
Prototipo para la Medición Automática de la Velocidad de un Automóvil con Cámara de Video
Mediante Procesamiento de Imágenes

Héctor Iván ARISTIZABAL GÓMEZ

Trabajo de grado para optar al título de Ingeniero Electrónico

Director
José Valentín Restrepo Laverde
Magíster en Finanzas-Ingeniero Electrónico

Universidad Pontificia Bolivariana
Escuela de Ingenierías
Facultad de Ingeniería Eléctrica y Electrónica
Programa de Ingeniería Electrónica
Medellín
2013

Dedicatoria

A mis padres Ivan Aristizabal y Helda Margarita Gómez, mis tíos Rubiela Gómez y Javier Gómez , Liborio Gómez y Alexandra López, Mauricio Gómez y Amparo Orozco, Luz Elena Gómez y Jorge Giraldo , por el gran sacrificio y apoyo durante toda mi carrera.

Agradecimiento

Agradezco a todos aquellos que de alguna u otra forma me apoyaron durante todo este proceso de formación, familia, compañeros y docentes.

Primero quiero agradecer a mis padres Iván Aristizabal y Helda Margarita Gómez y hermanos Juan Jacobo Aristizabal y Jacke Aristizabal.

También quiero agradecer a mi novia Diana Salazar por el apoyo incondicional durante el proceso de desarrollo de este trabajo de grado. Igualmente quiero agradecer a Sebastián Salazar por ayudarme en la realización de las pruebas de velocidad del trabajo.

Para finalizar quiero agradecer a mi director Valentín Restrepo por plantear este reto, su dedicación y guía durante la realización de este trabajo de grado, y a Carlos Andrés Arango por su disposición y colaboración.

Contenido

INTRODUCCIÓN	10
1. CONCEPTOS FUNDAMENTALES DE IMÁGENES Y VIDEO	12
1.1. Imagen digital.....	12
1.2. Resolución de una Imagen	12
1.3. Color de la Imagen	12
1.4. Secuencia de Imágenes (fps)	13
2. PROCESAMIENTO DE IMAGENES	14
2.1. Filtrado de Imágenes	14
2.2. Región de Interés.....	14
2.3. Características de Operaciones Morfológicas	15
2.4. Operaciones Morfológicas	15
2.5. Detección de bordes	16
2.6. Detección de Contornos	17
2.7. Transformada de Hough	17
3. DETECCION Y SEGUIMIENTO DE OBJETOS	18
3.1. Detección por color	18
3.2. Detección por sustracción de fondo	19
3.3. Seguimiento de objetos	21
3.4. Desafíos	21
3.5. Técnicas de seguimiento de objetos	21
4. SISTEMA DE MOVILIDAD EN MEDELLÍN	24
4.1. Parque Automotor	24
4.2. Velocidad promedio	24
4.3. Índices de accidentalidad.....	24
4.4. Sistema inteligente de transporte.....	24
5. ÓPTICA GEOMÉTRICA	27
5.1. Reflexión y refracción	27
5.2. Lentes	28
5.3. Calibración para iPhone4	28
6. LENGUAJE DE PROGRAMACIÓN	31
6.1. Java®	31
6.2. OpenCV	31
7. DESARROLLO DE SOFTWARE	32
7.1. Visualización	32
7.2. Configuración inicial	32
7.3. Pre Procesamiento	33
7.4. Procesamiento de Contornos	33
7.5. Detección.....	35
7.6. Seguimiento.....	36
7.7. Cálculo de velocidad	37
7.8. Mecanismo de alerta por infracción de velocidad	38
7.9. Integración.....	38
8. PRUEBAS	40
8.1. Pruebas en balcón Residencial	40
8.2. Pruebas en la Universidad	40
9. CONCLUSIONES	42
REFERENCIAS	43
AUTOR.....	44
ANEXOS	45

Lista de Figuras

Figura 1. Ejemplo región de interés.....	14
Figura 2. Concepto dilatación.....	15
Figura 3. Concepto Erosión.....	16
Figura 4. Ejemplo detección de bordes.....	17
Figura 5. Ejemplo detección de contornos.....	17
Figura 6. Ejemplo detección por color.....	18
Figura 7. Ejemplo detección fondo estático.....	19
Figura 8. Ejemplo detección fondo dinámico.....	20
Figura 9. Óptica geométrica.....	27
Figura 10. Formación imagen.....	29
Figura 11. Pruebas de calibración.....	29
Figura 12. Visualización.....	32
Figura 13. Pre Procesamiento.....	33
Figura 14. Diagrama flujo Pre Procesamiento.....	33
Figura 15. Procesamiento Contornos.....	34
Figura 16. Diagrama flujo Pre Procesamiento.....	34
Figura 17. Detección del vehículo.....	35
Figura 18. Diagrama flujo detección.....	36
Figura 19. Seguimiento del vehículo.....	36
Figura 20. Diagrama de flujo seguimiento.....	37
Figura 21. Diagrama de flujo final.....	39
Figura 22. Pruebas de velocidad 50 km/h.....	40
Figura 23. Pruebas de velocidad 35 km/h.....	41

Lista de Tablas

Tabla 1. Resolución de una imagen	12
Tabla 2. Resultados de calibración	30
Tabla 3. Pruebas en balcón Residencial.....	40
Tabla 4. Pruebas velocidad Universidad.....	41

Glosario

Cámara Web: dispositivo que captura imágenes y video de baja resolución, la cual transmite la información al computador donde se encuentra conectada.

Efecto Doppler: es el aparente cambio de frecuencia de una onda causado por el movimiento relativo de la fuente respecto a su observador. La frecuencia es la inversa del periodo, o del tiempo transcurrido entre dos ondas.

Escala de grises: en los sistemas informáticos, es una función para determinar la gama de claridad de cada pixel de la imagen, desde el negro (0) hasta el blanco (255), dando una escala de 256 niveles escalonados de grises.

Frame: es una imagen particular dentro de una sucesión de imágenes que componen una animación o video.

Imagen Binaria: es una imagen digital en donde sus pixeles solo tienen 2 valores de intensidad posibles (0, 1), que corresponden a negro o blanco.

Imagen digital: es aquella que está definida por medio de códigos informáticos que pueden ser interpretados por el computador con los programas adecuados y permite en cualquier momento modificar los contenidos de la imagen original.

Live Streaming: Es la transmisión de datos continua en forma de audio y video en tiempo real, si necesitas descargarla para transmitirla.

OpenCV: es una librería de visión artificial desarrollada por Intel, está escrita en C y C++ que utiliza algoritmos de procesamiento de imágenes y adquisición de video.

Vehicle Tracking: término utilizado para referirse al rastreo y seguimiento de vehículos.

Resumen

El propósito de este proyecto es el desarrollo de un prototipo para un dispositivo que permita medir la velocidad de un vehículo en tiempo real por medio del análisis de imágenes capturadas usando una cámara de video.

Para lograr esto se aplica un algoritmo de sustracción de fondo y detección de contornos, para detectar el vehículo; A continuación se crea una plantilla del vehículo, para que su detección a lo largo del video sea eficiente. Por medio de análisis geométrico de la posición de la cámara, podemos medir el desplazamiento del vehículo y con el cálculo del tiempo por cada *frame*, se obtiene finalmente la velocidad a la que el vehículo transita. *Copyright © UPB 2013*

Palabras Claves: OpenCV, Vehicle Tracking, cámara web, detección de objetos, seguimiento de imágenes.

Abstract

Abstract: The purpose of this project is to develop a prototype for a real time vehicle speed meter based on the analysis of webcam captured images. The vehicle is detected by a background subtraction algorithm and edge detection. After that a Template of the vehicle is created to make the tracking more efficient. Using geometrical analysis of the camera position, the distance and displacement of the vehicle can be measured, and finally with the time difference between events, the speed can be calculated.

Keywords: OpenCV, Vehicle Speed, Image processing.

INTRODUCCIÓN

Antes de comenzar a hablar acerca de lo que veremos en esta tesis, quiero que pensemos un momento sobre cuáles son los principales problemas de movilidad vehicular que tiene su ciudad?, ¿Cuáles son las causas que incrementan los índices de accidentalidad?, ¿Qué se ha hecho para mejorar esto?

El número de vehículos se ha incrementado en los últimos años en las grandes capitales del mundo, estudios como el que hace anualmente BBVA aseguran que en Colombia en el sector automotriz por cada 100 habitantes hay 7,4 carros, lo que equivaldría a tener 3,4 millones de vehículos en todo el territorio nacional, cifra que este año (2013) está alrededor de los 4,6 millones y que tiene una tendencia de crecimiento considerable para los próximos años. [1][2]

A lo anterior sumamos otros factores que incurren en una desorganización de nuestro sistema de movilidad vial, como la falta de vías alternas, el alto flujo de carros en horas pico lo cual se ha venido controlando por medio del pico y placa, y uno de los más importantes, los incidentes de tránsito, donde se hace especial énfasis en este documento, ya que la alta velocidad de las personas al conducir causa los mayores problemas de circulación en las ciudades, es así, como mejorando estándares de seguridad que se han aplicado en varias ciudades del país y desarrollando

procesos bajo nuevas tecnologías verán cómo se puede disminuir los accidentes viales.

Para el control y regulación de la velocidad existen diferentes tecnologías como los radares microondas, radares infrarrojos, de ultrasonidos, de captador magnético, y de lazos inductivos, sin embargo en Colombia adquirir tecnología de punta resulta muy costoso, y no se tiene una infraestructura claramente desarrollada para la ejecución de estas.

Medellín es la ciudad pionera en la implementación de fotomultas o comparendos electrónicos. Actualmente Sabaneta y Cali están incursionando en esta modalidad de control vehicular.

El inconveniente de este sistema es su costo, no solo la tecnología es de un precio considerable, sino que en el aprovisionamiento se deben llevar a cabo obras en las vías, ya que la técnica utiliza lazos inductivos en el asfalto para la detección de los vehículos. Son estos gastos lo hace que las demás ciudades del país tomen esta iniciativa como un proyecto a largo plazo.

En este documento conocerán una propuesta basada en desarrollo de software por medio de visión artificial, o también llamada visión por computadora, sub campo de la inteligencia artificial.

Este modelo estará al alcance de muchas ciudades, permitiendo que su implementación sea más económica, con un software más completo, que cumplirá con los estándares de medición de la

velocidad. Los procesos están desarrollados en Java ® con la ayuda de la librería OpenCV, la cual contiene diversas funciones para el procesamiento y análisis de imágenes, dando un resultado interesante que recopilará datos que va a ayudar a tener indicadores del estado de las vías.

1. CONCEPTOS FUNDAMENTALES DE IMÁGENES Y VIDEO

1.1. Imagen digital

La imagen digital es cualquier imagen fija, que se captura a través de un medio electrónico y que se representa como un archivo de información, compuesta a partir de una matriz binaria formada por dígitos (1,0) que representan un pixel.

Imágenes bitmap. Las imágenes de mapa de bits o *bitmap*, están construidas mediante un conjunto de pixeles, cada una de las cuales contiene información acerca del color. La calidad de la imagen dependerá de la cantidad de pixeles utilizados para representarla.

Imágenes vectoriales. Las imágenes vectoriales están compuestas por entidades geométricas simples: como segmentos y polígonos. Cada una de estas entidades está definida matemáticamente por un grupo de parámetros como lo son las coordenadas iniciales, grosor, color y contorno.

Al estar compuestas por entidades geométricas simples, las imágenes vectoriales se pueden cambiar de escala, para ampliarlas o reducir las, sin que la imagen pierda calidad.

1.2. Resolución de una Imagen

La resolución de una imagen nos indica la cantidad de pixeles que hay en una determinada medida de longitud (pulgada o

centímetro). La resolución de una imagen se mide en pixeles por pulgadas (*ppi*) y en pixeles por centímetro. Al aumentar la resolución de la imagen, mas pixeles hay en ello, por lo tanto más grande será su mapa de bits (*bitmap*).

Tabla 1. Resolución de una imagen

Resolución (px/in)	Superficie (in x in)	Pixeles (px)
8	1	64
72	3	64
266	3	64

1.3. Color de la Imagen

Se debe tener en cuenta que la percepción del color a través del ojo humano es según la longitud de onda de la luz que le llega. La luz contiene todo el espectro de color. Cuando hay superposición de colores el color resultante es blanco, mientras que la ausencia de luz es percibida el ojo como el color negro. Sin embargo, las propiedades del color pueden ser definidas matemáticamente de diferentes formas permitiendo que esta pueda ser capturada y clasificada por una computadora.

Color HSB. El modelo HSB, también es conocido como HSV define el color mediante tres componentes: tono, saturación y brillo. El tono describe la longitud de onda de luz relegada o

emitida por un objeto. La saturación es la intensidad del color y representa el matiz en relación con el tono, se mide en términos de porcentaje de 0 % a 100 % donde se encuentra la saturación máxima. El brillo hace referencia a la claridad u oscuridad relativa de cada tono de color e igualmente que la saturación se mide en porcentaje 0 % (oscuro) y 100 % (claro).

Color RGB. Este modelo se basa en la descomposición del color en términos de la intensidad de los colores primarios de la luz (Rojo, Verde, Azul), es el más usado en iluminación artificial, en la mayoría de los monitores y pantallas gráficas. Estos tres colores crean el blanco cuando se combinan simultáneamente, por eso también es llamado modelo aditivo.

1.4. *Secuencia de Imágenes (fps)*

La secuencia de imágenes en un video está relacionada con la velocidad de cuadro o frecuencia de cuadro, la cual indica la velocidad en la que el dispositivo de imágenes produce imágenes únicas y consecutivas llamadas macros (*frame*). Este término se aplica igualmente a las gráficas de un computador, filmadoras, cámaras de video, y sistemas de captura de movimiento. Esta velocidad de cuadros se expresa frecuentemente como fps (fotogramas por segundo).

El ser humano posee un sistema visual el cual puede procesar 10 a 12 imágenes por segundos por separado, la corteza visual se aferra a una imagen de una quinceava parte de un segundo, así que si otra imagen se recibe durante ese periodo, la ilusión de

continuidad se crea, lo que permite una secuencia de imágenes fijas para dar la sensación de movimiento. [3]

2. PROCESAMIENTO DE IMAGENES

2.1. Filtrado de Imágenes

El proceso de suavizar la imagen (*Smoothing*) es una operación muy usada en el procesamiento de imágenes para reducir el ruido, causado por el sensor en las cámaras digitales.

Algunos tipos de proceso de suavizado son:

Filtro desenfoque simple. En la operación de desenfoque simple, los píxeles de salida son la media simple de todos los píxeles de una ventana alrededor del píxel

Filtro de mediana. El filtro de mediana busca localizar dentro de la imagen, los píxeles que tengan valores extremos o excesivos para cambiarlos por valores real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que tienen las imágenes que han sufrido un filtro de media.

Este filtro revisa el valor de cada píxel y lo compara con los píxeles vecinos para definir si es valor es representativo en su entorno, en caso de que no lo sea, este píxel es cambiado por el valor de la mediana de los píxeles vecinos.

Filtro Gaussiano. Este filtro modifica los valores de los píxeles según la curva de Gauss, es decir en forma de campana.

El valor de píxel máximo se encuentra en el parte central de la matriz y disminuye hacia los extremos más rápido cuando es menor el parámetro de desviación típica s . El resultado será un conjunto de valores entre 0 y 1.

La ecuación matemática para calcular el valor de cada píxel según su posición x , y es

$$G(x, y) = \frac{e^{-\frac{x^2+y^2}{2s^2}}}{\min_{x,y} \left(e^{-\frac{x^2+y^2}{2s^2}} \right)} \quad (1)$$

donde x y y , es la posición del píxel, y s es la desviación típica.

2.2. Región de Interés

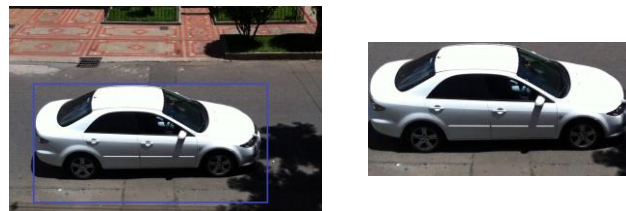


Figura 1. Ejemplo región de interés

Esta técnica de procesamiento es útil a la hora de trabajar con regiones de interés dentro de una imagen y así poder desechar cierta cantidad de píxeles que no nos interesa al momento de hacer el análisis de la imagen.

En este caso nuestra región de interés es el carro, pues es el objeto a seguir en las imágenes, este proceso ayuda a reducir la velocidad de procesamiento debido a que solo se estudia el área de interés.

2.3. Características de Operaciones Morfológicas

La palabra morfología en matemática, se refiere a la forma y estructura basada en teoría de conjuntos. En visión artificial se usa para indicar la forma y estructura de una imagen en un conjunto de píxeles. Estas operaciones pueden simplificar datos en una imagen, preservar características esenciales y eliminar aspectos irrelevantes, además de [4]:

- Supresión de ruido y simplificación de formas.
- Destacar la estructura de objetos. (extraer esqueleto, silueta del objeto, envolvente).
- Descripción cualitativa de objetos (área, perímetro).

2.4. Operaciones Morfológicas

Dilatación. La dilatación es una convolución de una imagen que llamaremos A, con un elemento estructural que llamaremos B en

la Figura 2, este elemento puede ser de cualquier forma o tamaño comúnmente llamado plantilla o máscara, y su efecto para la dilatación es la de un operador máximo local. [5]

Esta operación se define como:

$$A \oplus B = x | (\widehat{B}_x \cap A \neq \emptyset) \quad (2)$$

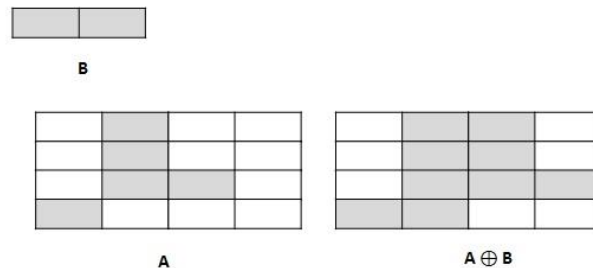


Figura 2. Concepto dilatación

Erosión. La erosión es la operación inversa a la dilatación, donde una imagen A, con un elemento estructural B mostrada en la Figura 3, busca un valor de un mínimo local en común sobre el área del elemento B. [5]

Esta operación se define como:

$$A \ominus B = x | (B_x \subseteq A \neq \emptyset) \quad (3)$$

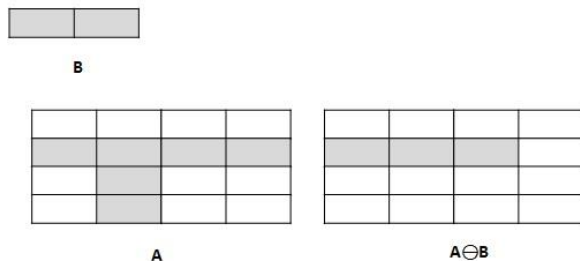


Figura 3. Concepto Erosión

Apertura. La apertura se obtiene de la operación morfológica erosión seguido de la operación dilatación con el mismo elemento estructural B. Este proceso tiene a eliminar todos los objetos que no están completamente contenidos en el elemento estructural, además no disminuye el tamaño de los las figuras de la imagen. Esta operación puede ser ideal para la eliminación de ruido, aunque no preserva la forma geométrica de las figuras de la imagen.

Se llama apertura debido a que su función es separar espacios entre las figuras de la imagen.

Está definido como:

$$A \circ B = (A \ominus B) \oplus B \quad (4)$$

Clausura. La apertura se obtiene de la operación morfológica dilatación seguido de la operación erosión con el mismo elemento estructural B. Este proceso tiene a juntar todos los objetos que se encuentran cerca y reducir segmentos no deseados.

Se llama clausura debido a que su función es unir espacios entre las figuras de la imagen.

Está definido como:

$$A \cdot B = (A \oplus B) \ominus B \quad (5)$$

2.5. Detección de bordes

La detección de bordes es un proceso en el análisis de imágenes, para encontrar discontinuidades o grandes cambios en la intensidad de la luz entre pixeles. Estos cambios se pueden usar para determinar la profundidad, tamaño, orientación y propiedades de la superficie de algún objeto en particular.

El algoritmo de Canny o comúnmente llamada detector de bordes se basa en el uso de la primera derivada, la cual toma el valor de cero en las regiones donde el cambio de intensidad no varía, y un valor constante en las demás transiciones de intensidad. Por lo tanto un cambio de intensidad, se manifiesta con un cambio en la primera derivada, esta característica es usada para la detección de borde.



Figura 4. Ejemplo detección de bordes

2.6. Detección de Contornos



Figura 5. Ejemplo detección de contornos

La detección de contornos es un proceso básico en el procesamiento de imágenes, pues cuenta con información útil acerca de los límites del objeto que pueden ser utilizados para el

análisis, aplicación de filtrado y para detección de objetos.

Este proceso por lo general se realiza después de hacer una buena detección de bordes, por el método de *Canny*.

El paso de nivel brillante a uno oscuro, o viceversa, establecen un contorno.

2.7. Transformada de Hough

La transformada Hough es un algoritmo utilizado en el reconocimiento de patrones de una imagen, la cual permite encontrar formas como círculos, líneas, entre otras, dentro de la imagen.

La original transformada de Hough era una transformada de línea, que es una forma relativamente rápida de búsqueda líneas rectas en una imagen binaria. La transformación puede ser más generalizada a casos distintos de líneas simples. [5]

Transformada Lineal. La teoría básica de la línea transformada de Hough es que cualquier punto en una imagen binaria podría ser parte de un conjunto de posibles líneas.

Transformada Circular. La transformada de Hough circular funciona de manera similar a la lineal, este proceso se basa en el método de gradiente de Hough, el cual permite encontrar formas circulares dentro de las imágenes.

3. DETECCIÓN Y SEGUIMIENTO DE OBJETOS

La detección y seguimiento de objetos es una de las funciones principales de la visión artificial. En este paso se realiza la segmentación la cual consiste en aislar en la imagen todos los conjuntos de píxeles candidatos a ser objeto a seguir e ignorar el conjunto de píxeles que no está dentro del rango de interés.

Existen varios caminos para lograr detección de objetos en una imagen, en este capítulo se van a mostrar dos formas de conseguir esta segmentación, por medio del color y mediante sustracción de fondo.

3.1. Detección por color

La detección de objetos por color tiene varios métodos, entre ellos está la segmentación por métodos de agrupamiento y clasificación (Celenk, 1990), detección de contornos (Zugaj y Lattuati, 1998), fusión de regiones (Trémeau y Borel, 1998) [6]. Otros de los métodos es la segmentación basada en histogramas hue sobre imágenes HSV, donde se define el color por medio de tres componentes: tono, saturación y brillo.

Para lograr la segmentación de un color en particular se debe seguir los siguientes pasos:

1. Se calcula un histograma en *Hue* para el color del objeto, asumiendo que la imagen contiene solo el objeto.
2. Se calcula la proyección del plano *Hue* de la imagen de entrada donde se busca el objeto, usando el histograma.

Para cada conjunto de píxeles se impone el valor del histograma correspondiente al sector de la imagen de salida. Estadísticamente, el valor de cada píxel de salida es la probabilidad de que coincida con el conjunto de valores observado del histograma.

3. Se realiza una segmentación del objeto contra al fondo de la imagen. Esta nueva imagen es de solos dos niveles.
4. Se encuentran las componentes asociadas a la imagen resultante.

La imagen de la Figura 6 muestra la segmentación de una manzana del objeto de fondo, mediante las componentes HSV.

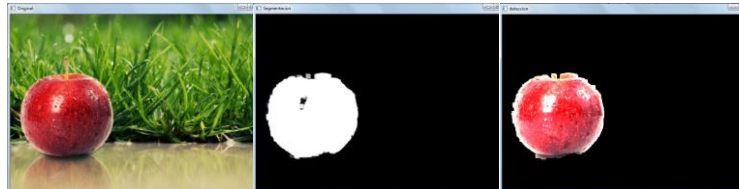


Figura 6. Ejemplo detección por color

Desventaja detección por color. Debido a que en nuestro caso deseamos la detección de vehículos, y estos no tienen un color en específico, pues existen diferentes gamas de colores, esto hace

que la detección sea más compleja. Por lo tanto no se utilizó este método.

3.2. Detección por sustracción de fondo

La sustracción de fondo es una práctica elemental de segmentación de imágenes. Tiene numerosas aplicaciones prácticas, como seguimiento (tracking) de vehículos, personas, segmentación de diferentes escenas de video y compresión de video, entre otras.

Existen dos tipos de fondo, el fondo estático el cual durante todo el procesamiento, la escena de segundo plano no tiene cambios drásticos, por el contrario el fondo dinámico presenta alteraciones en la escena de segundo plano, lo cual lleva hacer otro procedimiento para obtener la segmentación del objeto por cada cuadro de imagen.

Fondo estático. La sustracción de fondo estático se realiza comparando las imágenes cuadro a cuadro de un video, con la imagen inicial del fondo. Para lograr esto se debe seguir los siguientes pasos:

1. Primero obtenemos la imagen inicial del video, en el instante $t = 0$, esta será nuestra imagen fondo.
2. Después para cada imagen del video se debe
3. Obtener la imagen del siguiente cuadro (*frame*)

4. Realizar una conversión a escala de grises para poder hacer comparaciones
5. Se compara cada cuadro con la imagen inicial, esta comparación es una resta, en donde se puede apreciar los cambios que se ha producido en la imagen de fondo desde el instante inicial.

En la Figura 7 se puede observar un ejemplo de segmentación mediante fondo estático. (a) Imagen de fondo. (b) imagen fondo en escala de grises. (c) Imagen de video en $t = 1$. (d) Segmentación del objeto en $t = 1$. (e) Imagen de video en $t = 2$. (f) Segmentación del objeto en $t = 2$.

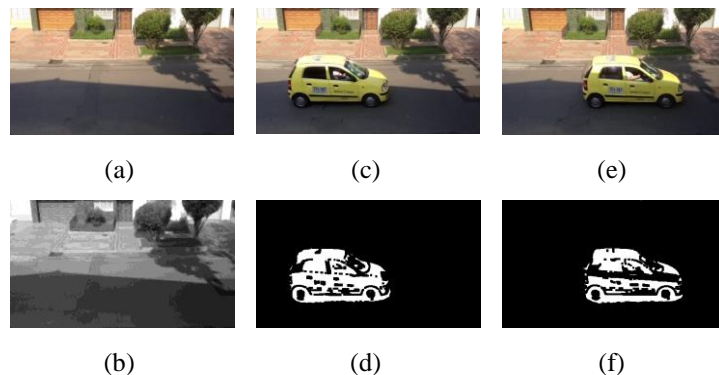


Figura 7. Ejemplo detección fondo estático

Fondo dinámico. Cuando se realiza segmentación por medio de fondo estático se encuentra con varios inconvenientes que pueden llegar distorsionar la segmentación del objeto de interés. Alguno de estos problemas pueden ser:

- La imagen de fondo no es estático, debido a que se presentan alteraciones en ella, como lo son los movimientos de los árboles, la sombra que produce el sol, la cual se va moviendo durante el transcurso del día, entre otros.
- La cámara no está fija y por lo tanto su movimiento debido a vibraciones producidas por carros o debido a fuertes vientos puede llegar a presentar perturbaciones en la imagen de fondo.
- Algunos objetos grandes y lentos que aparecen en la escena provocando que se confunda con la imagen de fondo de la escena.

Por lo tanto se propone realizar la sustracción de fondo de forma dinámica. Es decir que no exista para toda la secuencia una única imagen de fondo, sino que la imagen de fondo de cada cuadro, es la imagen anterior. Es decir una secuencia de video con 8 cuadros, tendrá 7 imágenes de fondo diferentes

La secuencia para lograr la segmentación por este método es la siguiente:

1. Obtenemos una imagen inicial de fondo.

2. La siguiente imagen del video, se le realiza una resta con la imagen anterior, para obtener el objeto segmentado.
3. Esta imagen a su vez, será la imagen de fondo para el siguiente cuadro de video.
4. Este ciclo se repite hasta procesar todos los cuadros del video.

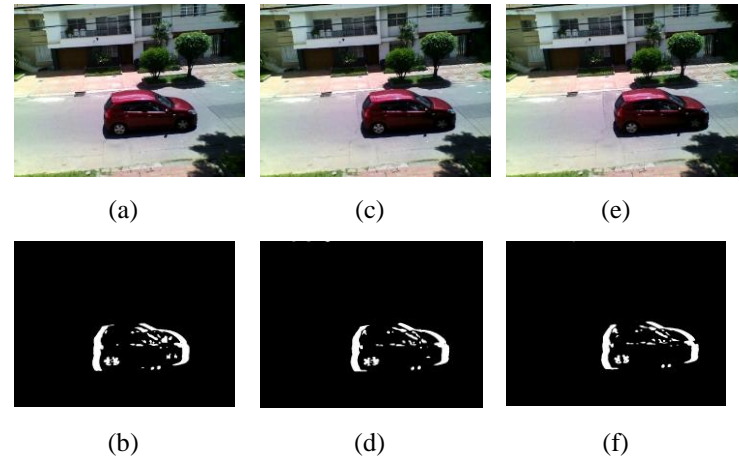


Figura 8. Ejemplo detección fondo dinámico

Como se puede observar en la Figura 8, no es necesario para la segmentación tener un fondo fijo o estático, sino un fondo

dinámico, como lo es la imagen (a) con respecto a la imagen (c), y a su vez la imagen (c) es el fondo de escena de la imagen (e), y sucesivamente con todos los cuadros del video.

3.3. Seguimiento de objetos

El seguimiento de imágenes (*tracking*) es la técnica de ubicación y seguimiento de uno o más objetos móviles en un video. Este proceso puede ser un poco lento debido a la gran cantidad de datos que se deben procesar.

Cuando se habla de *tracking* o seguimiento, se tiene que especificar el objeto o conjunto de interés que se quiere seguir, esto dependerá de la aplicación específica, por ejemplo en un equipo de vigilancia de los edificios, los objetivos pueden ser personas, mientras que en una aplicación para juegos interactivos, los objetivos pueden ser las manos o la cara de una persona. [4]

3.4. Desafíos

Los principales retos que tienen que ser tenidos en cuenta a la hora de diseñar un seguidor de objetos, está relacionado con la similitud de apariencia entre el objetivo y otros objetos en la escena, e inclusive variaciones del mismo objeto en los diferentes cuadros. [7]

Algunos factores que dificultan el seguimiento de objetos pueden ser causados por:

- Cambios en la posición: el objeto de interés puede variar su aspecto cuando gira o rota su posición frente al plano de la imagen.
- Iluminación de ambiente: el objeto de interés también puede cambiar de apariencia en la intensidad, dirección o color provocados por la luz del ambiente, esto se evidencia por ejemplo cuando aparecen nubes que tapan el sol y oscurecen la escena.
- Ruido: Esto se produce en el momento de adquirir las imágenes, se le introduce una señal a causa de la calidad del sensor, afectando el objeto de interés, lo que provoca un bajo rendimiento del seguidor.
- Oclusiones: Esto sucede cuando el objeto de interés es tapado parcial o totalmente por otros objetos en la escena. Las oclusiones son habitualmente generadas por que el objeto de interés se encuentra detrás de un objeto estático, como por ejemplo árbol (Figura 24.b). O cuando el objeto de interés está detrás de otros objetos móviles.

3.5. Técnicas de seguimiento de objetos

El objetivo principal de las técnicas de seguimiento es formar una trayectoria de un objeto a través del tiempo, posicionándolo dentro de una imagen.

La extracción de características es el primer paso en el seguimiento y permite destacar información de interés de las

imágenes. Estas características se reúnen en tres grandes grupos: de bajo nivel, nivel medio y alto nivel.

Bajo nivel. El objetivo es aprovechar las características de bajo nivel en las diferentes etapas para realizar seguimiento y detección, entre ellos están [7]:

1. **Color:** Un espacio de color es una representación matemática de nuestras percepciones visuales, y nos permite analizar y gestionar el color. Algunas de estas representaciones matemáticas son: CIELAB, CIELUV, RGB, YIQ, YUV, YCbCr.
2. **Derivadas y Gradientes:** Los cambios de intensidad locales tienen información importante acerca de la apariencia de los objetos de interés. Algunas operaciones son: operador Sobel, selección de escala.
3. **Movimiento:** La estimación de movimiento tiene como objetivo la caracterización de movimiento aparente mediante el flujo óptico. Este vector describe el desplazamiento de un píxel entre dos *frames*. Esta estimación de movimiento presenta dos problemas la oclusión y la apertura.

Nivel medio. El extracción de características del objeto de bajo nivel puede no ser adecuado para lograr un buen seguidor de vídeo. Una solución consiste en analizar el vídeo usando

subconjuntos de píxeles que representan las estructuras pertinentes. Entre ellos están [7]:

1. **Bordes:** Los detectores de bordes generalmente producen un mapa binario que pone en relieve la presencia de bordes o elementos de borde en una imagen. Los bordes se asocian con altas respuestas en el operador derivador de primer orden.
2. **Puntos y Regiones de Interés:** Tienen por objeto la selección de características de la imagen que pueden ser localizados con precisión y de forma fiable, cuando se plantean condiciones de iluminación que pueden cambiar. La mayoría de los detectores de punto de interés tienden a seleccionar los patrones locales, como esquinas, bordes, y las regiones con textura diferentes.
4. **Regiones Uniformes:** Utilizando las regiones de interés para localizar un objetivo, cuyos píxeles pueden compartir algunas propiedades tales como el color o movimiento. Para generar estas regiones uniformes, se pueden agrupar las funciones de bajo nivel seleccionados dentro del área objetivo.

Alto nivel. Nos referimos a los resultados de la detección de objetos como características alto nivel. Las características de alto nivel pueden ser el centroide, toda la zona o la orientación de un objetivo. Detectores de objetos basados en sustracción de fondo se

utilizan en la mayoría de los casos con cámaras estáticas, siendo sensibles a los cambios de iluminación. [7]

1. Modelos de Fondo: Una variedad de métodos para la detección de objetos en movimiento se basa en la sustracción de fondo. Los Algoritmos de sustracción de fondo usan el supuesto de cámara estática para detectar variaciones de intensidad de los píxeles. La estimación y la actualización del modelo de fondo se pueden realizar mediante la recopilación de estadísticas de los valores de píxel en el tiempo.
2. Modelos de objetos: Este modelo de extracción de característica y seguimiento, utiliza una clase de objeto predefinida. A esta clase se le aplica algoritmos de clasificación como *AdaBoost*, el cual crea un objeto *cascade*, para la detección y seguimiento de los objetos entrenados.[7]

En este trabajo se utilizó la extracción de características de alto y medio nivel para la detección y seguimiento de vehículos. Lo primero que se realizó fue la extracción de fondo, esto debido a que se utilizó una cámara estática, lo cual facilita segmentar el objeto del fondo. Para evitar cambios en la escena inicial, lo cual pueda generar en una mala detección, se actualiza el fondo de la imagen *frame* a *frame*.

Después de haber detectado el vehículo, utilizamos la extracción de características de medio nivel, donde se obtiene la región de interés del vehículo y se crea una plantilla o *template* la cual contiene la información como posición e intensidad de los píxeles del objeto de interés al cual se quiere seguir. Esta plantilla se compara con cada una de las imágenes usando una función de probabilidad, donde se sigue la posición que entregue el valor más óptimo.

4. SISTEMA DE MOVILIDAD EN MEDELLÍN

Medellín actualmente cuenta con un proyecto basado en un Sistema Inteligente para la Movilidad, llamado SIMM, este se encuentra enmarcado dentro de los objetivos de los Sistemas Inteligentes de Transporte, buscando resolver los problemas de movilidad a través de soluciones tecnológicas. [8]

La gran cantidad de personas y vehículos que se desplazan en las vías de la ciudad durante el día, pueden generar altos niveles de congestión, principalmente en las horas pico, además de accidentes debido a la falta de prudencia de algunos conductores y/o peatones que transitan por los cruces viales.

4.1. Parque Automotor

Se estima que en el Área Metropolitana en lo referente al parque automotor, se encontraban matriculados en el 2012 un total de 1'100.000 vehículos y 537.969 motocicletas. Solo en Medellín se encuentran matriculados hasta agosto de 2013 un total de 176.811 vehículos particulares, 26.327 motocicletas, 4.288 vehículos de transporte público colectivo, 19.126 taxis y 3.872 vehículos oficiales. [8]

4.2. Velocidad promedio

La interacción entre vehículos y personas que circulan en las vías de la ciudad, producen una velocidad promedio de desplazamiento vehicular alrededor de 34 km/h y un tiempo promedio de viaje de 25 minutos. [8]

4.3. Índices de accidentalidad

Los índices de accidentalidad en la ciudad de Medellín, en los últimos años se ha logrado reducir de manera importante, esto gracias a una combinación de mejora en el control de las vías y ampliación de campañas de cultura ciudadana. Medellín es la ciudad que ha obtenido los mejores logros en reducción de accidentalidad en Colombia. [9]

4.4. Sistema inteligente de transporte

El aumento de números de vehículos, el crecimiento de la población, la urbanización y la industrialización, generan un alto impacto sobre la movilidad en la ciudad, ocasionado como consecuencia la disminución de la eficiencia de la infraestructura de transporte, incremento de tiempo de viajes, accidentalidad, contaminación, entre otros.

Los Sistemas Inteligentes de Transportes (ITS) nacen con la intención de resolver los problemas de movilidad en las vías de la ciudad, estas soluciones integran tecnologías de información y comunicación en la infraestructura de transporte y vehículos con el fin de optimizar el uso de las vías y en consecuencia mejorar la seguridad, reducir tiempos de desplazamiento y contaminación.

Algunos países en América, como Brasil, Chile, México, Argentina y Panamá, cuentan con estos sistemas. En países como Francia y España su implementación en los años 90 permitió reducir la accidentalidad vial hasta un 30%. [8]

El proyecto SIMM, Sistema Inteligente para la Movilidad de Medellín, busca solucionar los inconvenientes de movilidad mediante soluciones tecnológicas, este proyecto se viene desarrollando con ayuda de los siguientes componentes:

Paneles informativos. En las principales vías, Medellín cuenta con 22 paneles informativos. Estas pantallas muestran mensajes variables con información en tiempo real sobre el estado de las vías, como accidentes, calles cerradas, entre otros.

Estos paneles informativos ofrecen los siguientes beneficios a los conductores:

- Reducción de la congestión y los tiempos de viaje
- Toma de mejores decisiones en los desplazamientos
- Optimización de los flujos vehiculares
- Reducción de la accidentalidad
- Información al ciudadano, oportuna y dinámica

La información que presentan los paneles informativos es enviada por el Centro de Control de Tránsito, con un personal capacitado, que está permanentemente monitoreando el estado de la red vial de la ciudad.

Optimización de semáforos. Actualmente Medellín cuenta con algunos semáforos inteligentes, los cuales se adaptan a las

condiciones de tráfico. Este sistema cuenta con un mecanismo de obtención de datos para el análisis vehicular, que permite diagnosticar en cuales lugares, zonas, e interacciones viales de la ciudad se presentan cambios de flujo vehicular, como también los horarios donde se presentan mayores congestiones y otras condiciones que influyan en el tráfico.

La red de semaforización cuenta con una tecnología aplicada, para realizar mejores análisis de circulación vehicular, lo que permiten tomar decisiones acertadas para la programación en función del tiempo y del tráfico en los semáforos de la ciudad.

- La optimización de Semáforos ofrece los siguientes beneficios:
- Reducir los tiempos de espera en los semáforos
- Minimizar las paradas de los vehículos
- Maximizar desplazamientos por la ciudad, reduciendo tiempos de recorrido
- Economizar combustible y desgaste de piezas del vehículo
- Ordenar el tránsito

Cámaras de detección. La secretaria de Movilidad de Medellín, cuenta con un sistema de fotodetección en la ciudad, el cual permite mejorar la movilidad y reducir accidentes viales. Este

sistema genera infracciones, las cuales tiene la misma validez que un comparendo impuesto por un agente de tránsito.

Actualmente Medellín cuenta con 70 puntos fijos, con 40 equipos móviles los cuales son rotados por diferentes sitios de la ciudad, esta rotación obedece a criterios que permitan reducir la accidentalidad en diferentes zonas. Las infracciones que estos equipos generan son: [8]

- No detenerse ante un semáforo en rojo
- Conducir un vehículo a velocidades superiores a la máxima permitida.
- Transitar por sitios restringidos o en horas prohibidas (Pico y Placa)
- No respetar el paso de peatones que cruzan la cebrera cuando el semáforo está en rojo.

Los beneficios que tiene este sistema son:

- Reducir la accidentalidad.
- Fortalecer el respeto por las normas de tránsito.
- Disminuir personas heridas y víctimas fatales debido a accidentes viales.

Circuito cerrado de televisión. Medellín cuenta con un Circuito Cerrado de Televisión (CCTV) el cual cuenta con cámaras

ubicadas estratégicamente en la red vial de la ciudad, como son glorietas e interacciones principales y el sistema vial del río.

Estas cámaras envían datos al Centro de Control de Tránsito, que son monitoreados a través de un software especializado, el cual recoge información sobre el estado de las vías. Esta información permite realizar estudios del tráfico vehicular y así poder tomar decisiones que ayuden a la disminución de las congestiones, además de conocer de inmediato incidente que ocurran en las vías y atenderlas de manera rápida.[8]

Estas cámaras permiten:

- Monitorear principales corredores viales, para conocer el estado de las vías y poder detectar incidentes en tiempo real.
- Capturar información para estudios de comportamiento vial, con la intención de identificar problemas que incidan en la circulación y la accidentalidad.
- Obtención de información en línea que permite reportar el estado de las vías a los cuidanos mediante redes sociales.

5. ÓPTICA GEOMÉTRICA

La óptica geométrica es un campo de la óptica que se ocupa del estudio de las imágenes producidas por refracción o por reflexión de la luz, como el diseño de lentes y otros componentes de instrumentos ópticos.

La óptica geométrica se ocupa de las trayectorias de los rayos luminosos, despreciando los efectos de la luz como movimiento ondulatorio, como las interferencias. Estos efectos se pueden despreciar cuando el tamaño la longitud de onda es muy pequeña en comparación de los objetos que la luz encuentra a su paso. [10]

Para poder estudiar la posición de una imagen con respecto al objeto, es necesario entender las siguientes definiciones:

Eje óptico: Es el eje de coordenada perpendicular al plano del lente. El sentido hacia la derecha se toma como positivo, es el sentido de avance de la luz.

Espacio objeto: Es el espacio en sentido izquierdo donde se encuentra el objeto.

Espacio imagen: Es el espacio en sentido derecho donde se encuentra la imagen.

Foco objeto: Punto F del eje óptico cuya imagen se encuentra en el infinito del espacio imagen.

Foco imagen: Punto F del eje óptico cuya imagen se encuentra en el infinito del espacio objeto.

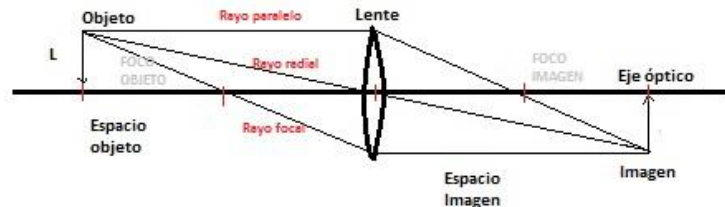


Figura 9. Óptica geométrica

5.1. Reflexión y refracción

Cuando la onda de luz alcanza la superficie de dos medios de diferente naturaleza se produce, dos nuevas ondas, una que atraviesa la superficie límite y se propaga en el segundo medio (Refracción) y otra que retrocede hacia el medio de partida (Reflexión).

Refracción. La refracción de una onda se produce cuando existe un cambio de dirección de la onda al paso de un medio a otro distinto. Este cambio de dirección se genera como resultado de la diferente velocidad de propagación que tiene la onda en ambos medios. [11]

Reflexión. La reflexión implica que la onda al llegar a un obstáculo no la atraviesa, sino que su dirección se invierte, en otras palabras, rebota. Aunque el obstáculo absorba parte de la

energía recibida se produce también reflexión en la que se trasmite de vuelta parte de la energía a las partículas del medio incidente. [12]

Según el tipo de superficie existen dos clases de reflexión. La reflexión especular ocurre cuando la superficie es completamente lisa, lo que permite que la luz se proyecte de manera uniforme, como es el caso de los espejos. La reflexión difusa ocurre cuando la superficie es rugosa, lo que origina que la luz se disperse en diferentes direcciones, como es el caso del cine, donde se refleja la luz en toda la pantalla para que todos los espectadores puedan observar la película.

5.2. Lentes

Los lentes son un medio de material transparente que tiene la capacidad de refractar la luz, es decir, desviar los rayos luminosos que procede de un objeto y formar una imagen del mismo.

Existen dos tipos de lentes: convexas y cóncavas.

Lente convexa. Una lente convexa o convergente tiene la parte central más gruesa que los bordes, haciendo converger los rayos hacia el centro. Esto hace que se forme la imagen del objeto al otro lado de la lente. El ser humano tiene una lente convexa, que puede cambiar de forma para enfocar objetos a diferentes distancias

La longitud focal de un lente convexo depende tanto de la forma como del índice de refracción del material del que está hecho.

Una propiedad muy importante de estas lentes es que producen imágenes reales.

Lente cóncava. Las lentes cóncavas o divergentes están curvadas hacia adentro, logrando desviar (divergir) los haces luminosos que inciden paralelamente sobre una de sus caras. A diferencia de las lentes convergentes o convexas, las cóncavas solo producen imágenes virtuales, es decir, imágenes de las que parece proceder los rayos de la luz.

5.3. Calibración para iPhone4

Problema. El objetivo principal es determinar el valor de conversión de pixeles a cm, el cual depende de la distancia de la cámara al objeto, un valor alfa el cual vamos hallar y el la distancia focal de la cámara, en este caso de un iPhone4.

Para lograr existen dos métodos, el primero corresponde a la toma de una imagen con una sola cámara, el segundo caso, es cuando el objeto de interés es visualizado por dos o más cámaras. Como en nuestro caso solo tenemos una sola cámara, se va a optar por el primer método. [13]

Captura de imagen. Cuando se toma la imagen del objeto con una cámara, la distancia del objeto en la imagen con respecto a la lente no se conoce. Según el principio de formación de una imagen virtual se puede observar en la Figura 10.

Para obtener el tamaño del objeto real, es necesario conocer la distancia del objeto, la distancia focal y el tamaño del objeto

proyectado en el plano de imagen. Estas variables se pueden relacionar con la ley básica de la óptica

$$\frac{L'}{f} = \frac{L}{d} \quad (6)$$

donde L es el tamaño del objeto real, f la distancia focal, L' es el tamaño del objeto proyectado y d es la distancia al objeto.

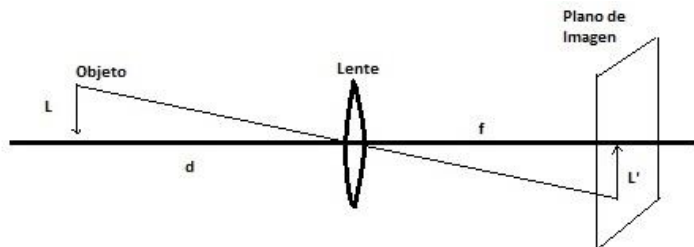


Figura 10. Formación imagen

En la Figura 10 se observa como es la proyección de la imagen en el plano proyectado, y su relación entre variables. [14]

Toma de datos. Para realizar pruebas y encontrar las constantes que son necesarias para encontrar el tamaño del objeto real, o el

valor α de conversión entre pixeles y cm, se utilizó un vehículo como modelo, debido a su gran tamaño y pruebas a distancias donde normalmente estará la cámara de detección.

Se realizó varias tomas del vehículo a diferentes distancias con la cámara trasera del iPhone4, los resultados se observan en la Figura 11.



Figura 11. Pruebas de calibración

Para encontrar el factor de conversión se evalúa los datos en (8) para distintas distancias conocidas. Para el cálculo del tamaño del objeto proyectado se utiliza

$$L' = \alpha \cdot n \quad (7)$$

donde n es el número de píxeles del objeto y α es el factor de conversión. Se reemplaza (9) en (8) y se despeja el factor de conversión, obteniendo

$$\alpha = \frac{L f}{d n} \quad (8)$$

Tabla 2. Resultados de calibración

d (cm)	n (px)	L (mm)	f (mm)	α (px/cm)
500	281	4150	30	0.8861
600	241	4150	30	0.8609
700	205	4150	30	0.8675
800	179	4150	30	0.8694
900	165	4150	30	0.8383
1000	149	4150	30	0.8357
Promedio				0.8596

Resultados. Para las diferentes distancias encontramos valores del

ancho del objeto L' . Con estos datos y utilizando (10), en donde despejamos el factor de conversión, obtenemos los resultados de la Tabla 2.

La distancia focal de la cámara trasera del celular iPhone4 es 30 mm. [15]

Haciendo un promedio de los valores de α , se obtiene un valor de conversión de 0.86.

6. LENGUAJE DE PROGRAMACIÓN

6.1. *Java*[®]

En 1995 Sun Microsystems creó su primera plataforma en lenguaje *Java*[®]. Actualmente se ejecuta *Java*[®] en más de 850 millones de computadoras y en miles de millones de dispositivos. [16]

Es un lenguaje de propósito general, orientado a objetos y basado en clases, permite escribir programas en interfaz gráfica o textual. También se pueden correr programas de manera incorporado en navegadores de internet en forma de *Java*[®] *applets*.

Java[®] es un lenguaje abierto, no propietario, que puede ser implementado por muchas compañías y venderse como un paquete.

6.2. *OpenCV*

OpenCV es una librería de código abierto orientada principalmente para visión artificial en tiempo real. Esta librería corre en diferentes plataformas como Windows, Linux y Mac OS X, enfocada en C y C++ aunque existen desarrollos en otros lenguajes de programación. [17]

7. DESARROLLO DE SOFTWARE

El desarrollo se implementó en *Java*[®], con ayuda de las librerías de procesamiento de imágenes OpenCV. Las funciones que se utilizaron están incluidas en las librerías *cv.h* donde están los principales algoritmos, *cxcore.h* que contiene las estructuras básicas y funciones para graficar, *higui.h* para el manejo de ventanas además de entrada y salida de video.

En el Anexo 1 se mostrara la instalación de la librería OpenCV en la plataforma de *Netbeans* donde se programa *Java*[®].

7.1. Visualización

Para mostrar un video, creamos una instancia del tipo de dato *CvCapture* el cual va a contener el archivo de video que deseemos mostrar o el video en vivo de nuestra cámara web.

Las imágenes se representan con tipo de dato *IplImage*, el cual contiene toda la información de la imagen, como es tamaño, ancho, alto, número de canales, entre otros.

Para mostrar el video, se captura *frame* por *frame* y se muestra con una velocidad fps de 15 *frames* por segundo, esto debido a que se utiliza una transmisión *streaming*, donde la cantidad de *frames* transmitimos en un segundo es menor a una transmisión de video cargada desde el mismo computador.

El diagrama de flujo que se muestra a continuación explica como es el procedimiento para mostrar el video que genera nuestra cámara web.

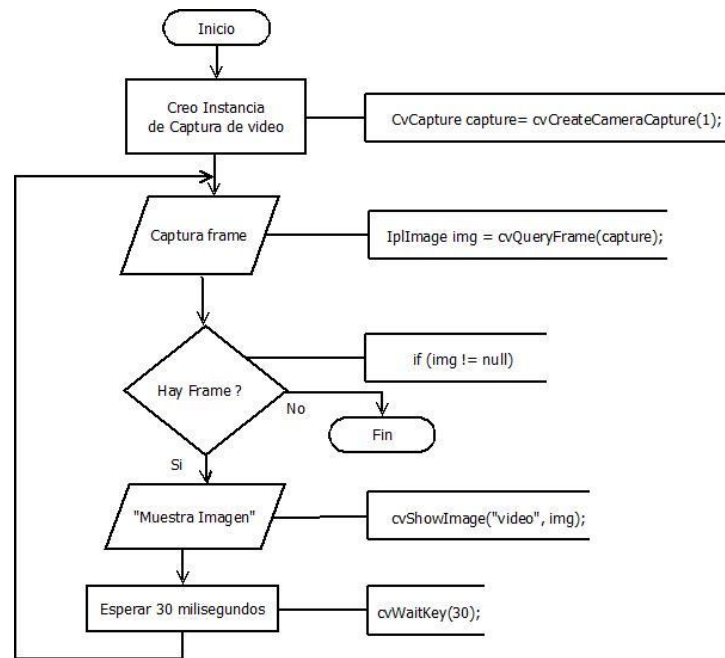


Figura 12. Visualización

7.2. Configuración inicial

En la configuración inicial, se obtienen variables como son el ancho y alto del *frame*. Se calcula la distancia euclidiana del

punto donde está la cámara, hasta el punto de la vía donde va a pasar los vehículos. Además se configuran algunas variables como el ancho y altura máximo y mínimos que tendrá aproximadamente los vehículos que van a transitar en la vía, esto se obtiene según la posición de la cámara.

7.3. Pre Procesamiento

En esta parte del desarrollo se realiza la sustracción de fondo dinámica, explicada en el capítulo 3. Además se utilizan algunas funciones morfológicas para segmentar, y obtener una buena detección del vehículo.



Figura 13. Pre Procesamiento

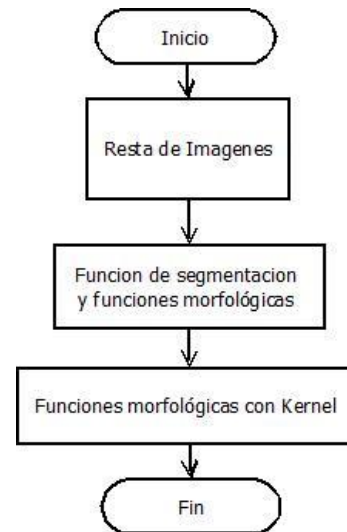


Figura 14. Diagrama flujo Pre Procesamiento

7.4. Procesamiento de Contornos

Después de haber obtenido la sustracción del fondo, y al haber aplicado algunas funciones de segmentación, se realiza un proceso de reconocimiento de contornos, para lograr detectar la silueta del carro en la imagen.

Para esto se utiliza la función de OpenCV llamada `cvFindContours`, la cual encuentra todos los contornos del objeto, esta secuencia de contornos se dibujan mediante el ciclo y la función `cvDrawContours`.

Solo se dibuja los contornos mayores a un área mínima y menores a un área máxima, esto se realiza para evitar dibujar elementos u objetos que no son de interés. El área máxima y mínima se calcula dependiendo de la posición de la cámara esto se realiza en la configuración inicial.



Figura 15. Procesamiento Contornos

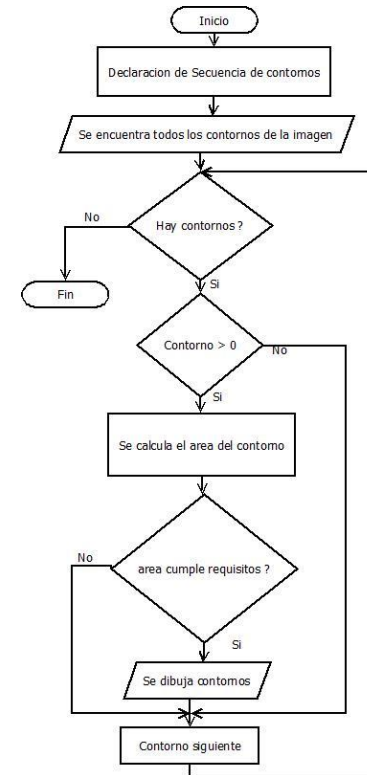


Figura 16. Diagrama flujo Pre Procesamiento

7.5. Detección

Este es el proceso más importante de todo el algoritmo, pues aquí es donde se debe detectar el vehículo que vamos a seguir.

Para realizar esto calculamos de nuevo los contornos de la imagen y con una función que se llama `cvBoundingRect` buscamos en la imagen figuras semejantes a rectángulos.

Solo se acepta rectángulos con una altura y un ancho que sea aproximadamente igual al de un vehículo que pase por la vía. Estas variables se definen al inicio en la configuración inicial.

Cuando detectamos el rectángulo con las medidas correctas, creamos una plantilla del vehículo, para poder en los próximos *frame* seguir al vehículo.



Figura 17. Detección del vehículo

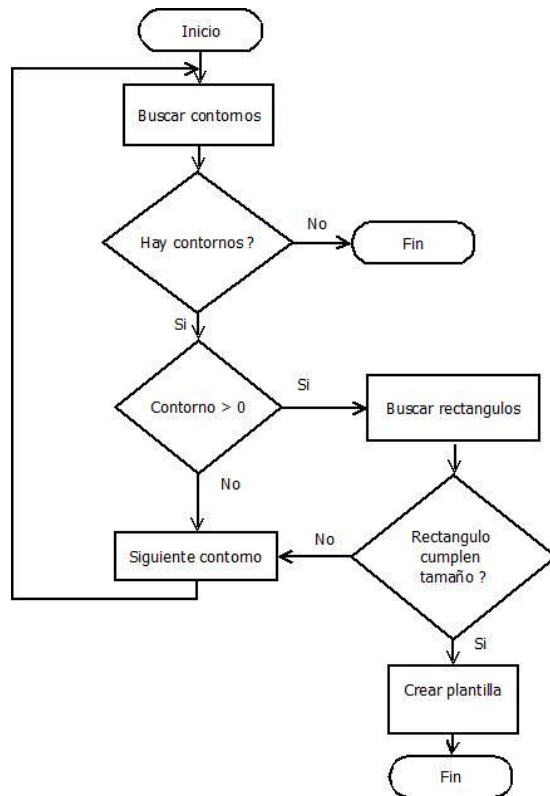


Figura 18. Diagrama flujo detección

7.6. Seguimiento

Después de haber obtenido la plantilla, se dispone a seguir el vehículo en los siguientes *frames*, utilizando una función de OpenCV llamada `cvMatchTemplate` que compara el siguiente *frame* con la plantilla y el resultado lo almacena en una matriz.

Esta técnica se utiliza para encontrar áreas que coincidan o sean similares a en las siguientes imágenes. El método que se utilizó para la comparación fue el de Diferencia Cuadrática Normalizada.

$$R(x, y) = \frac{\sum_{x'y'} [T(x', y') - I(x + x', y + y')]^2}{\sqrt{\sum_{x'y'} T(x', y')^2 \cdot \sum_{x'y'} I(x + x', y + y')^2}} \quad (9)$$

donde I es la imagen de entrada, T es la plantilla y R el resultado. Esta técnica arroja 0 si la coincidencia es exacta, y 1 cuando no existe relación alguna con la imagen. El valor varía entre 0-1.

Después de obtener la matriz resultado, de la función `cvMatchTemplate`, se revisa que área tuvo mayor coincidencia con la plantilla. Para esto se utiliza la función `cvMinMaxLoc`, que me encuentra el valor máximo y mínimo de la matriz resultado, como la posición (x, y) donde se encuentran estos valores. El valor que se utiliza para el seguimiento depende del

método utilizado, en este caso utilizamos el de Diferencias Cuadráticas, por lo tanto el valor que utilizamos es el mínimo global que indica la probabilidad máxima de similitud.

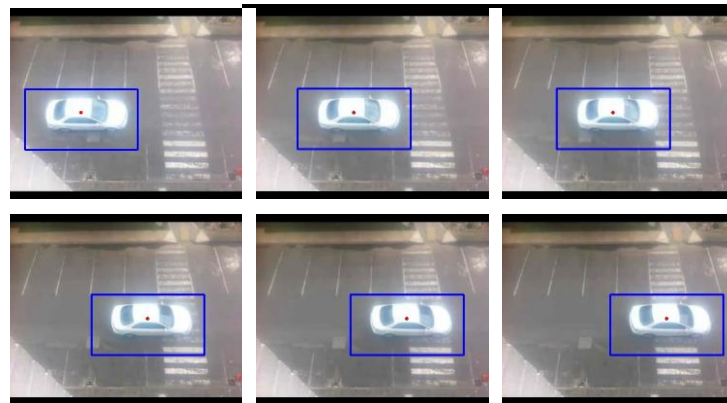


Figura 19. Seguimiento del vehículo

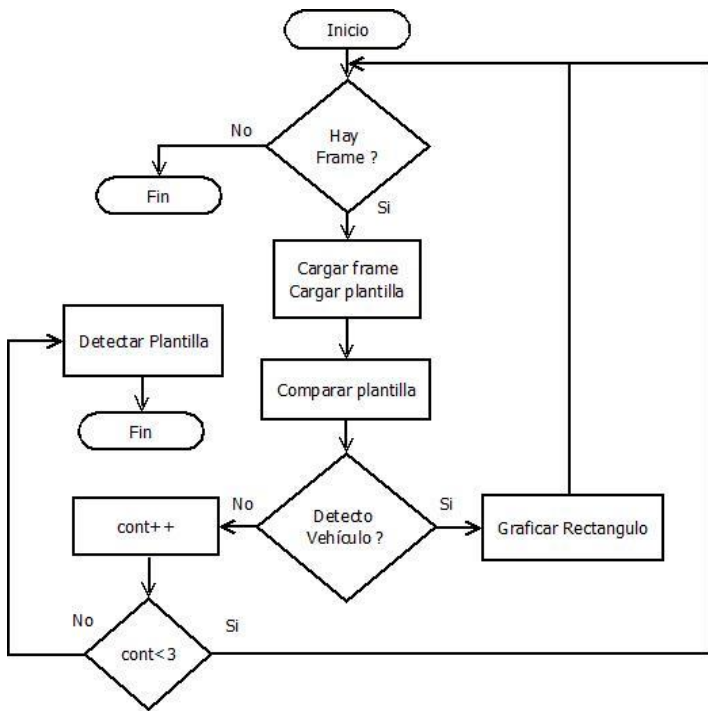


Figura 20. Diagrama de flujo seguimiento

7.7. Cálculo de velocidad

Para obtener la velocidad en la que pasa el vehículo por la cámara es necesario tener un punto de referencia del vehículo, por lo menos en dos *frames*. Se calcula la distancia euclidiana entre los dos puntos, que es la distancia recorrida. El tiempo transcurrido entre los *frames*, con un tiempo aproximadamente entregado por la cámara de 15 *frames* por segundos, se obtiene la velocidad del vehículo instantánea en cada *frame*. Se utiliza la fórmula básica de velocidad

$$v = \frac{d}{t} \quad (10)$$

donde v es la velocidad, d es la distancia recorrida entre dos puntos, y t es el tiempo transcurrido en recorrer ésta distancia.

La distancia entre dos puntos se obtiene aplicando

$$dist_{rec} = \frac{\alpha n d}{f} \quad (11)$$

donde n indica el número de píxeles entre los dos puntos, α es el valor de conversión obtenido en el capítulo 5, f es la distancia focal del iPhone4 y d es la distancia de la cámara al punto donde pasa el vehículo en la vía.

La distancia recorrida se encuentra en decímetros. Cada *frame* se muestra aproximadamente cada 66,67 ms, como se puede observar.

$$t_{frame} = \frac{1000 \text{ ms}}{15 \text{ frame}} = 66,67 \text{ ms/frame} \quad (12)$$

esto debido que en un segundo se recorren 15 *frames*.

Por último la fórmula de velocidad

$$V = \frac{\alpha n d}{f * t_{frame}} * 360 \quad (13)$$

entrega el valor de velocidad requerido, realizando conversión de distancia y tiempo, y así poder obtener las unidades en kilómetros por hora.

7.8. Mecanismo de alerta por infracción de velocidad

La aplicación representara con un recuadro de color rojo a los vehículos que sobrepasen la velocidad permitida, dependiendo del lugar donde se encuentre la cámara (Avenida, calle escolar, calle dentro de un barrio, etc.), además se guarda las imágenes como evidencia de la velocidad transitada.

7.9. Integración

Finalmente cada uno de las etapas del desarrollo es integrado para lograr la aplicación final. Esta aplicación detecta y sigue vehículos *frame a frame*, obteniendo la velocidad instantánea en cada punto en tiempo real, debido a que el video reproducido por el computador proviene de forma *streaming*. Es decir una difusión continúa de imágenes a través de una red local creada desde el portátil. Debido a que el flujo de datos es vía *streaming* es posible que algunos datos o *frames* se pierdan, es por eso que se decide obtener la velocidad del vehículo, haciendo un promedio de las velocidad instantáneas y desechando posible valores errados donde la velocidad encontrada no coincide con la velocidad encontrada en los demás puntos.

En la Figura 21 se observa el diagrama de flujo del desarrollo final, igualmente en el Anexo 3 se encuentra el código completo con cada uno de los algoritmos utilizados y referenciados con más detalle.

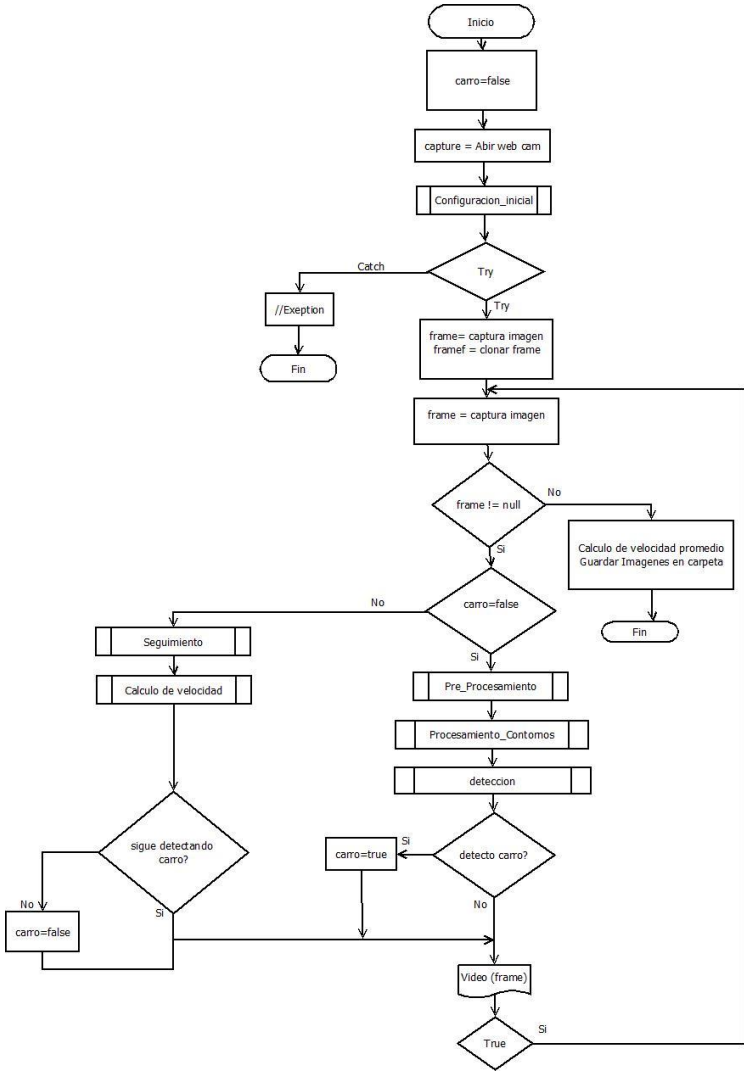


Figura 21. Diagrama de flujo final

8. PRUEBAS

Los resultados completos de las pruebas realizadas se pueden encontrar en el Anexo 2.

8.1. Pruebas en balcón Residencial

Se realizaron varias pruebas desde un balcón residencial. La cámara se encontraba a una distancia aproximada de 10 metros horizontalmente y 4.5 metros de altura. Se utilizó como velocidad máxima 40 km/h para verificar detección de infracción del vehículo.

Tabla 3. Pruebas en balcón Residencial

V_r (km/h)	V_m (km/h)	E_r (%)	Infracción
30	33.26	10.8	NO
40	46.17	16.7	SI
50	52.62	5.2	SI
58	57.7	0.5	SI

Como se puede observar en la Tabla 3, la velocidad obtenida por la aplicación es muy cercana a la real en velocidades altas,

aproximadamente en 58 km/h donde el error relativo es cerca del 0.5 %.



Figura 22. Pruebas de velocidad 50 km/h

La Figura 22 muestra la alerta que se genera en el sistema al detectar un carro a velocidades superiores, en este caso, a 40 km/h. Estas imágenes son almacenadas en una carpeta que indica la fecha, hora y velocidad en la cual paso el vehículo, como prueba de la infracción.

8.2. Pruebas en la Universidad

Estas pruebas se realizaron desde el 4 piso de Ingeniería de la Universidad Pontificia Bolivariana, en donde la cámara encontraba a una distancia aproximada de 6.6 metros horizontalmente y 11.94

metros de altura. Se utilizó como velocidad máxima 30 km/h para verificar detección de infracción del vehículo.

Tabla 4. Pruebas velocidad Universidad

V_r (km/h)	V_m (km/h)	E_r (%)	Infracción
20	21	5	NO
30	28.8	4	NO
35	33.1	5.4	SI
40	54	35	SI
40	42	5	SI

Las velocidades obtenidas mediante la aplicación son generalmente buenas, obteniendo un error relativo aproximadamente del 5 %. En una de las mediciones de velocidad el error fue del 35 %, esto sucedió debido a que algunos *frames* se perdieron cuando intentamos enviar el video en tiempo real al pc desde el celular iPhone4 vía *streaming*, por el sistema hace una detección errónea de la velocidad.

En la Figura 23 se observa la imagen del vehículo desde el 4 piso del bloque de Ingeniería. La velocidad obtenida es de 35 km/h, por lo tanto se encierra en un cuadro rojo, indicando la alerta de infracción por violar el límite de velocidad.



Figura 23. Pruebas de velocidad 35 km/h

9. CONCLUSIONES

El prototipo desarrollado en este trabajo es una herramienta útil para detectar, seguir y obtener la velocidad de los vehículos que transitan por una determinada vía. La aplicación se utiliza realizando una calibración sencilla que depende de la posición de la cámara respecto a la vía, además cuenta con un perfil de bajo costo, que solo requiere de un hardware donde se pueda correr la aplicación, junto a una cámara web, sin necesidad de hacer algún trabajo sobre la vía.

Al realizar las pruebas en dos ambientes y condiciones de posicionamiento de la cámara diferente se encontró un error relativo promedio del 6.5 % por encima de la velocidad real. Dentro de este valor existe un 5 % de incertidumbre dado que la velocidad del vehículo varía entre pequeños rangos. El 1.5 % restante se debe a la precisión de la distancia real de la cámara al punto de captura. Para un futuro se tendrán planes de mejora para precisar la instalación y utilización del sistema.

En algunos casos se identificó pérdida de *frames* en la transmisión vía *Streaming* de los datos tomados desde el Iphone4, al portátil mediante la aplicación *WebCamera*[®], lo cual generó valores de velocidad más elevados que la velocidad real del vehiculó.

Para facilitar el proceso de extracción de características de un objeto en una imagen, es necesario realizar un pre procesamiento usando técnicas de segmentación y operaciones morfológicas, los

cuales ayuda a eliminar ruido y conservando las principales características de forma de los objetos.

Se comprobó que la técnica de extracción de características de alto nivel usando sustracción de fondo fue efectiva a la hora de detectar el vehículo, debido a que se trabajó con una cámara estática, lo cual facilitó la detección de la silueta del vehículo.

La librería OpenCV sobre procesamiento de imágenes, es útil para las soluciones de diversos problemas gracias a la gran cantidad de funciones que posee para el análisis, detección o seguimiento de objetos, sus ventajas sobre otras librerías, es gracias a su código abierto, lo cual reduce costos y evita problemas de propiedad intelectual.

Se utilizó *Netbeans* para programar en *Java*[®], ya que los tiempos de ejecución son más rápidos con esta herramienta, lo que resulta de gran ayuda para este proyecto porque se busca que el procesamiento de imágenes sea en tiempo real.

Para trabajos futuros que complementen este sistema, se puede utilizar una cámara adicional que pueda detectar y capturar la placa del vehículo y así relacionar velocidad con la placa, obteniendo datos más completos para identificar a los infractores.

A este prototipo se le pueden agregar nuevas versiones, como es la adición de una interfaz gráfica de usuario (GUI), lo cual facilitará cambiar algunos valores como son el posicionamiento de la cámara, velocidad a la cual infraccionar, entre otros.

REFERENCIAS

- [1] El Espectador. (2012). *El sector automotor seguirá de plácemes en 2012*. Recuperado el 15 de julio de 2013, de <http://www.elespectador.com/noticias/economia/el-sector-automotor-seguira-de-placemes-2012-articulo-319643>
- [2] BBVA Research. (2011). Situación Automotriz Colombia Recuperado el 9 de julio de 2013, de http://serviciodeestudios.bbva.com/KETD/fbin/mult/1301_SitAutomotriz_Colombia_Ene13_tcm346-364303.pdf?ts=1422013
- [3] Capossello, H. (2011). Referencia conceptos Teoría del Video. Recuperado el 15 de agosto de 2013, de <https://we.riseup.net/assets/65966/guia%20edicion%20I.pdf>
- [4] Universidad Nacional de Quilmes (2005). *Operaciones Morfológicas en imágenes binarias*. Recuperado el 16 de agosto de 2013, de <http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Operaciones%20MorfoI%C3%B3gicas%20en%20Im%C3%A1genes%20Binarias%20-%20parte%201.pdf>
- [5] Bradsky, G. , Kaehler, A. *Learning Opencv*. O' Reilly. Primera Edición (2008).
- [6] Angulo, J., Serra, J. (2003) *Segmentación de imágenes en color utilizando histogramas bi-variables en espacio de color polares*. Recuperado el 18 de septiembre de 2013, de http://cmm.ensmp.fr/~angulo/publicat/AnguloSerra_CyS_05.pdf .
- [7] Maggio, E., Callavaro, A. *Video Tracking: Theory and Practice*. Wiley. UK. Primera Edición (2011).
- [8] STT Medellín. *Sistema Inteligente de Movilidad* (2013). Recuperado el 10 de septiembre de 2013, de http://www.medellin.gov.co/transito/sistema_inteligente_movilidad.html?utm_expId=37837193-0.N-aRCm33TeWEod4cht0dvg.1&utm_referrer=http%3A%2F%2Fwww.medellin.gov.co%2Ftransito%2F
- [9] Medellín cómo vamos. (2013) *Movilidad y espacio público*. Recuperado el 24 de septiembre de 2013, de <http://medellincomovamos.org/movilidad-y-espacio-p-blico>.
- [10] Intef. (2003) *Óptica Geométrica*. Recuperado el 23 de septiembre de 2013, de <http://acacia.pntic.mec.es/jruiz27/elementos.htm>.
- [11] Universidad de Córdoba. (2003) *Departamento de física aplicada*. Recuperado el 25 de septiembre de 2013, de <http://rabfis15.uco.es/lvct/tutorial/39/refraccion.htm>.
- [12] Intef. (2012) *Fenómenos ondulatorios* Recuperado el 24 de septiembre de 2013, de http://recursostic.educacion.es/secundaria/edad/4esofisicaquimica/4quincena1/4q11_contenidos_2a.htm
- [13] Jiménez C. E (2009) *Medición de distancias por medio de procesamiento de imágenes y triangulación, haciendo uso de cámara de video* Recuperado el 30 de septiembre de 2013, de http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/jimenez_c_e/portada.html
- [14] Grupo de Tratamiento de Imágenes. (2002) *Geometría del proceso de formación de la imagen* Recuperado el 12 de octubre de 2013, de <http://iie.fing.edu.uy/investigacion/grupos/gti/cursos/egvc/material/tema-3.pdf>
- [15] Carles Mitja (2013) *La cámara del iPhone4* Recuperado el 20 de septiembre de 2013, de <http://carlesmitja.net/2012/06/22/la-camara-del-iphone4/>
- [16] Java®. (2013) *Tecnología Java y sus necesidades* Recuperado el 29 de septiembre de 2013, de http://www.java.com/es/download/faq/whatis_Java.xml
- [17] OpenCv Wiki. (2013) *Introducción* Recuperado el 29 de septiembre de 2013, de <http://opencv.org/>

AUTOR



Héctor Iván *ARISTIZABAL GOMEZ*, nacido en Cali, Colombia. Estudiante de décimo semestre de Ingeniería Electrónica de la Universidad Pontificia Bolivariana. Diplomado en Gestión de Proyectos de Ingeniería de la UPB (2013). Curso de controladores programables de la UPB (2012). Semillero de dispositivos móviles de la UPB (2011).

ANEXOS

ANEXO 1. GUÍA INSTALACION OPENCV EN JAVA®

ANEXO 2. RESULTADOS DE PRUEBAS DE VELOCIDAD

ANEXO 3. CODIGO DE DESARROLLO FINAL