWorks, *Image Morphology*, 2009.
- [3] SURA, «SURA,» Cuidado a conducir por las intersecciones viales, [En línea]. Available: <https://www.sura.com/blogs/autos/cuidado-intersecciones-viales.aspx>.
- [4] Nilakorn Seenouvong y Ukrit Watchareeruetai, «A Computer Vision Based Vehicle Detection and Counting System,» [En línea]. Available: <http://ieeexplore.ieee.org/document/7440510/>.
- [5] S. Atev, , H. Arumugam y O. Masoud, «A Vision-Based Approach to Collision Prediction at Traffic Intersections. IEEE International Conference on Information Acquisition,» 2005. [En línea]. Available: <http://ieeexplore.ieee.org/document/1549845/>.
- [6] L. Liwei , X. Junliang y Haizhou , «IEEE International Conference on Information Acquisition,» Multi-view Vehicle Detection and Tracking in Crossroads, [En línea]. Available: <http://ieeexplore.ieee.org/document/6166688/>.
- [7] M. Abril y Mauricio Valencia, «redalyc.org,» Sistema De Visión Artificial Para El Registro De Densidad Peatonal En Tiempo Real, [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=84903515>.
- [8] University, Department of Computer Science and Information Engineering National Taipei, «IEEE International Conference on Information Acquisition,» Crossroad Traffic Surveillance Using Superpixel Tracking and Vehicle Trajectory, 2014. [En línea]. Available: <http://ieeexplore.ieee.org/document/6977103/>.
- [9] Shung-Tsang Tseng, Canción Kai-Tai y Jen-Chao Tai, «Departamento de Ingeniería Eléctrica y de control,» Real-Time Image Tracking For Automatic Traffic Monitoring And Enforcement Applications, [En línea]. Available: [www.elsevier.com/locate/imavis](http://www.elsevier.com/locate/imavis).
- [10] Kamijo, Matsushita, Ikeuchi y Sakauchi, «IEEE International Conference on Information Acquisition,» Traffic Monitoring and Accident Detection at Intersections, 2000. [En línea]. Available: <http://ieeexplore.ieee.org/document/880968/>.
- [11] W. Xiaoling, M. Li-Min y Z. Biaobiao , «A Video-based Traffic Violation Detection System. IEEE International Conference on Information Acquisition,» 2013- 2014, [En línea]. Available:

- 1] <http://ieeexplore.ieee.org/document/6885246/>.
- [1 H. M. Muhammad y H. A. Muhammad , «IEEE International Conference on Information Acquisition,» Vision Based Intelligent Traffic Management System. , [En línea]. Available: <http://ieeexplore.ieee.org/document/6137134/>.
- [1 C. F. Q. Urrego, «Adquisición de variables de tráfico vehicular usando visión artificial. Revista de ingeniería.,» (2009). [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=43021583005>.
- [1 Quiroga, Romero, Garcia y Parra, «Revista Facultad de Ingeniería Universidad de Antioquia.,» Adquisición de variables de tráfico peatonal utilizando vision por computador, (2010- 2011). [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=43021583005>.
- [1 FRAV, «FRAV,» Estudio de intersecciones semaforicas regulas por vision artificial, [En línea]. Available: <http://frav.es/index.php/trabajos-fin-de-grado/107-leidos/resumenes/158-estudio-de-intersecciones-semaforicas-reguladas-por-medio-de-vision-artificial>.
- [1 A. F. GRANADOS y J. I. MARIN , «redalyc.org,» Detección De Flujo Vehicular Basado En Visión Artificial, [En línea]. Available: <http://www.redalyc.org/articulo.oa?id=84903530>.
- [1 Katanyoo , Wittaya y Surasak , «Traffic Violation Detection Using Multiple Trajectories Evaluation of Vehicles. IEEE International Conference on Information Acquisition,» 2013. [En línea]. Available: <http://ieeexplore.ieee.org/document/64>.
- [1 Unad, «Ingeniería de Software,» El modelo DRA (Desarrollo Rápido de Aplicaciones), [En línea]. Available: [http://datateca.unad.edu.co/contenidos/301404/301404\\_ContentadoEnLinea/leccin\\_12\\_\\_el\\_modelo\\_de\\_construccion\\_de\\_prototipos.html](http://datateca.unad.edu.co/contenidos/301404/301404_ContentadoEnLinea/leccin_12__el_modelo_de_construccion_de_prototipos.html).
- [1 J. Portillo , «Instituto politecnico nacional,» Detección de movimiento de objetos mediante secuencias de video., [En línea]. Available: <http://www.sepi.esimez.ipn.mx/electronica/archivos/991.pdf>.
- [2 G. Santillán y I. D. Ibarra, «Visión Artificial y Procesamiento Digital de Imágenes usando Matlab,» García Santillán, Iván Danilo, Visión Artificial y Procesamiento Digital de Imágenes usando Matlab. Ibarra – Ecuador,2008.pag 73}, 2008. [En línea].
- [2 Universidad valladoid, «Vision artificial industrial,» Morfologia, [En línea]. Available: [www.librovision.eii.uva.es/ppt/5\\_Morfologia.pptx](http://www.librovision.eii.uva.es/ppt/5_Morfologia.pptx).
- [2 L. Hincapie, Desarrollo De Una Herramienta Sistémica Basada En procesamiento digital, [En línea]. Available: <https://red.uao.edu.co/bitstream/10614/6472/1/T04528.pdf>.

[2 MathWorks, *Correcting Nonuniform Illumination*.  
3]

[2 MathWorks, *Motion-Based Multiple Object Tracking*.  
4]

[2 The MathWorks, Inc., «The MathWorks, Inc.,» Image Processing Toolbox User's Guide, 1994-  
5] 2005. [En línea]. Available: <http://matlab.izmiran.ru/help/toolbox/images/enhanc17.html>.

[2 «Rotonde cam,» [En línea]. Available: <http://rotondecam.nl/>,. [Último acceso: 01 2017].  
6]

[2 . J. A. Somolinos Sánchez, *Avances en robótica y visión por computador* Escrito por, Ediciones  
7] universidad de castilla, 2002.

[2 José F. Valencia, *Evaluación del impacto del preprocesamiento de imágenes en la*  
8] *segmentación del gris*, 2014. [En línea]. Available:  
<http://www.scielo.org.co/pdf/teclo/v17n33/v17n33a04.pdf>.

[2 Conde, Pérez, González, Silva, Vaquero y Rodríguez, *Estudio De Intersecciones Semafóricas*  
9] *Reguladas Por Visión Artificial*, (2003). [En línea]. Available:  
<http://www.frav.es/pdf/2003/colima2003.pdf>.