

PROCESAMIENTO DIGITAL DE IMÁGENES PARA ESTIMACIÓN DE VELOCIDAD Y MEDICIÓN DE FLUJO VEHICULAR

DIGITAL IMAGING PROCESSING FOR SPEED DETERMINATION AND TRAFFIC FLOW MEASUREMENT

Pedro Freddy Huamani Navarrete¹ y José Luis Rojas Vara²

RESUMEN

En este artículo se propone una técnica de procesamiento de imágenes para la medición del flujo vehicular y estimación de la velocidad en un punto determinado de la avenida Paseo de la República de la ciudad de Lima. Para determinar el flujo vehicular y la estimación de la velocidad, se utilizó una operación de filtrado digital no lineal, una manipulación del histograma, una operación de segmentación y operadores morfológicos de dilatación, erosión, apertura, cerradura y etiquetado de imágenes binarias. Con el etiquetado de las imágenes, se consiguió representar cada vehículo por un objeto de color blanco. Esto facilitó la obtención del centro de masa de los objetos etiquetados en cada imagen binaria para, luego, realizar la comparación respecto a una línea de referencia. Esta técnica fue implementada en el GUI del *software matlab* y para ello se utilizó un grupo de 12 videos digitales capturados en horario diurno y por periodos de tiempo limitados. En cuanto a los resultados, el flujo vehicular promedio alcanzado tuvo un porcentaje de acierto del 90%, mientras que en el caso de la velocidad promedio aproximada se halló un valor no superior a los 80 km/h.

Palabras clave: Estimación de velocidad, flujo vehicular, Operaciones Morfológicas, Etiquetado de Imágenes, GUI del *Matlab*.

ABSTRACT

This article provides an image processing technique for speed determination and traffic flow measurement at a given point at Paseo de la Republica Avenue in Lima City, Peru. In order to do so we work with a nonlinear digital filtering operation, a histogram, an operation of segmentation and morphological operators of dilatation, erosion, opening, lock and binary image labeling. Image labeling made possible to represent each vehicle by a white color object. This helped to obtain the mass center of tagged objects in each binary image, and, at that point, to relate these tagged objects to a reference line. This technique was implemented by means of Matlab GUI software. Likewise, 12 digital videos caught in the target point for limited periods of times during day time were used. Regarding results average traffic flow had a 90% success rate and average speed value did not exceed 80 km/h.

Keywords: Speed determination, traffic flow measurement, Morphologic Operations, Image Labeling, Matlab GUI software.

1 Ricardo Palma, Av. Benavides s/n, Lima, Perú. <phuamani@urp.edu.pe>

2 Universidad Tecnológica del Perú, Av. 28 de Julio 826, Lima, Perú. <jlrojas23@gmail.com>

1. INTRODUCCIÓN

Con el transcurrir de los años, la población mundial ha tenido un aumento exponencial y, con ello, el uso del transporte motorizado se ha intensificado en diferentes localidades del planeta. Aunque aún existen muchas ciudades que todavía no se han anticipado a este crecimiento vertiginoso, se manifiesta un alto tráfico en las principales calles y avenidas hasta la fecha. En nuestro país, particularmente en las grandes ciudades como la capital Lima, el tránsito se hace insoportable en determinadas horas de algunos días de la semana. Por ello, es necesario contar con un medidor de flujo vehicular que permita que las instituciones de transporte puedan precisar los lugares donde es necesario construir vías alternas para que ayude a descongestionar el tráfico en algunos puntos críticos de la ciudad de Lima. Asimismo, para posteriores trabajos de investigación, dicho monitoreo de tráfico ayudará a establecer nuevos tiempos para los cambios de luces de los semáforos durante determinadas horas del día, lo que permitiría el paso a las llamadas olas verdes y dejaría de lado las olas rojas. Es así que este artículo propone la implementación de un algoritmo computacional en el *software Matlab* con ayuda del *Simulink* y del *Toolbox Image Processing* para la obtención y procesamiento de imágenes contenidas en un video digital. Así, se podrá conseguir la obtención del flujo vehicular y la estimación de la velocidad en un determinado punto de la ciudad.

2. MATERIALES PARA EL PROCESAMIENTO DIGITAL

Para el desarrollo de esta investigación se utilizaron ciertas técnicas matemáticas de procesamiento digital de imágenes. Algunas fueron implementadas algorítmicamente en el *software Matlab*, y otras pertenecieron a las librerías *Image Processing* y *Video and Image Processing Blockset* del *Simulink*.

Proceso de lectura de video digital

En este artículo, se realiza el procesamiento de archivos de video en tiempo *off line*, esto quiere decir, que el procesamiento no se realizó en tiempo real sino sobre un conjunto de archivos de video multimedia, los cuales fueron capturados a través de una cámara fotográfica digital por un periodo de tiempo igual a 10 segundos cada uno en formato MP4. La cámara posee una resolución por fotograma o cuadro igual a 720x1280 píxeles, una codificación de 24 bits/píxel y una velocidad de 30 cuadros/segundo. Sin embargo, por la amplia resolución presentada en el formato MP4, se convirtió tal formato en uno de menor resolución: formato AVI. Por ello, en este nuevo formato, el video conservó la codificación de 24 bits/píxel, pero disminuyó el número de cuadros/segundo a 25, así como la resolución a 240x320 píxeles. Una vez hecha la conversión, se utilizó la librería *Video and Image Processing Blockset*

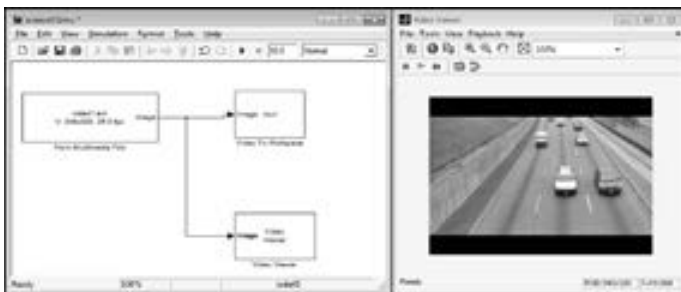


Fig. 1. Proceso de lectura de archivo AVI con Simulink.

del *Simulink* para realizar la lectura del archivo multimedia con la sub librería *From Multimedia File* de la opción *sources*, y las sub librerías *Video to Workspace* y *Video Viewer* de la opción *sinks*. A continuación, la figura 1 muestra el proceso de lectura del archivo AVI.

Manipulación del histograma

El proceso de manipulación del histograma tiene como objetivo principal mejorar el contraste de una imagen y facilitar la segmentación. Por ello, se eligió el método de manipulación del histograma por extracción de raíz cuadrada, porque con el resultado de tal operación se consiguió acentuar más los efectos que se producen en el momento de aclarar la imagen (González, 2006). Ver la figura 2. Por lo tanto, este método consistió en normalizar los píxeles de cada cuadro del video capturado. Es decir, en caso de encontrarse en el rango dinámico de 0 a 255, se tuvo que dividir cada cuadro entre el valor 255. Para este caso, no fue necesario porque los píxeles ya se encontraban entre 0 y 1. Luego, con estos se procedió a extraer la raíz cuadrada a cada uno de ellos en el intervalo de 0 a 1. Finalmente, se retornó al rango dinámico original, situado entre 0 y 255, acompañado de una operación de redondeo para así contar con píxeles enteros con facilidad de ser codificados con 8 bits. Este tipo de operación permitió que el conjunto de píxeles con valores centrales al rango dinámico alcancen nuevos valores, pero con mayor tendencia al color blanco. La siguiente expresión matemática muestra el proceso de extracción de la raíz cuadrada a cada uno de los píxeles de la imagen a manipular.

$$I_{\text{TRANSFORMADA}}(i, j) = \text{REDONDEO} \left(\frac{I_{\text{ORIGINAL}}(i, j)}{255} \right) * (255)$$

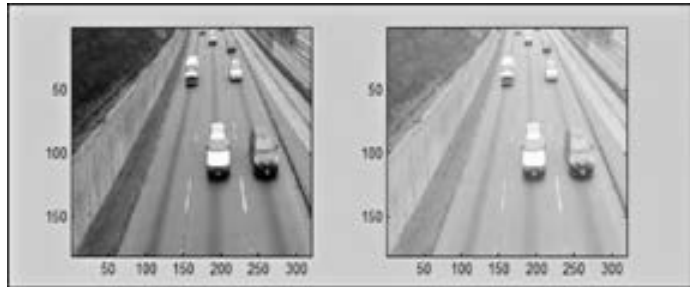


Fig. 2. Resultado de la manipulación del histograma.

La siguiente expresión matemática muestra el proceso de extracción de la raíz cuadrada a cada uno de los píxeles de la imagen a manipular.

Binarización de imágenes

Representa la transformación de una imagen a color o con tonos de gris a una nueva imagen que contiene solo píxeles de color blanco y negro. De esta manera, estos son presentados solamente por los números: 0 y 1. Ver la figura 3.

Este proceso de conversión se realiza a través de la siguiente expresión matemática, donde el umbral es seleccionado adecuadamente dentro del rango dinámico y con ayuda del histograma de la imagen digital a binarizar.

$$I_{\text{binaria}}[m, n] = \begin{cases} 1: & I_{\text{gris}}[m, n] \geq \text{umbral} \\ 0: & I_{\text{gris}}[m, n] < \text{umbral} \end{cases}$$

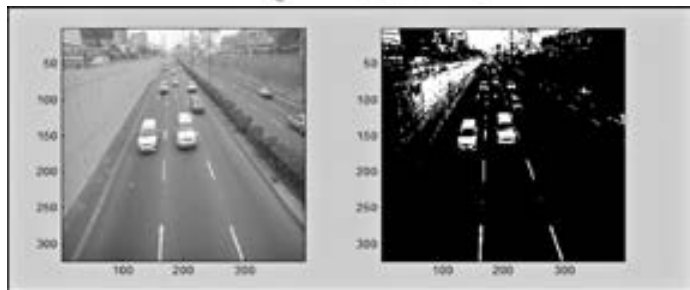


Fig. 3. Proceso de binarización de una imagen digital.

Transformaciones morfológicas

Se define como el conjunto de transformaciones que recibe una imagen digital, basado en un elemento estructurante morfológico y representado a través de una matriz. Esta matriz se compone de

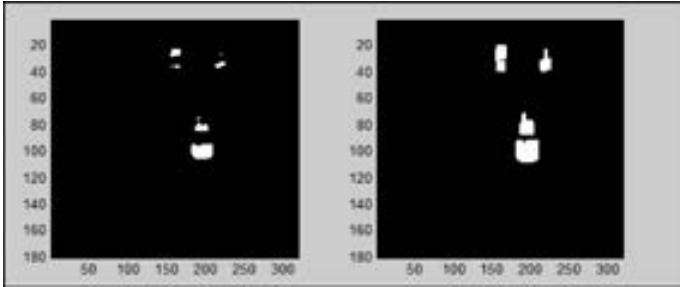


Fig. 4. Resultado de aplicar la operación morfológica de dilatación.

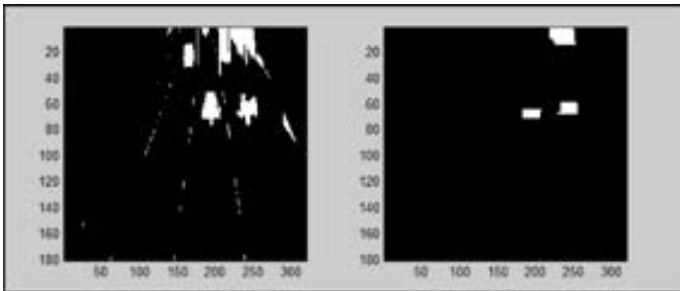


Fig. 5. Resultado de aplicar la operación morfológica de apertura.

te en forma de una recta horizontal. Como la apertura es el procedimiento secuencial de las operaciones de erosión y dilatación, en la figura 5 se observa el proceso, por el cual se separan conexiones finas y se remueven protuberancias muy delgadas.

3. PROCEDIMIENTO PARA EL PROCESAMIENTO DIGITAL

El procedimiento realizado para el procesamiento digital de los cuadros contenidos en un video se basó en la utilización secuencial de las técnicas anteriormente descritas.

Determinación del flujo vehicular

Se determinó a través del histograma realizado a cada componente de color: *red*, *green* y *blue*, donde el componente *blue* representaba un color menos común en promedio. Por ese motivo, solo se trabajó con este componente en cada cuadro o imagen del video digital capturado. Luego, se recortó cada cuadro del componente *blue* a una resolución de 180x320 píxeles con la finalidad de suprimir la franja negra que surgió al momento de realizar la conversión del formato MP4 a AVI. El siguiente paso consistió en aplicar un filtrado de mediana sobre cada bloque de 3x3 píxeles de cada cuadro del video digital. Este procedimiento permitió disminuir el ruido existente sin llegar a deteriorar los bordes y detalles abruptos de los objetos de interés. Seguidamente, se procedió a manipular el histograma a través de la extracción de la raíz cuadrada tal como se mencionó en la sección anterior.

Como los cuadros del video presentaban dos zonas que no son de interés (vía auxiliar de la izquierda y autopista contraria), se procedió a la implementación de una plantilla construida en base a dos triángulos rectángulos formados por píxeles 0 y la parte central por píxeles 1. Para ello, se realizó un algoritmo computacional que permitió construir tal plantilla en base a dos pendientes diferentes, que posteriormente dio origen a la plantilla para utilizarse en el proceso de segmentación de cada cuadro

1 y 0 que forman una recta, un cuadrado, un círculo, o alguna forma arbitraria. Entre algunas de las transformaciones morfológicas más utilizadas, encontramos a la dilatación, erosión, cerradura y apertura. A continuación, la figura 4 muestra el resultado de una operación de dilatación sobre una imagen binarizada que hace uso de un elemento estructurante en forma de una recta vertical. Eso permite dilatar o aumentar los píxeles blancos en ese sentido, mientras que la figura 05 muestra el resultado de una operación de apertura que hace uso de otro elemento estructuran-

del video digital. A continuación, la figura 6 muestra el resultado de la construcción de la plantilla para el proceso de segmentación a partir de un producto.

La intensidad de gris, producto de la luz natural, no fue uniforme porque el video fue adquirido en la intemperie. Por tal razón,

se optó por utilizar un filtrado *Top-Hat* que, asimismo, es un tipo de transformación morfológica. Este filtro tiene como propósito resaltar los detalles claros situados en zonas de poco contraste y para ello se utilizó un elemento estructurante del tipo recta en forma vertical. Luego, se continuó con el proceso de binarización que hizo uso de dos umbrales. El primero fue de 26 para las primeras 100 filas y el segundo fue de 52 para las últimas 80 filas.

Posteriormente, se continuó con la transformación morfológica de cerradura con un elemento estructurante del tipo recta vertical con 20 píxeles de longitud y, luego, se utilizó la transformación de apertura con un elemento estructurante del tipo recta horizontal y también con 20 píxeles de longitud.

Seguidamente, se aplicó la transformación morfológica de dilatación con otro elemento estructurante del tipo recta vertical y también con 20 píxeles de longitud. Esta combinación se obtuvo a través de un conjunto de pruebas y ensayos, lo que permitió identificar cada vehículo por un conjunto de píxeles blancos, tal como se observa en la figura 7.

Una vez lograda la segmentación de cada vehículo, se optó por trazar una línea de referencia en la fila 60 de cada cuadro.

Esto tuvo como propósito contar con una guía para la cuenta de vehículos. Es decir, se procedió a contabilizar el número de objetos de color blanco que atravesaron la línea de referencia, pero en función a un tiempo determinado y a una cantidad específica de cuadros.

Según el análisis realizado, no fue necesario procesar los 25 cuadros por segundo.

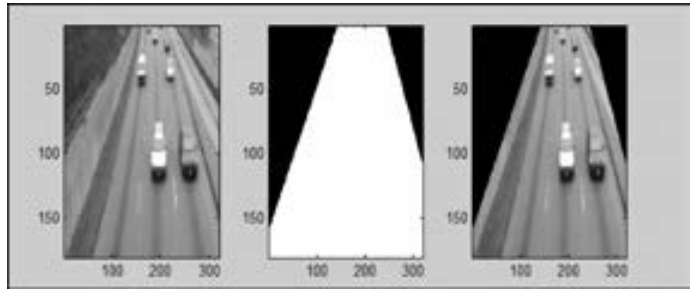


Fig. 6. Resultado de la construcción de la plantilla para la segmentación.

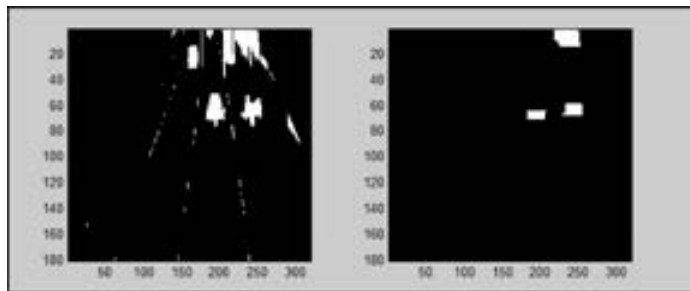


Fig. 7. Proceso de transformación morfológica para segmentar los vehículos.

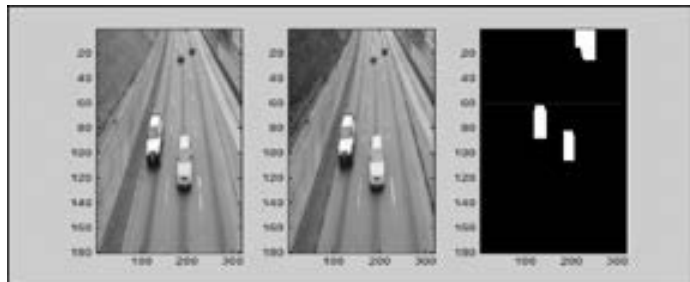


Fig. 8. Línea de referencia para determinar el flujo vehicular.

4. RESULTADOS

Las pruebas y los resultados obtenidos después de realizar el procesamiento a un conjunto de 12 videos digitales, que fueron capturados en diferentes horarios y días de la semana entre lunes y viernes, se muestran en las siguientes tablas. Además, por la cantidad de información que se maneja al momento de digitalizar un video, aún encontrándose en formato AVI, solo se utilizaron archivos con 20 segundos de duración. A continuación, se muestran los resultados obtenidos en cuanto al flujo vehicular por cada video procesado.

TABLA 1. RESULTADOS DE LA VELOCIDAD PROMEDIO APROXIMADA SEGÚN EL HORARIO

	VIDEOS	FECHA	HORA	VELOC. PROMEDIO APROX.
01	Video_01.avi al Video_06.avi	Entre el lunes 18 de agosto y jueves 04 de setiembre de 2014.	10:30 am	78.8 Km/hora
02	Video_07.avi al Video_12.avi	Entre el viernes 05 de setiembre y viernes 19 de setiembre de 2014.	02:45 pm	72.0 Km/hora

TABLA 2. RESULTADOS DE LA VELOCIDAD PROMEDIO APROXIMADA SEGÚN EL DÍA LABORABLE DE LA SEMANA

	VIDEOS	DÍA LABORABLE	VELOC. PROMEDIO APROX.
01	Video_01.avi y Video_08.avi	Lunes	78.1 Km/hora
02	Video_02.avi, Video_04.avi y Video_09.avi	Martes	71.8 Km/hora
03	Video_03.avi, Video_05.avi y Video_10.avi	Miércoles	77.7 Km/hora
04	Video_06.avi y Video_11.avi	Jueves	75.1 Km/hora
05	Video_07.avi y Video_12.avi	Viernes	69.9 Km/hora

5. CONCLUSIONES

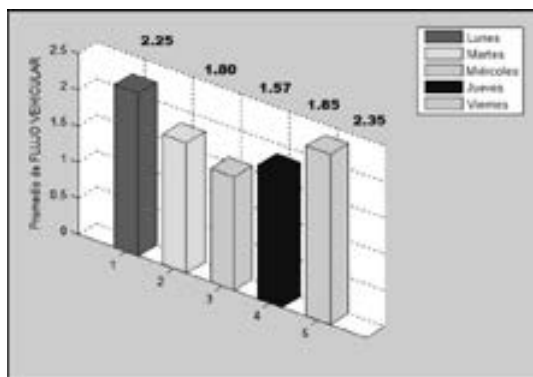


Fig. 9. Promedio de flujo vehicular.

Esta técnica matemática hizo uso de una combinación y secuencia de varias herramientas de procesamiento digital de imágenes que se realizaron en la plataforma del *software Matlab*. Asimismo, cabe mencionar que este software posee un *Toolbox* con un conjunto de funciones relacionadas al tratamiento digital de imágenes, lo cual facilitó de cierta forma la programación de algoritmos complementarios para determinar el flujo vehicular. Sin embargo, tales algoritmos fueron utilizados en el orden correcto y en la forma adecuada, lo que permitió verificar previamente uno a uno los videos empleados para este proyecto de tesis, antes de proceder a la implementación final de la interfaz gráfica desarrollada en el propio *software Matlab*. En resumen, la técnica empleada partió de la lectura del archivo de video digital para luego continuar con una operación de recorte y conversión; luego, se siguió con un filtrado de mediana y manipulación del histograma, para después proseguir con



Fig. 10. Medidor de flujo vehicular

principalmente, por encontrar vehículos de un color muy próximo al asfalto de la vía donde se realizó el procesamiento.

En relación a la determinación de la velocidad promedio aproximada de los vehículos que circulan por un punto de la Vía Expresa y Paseo de la República de la ciudad de Lima, se concluye que dicho cálculo presentó ciertos errores, sobre todo cuando el algoritmo de segmentación por etiquetado mostró uno que otro caso de un automóvil fantasma. Para esto, es necesario realizar eficazmente la técnica de segmentación, donde se aísla o ignora todo aquello que no pertenece a la zona de interés. Asimismo, no fue posible verificar el grado de error obtenido en la obtención de la velocidad promedio, debido a que resultó imposible obtener el valor real de velocidad a partir de una observación directa del video capturado. Por lo tanto, se aclara que la técnica empleada para determinar la velocidad promedio se basó en un cálculo de comparación de regla de tres. De la misma manera, se concluye que no se encontró ningún exceso de velocidad en el conjunto de archivos de videos analizados; es decir, no se llegó a encontrar vehículos con una velocidad superior a 80 Km/hora, y los resultados de las velocidades promedio aproximadas se observan según el horario y el día de la semana indicadas en la tabla. Esta deducción quizás se deba a que, hoy en día, la Policía Nacional del Perú supervisa la velocidad de los automóviles constantemente, desde los puentes peatonales, y utiliza un equipo de medición de velocidad basado en un sistema radar. De esta forma, se asume que los conductores toman conciencia y evitan sobrepasar tal límite de velocidad. Asimismo, es necesario aclarar que, en este trabajo de tesis, se utilizó una muestra pequeña de video digital en relación al tránsito existente por dicha zona, por lo cual no se puede afirmar completamente que la velocidad de un vehículo nunca superará los 80 km/hora.

6. REFERENCIAS

- Cabello Hernández, V. (2010). *Reconocimiento Inteligente de Flujo Vehicular basado en el Procesamiento de Imágenes*. (Tesis de Ingeniero Electrónico, Universidad Michoacana, San Nicolás, Hidalgo-México). Recuperado de <http://bibliotecavirtual.dgb.umich.mx:8083/jspui/bitstream/123456789/5046/1/CONOCIMIENTOINTELIGENTEDEFUJOVEHICULARBASADOENELPROCESAMIENTODEIMANES.pdf>

- Calderón, F., y Urrego, G. (2008). *Conteo Automático de Vehículos*. (Tesis de Ingeniero Electrónico, Pontificia Universidad Javeriana de Bogotá, Colombia). Recuperado de <http://opencvjaveriana.wikispaces.com/file/view/Conteo+Automatico+De+Vehiculos.pdf>
- Chávez, A. (2007). *Algoritmo que permite el ingreso de vehículos a una vía de alto tráfico y doble sentido, basado en procesamiento de imágenes*. (Tesis de Ingeniero Electrónico, Pontificia Universidad Católica de Perú, Perú). Recuperado de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/224/CHAVEZ_CORDOVA_ALEX_ALGORITMO_INGRESO_VEH%C3%8DCULOS.pdf?sequence=1
- Cuevas, E., Zaldívar, D., y Pérez, M. (2010). *Procesamiento Digital de Imágenes usando Matlab y Simulink*. México: Alfaomega.
- González, E. (2004). *Descriptores de Fourier para Identificación y Posicionamiento de Objetos en Entornos 3D*. XXV Jornadas de Automática en Cuba. Recuperado de <http://www.ceautomatica.uji.es/old/actividades/jornadas/XXV/documentos/140-liezgarthg.pdf>
- González, M., Martínez de Pisón, F., Pemía, V., Alba, F., Castejón, M., Ordieres, J., & Vergara, E. (2006). *Técnicas y Algoritmos Básicos de Visión Artificial*. España: Editorial de la Universidad de La Rioja. Servicio de Publicaciones.
- González, R., Woods, R., and Eddins, S. (2004). *Digital Image Processing using Matlab*. New Jersey: Prentice Hall.
- González, R., Woods, R. and Eddins, S. (2004). *Digital Image processing using MATLAB*. (s.c.). Dorling Kindersley.
- Granados, A. (2006). *Medición de flujo vehicular basado en visión artificial*. XI Simposio de tratamiento de señales, imágenes y visión artificial. Stsiva 2006, Recuperado de http://gdsproc.com/congreso/granados_stsiva_2006.pdf
- Jain, A. (1989). *Fundamentals of Digital Image Processing*. New Jersey: Editorial Prentice-Hall, Inc.
- Marqués, F., e Vieira, H. (1999). *Processamento Digital de Imagens*. Brasil: Brasport.
- Pajares, G., y De la Cruz, J. (2007). *Visión por Computador: Imágenes digitales y aplicaciones*. (2da ed.). México: RA-MA S.A.
- Prieto, M. (2010). *Algoritmo para conteo vehicular en tiempo real con base en franjas de interés*. (Tesis de Grado para optar el Título de Ingeniero Electrónico, Pontificia Universidad Javeriana de Bogotá, Colombia). Recuperado de <http://repository.javeriana.edu.co/bitstream/10554/7034/1/tesis480.pdf>
- Vieira, H., Marqués, F. (1999). *Processamento Digital de Imagens*. Brasil: Brasport.