DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA MEDICIÓN DE FLUJO VEHICULAR UTILIZANDO VISIÓN POR COMPUTADOR BAJO SOFTWARE LIBRE USANDO TECNOLOGÍA RASPBERRY PI

AUTORES: ISAAC ANÍBAL TRÁVEZ TOCA

DANILO FERNANDO VIRACOCHA SORIA

DIRECTOR: ING. EDDIE GALARZA

LATACUNGA

2016

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Certifico que el proyecto de investigación**, "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA MEDICIÓN DE FLUJO VEHICULAR UTILIZANDO VISIÓN POR COMPUTADOR BAJO SOFTWARE LIBRE USANDO TECNOLOGÍA RASPBERRY PI"** realizado por los señores **ISAAC ANÍBAL TRÁVEZ TOCA, DANILO FERNANDO VIRACOCHA SORIA,** ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar a los señores **ISAAC ANÍBAL TRÁVEZ TOCA, DANILO FERNANDO VIRACOCHA SORIA** para que lo sustente públicamente.

Latacunga, 05 de agosto de 2016

ING. EDDIE GALARZA.
DIRECTOR

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORÍA DE RESPONSABILIDAD

ISAAC ANÍBAL TRÁVEZ TOCA

DANILO FERNANDO VIRACOCHA SORIA

DECLARAMOS QUE:

El trabajo de titulación denominado "**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA MEDICIÓN DE FLUJO VEHICULAR UTILIZANDO VISIÓN POR COMPUTADOR BAJO SOFTWARE LIBRE USANDO TECNOLOGÍA RASPBERRY PI**", ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaramos que este proyecto es de nuestra autoría, en virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 05 de agosto de 2016

Atentamente,

Isaac Aníbal Trávez Toca
C.I. 0503153272

Danilo Fernando Viracocha Soria
C.I. 0503481590

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA DE ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Nosotros, ISAAC ANÍBAL TRÁVEZ TOCA

DANILO FERNANDO VIRACOCHA SORIA.

Autorizamos a la Universidad de las Fuerzas Armadas ESPE la publicación, en la biblioteca virtual de la Institución el presente trabajo "**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA PARA LA MEDICIÓN DE FLUJO VEHICULAR UTILIZANDO VISIÓN POR COMPUTADOR BAJO SOFTWARE LIBRE USANDO TECNOLOGÍA RASPBERRY PI**", cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Latacunga, 05 de agosto de 2016

Atentamente,

Isaac Aníbal Trávez Toca

C.I. 0503153272

Danilo Fernando Viracocha Soria

C.I. 0503481590

DEDICATORIA

Este proyecto dedicamos a Dios, por darnos la vida y guiarnos en cada paso que dimos para culminar nuestra formación profesional, que es el sueño más deseado de todo estudiante.

A nuestros padres por su apoyo, consejos, amor, comprensión, ayuda en los momentos difíciles, y lo más importante, por ayudarnos con todos los recursos para estudiar. Han sido una guía para alcanzar nuestros objetivos.

A todas las personas que creyeron y estaban seguras de nuestras capacidades para llegar a este momento tan especial en cada una de nuestras vidas.

ISAAC, DANILO

AGRADECIMIENTO

Este presente proyecto agradecemos a nuestros padres y familiares porque nos brindaron su apoyo tanto moral y económicamente para seguir estudiando y lograr terminar una meta muy importante en nuestras vidas.

Agradecemos también a nuestro Tutor de tesis el Ing. Eddie Galarza por compartirnos su capacidad y gran conocimiento científico. Además de tener toda la paciencia para el desarrollo del proyecto.

De la misma manera agradecemos a nuestros profesores a quienes les debemos gran parte de nuestros conocimientos, gracias a su paciencia y enseñanza.

Finalmente un eterno agradecimiento a esta prestigiosa universidad la cual nos abrió sus puertas preparándonos para un futuro competitivo y formándonos como personas de bien.

ISAAC, DANILO

ÍNDICE

CAPÍTULO III

CAPÍTULO IV

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente proyecto consiste en la implementación de un sistema para el conteo de vehículos en una determinada calle o avenida de la ciudad el mismo que facilite el conteo de vehículos y por ende tener un estimado de qué cantidad de vehículos transitan por una vía para de esta manera poder ayudar con la movilidad del tránsito y que no exista congestionamiento en las horas pico, por medio de una señal de video utilizando un sistema de visión artificial.
Para lograr este objetivo se desarrolla un algoritmo programado en software libre, con la libreria OPENCV en conjunto con una tarjeta RASPBERRY PI, los mismos que se utilizan para determinar el conteo de los vehículos, adicionalmente se crea una base de datos donde se almacena la información obtenida por el sistema. Los objetivos de este proyecto serán la instalación y configuración de un sistema de video que permitirá adquirir una señal de video para el posterior análisis y procesamiento de los diferentes sitios de la calle o avenida.

PALABRAS CLAVE:

- **FLUJO VEHICULAR - MEDICIÒN**
- **INGENIERÌA DE TRANSITO - ECUADOR**
- **TARJETA RASPBERRY PI**

ABSTRACT

This project involves the implementation of a system for counting vehicles on a given street or avenue of the city to facilitate the counting of vehicles and therefore having an estimate of how many vehicles pass through an avenue or street. The results of the implemented system will to help to mobility and at the same time to decrease the traffic congestion during the peak hours. The system uses the video signal within a machine vision system. To achieve this goal an algorithm that was programmed into free software OPENCV installed on a RASPBERRY PI card. The system is used to determine the number of vehicles in a time period, additionally a database to stared the acquired information was developed. The objectives of this project were the installation and configuration of a video system which will allow us to acquire information about traffic using a video signal that will be processed for further analysis determining the situation of streets or avenues.

KEYWORDS:

- **VEHICULAR FLOW – Measurement**
- **Traffic Engineering - ECUADOR**
- **CARD RASPBERRY PI**

CAPÍTULO I

1. INTRODUCCIÓN

1.1 Antecedentes

El procesamiento de imágenes es un proceso que aspira extraer información visual de una imagen o de algún conjunto de imágenes, para este propósito por medios de algoritmos, se entrena al ordenador para que pueda interpretar la información de las imágenes y distinguir los datos para un propósito concreto.

Esta es una tecnología que cada vez está más ajustada en la sociedad, una de las principales utilidades del procesamiento de imágenes, que cada vez es más frecuente, se la puede encontrar en el sector de la automatización, esta técnica se la utiliza para poder inspeccionar y supervisar las líneas de producción de grandes empresas o fábricas, éste sistema reduce el error humano a la vez que facilita estabilidad y precisión ofreciendo continuos controles de calidad para todas las piezas.

Otro principal uso que tiene el procesamiento de imágenes se encuentra en la video vigilancia, a cualquier otro sensor se lo puede burlar, pero a una cámara inteligente no es muy fácil engañar, ya que ésta detecta cualquier cambio en la imagen que esta recepta.

Con el paso de los años, los sistemas para medir el nivel de tráfico no se han incrementado o desarrollado de una manera automatizada. El conteo de vehículos es una parte fundamental cuando se trata de grandes volúmenes de tráfico tanto para la optimización del tráfico urbano y para el diseño de nuevas

vías que puedan disminuir el congestionamiento que genera la gran afluencia de vehículos.

En las grandes ciudades, este conteo actualmente se realiza en forma manual, pero puede hacerse automáticamente empleando técnicas de visión por computador. En el presente proyecto se desarrolló un algoritmo de detección y seguimiento de vehículos en tiempo real a partir de video, el cual permite el conteo de vehículos en la vía y la estimación del volumen de tráfico.

1.2 Planteamiento del problema

La congestión vehicular es algo normal en una ciudad grande, por lo que siempre se ha buscado una alternativa para poder controlar de una mejor manera la movilidad dentro de las mismas; una de las propuestas es convertir el conteo vehicular en un acto automático y no manual. Manual porque gracias a un número de personas que se ubican en sitios estratégicos a contar vehículos en diferentes momentos del día, se ha logrado llegar a un estimado que determina el intervalo de tiempo que demora encendida una luz de semáforo o su ubicación en una determinada intersección, sin embargo, para una ciudad grande las personas que registran ya no son suficientes, ahora existen más carros, más congestión y más movimiento de lo que un par de ojos puedan registrar.

La congestión vehicular está asociada a las grandes ciudades, ya sea porque la infraestructura no creció o porque las ciudades no se desarrollaron de una manera en la que los vehículos puedan transitar de una mejor forma. En general una de las opciones que siempre se presenta para mejorar la movilidad de una ciudad es optimizar la manera en cómo se usan los recursos de infraestructura existentes. Para ello, es indispensable adquirir y almacenar de manera adecuada la información en tiempo real, la cual permita actualizar el modelo de tráfico que se pretenda optimizar.

El uso de visión artificial al ser utilizado en sistemas de monitoreo tiene muchas ventajas respecto a otras técnicas que se puedan utilizar al momento de realizar dicho ejercicio. Entre las ventajas que posee se incluye la capacidad para poder extraer varias variables de tráfico simultáneamente como también posee la flexibilidad de escoger la escena de tráfico que se desee monitorear, además las diversas técnicas de visión artificial que existen se puede implementar en tiempo real y unido a los avances en comunicaciones, es posible transmitir la información obtenida a las autoridades de tránsito de una manera remota para su posterior análisis; así mismo la instalación que requiere el equipo para poder ser instalado en la vía son mínimas.

1.3 Justificación

El conteo de vehículos actualmente se lo realiza manualmente, por lo que se ha comprobado que esta técnica no es de todo confiable, ya que, una persona bien dedicada y además bien intencionada no brinda un apoyo suficiente y eficaz cuando se trata de un sistema de monitoreo, la concentración de una persona para estos casos se degenera muy por debajo de los niveles aceptables en tan solo 20 minutos.

El presente apartado muestra el desarrollo de un algoritmo de detección y seguimiento de vehículos en tiempo real a partir de video, el cual permite el conteo de vehículos en la vía y la estimación del volumen de tráfico.

El sistema a emplearse en el proyecto está orientado a facilitar el conteo de vehículos y por ende tener un estimado de que cantidad de vehículos transitan por una vía específica para de esta manera poder ayudar con la movilidad del tránsito y que no exista congestionamiento en las horas pico.

1.4 Objetivos del proyecto

1.4.1 General

Diseñar e implementar un sistema para la medición de flujo vehicular utilizando visión por computador bajo software libre usando tecnología RASPBERRY PI.

1.4.2 Específicos

- Adquirir conocimientos sobre el software libre OPENCV y funcionamiento de las distintas librerías, para la utilización correcta en la aplicación
- Investigar sobre el funcionamiento de la tarjeta RASPBERRY PI para almacenar los datos requeridos del software libre OPENCV.
- Crear un algoritmo compacto y eficiente en el software libre OPENCV, de tal manera que no consuma gran cantidad de recursos en la tarjeta RASPBERRY PI.
- Establecer la adecuada ubicación de la cámara para monitorear el flujo vehicular.
- Implementar un SERVIDOR WEB en una tarjeta RASPBERRY PI, para gestionar de manera remota los datos adquiridos del sistema para la medición de flujo vehicular.
- Relacionar las imágenes obtenidas en tiempo real por la cámara en OPENCV, para ser procesadas en el mismo mediante la implementación de un algoritmo de control.
- Realizar pruebas de funcionamiento para verificar el correcto funcionamiento del sistema y poder corregir cualquier error que se presente en el mismo.
- Establecer un análisis estadístico del flujo vehicular con la información obtenida del algoritmo de control implementado en el software libre OPENCV.

1.5 Características de OpenCV

Es una librería bajo una licencia BSD (Berkeley Software Distribution) y por lo tanto es gratis, tanto para uso académico y comercial. Fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real, puede tomar ventaja de procesamiento multi-core.

Es una biblioteca de código abierto de visión por computador, la biblioteca está escrita en C y C++, se ejecuta en Linux, Windows y Mac OS X. Se ha desarrollado en las interfaces de Python, Ruby, Matlab, entre otros lenguajes.

OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real, optimizado y puede tomar ventajas de los procesadores multi-core. Utiliza automáticamente la biblioteca IPP propiedades en tiempo de ejecución si se ha instalado la biblioteca.

Características principales

- Interfaz principal de OpenCV es en C ++
- También hay interfaces de plenos C, Python y Java
- Optimizado para tiempo real de procesamiento de imágenes y de computación en aplicaciones de visión
- Optimizado para procesadores de Intel
- Las aplicaciones de OpenCV se ejecutan en Windows, Android, Linux, Mac e iOS.

1.5.1 Usuarios de OpenCV

La mayoría de los informáticos y programadores prácticos son conscientes de alguna faceta del papel que juega la visión por computador. Pero pocas personas son conscientes de todas las formas en las que se utiliza la visión por computador. Por ejemplo, la mayoría de la gente es un poco consciente de su uso en la vigilancia y muchos también saben que cada vez se está utilizando para imágenes y video en la web. Unos pocos han visto algún uso de la visión por computador en las interfaces de juego. Sin embargo, pocas personas se dan cuenta que la mayoría de las imágenes aéreas y de la calle hacen pesada el uso de la calibración de la cámara y las técnicas de reconstrucción de imágenes.

1.5.2 Visión por computador

Visión por computador es la transformación de los datos de una cámara de fotos o video en una nueva representación. Todas estas transformaciones se realizan para lograr algún objetivo en particular. Los datos de entrada pueden incluir alguna información contextual.

Una nueva representación podría significar convertir una imagen de color en escala de grises o eliminación de movimientos de la cámara de una secuencia de imágenes. (G.R. Bradski, 2008)

Debido a que somos criaturas visuales, es fácil ser engañado en el pensamiento de que las tareas de visión por computador son fáciles. Puede ser muy difícil mirar fijamente una imagen y encontrar un coche, porque las intuiciones iniciales son equivocadas. El cerebro humano divide la señal de visión en muchos canales que transmiten diferentes tipos de información en su cerebro. Su cerebro tiene un sistema de atención que identifica en una tarea dependiente.

En un sistema de visión artificial, un ordenador recibe una cuadricula de números de la cámara o desde el disco. En su mayor parte, no hay una función de reconocimientos de patrones, sin control automático de enfoque y apertura, no hay cruzadas asociaciones con años de experiencia. En su mayor parte los sistemas de visión son todavía bastante ingenuos. La Figura 1 muestra la foto de un automóvil, en esta foto se presenta un espejo lateral en el lado del conductor del auto. Lo que el equipo “ve” es solo una cuadricula de números. Cualquier número determinado dentro de esa rejilla tiene un gran componente de ruido y en si proporcionan poca información, pero esta cuadricula de números es todo el equipo. El trabajo consiste en convertir esta ruidosa cuadricula de números en la percepción “retrovisor”.

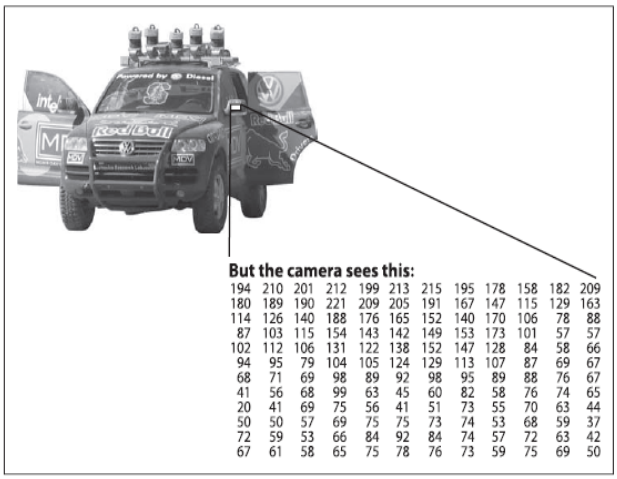

Figura 1. ***Representación numérica del espejo lateral de un auto, para un computador.***

Fuente: (G.R. Bradski, 2008)

El problema que perdura y formalmente es imposible de resolver es, dada una vista de dos dimensiones (2D), de un mundo (3D), no hay una única manera de reconstruir la señal 3D. Formalmente, un problema mal planteado no tiene única o definitiva solución. La misma imagen 2D puede representar a cualquiera de una combinación infinita de escenas en 3D, incluso si los datos eran perfectos, sin embargo como ya se ha mencionado, los datos están dañados por el ruido y distorsiones. Esta corrupción se deriva de variaciones en el mundo (clima, iluminación, reflexiones, movimientos), las imperfecciones del lente y configuración mecánica, tiempo de integración finito en el sensor (desenfoque de movimiento), el ruido eléctrico en el sensor u otros aparatos eléctricos y artefactos de compresión. Teniendo en cuenta estos enormes desafíos en el diseño de un sistema práctico, el conocimiento contextual adicional se puede utilizar para trabajar en torno a las limitaciones que imponen los sensores visuales. En la Figura 2 se puede tener un panorama más claro porque la visión por computador es tan difícil, la aparición de objetos 2D puede cambiar radicalmente con un punto de vista.

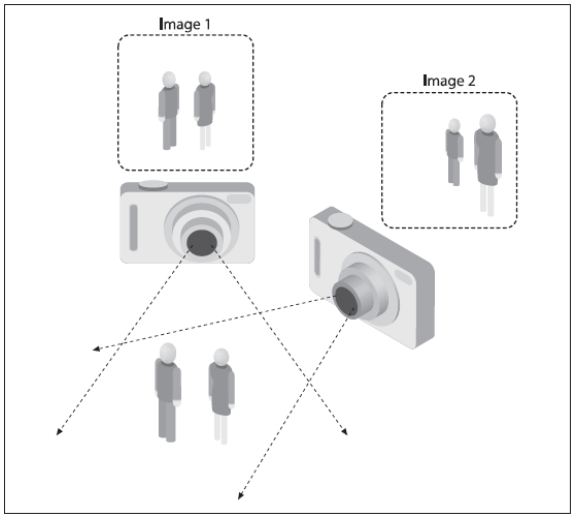

Figura 2. ***Posicionamiento de cámaras***

Fuente: (G.R. Bradski, 2008)

Las personas también tienden a centrar los objetos a la hora de tomar imágenes y tienden a poner en diferentes orientaciones. Para nosotros existe a menudo un poco de información implícita no intencional dentro de las fotos tomadas por el ser humano. La información contextual también se puede modelar de forma explícita con las técnicas de aprendizaje automático. Variables ocultas tales como el tamaño, la orientación a la gravedad, y así sucesivamente puede ser correlacionada con sus valores en un conjunto de entrenamiento etiquetados. Alternativamente, se puede intentar medir las variables de polarización ocultas mediante el uso de sensores adicionales. El uso de la gama de láser para medir la profundidad permite medir con precisión el tamaño de un objeto.

El próximo problema que enfrenta la visión por computador es el ruido. Normalmente se analiza el ruido mediante el uso de métodos de estadística. Por ejemplo, puede ser imposible detectar un borde de una imagen simplemente mediante la comparación de un punto para sus vecinos inmediatos. Pero las estadísticas a través de una región local, la detección de bordes se vuelve mucho más fácil. Una ventaja real debe aparecer como una serie de tales respuestas de vecinos inmediatos a través de una región local, cada una de su orientación es consistente con sus vecinos. También es posible para compensar el ruido mediante la adopción de estadísticas a través del tiempo. Todavía otras técnicas representan el ruido o distorsiones mediante la

construcción explicita de modelos que aparecieron directamente de los datos disponibles. Por ejemplo, debido a distorsiones de lente que se conocen bien, solo hay que aprender de los parámetros para un modelo simple polinomial, con el fin de describir y de este modo corregir casi en su totalidad, tales distorsiones.

1.5.3 Aplicaciones similares.

- OpenCV ha sido usado en el sistema de visión del vehículo no tripulado Stanley de la Universidad de Stanford, el ganador en el año 2005 en el gran desafío DARPA.
- OpenCV se usa en sistemas de vigilancia de video.
- OpenCV es la clave en el programa Swistrack, una herramienta de seguimiento distribuida.

1.5.4 Estructura y contenido de OpenCV

OpenCV comprende de cinco componentes principales, cuatro se muestran en la Figura 3. El componente de CV contiene el procesamiento de imágenes básico y algoritmos de visión por computador de alto nivel; ML es la biblioteca de aprendizaje de máquina que incluye muchos clasificadores estadísticos y herramientas de agrupación. HighGUI contiene rutinas y funciones de E / S para el almacenamiento y carga de vídeo e imágenes, y CXCore contiene las estructuras de datos básicos y contenidos.

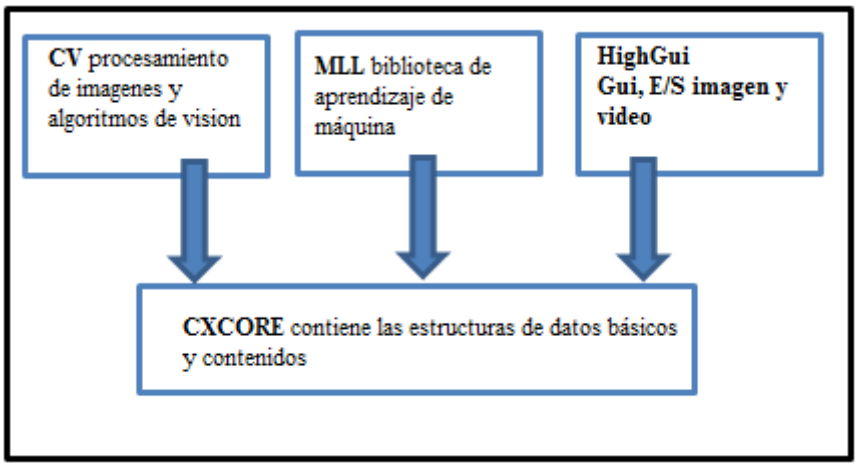

Figura 3*. Estructura y contenido de OpenCV*

- Eigen objects, una técnica de reconocimiento computacionalmente eficiente, que es, en esencia, un procedimiento de comparación de plantillas.
- Modelos 1D y 2D ocultos de Markov, una técnica de reconocimiento estadístico resueltos por programación dinámica.
- Reconocimiento de gestos de apoyo a la visión estereoscópica.
- Extensiones para la triangulación de Delaunay, secuencias, etc.
- Visión estéreo
- Descriptores de textura
- Seguimiento de ojos y boca
- seguimiento 3D
- Deformaciones intermedias entre dos vistas de cámara
- Segmentación de fondo plano
- Video vigilancia

Algunas de estas características pueden migrar a CV en el futuro; otros probablemente nunca lo harán.

1.6 Segmentación de imágenes

El objetivo de esta etapa es agrupar los pixeles por medio de algún método de homogeneidad y de esta manera poder tener una imagen particionada en regiones significativas. Luego de esta etapa la imagen esta lista

para poder ser interpretada o reconocida de acuerdo a la aplicación que se requiera dar a la misma.

Para poder implementar la segmentación existen dos técnicas muy utilizadas, la primera es la técnica basada en los bordes y la segunda búsqueda de regiones homogéneas.

1.6.1 Técnica basada en los bordes

Esta técnica básicamente consiste en agrupar los pixeles determinados como bordes en la etapa de procesamiento. Para que un píxel borde pueda de ser definido como píxel de frontera, necesariamente los otros píxeles deben tener la misma dirección y módulo del gradiente. Para poder tener una conectividad se puede utilizar la transformada de Hough.

Esta transformada examina y detecta las formas geométricas sencillas en una imagen, como pueden ser la detección de círculos o líneas rectas e incluso se puede tener una transformada generalizada para poder obtener de estas una curva cuando no exista una expresión analítica para poder describir dicha curva. Como se muestra en la figura 4. (Méndez, 2015)

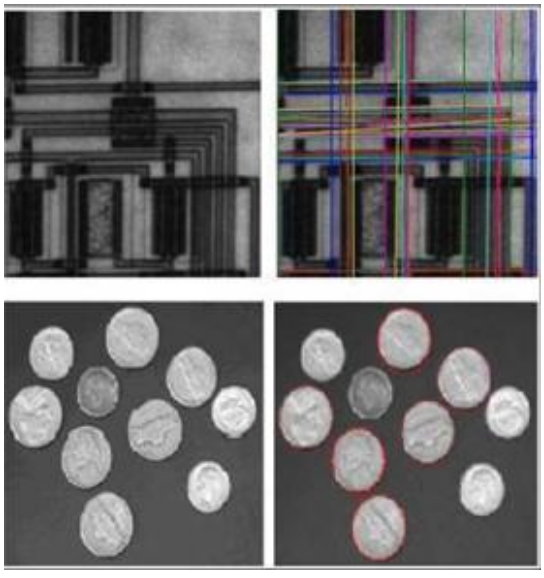

Figura 4. ***Transformada de Hough para detección de líneas y círculos.***

Fuente: (Méndez, 2015)

Numeración: Es recomendable utilizar esta técnica cuando existe una resaltada diferencia entre los objetos a ser analizados y el fondo de la imagen. Para que se pueda utilizar esta técnica se debe emplear la similitud entre los píxeles pertenecientes a un objeto y sus diferencias respecto a otros.

Cuando se realiza una umbralización se tiene como resultado una imagen binarizada, en la cual cada píxel solo puede tener un valor ya sea este “0” ó “1” este valor va a depender de la tonalidad de gris que posea la imagen. Para utilizar esto se utiliza dos límites de umbral un superior y un inferior, si el pixel no posee el nivel de gris dentro de estos límites, entonces el píxel toma el valor de “0” caso contrario será “1”. (Méndez, 2015)

Para lograr la umbralización, la segmentación por histograma se convierte en una buena opción. Un histograma se compone generalmente de un valle y dos picos, en este caso el umbral necesariamente debe ser colocado en el valle. Ver figura 5.

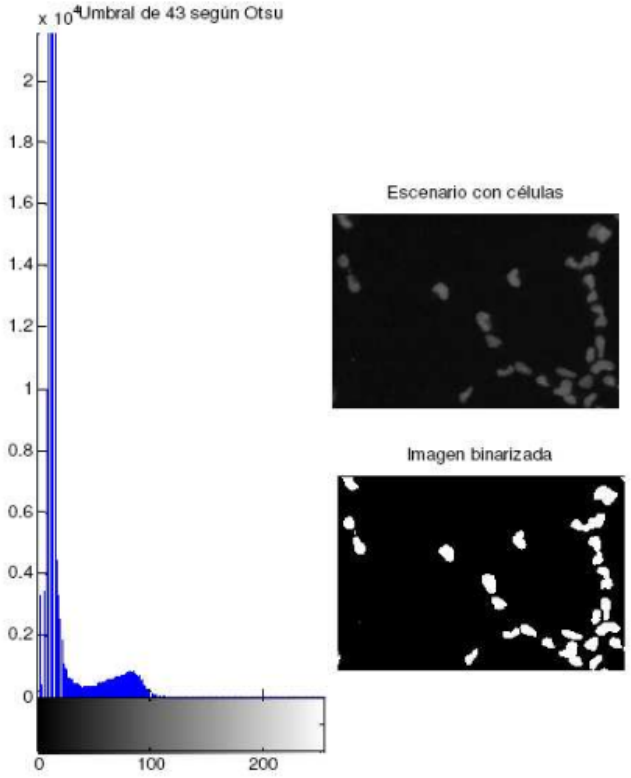

Figura 5. ***Umbralización utilizando histograma***

Fuente: (Méndez, 2015)

1.7 Selección de región de interés

El objetivo de una región de interés, es que la región a observarse pueda ser configurable por el usuario, con el fin de no procesar regiones que no formen parte de la vía o que para el estudio sean totalmente inútiles y no brinden una información veraz para poder considerarles como parte de la imagen.

Existen también otro tipo de perspectivas que demandan de correcciones para poder ser procesadas, para poder hacer este tipo de correcciones se deben usar funciones especiales de los diferentes programas para el tratamiento de imágenes en este caso OpenCV. Gracias a estas técnicas se obtiene una matriz la cual se la utiliza en una transformación de perspectiva, la creación de los puntos para deducir dicha matriz se hace con una interfaz con el usuario, en la cual se pide indicar con el mouse del computador la región de interés y luego dos puntos más para poderlos utilizar como destino. Se muestra en la figura 6. (Baque J. , 2015)

Figura 6*. Ejemplo selección de región de interés.*

Fuente: (Baque J. , 2015)

1.7.1 Estimación previa del fondo

La estimación previa de fondo es un método comúnmente utilizado para detectar y segmentar regiones de interés en movimiento, independientemente del tipo de imágenes que se obtenga de la secuencia de video. Este método provee la información más completa del objeto a estudiar, pero al mismo

tiempo es considerablemente sensible a los cambios dinámicos propios que posee la escena, como son movimientos de la cámara y los cambios de iluminación.

Cuando se trabaja con un fondo estático se puede modelar como una constante o también se puede emplear el filtro de media; este método funciona para casos simples en el cual no se tenga un movimiento continuo del objeto, ya que para casos donde el fondo se encuentra en movimiento como en el caso de que el viento mueva a los objetos, se aplican métodos más sofisticados tales como modelos estadísticos para cada píxel, la principal desventaja de trabajar con un método de estas características es el elevado costo computacional, ya que impide su aplicación en sistemas que funcionen en tiempo real.

1.7.2 Máscaras de borde de fondo

La manera más usual de detectar contornos es de forma, diferencial o derivada. La derivada permite calcular variaciones entre un punto y su vecindario (máscara). Un contorno implica una discontinuidad en dicha función de la imagen, es decir donde la función tiene un valor de gradiente o derivada alta. (Cofre, 2015)

La mayoría de las técnicas para detectar bordes emplean operadores basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen como se muestra en la Figura 7(c).

Figura 7. ***Ejemplo de mascara de fondo (B) de una imagen (A) con el fondo calculado por medida aumentada (B)***

Fuente: (Cofre, 2015)

1.7.2.1 Operadores de Sobel.

La desventaja de los operadores de gradiente es que aumentan el ruido en la imagen, tanto los operadores de Sobel como el resto de operadores de vecindad tienen la propiedad de suavizar la imagen, eliminando parte del ruido y minimiza la aparición de falsos contornos como se puede apreciar en la Figura 8. (Garcia, 2008)

Figura 8. ***A) Imagen original, B) imagen aplicando el operador Sobel.***

Fuente: (Santillan, 2015)

1.7.2.2 Operador de Roberts.

Este operador es muy similar al de Sobel, lo que cambia son los coeficientes de las máscaras. El operador de Roberts es muy simple, ya que trabaja muy bien con imágenes binarias, además este operador obtiene una buena respuesta ante bordes diagonales, también ofrece buenas prestaciones en cuanto a localización. El inconveniente con este operador es su extremada sensibilidad al ruido. (Garcia, 2008)

1.8 Representación de objetos

En este aparato se indaga las diferentes representaciones de objetos utilizadas en el procesamiento digital de imágenes y de videos, se estudian modelos de objetos genéricos y específicos. Con respecto a los objetos genéricos, a estos se los estudia la representación basada en formas geométricas y puntos. Respecto a los objetos específicos se analizan la representación basada esqueletos y contornos. (Cofre, 2015)

1.8.1 Representación de objetos genéricos

El reconocimiento de un objeto es un paso importante cuando se quiere estudiar una imagen o secuencias de las mismas. El proceso se entiende en este contexto como el aprendizaje de objetos nuevos para que posteriormente puedan ser reconocidos cada vez que se le presentan al sistema. El reconocimiento se suele basar en la creación de los modelos estudiados, que en lugar de preservar toda la información pertinente a dichos objetos, de esta forma tratan de minimizar dicha información almacenando la menor cantidad de ella posible para optimizar el proceso de reconocimiento. La selección de los parámetros que definen un modelo es uno de los pasos más críticos del proceso de reconocimiento, los modelos pueden ser introducidos al sistema mediante aprendizaje supervisado, pero también existe la posibilidad de permitirle al sistema adquirir modelos nuevos a partir de ejemplos establecidos mediante aprendizaje no supervisado. (Otero, 2004)

Estas representaciones se las puede clasificar como:

1.8.1.1 *Representación basada en puntos:*

En la Figura 9 se observa que el objeto a estudiar es representado como un simple punto, esta representación es ideal para seguir objetos que ocupan pequeñas regiones en una imagen como se los utiliza en aplicaciones con radar, que utilizan el punto como una representación para el rastreo de múltiples objetivos.

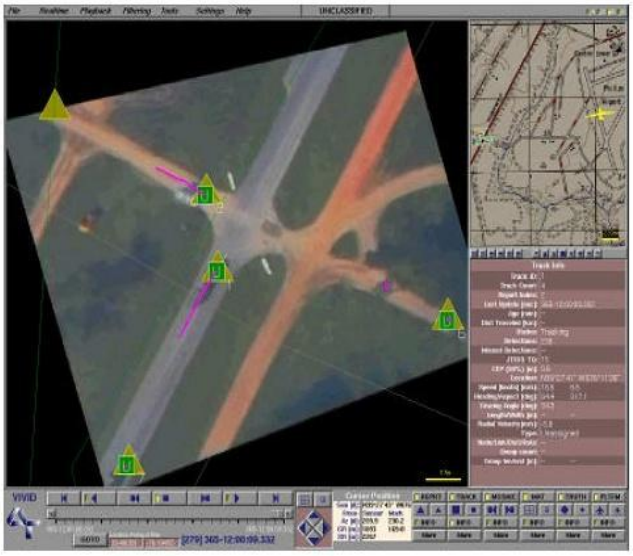

Figura 9*. Ejemplo representación basada en puntos*

Fuente: (Cofre, 2015)

1.8.1.2 Forma geométrica en 2D:

Esta técnica consiste en encerrar el objeto por medio de una forma geométrica al objeto, ya sean estas circunferencias o cuadrados como se aprecia en la Figura 10. Existen una gran variedad de representaciones para poder encerrar al objeto ayudando así a tener referencia en la imagen que se la está estudiando. (Mendieta, 2015)

Figura 10*. Representación de objetos en forma geométrica.*

Fuente: (Mendieta, 2015)

1.8.1.3 Forma geométrica en 3D:

La principal ventaja de esta representación con respecto a la anterior es que posee un mayor grado de precisión, pero se debe tener en cuenta que su principal desventaja es que involucra un mayor procesamiento respecto a las formas geométricas en 2D. Esta representación consiste básicamente en cerrar al objeto en una imagen tridimensional. Para las diferentes imágenes existen distintos tipos de representaciones de formas geométricas en 3D, por ejemplo, en el seguimiento para utilizarlos con personas, a estos se los representa con forma de cilindros, para los vehículos se tiene una representación con paralepípedos como se muestra en la Figura 11. (Cofre, 2015)

Figura 11. Representación de un vehículo en 3 dimensiones

Fuente: (Cofre, 2015)

1.8.2 Representación de objetos específicos

La forma de un objeto conocido puede ser generada por una descripción aproximada por medio de la representación de objetos específicos. La principal ventaja de utilizar esta representación es la fidelidad en la descripción del objeto, teniendo en cuenta que su primordial desventaja es la incapacidad de describir otros tipos de objetos en la imagen, otra desventaja que posee es que cuando se necesita encontrar los bordes de un objeto, se requiere un alto tiempo de procesamiento para poder realizar este tipo de análisis. (Bellesteros, 2015)

Este tipo de representaciones se las clasifica como:

1.8.2.1 Representación por esqueletos:

Este tipo de representación reduce los objetos de una imagen, a un esqueleto del grosor de un píxel, es decir cuando se obtienen un objeto cualquiera se va reduciendo de grosor hasta dejarlo del grosor de un píxel. El esqueleto que se consigue luego de realizar dicho procedimiento da una idea de la forma inicial del objeto como muestra la Figura 12.

El inconveniente cuando se realiza este tipo de transformación, es el tiempo de cómputo necesario para poder llegar hasta dicha transformación.

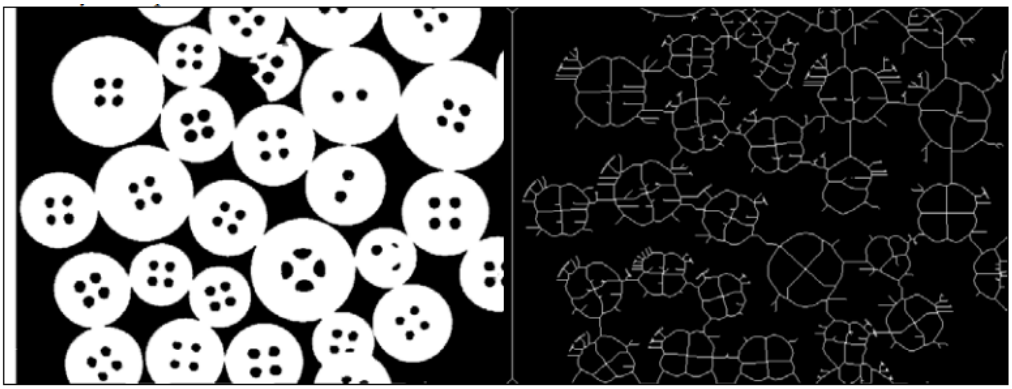

Figura 12*. **Resultado de la esqueletización***

Fuente: (Gonzales, 2015)

1.8.2.2 Representación basada en contorno:

Define el límite o borde de un objeto, la velocidad depende del detalle en el contorno y del método de cálculo para desarrollarlo. Tiene la ventaja de poseer una alta precisión y calidad. La representación puede ser por medio de puntos parametrizados, por puntos representativos o por los píxeles de borde de la imagen. (Brown, 2000)

1.9 Detección de vehículos

La finalidad de los sistemas de detección de vehículos, los cuales centran su atención en secuencias de video de escenas de tráfico, es detectar

los vehículos que transitan por una determinada avenida. Las imágenes obtenidas por las cámaras previamente instaladas son procesadas y se establece si se hallan vehículos en la imagen y determinan también su posición.

Los inconvenientes más comunes a los que se enfrentan dichos sistemas son (Bellesteros, 2015):

- Movimiento de la cámara: los elementos estáticos como son el fondo o la carretera misma pueden aparecer como un movimiento en la secuencia del video, el reto es poder eliminar dicho movimiento creado por la cámara.
- Cambio de patrones de apariencia: la apariencia tanto del vehículo o del fondo pueden verse muy afectados por las condiciones climáticas que se pueden presentar y también por los cambios de iluminación que se puedan dar en diferentes entornos como se muestra en la Figura 13.
- Pérdida de información debido al cambio de secuencias en 3D a imágenes 2D: cuando se adquiere una imagen para su posterior estudio siempre implica una trasformación de 3D al espacio en 2D. Al hacer esta transformación de la imagen esta se distorsiona, por lo que la forma de la imagen no tiene una forma homogénea, esto hace que vehículos del mismo tamaño se puedan apreciar son diferentes dimensiones en la imagen según su posición relativa a la cámara.
- Variación intrínseca del vehículo: muchos factores como: color, forma, tamaño, entre otros, presentan una alta variabilidad al momento de obtener una imagen de un vehículo aunque la mayoría suelen tener una misma estructura.
- Procesamiento en tiempo real: los sistemas de detección de vehículos requieren que la detección del vehículo se efectúe en tiempo real. Para poder lograr que se cumpla con esto se debe tener muy en cuenta el tiempo de cómputo de los algoritmos que se estén utilizando.

La detección de vehículos en si es muy costosa y un poco dificultosa, para poder facilitar la detección esta se la divide en dos etapas: generación de hipótesis (GH) y verificación de hipótesis (VH) [12,5]. La primera etapa consiste en una búsqueda rápida en la imagen de regiones que potencialmente pueda

existir un vehículo. La segunda etapa las hipótesis obtenidas en la anterior son estudiadas, este estudio se lo realiza mediante el uso de características relacionadas con la apariencia de los vehículos.

Figura 13. ***En (a) y (b) se tiene efectos producidos por cambios de iluminación y condiciones meteorológicas. También se tiene variaciones en la apariencia del vehículo (color forma) (c) y (d).***

Fuente: (Bellesteros, 2015)

Generación de hipótesis (GH): Esta etapa realiza un procesado rápido de la secuencia a estudiar obteniendo regiones con una alta probabilidad de que pueda existir vehículos.

Se encontraron los siguientes métodos para la generación de esta hipótesis:

- Métodos basados en el conocimiento: principalmente se basan en la aplicación del conocimiento previo de las hipótesis, por lo que se busca el seguimiento de varias características fijadas con anterioridad y que se encuentren presentes en los vehículos y sus entornos.
- Métodos basados en visión artificial estereoscópica: básicamente consiste en los diferentes procedimientos para la determinación de la forma de los objetos en la escena. La forma se la determina mediante la distancia desde el objeto respecto a un sistema de referencia, estos métodos típicamente se los realizan con el uso de sistemas que comprenden de dos cámaras separadas, el objetivo de estas es adquirir una imagen en tres dimensiones.
- Métodos basados en el análisis del movimiento (Woel, 2005): los métodos que se discutieron anteriormente abordan el problema de la separación entre el fondo y el objeto. Los métodos basados en el movimiento de los objetos se basan en la distinción de vehículos y no

vehículos mediante el movimiento que se genera en los movimientos de las imágenes extraídas de las secuencias de video.

Verificación de hipótesis esta etapa tiene como objetivo la verificación de las regiones obtenidas en la anterior. Para ello se encontró dos categorías importantes (Bellesteros, 2015):

- Métodos basados en el aprendizaje (Gupte, 2002): estos métodos se fundamentan en la clasificación de la hipótesis de los vehículos mediante el aprendizaje que obtiene luego de que este pasará por un entrenamiento previo.
- Métodos basados en modelos (Baker, 2002): este método se basa fundamentalmente en el hecho de que los vehículos son prácticamente figuras geométricas o poliedros, por lo que estas técnicas recurren a compararlas con varias características básicas (bordes, simetrías,…) para la descripción de los distintos modelos.

1.10 PYTHON

Python es un lenguaje de programación que permite varios estilos de programación en lugar de forzar a los programadores a adoptar un estilo específica de programación, por lo que Python en un lenguaje de programación poderoso y cómodo de aprender, cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. (Rossum, 2009)

Python fue introducido como un sucesor del lenguaje de programación ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba. Python fue creado a finales de los Ochenta por Guido van Rossum en el Centro de las Matemáticas y la Informática en los Países Bajos.

Python usa tipiado dinámico y conteo de referencias para la administración de memoria, lo que quiere decir, que una misma variable puede tomar valores de distinto tipo en diferentes momentos.

Una característica importante de Python es la resolución dinámica de nombres; es decir, lo que enlaza un método y un nombre de variable durante la ejecución del programa.

Python ofrece varios toolkits alternativos para la implementación de una aplicación gráfica. Entre las más conocidas están: Tkinter, PyGTK, PyQT, Pythonwin y wxPython.

Tkinter es un toolkit para el desarrollo de las interfaces gráficas que viene integrado a la distribución de Python. Tkinter en una adaptación en Python de la API gráfica Tk ofrecida por el lenguaje de programación Tcl. Esta se distingue por ser multiplataforma y además viene integrada con la distribución estándar de Python, lo que quiere decir que no necesita de instalaciones adicionales.

PyGTK es otro de los toolkits que definen finas capas de Python, su principal inconveniente es que están orientados a una plataforma definida (UNIX o Windows), aunque las interfaces que muestran son muy rápidas dado que interactúan directamente con APIs nativas de la plataforma. El uso de esta es más complejo que Tkinter por lo que no llama la atención a la mayoría de programadores.

El toolkit wxPython es multiplataforma, presenta una apariencia nativa y además ofrece un alto rendimiento gráfico, está basado en la API en C++. wxWiggest ha venido simbolizando desde hace unos años una alternativa muy capaz y sencilla tanto a los toolkits gráficos de Linux como son GTK y QT y como también a las APIs de Windows. El combo wxWiggest/wxPython permite

realizar interfaces gráficas muy sofisticadas de una manera tan sencilla como Tkinter.

Muchos usuarios afirman que wxPython es el mejor toolkit para el desarrollo de interfaces gráficas con Python, el cual combina eficacia, con sencillez de uso y permite escribir código fuente una sola vez independientemente de la plataforma que esté utilizando, la aplicación adopta una apariencia y formas más apropiadas a la plataforma que se esté utilizando.

1.11 Raspberry Pi

La Fundación Raspberry Pi es una organización que dió sus primeros pasos como fundación en 2008, pero que en realidad llevaba gestionándose desde mucho tiempo atrás. En 2011 desarrolló la Raspberry Pi como ordenador de bajo costo para facilitar la enseñanza de la informática en los colegios, pero hasta 2012 no comenzó a fabricarse. La fundación recibe apoyos del laboratorio de informática de la universidad de Cambridge y de Broadcom. [15]
La evolución de los diferentes modelos se ha complementado hasta ahora tener una tarjeta modelo B con procesador de 1GB.

Es un pequeño computador del tamaño de una tarjeta de crédito, el cual se puede conectar fácilmente a una televisión vía HDMI o RCA. Además, se puede utilizar, lógicamente, con un teclado y un mouse. Tiene conexión a internet y unos pines GPIO (General Purpose Input/Output), para que se pueda interactuar una placa con sensores, botones, o lo que se requiera. Por otra parte, reproduce video en alta definición y también sirve como procesador de textos o para jugar, pero realmente está pensada para programarla. (Pinedo, 2015)

Existen dos modelos: el modelo A y el modelo B. El más reciente es el modelo B.

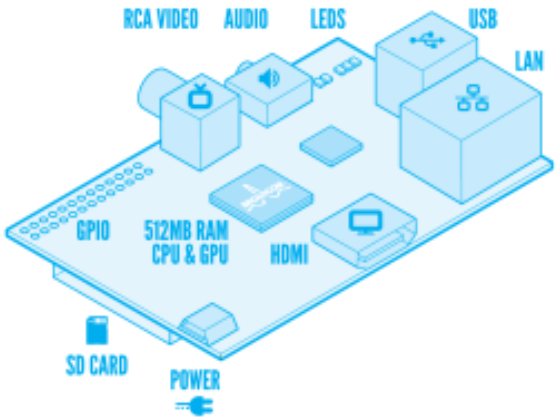

Figura 14. ***Tarjeta Raspberry Pi modelo B***

Fuente: (Pinedo, 2015)

DIFERENCIAS ENTRE EL MODELO A Y EL MODELO B

- Modelo A: 256 MB de RAM, 1 puerto USB, no tiene conexión a Ethernet.
- Modelo B : 512 MB de RAM, dos puertos USB, si tiene conexión Ethernet.

No hace falta tener conectado un monitor, teclado y mouse ya que si la tarjeta se la tiene conectada a un Router se puede acceder a ella a través de SSH o VNC (Virtual Network Computing).

1.11.1 ESPECIFICACIONES RASPBERRY PI 2

La nueva Raspberry Pi 2 es aproximadamente 6 veces más rápida que los modelos anteriores en las tareas más frecuentes. Atrás quedan también los modelos con 256 y 512 KB, ya que ahora la memoria RAM es de 1 GB.

A continuación se muestra una comparación, y especificaciones de la tarjeta Raspberry Pi 2 modelo B+ y Raspberry Pi 2 Modelo B [20]

Tabla 1.

Comparación de tarjetas Raspberry Pi 2 B y B+

Características	Raspberry Pi Model B+	Raspberry Pi Model B
SoC	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM11 ARMv6 700 MHz	ARM11 ARMv7 ARM Cortex A7 4 nucleos @ 900 MHz
Overclocking	Si, hasta velocidad Turbo; 1000 MHz ARM, 500 MHz core, 600 MHz SDRAM, 6 overlot de forma segura	Si, hasta ARM freq= 1000 SDRAM freq=500 core freq=500 over voltaje=2 de forma segura
GPU	Broadcom VideoCore IV 250 MHz. Open ES 2.0	Broadcom VideoCore IV 250 MHz. Open GL ES 2.0
RAM	512 MB LPDDR SDRAM 400 MHz	1GB LPDDR2 SDRAM 450 MHz
USB 2.0	4	4
Salidas de Video	HDMI 1.4 @ 1920x1200 píxeles	HDMI 1.4 @ 1920X1200 píxeles
Almacenamiento	microSD	microSD
Ethernet	Si, 10/100 Mbps	Si, 10/100 Mbps
Tamaño	85,60x56,5mm	85,60x56,5mm
Peso	45g	45g
Consumo	5v, 600mA	5v, 900mA, aunque depende de la carga de trabajo de los 4 cores

Fuente: (Doutel, 2015)

CAPÍTULO II

2. DESARROLLO

2.1 Análisis de flujo vehicular

La importancia del análisis de flujo vehicular radica en el entendimiento de las características y el comportamiento del tránsito para el planteamiento de proyectos de obra y operación de carreteras. Mediante la matemática y la física el análisis del flujo vehicular observa la circulación de los vehículos en cualquier tipo de vialidad y de esta manera determina el nivel de eficiencia de la operación. (Armequiza, 2010)

De la misma manera, las diferentes metodologías estadísticas para este análisis y para cualquier modelo microscópico y macroscópico de proyecto, basa su estudio en el entendimiento de las tres principales variables básicas en el estudio del flujo vehicular: *flujo, velocidad y densidad.* Las tres principales variables también dan origen a las variables asociadas: el volumen, el intervalo, el espaciamiento, la distancia y el tiempo, con las cuales se determina genéricamente la situación del proyecto para fundamento de temas posteriores. La comprensión de las tres variables de igual manera son plasmadas en modelos básicos matemáticos de cálculo tales como modelos lineales y no lineales; así mismo no puede faltar la predicción y el entendimiento de la probabilidad de flujo vehicular en un determinado proyecto, por tal motivo, el análisis probabilístico se encuentra dentro de un análisis de flujo vehicular.

De igual manera el conocimiento de las tres variables marca la pauta para el conocimiento de revestir singular importancia, ya que éstas indican la calidad o nivel de servicio experimentado por los usuarios de cualquier sistema vial; cabe destacar que el análisis de flujo vehicular generalmente en su estudio tiene la metodología como base fundamental del cálculo.

El análisis de flujo vehicular, dictamina el comportamiento y funcionamiento de la eficiencia de los vehículos sobre una determinada vía y marca la medida estándar ya sea por unidad de tiempo, por carril o por ambas a la vez.

2.1.1 Congestión vehicular

Es la obstrucción del paso de la circulación vehicular. Además se conoce como la condición en la cual existen varios vehículos circulando y cada uno de ellos avanza lentamente. La congestión aparece cuando el requerimiento del espacio vehicular es más grande que la infraestructura vial en que transcurren un determinado número de autos y/o carros de carga.

El fenómeno de la congestión está relacionado con el impedimento de unos vehículos a que los otros puedan avanzar en unas carreteras de capacidad limitada, aunque dicho fenómeno está relacionado además con la diferencia entre las expectativas que tienen los usuarios del comportamiento del sistema y la situación real que se encuentran. A medida que aumenta el tránsito, se reducen fuertemente las velocidades de circulación.

2.1.2 Causas de la congestión vehicular

Las causas de la congestión vehicular son variadas. Sin embargo, entre los factores que la provocan se encuentran:

- Rápido crecimiento poblacional y de trabajo.
- Un uso más intensivo de vehículos automotores.
- Deficiente construcción de infraestructura vial.
- Concentración de los viajes de trabajo en el tiempo.
- Deseo de escoger donde vivir y donde trabajar.

- Deseo de vivir en zonas con baja densidad de población.
- Deseo de viajar en vehículos privados.
- Un inadecuado diseño o mantenimiento de la viabilidad es causa de una congestión innecesaria.
- Ubicación de los paraderos de buses justo en puntos de una reducción en el ancho de la calzada.
- El mal estado del pavimento.
- El excesivo número de vehículos de transporte público contribuyen a agravar la congestión.

2.1.3 Flujo de saturación

El flujo de saturación es la cantidad de vehículos por horas que pueden ser acumulados por grupo de carriles asumiendo que la fase verde está al 100% del tiempo de ciclo semafórico. (Sierra, 2016)

$$S = S_0\, N\, f_w\, f_g\, f_{HV}\, f_P\, f_{bb}\, f_a\, f_{LU}\, f_{LT}\, f_{RT}\, f_{Lpb}\, f_{Rpb} \quad (1)$$

Dónde:

S: Flujo de saturación

S_0: Flujo de saturación base

N: número de carriles

f_w: Factor de ajuste por ancho de carril

f_{HV} : Factor de ajuste por vehículos pesados

f_g : Factor de ajuste por pendiente del acceso

f_p: Factor de ajuste por carril de parqueo

f_{bb}: Factor de ajuste por bloqueo local por paradas de buses

f_a: Factor de ajuste por tipo de área

f_{LU}: Factor de ajuste por utilización de carril

f_{LT}: Factor de ajuste por giros a la izquierda

f_{RT}: Factor de ajuste por giros a la derecha

f_{Lpb}: Factor de ajuste por giros a la izquierda peatonales

f_{Rpb}: Factor de ajuste por giros a la derecha peatonales.

Los factores mencionados se calculan de acuerdo a cada una de las características, geométricas y de tránsito que la intersección contiene en su estado actual. Para la evaluación del flujo vehicular, se usa los factores de ajuste por ancho de carril, pendiente del acceso, tipo de área, utilización de carril, giros a la izquierda y derecha.

2.1.4 Circulación ininterrumpida

El desplazamiento de vehículos se verifica en un ambiente físico que comprende tanto dimensiones especiales como temporales en forma simultánea. No obstante, a diferencia del transporte marítimo o aéreo donde la dimensión espacial posee dos o tres componentes, el tráfico urbano tiene una componente espacial que puede considerarse casi-unidimensional: la vía. (Huenupi, 2016)

Por otra parte, en la circulación vehicular los objetos en movimiento no solo obedecen a leyes físicas sino que son conducidos y ocupados por personas. Luego, están dotados de una dimensión humana: la voluntad. Así, la circulación vehicular es un problema complejo, pero a la vez muy interesante de analizar. [23]

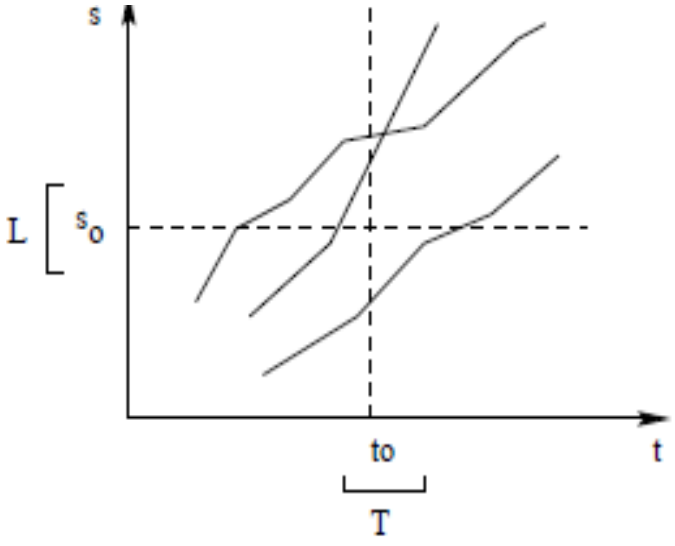

Figura 15. *Diagrama Espacio-Tiempo*

Fuente: (Huenupi, 2016)

En este caso la circulación se representa a través de un diagrama espacio-tiempo, como el de la Figura 15. En dicho diagrama se muestra la posición en cada instante de los vehículos.

2.2 Procesamiento de imágenes

El procesamiento de imágenes es un campo de investigación muy grande, porque se han desarrollado una variedad de investigaciones con énfasis en visión artificial. El crecimiento de esta área no ha sido sola, sino en conjunto con otras áreas, con las que está relacionada como son: las matemáticas, la computación, y el conocimiento cada vez mayor de algunos órganos del cuerpo humano que participan en la percepción y en la manipulación de las imágenes. Juntando a esto, la necesidad del hombre por simular algunas características del ser humano como apoyo en la solución de problemas. El avance del Procesamiento Digital de imágenes se ve reflejado principalmente en sistemas informáticos, en la medicina, la astronomía, geología, microscopia, etc. La información meteorológica, la transmisión y despliegue agilizado de imágenes por Internet tienen sustento gracias a los avances antes mencionados.

2.2.1 Representación de una imagen digital

El Término “imagen monocromática” o imagen simplemente, se refiere a una función de intensidad de luz bidimensional *f(x,y),* donde *x* e *y* indican las coordenadas espaciales y el valor de *f* en cualquier punto *(x,y)* es proporcional a la luminosidad (nivel de grises) de la imagen en dicho punto.

Una imagen digital es una imagen (función) *f(x,y)* que ha sido discretizada tanto en coordenadas espaciales como en luminosidad. Una imagen digital puede ser considerada como una matriz cuyos índices de renglón y columna identifican un punto (un lugar en el espacio bidimensional) en la imagen y el correspondiente valor de elemento de matriz identifica el nivel de gris en aquel punto. Los elementos de estos arreglos digitales son llamados elementos de imagen o pixeles. (Escalante, 2016)

En el tratamiento de imágenes se puede distinguir tres etapas principales:

1.- Adquisición de la imagen

2.- Procesamiento de la imagen

3.- Presentación al observador

Figura 16*. Etapas del tratamiento de imágenes*

Fuente: (Escalante, 2016)

En la Figura 16 se muestra las etapas del procesamiento de imágenes. La adquisición de la imagen está a cargo de un transductor o conjunto de transductores que a través de la manipulación de la luz o de alguna otra forma de radiación que es emitida por los cuerpos, se consigue construir una representación del objeto dando lugar a la imagen. Ejemplos: el ojo humano, sensores de una cámara fotográfica o de video, tomógrafos. Es importante conocer que durante la etapa de adquisición, los transductores agregan ruido a la imagen logrando una distorsión en la misma. Además, los transductores poseen una resolución limitada, lo cual afecta en la apreciación de dicha imagen. El procesamiento digital de la imagen consiste en eliminar la mayor cantidad de ruido que se agrega durante la adquisición, así como también, mejorar las características de dicha imagen en la etapa de reconstrucción de la misma tales como: definición de contornos, color, brillo, etc. valiéndose de procedimientos y herramientas matemáticas. En esta etapa también se encuentran técnicas de codificación para el almacenamiento o bien para la transmisión.

La presentación consiste en el método empleado para exponer la imagen la cual puede ser impresa o por medios electrónicos como la televisión, el monitor de una computadora, o algún otro medio. Para la presentación de la imagen se debe considerar algunos aspectos de percepción humana así como las velocidades de despliegue del dispositivo utilizado.

Algunos de los problemas más comunes en el diseño de estos subsistemas que involucran el uso de representaciones de señales son las siguientes:

- Los dispositivos sensoriales realizan un número limitado de mediciones sobre las señales de entrada; dichas mediciones deben ser acopladas para obtener aproximaciones útiles. Decidir que mediciones realizan y como usarlas de tal manera que aproximen mejor a la señal de entrada son los problemas que deben ser resueltos.
- Para la selección del procesamiento y/o codificación que se hará sobre una señal, es necesaria una interpretación de las componentes de la señal. El modelo del sistema de visión humano puede ser utilizado en ciertas etapas de procesamiento para dicha interpretación.
- Los dispositivos de despliegue sintetizan una imagen usando un número finito de respuesta básicas de despliegue, como los puntos de fosforo utilizando en un tubo de rayos catódicos. Seleccionar el tamaño y la forma de estas respuestas de despliegue, la configuración (número y posición relativa) y como pueden ser controlados de la mejor manera óptima para obtener imágenes con la calidad/fidelidad requerida son aspectos que deben ser cubiertos.

2.2.2 Análisis de imágenes

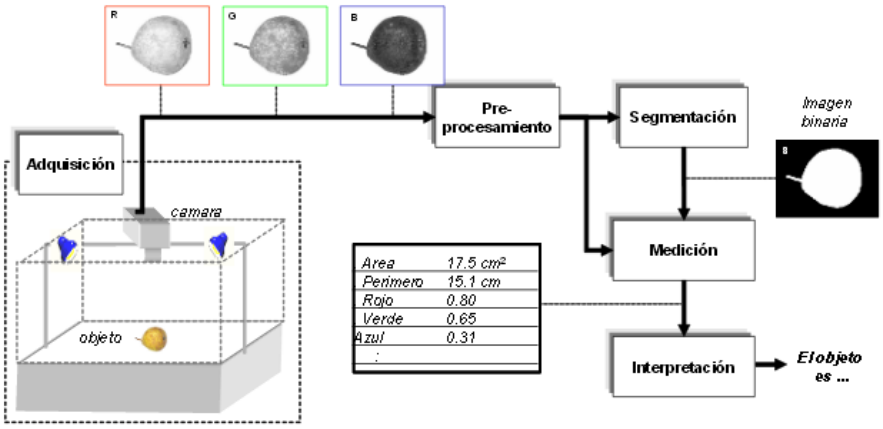

Figura 17. ***Esquema de un proceso de análisis de imágenes.***

Fuente: (Domingo, 2016)

Es el proceso por el cual a partir de una imagen se obtiene una percepción, medición o decisión.

Existen diversos ejemplos de análisis de imágenes como en los procesos de visión artificial, que abarcan procesos que al partir de una imagen se pretenden saber si un producto tiene fallas. (Domingo, 2016)

Otro ejemplo es en el área de la industria de alimentos, en donde se analizan imágenes a color con el fin de detectar grados de calidad o anomalías en los alimentos. El esquema típico del análisis de imágenes se ilustra en la Figura 17. El análisis consiste en cinco etapas:

- Adquisición de la imagen: Se obtiene la imagen adecuada del objeto de estudio. Dependiendo de la aplicación la imagen puede ser una fotografía, radiografía, termografía, etc.
- Reprocesamiento: Con el fin de mejorar la calidad de la imagen obtenida se emplea ciertos filtros digitales que eliminan el ruido en la imagen o bien aumenten el contraste.
- Segmentación: Se identifica el objeto de estudio en la imagen

- Medición (extracción de características): Se realiza una medición objetiva de ciertos atributos de interés del objeto de estudio.
- Interpretación (clasificación): De acuerdo a los valores obtenidos en las mediciones se lleva a cabo una interpretación del objeto.

2.3 Software en procesamiento de imágenes

A continuación se presentan tres programas que son los más conocidos para el procesamiento de imágenes y al mismo tiempo los más usados en el campo de visión artificial.

2.3.1 Procesamiento de Imágenes con Matlab

MATLAB es un software robusto que posee un toolbox orientado al procesado de imágenes que proporcionan un conjunto de funciones y herramientas para el análisis y visualización de imágenes digitales, así como para el desarrollo de algoritmos.

Muchas de las funciones del toolbox están orientadas en base a un sistema abierto, es decir se puede acceder a su código fuente y alterar los algoritmos según las necesidades, o incluso desarrollar otros nuevos. Mediante esta toolbox se puede restaurar o degradar las imágenes, así como extraer y analizar datos de las mismas. Estas técnicas resultan muy útiles en campos como la astronomía, el control remoto, la industria aeroespacial, la medicina y la bioingeniería. (Berzal, 2016)

2.3.2 SDC

Morphology Toolbox es un software para el análisis de imágenes y procesamiento de señales que funciona con Matlab. Está compuesto por una familia de filtros discretos, no lineales. Estos filtros, llamados operadores morfológicos, son muy útiles para restauración, segmentación y análisis cuantitativo de imágenes y señales.

Las SDC Morphology Toolbox´s trabajan con escala de grises e imágenes binarias. Así, la mayoría de los operadores realizan el procesamiento en escala de grises, imagen binaria o también la selección automática de la imagen apropiada para el algoritmo. Las imágenes pueden ser representadas por los formatos: binario, 8- báscula de bit-gris y 16-báscula de bit-gris, donde cada pixel es representado, respectivamente, por un tipo de datos lógico. Los operadores morfológicos pueden ser escritos jerárquicamente a partir de los operadores elementales: dilatación y erosión. SCD tiene unas herramientas muy eficientes para la dilatación y erosión, sin embargo, para obtener más eficiencia, varios operadores son tratados también por algoritmos rápidos espaciales. Algunos de estos operadores son de transformación de distancia, reconstrucción, etiqueta y apertura de áreas. Las dilataciones y erosiones son parametrizadas por imágenes particulares, llamadas elementos estructurantes. (Berzal, 2016)

Figura 18*. Logo del Software SDC*

Fuente: (Berzal, 2016)

La SCD Morphology Toolbox funciona bajo 3 plataformas: Win 95/98/NT, Linux y Solaris. SDC depende de la versión de Matlab 5, o superior. No dependen de ninguna otra toolbox.

2.3.3 VTK

VTK (Visualization Toolkit) es un conjunto de librerías de código y distribución libres destinadas a la visualización y el proceso de imágenes, basadas en la programación orientada a objetos. Son muy amplias y complejas, pero aun así, están diseñadas para ser sencillas de emplear con cualquier lenguaje de programación orientado a objetos, como pueden ser C++,

Java, Tcl. Son capaces de realizar operaciones sobre imágenes en dos y tres dimensiones y de generar modelos en las mismas con pocas líneas de código. Debido a su gran potencia, se hacen necesarios amplios recursos de memoria en el PC para poder aprovechar en su totalidad sus funcionalidades.

Figura 19. ***Logo del Software VTK***

Fuente: (Berzal, 2016)

El modelo gráfico de VTK posee un nivel de abstracción mucho mayor que el de otras librerías de procesamiento de imágenes como OpenGL o PEX. Esto se traduce en una mayor sencillez a la hora de implementar aplicaciones gráficas o de visualización con VTK. Además, las aplicaciones creadas empleando VTK pueden ser escritas directamente en Tcl, Java, Python o C++, lo que aumenta y facilita la posibilidad de implementar aplicaciones en poco tiempo.

Por otra parte, este software es un sistema de visualización que no solo permite visualizar geometría, sino que además soporta una amplia variedad de algoritmos de visualización, incluyendo métodos escalares, vectoriales, tensores, de textura y volumétricos, además de otras modernas técnicas de modelado, como la reducción poligonal, el contorno, etc.

2.3.4 OpenCv

OPENCV (Open Source Computer Visión Library), es una librería de funciones, para programar en tiempo real, soporta plataformas como Linux y Windows. Es la libreria más usada por los programadores en la actualidad para el desarrollo de Visión Artificial. El proyecto se ejecuta en base a ésta librería.

A continuación se mencionan algunas características más relevantes que posee la librería OpenCV:

- Manipulación de los datos de la imagen
- Imagen y I/O de video (archivos de video, entrada de video en tiempo real)
- Procesamiento básico de imagen (filtrado, detección de bordes, muestreo e interpretación, conversión de color, operaciones morfológicas)
- Análisis estructural (componentes conectados, procesamiento de contorno, transformar la distancia, aproximación poligonal, ajuste de línea, elipse)
- Calibración de la cámara (búsqueda y seguimiento de los patrones de calibración, estimación de la matriz fundamental)
- Análisis de movimientos (flujo óptico, segmentación de movimiento, seguimiento)
- Reconocimiento de objetos (Agam, 2016)

Figura 20. ***Logo de librería OpenCV***

Fuente: (Agam, 2016)

2.4 Descripción de la cámara

Se usa la cámara web Klip Xtreme que se muestra en la Figura 20, por las altas prestaciones de video que posee con respecto a las webcam

convencionales, a continuación se muestran las características y funcionalidad que posee la misma.

La cámara web Klip Xtreme, cuenta con el balance automático de blancos para garantizar la reproducción precisa de colores, aun cuando la luz sea muy tenue. Para hacer éste dispositivo aún más práctico, la base universal con clip permite fijar la cámara web en el pedestal construido para la elaboración de este proyecto como se especifica en la sección de posicionamiento de la cámara web. También se puede fijar en el computador, pantalla plana o monitor LCD, o se puede plegar para usarla como base independiente ya sea sobre el escritorio o una repisa. (Xtreme, 2016)

Características Adicionales

A continuación se muestran las características de la cámara que ayudan al desarrollo del proyecto.

- Resolución de video mejorada a través de software hasta 1600 x 1200 pixeles.
- Sensor APS de detección de luz para un mejor manejo de iluminación y contraste.
- Seguimiento automático, permite centrar al sujeto y seguir su movimiento dentro del marco de referencia.
- Captura de imágenes hasta 30 fps (cuadros por segundo)
- Formato de imagen RGB24 y YUY2
- Interfaz USB 2.0
- Detección automática Plug and Play
- Longitud del cable 1.5 m
- Formato de video: 24 bit true color
- Interfaz: USB 2.0 y compatible con USB 1.1
- Distancia focal: 30mm (VGA)

Figura 21. ***Camara Klip Xtreme***

Fuente: (Xtreme, 2016)

2.5 Puntos de análisis de la imagen

Los puntos de análisis en tiempo real que se utiliza para la elaboración de este proyecto poseen las características que se muestran a continuación, con ayuda de algoritmos y métodos de detección de vehículos.

2.5.1 Imagen a escala de grises

La imagen a escala de grises es la que se modifica la intensidad de colores del rango requerido RGB [0-255]. También llamada imagen de intensidad, donde cada pixel representa un valor en la escala como se puede ver en la Figura 22, teniendo en cuenta que la escala va desde 0 que representa el color negro hasta 255 que representa el color blanco.

Figura 22. ***Imagen a color(A), imagen a nivel de grises (B)***

Fuente: (Baque J. , 2016)

2.5.2 Imágenes binarias

La imagen binaria es primordial para los procesos de filtrado, porque es un arreglo que solo contiene unos y ceros, donde cada pixel toma valores de ceros y unos para la representación de ciertos objetos, como se presenta en la Figura 22 (B).

Los unos y ceros son especiales, porque no implican valores numéricos sino más bien son banderas de activación que indican el estado de falso (0) o verdadero (1) en el proceso de un determinado algoritmo, en este caso es el seguimiento del contorno de un auto. En la Figura 23 (C), se representa en mapa de bits las banderas de activación (unos y ceros) partiendo de una imagen a escala de grises como se muestra en la Figura 22 (A). (Baque J. , 2015)

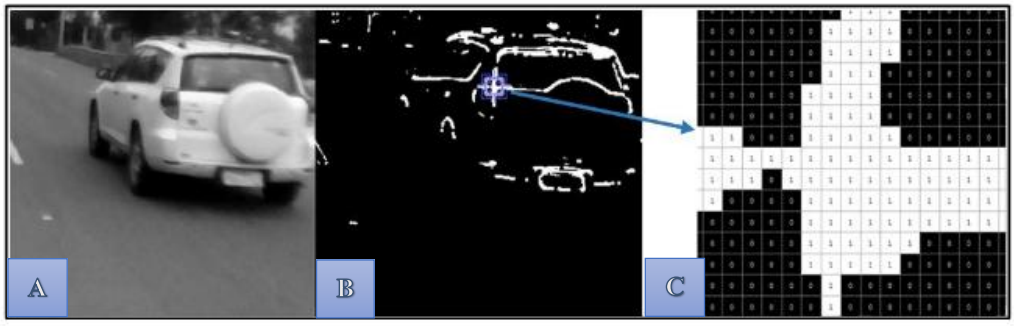

Figura 23. ***Imagen a nivel de escala de Grises(A), Imagen Binaria(B), Representación en mapa de Bits de la Imagen Binaria (C)***

Fuente: (Baque J. , 2015)

2.5.3 Características

En visión artificial, el proceso de detección de objetos se realiza por medio de características propias como son el color, tamaño o forma, pero esto no suele ser suficiente porque en la mayoría de los casos, algunos objetos pueden compartir estas mismas singularidades o incluso que dichas características cambien para un mismo objeto en función de las condiciones en las que se distingan (distancia, perspectiva, iluminación). Pero existen técnicas desarrolladas como la umbralización o la detección de bordes que se utiliza

para la elaboración de éste proyecto, para encontrar ciertas características de cada uno de los vehículos. Hay casos que dependiendo de la aplicación no son suficientes. [26]

A continuación se mencionan cuatro características para la detección de objetos con su respectiva particularidad de operación.

- Características HAAR: Detección de caras y peatones
- Características HAAR-LIKE FEATURES: Detección de objetos
- Características LBP: Patrones binarios locales
- Características HOG: Histograma de los gradientes orientados

2.6 Técnicas de identificación de vehículos

Existen varias técnicas para la detección, seguimiento y conteo de vehículos entre las principales se tiene: Haar Cascade, esta es una técnica sencilla para el reconocimiento de objetos, al utilizar ésta técnica se debe entrenar al algoritmo para que pueda reconocer al objeto pero se dificulta la detección cuando el objeto adopta un cambio de posición diferente o también cuando el objeto tiene una forma diferente para la que fue entrenado. La otra técnica es Background Subtractor (sustracción de fondo), con ésta lo que se hace es solo tomar los objetos que se encuentran en movimiento independientemente con lo que tenga de fondo la imagen, siempre y cuando la cámara se encuentre sin movimiento.

2.6.1 Haar Cascade

Detección de objetos utilizando Haar, se basa en características como clasificadores en cascada que es un método de detección de objetos efectivo propuesto por Pablo Viola y Michael Jones en su artículo, " Rapid Object Detection using a Boosted Cascade of Simple Features " en el año 2001. Trata de un enfoque basado en una función cascada, que es entrenado para una gran cantidad de imágenes positivas y negativas. De esta manera se utiliza para detectar objetos en otras imágenes. (OpenCV 3.0.0, 2015)

A continuación se muestra un ejemplo para trabajar en la detección de rostros. Inicialmente, el algoritmo necesita una gran cantidad de imágenes positivas (imágenes de rostros) e imágenes negativas (imágenes sin caras) para entrenar el clasificador. De esta manera se extraer las características de la misma. Para esto, se utilizan características Haar que se muestran a continuación en la Figura 24.Cada característica es un solo valor obtenido restando la suma de los píxeles bajo rectángulo blanco de la suma de píxeles bajo rectángulo negro. (OpenCV 3.0.0, 2015)

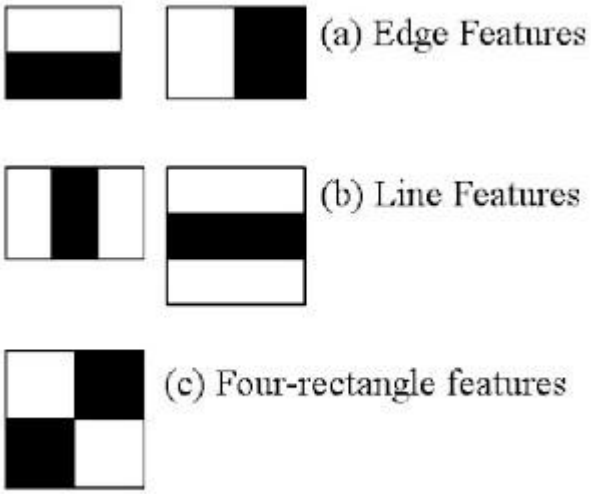

Figura 24. ***Entrenador HAAR CASCADE***

Fuente: (OpenCV 3.0.0, 2015)

Ahora todos los tamaños y ubicaciones posibles de cada núcleo se utilizan para calcular varias características. Para cada cálculo de función, se tiene que encontrar la suma de los píxeles en función de rectángulos blancos y negros. Para solucionar esto, se introducen las imágenes. Se simplifica el cálculo de la suma de píxeles, lo grande que puede ser el número de píxeles, a una operación que implica sólo cuatro píxeles.

Pero entre todas estas características que se calcularon, la mayoría son irrelevantes. Por ejemplo, considere la Figura 25. La fila superior muestra dos características. La primera característica seleccionada parece centrarse en la zona de los ojos, es a menudo más oscura que la zona de la nariz y las mejillas. La segunda característica seleccionada se basa en la propiedad de que los ojos son más oscuros que el puente de la nariz. Pero las mismas ventanas aplicando en las mejillas o en cualquier otro lugar son irrelevantes.

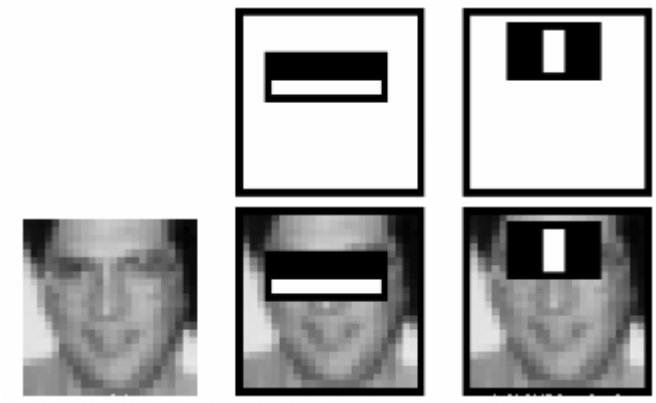

Figura 25. ***Ejemplo Clasificador HAAR***

Fuente: (OpenCV 3.0.0, 2015)

Para seleccionar las mejores características se lo realiza mediante el comando Adboost. Para cada característica, se encuentra el mejor umbral que clasifica las caras positivas y negativas. Pero, obviamente, no habrá errores o errores de clasificación. Se selecciona las características con una tasa de error mínimo, lo que significa que son las características que mejor se clasifica el rostro.

El clasificador final es una suma ponderada de estos clasificadores débiles. Se llama débil porque por sí solo no puede clasificar la imagen, pero en conjunto con los demás constituye un fuerte clasificador.

En una imagen, la mayor parte de la región de imagen es la cara. Por lo tanto, es una buena idea tener un método sencillo para comprobar si una ventana no es una región de la cara. Si no es así, se elimina, no procesa de nuevo. De esta manera se centran en la región que no puede haber una cara. Así, se facilita el proceso de encontrar rápidamente la cara.

2.6.2 Background Subtractions

El método sustracción de fondo es uno de los principales pasos de pre-procesamiento de imágenes en muchas aplicaciones basadas en visión artificial. Por ejemplo, se considera, el contador de visitas, donde una cámara

estática lleva el número de visitantes que entren o salgan de la habitación, o una cámara de tráfico de información acerca de la extracción de los vehículos, etc. En todos estos casos, primero que hay que extraer la persona o vehículos por sí solos. Técnicamente, es necesario extraer el primer plano en movimiento de fondo estático. Como se muestra en la figura 26.

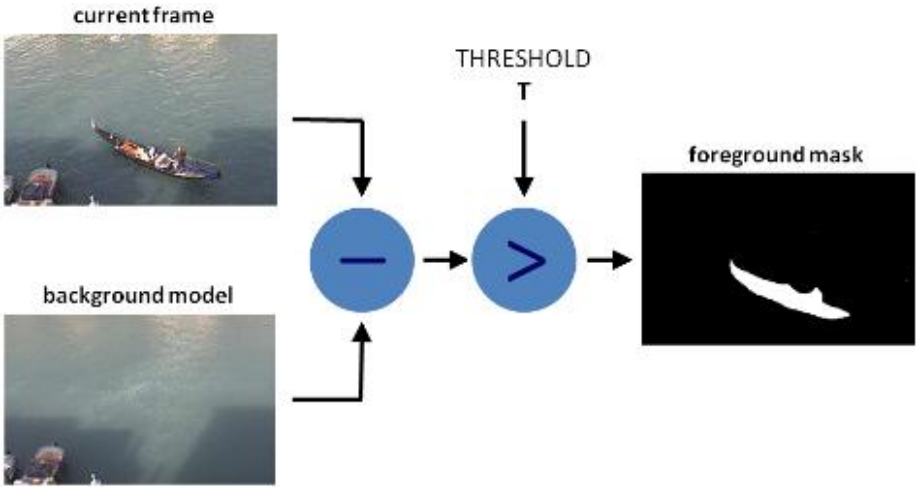

Figura 26. ***Diagrama de aplicación de Background Subtraction***

Fuente: (OpenCV 3.0.0, 2015)

El modelado de fondo consiste en dos pasos principales:

1. Inicialización en segundo plano;
2. Actualización de fondo.

En el primer paso, se calcula un modelo inicial de los antecedentes, mientras que en el segundo paso, el modelo se actualiza con el fin de adaptarse a los posibles cambios en la escena.

Si se tiene una imagen de fondo solo, como la imagen de una habitación sin personas, la imagen de carretera sin vehículos, etc. Es un trabajo fácil. Sólo se resta la nueva imagen del fondo. Se obtiene los objetos en primer plano solo. Pero en la mayoría de los casos, pueda que no tenga una imagen de esta manera, por lo que se necesita extraer el fondo de cualquier imagen que se tenga. Se vuelven más complicados cuando hay sombra de los vehículos principalmente. Desde la sombra también se mueve. Se complica las cosas.

Se introducen varios algoritmos para este propósito. OpenCV ha puesto en marcha tres de estos algoritmos, que son fáciles de usar. A continuación se describe uno por uno.

2.6.2.1 Background Subtractor MOG

Fue introducido en el artículo "An improved adaptive background mixture model for real-time tracking with shadow detection" por P. y R. KadewTraKuPong Bowden en 2001. Utiliza un método para modelar cada píxel del fondo por una mezcla de distribuciones gaussianas K (K = 3 a 5). Los pesos de imagen de la mezcla representan las proporciones de tiempo que esos colores se quedan en la escena. Los colores de fondo probables son los que permanecen más tiempo y más estática.

Figura 27. ***(A)Imagen original (B) Imagen aplicada la sustracción de Fondo MOG***

Fuente: (OpenCV 3.0.0, 2015)

Para la codificación, se debe crear un objeto de fondo usando la función cv2.createBackgroundSubtractorMOG(). Tiene algunos parámetros opcionales, como la longitud de la imagen, el número de mezclas gaussianas, etc. El umbral que está listo para algunos valores por defecto. En el interior del bucle de vídeo, se utiliza el método backgroundsubtractor.apply () para obtener la máscara de primer plano. Como se muestra en la figura 27.

2.6.2.2 Background Subtractor MOG2

Se basa en dos artículos, por Z.Zivkovic, "Improved adaptive Gausian mixture model for background subtraction" en 2004 y "Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction" en 2006. Una característica importante de este algoritmo es que selecciona el número apropiado de distribución de Gauss para cada píxel. Proporciona una mejor adaptabilidad a escenas distintas, debidos cambios de iluminación, etc. Un ejemplo en la figura 28.

Figura 28. ***(A)Imagen original (B) Imagen aplicada la sustracción de Fondo MOG2***

Fuente: (OpenCV 3.0.0, 2015)

Al igual que en el caso anterior, se debe crear un objeto de fondo restador. Aquí, existe la posibilidad de seleccionar si la sombra que se detecte aparezca o no. Si detectShadows = true (es por defecto), que detecta y marcas de sombras, pero disminuye la velocidad. Las sombras se marcan en color gris.

2.6.2.3 Background Subtractor GMG

Este algoritmo combina la estadística de estimación de imagen de fondo y la segmentación por píxel. Se introdujo por Andrew B. Godbehere, Akihiro Matsukawa, Ken Goldberg en su documento "Visual Tracking of Human Visitors under Variable-Lighting Conditions for a Responsive Audio Art Installation" en 2012. Los resultados al aplicar este algoritmo se muestra en la figura 29.

Figura 29. ***(A)Imagen original (B) Imagen aplicada la sustracción de Fondo GMG***

Fuente: (OpenCV 3.0.0, 2015)

Se utiliza de forma predeterminada los 120 primeros marcos para el modelado de fondo. En primer plano se emplea un algoritmo de segmentación que identifica posibles objetos en el uso de la inferencia bayesiana. Las estimaciones son de adaptación, nuevas observaciones están más fuertemente ponderadas que las observaciones antiguas para dar cabida a la iluminación variable. Varias operaciones de filtrado morfológicas como cierre y la apertura se realiza para eliminar el ruido no deseado.

2.7 Filtrado de imágenes

El proceso de filtrado consiste en cambiar las características de una imagen, añadiendo y/o quitando ruido. A continuación se describe las diferentes transformaciones que se usó en el desarrollo del proyecto.

2.7.1 Transformaciones morfológicas

Las Transformaciones Morfológicas (*Morphological Transformations* o *Morph Transforms*) son operaciones que se puede hacer sobre una máscara para transformarla. Por ejemplo, un ‘close’ sirve para eliminar manchas negras dentro de la zona blanca de la máscara (imagen que se quiere procesar). ‘Erode’ elimina las regiones blancas y las reduce. (ROBOLOGS, 2015)

Para hacer cualquier Transformación Morfológica se necesitan dos cosas: la máscara original y un kernel.

Si se decide mostrar la máscara por la pantalla, OpenCV mostrará una imagen negra con manchas blancas.

En cuanto al kernel (o matriz de convolución) se trata de un array más pequeño lleno de valores como se muestra en la Figura 30 en un ejemplo de convolución de una imagen a color.

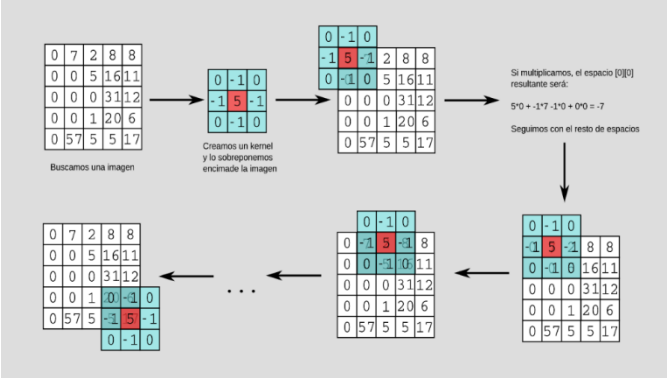

Figura 30. ***Ejemplo de convolución de una imagen a color***

Fuente: (ROBOLOGS, 2015)

2.7.1.1 Erosión

Figura 31. ***Ejemplo de una imagen aplicando Erosión***

Fuente: (ROBOLOGS, 2015)

Para entender el operador ‘Erosión’ hay que tomar en cuenta el efecto del agua sobre una roca que se va desgastando por el exterior.

El operador Erosión hace lo mismo porque reduce el tamaño de los objetos blancos de la máscara “comiéndose” su límite. A la hora de hacer la convolución, el píxel será ‘1’ sólo si todos los píxeles que quedan debajo el kernel también son ‘1’. En la figura 31 se muestra un ejemplo.

2.7.1.2 Dilation

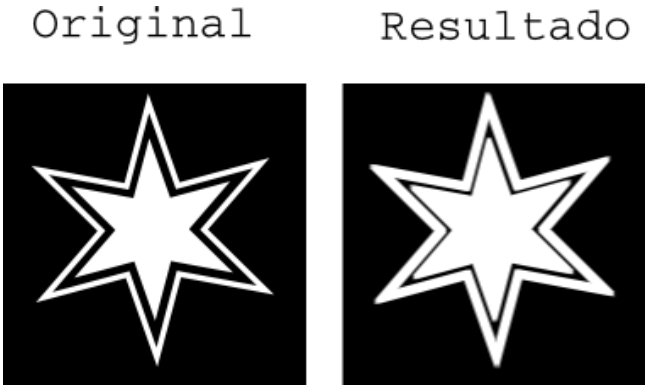

Figura 32. ***Ejemplo de una imagen aplicando Dilation***

Fuente: (ROBOLOGS, 2015)

Esta operación hincha la zona blanca de la máscara. Al hacer la convolución, el píxel resultante será ‘1’ si al menos un píxel que quede debajo del kernel también es ‘1’. Como se muestra en la figura 32.

2.7.1.3 Morphological Gradient

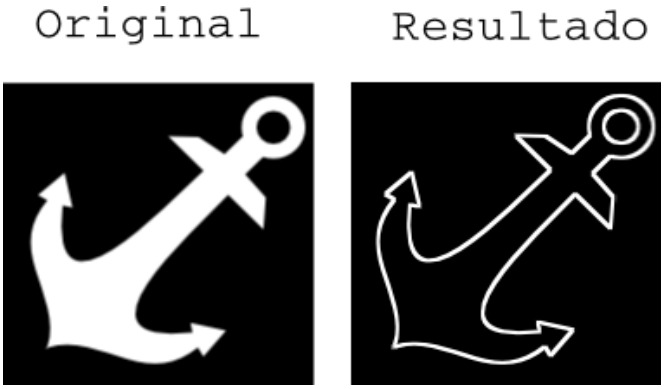

Figura 33. *Ejemplo de una imagen aplicando Morphological Gradient*

Fuente: (ROBOLOGS, 2015)

No es más que la diferencia entre el ‘Dilation’ y el ‘Erode’. Es muy útil para encontrar la silueta de los objetos

2.7.1.4 Opening

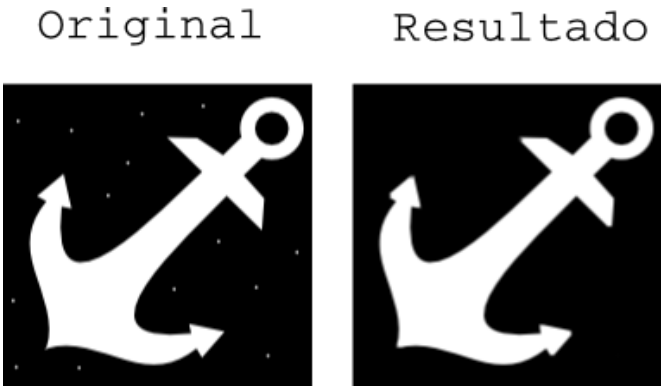

Figura 34. ***Ejemplo de una imagen aplicando Opening***

Fuente: (ROBOLOGS, 2015)

Es un ‘Erosion’ seguido de un ‘Dilation’. Nos sirve para eliminar el ruido blanco sobre las zonas negras.

2.7.1.5 Closing

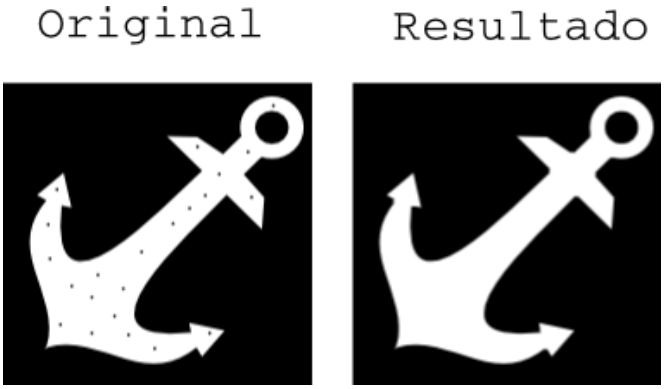

Figura 35. ***Ejemplo de una imagen aplicando Closing***

Fuente: (ROBOLOGS, 2015)

Justo lo contrario que el Opening, el Closing es un ‘Dilation’ seguido de un ‘Erosion’. Si hay ruido negro en las áreas blancas, el Closing lo limpiará.

2.7.1.6 Umbral simple

Aquí, el asunto es sencillo. Si el valor de pixel es mayor que un valor umbral, se le asigna un valor (puede ser de color blanco), de lo contrario se le asigna otro valor (puede ser de color negro). La función utilizada es cv2.threshold. El primer argumento es la imagen de origen, que debe ser una imagen de escala de grises. Segundo argumento es el valor umbral que se utiliza para clasificar los valores de los píxeles. En tercer argumento es el Maxval que representa el valor que debe darse si el valor de pixel es más que (a veces menos) el valor de umbral. OpenCV proporciona diferentes estilos de umbral y se decide por el cuarto parámetro de la función. (OpenCV 3.0.0, 2015)

2.8 Servidor FTP

El servidor FTP se usa para la transferencia de archivos en la Tarjeta Raspberry pi. A continuación se describe el funcionamiento del servidor usado en la ejecución del proyecto. (Cobo, 2016)

El servidor FTP permite copiar fichero de muchos ordenadores diferentes de todas las partes de la nube. Estos ficheros contienen todo tipo de información que se puede guardar en un computador.

Utiliza un sistema de tipo Cliente/ Servidor. Es decir es necesario ejecutar un programa cliente en un computador que será en encargado de conectarse al programa servidor, que se encuentra en un ordenador remoto.

FTP utiliza dos modos de transferencia de archivos.

- Modo Texto: Utiliza caracteres Ascci y emplea retorno de carácter de nueva línea
- Modo Binario: Cuando funciona en este modo no realiza ninguna conversión de formato.

CAPÍTULO III

3. PRUEBAS EXPERIMENTALES Y ANÁLISIS DE RESULTADOS

A continuación se detalla el manejo de la tarjeta Rapberry Pi con su adecuado Sistema Operativo, para el desarrollo del proyecto, además de explicar el posicionamiento de la cámara y el algoritmo que se utiliza.

3.1 Reconocimiento del entorno de programación e Instalación del software requerido

Como para cualquier ordenador lo primero que se debe hacer es elegir el sistema operativo que se va utilizar. Aunque la tarjeta es compatible con Windows 10 como también con Ubuntu Mate y otros sistemas, se eligió trabajar con el sistema operativo Raspbian, que es un Debian Linux adaptado a las capacidades de la tarjeta Raspberry Pi, éste es el más versátil y el que permite instalar y modificar más cosas, además de poder convertirlo en un Sistema Operativo similar a los que generalmente se usa.

La manera más sencilla de iniciar a la tarjeta Raspberry Pi es con Noobs (New Out Of the Box Software) que es un instalador de distribuciones creado para trabajar con la Raspberry Pi el cual es muy sencillo y básico de manejar.

Figura 36. ***Logotipo del Sistema Operativo Raspbian***

Fuente: (Raspberry Pi Foundation, 2016)

Para que la Raspberry Pi funcione con este sistema operativo, se debe descargar la imagen del sistema operativo, la cual se la encuentra en la web de descargas oficial de la Raspberry Pi. Una vez descargado el ZIP, hay que

descomprimirlo y cargar los archivos que vienen dentro de la carpeta en una tarjeta micro SD, pero formateándola de modo que sea booteable, la capacidad de la micro SD depende de lo que necesita almacenar o instalar en la Raspberry Pi, como mínimo se debe utilizar una de 4 GB, luego se debe conectar la Raspberry a un monitor con un cable HDMI como también un teclado y un mouse para poder instalar el sistema operativo y poder utilizarlo.

Al momento de introducir la micro SD en la Raspberry Pi y alimentarla con una fuente energía, se muestra la pantalla que se presenta en la Figura 37. Con la opción de instalar el sistema Raspbian previamente cargado en la micro SD.

En esta ventana se debe escoger el S.O. que se desee instalar en tarjeta, si se tiene una conexión a internet se presentan más opciones de instalación. Se presenta en la figura 37

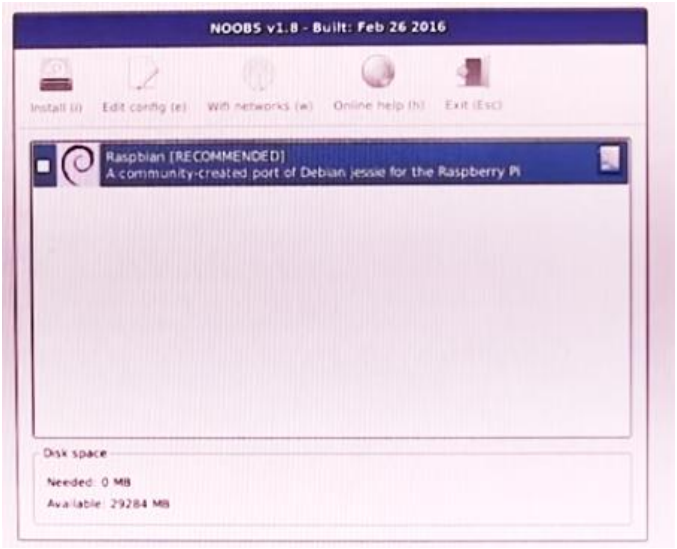

Figura 37. ***Instalación del S.O. Raspbian***

Terminada la instalación del sistema Raspbian, la tarjeta inicia la interfaz gráfica de la misma, el escritorio del sistema es el siguiente. Ver figura 38.

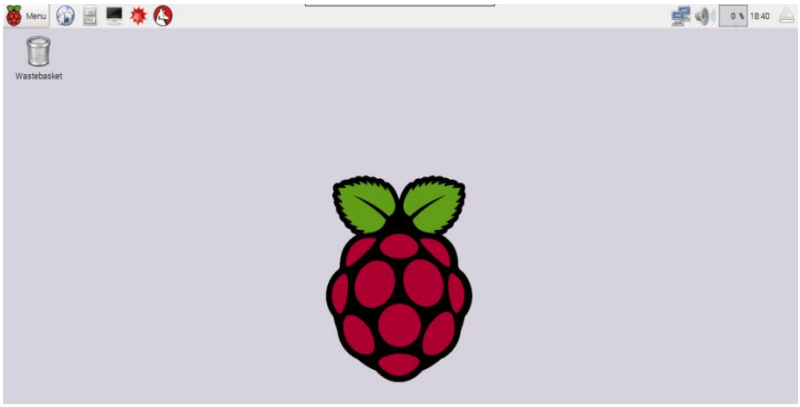

Figura 38. ***Escritorio de Raspbian***

Como la aplicación a la que se la sometió la tarjeta es la de visión artificial, se debe aprovechar al máximo los recursos que ofrece el Sistema Operativo Raspbian, por esta razón la codificación del programa está desarrollada en Python, este lenguaje de programación viene ya precargado en el sistema operativo Raspbian, por lo que no fue necesario instalar otro lenguaje de programación. La figura 38 muestra el escritorio del sistema operativo.

La instalación del software OpenCv se lo hace como en cualquier sistema operativo de Ubuntu, se lo descarga mediante la terminal y la utilización de códigos propios para la instalación, la diferencia es que para la tarjeta se debe descargar algunas dependencias extras que se necesita para el desarrollo del proyecto como es el de la cámara web, ya que se utiliza una genérica que va conectada al puerto USB de la tarjeta.

El OpenCv instalado es 3.0.0, las librerías de este software en una PC normal no ocupan mucho espacio, pero en la tarjeta es primordial el espacio que se va ocupar, por lo que se ocupó una micro SD de 16 GB de capacidad, ya que esta capacidad es la adecuada para que se puedan instalar las librerías que se necesitan para el desarrollo de la aplicación. Ver figura 39.

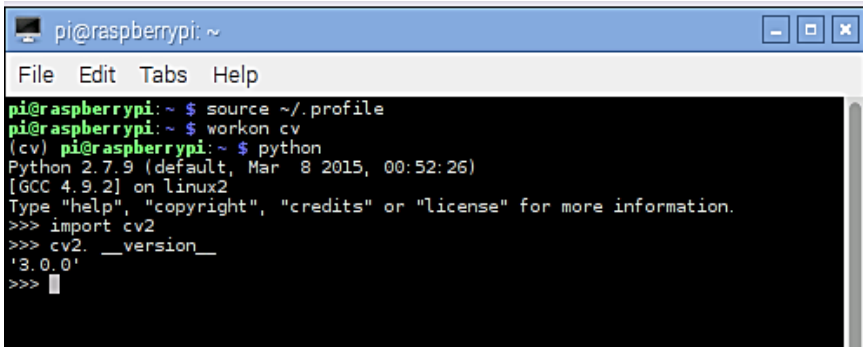

Figura 39. ***Librerías de OpenCv instaladas correctamente***

Para poder editar el *código* de programación en Python, en la terminal se debe editar *nano ejemplo.py*, esto para crear un nuevo documento de Python (figura 40) o también se puede dirigir al icono de *Menu* y en *programming* seleccionar el IDLE de Python (figura 41), luego para correr el programa en la terminal se debe editar *python ejemplo.py.*

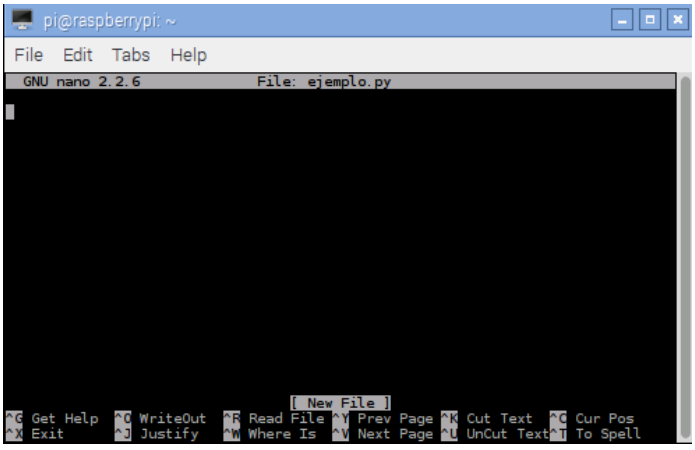

Figura 40. *Interfaz de programación desde la terminal*

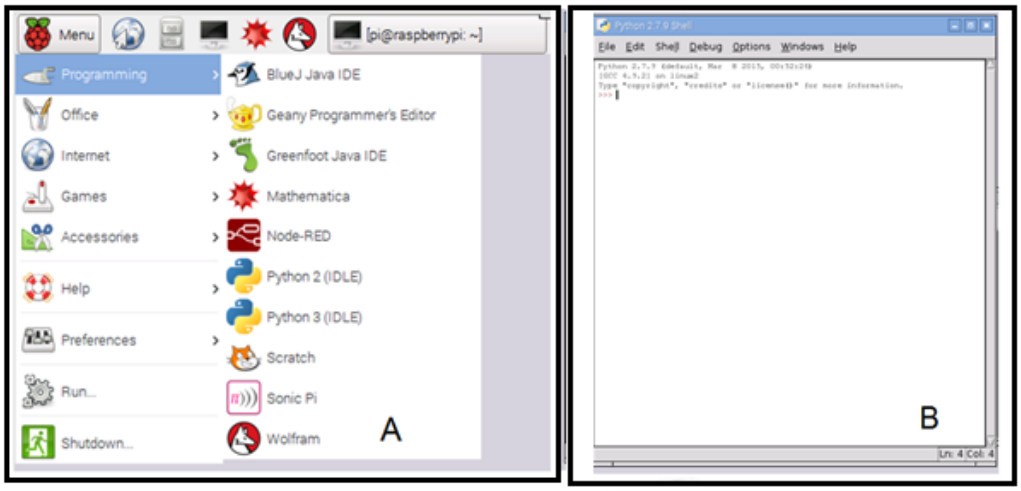

Figura 41. ***A. selección del IDLE desde Menu, B. interfaz del IDLE Python***

3.2 Posicionamiento de la cámara.

Para el correcto posicionamiento de la cámara se realizaron diferentes pruebas de funcionamiento del programa de control, para de esta manera establecer una posición adecuada al funcionamiento versátil del algoritmo implementado.

Para tener mejores resultados con la adquisición de imágenes se tomaron diferentes muestras con variación en altura y enfoque hacia los vehículos, teniendo en cuenta que la mejor forma para que el vehículo se encuentre dentro del foco de la cámara es a un ángulo de 45° respecto al plano horizontal, como también a una altura mínima de 3.5 m o superiores, ya que a partir desde esta altura los vehículos entran dentro de foco de la cámara, pero para probar con camiones largos la altura debe ser superior a la propuesta, por lo que las pruebas también se las hicieron a una altura de 5.5 m obteniendo mejores resultados al momento de identificar vehículos pesados con una mayor altura.

En la Figura 42 se muestra el posicionamiento de la cámara en el plano horizontal y la ubicación final de la cámara web para el desarrollo del proyecto.

Figura 42. ***Angulo de inclinación respecto al eje horizontal(A), Ubicación de la cámara web(B)***

Para la primera prueba realizada a la cámara se la ubicó enfocándole al vehículo en la parte lateral, como se muestra en la Figura 43. A una altura de 4 metros y con un ángulo de inclinación de 45° en el eje horizontal, al ejecutar el programa en el computador se divisa que no hay ningún tipo de inconvenientes, pero al momento de utilizar la tarjeta los resultados no fueron los mismos, porque los vehículos que pasan a una velocidad superior a los 40 km/h, casi no se los detecta, por lo que esta configuración de la cámara no es la óptima para el desarrollo de la aplicación.

Figura 43. ***Enfoque de la cámara lateral del vehículo***

En la segunda configuración se enfoca al vehículo de frente, a una altura de 4 m y un ángulo de 45° respecto al eje horizontal como se muestra en la Figura 44, esta alternativa es aceptable para la identificación de los vehículos livianos, pero que no circulan muy seguidos, los problemas se dan al momento de identificar un vehículo pesado y uno liviano al mismo tiempo, las dimensiones con las que se tiene que configurar para poderlos detectar había una variación al momento que entran dentro del enfoque de la cámara, por lo que se dificulta tomar una medida exacta de los vehículos pesados y livianos.

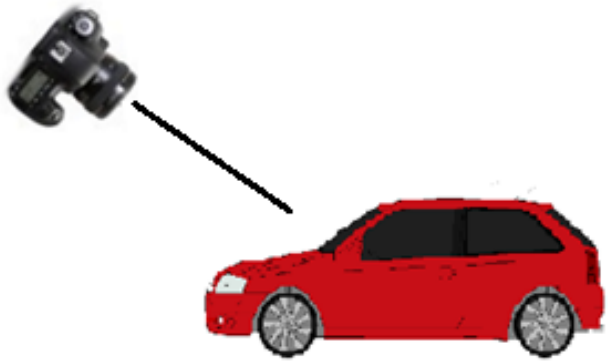

Figura 44. ***Enfoque frontal de la cámara al vehículo***

Para la siguiente prueba que se realizó, los resultados con respecto al anterior no varían mucho, ya que al momento de tomar las dimensiones de los vehículos éstas sufrían una variación cuando circulan vehículos casi unidos de forma continua, conocidos como vehículos con un mismo bloque, por lo que se dificulta el poder identificar al vehículo de forma independiente, la cámara enfoca al vehículo en la parte posterior y con un una altura de 4 m y un ángulo de 45° con respecto al eje horizontal (eje x) como se muestra en la Figura 45.

En la sección 3.4, se puede apreciar el análisis porcentual del posicionamiento de la cámara en todos los sectores que se realizó las pruebas.

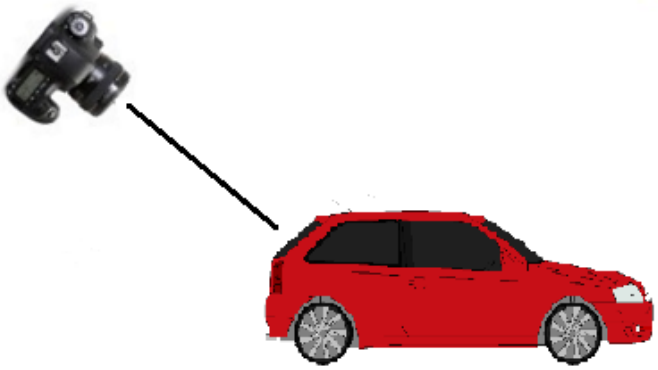

Figura 45. ***Enfoque de la cámara posterior al vehículo***

Con la última prueba que se realizó, se obtuvo mejores resultados, la configuración de la cámara está enfocando a la parte trasera-lateral del vehículo con una altura de 4 m y a un ángulo de 45°con respecto al eje horizontal (x), con éste posicionamiento de la cámara se pudo reducir los errores que se tenía con las anteriores configuraciones, en las cuales no se

podía discriminar la presencia de dos vehículos que circulaban en una forma continua y además se puede detectar cuando circula un carro pesado mediante las dimensiones apropiadas del contorno del foco de la cámara como se muestra en la Figura 46(B). Los vehículos que pasan por el sector entran dentro del foco de la cámara, por lo que es más fácil la toma de dimensiones de los vehículos, de esta manera la tarjeta actúa con una mejor eficiencia.

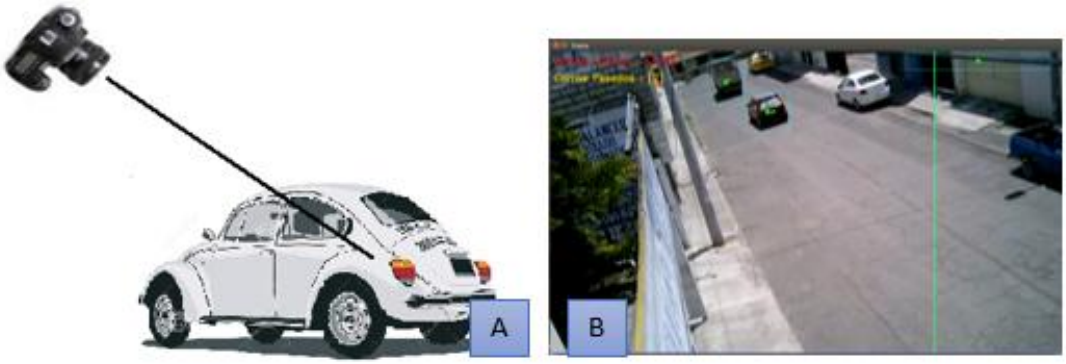

Figura 46. ***Enfoque de la cámara lateral-trasera al vehículo(A), discriminación de un carro liviano – pesado (B)***

3.3 Implementación del algoritmo de control

Al momento de realizar la implementación, se evaluaron computacionalmente varias secuencias de video para obtener un algoritmo robusto, fiable y sencillo que sea capaz de detectar y dar seguimiento a los vehículos y luego poder codificar el mismo programa en la Raspberry Pi para ser probado en la calle en donde se hicieron las pruebas con la misma, mostrándonos como resultados de que aparte de utilizar la técnica de seguimiento y detección que es la de Background Subtraction, se debe ocupar una serie de filtros adicionales que apoyan al seguimiento e identificación de los vehículos. También ayudan a eliminar ruidos generados por varios factores al momento de analizar una secuencia de imágenes.

Los filtros a los que son sometidos las secuencias son transformaciones morfológicas, éstas son algunas operaciones sencillas basadas en la forma de la imagen, normalmente se las lleva a cabo en las imágenes binarias. Para el

desarrollo de dichas transformaciones se necesita de dos entradas, una es la imagen original y la segunda a la que se le llama elemento estructurante o núcleo que decide la naturaleza de la operación.

Inicialmente es necesario suavizar a la imagen original aunque ello provoque pérdida del detalle de la imagen, esto para poder solucionar el problema del ruido de compresión, cuantización y de sensibilidad que se da al momento de obtener una imagen con una cámara. Esto se lo realiza por medio de un filtro lineal gaussiano como se muestra en la Figura 47.

Figura 47. ***Aplicación del filtro gaussiano a la imagen original*****.**

Luego a la imagen que da como resultado del filtro gaussiano se la transforma a escala de grises como se ilustra en la Figura 48, para de esta forma ayudar al programa a familiarizarse con las graduaciones de grises entre el blanco y el negro como también para medir los valores reales del modelo que se está representando. Esto debido a los efectos del contraste simultáneo, ya que cuando un tono claro y otro oscuro entran en contacto, hace que se produzcan ilusiones ópticas que perturban el grado de luminosidad que los dos son percibidos.

Figura 48. ***Transformación de la imagen de filtro gaussiano a escala de grises***

Después de tener a la imagen a escala de grises se aplica Background Subtractor o conocida también como selección de primer plano como se ilustra en la Figura 49, a la imagen obtenida anteriormente, teniendo como una región de interés a los objetos en movimiento en un primer plano. La razón de uso de esta librería es detectar los objetos que se mueven a partir de la diferencia entre la trama actual y un marco de referencia, a menudo llamado imagen de fondo.

Figura 49. ***Transformación de la imagen en nivel de grises a Background Subtraction***

Luego de tener solamente la imagen de interés al aplicar el Background, desde este punto a las imágenes se les aplica las trasformaciones morfológicas para poder filtrar el ruido, la imagen debe ser sometida a una erosión como se muestra en la Figura 50, esto quiere decir que se va a erosionar los límites del

objeto. Un píxel en la imagen sea este 1 ó 0, será considerado 1 si todos los píxeles bajo el kernel es 1, de lo contrario se erosiona (se hacen 0).

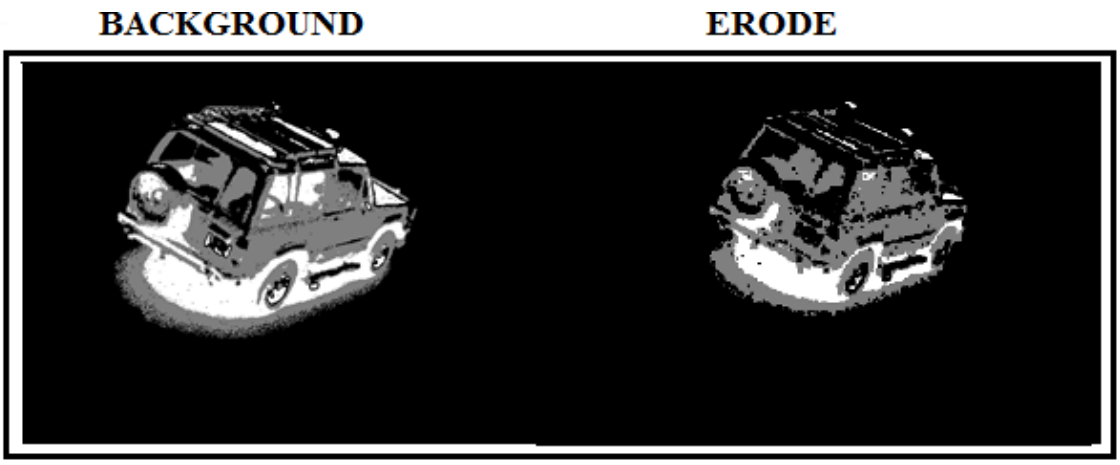

Figura 50. ***Imagen al aplicar Erode***

Para asegurarse que el ruido no afecte a la imagen, se debe pasarla por un filtro adicional que es open o apertura, este es sólo otro nombre que se la da al momento de hacer una erosión seguida de una dilatación a la imagen. Con esta transformación morfológica a la que es sometida la imagen, se puede eliminar los puntos blancos o negros no deseados que se puede encontrar en el contorno la imagen. En la Figura 51 claramente se puede apreciar que los puntos negros que aparecían con Erode ya se los eliminó.

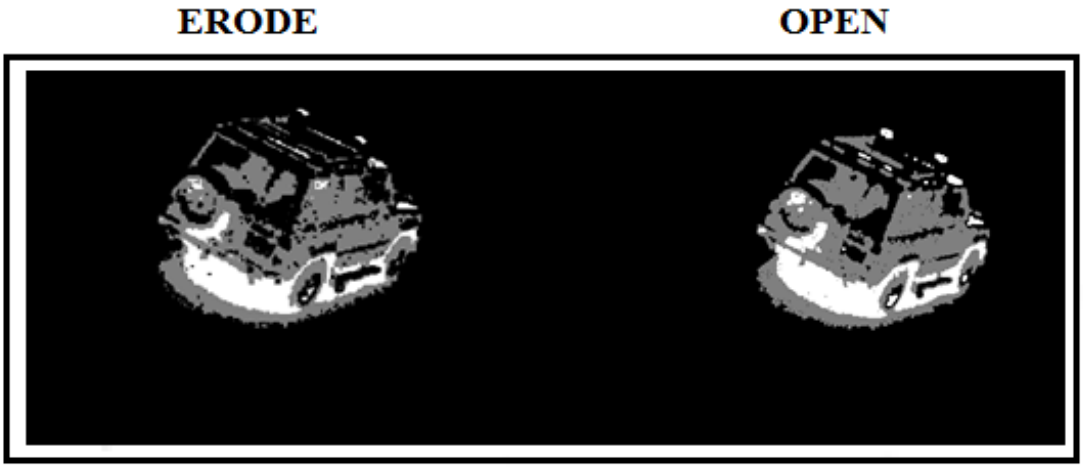

Figura 51. ***Resultado al utilizar open***

Después de pasar a la imagen por los diferentes filtros se tiene que umbralizarla con el objetivo de convertir una imagen en escala de grises a una

nueva con sólo dos niveles, de manera que los objetos queden separados del fondo como se ilustra en la Figura 52. Si el valor de pixel es mayor que el valor de umbral, se le asigna un valor (en este caso un color blanco), de lo contrario se le asigna otro valor (color negro).

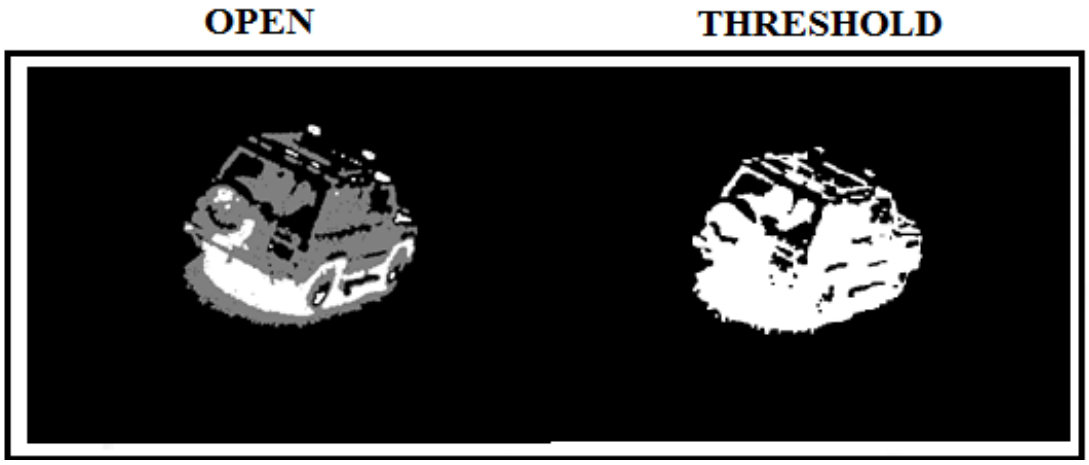

Figura 52. ***Imagen umbralizada***

Finalmente para que la imagen sea una sola masa y poder dibujar su contorno se debe hacerle una dilatación, esto con el fin de que aumente el área de los objetos o también para ayudar a unir partes separadas del objeto. A continuación en la Figura 53, se muestra claramente la imagen resultante después de aplicar una dilatación.

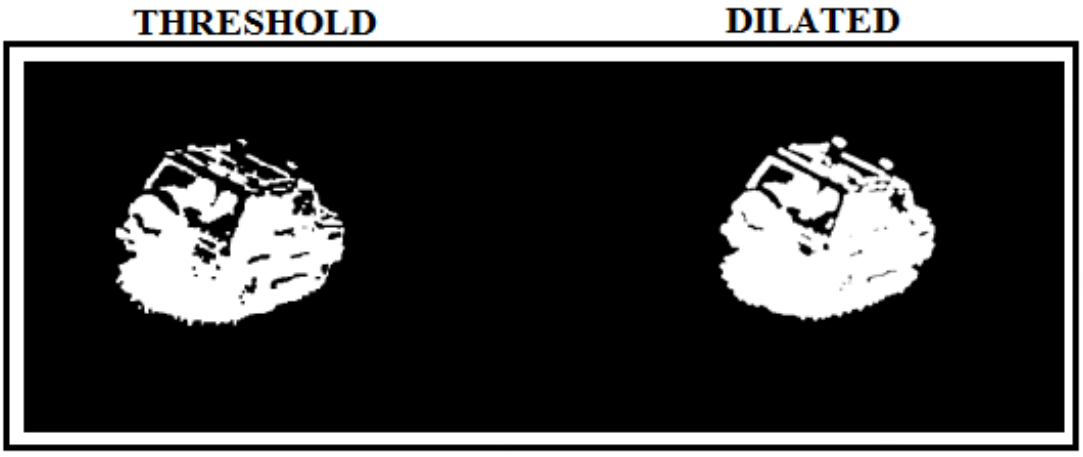

Figura 53. ***Imagen dilatada***

Todas las transformaciones anteriores a las que fue sometida la imagen ayudan a dibujar un solo contorno de la misma, si a la imagen no se le sometía a los procesos anteriores, se obtienen varios contornos dentro y fuera de la

imagen, por lo que no ayudan al momento de poder tomar las dimensiones del vehículo para así saber si es un vehículo pesado o un liviano. De esta manera en la Figura 54 se puede observar claramente el contorno dibujado del vehículo tomado para realizar las prácticas de filtrado y transformaciones morfológicas usadas para el desarrollo del proyecto.

Figura 54. ***Imagen del contorno del vehículo***

Con el área de los contornos de una imagen, se puede guiar para poder dibujar los rectángulos que ayudarán en el reconocimiento y seguimiento de los objetos en la imagen, de esta forma se puede ver tanto el ancho como el alto de un vehículo y así clasificarlos como uno pesado o uno liviano, dependiendo del área que tengan los mismos, la altura y el ancho de los vehículos varían de acuerdo al posicionamiento y a la altura que adopte la cámara. También se dibuja un círculo en el centro del rectángulo del mismo color, esto para tener una guía con la cual se puede contar los vehículos, además este círculo ayuda también a tener una referencia la cual indica por qué sector de la pantalla está pasando el vehículo y así calibrar la cámara según la región de interés que posea al momento de calibrarla en un sector diferente.

A continuación se muestra en la Figura 55, el contorno resultante de la imagen (A) y la imagen aplicada todos los procesos de filtrado (B).

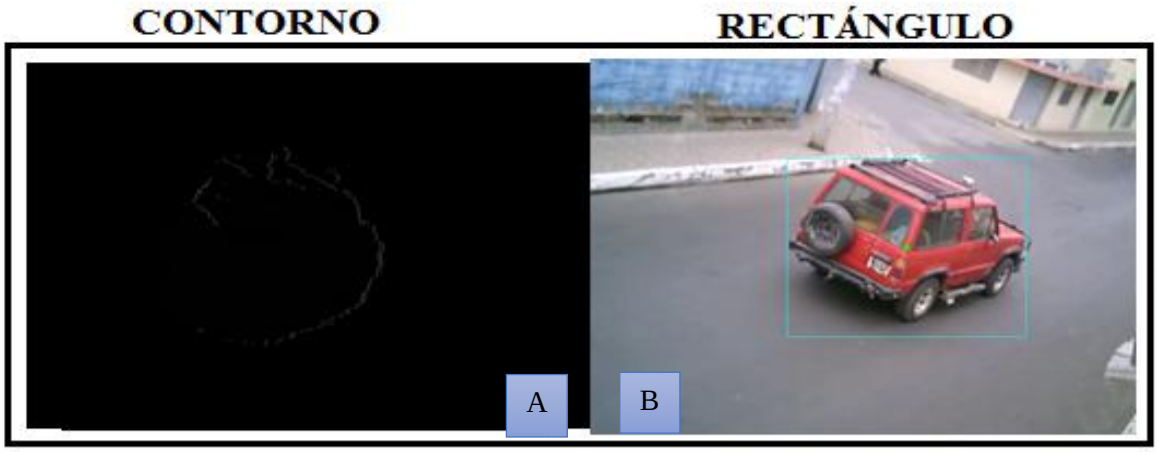

Figura 55. ***Contorno de la imagen(A), rectángulo dibujado en la imagen original (B).***

En la Figura 56 se tiene el diagrama de flujo del programa, el cual está dividido en secciones para un mejor estudio. La sección A carga las librerías de OpenCv necesarias para desarrollar el programa y lee la cámara entrando en un lazo While para mostrar los frames capturados por la cámara. En la sección B los frames capturados por la cámara deben pasar por un filtro Gaussiano, en esta sección a los frames también se los transforma a escala de grises para poder utilizar el método de sustracción de fondo (Background Subtraction), luego de este proceso se tiene que eliminar el ruido que es generado por diferentes factores que se verán más adelante, los que se elimina aplicando una transformación morfológica a las imágenes. La sección C está encargada en dibujar un triángulo en el vehículo deseado de acuerdo al área de contorno que este posea. Al final de todo el proceso anterior se debe verificar si es un vehículo liviano o un pesado, este proceso se lo realiza en la sección D, para efectuar este paso se debe verificar la altura del vehículo que se encuentra dentro del frame, de esta manera se detecta un vehículo pesado o un liviano, y por último se imprime el conteo total de los vehículos.

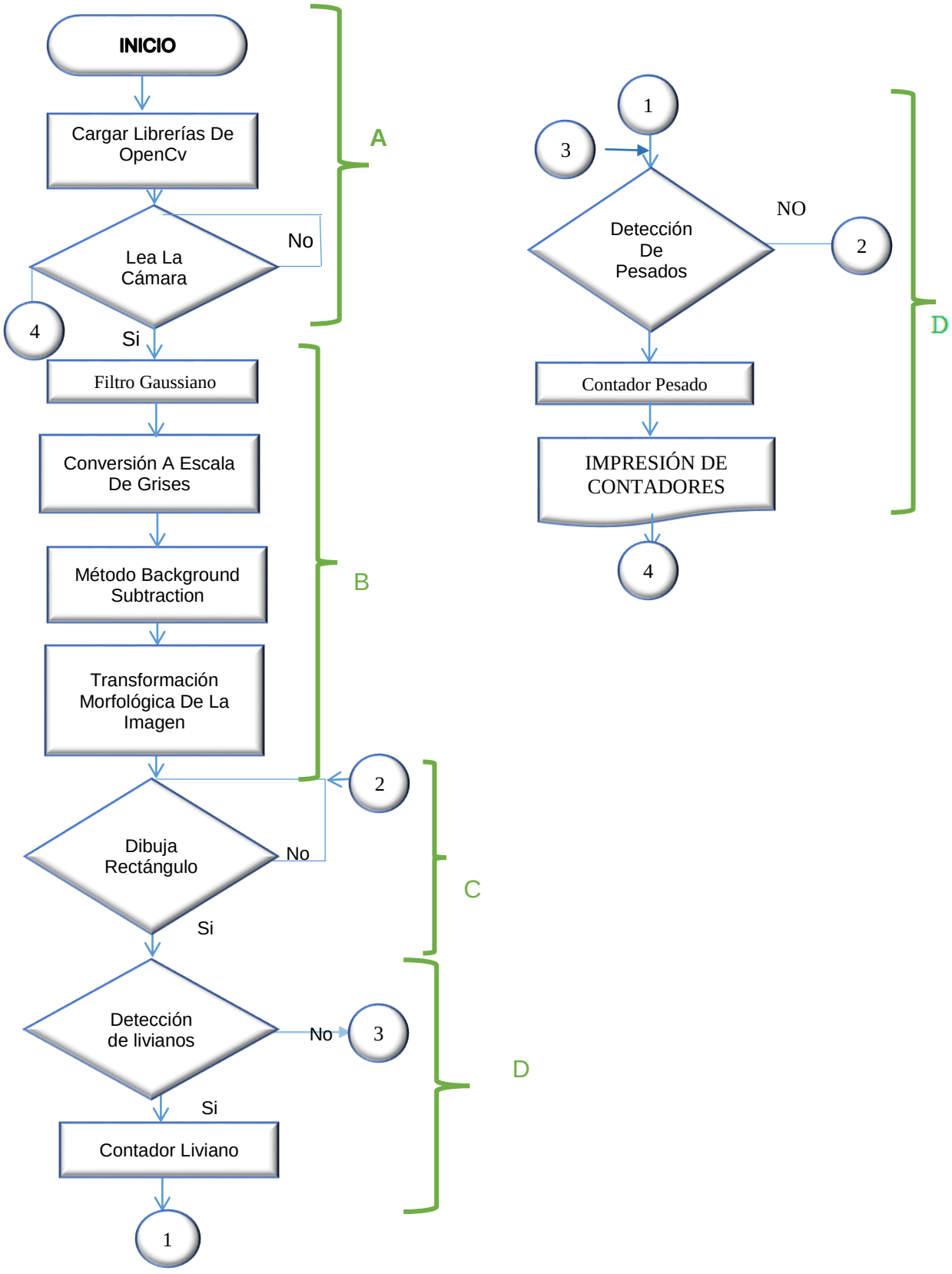

Figura 56. ***Diagrama de Flujo del Algoritmo de control***

3.3.1 Interfaz gráfica del usuario

Para una mayor facilidad al momento de calibrar la cámara y que el lugar en donde se desee ubicar a la tarjeta no represente algún inconveniente, la interfaz creada contiene lo básico para poder detectar y seleccionar una región de interés de acuerdo a la naturaleza del lugar seleccionado.

Básicamente lo que permite hacer es limitar las zonas superiores e inferiores del frame (sliders SUPERIOR e INFERIOR), esto con la finalidad de eliminar sombras que se crean en las veredas como cuando pasan las personas o el reflejo de la luz de una pared, así como también dejar fuera de la región de interés los vehículos que se pueden estacionar o entrar en un garaje. También permite limitar la región de detección de los vehículos tanto para vehículos pesados como livianos, (slider MEDIA), esto con la finalidad de que un vehículo pequeño quede dentro del frame, ya que la velocidad con la que suelen transitar causa problemas.

Al momento de cambiar de posición la cámara, sea esta de altura o de ángulo (sliders Y.LIVIANO, Y.PESADO, AREA), para el análisis de los vehículos no son las mismas, por lo que dispone de la opción de variar la altura del vehículo para poder detectarlo como un vehículo pesado o liviano. Todo esto se puede visualizar en la figura 57.

Figura 57. ***Interfaz gráfica del usuario***

3.4 Pruebas de funcionamiento de la aplicación realizada.

Para determinar la calidad de la operación del sistema, las pruebas se realizaron en la Tarjeta Raspberry Pi y un computador. De esta manera se pudo verificar la respuesta que tiene la tarjeta al realizar trabajos de procesamiento de imágenes. Se toman diferentes medidas en algunos sectores, teniendo en cuenta varios factores que afectaron las medidas, uno de las causas más importantes por las que no se puede identificar de una forma correcta a los vehículos es la calibración de la cámara. Las condiciones de adquisición de datos, el posicionamiento de la cámara web en la Tarjeta, y la PC son las mismas que se establecieron inicialmente.

Las pruebas realizadas se desarrollaron en tres sectores diferentes, en cada sector se tuvo una configuración diferente de la cámara, tanto en ángulo como en altura. El primer sector en el que se efectuaron las pruebas, las cámaras se las ubicaron una altura de 3.5 m sobre el pavimento y se las enfocó en una forma trasera-lateral a los vehículos, las calles de este sector es 9 de octubre y Pichincha ubicadas en el Cantón Saquisilí, se lo denominó como sector A, como se muestra en la Figura 58.

Figura 58. ***Sector A***

Para la siguiente prueba realizada se cambió de posición a las cámaras, se las ubicó a una altura de 5 m sobre el pavimento, este sector es una intersección en la cual los vehículos se los enfoca en forma frontal y lateral, a

este se lo denominó sector B. Las calles en donde está situado este sector son las Pampas y Saquisilí ubicadas en el barrio la Maldonado Toledo en la ciudad de Latacunga la cual se aprecia en la Figura 59.

Figura 59. ***Sector B***

Para el sector C, el cual se observa en la Figura 60, la configuración de las cámaras se las cambió el ángulo de enfoque y la altura la cual fue de 5.50 m sobre el pavimento, las cámaras enfocan a los vehículos de una forma trasera-lateral. La ubicación del sector B facilita para tener otro enfoque hacia los vehículos, por lo que para las pruebas descritas en el sector C se tomó la misma localización del sector B pero con un enfoque de las cámaras distinto al del anterior.

Figura 60. ***Sector C***

3.4.1 Pruebas realizadas en el sector A.

Para la primera prueba realizada en el sector A, el enfoque de la cámara fue hacia la parte trasera-lateral del vehículo, la misma se ubicó a 3.5 m sobre la vía.

Tabla 2.

Conteo de vehículos primer sector A, posición de cámara trasera-lateral al vehículo.

Horas del día	Conteo en una Pc		Conteo en tarjeta Raspberry pi 2		Conteo manual		Error % (pc)	Error % (tarjeta)
	Carros Livianos	Carros Pesados	Carros Livianos	Carros Pesados	Carros Livianos	Carros Pesados		
8am – 10am	15	0	14	1	37	1	60,5	60,5
10am– 12am	9	0	8	0	15	2	47,1	52,9
12 am – 2pm	22	1	20	0	43	4	51,1	57,4
2 pm – 4pm	24	1	17	2	29	1	16,7	36,7
4 pm – 6pm	12	1	13	1	31	4	62,9	60,0
Total	**82**	**3**	**72**	**4**	**155**	**12**	**49,1**	**54,5**

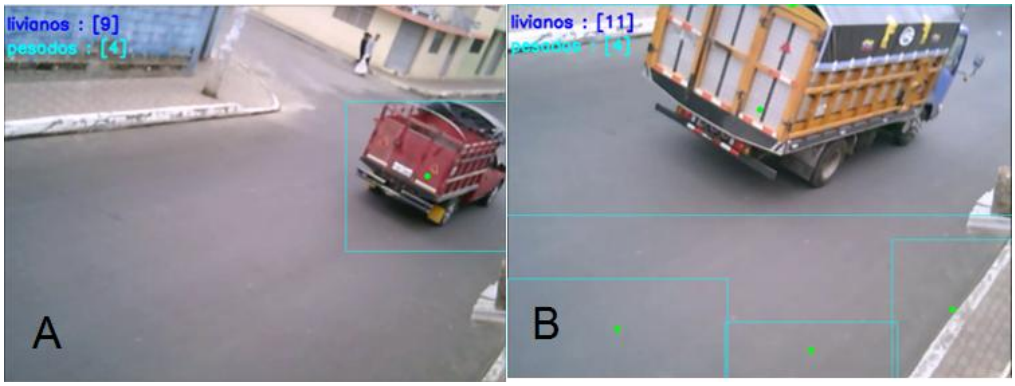

Figura 61. ***Sector A, tipo de posición de cámara trasera-lateral al vehículo***

En esta posición los errores para la toma de datos se debieron a varios motivos, pero entre los principales están el movimiento de la cámara (Figura 61: B) que hace que detecte otros objetos, la distancia entre la cámara y la vía es muy cercana por lo que los vehículos pesados ocupan casi toda la pantalla haciendo que ésta pierda el foco con los cambios de nivel, lo que dificulta el trabajo del algoritmo.

3.4.2 Pruebas realizadas en el sector B.

Para la siguiente prueba realizada se tiene el segundo sector en donde se realizaron las pruebas, posición de cámara Frontal al vehículo, la cámara se encuentra ubicada a 5 m sobre la vía.

Tabla 3.

Conteo de vehículos segundo sector, posición de cámara frontal al vehículo.

Horas del día	Conteo en una Pc		Conteo en tarjeta Raspberry pi 2		Conteo manual		Error % (pc)	Error % (tarjeta)
	Carros Livianos	Carros Pesados	Carros Livianos	Carros Pesados	Carros Livianos	Carros Pesados		
8 am – 10 am	9	1	5	7	10	5	33,3	20,0
10 am –12am	13	6	3	4	13	3	-18,8	56,3
12 am – 2 pm	11	7	7	2	12	2	-28,6	35,7
2 pm – 4 pm	12	2	3	5	11	1	-16,7	33,3
4 pm – 6pm	16	4	5	5	14	4	-11,1	44,4
total	**61**	**20**	**23**	**23**	**60**	**15**	**-8,0**	**38,7**

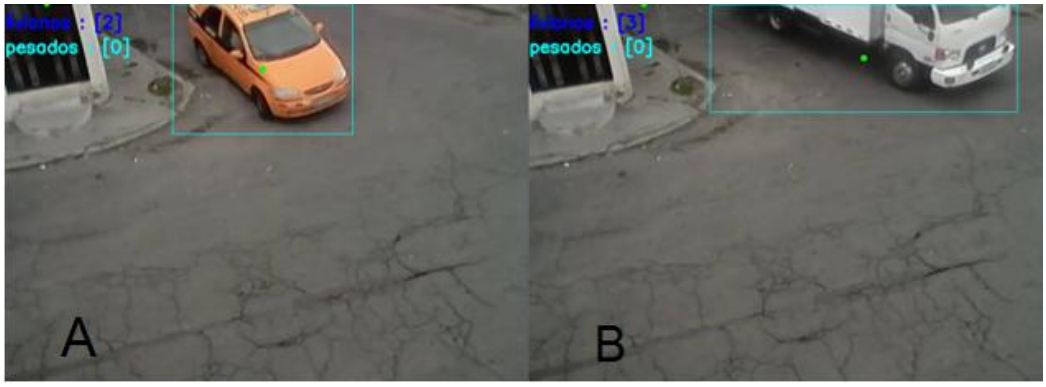

Figura 62. ***Sector B, tipo de posición de cámara Frontal al vehículo***

El sector en donde está ubicada la cámara es una intersección, cuando aparecen los vehículos pesados como se muestra en la Figura 62 B, se detectaban en algunas ocasiones como un liviano y cuando pasan en forma horizontal se detectan más de uno y el problema con los livianos es que la velocidad con la que circulan dificulta la detección.

3.4.3 Pruebas realizadas en el sector C.

Para la siguiente prueba se cambió de sector y de posición de la cámara, la cual se ubicó a 5.50 m sobre el pavimento y se la posicionó de una forma trasera-lateral al vehículo.

Tabla 4.

Conteo tercer sector, posición de cámara posterior al vehículo.

Horas del día	Conteo en una Pc		Conteo en tarjeta Raspberry pi 2		Conteo manual		Error % (pc)	Error % (tarjeta)
	Carros Livianos	Carros Pesados	Carros Livianos	Carros Pesados	Carros Livianos	Carros Pesados		
8 am – 10 am	52	2	50	3	55	2	5,3	7,0
10 am –12am	33	4	30	3	33	4	0,0	10,8
12 am – 2 pm	25	8	25	7	26	9	5,7	8,6
2 pm – 4 pm	33	3	25	2	31	2	-9,1	18,2
4 pm – 6pm	58	2	57	1	60	1	1,6	4,9
Total	**201**	**19**	**187**	**16**	**205**	**18**	**1,3**	**9,0**

Esta es la posición y la altura de la cámara más óptima para poder detectar y contar un vehículo (figura 63), de esta manera se puede detectar a un liviano como un pesado, los errores que se presenta son mínimos y que se deben a la velocidad de circulación de los vehículos livianos y la altura de los pesados los cuales dificultaban la detección en la anteriores posiciones.

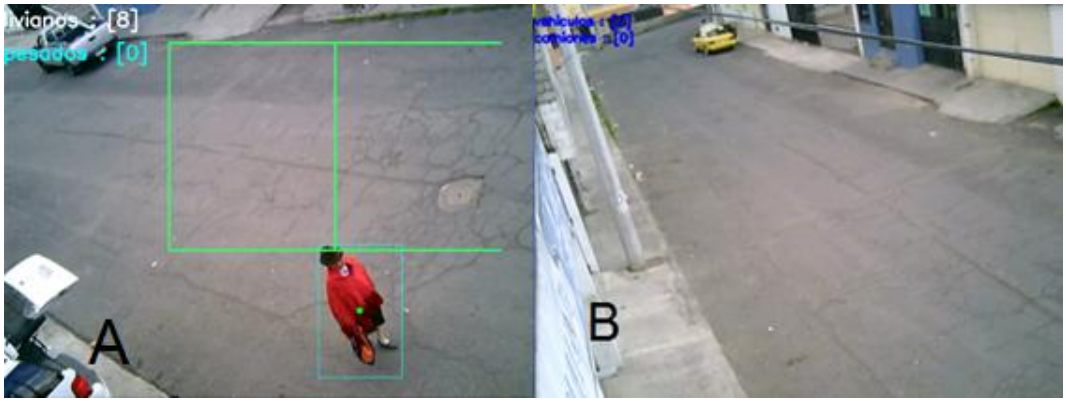

Figura 63. ***Sector C, tipo de posición de cámara trasera-lateral al vehículo***

La posición y altura aplicada en las pruebas del sector A no fueron las mejores como se mostró en los resultados de la Tabla 2, el error de conteo con la PC es de 49.1% y con la tarjeta de 54.5%, luego se cambió de posición y altura al sector B, no se obtuvo una buena respuesta por parte de la tarjeta(los resultados se muestran en la tabla 4), se obtuvo un error de conteo con la PC de -8% y con la Raspberry Pi de 38.7%.

Para la siguiente prueba en el sector C se obtuvo un error de conteo con la PC de 0.4% y con la Raspberry Pi de 3.6%, por lo que las siguientes pruebas se desarrollarán en esta posición, pero se debe tener en cuenta que aún en ésta posición existen factores que afectan el conteo.

3.4.4 Ajuste de las cámaras y el programa para las pruebas desarrolladas en el sector C

Al ejecutar en forma paralela las pruebas tanto en la tarjeta como en el computador con cámaras independientes, pero de las mismas características, ubicadas en el sector C como se muestra en la Figura 64, se presenta el análisis del proceso final.

Figura 64. ***Identificación de las cámaras independientes tanto para la Tarjeta como para la Pc***

Como se observa en la Tabla 3 la respuesta en la tarjeta respecto al de la Pc no es la esperada, esto ocurre porque el rendimiento del procesador del computador es más eficiente que el de la Tarjeta, debido a su mayor capacidad.

En el computador tampoco se obtienen los resultados deseados. Se presenta un resultado porcentual negativo porque el valor medido excede el valor verdadero que se tomó manualmente, por lo que se procede a realizar cambios en los programas del computador como de la tarjeta.

Para el computador se afina el programa buscando los posibles problemas que se dan para que trabaje en forma correcta.

Los problemas que se encontraron se describen a continuación.

- La región de interés para realizar la detección de los vehículos es muy corto. Se refiere a la distancia estimada para la detección vehículos. Debido al corto rango de detección, los carros que sobrepasan la velocidad de 50 km/h no se detectan y por ende no se realiza el conteo. Las regiones de interés tanto para la tarjeta como para la Pc se muestran en la figura 65.

Figura 65. ***Rango de detección de vehículos***

- El rango del contorno de la imagen es demasiado extenso. Se refiere a que los vehículos que circulan muy cerca el uno del otro se los dibuja como un solo contorno, representa un gran problema porque al realizar el conteo, no detecta a otro vehículo si pasan en forma continua. Por ésta situación es que las regiones de interés se encuentran en la parte derecha de la imagen, cuando el vehículo aparece en el frame, este método esta aplicado en la Tarjeta y en la Pc. La construcción de un solo contorno en la unión de dos vehículos se aprecia en la Figura 66.

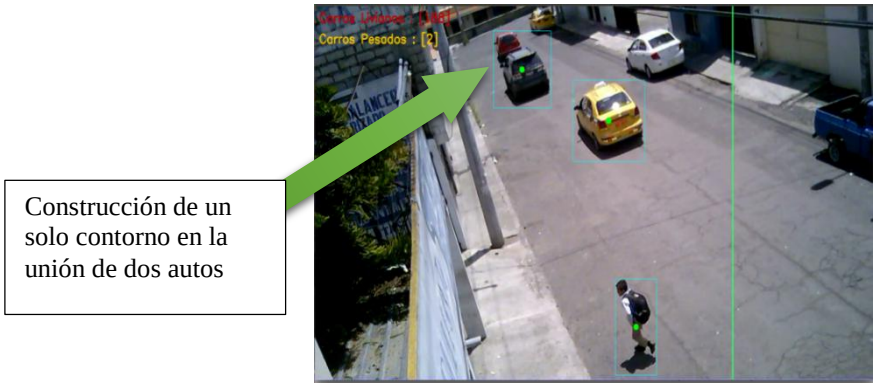

Figura 66. ***Construcción de un solo contorno en vehículos que circulan muy cerca***

Discriminación de los bordes del área de trabajo. Se refiere a:

- ✓ Las aceras de la calle se las discrimina, debido a que cuando caminan un grupo de personas juntas el sistema las reconoce como un auto pequeño por la dimensión del contorno detectado como se muestra en la Figura 67.

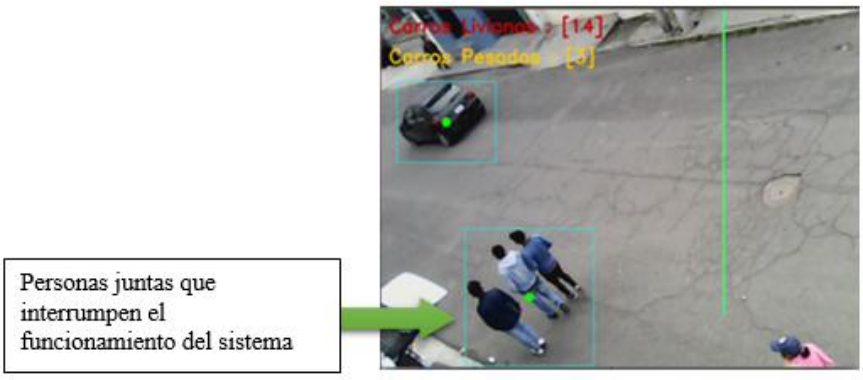

Figura 67. ***Dimensión de grupo de personas que interfieren en la medición vehicular***

- ✓ Se tiene el problema de entrada y salida de vehículos a estacionamientos cercanos al área de detección, esto hace que el sistema cuente los autos o carros pesados aceleradamente como se muestra en la Figura 68. Con la delimitación del área de trabajo y la corta aparición del contorno del vehículo, una escena

como el de las figuras 67 y 68 no causará problemas en el funcionamiento del sistema.

Figura 68. ***Entrada y salida de vehículos que interfieren en la medición del flujo vehicular***

- ✓ Parqueo de autos cerca de la región de detección y medición de vehículos. Esta situación determina que el sistema trabaje de manera errónea, porque se dispara el conteo de autos y no detecta a ningún otro vehículo, como se muestra en la Figura 69.

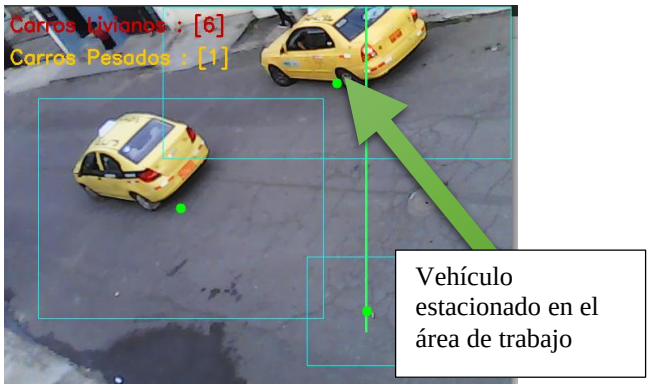

Figura 6968. ***Parqueo de vehículos cerca de la región de trabajo***

- Retardos en el procesamiento de imagen en la tarjeta

Este es el principal problema que presenta la Tarjeta Raspberry Pi, se lo soluciona tomando regiones de interés de acuerdo al medio en donde se desee detectar para que de esta manera se pueda determinar la presencia tanto los vehículos pesados como los livianos. En la Figura 70 (A) se aprecia claramente el retardo que posee la tarjeta al momento de presentar un frame, debido a las limitaciones que esta posee. En la Figura 70 (B) se tiene la pantalla dividida en

rangos de trabajo para que solo se tome en cuenta al vehículo que está transitando y no a los que se pueden estacionar o pueden estar saliendo de algún garaje, en la región de interés se presenta una línea en forma vertical dividiendo la pantalla, de esta manera se puede despreciar a un vehículo liviano de un pesado.

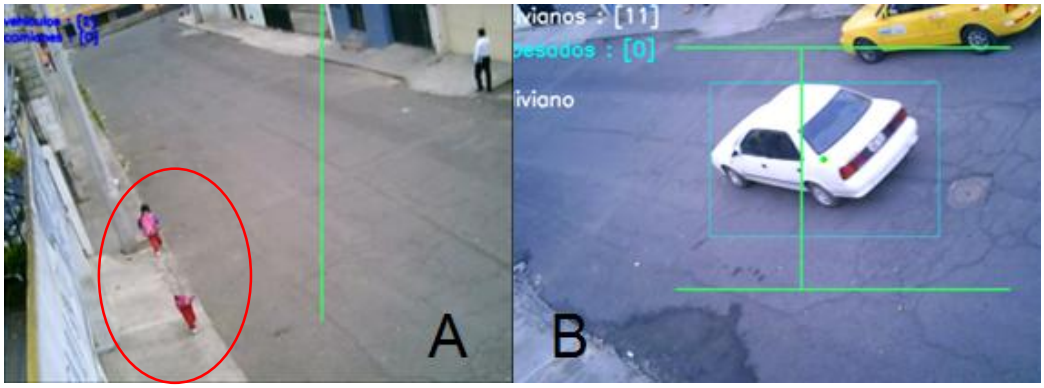

Figura 7069. ***(A)Retardos de la imagen en la tarjeta, (B) rangos de trabajo para un óptimo desempeño***

- El viento y cambios de luz

Otro factor por el que se tiene mediciones erróneas es cuando existen fuertes vientos y mueven las cámaras haciendo que aparezcan sombras al momento en el que se debe identificar un vehículo, algo parecido se da con los cambios de luz por lo que se tienen diferentes errores de medición, este es el motivo por el cual se tomaron mediciones en diferentes horas del día dando como resultado un mayor número de errores en las mediciones a partir de las 4 de la tarde, ya que a partir de esta hora se tiene una mayor presencia del sol, el cual se presenta en forma angular, afectando directamente a las medidas que se toman para poder identificar a los vehículos .

En la Figura 71 (A) se presenta la imagen original, mientras que en la imagen (B) se presentan sombras que dificultan las medidas debido a los movimientos de la cámara y cambios de luz al momento de aplicar Background Subtraction.

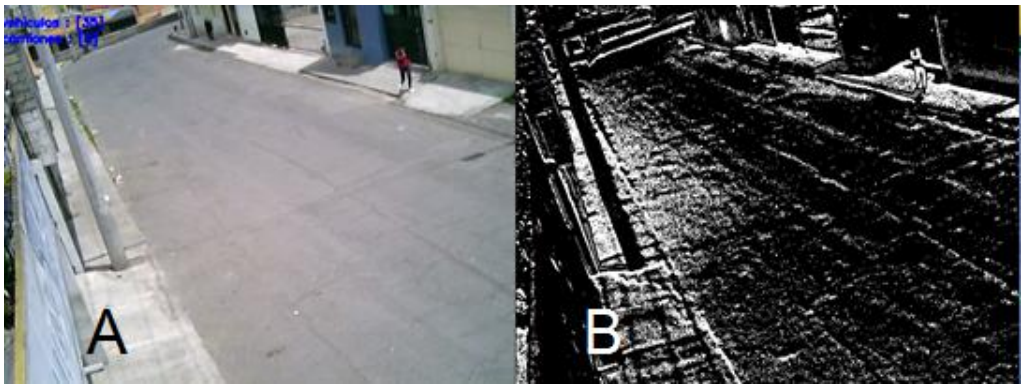

Figura71. ***Frame de origen (A), luego de pasar por Background Subtraction (B)***

Para la detección de los problemas antes mencionados se realizaron varias pruebas hasta que el programa trabaje de la manera que se desea con la Tarjeta y el computador, el mismo que fue una guía para la correcta configuración de datos en el programa de la tarjeta Raspberry Pi.

Al realizar las debidas correcciones en cada problema se obtuvo un porcentaje de error relativo de 0.9 % en la implementación del algoritmo en el computador, mientras que en la tarjeta presenta un porcentaje de error relativo de 6.1 %. Como se muestra en la Tabla 5.

Tabla 5.

Análisis del sistema con correcciones de mejora.

	Conteo en una Pc		Conteo en tarjeta Raspberry Pi 2		Conteo manual			
Horas del día							**Error % (pc)**	**Error % (tarjeta)**
	Carros Livianos	**Carros Pesados**	**Carros Livianos**	**Carros Pesados**	**Carros Livianos**	**Carros Pesados**		
8 am – 10 am	10	1	10	1	10	1	0,0	0,0
10 am –12am	25	2	23	1	25	2	0,0	11,1
12 am – 2 pm	30	4	31	3	30	4	0,0	0,0
2 pm – 4 pm	14	1	13	1	15	1	6,3	12,5
4 pm – 6pm	22	4	21	3	22	4	0,0	7,7
total	**101**	**12**	**98**	**9**	**102**	**12**	**0,9**	**6,1**

Como se aprecia en la figura 72 el principal factor que afecta al momento de detectar y contar un vehículo es el retardo de procesamiento de imagen que tiene la tarjeta, por este motivo es el que la región de interés varían en la pc y en la tarjeta, además en la tarjeta a la región de interés se la limita en la parte superior e inferior, esto con el objetivo de despreciar zonas como son las de veredas y no afecten al programa al momento de contar. Con el estudio comparativo que se realizó con la PC y la Raspberry Pi se puede apreciar que el conteo final se ve afectado significativamente por dicho problema. Otro de los inconvenientes que se presentó es el cambio de luz que se generan en las diferentes horas del día, concluyendo que la mejor hora para que se desarrolle con normalidad el conteo es entre las 8 am y 4 pm. El viento es otro de los factores por el cual el conteo se ve afectado, esto produce movimientos en la cámara y se detectan objetos que no se desean los demás factores que afectan en ocasiones se presentan de una manera no muy frecuente, por ejemplo un grupo de personas que caminan junto al vehículo y pasan dentro de la zona de interés, afectando principalmente a la determinación de las dimensiones cuando se detecta el vehículo. También afectan al conteo los vehículos que entran o salen de un garaje.

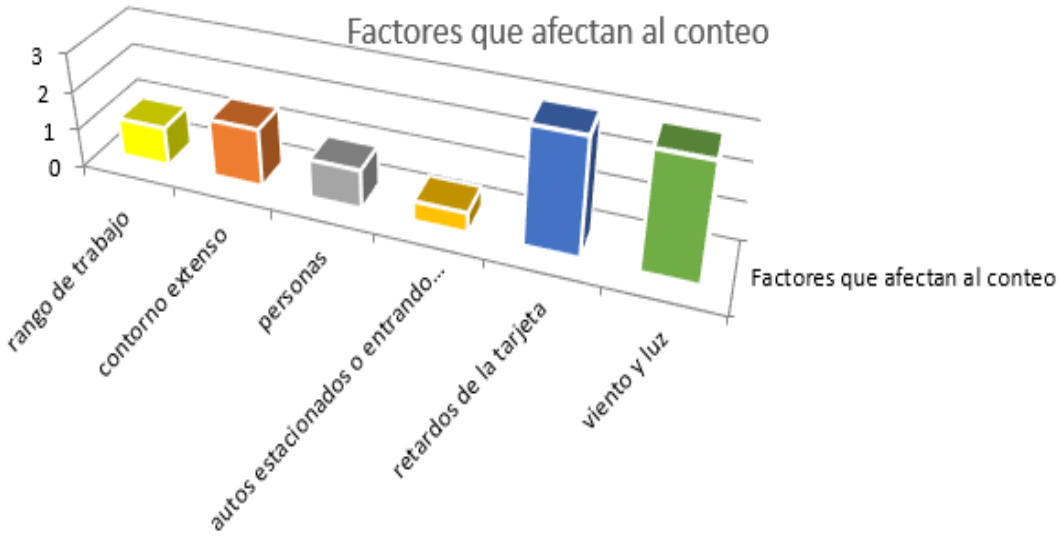

Figura 7270. ***Factores que afectan al conteo***

3.5 Análisis de los beneficios que se obtienen al usar la aplicación.

En una ciudad grande es algo normal el tráfico que se genera a causa de los vehículos que transitan por ésta, con el paso del tiempo se ha buscado diferentes mecanismos para poder controlar el tráfico. La obtención de los valores de los parámetros del tráfico muchas veces se los realiza de una forma manual, esto quiere decir que se ubica a personas en una determinada vía para que cuenten los vehículos que pasan por la misma, mediante la obtención de estos datos se puede hacer diversos estudios y caracterizaciones de las vías en las cuales se puede ver el comportamiento a futuro, además permite la elaboración de estimaciones y predicciones de obras de una manera confiable.

La propuesta es implementar en la tarjeta Raspberry Pi para que se pueda realizar el conteo de manera automática y no invasiva, para poder reducir costos y también disminuir el error que se tiene con el conteo manual, este error es producido por diversos factores como pueden ser: la fatiga al momento de contar, también las posibles distracciones que puede tener la persona, además de esto la persona debe poseer una desarrollada capacidad de atención y concentración para poder estar en un mismo sitio durante varias horas del día.

La persona que esté a cargo del conteo debe estar en la intemperie, lo que quiere decir que debe soportar el sol, el polvo y la lluvia si es necesario para poder realizar el trabajo de contar los vehículos circulantes, lo que la tarjeta solo necesita un lugar en donde situarse para poder realizar su trabajo.

Como se puede ver en la figura 73 los errores que presentan las mediciones realizadas con la tarjeta son mínimos, por lo que comparando el precio que representa al pagar a una persona para que realice el trabajo manualmente, y el precio de implementación de la tarjeta para que se realice el mismo trabajo de una manera automática, se tiene un mejor beneficio al ocupar la tarjeta, ya que la tarjeta se enfoca solo a realizar su trabajo, y no tiene las

posibles distracciones, cansancio y algún accidente que puede sufrir una persona al momento de realizar dicha tarea, además se debe tomar en cuenta los demás beneficios que debe tener una persona al ser contratada para realizar dicho trabajo.

Al momento de emplear ésta aplicación, también nos aseguramos que los datos no se pierdan o sufran algún tipo de manipulación o alteración, nos aseguramos que los datos sean los correctos para emplearlos en futuros estudios, además a la tarjeta y la cámara se los puede ubicar en cualquier sitio de la ciudad sin correr el riesgo de tener algún tipo de accidente.

El sistema que posee la tarjeta para el conteo es adaptativo, lo que quiere decir que no tiene una imagen de referencia de fondo. Para ser ubicada en otro sitio de la ciudad tiene que ser calibrada tanto en la posición como en el ángulo de enfoque para que pueda trabajar de una buena manera.

Se debe tener en cuenta que las horas más idóneas para poner en marcha al proyecto son desde las 8 am hasta las 4 pm, ya que en ese intervalo de tiempo no afecta de una gran manera el cambio de luz que se produce durante el día, luego de las 4 pm las sombras que se producen por la posición misma del sol, hacen que no se puedan conseguir una buena lectura de los datos.

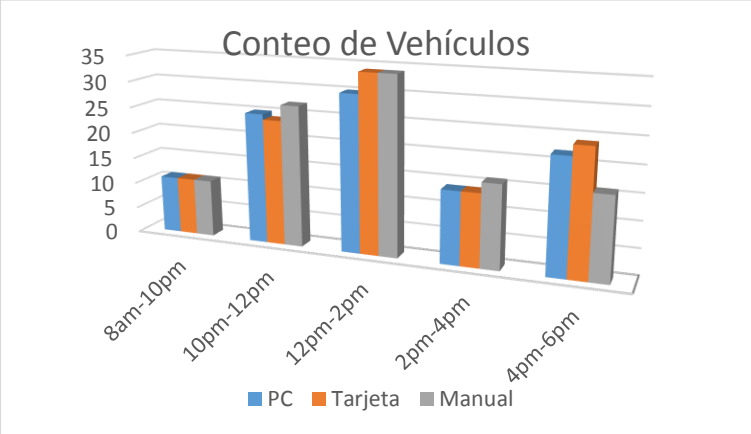

Figura73. ***Conteo de vehículos en diferentes horas del día***

3.6 Diagrama de bloques del sistema

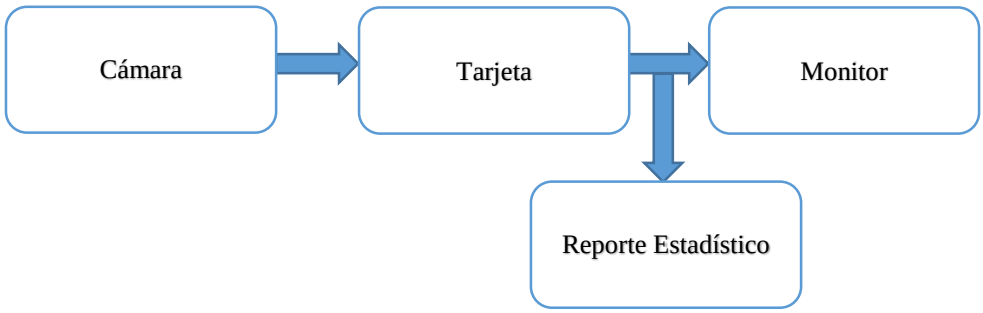

Figura 74. ***Diagrama de bloques del sistema***

En la Figura 74 se muestra el diagrama de bloques del sistema. La cámara captura las imágenes que serán procesadas en el programa desarrollado en la tarjeta Raspberry Pi para la detección y conteo de vehículos, para luego mostrar los resultados estadísticos en Bloc de Notas y del tratamiento de imágenes en un monitor el cual se conecta mediante el puerto HDMI de la tarjeta.

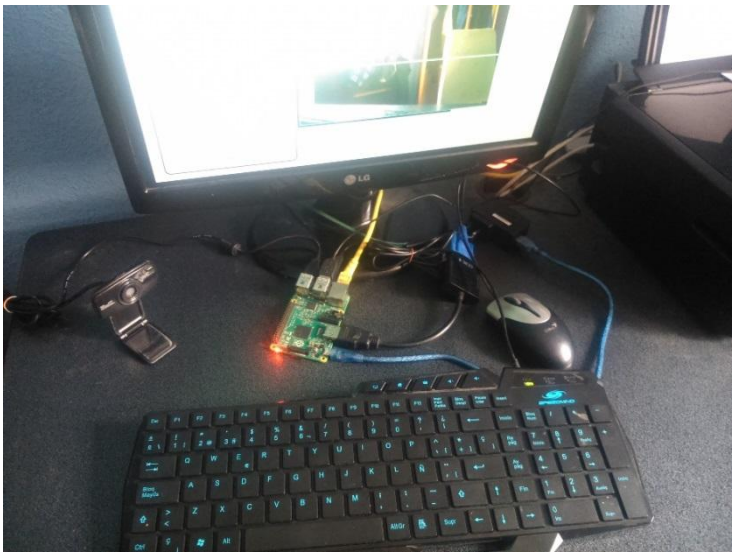

Figura 75. ***Sistema implementado***

En la Figura 75 se muestra el resultado del sistema implementado, en el cual se muestra elementos como mouse, teclado, cámara web que se conectan mediante el puerto USB a la tarjeta.

Se puede ingresar a la tarjeta mediante un escritorio remoto que se lo puede crear en una pc y utilizando una red local, para lo cual la tarjeta necesita ser conectada al modem de la red.

3.7 Alcances y limitaciones

3.7.1 Alcances

- El proyecto implementado cuenta vehículos en una determinada avenida de la ciudad.
- El algoritmo que se implementó en la elaboración del proyecto está entrenado para discriminar carros livianos y carros pesados.
- Se elimina todos los ruidos que afectan en la detección de los objetos en movimiento para posteriormente su respectiva clasificación.
- Se implementa un algoritmo de control compacto, para el desarrollo del sistema de medición de flujo vehicular implementado en la tarjeta Raspberry Pi.
- Se discrimina a todo objeto en movimiento que no sea un auto pequeño o un vehículo de tipo pesado.

3.7.2 Limitaciones

- El sistema desarrollado para contar carros funciona correctamente con vehículos en movimiento.
- La tarjeta Raspberry Pi tiene un retardo de 2 segundos en mostrar un Frame.
- De cualquier manera, es prácticamente imposible no tener limitaciones y generar un modelo perfecto ya que la capacidad de procesamiento de imágenes de la tarjeta Raspberry Pi es limitada, y como también las variables y factores, tanto conscientes como inconscientes, influyen en

las decisiones de los usuarios y subsecuentemente en la detección y conteo. Por estas razones es que en definitiva, las condiciones están en constante cambio, y por tanto el modelo nunca podrá llegar a contar a la perfección.

CAPÍTULO IV

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El funcionamiento correcto del algoritmo de control se debe a la investigación preliminar que se realizó sobre la librería OPENCV, para de esta manera poder identificar y discriminar vehículos pequeños de vehículos pesados.
- Las características (procesador, memoria interna y externa) funcionales de la tarjeta Raspberry Pi permite almacenar las librerías requeridas de OpenCv y poder trabajar en tiempo real con procesamiento de imágenes.
- Los recursos que se consumen al momento de trabajar con procesamiento de imágenes en tiempo real con la tarjeta Raspberry Pi, son considerables por esta razón, el algoritmo que se desarrolló para el proyecto posee características de eficiencia y robustez.
- Los limitados recursos de la tarjeta Raspberry Pi no permiten desarrollar una interfaz gráfica en el software Qt Creator, por esta razón es necesario trabajar en el lenguaje de programación nativo que posee la misma (Python).
- Se puede implementar una red de tarjetas que trabajen de forma simultánea, pero cada tarjeta ejecuta una tarea diferente, en procesamiento de imágenes las tarjetas Raspberry Pi no pueden ejecutar la misma aplicación en forma paralela, porque deben compartir la misma memoria RAM y el mismo bus de datos y direcciones para poder desarrollar una misma aplicación. El no procederse de esta manera significará tiempos de procesamiento superiores a los que se obtendrían si se ejecuta en forma serie, debido a los tiempos requeridos en el proceso requerido para la comunicación entre tarjetas. Esta razón

determina que el proyecto se desarrolló con una tarjeta Raspberry Pi para la detección de vehículos.

- La ubicación y la estabilidad de la cámara es primordial al momento de discriminar un vehículo liviano de un pesado, luego de las pruebas realizadas con las diferentes ubicaciones de la cámara, se tuvo como resultado que la altura mínima adecuada con que se puede detectar a los dos tipos de vehículos que es de 4.5 metros, porque a esta altura se puede manejar los contornos de los mismos que se encuentran dentro del foco de la cámara.
- Para tener una conexión remota con la tarjeta Raspberry Pi, se implementó un servidor FTP para descargar el reporte que se desarrolla al finalizar la ejecución del programa, ya que para poder desarrollar un servidor Web en la misma es necesario tener una dirección IP estática, ya que una IP pública cambia automáticamente su dirección, por lo que se tiene que pagar por este servicio, y no resulta conveniente cuando se desee cambiar de localización a la tarjeta.
- Un servidor WEB puede ser reemplazado por un servidor FTP con las ventajas de que no se necesita internet, IP estática, entre las más importantes que se consideran para el desarrollo de este proyecto.
- El proyecto se ejecuta de manera requerida porque se realizó algunas pruebas de funcionamiento, las que ayudaron a corregir todos los errores que se presentaron durante la ejecución del mismo.
- Los resultados del presente proyecto permiten desarrollar a futuro un sistema de administración de la circulación vehicular, porque presenta un análisis estadístico con fecha, hora y número de vehículos livianos y pesados que transcurren, así también se podría usar para el estudio y desarrollo de la calzada en alguna avenida o calle principal en donde sea ubicado el trabajo realizado.
- El sistema desarrollado en el proyecto se puede adaptar a cualquier entorno, sin importar la imagen que se presente de fondo, siempre y cuando se tome en cuenta las consideraciones necesarias para que se pueda detectar a los vehículos de mejor manera, la consideración más

importante es que la región de interés necesariamente debe quedar dentro del foco de la cámara.

- Las horas más idóneas para poner en marcha al proyecto son desde las 8 am hasta las 4 pm, ya que en ese intervalo de tiempo no afecta de una gran manera el cambio de luz que se produce durante el día, luego de las 4 pm las sombras que se producen por la posición misma del sol, hacen que no se puedan conseguir una buena lectura de los datos.
- El viento es un factor por el cual el conteo de los vehículos se ve afectado, porque produce movimientos en la cámara haciendo que se cambie la referencia de la imagen y el sistema considera que existe la presencia de objetos.
- La altura apropiada para el desarrollo de este proyecto es de 5.50 metros, porque a dicha altura se puede diferenciar el tamaño de un carro pequeño y un carro liviano.
- Una cámara exclusiva para la Raspberry Pi es más costosa que una cámara web normal, al momento de ponerlas en funcionamiento tienen la misma respuesta de funcionamiento, pero la diferencia es que, la cámara de la Raspberry Pi se conecta a la tarjeta mediante un conector de bus CSI (Camera Serial Interface), dedicado para la conexión de estas cámaras y la cámara web convencional se conecta mediante un cable de datos normal al puerto USB de la tarjeta. Comparando el precio y las prestaciones que brindan cada una, se optó por utilizar una cámara WEB ya que el cable de datos es más largo que el cable con el que viene la cámara de la Raspberry, además que tiene mayores protecciones para poder utilizarla a la intemperie.
- El algoritmo de control implementado en la elaboración del proyecto está entrenado para contar vehículos en movimiento.
- El Rango de detección para el conteo de vehículos depende de la capacidad del procesamiento del dispositivo en el que se implemente el algoritmo de control.

- Los colores de los vehículos no presentan ningún tipo de inconveniente, lo único que afecta en un vehículo que no sea reflector de luz solar, es que tenga exactamente el mismo color de la calzada.

- El brillo en la calzada y en los vehículos afecta la correcta detección de los vehículos, peor aún si se produce un reflejo directo del sol de una ventana de un vehículo o de un edificio.

4.2 Recomendaciones

- Se debe evitar cualquier tipo de obstáculo que produzcan variaciones en la imagen (sombras de árboles, edificios y luces de tráfico). Esto puede causar falsas estadísticas de detección o falsas alarmas.
- Para un mejor desempeño de la tarjeta siempre asegurarse que la posición de la cámara se encuentre enfocando el carril o el área de detección.
- Evitar que la posición de la cámara esté demasiado alta o baja, la posición adecuada es de 4.50 a 5.50 metros dependiendo los distintos ambientes de aplicación.
- Para identificar y contar vehículos en la noche es necesario tener una iluminación artificial adecuada o una cámara infrarroja.
- Se debe utilizar una cámara web que ayude con el procesamiento de imágenes en la tarjeta, no es recomendable utilizar de muy alta resolución ni tampoco una de resolución menor a 5 mp, debido a que las imágenes capturas por cámaras de una baja resolución no brindan una buena imagen y en ocasiones se ven afectadas por la luz que reflejan los vehículos, lo recomendable es utilizar una cámara de hasta 10 mp.
- Al momento de mostrar los frames capturados por la cámara en la pantalla es recomendable que el tamaño de los mismos sean de 640x480 pixeles, ya que a esta resolución se puede capturar de 40 a 60 frames por segundo.

- El procesador de la tarjeta Raspberry Pi puede trabajar al límite de su capacidad, es decir, trabajar a 1 GHz de velocidad, este proceso se llama overclock, pero la vida útil de la tarjeta se va agotando con mucha más facilidad, lo que es recomendable que trabaje solo a 900 MHz de su capacidad. La combinación al momento de hacer un overclock a la Raspberry Pi con tarjetas SD rápidas de tipo clase 6 ó 10 provocan fallos y terminan haciendo inestable al sistema operativo de la Raspberry Pi.
- La capacidad de la micro SD va a depender de lo que se necesite instalar, almacenar o para lo que vaya a ser utilizada la tarjeta Raspberry Pi, teniendo en cuenta que la micro SD debe ser mínimo de clase 6 para tener una mejor respuesta de procesamiento con la Raspberry.
- Para aprovechar al máximo el rendimiento de la micro SD, se debe formatear de tal manera que pueda ser Booteable, lo que quiere decir que la Raspberry Pi pueda arrancar sin inconvenientes con el sistema operativo previamente instalado, para hacer Booteable a una micro SD existen programas dedicados a ello, por lo que quiere decir que no basta con formatearla como comúnmente se lo hace, se lo debe realizar con un programa específico como por ejemplo el SDK.
- Existen sistemas operativos que ocupan solamente una parte del tamaño total de la micro SD, ya que el sistema operativo se instala en una partición con el espacio necesario, este espacio varía en función del sistema operativo como es el caso de Raspbian Wheezy, lo que produce un mayor espacio de la micro SD sin utilizar. Para poder utilizar este espacio "vacío" se debe expandir la partición para ocupar todo el espacio de la micro SD, para lo cual la forma más recomendable de hacerlo es mediante el programa de configuración de Raspberry Pi, pero se debe tener en cuenta que este proceso se debe realizar antes de instalar algún programa o guardar algún tipo de información, ya que se puede perder la información guardada.
- Al momento que se desee ingresar a la tarjeta mediante un escritorio remoto es recomendable que la tarjeta se configure con una IP estática,

esto con el fin de poder comunicarse con la tarjeta de una manera rápida.

- Los vehículos deben circular a una velocidad mayor a los 20 km/h, porque el algoritmo está entrenado con un rango de detección grande, para contar sin problema a los vehículos livianos y pesados.
- Para tener un mejor rendimiento al momento de trabajar en la Raspberry Pi, la principal característica con la que debe contar la micro SD en la cual se instala el sistema operativo, es que debe ser de clase 6 o de clase 10, la clase de una micro SD indica la velocidad máxima a la que graba, es recomendable utilizar estas ya que tienen una mejor respuesta al momento de transferir información.

REFERENCIAS BIBLIOGRÁFICAS

Agam, G. (15 de Febrero de 2016). Introduction to program

ming with OpenCV. Chicago, Estados Unidos.

Armequiza, A. (2010). Análisis de Flujo Vehicular, Análisis de Congestión, Semaforización y Estacionamientos. Guadalajara, México.

Baker, K. (2002). Efficient image gradient based vehicle localization.

Baque, J. (9 de Enero de 2016). Desarrollo de un algoritmo para la adaptación inteligente de la velocidad de un vehículo automotor mediante el uso de visión artificial para la detección del trazado de la vía y la proximidad entre vehículos. Cuenca, Ecuador.

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up Robust Features (SURF). *Computer Vision and Image Understanding (CVIU)*.

Bellesteros, G. (19 de Julio de 2015). Verificación de vehículos mediante técnicas. Madrid, España.

Berzal, I. (20 de Enero de 2016). Desarrollo de algoritmos de procesamiento de imágenes con VTK. Madrid, España.

Bradski, G. (s.f.). *Learnin OpenCV: compuyer vision wih the .*

Bradsky, G. (2010). Learnin Opencv.

Brown, C. (2000). The Vision Image-Understanding System.

Cobo, A. (23 de Febrero de 2016). Protocolo de Transferencia de Archivos FTP. Cordova, Argentina.

Escalante, B. (23 de Agosto de 2016). Procesamiento Digital de imagenes. Verona, Italia.

G.R. Bradski, A. K. (2008). *“Learning OpenCV: Computer Vision with the OpenCV ”.* New York: O Reilly & Associates.

Garcia, E. (2008). Detección y clasificación de objetos dentro de una aula. México.

Gupte, S. (2002). Detection and classification of vehicles.

Mendieta, V. (14 de noviembre de 2015). DETECCIÓN Y RECONOCIMIENTO DE SEMÁFOROS. Madrid, España.

Otero, C. (2004). Sistema De Aprendizaje Y Reconocimiento De Objetos 3D A Partir De Imágenes Planas. Málaga, España.

Rossum, G. (2009). Tutorial de Python.

Sierra, C. (19 de Enero de 2016). Evaluación del flujo vehicular en la intersección de la carrera 70 con la calle 9 por la construcción de una vía subterránea en el aeropuerto Enrique Olaya Herrera. Antoquia, Colombia.

Woel, F. (Mayo de 2005). A monocular collision warning system.

LINKCOGRAFÍA

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). *SURF: Speeded Up Robust Features.* Recuperado el 10 de 04 de 2015, de http://www.vision.ee.ethz.ch/~surf/eccv06.pdf

Acurio Méndez, E. M. (15 de Mayo de 2001). *Biblioeca Digita EPN*. Obtenido de http://bibdigital.epn.edu.ec/handle/15000/4119?mode=full

Baque, J. (15 de Septiembre de 2015). *Repositorio UPS*. Obtenido de http://dspace.ups.edu.ec/bitstream/123456789/7126/1/UPS-CT003932.pdf

Cofre, A. (17 de Agosto de 2015). *Repositorio UNIVERSIDAD TÉCNICA FEDERICO SANTA.* Obtenido de http://www.telematica.utfsm.cl/telematica/site/artic/20121008/asocfile/20121008171131/cofrealvaro.pdf

Cruz, C. L. (2013). *Seguimiento automático de objetos en sistemas con múltiples cámaras.* Obtenido de http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20130527CarlosLeguaCruz.pdf

Domingo, M. (12 de Febrero de 2016). Visión por compuador. Santiago de Chile.

Doutel, F. (13 de Agosto de 2015). *XATAKA SMATR HOME*. Obtenido de http://www.xatakahome.com/trucos-y-bricolaje-smart/probamos-la-nueva-raspberry-pi-2-a-fondo

Gonzales, A. (13 de Diciembre de 2015). Obtenido de https://publicaciones.unirioja.es/catalogo/online/VisionArtificial.pdf

Huenupi, M. (13 de Enero de 2016). Teoria de flujos vehiculares. Santiago de Chile.

Méndez, A. (15 de Agosto de 2015). *Biblioteca digial EPN*. Obtenido de http://bibdigital.epn.edu.ec/handle/15000/4119?mode=full

OpenCV 3.0.0. (10 de Agosto de 2015). Obtenido de http://docs.opencv.org/master/d1/dfb/intro.html#gsc.tab=0

Pinedo, J. (17 de Octubre de 2015). *Raspberry Pi tutorial*. Obtenido de https://www.raspberrypi.org/forums/viewtopic.php?f=31&t=72254

Raspberry Pi Foundation. (20 de Enero de 2016). Obtenido de https://www.raspberrypi.org/

ROBOLOGS. (9 de Julio de 2015). Obtenido de http://robologs.net/2015/07/26/como-filtrar-el-ruido-de-una-mascara-con-opencv/.

Santillan, E. (20 de Noviembre de 2015). *Repositorio Universidad Autónoma Metropolitana.* Obtenido de http://newton.azc.uam.mx/mcc/01_esp/11_tesis/tesis/terminada/080513_garcia_santillan_elias.pdf

Xtreme, K. (16 de Marzo de 2016). *Klip Xtreme Products.* Obtenido de http://www.klipxtreme.com/ec/catalogsearch/result/?q=drivers+webcam+kdc-600

ANEXOS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por los señores: **ISAAC ANÍBAL TRÁVEZ TOCA Y DANILO FERNANDO VIRACOCHA SORIA.**

En la ciudad de Latacunga, a los 05 días del mes de agosto del 2016.

.....

Ing. Eddie Galarza

DIRECTOR DEL PROYECTO

Aprobado por:

.....

Ing. Franklin Silva

DIRECTOR DE LA CARRERA

.....

Dr. Rodrigo Vaca

SECRETARIO ACADÉMICO