



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E**  
**INDUSTRIAL**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y**  
**COMUNICACIONES**

TEMA:

---

“Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes”

---

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Electrónica y Comunicaciones.

SUBLÍNEA DE INVESTIGACIÓN: Procesamiento Digital de Señales e Imágenes

AUTOR: Chávez Carla

TUTOR: Ing. Jurado Marco

Ambato - Ecuador

Julio, 2015

## APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

“Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes”, de la señorita, Chávez Fuentes Carla Patricia, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, julio de 2015

EL TUTOR

---

Ing. Marco Jurado

## AUTORÍA

El presente trabajo de investigación titulado: Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, Julio de 2015

---

Chávez Fuentes Carla Patricia

CC: 1804792602

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación. Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad

---

Chávez Fuentes Carla Patricia

CC: 1804792602

## APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Juan Pablo Pallo e Ing. Santiago Manzano, revisó y aprobó el Informe Final del Proyecto de Investigación titulado “Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes”, presentado por la señorita Carla Patricia Chávez Fuentes de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

---

Ing. Vicente Morales Lozada Mg.

PRESIDENTE DEL TRIBUNAL

---

Ing. Juan Pablo Pallo Mg  
DOCENTE CALIFICADOR

---

Ing. Santiago Manzano Mg  
DOCENTE CALIFICADOR

## **DEDICATORIA**

El presente trabajo lo dedico a toda mi familia, especialmente a mis padres que me apoyan cada día y hermanas que son un pilar fundamental de mi vida, que me han apoyado para luchar por mis sueños y hacerlos realidad.

Carla Patricia Chávez Fuentes

## **AGRADECIMIENTO**

Agradezco de todo corazón a mi madre y hermanas que alimentan mi espíritu con su amor para siempre seguir adelante. A mis maestros que día a día han sabido transmitir sus conocimientos.

A Jesús Guamán que es y será mi apoyo en los buenos y malos momentos.

Carla Patricia Chávez Fuentes

## ÍNDICE

<b>APROBACIÓN DEL TUTOR</b>	<b>ii</b>
<b>AUTORÍA</b>	<b>iii</b>
<b>APROBACIÓN COMISIÓN CALIFICADORA</b>	<b>v</b>
<b>Dedicatoria</b>	<b>vi</b>
<b>Agradecimiento</b>	<b>vii</b>
<b>Introducción</b>	<b>xviii</b>
<b>CAPÍTULO 1 El Problema</b>	<b>1</b>
1.1 Tema de Investigación . . . . .	1
1.2 Planteamiento del problema . . . . .	1
1.3 Delimitación . . . . .	2
1.4 Justificación . . . . .	2
1.5 Objetivos . . . . .	3
1.5.1 General . . . . .	3
1.5.2 Específicos . . . . .	3
<b>CAPÍTULO 2 Marco Teórico</b>	<b>4</b>
2.1 Antecedentes Investigativos . . . . .	4
2.2 Fundamentación Teórica . . . . .	5
2.2.1 Semáforo. Conceptos Fundamentales . . . . .	5
2.2.2 Tipos de Sistemas de Semaforización Inteligente . . . . .	7
2.2.3 Procesamiento de Imágenes . . . . .	12
2.2.4 Operaciones Morfológicas . . . . .	14
2.2.5 Técnicas de segmentación . . . . .	20
2.2.6 Técnicas de detección aplicadas al flujo vehicular . . . . .	27
2.2.7 Software´s para Procesamiento de Imágenes . . . . .	29
2.2.8 Tarjetas de Adquisición de datos . . . . .	32



2.3	Propuesta de Solución . . . . .	33
<b>CAPÍTULO 3 Metodología</b>		<b>34</b>
3.1	Modalidad Básica de la Investigación . . . . .	34
3.1.1	Investigación Bibliográfica . . . . .	34
3.1.2	Investigación de Campo . . . . .	34
3.1.3	Investigación Experimental . . . . .	34
3.2	Recolección de la Información . . . . .	34
3.3	Procesamiento y Análisis de datos . . . . .	35
3.4	Desarrollo del Proyecto . . . . .	35
<b>CAPÍTULO 4 Desarrollo de la propuesta</b>		<b>36</b>
4.1	Antecedentes . . . . .	36
4.2	Factibilidad . . . . .	37
4.2.1	Técnica . . . . .	37
4.2.2	Económica . . . . .	37
4.2.3	Bibliográfica . . . . .	37
4.3	Investigación sobre la situación actual del tráfico vehicular en la ciudad de Ambato . . . . .	37
4.4	Análisis de requerimientos . . . . .	40
4.4.1	Software . . . . .	40
4.4.2	Hardware . . . . .	41
4.5	Elección del Software . . . . .	41
4.6	Comparación de las tarjetas de adquisición. . . . .	41
4.7	Elección de la escala para el desarrollo del prototipo. . . . .	42
4.8	Ubicación de las cámaras y semáforos . . . . .	43
4.9	Elección de la cámara . . . . .	45
4.10	Descripción del Sistema . . . . .	48
4.11	Procesamiento Digital de las imágenes para la detección de vehículos . . . . .	49
4.12	Resultados del Procesamiento de las imágenes . . . . .	52
4.13	Descripción lógica para la duración de las fases de los semáforos . . . . .	57
4.14	Diseño de la Interfaz . . . . .	60
4.15	Conexiones del Sistema . . . . .	63
4.16	Instalación de las cámaras . . . . .	64
4.17	Establecer la Comunicación entre Matlab y Arduino . . . . .	66
4.18	Presupuesto del Prototipo . . . . .	68
4.19	Requerimientos para implementación real. . . . .	69
4.20	Presupuesto Real . . . . .	70

4.21 Pruebas de Funcionamiento del sistema de semaforización. . . . .	70
<b>CAPÍTULO 5 Conclusiones y Recomendaciones</b>	<b>73</b>
<b>Bibliografía</b>	<b>74</b>
<b>ANEXOS</b>	<b>78</b>

## ÍNDICE DE TABLAS

2.1	Clasificación de tarjetas RFID . . . . .	9
4.1	Horarios de mayor concentración vehicular . . . . .	40
4.2	Comparación de las Tarjetas de Adquisición . . . . .	42
4.3	Medidas Reales . . . . .	42
4.4	Medidas en escala 1:64 . . . . .	43
4.5	Webcam con características y precios . . . . .	47
4.6	Fases de los semáforos . . . . .	58
4.7	Asignación de tiempos . . . . .	59
4.8	Presupuesto del Prototipo . . . . .	68
4.9	Presupuesto del Prototipo . . . . .	70
C.1	Características Generales del Arduino Uno . . . . .	83

## ÍNDICE DE FIGURAS

2.1	Partes de un semáforo vehicular . . . . .	5
2.2	Componentes Básicos de un Sistema RFID . . . . .	8
2.3	Esquema típico del sistema de semaforización inteligente usando sensores inalámbricos . . . . .	10
2.4	Elementos de un sistemas de semaforización mediante procesamiento de imágenes . . . . .	11
2.5	Imagen en Modo Monocromático . . . . .	13
2.6	Imagen en modo Escala de grises . . . . .	13
2.7	Imagen en Modo RGB . . . . .	14
2.8	Máscara B . . . . .	14
2.9	Máscara de la imagen. Matriz A . . . . .	15
2.10	Matriz Resultante . . . . .	15
2.11	Imagen Original(izquierda) y dilatación derecha . . . . .	16
2.12	Máscara B . . . . .	16
2.13	Matriz A . . . . .	17
2.14	Matriz Resultante . . . . .	17
2.15	Imagen Original(izquierda) y erosión derecha . . . . .	18
2.16	Imagen Original(izquierda) y su apertura (derecha) . . . . .	18
2.17	Imagen Original(izquierda) y su cierre (derecha) . . . . .	19
2.18	Imagen con un objeto obscuro sobre un fondo gris. . . . .	21
2.19	Histograma de la Figura de 2.17 . . . . .	21
2.20	Ejemplo de Segmentación Basada en Regiones . . . . .	22
2.21	Imagen Dividida . . . . .	23
2.22	Máscara Operador de Roberts . . . . .	23
2.23	Imagen original y Aplicación del Operador Roberts . . . . .	24
2.24	Matrices de Kernell . . . . .	24
2.25	Máscara de filtro Gaussiano . . . . .	26
2.26	Matrices de Sobel y Canny . . . . .	26
2.27	Comparación del método de Canny y Prewitt . . . . .	27
2.28	Ejemplo de la segmentación por redes neuronales . . . . .	28

2.29	Ejemplo de la segmentación por redes neuronales . . . . .	29
2.30	Placa Arduino . . . . .	33
4.1	Concentración diaria vehicular . . . . .	38
4.2	Concentración total vehicular . . . . .	39
4.3	Afluencia vehicular en la Av. Cevallos y la calle Mera . . . . .	44
4.4	Ubicación de semáforos y cámaras . . . . .	44
4.5	Relación distancia focal y ángulo de visión . . . . .	45
4.6	Ubicación de la cámara . . . . .	46
4.7	Cámara KLIP Xtreme . . . . .	47
4.8	Diagrama de bloques del sistema . . . . .	49
4.9	Matrices de una imagen a color . . . . .	49
4.10	Ejemplos de histogramas . . . . .	50
4.11	Histograma de una imagen con un contraste homegeneo . . . . .	51
4.12	a) Imagen RGB de la cámara uno b) Imagen RGB de la cámara dos . . . . .	52
4.13	a) Imagen de la cámara uno con mayor contaste b) Imagen de la cámara dos con mayor contraste . . . . .	53
4.14	a) Imagen binaria de la cámara uno b) Imagen binaria de la cámara dos . . . . .	54
4.15	a) Operación cierre de la cámara uno b) Operación cierre de la cámara dos . . . . .	55
4.16	a) Detección de bordes de la imagen uno b) Detección de bordes de la imagen dos . . . . .	56
4.17	a) Detección de bordes de la imagen uno b) Detección de bordes del la imagen dos . . . . .	57
4.18	Diagrama de Flujo de las Fases de los semáforos . . . . .	60
4.19	Ventana de inicio . . . . .	61
4.20	Entorno de Diseño . . . . .	61
4.21	Entorno de Diseño . . . . .	62
4.22	Interfaz en funcionamiento . . . . .	62
4.23	Entorno de Diseño . . . . .	63
4.24	Conexiones para la aplicación . . . . .	64
4.25	Información de los Adaptadores Instalados . . . . .	65
4.26	Información de los ID de los dispositivos . . . . .	65
4.27	Información de los Adaptadores Específico . . . . .	66
4.28	Conexión del Arduino a la PC . . . . .	66
4.29	Elección de la placa Arduino . . . . .	67
4.30	Información sobre el puerto COM . . . . .	67

4.31	Instalar Arduino . . . . .	67
4.32	Asignar variabley puerto COM . . . . .	68
4.33	Vehículos diagonales en la calle secundaria . . . . .	71
4.34	Vehículos diagonales en ambas calles . . . . .	71
4.35	Vía Principal llena de vehículos . . . . .	71
4.36	Calle Secundaria llena devehículos . . . . .	72
4.37	Efecto al presionar el botón reset . . . . .	72
E.1	Prototipo del sistema . . . . .	86

## RESUMEN

En los últimos años la necesidad de una movilidad vehicular sostenible, hace necesario el desarrollo e implementación de nuevas tecnologías con el fin de mejorar el tráfico vehicular en muchos aspectos como mayor fluidez, reducción de accidentes, entre otros.

La presente investigación se enfoca en el desarrollo de un prototipo, encaminado al control de los tiempos de duración de las fases de los semáforos, directamente ligado con el número de vehículos existentes en la vía, dando siempre preferencia a la vía considerada principal, mediante el procesamiento de imágenes, para el presente estudio se consideró la intersección formada por la Av. Cevallos y la calle J.L. Mera.

El sistema está constituido por dos cámaras que adquieren imágenes de las áreas de interés, para posteriormente realizar el procesamiento de las imágenes que constituye de transformaciones sistemáticas que permiten mejorar la imagen y obtener el número de vehículos existentes mediante la segmentación de bordes aplicando el método de Canny.

Con el número de vehículos de ambas calles, se procede a la toma de decisiones para determinar los tiempos de duración de las diferentes fases, esta información se extrae mediante la placa arduino para ser enviadas directamente a los semáforos.

La interfaz aplicada muestra un botón para iniciar el sistema, se indica el número de vehículos registrados, el tiempo en verde para cada calle y cuenta con un botón que permite resetear el programa.

## ABSTRACT

In recent years the need for sustainable vehicular mobility, makes the development and implementation of new technologies in order to improve traffic in many aspects more smoothly, reducing accidents, among others.

This research focuses on the development of a prototype and taking control of the time duration of the phases of traffic lights, directly linked to the number of existing vehicles on the road, always giving priority to the main route considered by the image processing, for this study formed by the intersection was considered Av. Cevallos and J.L. Mera.

The system consists of two cameras acquiring images of the areas of interest, then performing image processing constitutes systematic changes that improve the image and get the number of existing vehicles by segmenting edges by applying the method Canny.

With the number of vehicles on both streets, proceed to decision making to determine the time duration of the different stages, this information is extracted by the Arduino board to be sent directly to the traffic lights.

Applied interface displays a button to start the system, the number of registered vehicles, the green time for each street and has a button that allows the program Resetar indicated.



## GLOSARIO DE TÉRMINOS

- **Analógico:** Señal de entrada o salida cuyos valores son continuos en el tiempo, es decir que puede tomar infinitos valores.
- **Bimodal:** Distribución numérica en la que aparecen unicamente dos picos.
- **Binario:** Sistema de numeración que consta de dos cifras 0 “cero” y 1 “uno”, es decir que su base es dos. Generalmente es utilizado en sistemas computacionales
- **Bit:** Unidad mínima de información, puede tomar el valor de cero o uno.
- **Digital:** Señal de entrada o salida cuyos valores son dicretos en el tiempo, es decir que puede tomar valores fijos.
- **Electromagnético:** Es una rama de la física que estudia los fenómenos eléctricos y magnéticos creados por cargas eléctricas en reposo o movimiento.
- **Led:** Diodo emisor de luz o light-emitting diode. Es un dispositivo semiconductor el cual emite luz cuando es polarizado directamente.
- **Matriz:** En matemáticas es un conjunto ordenado de elementos matemáticos, dispuestos en filas y columnas.
- **Microondas:** Son ondas electromagnéticas con frecuencias altas, entre los 500 MHz a los 300 GHz. Al tener altas frecuencias las longitudes de onda son relativamente cortas, de ahí su nombre.
- **Sketch:** Plantilla de un programa disponible en Arduino.

## INTRODUCCIÓN

En el presente proyecto se desarrolla el prototipo del Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes. Se basa fundamentalmente en el uso de cámaras para capturar la zona de interés, a través de la programación desarrollada en Matlab identificar la cantidad de vehículos existentes y así asignar los tiempos de fase de los semáforos, mediante las salidas del Arduino.

El proyecto se compone de cinco capítulos, que se describen a continuación:

En el primer capítulo se describen los problemas existentes con el actual sistema semafórico, se incluyen los objetivos que se desean cumplir al finalizar el proyecto y se establece la duración y el lugar para el desarrollo del proyecto.

En el segundo capítulo se resume los conceptos básicos necesarios para el desarrollo del proyecto. Abordando varios temas como: Dispositivos para el control de tránsito, semáforos, procesamiento de imágenes .

En el tercer capítulo se establece la modalidad de investigación, tipos de investigación realizadas para el desarrollo del proyecto, como fueron procesados y analizados los datos. También se enumeran las actividades realizadas para el desarrollo del proyecto.

Dentro del cuarto capítulo se desarrolla la propuesta de solución para el problema planteado, se resuelve la mayoría de los objetivos establecidos, requerimientos del sistema y se muestra el desarrollo y funcionamiento del prototipo. También se encuentra el estudio del presupuesto del prototipo, como el de una instalación real y su factibilidad.

En el quinto capítulo se muestran las conclusiones y recomendaciones.

# CAPÍTULO 1

## El Problema

### 1.1. Tema de Investigación

Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes.

### 1.2. Planteamiento del problema

La congestión vehicular en Sudamérica se ha convertido en un gran problema tanto para conductores como para peatones, esto causa conflictos en la población, ciudades como Buenos Aires, Sao Paulo, Ciudad de México, Lima, Quito, Bogotá o Santiago son protagonistas de largas filas en las vías principales, conductores que se impacientan, bocinas, imprudencia, invasión de carril, ruido, violencia verbal, estrés, gases de los vehículos y embotellamientos [1].

Según los reportes de la Compañía de Ingeniería de tráfico, Sao Paulo-Brasil es una de las ciudades con mayor congestión vehicular, las cifras indican que existe 1 vehículo por cada dos habitantes. Sao Pablo alcanzó una cifra histórica el 1 de junio de 2012 con 295 Km de filas acumuladas alrededor de la ciudad durante la hora pico de la tarde[2] .

El aumento excesivo del parque automotor no es una excepción en el Ecuador, aquí el número de automóviles matriculados ha crecido aproximadamente en un 350 % desde 1997 cuando las cifras indicaban un total de 561,864 vehículos hasta el 2013 en el cual el número creció a 1'717.886 (Cifras tomadas del Instituto Nacional de Estadísticas y Censos del Ecuador).

Del total de vehículos que circulan en el país se establece que el 95.1 % corresponde a vehículos privados, vehículos de renta 3.3 %, estatales un 1.2 % y el menor porcentaje los de uso municipal con 0.4 %[3] .

Según datos de la Asociación de Empresas Automotrices del Ecuador (Aeade), en

el país existe aproximadamente un vehículo por cada 8 habitantes, este crecimiento se debe al poder adquisitivo de los últimos años y por la dolarización[4].

Las largas filas de autos en las calles de las ciudades del Ecuador, manifiestan que es necesario el desarrollo de actividades para mejorar la movilidad vehicular. En las calles se observa como un agente de tránsito se coloca en la vía para permitir que el tráfico fluya por consecuencia de la cantidad de vehículos sin tomar en cuenta lo que indique el semáforo, pero esta no es una medida óptima, principalmente por que el agente corre peligro de ser atropellado, no cumple sus funciones y se está desperdiciando la funcionalidad del semáforo.

El actual sistema semafórico tiene dos inconvenientes principales:

Los conductores se ven obligados a esperar tiempos extensos cuando el semáforo se encuentra en rojo aunque en la vía contigua no existan vehículos.

Están programados de formas diferentes en las horas valle y en las horas pico, pero la mayor concentración vehicular no solo se genera en las consideradas horas pico.

### **1.3. Delimitación**

Área Académica: Comunicaciones.

Línea de Investigación: Tecnologías de Comunicaciones.

Sublínea de Investigación: Procesamiento Digital de Señales e Imágenes.

Delimitación Espacial: La presente investigación se realizó en la ciudad de Ambato.

Delimitación Temporal: El proyecto se realizó en un periodo de 12 meses, desde el 17 de abril de 2014 al 17 de abril de 2015, luego de ser aprobado por el Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

### **1.4. Justificación**

El incremento del parque automotor en los últimos tres años obligan a tomar de medidas de descongestión vehicular, por lo que ha sido necesario abordar en temas de investigación que permitan el descongestión vehiculares. En el Ecuador diversos Gobiernos Autónomos Descentralizados Municipales han adoptado medidas como: Sistema pico y placa, Sistema Integral de Transporte Masivo Urbano “Metrovía”, la motivación al uso de transporte no motorizado y mejoramiento del sistema semafórico,

La implementación de sistemas de transporte Inteligente (ITS), presentan una eficiente y tecnológica solución para el descongestión vehicular. Los sistemas ITS permiten tener un control vehicular y peatonal más eficiente que los sistemas semafóricos adaptativos.

En la actualidad en la ciudad de Ambato existen sistemas de semaforización vehicular adaptativos es decir que las duraciones de fases se encuentran programadas por horarios, el sistema de semaforización inteligente mediante el procesamiento de imágenes permite controlar la duración de las fases de los semáforos en dependencia del número de vehículos existentes en la vía. El tiempo máximo en estado verde es de 90 y 60 segundos en la vía principal y secundaria, respectivamente. Mencionado tiempo será distribuido de acuerdo al número de vehículos existentes en la vía. Dicha distribución de tiempo permitirá un mejor flujo vehicular del parque automotor de la ciudad de Ambato.

Los beneficiarios de este sistema son todas las personas que transiten por las vías ya que minimizará los tiempos de espera innecesarios generados por los semáforos. Los conductores tendrán el beneficio de transitar de forma más fluida, consumir menos combustible y minimizar niveles de estrés.

Los tiempos de viajes se verán drásticamente reducidos.

El presente proyecto es base para el desarrollo de trabajos futuros en sistemas de semaforización con procesamiento de imágenes mejorados como la detección de vehículos de emergencia.

## **1.5. Objetivos**

### **1.5.1. General**

- Diseñar un sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes.

### **1.5.2. Específicos**

- Investigar la situación actual del tráfico vehicular en la ciudad de Ambato.
- Estudiar los distintos tipos de semaforización inteligente.
- Analizar las diferentes técnicas para el procesamiento digital de las imágenes aplicadas al flujo vehicular.
- Implementar un prototipo de sistema de semaforización inteligente para el control de flujo vehicular mediante el procesamiento digital de Imágenes.

## CAPÍTULO 2

### Marco Teórico

#### 2.1. Antecedentes Investigativos

En los últimos años se han desarrollado varios proyectos relacionados con los sistemas de semaforización inteligente, entre los cuales se puede citar:

En la Escuela Técnica Superior de Ingeniería de la Universidad Pontificia Comillas en Madrid reposa el proyecto de fin de carrera de Ernesto de la Rocha Gómez titulado “Mejora del Tráfico en un cruce regulado por semáforos mediante un Sistema basado en Visión Artificial”. El proyecto analiza las distintas técnicas de segmentación orientadas al conteo de vehículos existentes en una vía se estudian: redes neuronales, segmentación basada en luminosidad, segmentación basada en contraste llegando a la conclusión que la mejor para un sistema de semaforización es la basada en bordes, esto en relación al número de vehículos detectados en cada una de las técnicas. Se realiza la implementación de la segmentación basada en bordes en imágenes ya capturadas no en tiempo real, el código de programación se realiza en Matlab, la segmentación para la detección de objetos se realiza mediante el método de Sobel. Su limitación es el tiempo de respuesta del sistema de procesamiento de imágenes, únicamente se realiza el conteo de vehículos no se realiza aplicación[5].

En la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú reposa la Tesis para optar el título de Ingeniero Electrónico de Alex Santos Chávez Córdova denominada “Algoritmo que permite el ingreso de vehículos a una vía de alto tráfico y doble sentido, basado en procesamiento de imágenes”, el algoritmo se enfoca en el ingreso a dicha universidad, se basa en la segmentación por regiones el procesamiento de imágenes se realiza en el programa VIP4L, se obtiene datos como: el número de regiones, cantidad de píxeles que forma cada región, posición del centro de la masa, coordenadas de cada región, para determinar si los vehículos van a ingresar al campus universitario. [6]

El proyecto de Vicente Blas en la Universidad Michoacana, el estudio se realiza en imágenes previamente capturadas del entorno de interés, el procesamiento de imágenes se desarrolla en el software Matlab, se basa en la obtención del fondo , es decir la imagen sin vehículos, para compararla con la nueva imagen, entonces se obtiene los píxeles que no coinciden, posteriormente se obtienen las siluetas de los vehículos, el conteo de vehículos se realiza mediante la conectividad de pixeles[7].

En la Facultad de Ciencias y Tecnologías reposa la investigación realizada por Manuel Martínez donde describe las diferentes tecnologías para la implementación de semáforos inteligentes entre estas consta el procesamiento digital de imágenes aquí exhibe que mediante esta tecnología es posible medir la densidad de tráfico y mediante estos datos modificar los tiempos del semáforo, también manifiesta que esta es una de las tecnologías más eficientes pero una de las más costosas. La investigación también plantea algunas ideas futuristas pero bastante posibles en el desarrollo de los ITS [8].

## 2.2. Fundamentación Teórica

### 2.2.1. Semáforo. Conceptos Fundamentales

Los semáforos son dispositivos electromagnéticos y electrónicos proyectados específicamente para facilitar el control de tránsito de vehículos y de peatones, mediante indicaciones visuales de luces de colores universalmente aceptados como lo son: verde, amarillo (ambar) y rojo.

En la figura 2.1 se observa un semáforo vehicular básico y sus partes:

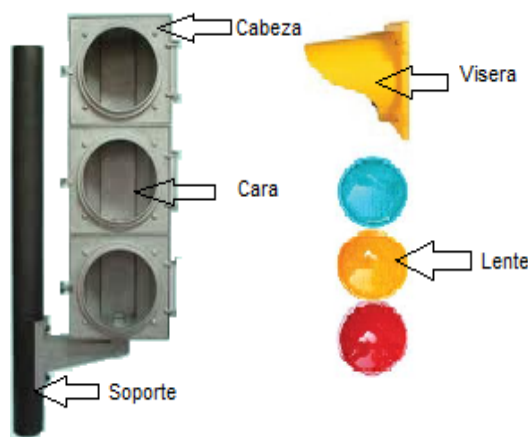


Figura 2.1: Partes de un semáforo vehicular  
Fuente: El Investigador basado en el Reglamento Técnico, INEN

Su finalidad principal es la de permitir el paso alternadamente, a las corrientes de tránsito que cruzan, permitiendo el uso ordenado y seguro del espacio disponible[9].

Los colores de luces del semáforo se definen como sigue:

- Rojo Fijo

Los conductores frente a esta luz deben detenerse antes de la raya de parada.

Los peatones no tienen derecho a cruzar la vía, a menos que algún semáforo les de la indicación que deben hacerlo.

- Amarillo Fijo

Es una advertencia hacia los conductores que el derecho al paso otorgado por la luz verde está por terminar, y que inmediatamente se encenderá la luz roja.

Los conductores deben reducir la velocidad y detenerse antes de la línea de parada. También avisa a los peatones que no cuentan con tiempo suficiente para cruzar, a menos que un semáforo peatonal indique que pueden cruzar.

Sirve para despejar el tránsito en una intersección y evitar a los conductores frenadas bruscas.

- Verde Fijo

Los conductores podrán circular de frente o dar vuelta a la derecha o izquierda, a menos que alguna señal lo impida, los vehículos que giren deberán ceder el paso a otros vehículos o peatones en caso de existir peligro de colisión o atropellamiento.

Los peatones que avancen hacia el semáforo podrán cruzar, a menos que un semáforo lo prohíba.

- Rojo Intermitente

Los conductores deben detenerse inmediatamente antes de la raya de parada, pueden circular con precaución solamente si no existe peligro de colisión con otro vehículo ó atropellamiento a un peatón.

Se aplica esta luz en circunstancias de emergencia, caso de fallas en el sistema semafórico ó en horarios nocturnos. Se aplicará en el acceso de la vía principal

- Amarillo Intermitente

Los conductores pueden fluir pero con precaución, evitando que exista peligro de colisión con otro vehículo o atropellamiento a un peatón Se aplica en la vía con preferencia.

Se aplica esta luz en circunstancias de emergencia, caso de fallas en el sistema semafórico ó en horarios nocturnos



- Verde Intermitente

Advierte a los conductores que el final de la luz verde y que su derecho al paso está por terminar, los conductores deben disminuir su velocidad. [9].

### **2.2.2. Tipos de Sistemas de Semaforización Inteligente**

Sistema de semaforización inteligente es todo aquel sistema capaz de tomar decisiones dependiendo de una serie de parámetros de entrada (flujo vehicular, velocidad media, comparación de flujo vehicular entre calles, identificación de vehículos de emergencia). En otras palabras, el comportamiento del mismo, es de forma dinámica y se ajusta de acuerdo a varios parámetros. Aunque existen distintos tipos de sistemas de semaforización inteligente todos ellos tienen los mismos objetivos que son mejorar la seguridad del tráfico, la optimización de la velocidad del flujo de tráfico, y minimizar el consumo de energía de vehículos que se movilizan en las carreteras [10, 11]

Existen diferentes sistemas de semaforización inteligente, estos son:

Semaforización Inteligente usando Tarjetas RFID

Semaforización Inteligente usando Redes de Sensores Inalámbricos

Semaforización Inteligente usando Procesamiento de Imágenes

#### **■ Semaforización Inteligente usando Tarjetas RFID**

La semaforización inteligente usando tarjetas RFID, consiste fundamentalmente en colocar dichas tarjetas en cada vehículo que se desee contabilizar, a través de una antena se capta el código único de cada tarjeta, entonces se envía una señal de radiofrecuencia de baja potencia y se registra la presencia de un vehículo, esta información se puede almacenar y procesar en un servidor mediante software.

Esta señal de radio transmitida no requiere que la tarjeta esté dentro de la línea visual del lector, ya que las señales de radio pueden propagarse fácilmente a través de materiales no metálicos.

En la figura 2.1 los elementos que forman este sistema, los mismos que se detallan a continuación:

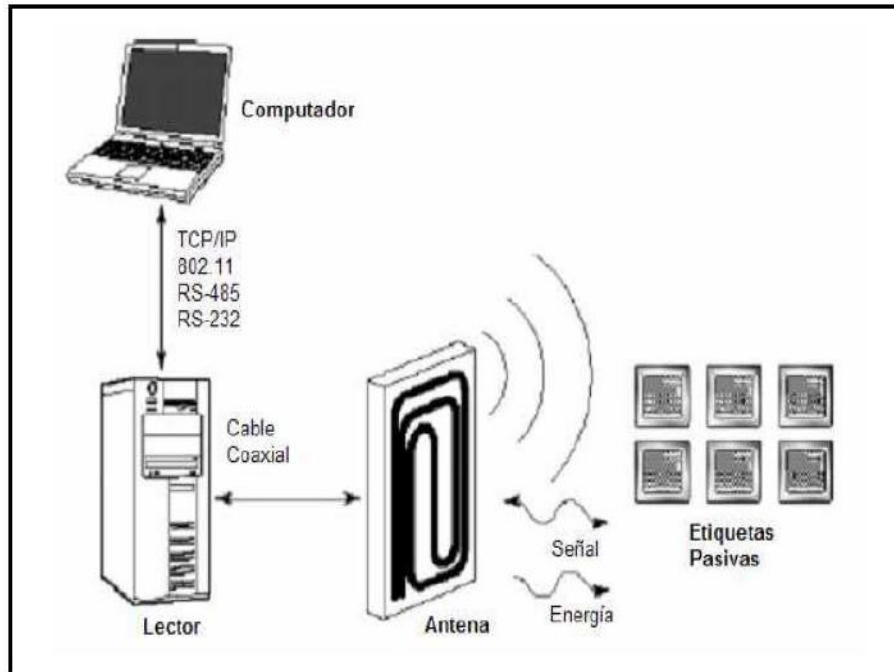


Figura 2.2: Componentes Básicos de un Sistema RFID  
Fuente: RFID, Tecnología, Aplicaciones y Perspectiva, Whitepaper Series

Transponder: (Etiqueta) que contiene un código identificativo.

Antena: Usada para transmitir las señales de RF entre el lector y el dispositivo RFID.

Lector o módulo digital: Encargado de recibir las transmisiones RF desde el dispositivo RFID y proporciona los datos al sistema servidor para su procesamiento.

Al ser un sistema que trabaja con transmisiones RF es vulnerable a reflexiones que causan variaciones de fase. La onda recibida es resultante directa de las ondas reflejadas tanto en frecuencia como en fase, si estas ondas tienen fase opuesta da como resultado que la onda recibida será nula. Es un sistema que no necesita de línea de vista es decir N-LOS entonces se debe considerar que este tipo de transmisiones tiene mayores pérdidas.

El elemento que se desee detectar debe tener colocada la tarjeta RFID o también denominada "tag". En la tabla 2.1 se resume la clasificación de las tarjetas RFID según el tipo de acceso, memoria, material del objeto a detectar y alimentación.

Tabla 2.1: Clasificación de tarjetas RFID

Clasificación	Características
Acceso	Solo Lectura
	Una escritura y múltiples lecturas
	Escrituras y lecturas múltiples
Memoria	Solo ID del tag
	ID del tag + memoria de usuario
Material de objeto a etiquetar	Papel, cartón, plástico, madera
	Recipientes con líquidos
	Metales
Alimentación	Pasivas: Se alimentan de la señal F que emite el lector
	Semi-Pasivas: Tiene una pequeña batería incluida.
	Activas: Contiene una batería de alimentación que amplifica la señal

Fuente: RFID, Tecnología, Aplicaciones y Perspectivas

Sin importar que tipo de tarjeta se utilice este es un sistema en tiempo real lo cual lo hace realmente confiable[12, 13, 14] .

El inconveniente general de este tipo de sistema radica que los objetos que deseen ser detectados, en caso de semaforización los autos, deben contar todos con la tarjeta rfid para ser identificados lo que consiste en un gran inconveniente ya que cada usuario (automóvil) debe adquirir una de estas para que pueda ser detectado y el sistema cumpla con su objetivo.

#### ▪ Semaforización Inteligente Usando Redes De Sensores Inalámbricos

El sistema de semaforización usando redes de sensores inalámbricos, cuenta con 2 partes importantes:

- Red de sensores inalámbricos (Wireless Sensor Network o WSN)
- Estación Base (Base Station o BS).

La red de sensores inalámbricos es un conjunto de sensores, cada sensor se encarga de generar los datos de tráfico como el número de vehículos, velocidad de cada vehículo y longitud de separación entre ellos según la aplicación que se desee. Los datos adquiridos por los sensores se envían a la estación base para almacenarlos y de ser necesario poder utilizarlos.

En la Figura 2.3 se muestra el esquema típico del sistema de semaforización inteligente usando sensores inalámbricos.

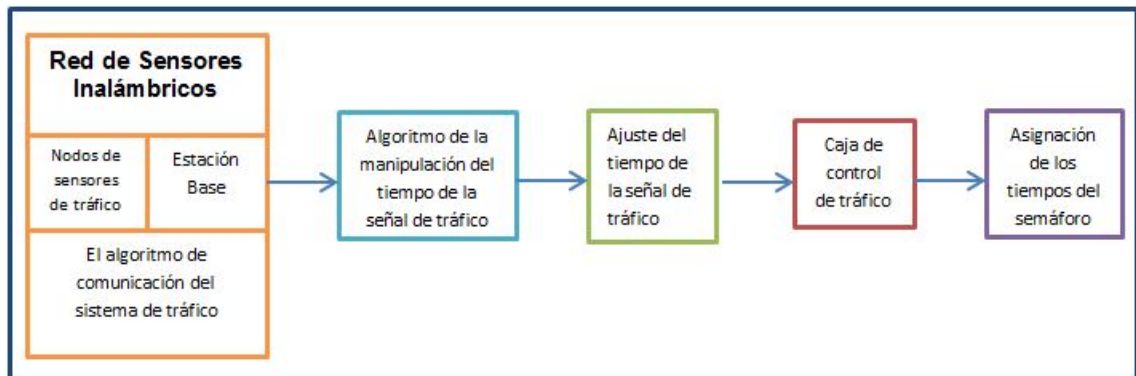


Figura 2.3: Esquema típico del sistema de semaforización inteligente usando sensores inalámbricos

Fuente: Investigador basado en Intelligent traffic light flow, K. Yousef

- El nodo de sensores de tráfico se encarga de detectar los vehículos, contarlos y transmitir la información continuamente hacia la Estación Base.

- El sistema de detección de vehículos requiere de cuatro componentes: el sensor el cual se encarga de detectar las señales generadas por los vehículos, un procesador para los datos detectados, una unidad de comunicación para transferir los datos a una estación base y una fuente de energía.

- El algoritmo de comunicación es el encargado de intercambiar los datos obtenidos por los sensores y la estación base, así como con la caja de control de tráfico.

- Luego los datos obtenidos por la estación base pasan por el algoritmo de la manipulación del tiempo de la señal de tráfico aquí se fijan las duraciones de tiempo de señales de manera dinámica de acuerdo a los parámetros recibidos.

- Asignación de tiempos del semáforo es un algoritmo de tiempo de control adaptativo desarrollado para calcular la duración de la luz roja-verde [11, 15, 16, 17].

Las ventajas y desventajas de este sistema depende de varios factores: tipos de sensores, gestión de datos, entre otros.

El tipo de sensor que se utilice tendrá un gran impacto en el desarrollo del proyecto tanto en complejidad como en gastos. , ya que si se colocan sensores de presión se deben ubicar debajo del asfalto esto implica dificultad de instalación y mantenimiento.

Se puede utilizar sensores ultrasónicos para evitar gastos en construcción vial.

### ■ **Semaforización Inteligente Mediante Procesamiento de Imágenes**

La semaforización inteligente mediante procesamiento de imágenes, consiste básicamente en la adquisición de información respecto al lugar de interés a través de un sensor óptico, mediante la imagen captada se pueden obtener diferente información como la velocidad, distancia, posición de los vehículos, etc. Dicha información depende directamente la programación desarrollada.

Este sistema se compone básicamente de tres elementos: sensor óptico, semáforos y un controlador de tráfico, se muestra en la figura 2.4.

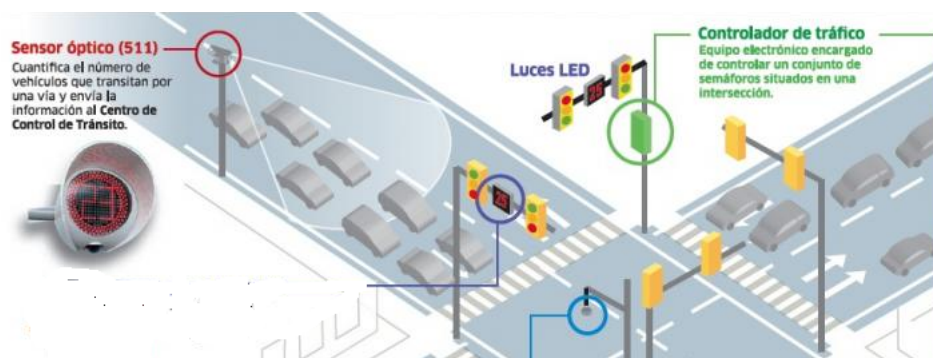


Figura 2.4: Elementos de un sistemas de semaforización mediante procesamiento de imágenes

Fuente:

Para procesar las imágenes se debe cumplir con un flujo de trabajo los cuales son:

- Acceder a los datos de las imágenes.
- Exploración y descubrimiento
- Aplicación

**Acceder a los datos de las imágenes:** Adquirir imágenes en vivo o ya capturadas del objeto de interés. Esto se realiza a través de una cámara ya sea digital o analógica.

**Exploración y descubrimiento:** Establecer las características de la imagen como el número de píxeles. Crear y modificar algoritmos. Esto se consigue mediante software (ej. Matlab, Labview, Opencv)

**Aplicación:** Obtener resultados y aprovecharlos en una aplicación.

Aunque el desarrollo de algoritmos es complejo este permite adaptar al sistema según los requerimientos de la aplicación y mejorarlo en cualquier momento. Se han desarrollado varios software que a través de sus herramientas pueden hacer de este trabajo mucho más simple.

A través de la obtención de imágenes se puede identificar cuando en la carretera transitan vehículos de emergencia como ambulancias, patrullas, bomberos y poder tomar la mejor decisión con respecto al control de fases de los semáforos mediante la aplicación de algoritmos. El algoritmo puede ser modificado y mejorado a partir de las necesidades del sistema [8, 18].

El mayor limitante en este sistema radica que al transcurrir el día la cantidad de luminosidad no es la misma, la reducción de luminosidad en la noche no permite tener imágenes claras para la ejecución del respectivo algoritmo, este problema puede ser superado con la iluminación correcta.

### 2.2.3. Procesamiento de Imágenes

El procesamiento digital de imágenes es una rama de la electrónica que permiten mejorar la calidad de una imagen para la percepción del ojo humano, y permite extraer información de la misma para su posterior aplicación.

Para procesar una imagen es necesario que la misma al ser capturada como una señal física (luz) sea transformada a una señal eléctrica es decir como una imagen digital. El procesamiento de imágenes se encuentra ligado a los siguientes conceptos.

#### ■ Modos de Color de una imagen

Modos de color de una imagen son la representación de colores que conforman cada píxel usando valores numéricos. Los modos de color más utilizados se describen a continuación:

Modo Monocromático: Este modo de color forma a la imagen unicamente con el color blanco y negro, ya que se emplea un bit con el cual se puede representar 0 y 1, no existen otros tonos o valores intermedios. Este tipo de imagen se almacena en una sola matriz se observa en la figura 2.5.



Figura 2.5: Imagen en Modo Monocromático  
Fuente: Imagen Digital, Folgado Laura

Modo Escala de Grises: Este modo permite representar la imagen en tonos o valores intermedios de gris pero también incluye al blanco y negro puros. Este cuenta con una gama de 256 tonos entre 0-255. También se puede definir el color por la cantidad de color negro presente en el mismo es decir entre el 0-100%. En la figura 2.6 se muestra un ejemplo.



Figura 2.6: Imagen en modo Escala de grises  
Fuente: Imagen Digital, Folgado Laura

Modo RGB: Este modo representa a la imagen con 3 canales, Rojo (R), Verde (G) y Azul (B)-RGB-, cada canal contiene 8 bits por lo que su valor de intensidad puede variar entre 0-255, cada píxel puede elegir entre 256 colores. En tipo de imagen se la representa en tres matrices -RGB-. De esta combinación de los tres colores primarios se pueden obtener hasta 16,7 millones de colores algunos de ellos que ni siquiera el

ojo humano puede diferenciar. En la figura 2.7 se observa una imagen en modo RGB [19].



Figura 2.7: Imagen en Modo RGB  
Fuente: Imagen Digital, Folgado Laura

#### 2.2.4. Operaciones Morfológicas

Las operaciones Morfológicas se basan en la teoría de conjuntos, a continuación se describen las más básicas.

- **Dilatación Binaria**

Dadas las matrices  $A$  y  $B$  se denota como dilatación a  $A \oplus B$ , donde  $A$  es la matriz de la imagen y  $B$  es la máscara. Se muestra un ejemplo en la figura 2.8 de una máscara  $B$ .

0	1	0
1	1	1
0	1	0

Figura 2.8: Máscara  $B$

Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

Un ejemplo de matriz  $A$  se muestra en la figura 2.9 que es la matriz de la imagen.



0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 2.9: Máscara de la imagen. Matriz A

Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

Entonces la dilatación consiste en solapar los elementos de B en A, si algún 1 de B coincide con al menos un uno de A, se pone en 1 el elemento de A donde se centro la matriz B. La erosión aplicada con las matrices anteriores da como resultado la matriz mostrada en la figura 2.10

0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0

Figura 2.10: Matriz Resultante

Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

La dilatación tiende a unir los objetos entre ellos y rellena los agujeros en el interior de los mismos. En la figura 2.11 se observa una imagen original y su dilatación.

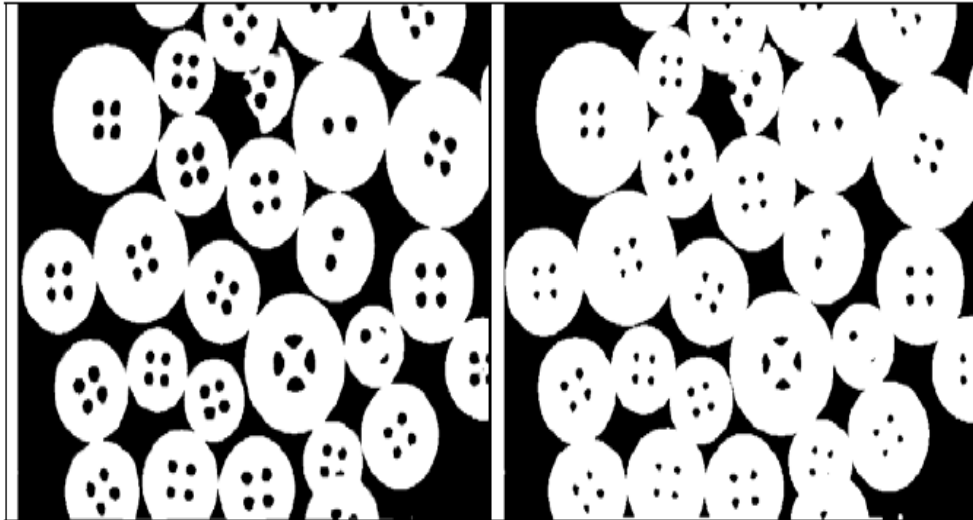


Figura 2.11: Imagen Original(izquierda) y dilatación derecha  
 Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

▪ **Erosión Binaria**

Dadas las matrices  $A$  y  $B$  se denota como erosión a  $A \ominus B$ , donde  $A$  es la matriz de la imagen y  $B$  es la máscara. Se muestra un ejemplo en la figura 2.12 de una máscara  $B$ .

<b>0</b>	<b>1</b>	<b>0</b>
<b>1</b>	<b>1</b>	<b>1</b>
<b>0</b>	<b>1</b>	<b>0</b>

Figura 2.12: Máscara  $B$   
 Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

y un ejemplo de matriz  $A$  se muestra en la figura 2.13 que es la matriz de la imagen.

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Figura 2.13: Matriz A

Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

Entonces la erosión consiste en solapar la máscara B en A y ver si todos los 1 de B coinciden con los de 1 de entonces, se pone en 1, en caso contrario se pone en cero. La matriz resultante se muestra en la figura 2.14.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	1	1	1	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 2.14: Matriz Resultante

Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

Con la dilatación se logra separar objetos entre si ya aumentar el diámetro de agujeros. En la figura 2.15 se observa una imagen original y su erosión.

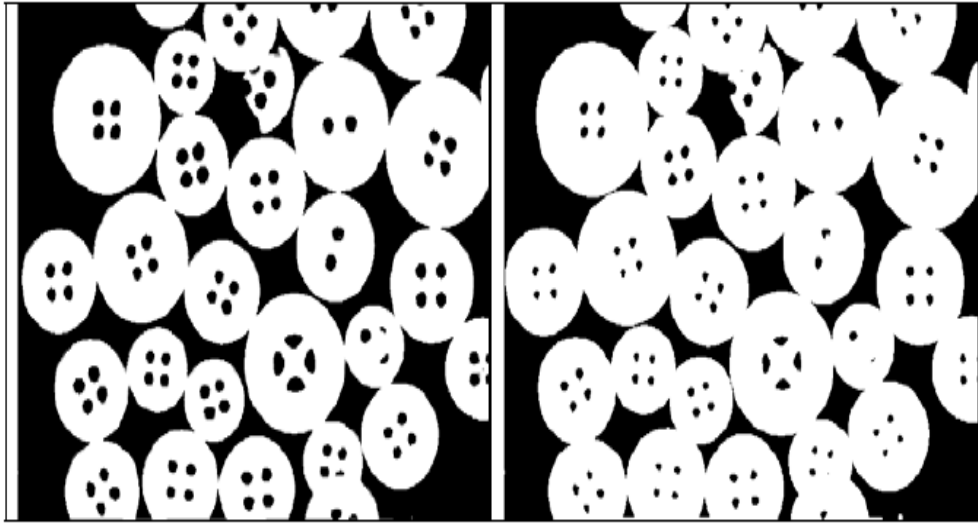


Figura 2.15: Imagen Original(izquierda) y erosión derecha  
 Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

- **Apertura**

La apertura es una operación en la que primero la imágenes sometida a la erosión y luego a la dilatación, la imagen resultante no será la original debido a que estas operaciones no son inversas.

En la figura 2.16 se puede observar una imagen con ruido y su apertura.

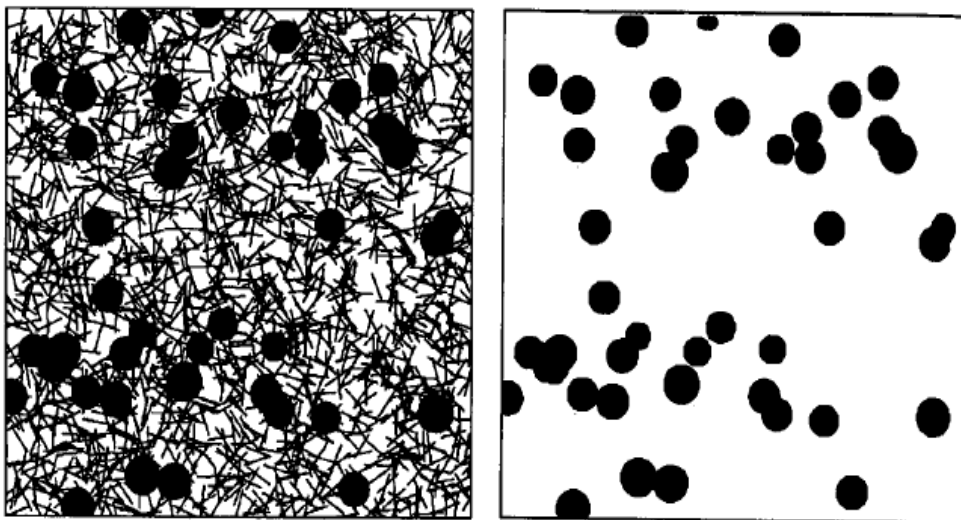


Figura 2.16: Imagen Original(izquierda) y su apertura (derecha)  
 Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

- **Cierre**

El cierre realiza las mismas operaciones que la apertura, pero primero realiza la dilatación y luego la erosión. Con esta operación se suaviza contorno y remueve agujeros. En la figura 2.17 se observa un ejemplo de cierre.[20]

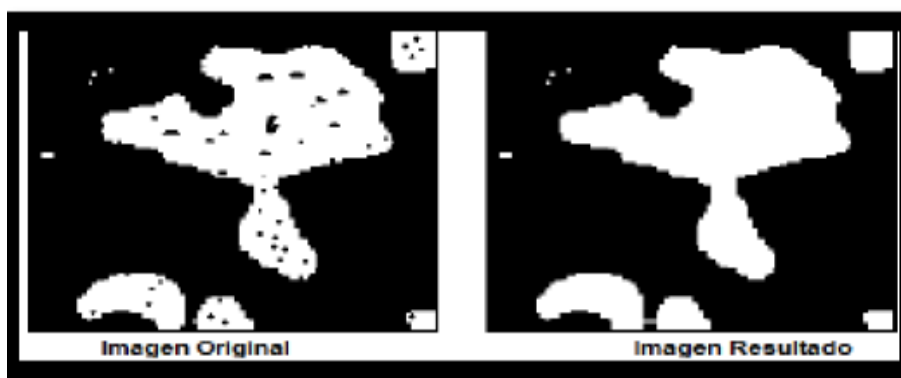


Figura 2.17: Imagen Original(izquierda) y su cierre (derecha)  
Fuente: Técnicas y Algoritmos Básicos de Visión Artificial, Grupo de Investigación EDMANS

- **Segmentación de una Imagen**

Para procesar una imagen es necesaria la segmentación o división de la misma. Este proceso permite encontrar objetos presentes en el entorno de estudio y clasificarlos entre sí. Con una buena segmentación permitirá que el procesado de la imagen sea más eficiente, es decir que se podrán realizar filtrados, obtención de características entre otras operaciones morfológicas de forma independiente a cada objeto presente en la imagen. La segmentación se basa en tres conceptos:

**Similitud:** Los píxeles pueden contener valores parecidos con respecto alguna propiedad determinada.

**Discontinuidad:** Los objetos en un entorno se diferencian del mismo ya que contienen puntos aislados, líneas y aristas.

**Conectividad:** Los píxeles que forman una región o imagen se agrupan es decir se conectan para formar una región homogénea.

### 2.2.5. Técnicas de segmentación

A medida que pasa el tiempo se desarrollan técnicas de segmentación para resolver problemas más complejos, entre estas:

Segmentación Basada en umbralización

Segmentación Basada en Regiones

Segmentación Basada en Bordos o en Contornos

Cada una de estas técnicas se describe a continuación:

- **Segmentación Basada en umbralización por histograma o luminosidad:**

La segmentación basada en umbralización se puede aplicar en una imagen en escala de grises

Este tipo de segmentación es aplicable en imágenes donde el histograma muestra valores claramente diferenciados entre el objeto de interés y el resto de la imagen en cuyo caso el histograma tendrá una distribución bimodal, es decir contará con zonas de valores claramente característicos del objeto de interés lo que permitirá tener una mejor separación de objetos.

Si se aplica un umbral  $T$ , la imagen  $f(x,y)$  se binariza, es decir los objetos distintos del fondo tomarán el valor de '1' y el fondo tomará el valor de '0', esto se aplica si los objetos son más claros que el fondo, la función será:

$$g(x, y) = \begin{cases} 1 & - f(x, y) > T \\ 0 & - f(x, y) \leq T \end{cases} \quad (2.1)$$

Pero si los objetos son más oscuros que el fondo se aplica:

$$g(x, y) = \begin{cases} 1 & - f(x, y) < T \\ 0 & - f(x, y) \geq T \end{cases} \quad (2.2)$$

En la figura 2.18 se observa una imagen con un objeto oscuro claramente identificado sobre un fondo gris.[21, 22, 23]

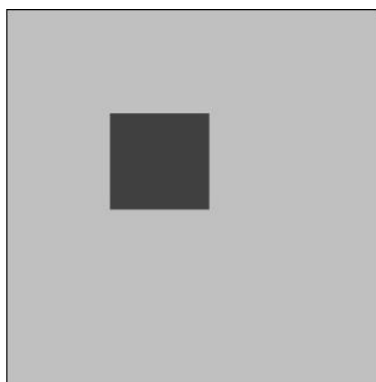


Figura 2.18: Imagen con un objeto obscuro sobre un fondo gris.  
Fuente: Procesamiento digital de imágenes usando Matlab y Simulink, Erick Cuevas

En la figura 2.19 puede verse que el histograma se compone por dos líneas bien diferenciadas, que identifican las regiones existentes en la imagen, en este caso concreto y dada la imagen tan sencilla, se trata de un histograma bimodal. .[18, 20]

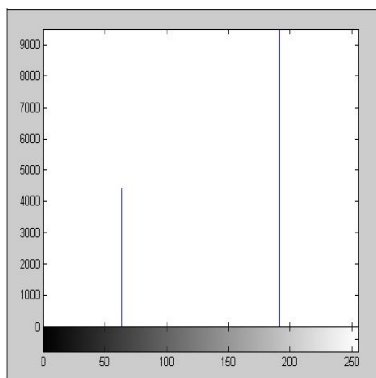


Figura 2.19: Histograma de la Figura de 2.17  
Fuente: Procesamiento digital de imágenes usando Matlab y Simulink, Erick Cuevas

#### ■ Segmentación Basada en Regiones:

La segmentación basada en regiones permite identificar imágenes que comparten características específicas. En la figura 2.20 se muestra una imagen dividida en tres regiones identificadas por su color. La región 1 es el fondo de la imagen, la región 2 y 3 son elementos que se encuentran en la imagen.

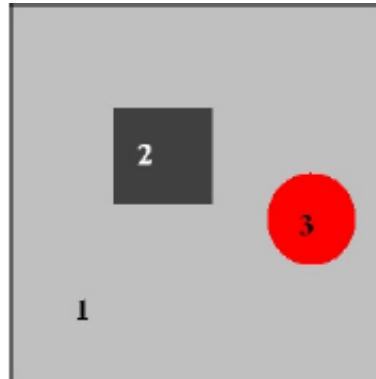


Figura 2.20: Ejemplo de Segmentación Basada en Regiones  
Fuente: Análisis de Métodos de Procesamiento de Imágenes, Alvaro Suárez

Para aplicar la segmentación basada en regiones existen tres algoritmos clásicos:

### **1. Algoritmo de crecimiento de regiones:**

Este algoritmo se basa en dos conceptos: proximidad y similitud mencionados anteriormente. La segmentación de cada región se inicia desde un punto inicial (semilla) desde este punto la región crece y agrupa a los píxeles inmediatos que satisfacen a un criterio determinado, este proceso concluye cuando ya no existen píxeles que cumplen dicho criterio. Finalmente se obtienen las regiones que están formadas por un conjunto de puntos.

### **2. Algoritmo de Fusión y División ,“Split & Merge”**

Como su nombre lo anuncia este algoritmo se basa en la división y fusión de regiones, estas fases se realizan en este orden.

Primero se realiza la división en regiones, si estas regiones son uniformes permanecen intactas y si no lo son este proceso se repite hasta que estas sean uniformes. Finalizado este proceso se realiza la fusión este consiste en unir las regiones con características similares (ej. intensidad de píxeles) y fusionarlas este proceso finaliza cuando ya no existen más regiones por unirse. En la figura 2.21 se muestra una imagen que se ha sometido al algoritmo de división.[18, 20]



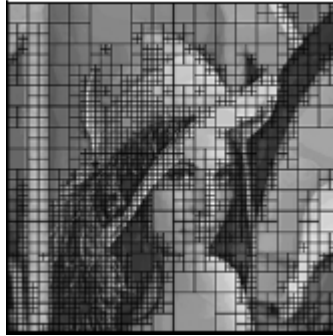


Figura 2.21: Imagen Dividida

Fuente: Análisis de Métodos de Procesamiento de Imágenes, Alvaro Suárez

■ **Segmentación Basada en Bordos o en Contornos:**

La segmentación basada en bordes es la técnica más usada. Los bordes de una imagen se definen como cambios bruscos de nivel de gris que se producen entre regiones.

. Este método se basa en realizar la búsqueda del valor máximo del gradiente, sobre cada línea que forma la imagen. Cuando un máximo es encontrado, un algoritmo de trazado trata de seguir el máximo del gradiente alrededor del objeto, hasta encontrar de nuevo el punto inicial, para luego buscar el próximo máximo en el gradiente. Para la detección de bordes existen diferentes métodos u operadores, se describen a continuación:

**1. Operador Roberts**

Este operador proporciona una aproximación de la gradiente, es más efectivo en bordes diagonales.

Las máscaras usadas en el operador Roberts se muestran en la figura 2.22.

Gradiente fila	Gradiente columna
0	-1
0	0
0	0
0	1
-1	0
0	0

Figura 2.22: Máscara Operador de Roberts

Fuente: Detección de bordes de una imagen, Departamento de Ingeniería Electrónica

Sus ventajas son: Buena respuesta en bordes horizontales y verticales, cálculos simples y tiempo de respuesta rápido.

Sus desventajas: Mala respuesta en bordes diagonales, sensibilidad al ruido y efecto de doble borde.

En la figura 2.23 se muestra una aplicación del operador Roberts.[24]

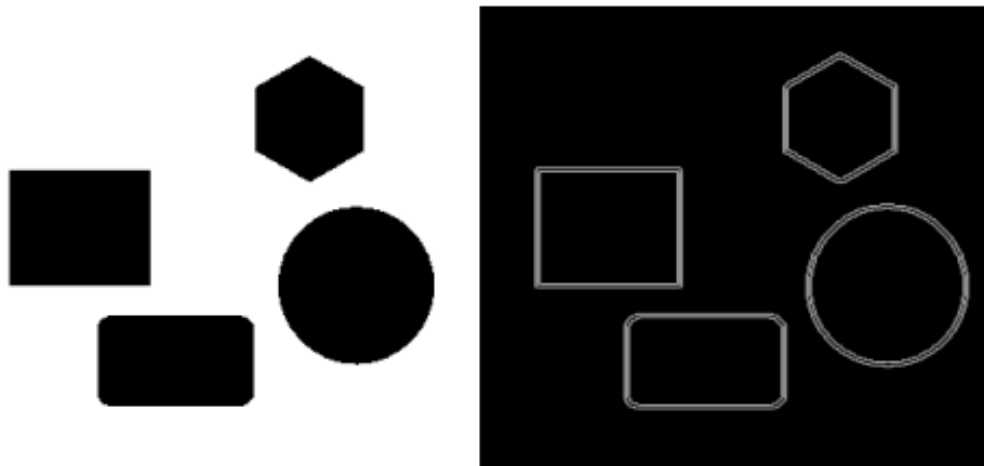


Figura 2.23: Imagen original y Aplicación del Operador Roberts  
Fuente: Visión Computacional, Gómez. G. y Sucar. E.

## 2. Operador de Prewitt, Sobel y Frei Chen

Estos tres métodos pueden formularse de manera conjunta con las matrices de kernell mostradas en la figura 2.24.

	Gradiente fila		Gradiente columna																		
$\frac{1}{2+K}$	<table border="1" style="border-collapse: collapse; margin: auto;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>K</td><td>0</td><td>-K</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	K	0	-K	1	0	-1	$\frac{1}{2+K}$	<table border="1" style="border-collapse: collapse; margin: auto;"> <tr><td>-1</td><td>-K</td><td>-1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>K</td><td>1</td></tr> </table>	-1	-K	-1	0	1	0	1	K	1
1	0	-1																			
K	0	-K																			
1	0	-1																			
-1	-K	-1																			
0	1	0																			
1	K	1																			

Figura 2.24: Matrices de Kernell  
Fuente: Detección de bordes de una imagen, Departamento de Ingeniería Electrónica

Para el método de Prewitt, K toma el valor de 1, se involucran a los vecinos adyacentes tanto de filas y columnas, es menos sensible al ruido.

Para el método de Sobel, K toma el valor de 2, sus resultados en la práctica es muy parecido a Prewitt.

Para el método de Frei Chen, K toma el valor de  $\sqrt{2}$ , el gradiente es igual para bordes verticales, horizontales y diagonales.

Las ventajas de estos operadores son: Buena respuesta en bordes horizontales y verticales y robusto frente al ruido.

Las desventajas de estos operadores: Mala respuesta en bordes diagonales, complejidad y lento tiempo de respuesta y efecto de doble borde.

### 3. Operador de Canny

En 1986 Canny dió a conocer un método para la detección de bordes con mejores resultados que los operadores anteriormente mencionados pero con mayor complejidad computacional.

El método de Canny se basa en tres criterios:

- Criterio de detección, tiene como finalidad evitar la eliminación de bordes importantes y no generar bordes falsos.
- Criterio de localización, que establece que la distancia entre la posición real y la posición localizada para el borde debe ser minimizada.
- Criterio de respuesta única, permite obtener un pixel por cada punto de borde, es decir no se generan múltiples bordes.

Para realizar el método de canny se siguen los siguientes pasos:

**Suavizado de la imagen:** Se obtiene mediante la aplicación de un filtro Gaussiano que consiste en aplicar la convolución como se muestra en la ecuación 2.3 entre la matriz de la imagen y una máscara determinada.

$$I = A * B \tag{2.3}$$

Donde:

I, es la matriz de la imagen resultante.

A, es la matriz de la imagen original.

B, es la máscara.

En la figura 2.25 se muestran la máscaras típicas para el filtro guassiano.

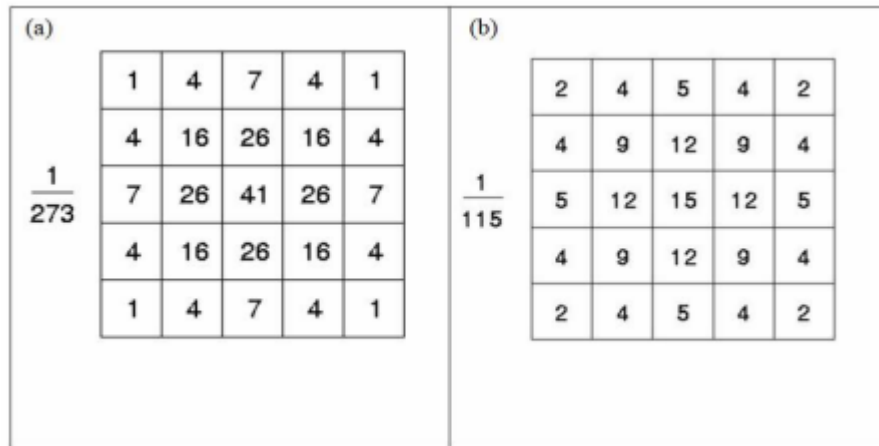


Figura 2.25: Máscara de filtro Gaussiano

**Obtención del Gradiente:** El gradiente de una ina imagen  $f$  en un punto  $x,y$  se define por la ecuación 2.4

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta}{\delta x} f(x, y) \\ \frac{\delta}{\delta y} f(x, y) \end{bmatrix} \quad (2.4)$$

En la figura 2.26 se muestra un ejemplo de aplicación del método de sobel y canny.

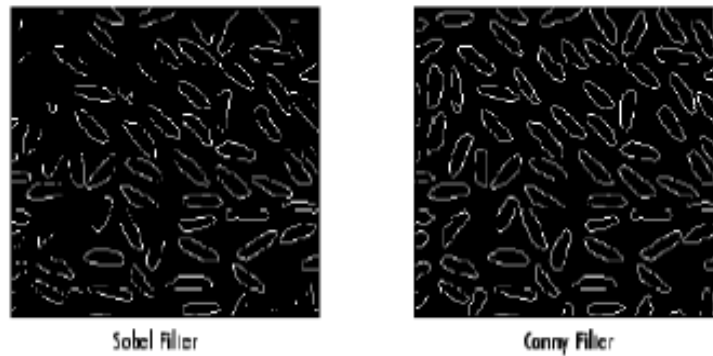


Figura 2.26: Matrices de Sobel y Canny

Fuente: Detección de bordes de una imagen, Departamento de Ingeniería Electrónica

Se observa que el método de Canny arroja objetos bordes más definidos que el método de Sobel.

Esta clara superioridad del método de Canny también ocurre en la figura 2.27 donde se compara Canny y Prewitt.[20, 25, 26]



Figura 2.27: Comparación del método de Canny y Prewitt  
Fuente: Detección de bordes de una imagen, Departamento de Ingeniería Electrónica

Este método es uno de los mejores para la detección de bordes, debido a su robustez frente al ruido, eficiencia y sencillez, la cual permite que la velocidad de procesamiento sea mínima[24]

#### 2.2.6. Técnicas de detección aplicadas al flujo vehicular

- **Detección basada en redes neuronales:**

Esta técnica se basa en la recolección de patrones de vehículos, el correcto funcionamiento de este sistema depende del número de patrones ingresados en el mismo, este sistema tiene una gran desventaja, el patrón de un mismo vehículo será distinto solo si su ángulo de ubicación cambia, es decir que el número de patrones de un mismo vehículo debe ser amplio para que sea detectado correctamente, esto implica una gran capacidad de memoria. Esta técnica no muestra resultados favorable.[5]

En la figura 2.28 se muestra el resultado de esta técnica.



Figura 2.28: Ejemplo de la segmentación por redes neuronales  
Fuente: Mejora del tráfico, Ernesto Rocha

- **Detección basada en luminosidad:**

Se basa en el cambio de luminosidad que existe entre el asfalto y el vehículo, se discriminan objetos con un cierto umbral de luminosidad. Los problemas generados en esta técnica son:

No detecta la presencia de vehículos de colores oscuros (ej azules, plomos, negros).

Considera sombras generadas por el contraste de la luz como vehículos.

Su confiabilidad es del 60 %

En la imagen 2.29 se muestra el resultado de esta técnica.



Figura 2.29: Ejemplo de la segmentación por redes neuronales  
Fuente: Mejora del tráfico, Ernesto Rocha

### 2.2.7. Software´s para Procesamiento de Imágenes

Existen diferentes software orientados al procesamiento de imágenes, tanto de paga como libres. Se estudiarán: Matlab, Labview, OpenCV y Phyton.

- **Matlab**



Matlab (MATrix LABoratory) es un lenguaje de alto nivel desarrollado por la empresa Mathworks que permite analizar datos, desarrollar algoritmos, modelos, aplicaciones y realizar cualquier tipo de cálculo con vectores y matrices. Adicionalmente permite mostrar gráficos en dos y tres dimensiones. Es uno de los lenguajes más empleados.

Matlab es mejorado cada día ya que se desarrollan librerías específicas para determinadas aplicaciones estas se denominan toolboxes.

Los toolbox son archivos con extensión .m que contiene una serie de funciones que mejoran la funcionalidad de Matlab en alguna aplicación en especial como por ejemplo: el procesado de señales y de imágenes, la lógica difusa, los sistemas de control, etc.

Es una herramienta potente y precisa para realizar operaciones complejas. En una sesión de trabajo típica, no hay que compilar o crear ejecutables y los ficheros son de texto, por lo que ocupan poca memoria.

Es capaz de intercambiar datos con otros lenguajes, admite cualquier dispositivo de hardware como tarjetas de sonido, tarjetas de adquisición de datos y DSPs (Digital Signal Processors).

Para facilitar el tratamiento de imágenes Matlab ha desarrollado un toolbox denominado Image Processing , permite utilizar funciones y herramientas probadas y robustas, este permite realizar varias funciones, entre las más importantes: análisis de imágenes y estadística, diseño de filtros y recuperación de imágenes, mejora de imágenes, operaciones morfológicas, definición de mapas de colores, modificación gráfica, operaciones geométricas y transformación de imágenes, soporta casi todos los formatos y tamaños de imagen. [27, 24]

Matlab es un software comercial que se divide en 3 categorías:

Estandar: Para uso de una organización comercial, de gobierno o privada. Precio de Matlab: \$2500 y Image Processing Toolbox: \$1250

Educación: Impartir cursos en una entidad educativa. Precio de Matlab: \$625 y Image Processing Toolbox: \$250

Estudiante: Uso para estudiantes con fines educativos y de investigación. Precio de Matlab: \$89 y Image Processing Toolbox: \$25

Casa: Únicamente uso personal. Precio de Matlab: \$135 y Image Processing Toolbox: \$39.[27]

- **Labview**



LabVIEW es una plataforma de programación gráfica desarrollada por la empresa National Instrument que ayuda a ingenieros a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Este software emplea una



programación gráfica o también denominada lenguaje G, crea programas basados en bloques. Es un lenguaje fácil de aprender e intuitivo, ya que se basa en símbolos. LabVIEW posee extensas librerías de funciones y subrutinas. Adicionalmente de las funciones básicas de cualquier lenguaje de programación, cuenta con librerías específicas para: Adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos. LabVIEW también proporciona potentes herramientas que facilitan la depuración de los programas.

Labview se puede conectar con todo tipo de hardware como: tarjetas insertables, controladores de movimiento y controladores lógicos programables (PLCs), cabe mencionar que National Instruments cuenta con sus propias tarjetas de Adquisición denominadas DAQ, Labview responde de manera más óptima en cuanto a tiempo de respuesta con el uso de estas tarjetas.[24]

Labview es un software comercial con tres distribuciones:

Sistema de Desarrollo Base de NI LabVIEW para Windows, precio:\$1100

Sistema de Desarrollo Completo de LabVIEW, precio:\$3300

Sistema de Desarrollo Profesional de LabVIEW, precio:\$5500

De las distribuciones expuestas la que incluye funciones para análisis y procesamiento de imágenes es la segunda.

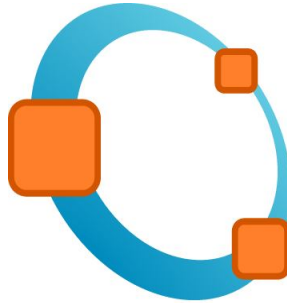
- **OpenCV**



Open CV ( Open Source Computer Vision ) es una poderosa herramienta orientada a la visión artificial, desarrollada originalmente por Intel. Aplicada en distintas áreas como: Sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Cuenta con unalicensia BSD, esta permite que su uso sea libre para aplicaciones comerciales y de investigación. Se encuentra programado en lenguaje C y C++.

Open CV puede ser instalado en GNU/Linux, Mac OS X y Windows es decir que es multiplataforma . Contiene más de 500 funciones y algoritmos para el manejo de dispositivos de captura de video y procesamiento digital de imágenes. que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estérea y visión robótica.[28]

- **Octave**



Octave es un programa desarrollado dentro del proyecto GNU, originalmente es compatible con UNIX pero existen versiones multiplataforma. Su programación se basa en vectores y matrices por lo que se lo considera el equivalente de Matlab pero en la categoría de software libre.

Octave se ha aplicado en diferentes áreas como: cálculo numérico, estadística, procesamiento de señal, de imagen, entre otros. [29]

### **2.2.8. Tarjetas de Adquisición de datos**

La adquisición de datos es el proceso de medir con una PC un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido.

- **DAQ (NI)**

Los dispositivos de adquisición de datos (DAQ) de National Instruments, tienen la ventaja de su simplicidad y portabilidad de USB y ofrecen medidas reales de alto rendimiento. Van desde dispositivos de bajo costo y de una sola función a sistemas modulares de alto rendimiento, ideales para una variedad de aplicaciones desde registro de datos simple hasta sistemas embebidos OEM. Estas muestran un mejor desempeño trabajando con el software de National Instruments.

#### **Tipos de DAQ**

Existen varios tipos, se presentan las más básicas:

USB 2000

USB 6501

USB 6001

## ■ ARDUINO



Figura 2.30: Placa Arduino

Fuente: <http://media.digikey.com/Photos/Arduino/A000046.jpg>

Arduino es una placa electrónica como se ilustra en la figura 4.30, basada en hardware y software libre, lo que la convierte en una plataforma flexible y de fácil uso. Arduino es ideal para desarrolladores, aficionados y cualquier persona que desee construir proyectos electrónicos.

Arduino es capaz de tomar información de señales analógicas o digitales las cuales pueden accionar luces, motores, etc. Arduino es capaz de comunicarse con otros softwares entre estos matlab.

### **Tipos de Arduino**

- Arduino Mini
- Arduino Uno
- Arduino Leonardo
- Arduino Mega
- Arduino Due

### **2.3. Propuesta de Solución**

La implementación de este sistema permitirá mejorar el flujo vehicular, ya que este actuará en dependencia al número de vehículos registrados a través de un detector (cámara), reduciendo así el tiempo de espera innecesarios a los que se ven sometidos los conductores con el sistema semafórico actual. El sistema es capaz de detectar la vía con mayor flujo vehicular y configurar la fases de los semáforos de la forma más óptima, para que exista un mejor flujo vehicular.

## **CAPÍTULO 3**

### **Metodología**

#### **3.1. Modalidad Básica de la Investigación**

##### **3.1.1. Investigación Bibliográfica**

En el presente proyecto se ha hecho una investigación bibliográfica ya que se ha recopilado información de libros, tesis de postgrado y tutoriales, relacionados al tema tanto en teoría como en el uso de los software. Ampliando así los conocimientos para el desarrollo del proyecto

##### **3.1.2. Investigación de Campo**

En el presente proyecto se ha hecho una investigación de campo debido a que se cuenta con datos reales sobre el flujo vehicular en la ciudad de Ambato, lo que permitió conocer de mejor forma la situación actual del tráfico vehicular.

##### **3.1.3. Investigación Experimental**

Se realizó una investigación experimental ya que se hicieron pruebas del prototipo para poder conocer su efectividad en tanto a la detección correcta de vehículos y el tiempo de respuesta del sistema.

#### **3.2. Recolección de la Información**

Para el desarrollo del proyecto “Sistema de semaforización inteligente para el control de flujo vehicular mediante el Procesamiento Digital de Imágenes” se obtuvo información real sobre el tráfico vehicular en la ciudad de Ambato proporcionada por la UMT. Se hicieron pruebas del prototipo simulando circunstancias reales.

También se obtuvo información respecto al tema de libros e internet.

### **3.3. Procesamiento y Análisis de datos**

La información consultada tanto en libros y en Internet fue analizada y ordenada de forma objetiva para el desarrollo del prototipo y del informe final.

Los datos proporcionados por la UMT de los aforos vehiculares realizados en distintos puntos de la ciudad de Ambato fueron tabulados y analizados para obtener información sobre la situación vehicular actual en la ciudad de Ambato.

### **3.4. Desarrollo del Proyecto**

Acudir a la Unidad Municipal de Tránsito (UMT) de Ambato, específicamente al centro de monitoreo de tránsito.

Tabular los datos proporcionados por la UMT sobre el tráfico vehicular en distintos puntos de la ciudad.

Ubicar geográficamente los puntos de aforo vehicular proporcionados por la UMT.

Ejecutar el estudio del tráfico vehicular en la ciudad de Ambato.

Elaborar una maqueta basada en la intersección Av.Cevallos y J. L. Mera en una escala 1:64.

Construir las placas electrónicas de los semáforos.

Ubicación de las cámaras y los semáforos.

Desarrollo del software aplicativo con técnicas del Procesamiento Digital de Imágenes.

Definición de conclusiones y recomendaciones.

Elaboración del informe final.

## CAPÍTULO 4

### Desarrollo de la propuesta

#### 4.1. Antecedentes

Los primeros sistemas semafóricos de la ciudad de Ambato fueron instalados por la Policía Nacional del Ecuador cuya tecnología se basaba en relés y bombillas incandescentes. Al transcurrir de los años esta fue reemplazada por un control basado PLC (Controlador Lógico Programable), su mayor inconveniente radicaba en el tiempo para determinar los daños de los semáforos, los cuales no se podían solucionar de forma inmediata.

El jueves 26 de abril de 2012 el Consejo Nacional de Competencias (CNC), otorga las competencias de tránsito al Municipio de Ambato para planificar, regular y controlar el tránsito y el transporte público dentro de su territorio cantonal información que consta en el Art. 264 numeral 6 de la Constitución de la República [30].

El mismo año se inicia el proceso de transferencia de competencia, el cual culmina dos años más tarde. Durante este tiempo la Policía Nacional y el GAD Municipalidad de Ambato trabajan conjuntamente en el mantenimiento preventivo y correctivo del sistema semafórico.

Entonces se inicia la semaforización en la periferia de la ciudad, basándose en controladores electrónicos y luminarias con tecnología led, este tipo de luminarias representan grandes beneficios por su mayor tiempo de vida útil, ahorro energético y mejor visibilidad para los usuarios.

En el año 2013 se implementa la semaforización centralizada en el casco central urbano de la ciudad, el cual tiene varias ventajas. Consiste en el adicional del monitoreo del sistema semafórico en tiempo real, capaz de ser reconfigurado los tiempos de las fases de los planes horarios programados en cada intersección desde la central de monitoreo, cuenta con una conexión tanto inalámbrica como alámbrica establecida con fibra óptica, este monitoreo permite detectar fallas inmediatamente.

También se han colocado en las calles con mayor circulación vehicular cámaras que incorporan un sistema de conteo vehicular, que permite establecer la configuración de los tiempos de fases más convenientes en los distintos horarios [31].

## **4.2. Factibilidad**

### **4.2.1. Técnica**

El desarrollo del proyecto es técnicamente factible, debido a que los elementos electrónicos utilizados para la elaboración del prototipo planteado son de fácil adquisición en el país.

### **4.2.2. Económica**

El desarrollo del proyecto no representa grandes gastos y se encuentran al alcance del investigador. Los elementos electrónicos y recursos bibliográficos a utilizar son fácilmente accesibles en el país y no ha sido necesaria ninguna exportación lo cual implica gastos adicionales.

### **4.2.3. Bibliográfica**

La información necesaria para el desarrollo del presente proyecto se encuentra tanto en libros, revistas científicas, tesis de postgrado, investigaciones y publicaciones técnicas relacionadas al temas. Matlab es uno de los software más utilizados a nivel de ingeniería, de este existen diversos tutoriales para su uso tanto en libros como en la web. Con respecto a Arduino está es una plataforma de hardware libre, del cual existen muchos proyectos desarrollados en la web con su código y conexiones respectivas.

## **4.3. Investigación sobre la situación actual del tráfico vehicular en la ciudad de Ambato**

En la ciudad de Ambato existen varios sitios y horarios donde se concentra la mayor afluencia vehicular. Los horarios considerados con mayor circulación vehicular se denominan horas pico.

La hora pico de la mañana es de 7:00/8:30 horas, debido al ingreso de personas a distintas entidades tanto públicas como privadas, los estudiantes a los establecimientos educativos entre otras actividades que se realizan en este horario.

En la tarde la hora pico consiste de 12:00/14:00 horas, ya que en este intervalo de tiempo culmina la jornada matutina e inicia la jornada vespertina de trabajadores

y estudiantes.

De 18:30/20:00 horas, se establece como la hora pico de la noche, la concentración vehicular se debe a que la mayoría de actividades realizadas por los ciudadanos culminan y se movilizan a sus hogares.

Las calles con mayor concentración vehicular se encuentran en el casco central de la ciudad, según datos de la UMT, para obtener una muestra del tráfico existente se analizan 5 calles pertenecientes al casco central y una calle fuera de este.

- Luis A. Martínez - Entre Juan Benigno Vela y Cevallos
- Av. Cevallos sentido Sur (Entre Martínez y Mera)
- 13 de Abril (Ingreso al centro)
- Pérez de Anda y Montalvo
- Eugenio Espejo (Entre 12 de Noviembre y los Andes)
- Av. Atahualpa (Entre Mercurial y Victor Hugo)

En la figura 4.1 se muestra el número de vehículos que circulan diariamente en la ciudad de Ambato, resultados de los aforos realizados del 2 al 8 de junio de 2014, entre las 6 de la mañana a 10 de la noche, proporcionados por la UMTA.

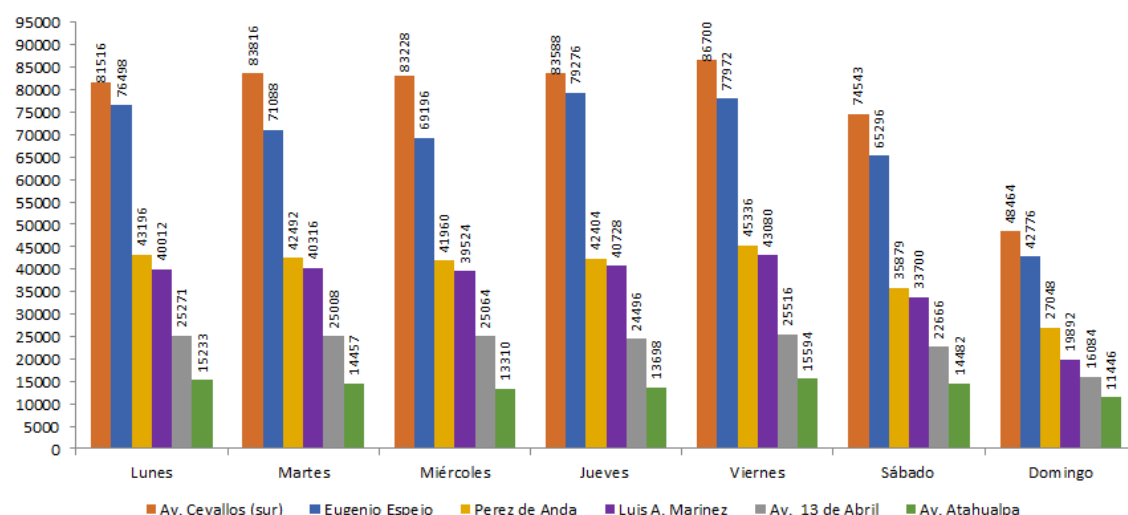


Figura 4.1: Concentración diaria vehicular  
Fuente: El Investigador basado en datos proporcionados por la UMTA



En la figura 4.1, se observa que no hay un día específico en el que se exista un mayor flujo vehicular. Sin embargo el domingo, es el día que presenta un menor flujo vehicular.

En la Av. Atahualpa entre Mercurial y Victor Hugo se concentra la menor cantidad de vehículos diariamente, con una cifra máxima de 15233. En este sitio se genera la menor concentración de vehículos en comparación al resto de los puntos de aforo vehicular.

En la figura 4.2, se presentan el número de vehículos que circulan por las diferentes sitios anteriormente mencionados. La Av. Cevallos, es la más transitada con un total de 541855 de vehículos a la semana seguida por la calle Eugenio Espejo con un total de 482102, mientras que la Av. Atahualpa y la Av. 13 de Abril son la menos transitados con un total de 98200 y 164105 a la semana, respectivamente. Estos datos permiten concluir que el mayor concentración de vehículos en la ciudad de Ambato se encuentra en el casco central.

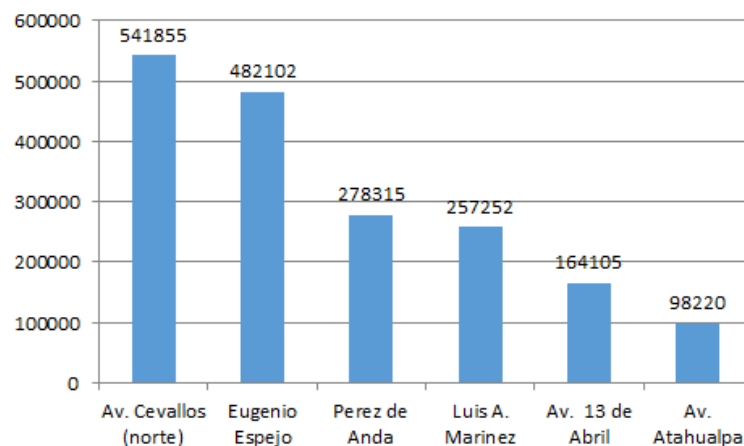


Figura 4.2: Concentración total vehicular

Fuente: El Investigador basado en datos proporcionados por la UMTA

En la tabla 4.1, se muestran los horarios de mayor concentración vehicular en las diferentes calles y avenidas de Ambato, en las cuales se realizaron los aforos vehiculares. No existen un horario fijo en el que se concentre un mayor flujo vehicular, este varia según el día y el sector. Sin embargo, en la Av. Atahualpa del día lunes a sábado en horarios de 19 a 20 horas se presenta la mayor concentración vehicular en el día.

Tabla 4.1: Horarios de mayor concentración vehicular

	Lunes	Martes	Miércoles	Jueves	Viernes	Sabado	Domingo
Luis A. Mart.	8:00/9:00	8:00/9:00	10:00/11:00	15:00/16:00	15:00/16:00	10:00/11:00	10:00/11:00
Av. Cevll. (S)	15:00/16:00	12:00/13:00	17:00/18:00	13:00/14:00	15:00/16:00	10:00/11:00	13:00/14:00
Av 13 de Abril	7:00/8:00	15:00/16:00	7:00/8:00	7:00/8:00	9:00/10:00	10:00/11:00	11:00/12:00
Perez de Anda	19:00/20:00	13:00/14:00	13:00/14:00	13:00/14:00	13:00/14:00	13:00/14:00	13:00/14:00
Eugen. Espejo	15:00/16:00	16:00/17:00	16:00/17:00	13:00/14:00	15:00/16:00	10:00/11:00	10:00/11:00
Av. Atahualpa	19:00/20:00	19:00/20:00	19:00/20:00	19:00/20:00	19:00/20:00	19:00/20:00	15:00/16:00

Fuente: Realizado por Investigador de acuerdo a los datos de UMT Ambato

De los datos analizados se concluye que la mayor concentración vehicular se concentra en el casco central de Ambato. La Av. Cevallos entre Martínez y Mera, presenta una concentración vehicular de 3 veces la Av 13 Abril y 5 veces la Av. Atahualpa, convirtiendo a este sitio en un sector idóneo para realizar el modelamiento del sistema de semaforización inteligente mediante procesamiento de imágenes.

#### 4.4. Análisis de requerimientos

Después de analizados los conceptos de semaforización inteligente mediante el procesamiento de imágenes se define que sistema se compone de tres fases fundamentales: adquisición, procesamiento y aplicación.

**Adquisición:** Permite obtener la imagen de interés para su posterior procesamiento.

**Procesamiento:** Permite establecer las características de un imagen y modificarlas mediante operaciones morfológicas.

**Aplicación:** Aprovechar los datos obtenidos por medio del procesamiento para aplicarlos en una determinada función

Para el desarrollo del sistema es necesario establecer los siguientes requerimientos:

##### 4.4.1. Software

Para el desarrollo del sistema se requiere software que contenga librerías para procesamiento de imágenes, fácil de usar, amigable con el usuario y además de ser capaz de comunicarse con tarjetas para adquisición de datos.

#### **4.4.2. Hardware**

Para el funcionamiento del sistema de semaforización inteligente son necesarios los siguientes dispositivos:

- **Computador:** Este debe contar con un procesador de alto nivel para el sistema trabaje en tiempo real. Debe soportar como mínimo 3 puertos usb, para conectar las cámaras y la tarjeta de adquisición.
- **Cámaras:** Las cámaras deben tener alta resolución y de preferencia ambas con las mismas características, que sean digitales y con conexión de puerto usb, de preferencia pequeñas ya que es un prototipo.
- **Semáforos:** El prototipo se enfoca en semáforos vehiculares por lo que son necesarias placas que simulen su funcionamiento es decir que contengan las tres luces: roja, ambar y verde.
- **Tarjeta de Adquisición de Datos:** Debe permitir extraer los datos del software de procesamiento de imágenes para enviarlos cada una de las entradas de los semáforos, debe contener como mínimo 6 salidas digitales y ser alimentada a través de la conexión usb.

#### **4.5. Elección del Software**

El estudio de los diferentes softwares de procesamiento de imágenes, realizados en el capítulo 2 sección 2.7. A permitido inclinarse por el uso de Matlab, debido a que: El software de MATLAB no necesita una gran capacidad de memoria.

Es un software es robusto.

Es un software que presenta un interface amigable para el uso estudiantil.

Los toolbox de MATLAB contienen funciones y algoritmos totalmente confiables y comprobados.

Acceso a seminarios en línea sin costo.

Abundante información en internet para su uso.

#### **4.6. Comparación de las tarjetas de adquisición.**

Las diferentes tarjetas de adquisición de datos abordadas en el capítulo 2 sección 2.8, se resumen en la tabla 4.2, en la que se muestran, el costo de cada una de las tarjetas y la compatibilidad con el software MATLAB. Las tarjetas de adquisición que son compatibles con MATLAB y presentan un costo bajo son: la usb 6001 de la National Instrument y la placa Arduino Uno. Para la implementación

práctica del prototipo del sistema de semaforización vehicular inteligente mediante el procesamiento de imágenes se ha decidido utilizar la placa Arduino Uno, ya que se reducido precio y su compatibilidad con MATLAB son óptimos para el desarrollo del sistema.

Tabla 4.2: Comparación de las Tarjetas de Adquisición

Tarjeta	# de E/S Digitales	Compatibilidad con Matlab	Precio en \$
NI USB 6000	4	si	165
NI USB 6001	13	si	210
NI USB 6501	24	no	110
NI Arduino Mini	14	no	12
Arduino Uno	14	si	20
Arduino Leonardo	20	si	25
Arduino Mega	54	si	32
Arduino Due	54	si	25

Fuente:El Investigador

#### 4.7. Elección de la escala para el desarrollo del prototipo.

El sitio el que se realizo el modelameinto del sistema para el desarrollo del prototipo es la Av. Cevallos y Mera. La escala del prototipo se define en base a las medidas reales de diferentes automóviles de distintas marcas y modelos, también se consideran el ancho y largo de las calles y la señalización respectiva como paso cebra y división de carril.

Las medidas que se presentan en la tabla 4.3 corresponden a vehículos medianos, medidas de las calles y la señalizació, medidas estandar.

Tabla 4.3: Medidas Reales

<b>MEDIDAS REALES [cm]</b>											
<b>VEHÍCULOS MEDIANOS</b>											
Modelo	Elantra	Tucson	Mazda	Lancer	Renault	Citroen	Volvo	Passat	Accord	Aveo	Promed.
<b>Alto</b>	142.3	168.0	146.2	150.5	158.7	151.2	161.0	142.2	144.0	150.1	151.4
<b>Long.</b>	452.5	432.5	443.5	456.0	403.4	477.1	480.7	479.1	472.5	431.0	452.8
<b>Ancho</b>	172.5	183.0	175.3	176.0	170.9	177.3	178.4	185.2	184.0	170.1	177.3
<b>MEDIDAS DE LAS CALLES</b>						<b>LÍNEAS DE SEÑALIZACIÓN</b>					
<b>Ancho [cm]</b>			680			<b>Ancho [cm]</b>			15		
<b>Largo [cm]</b>			2200			<b>Largo [cm]</b>			180		

Fuente: El investigador

Tomando como base las medidas reales de los vehiculos a utilizar en prototipo, se estableció una escala de 1:64, esta escala tiene relación directa con la medida vehículos medianos de tamaño real. En la tabla 4.4, se presentan las medidas para la elaboración del prototipo, en base a la escala establecida anteriormente.

Tabla 4.4: Medidas en escala 1:64

<b>MEDIDAS EN ESCALA 1:64 [cm]</b>											
<b>VEHÍCULOS</b>											
<b>Modelo</b>	Elantra	Tucson	Mazda	Lancer	Renault	Citroen	Volvo	Passat	Accord	Aveo	Promed.
<b>Alto</b>	1.6	2.2	2.8	2.6	2.4	2.0	2.1	2.0	2.6	2.6	2.3
<b>Long.</b>	7.0	6.8	7.4	7.4	7.3	7.0	7.4	7.5	7.0	6.5	7.1
<b>Ancho</b>	2.7	2.7	2.9	2.8	2.8	3.0	3.0	2.6	2.9	3.0	2.8

<b>MEDIDAS DE LAS CALLES</b>		<b>LÍNEAS DE SEÑALIZACIÓN</b>	
<b>Ancho [cm]</b>	10.6	<b>Ancho [cm]</b>	0.25
<b>Largo [cm]</b>	35	<b>Largo [cm]</b>	2.8

Fuente: El investigador

#### 4.8. Ubicación de las cámaras y semáforos

El sitio anteriormente modelado, cuenta con una Av. Principal (Av. Cevallos) y una calle secundaria (calle Mera), le cual fue recreado a una escala de 1:64 para la simulación del prototipo..

Para ubicar las cámaras se realizó un estudio de tráfico vehicular del sitio de interés, para cada uno de los sentidos tanto de la calle y la avenida. Este análisis se basa en los datos proporcionados por la UMT de Ambato, aforos realizados entre la primera semana de junio del 2014. Los resultado del flujo vehicular se muestra en la figura 4.3.

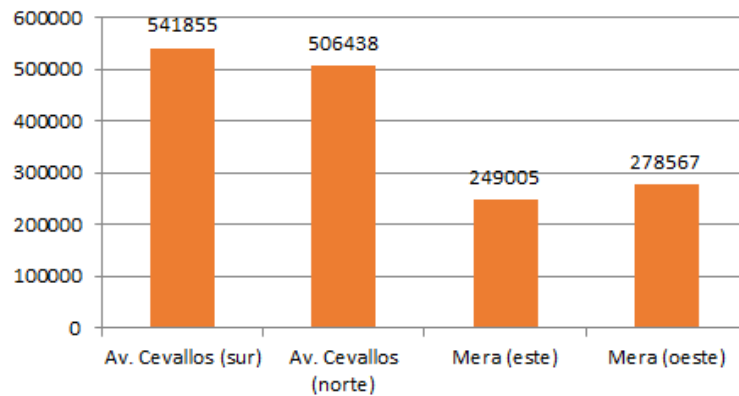


Figura 4.3: Afluencia vehicular en la Av. Cevallos y la calle Mera  
 Fuente: Realizado por Investigador de acuerdo a los datos de UMT Ambato

Según los resultados la mayor afluencia vehicular se genera: en la Av. Cevallos en sentido Norte a Sur y en la calle Mera en el sentido Este a Oeste. En la figura 4.4 se ilustra a ubicación de las cámaras y de los semáforos.

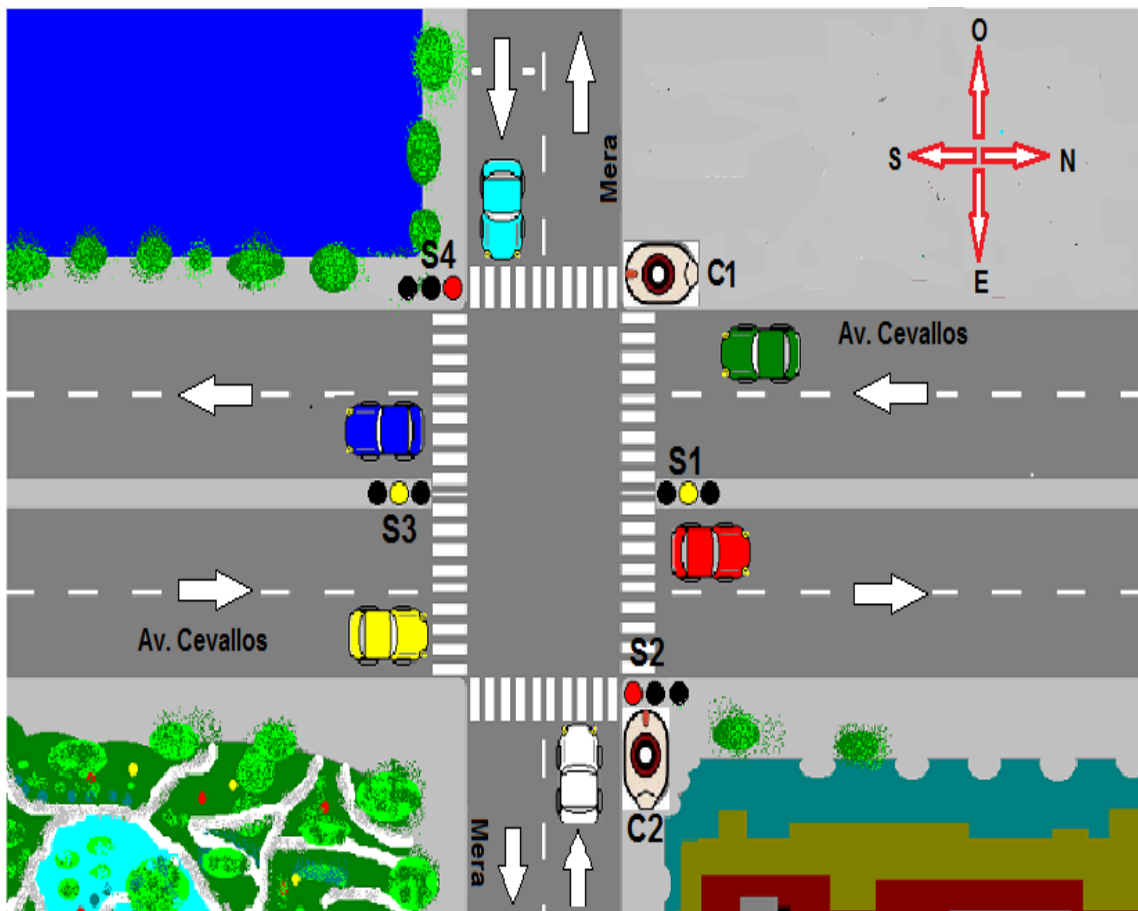


Figura 4.4: Ubicación de semáforos y cámaras  
 Fuente: El Investigador

Donde:

C1, es la cámara que captura los vehículos en la Av. Cevallos en sentido Norte a Sur.

C2, es la cámara que captura los vehículos en la calle Mera en sentido Este a Oeste.

S1 y S3, controlan el flujo vehicular en la Av. Cevallos hacia el Sur y hacia el Norte, respectivamente.

S2 y S4, controlan el flujo vehicular en la calle Mera hacia el Este y hacia el Oeste, respectivamente.

#### 4.9. Elección de la cámara

Para determinar las características de la cámara a utilizar se debe tomar en cuenta la distancia focal, ya que de ella depende el ángulo de visión de la cámara, esta relación se observa en la figura 4.5.

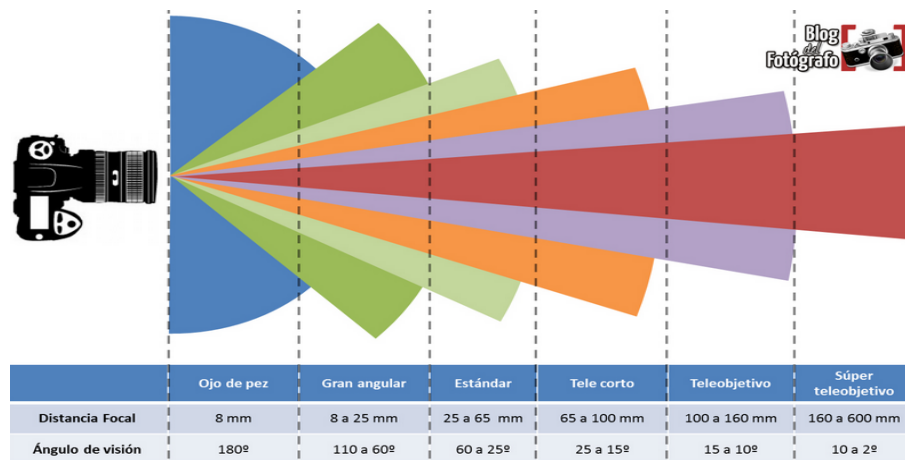


Figura 4.5: Relación distancia focal y ángulo de visión  
Fuente:<http://www.blogdelfotografo.com/distancia-focal/>

Se relaciona la altura de la cámara promedio de los báculos instalados en la ciudad las cuales de 14m, en escala esto representa 23cm. Entonces se hace uso de la figura 4.6 para determinar el ángulo de visión de la cámara.

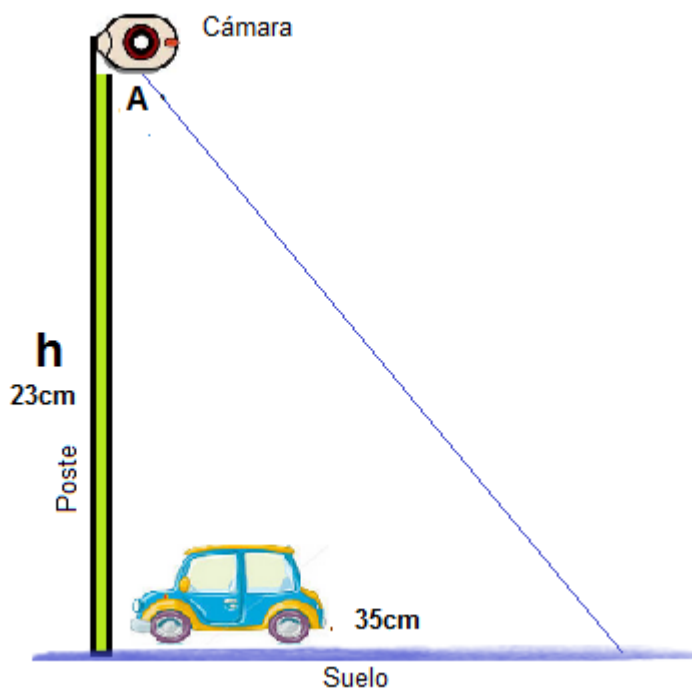


Figura 4.6: Ubicación de la cámara  
Fuente: El Investigador

Donde:

A, es el ángulo de visión.

23cm, es la altura a la que se debe colocar la cámara.

35cm, es el área que se desea cubrir

Entonces para determinar el ángulo de visión necesario se aplica el teorema de pitágoras, como sigue en la ecuación 4.1.

$$tg(A) = \frac{35}{23} \quad (4.1)$$

$$A = tg^{-1}\left(\frac{35}{23}\right) = 57^\circ$$

El rango de la distancia focal que cubre este ángulo es de 25 a 65mm. En el cuadro 4.5 se muestra distintas cámaras con sus características y precios.



Tabla 4.5: Webcam con características y precios

Descripción	Características	Precio en \$
Logitech HD Webcam C270	Captura de vídeo: Hasta 1280 x 720 píxeles Distancia focal: 7mm	45
Logitech HD Webcam C525	Captura de vídeo HD: Hasta 1280 x 720 píxeles Distancia focal: 7mm	60
WebCam Genius FACECAM 321	Captura de vídeo HD: 1280 x 720 píxeles Distancia focal: 10mm	77
MODECOM +88075	Captura de vídeo HD: 1600 x1200 píxeles Distancia focal: 30mm	40
Klip Xtreme KDC-600 Voilà cámara HD	Captura de vídeo HD: 1600 x1200 píxeles Distancia focal: 30mm	29

Fuente: El Investigador

Las cámaras óptimas para la aplicación son la Modecom y la KlipXtreme, las que tienen similares características en cuanto a distancia focal y a resolución se refiere. Para el desarrollo del prototipo se optó por la cámara Klip Xtreme, ya que es la más económica y presenta un distancio focal de 30 mm, óptima para un ángulo de visión de 57°.

**Características de la cámara KLIP** La webcam seleccionada es la klip xtreme kdc-600, se ilustra en la figura 4.7.



Figura 4.7: Cámara KLIP Xtreme

Fuente: [http://cnx.org/contents/Webcam\\_Color\\_Tracking\\_in\\_Matlab](http://cnx.org/contents/Webcam_Color_Tracking_in_Matlab)

Las características principales de esta cámara son:

Resolución: 1600 x1200 píxeles  
Captura de imágenes: Hasta 30 fps (cuadros por segundo)  
Distancia focal: 30mm  
Formato de imagen: RGB24 y YUY2  
Interfaz: USB 2.0  
Conguración: Detección automática Plug and play  
Longitud del cable: 1,5 m  
Requisitos del sistema: Familia Windows o Mac OS

#### **4.10. Descripción del Sistema**

El sistema de semaforización inteligente mediante el procesamiento digital de imágenes tiene como objetivo principal controlar el tiempo de las fases de los semáforos en dependencia al número de vehículos existentes en la vía. El sistema está formado tres fases fundamentales: adquisición -las cámaras-, el procesamiento -el ordenador- y la aplicación -placa de control (Tarjeta Arduino), los semáforos y la interfaz del sistema-.

El sistema se pone en funcionamiento cuando se presiona un el botón de Ejecutar en la interfaz, entonces las cámaras se encienden y se capturan las imágenes. Una vez adquirida las imágenes se realiza el procesamiento de las mismas con el fin de detectar el número de vehículos en ambas calles, entonces se toma decisiones y se genera los tiempos de fases para cada uno de los semáforos, esto se realiza mediante programación en el editor de MATLAB.

En la interfaz se muestra el video en tiempo real de las calles, el número de vehículos detectados y el tiempo estimado en verde para la calle y la avenida.

En la figura 4.8 se muestra el diagrama de bloques del funcionamiento del sistema de semaforización inteligente mediante el procesamiento de imágenes. En el diagrama se puede apreciar las tres etapas fundamentales de un sistema de procesamiento de imágenes.

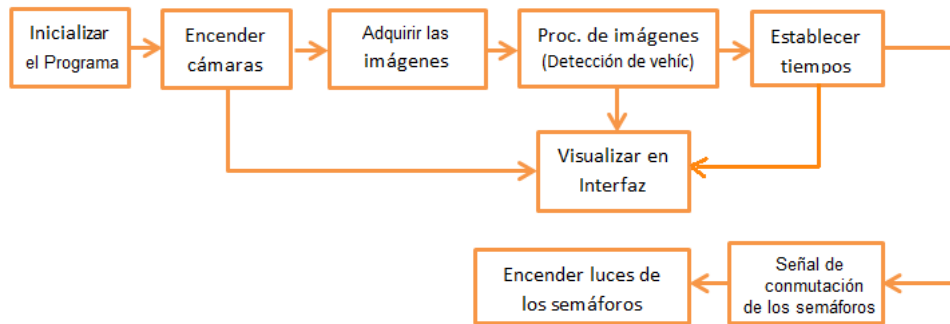


Figura 4.8: Diagrama de bloques del sistema  
Fuente:El Investigador

#### 4.11. Procesamiento Digital de las imágenes para la detección de vehículos

Para lograr la detección correcta de los vehículos fue necesario realizar el siguiente proceso:

- Adquisición de imágenes: La imágenes se adquieren como matrices de  $N \times M$ . La imagen adquirida es a color por lo que se forma de tres matrices bidimensionales de los planos R, G y B, como se muestra en la figura 4.9.

$$I_R(m, n, 1) \begin{bmatrix} r_{11} & \dots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \dots & r_{mn} \end{bmatrix}; I_G(m, n, 2) \begin{bmatrix} g_{11} & \dots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{m1} & \dots & g_{mn} \end{bmatrix}; I_B(m, n, 3) \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mn} \end{bmatrix}$$

Figura 4.9: Matrices de una imagen a color  
Fuente: [http://cnx.org/contents/Webcam\\_Color\\_Tracking\\_in\\_Matlab](http://cnx.org/contents/Webcam_Color_Tracking_in_Matlab)

- Mejora del contraste: El contraste se define como la relación existente entre los diferentes valores de intensidad en una imagen. Matemáticamente se expresa con la ecuación:

$$R(x, y) = I(x, y) \pm k \quad (4.2)$$

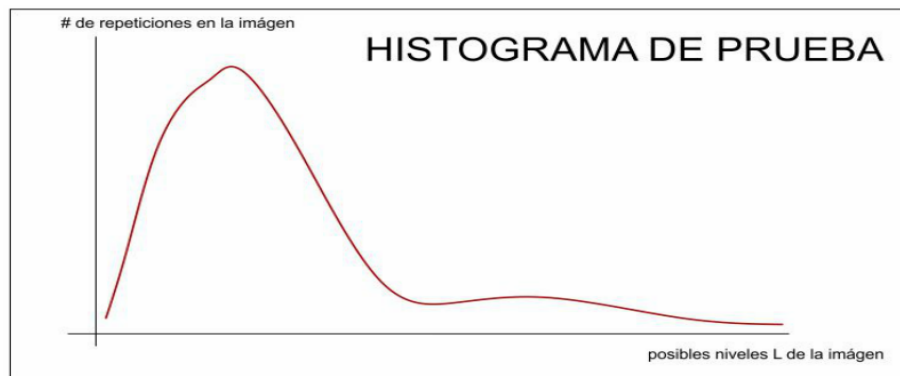
Donde:

$R$ = Matriz de la imagen resultante

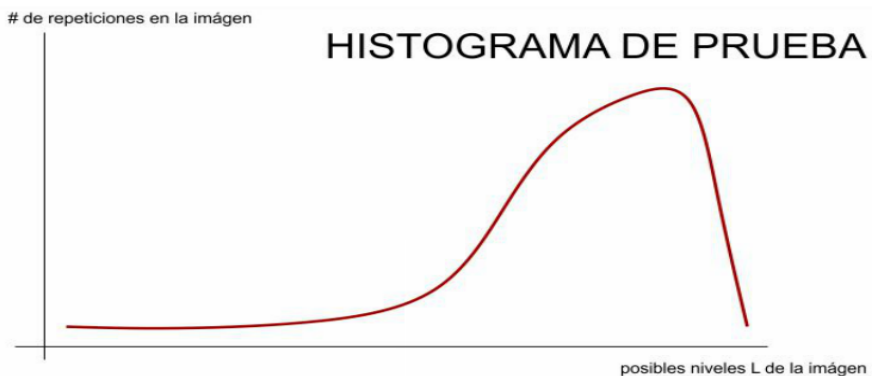
$I$ = Matriz de la imagen original

$k$ =Constante

Cabe mencionar que  $k$  es positivo se se tiene una imagen oscura con un histograma como se muestra en a figura 4.10a y es negativa si se tiene una imagen clara correspondiente al histograma de la figura 4.10b.



(a) Histograma de una imagen con bajo contraste



(b) Histograma imagen de alto contraste

Figura 4.10: Ejemplos de histogramas

Fuente:Clase de Procesamiento de Imágenes, Julian Figueroa

El valor de constante  $k$ , debe ser fijado hasta obtener un histograma parecida al de la figura 4.11, en donde las distribuciones de intensidad de los pixeles son homogeneas.

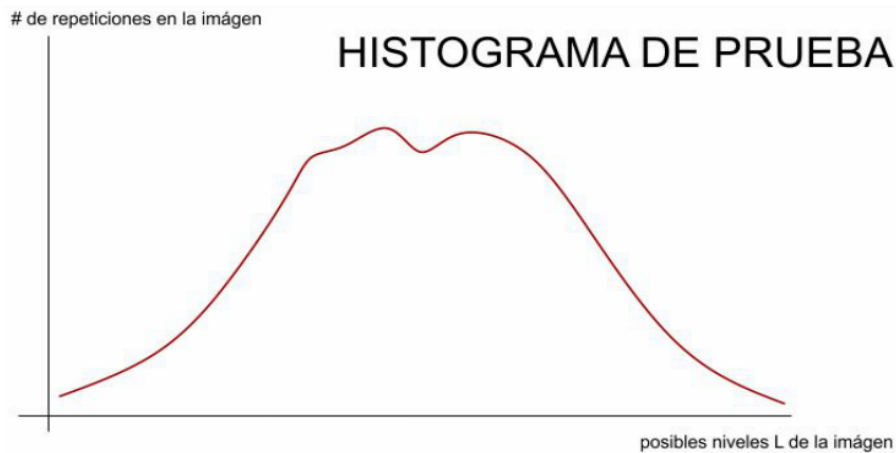


Figura 4.11: Histograma de una imagen con un contraste homogéneo  
Fuente: Clase de Procesamiento de Imágenes, Julian Figueroa

- Imagen RGB a Binaria: Esta transformación consiste en un valor denominado umbral para el cual los píxeles con una intensidad menor a este tomarán un valor de cero y los mayores a este umbral uno, como se expresa en la ecuación.

$$B(p) = \begin{cases} 0 & \text{si } p < u; \\ 1 & \text{si } p \geq u; \end{cases} \quad (4.3)$$

Donde:

$p$ , es el píxel de la imagen

$u$ , es el umbral determinado por el usuario.

Se debe considerar que esta operación se repite por cada píxel de la imagen.

- Rellenar Agujeros: Se realiza la operación morfológica denominada cierre, esta consiste en aplicar sobre la imagen la dilatación seguida de la erosión.

Erosión:  $A \ominus B$

Dilatación:  $A \oplus B$

Donde:  $A$  es la matriz de la imagen y  $B$  es la máscara (ver apartado 2.2.4)

Se consigue el relleno de agujeros y la definición de contornos.

- Segmentación con Método de Canny.

Se elige el método de Canny ya que es el más robusto y rápido de los métodos para detección de bordes. El método de Canny consiste en dos pasos:

Suavizar la imagen, se consigue con la aplicación de la siguiente ecuación.

$$I = A * B$$

Donde:

I, es la matriz de la imagen resultante.

A, es la matriz de la imagen original.

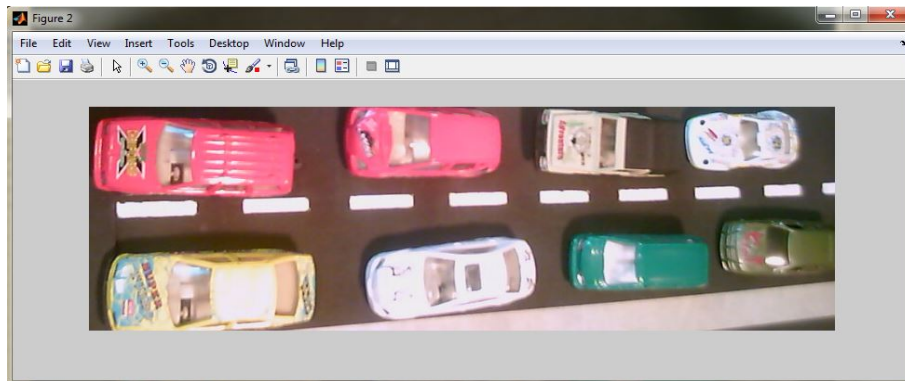
B, es la máscara.

Cálculo de la gradiente que se obtiene como un vector bidimensional.

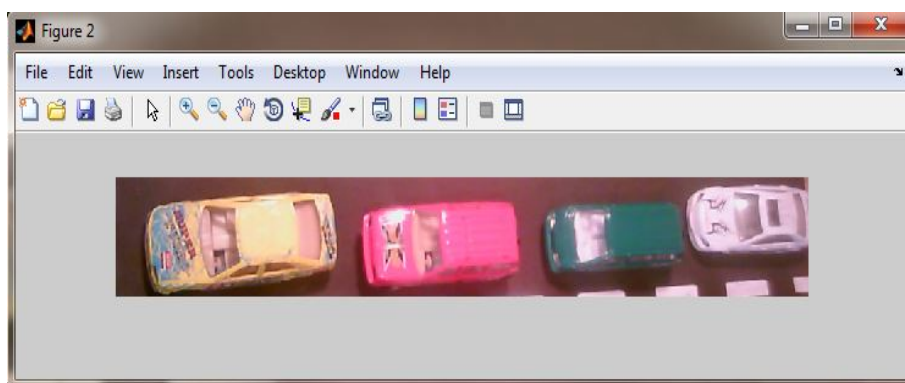
$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta}{\delta x} f(x, y) \\ \frac{\delta}{\delta y} f(x, y) \end{bmatrix}$$

#### 4.12. Resultados del Procesamiento de las imágenes

- Adquisición de las imágenes: En la figura 4.12a y 4.12b, se muestra las imágenes adquiridas.



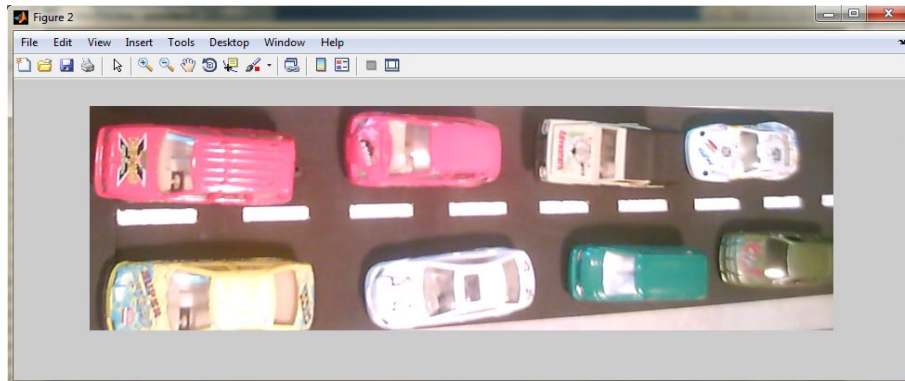
(a)



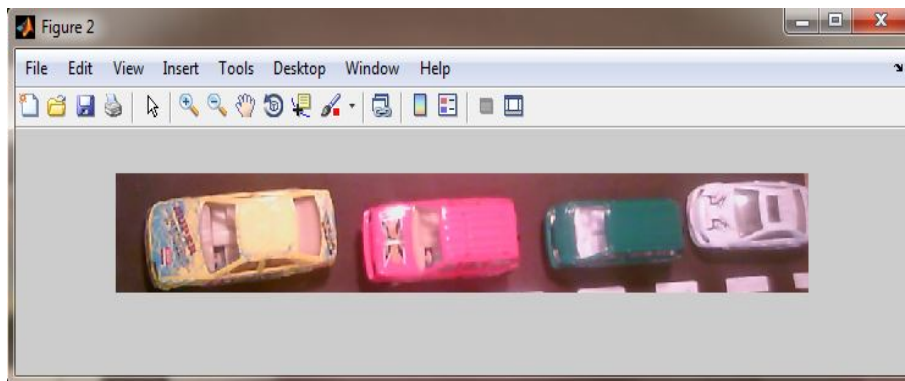
(b)

Figura 4.12: a) Imagen RGB de la cámara uno b) Imagen RGB de la cámara dos  
Fuente:El investigador

- Mejorar el contraste: En la figura 4.13a y 4.13b, se muestra las imágenes con un mayor contraste.



(a)

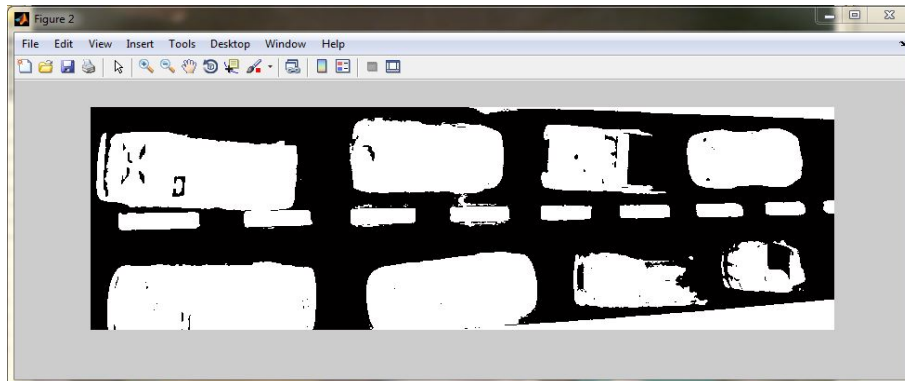


(b)

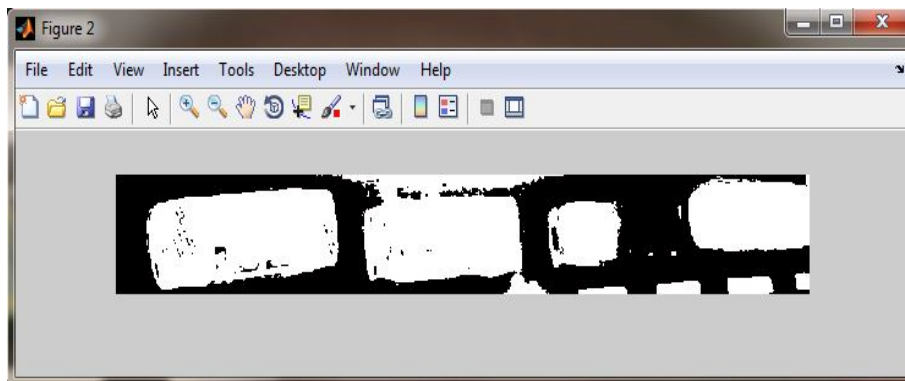
Figura 4.13: a) Imagen de la cámara uno con mayor contraste b) Imagen de la cámara dos con mayor contraste

Fuente:El investigador

- Transformación de RGB a binario: En la figura 4.14a y 4.4b se muestra las imágenes en binario.



(a)



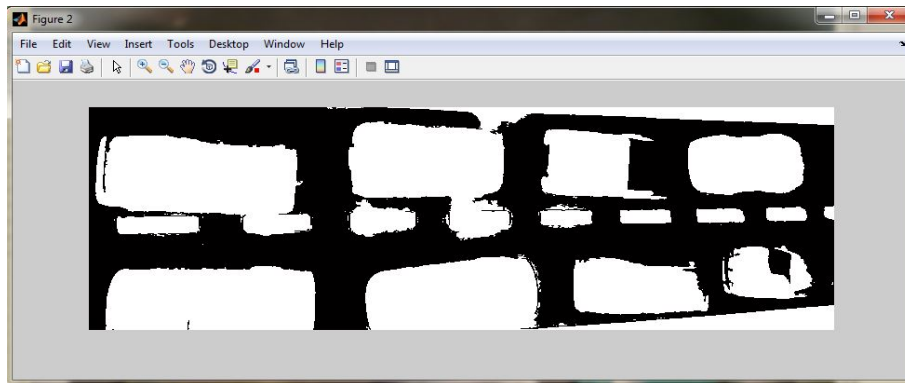
(b)

Figura 4.14: a) Imagen binaria de la cámara uno b) Imagen binaria de la cámara dos

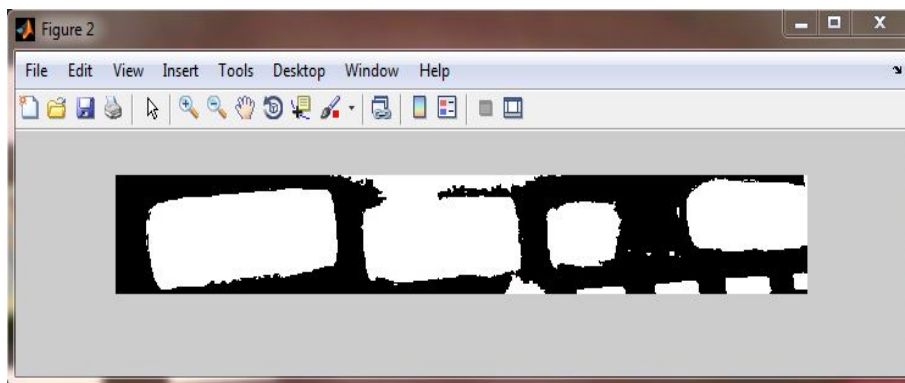
Fuente: El investigador



- Rellenar Agujeros: En la figura 4.15a y 4.15, se muestran las imágenes imágenes más definidas.



(a)

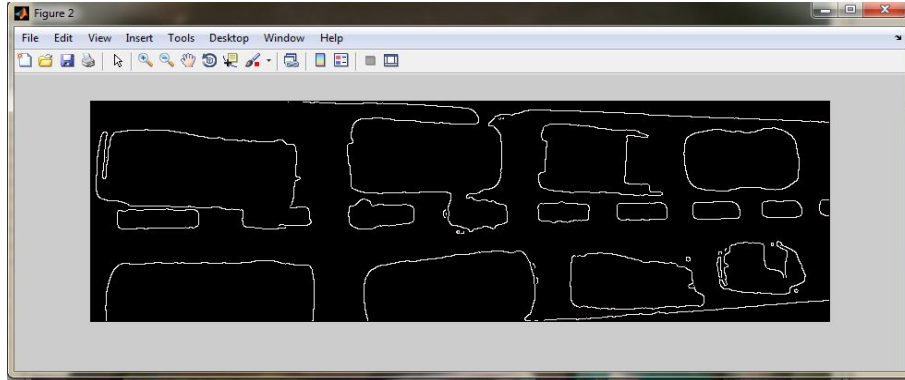


(b)

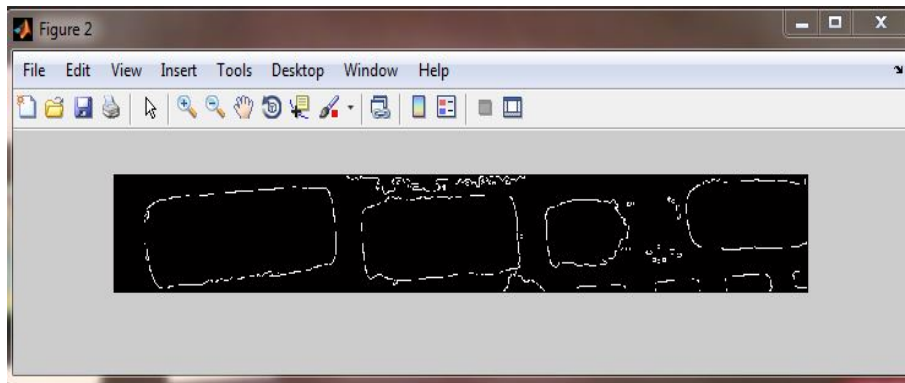
Figura 4.15: a) Operación cierre de la cámara uno b) Operación cierre de la cámara dos

Fuente:El investigador

- Detección de bordes:



(a)

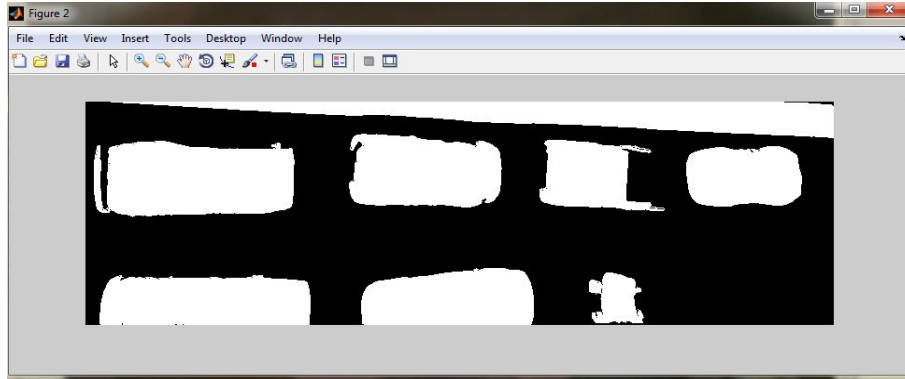


(b)

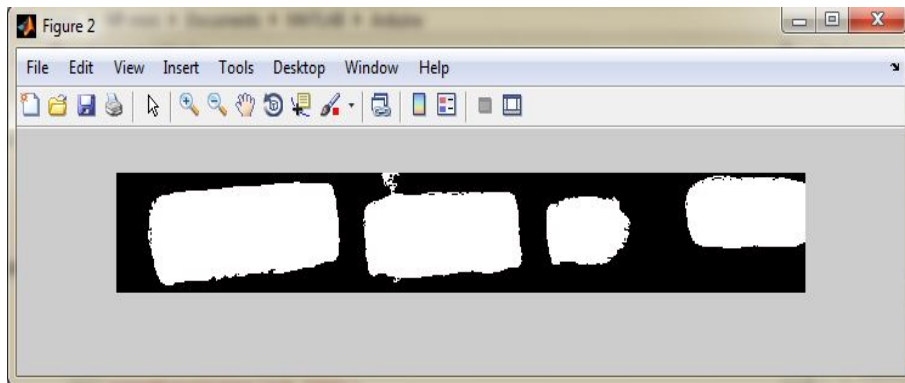
Figura 4.16: a) Detección de bordes de la imagen uno b) Detección de bordes de la imagen dos

Fuente:El investigador

- Discriminación por áreas:



(a)



(b)

Figura 4.17: a) Detección de bordes de la imagen uno b) Detección de bordes del la imagen dos

Fuente:El investigador

#### 4.13. Descripción lógica para la duración de las fases de los semáforos

El ciclo de las fases de los semáforos se presenta en la tabla 4.6, este se compone de 4 fases en la calle principal y en la calle secundaria.

- FASE1: Permite el paso vehicular en la calle principal (luz verde) y lo impide en la calle secundaria (luz roja).
- FASE2: Enciende la luz Ambar como anuncio a cambio a luz roja en la calle principal y mantiene encendida la luzroja como anuncio a cambio a luz verde en la calle secundaria.

- FASE3: Impide la circulación vehicular (luz roja) en la calle principal y permite la circulación en la calle secundaria (luz verde). Realiza el conteo de vehículos en la calle secundaria.
- FASE4: Mantiene encendida la luz roja en la calle principal y enciende la luz ambar en la calle secundaria.

Tabla 4.6: Fases de los semáforos

SEMÁFORO	FASE1	FASE2	FASE3	FASE4
<b>Calle Principal</b>				
<b>Calle Secundaria</b>				
Tiempo de duración	tv1	5seg	tv2	5seg

Fuente: El Investigador

En la tabla 4.7a se muestra la designación de tiempo de la FASE1 y en la tabla 4.7b la de la FASE3.

N1 y N2, son el número de vehículos existentes en la vía principal y secundaria, respectivamente.

La FASE1 y FASE3 tienen una duración de 5seg, es el tiempo requerido para ejecutar la programación.

Tabla 4.7: Asignación de tiempos

(a) Distribución de tiempo en la calle principal

N1	tv1(s)
0	10
1-2	30
3-4	50
5-6	70
7-8	90

(b) Distribución de tiempo en la calle secundaria

N2	tv2(s)
0	0
1	15
2	30
3	45
4	60
Ec. Final	$tv2=10*N2$

Fuente: El Investigador

Hay que tener en cuenta que cuando no existen vehículos en la calle secundaria la FASE1 se mantiene. En la figura 4.18 se muestra el diagrama de flujo para la coordinación de las fases de los semáforos.

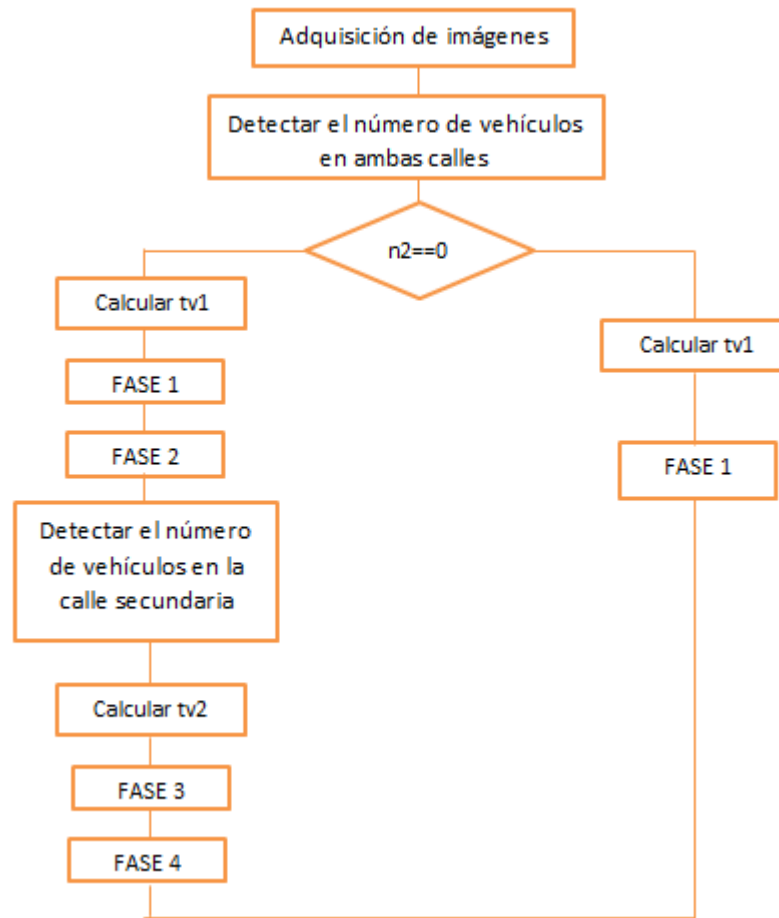


Figura 4.18: Diagrama de Flujo de las Fases de los semáforos  
Fuente: El Investigador

#### 4.14. Diseño de la Interfaz

La interfaz se realizó en GUI es un entorno de programación visual disponible en Matlab. Para iniciar la programación en Guide deben seguirse los siguientes pasos:

Para iniciar GUI, se debe ejecutar la siguiente instrucción en Matlab:

>>guide

Se presenta una ventana coo se muestra en la figura 4.19

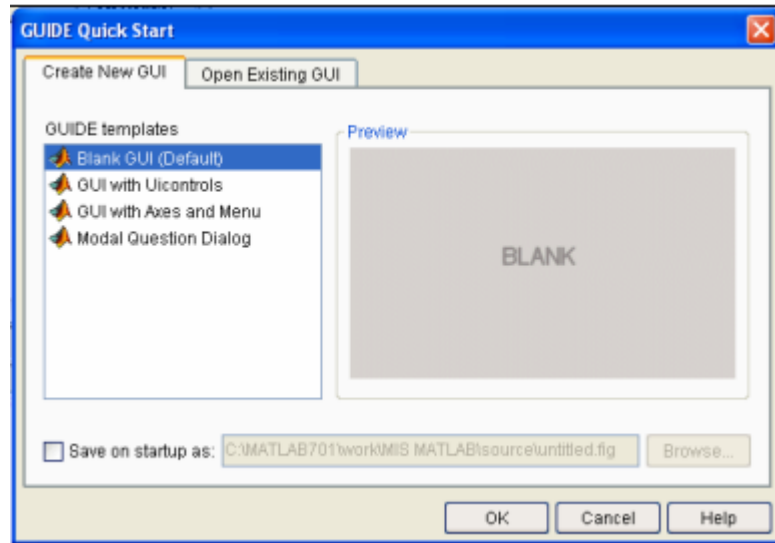


Figura 4.19: Ventana de inicio

Fuente: Manual de Interfaz Gráfica de Usuario, Barragán Diego.

Se elige la primera opción “Blank GUI”, como se ilustra en la figura 4.20.

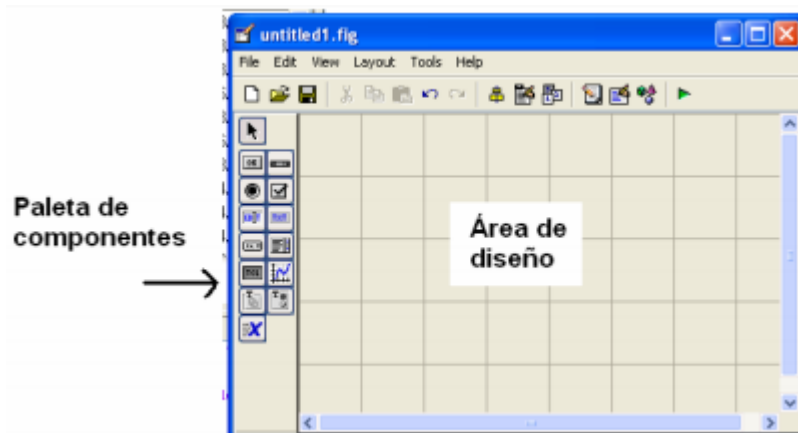


Figura 4.20: Entorno de Diseño

Fuente: Manual de Interfaz Gráfica de Usuario, Barragán Diego.

Los elementos a usarse en la interfase son: 2 Axes, 2 botones y cuatro cuadros de texto, como se muestra en la figura 4.21

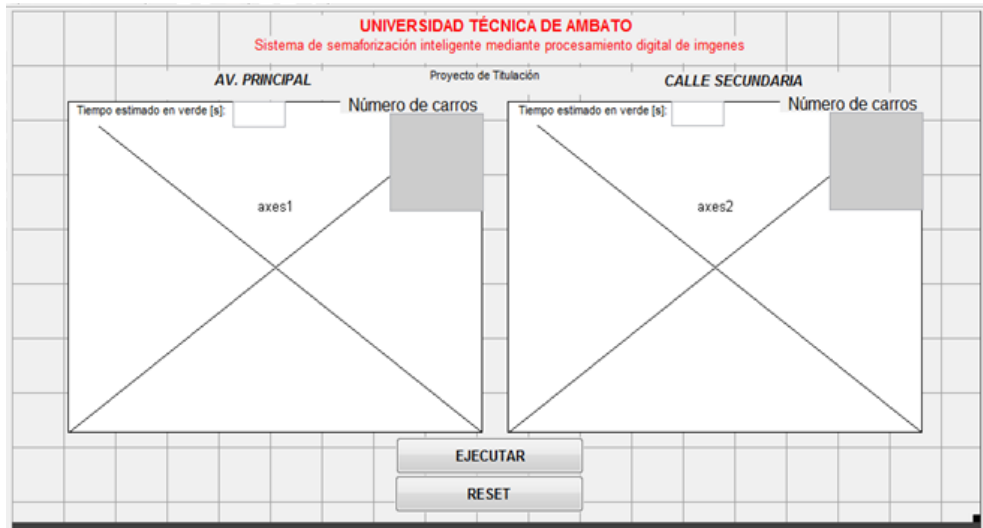


Figura 4.21: Entorno de Diseño  
Fuente: El Investigador.

La interfaz completa se encuentra dispuesta de la siguiente manera:

El botón Ejecutar pone en marcha al sistema y el botón Resetear pone en cero al sistema listo para ser ejecutado nuevamente.

En los axes se muestra la captura de la calle principal y la secundaria.

Dos cuadros de texto muestran el número de vehículos detectados en cada calle.

Dos cuadros de texto muestran el tiempo en verde correspondiente para cada una de las calles.

Se observa la interfaz en funcionamiento en la figura 4.22



Figura 4.22: Interfaz en funcionamiento  
Fuente:El Investigador



#### 4.15. Conexiones del Sistema

Las conexiones se analizarán por fases:

- Adquisición: Se realiza la conexión de ambas cámaras para la captura de las calles de interés como se muestra en la figura 4.23.

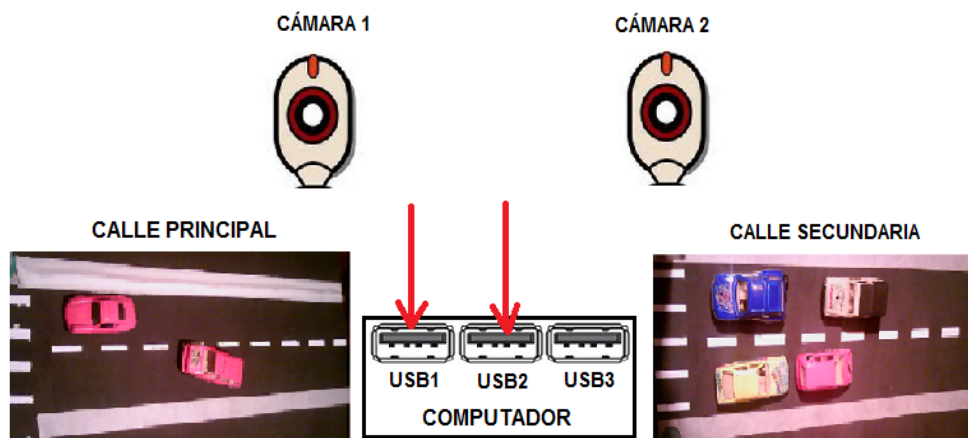


Figura 4.23: Entorno de Diseño  
Fuente: El Investigador

- Aplicación: La conexión del Arduino y de las placas de los semáforos se muestra en la figura 4.24.

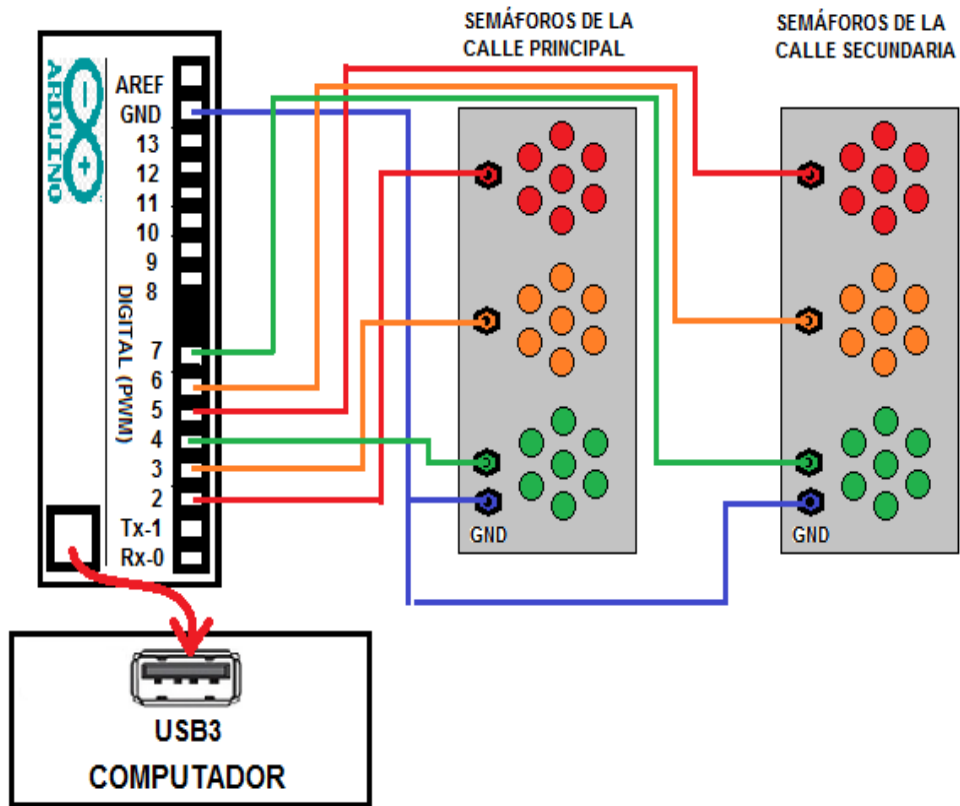


Figura 4.24: Conexiones para la aplicación  
Fuente: El Investigador

#### 4.16. Instalación de las cámaras

Para la adquisición de las imágenes se utilizó dos cámaras de las mismas características, estas fueron descritas en el apartado 4.6.2. La captura de la imágenes se realizan a través del software Matlab R2012b, para que las cámaras sean instaladas correctamente se deben ejecutar los siguientes comandos.

En la interfaz de matlab se ingresa el comando (`imaqhwinfo`) como se muestra en la figura 4.25, mediante este se conoce que adaptadores se encuentran instalados en Matlab R2012b.

```
Command Window
>> camaras=imacqhwininfo

camaras =

    InstalledAdaptors: {'gentl' 'gige' 'matrox' 'winvideo'}
    MATLABVersion: '8.0 (R2012b)'
    ToolboxName: 'Image Acquisition Toolbox'
    ToolboxVersion: '4.4 (R2012b)'
```

Figura 4.25: Información de los Adaptadores Instalados  
Fuente: El Investigador

El adaptador que utiliza matlab para adquisición de imágenes es 'winvideo'. En la figura 4.26 se observa que al ejecutar este comando se despliega una serie de información una de ellas se describe como DevicesIDs esta indica cuantos dispositivos de captura se encuentran instalados en el computador, en este caso se observan 3, las dos cámaras a utilizar y la cámara incorporada de fábrica al computador.

```
Command Window
>> camaras=imacqhwininfo('winvideo')

camaras =

    AdaptorDllName: [1x81 char]
    AdaptorDllVersion: '4.4 (R2012b)'
    AdaptorName: 'winvideo'
    DeviceIDs: {[1] [2] [3]}
    DeviceInfo: [1x3 struct]
```

Figura 4.26: Información de los ID de los dispositivos

Los IDs pertenecientes a las cámaras utilizadas para la aplicación son: [2] y [3], para obtener más información de los dispositivos se ejecuta el comando como se muestra en la figura 4.27.

```
Command Window
>> camara1=imqhwinfo('winvideo',2)

camara1 =

    DefaultFormat: 'MJPG_1024x576'
    DeviceFileSupported: 0
    DeviceName: 'HD 720P Webcam'
    DeviceID: 2
    VideoInputConstructor: 'videoinput('winvideo', 2)'
    VideoDeviceConstructor: 'imaq.VideoDevice('winvideo', 2)'
    SupportedFormats: {1x36 cell}
```

Figura 4.27: Información de los Adaptadores Específico  
Fuente: El Investigador

#### 4.17. Establecer la Comunicación entre Matlab y Arduino

Para establecer la comunicación entre Matlab y la placa Arduino Uno se deben seguir los siguientes pasos:

1. Conectar la placa Arduino mediante el cable usb.Figura 4.28



Figura 4.28: Conexión del Arduino a la PC  
Fuente: El Investigador

- 2.Verificar que se encuentre seleccionada la tarjeta correcta como se muestra en la figura 4.29

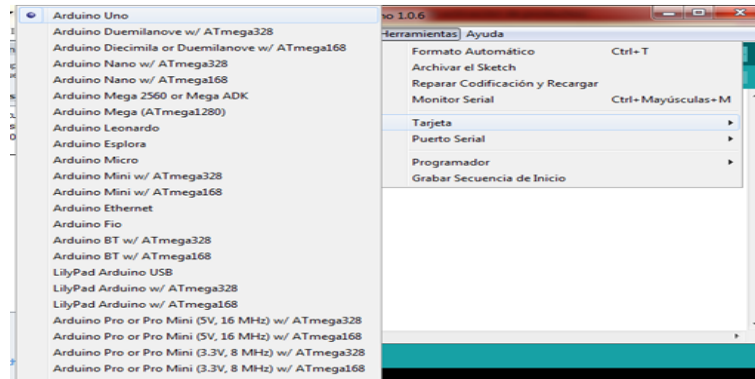


Figura 4.29: Elección de la placa Arduino  
Fuente: El Investigador

3. Identificar el puerto al que se encuentra conectado el Arduino, se muestra en la figura 4.30

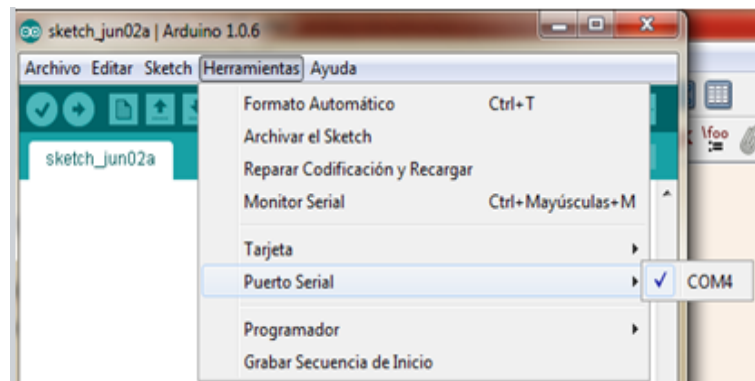


Figura 4.30: Información sobre el puerto COM  
Fuente: El Investigador

4. Se carga un sketch que permite realizar la comunicación. Dentro del Arduino: Archivo > Abrir. Buscar dentro de los archivos descargados en la siguiente dirección: ArduinoIO/pde/adiosrv/adiosrv.pde y se carga el sketch en el Arduino.

5. Se inicia Matlab se ejecuta el comando como se muestra en la figura 4.31.

```
>> install_arduino
Arduino folders added to the path
Saved updated MATLAB path
```

Figura 4.31: Instalar Arduino  
Fuente: El Investigador

6. Se asigna el COM en el que se encuentra instalada la placa Arduino, como se muestra en la figura 4.32. En este caso es 4.

```
>> f=arduino('COM4')
Attempting connection .....
```

Figura 4.32: Asignar variable y puerto COM  
Fuente: El Investigador

Se establece la conexión correctamente.

Nota: El proceso anteriormente detallado se realiza una sola vez.

#### 4.18. Presupuesto del Prototipo

En la tabla 4.8 se presenta el número de materiales con su precio individual y colectivo, haciendo posible plantear el presupuesto necesario para el desarrollo del prototipo del Sistema de semaforización Inteligente mediante el Procesamiento de imágenes.

Tabla 4.8: Presupuesto del Prototipo

Descripción	Cantidad	Precio Unidad en \$	Precio Total en \$
HP Mini 110-4100LA	1	500	500
Webcam Klip Xtreme	2	30	60
Arduino Uno	1	35	35
Placas de semáforos	4	5	20
Led	84	0.10	8.4
Lámpara de escritorio	1	15	15
Tubo PVC	2mts	8	8
Tabla MDF	1	20	20
Cable Multipar	3mts	1	3
<b>Total</b>			<b>669.4</b>

Fuente: Investigador

En el presupuesto anterior mente presentado no se incluye el coste del software, puesto que este fue proporcionado por la universidad y consta con una licencia educativa.

#### **4.19. Requerimientos para implementación real.**

Una vez analizados los materiales del prototipo y presentada la escala con la que se realizó el mismo, se procedió a realizar un presupuesto estimativo de la implementación real del sistema de semaforización inteligente. El Presupuesto real se realizó tomando en cuenta que las calles que recreadas son la Avenida Cevallos y Mera las cuales se encuentran a 350 metros aproximadamente del Centro de Control de la Unidad Municipalidad de Tránsito de Ambato, ubicado en la calle Simón Bolívar entre Castillo y Juan Montalvo. Los materiales seleccionados con sus respectivos costos, responden a materiales de bajos costos y altas prestaciones. El costo de los materiales ha sido calculado mediante el promedio de diferentes distribuidores tanto nacionales como internacionales.

#### 4.20. Presupuesto Real

Tabla 4.9: Presupuesto del Prototipo

Descripción	Cantidad	Precio Unidad en \$	Precio Total en \$
Cámara IP de vigilancia 25mm HD con transductor incluido	2	1567	3134
Transceiver	2	50	100
Arduino Uno	1	25	50
Cable multipar de 8 hilos	700m	3,10	196
Software MATLAB	1	2500	2500
Toolbox ImageProcessing	1	1250	1250
Báculo 14m	2	720	1440
Obra civil ( Instalación del báculo y canalización del cableado)	1	1440	1440
Ordenador de escritorio, con procesador de 7 núcleos	1	1115	1115
Mano de obra (Cableado, montaje de elementos, instalación y puesta en marcha del sistema)	1	1000	1000
<b>Total</b>			10785

Fuente: Investigador

Los valores de instalación, mano de obra y el báculo de alojamiento de la cámara han sido proporcionados por el Presupuesto referencial para instalación de un sistema semafórico en la ciudad de Ambato realizado por la Dirección de Tránsito y Movilidad.

#### 4.21. Pruebas de Funcionamiento del sistema de semaforización.

La figura 4.32 y 4.33 muestran que el sistema reconoce los vehículos aunque estos estén ubicados en forma diagonal.





Figura 4.33: Vehículos diagonales en la calle secundaria  
Fuente: El Investigador



Figura 4.34: Vehículos diagonales en ambas calles  
Fuente: El Investigador

En la figura 4.35y 4.36, se muestran las vías llenas.



Figura 4.35: Vía Principal llena de vehículos  
Fuente: El Investigador



Figura 4.36: Calle Secundaria llena de vehículos  
Fuente: El Investigador

En la figura 4.37, se muestra lo que ocurre al presionar el botón Reset.



Figura 4.37: Efecto al presionar el botón reset  
Fuente: El Investigador

## Capítulo 5

### Conclusiones y Recomendaciones

#### 1. Conclusiones:

- La investigación realizada sobre el tráfico vehicular en la ciudad de Ambato permite concluir que el porcentaje más alto de tráfico no responde a las horas pico ya que este varía dependiendo del día y el lugar.
- Se determinó la fórmula para la distancia focal de la cámara a utilizar en dependencia a la altura y el área focal que se desea cubrir.
- El prototipo se ve afectado directamente por la luminosidad existente. Esta es una variable que no se puede controlar ya que esta varía en el transcurso de las horas y por el clima.

#### 2. Recomendaciones:

- Modernizar el sistema semaforico actual por el sistema del prototipo para mejorar la movilidad vehicular en la ciudad de Ambato.
- Para que el sistema funcione de forma óptima la cámara debe presentar la distancia focal menor a 25mm.
- Para detectar el cambio de luminosidad se recomienda la integración de un sensor de lúmenes, cuya señal permita la elección automática del factor de contraste aplicado en el sistema.

## Bibliografía

- [1] I. Thomson, “La congestión del tránsito urbano: causas y consecuencias económicas y sociales,” Master’s thesis, Revista de la CEPAL, 2002.
- [2] S. Denig, “Proyecto de investigación independiente realizado por la economist intelligence unit,” Master’s thesis, Siemens AG, 2010.
- [3] F. Camacho and W. Albuja, “Anuario de estadísticas de transporte,” *INEC*, 2013.
- [4] A. Meléndez, “El abc del sector automotor,” in *Comparación Regional del parque automotor*, 2011.
- [5] E. Rocha, “Mejora del tráfico en un cruce regulado por semáforos, mediante un sistema basado en visión artificial,” Master’s thesis, Universidad Pontificia Comillas, 2009.
- [6] A. Santos, “Algoritmo que permite el ingreso de vehículos en una vía de alto tráfico y doble sentido, basado en procesamiento de imágenes,” Master’s thesis, Pontificia Universidad Católica del Perú, 2007.
- [7] V. Blas, “Reconocimiento inteligente de flujo vehicular basado en el procesamiento de imágenes,” Master’s thesis, Universidad Michoacana de San Nicolás de Hidalgo, 2010.
- [8] M. Martinez, “Proyecto de investigación, semáforos inteligentes,” Master’s thesis, Universidad Católica de Nuestra Señora de la Asunción, 2013.
- [9] Cal and Mayor, *Ingeniería de Tránsito, Fundamentos y Aplicaciones*. México, 2000.
- [10] *Modificación al Manual de Normas Técnicas para la Señalización, Control y Regulación del Tránsito en Vías donde se realizan Trabajos*.

- [11] B. Zhou, J. Cao, X. Zeng, and H. Wu, "Adaptive traffic light control in wireless sensor network-based intelligent transportation system," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pp. 1–5, Sept 2010.
- [12] A. Khalid, A. Jai, and F. Wajdi, "Dynamic traffic light sequence algorithm using rfid," *Computer Science*, 2008.
- [13] A. Kalea, *Introducción a los Sistemas R.F.I.D.* LASARTE.
- [14] L. Networks, *RFID: Tecnología, Aplicaciones y Perspectivas*. Whitepaper Series, 2010.
- [15] F. Pedraza and C. Hernández, "Sistema de comunicación tcp/ip para el control de una intersección de de tráfico vehicular," *Ingeniería investigación y Tecnología, volumen XIV, 04*, 2013.
- [16] K. Yousef, J. Al-Karaki, and A. M. Shatnawi, "Intelligent traffic light flow control system using wireless sensors networks," *Information Science and Engineering*, 2010.
- [17] A. Goel, S. Ray, and N. Chandra, "Intelligent traffic light system to prioritized emergency purpose vehicles based on wireless sensor network," *Computer Applications*, 2012.
- [18] Álvaro Suárez Bravo, "Análisis de métodos de procesamiento de imágenes estereoscópicas forestales," Master's thesis, Universidad Complutense de Madrid, 2009.
- [19] J. Bulnes, "El método de correlación cruzada de matrices y la generación de un campo de desplazamientos para pequeñas esferas," *Grupo de Mecánica Quántica, Informação Quântica e Física Aplicada*, 2013.
- [20] A. Gonzáles, F. Martínez, A. Pernía, F. Alba, and M. Castejón, "Técnicas y algoritmos básicos de visión artificial," *Universidad de Rioja*, 2006.
- [21] C. Luján, "Adquisición y procesamiento digital de imágenes para la obtención de la trayectoria de los vectores de posición del camarón y la jaiba," Master's thesis, Universidad Politécnica de Valencia, España, Septiembre 2012.
- [22] E. Cuevas and D. Zaldivar, *Procesamiento digital de imagenes usando Matlab y Simulink*. Rústica, 2010.

- [23] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. capítulo 10.
- [24] G. Gómez and E. Sucar, “Vision computacional,” *Instituto Nacional de Óptica y Electrónica*, 2010.
- [25] A. Miguel and F. Alvaro, “Implementación del detector de bordes de canny sobre redes neuronales celulares,” *España*, 2012.
- [26] J. V., “Detección de bordes mediante el algoritmo de canny,” *Universidad Nacional de Trujillo*, 2010.
- [27] Mathworks, “Image processing toolbox,” *Disponible en Línea: <http://es.mathworks.com/products/image/>*.
- [28] J. Soso, “Compilacion de opencv y primeros pasos en la visión por computadora,” 2013.
- [29] “Gnu octave,” *Disponible en Línea: <http://es.kioskea.net/download/descargar-7418-gnu-octave>*.
- [30] *Consejo Nacional de Competencias (CNC) del Ecuador*.
- [31] *Unidad Municipal de Tránsito de Ambato*.

## **Anexos y Apéndices**

## Anexo A

### Anexo A. Programación del botón Ejecutar

```
%Declaración de las variables: Camara 1 y Cámara 2
global video1
global video2
global f
% Declaración de los pines de Arduino
f=arduino('COM4');
f.pinMode(2,'output');
f.pinMode(3,'output');
f.pinMode(4,'output');
f.pinMode(5,'output');
f.pinMode(6,'output');
f.pinMode(7,'output');
handles.output=hObject;
%Visualización del video de la Camara1 y 2 en el primer eje de coordenadas en la
interfaz
axes(handles.axes1); axes(handles.axes2);
video1=videoinput('winvideo',3); video2=videoinput('winvideo',2);
imagen1=image(zeros(500,1000,3),'Parent',handles.axes1);
imagen2=image(zeros(500,1000,3),'Parent',handles.axes2);
preview(video1,imagen1); preview(video2,imagen2);
%Creación del bucle
i=0;
while (i<10)
i=i+1;
f.digitalWrite(4,1);f.digitalWrite(5,1);f.digitalWrite(2,0);f.digitalWrite(6,0);
%Captura de imagen, Cámara1 y Cámara2
i1=getsnapshot(video1); i2=getsnapshot(video2);
```



```

%Recorte de la imagen para obtener unicamente el area de interes
rec1=imcrop(i1,[200 170 900 250]);rec2=imcrop(i2,[150 0 900 100]);
%Aumento del contraste
img1=rec1+30;img2=rec2+30;
%Transformación de las imagenes rgb a binarias
bw1=im2bw(img1);bw2=im2bw(img2);
%Mejoramiento de la imagen a través del relleno de agujeros
fill1=imfill(bw1,'holes');fill2=imfill(bw2,'holes');
%Detección de bordes de la imagen mejorada
bordes1=edge(fill1,'canny',0.4);bordes2=edge(fill2,'canny',0.4);
%Combinación de las imágenes
com1=bordes1+fill1;com2=bordes2+fill2;
%Discriminación de objetos con un areas menores a 2cm
area1=bwareaopen(com1,2000);area2=bwareaopen(com2,2000);
%Creación de un vector con el número de vehículos detectados
[L1, N1]=bwlabel(area1);[L2, N2]=bwlabel(area2);
%Visualización del número de vehículos detectados en la interfaz
n1=N1-1; n2=N2;
set(handles.n1,'String',(n1));set(handles.n2,'String',(n2));
% Toma de descsiones para laduración de la fase1 en dependencia al numero de
vehículos
if(n1==0);
f.digitalWrite(4,1);f.digitalWrite(5,1);f.digitalWrite(2,0);f.digitalWrite(6,0);
set(handles.tv1,'String','(10)');
pause(10)
elseif(n1==1||n1==2)
f.digitalWrite(4,1);f.digitalWrite(5,1);f.digitalWrite(2,0);f.digitalWrite(6,0);
set(handles.tv1,'String','(30)');
pause(30)
elseif(n1==3||n1==4)
f.digitalWrite(4,1);f.digitalWrite(5,1);f.digitalWrite(2,0);f.digitalWrite(6,0);
set(handles.tv1,'String','(50)');
pause(50)
elseif(n1==5||n1==6)
f.digitalWrite(4,1);f.digitalWrite(5,1);f.digitalWrite(2,0);f.digitalWrite(6,0);
set(handles.tv1,'String','(70)');
pause(70)

```

```

else
f.digitalWrite(4,1);f.digitalWrite(5,1);f.digitalWrite(2,0);f.digitalWrite(6,0);
set(handles.tv1,'String','90');
pause(90)
end
% Cuando n2 es igual a cero la fase1 se mantiene
if n2==0
f.digitalWrite(4,1);f.digitalWrite(5,1);f.digitalWrite(2,0);f.digitalWrite(6,0);
set(handles.tv1,'String','10');
pause(10)
else
%Generación de la fase2
f.digitalWrite(4,0); f.digitalWrite(5,1);f.digitalWrite(3,1);
%Adquisicion del numero de vehiculos en la calle secundaria
i2=getsnapshot(video2);
pause(3)
%Recorte de la imagen para obtener unicamente el area de interes
rec2=imcrop(i2,[150 0 900 100]);
%Aumento del contraste img2=rec2;
%Transformación de las imagenes rgb a binarias
bw2=im2bw(img2);
%Mejoramamiento de la imagen a través del relleno de agujeros
fill2=imfill(bw2,'holes');
%Detección de bordes de la imagen mejorados
bordes2=edge(fill2,'canny',0.4);
%Combinación de las imágenes
com2=bordes2+fill2;
%Discriminación de objetos con un areas menores a 2cm area2=bwareaopen(com2,2000);
%Creación de un vector con el número de vehículos detectados
[L2, N2]=bwlabel(area2);
%Visualización del número de vehículos detectados
n2=N2;
set(handles.n2,'String',(n2));
%Cálculo de la duración de la fase3
tv2=15*n2;
%Generación de la fase3
f.digitalWrite(3,0);f.digitalWrite(5,0);f.digitalWrite(2,1);f.digitalWrite(7,1);

```

```
set(handles.tv2,'String',(tv2));
pause(tv2)
%Generación de la fase4
f.digitalWrite(2,1);f.digitalWrite(6,1);f.digitalWrite(7,0);
pause(5)
end
end
%Poner en cero todos los elementos
f.digitalWrite(2,0);f.digitalWrite(3,0);f.digitalWrite(4,0);
f.digitalWrite(5,0);f.digitalWrite(6,0);f.digitalWrite(7,0);
delete(instrfind({'Port'},{'COM4'}));set(handles.tv1,'String','');
set(handles.tv2,'String','');
set(handles.n1,'String',''); set(handles.n2,'String','');
closepreview(video1);closepreview(video2);
```

## Anexo B

### Anexo B. Programación del botón Reset

```
%Declaración de Variables
global video1
global video2
global f
% Apagar todos las luces de los semáforos
f.digitalWrite(2,0);f.digitalWrite(3,0);f.digitalWrite(4,0);
f.digitalWrite(5,0);f.digitalWrite(6,0);f.digitalWrite(7,0);
% Dejar libers los puertos
delete(instrfind({'Port'},{'COM4'}));
closepreview(video1);closepreview(video2);
%Dejar en cero los cuadros de texto
set(handles.tv1,'String','');set(handles.tv2,'String','');
set(handles.n1,'String',''); set(handles.n2,'String','');
```

## Anexo C

### Anexo C. Características Generales del Arduino Uno

Tabla C.1: Características Generales del Arduino Uno

<b>Microcontrolador</b>	ATmega328
<b>Voltaje de operación</b>	5V
<b>Voltaje de Entrada (recomendado)</b>	7-12V
<b>Voltaje de Entrada (limites)</b>	6-20V
<b>Pines Digitales I/O</b>	14( de los cuales 6 pueden ser salidas PWM)
<b>Pines Entradas Analógicas</b>	6 (A0-A5)
<b>Corriente DC por pin I/O</b>	40 mA
<b>Corriente DC por 3.3V pin</b>	50 mA
<b>SRAM</b>	2KB
<b>EEPROM</b>	1KB
<b>Velocidad de Reloj</b>	16MHz
<b>Largo</b>	68.6 mm
<b>Ancho</b>	53.4 mm
<b>Peso</b>	25 g

## Anexo D

### Anexo D. Características del Ordenador Utilizado

#### Características Generales:

- Fabricante: Hewlett-Packard
- Nombre del Producto: 110-4100LA
- Procesador: Intel Atom N2600
- Disco duro: 320 GB (5400 RPM)
- Memoria Instalada RAM: 2.00GB
- Tarjeta de red: LAN Ethernet 10/100BASE-T integrada (conector RJ-45)
- Conectividad inalámbrica: WLAN 802.11b/g/n
- Pantalla: Pantalla con retroiluminación anti reflejo de 10,1 pulgadas en diagonal (1024 x 600)
- Peso: 1.26Kg

#### Energía:

- Adaptador de CA de 40 W
- Batería de ion de litio de 3 celdas (28 Wh)

#### Puertos Externos:

- 3 USB 2.0
- 1 VGA (15 pines)
- 1 RJ -45 (LAN)

- 1 salida de auriculares
- 1 entrada de micrófono
- Tipo de sistema: Sistemas Operativo de 32bits

### **Seguridad**

- Ranura para traba Kensington MicroSaver
- Contraseña de arranque
- Acepta dispositivos de bloqueo de seguridad de otros fabricantes

## Anexo E

### Anexo E. Prototipo del Sistema

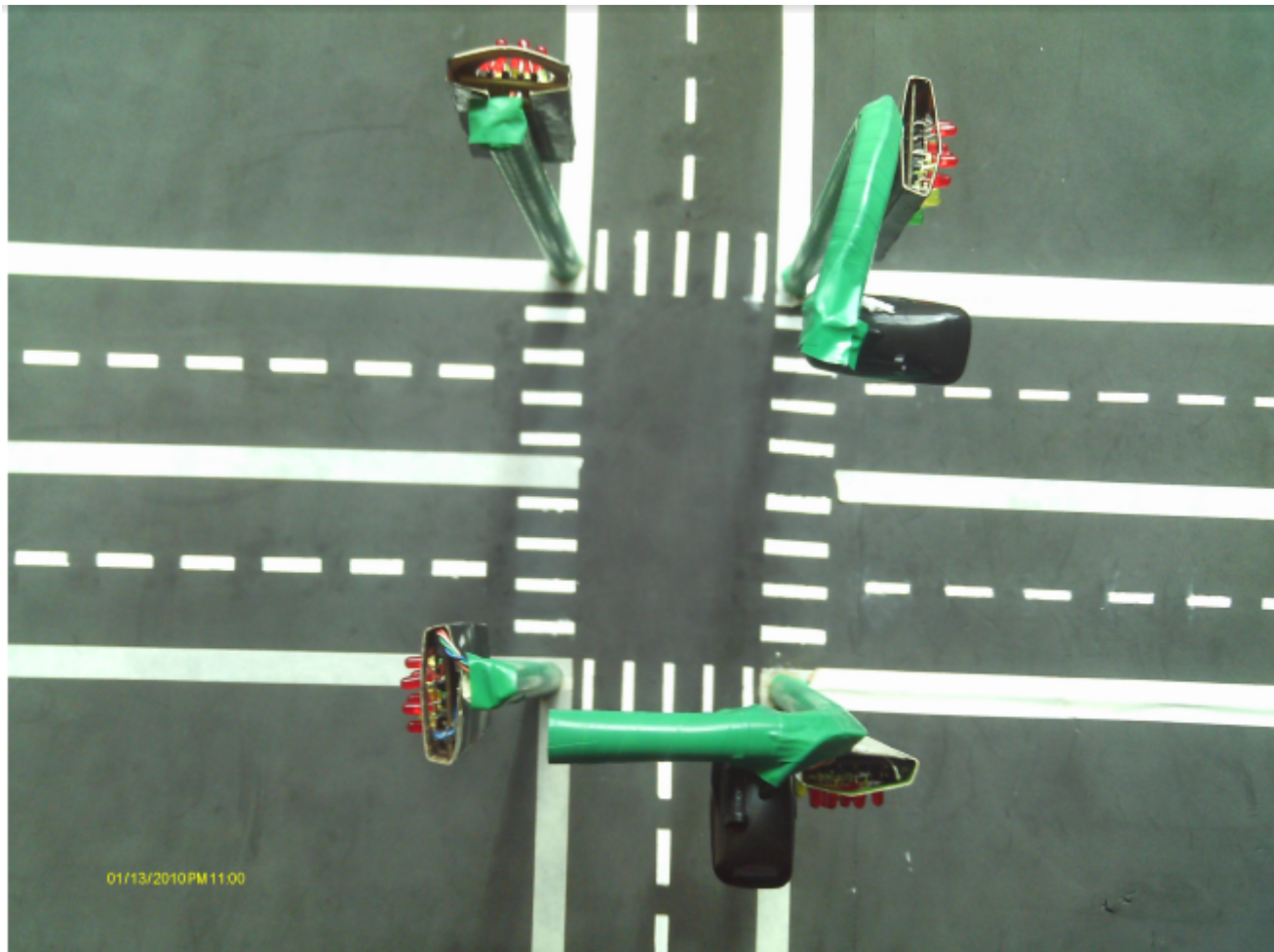


Figura E.1: Prototipo del sistema