

# Adquisición de variables de tráfico vehicular usando visión por computador

## Acquisition of Vehicular Traffic Variables using Computer Vision

7  
técnica

Recibido 31 de Marzo de 2009, modificado 20 de octubre de 2009, aprobado 26 de octubre de 2009.

### **Germán Enrique Urrego**

Ingeniero Electrónico. Pontificia Universidad Javeriana. Bogotá D.C., Colombia.  
gurrego@javeriana.edu.co ✉

### **Francisco Carlos Calderón**

Ingeniero Electrónico. Pontificia Universidad Javeriana. Bogotá D.C., Colombia.  
calderonf@javeriana.edu.co ✉

### **Alejandro Forero**

M.Sc. Profesor Asistente, Departamento de Electrónica, Facultad de Ingeniería, Pontificia Universidad Javeriana. Bogotá D.C., Colombia.  
alejandro.forero@javeriana.edu.co ✉

### **Julián Armando Quiroga**

M.Sc. Profesor Asistente, Departamento de Electrónica, Facultad de Ingeniería, Pontificia Universidad Javeriana. Bogotá D.C., Colombia.  
quiroga.j@javeriana.edu.co ✉

#### **PALABRAS CLAVES**

Ingeniería de tráfico, visión por computador.

#### **KEY WORDS**

Computer vision, traffic engineering.

#### **RESUMEN**

El conteo de vehículos es una tarea fundamental en los aforos de tráfico, tanto para el diseño de nuevas vías como para el control y la optimización del tráfico urbano. En ciudades como Bogotá, este conteo actualmente se realiza en forma manual, pero puede hacerse automáticamente empleando técnicas de visión por computador. El presente artículo muestra el desarrollo de un algoritmo de detección y seguimiento de vehículos en tiempo real a partir de video, el cual permite el conteo de vehículos en la vía y la estimación del volumen de tráfico o velocidad promedio.

#### **ABSTRACT**

The count of vehicles is a key task in traffic gauging, as it provides information for designing new roads, and for the control and optimization of urban traffic. Nowadays in cities like Bogota, vehicle counting is done manually, but it can also be done automatically using computer vision techniques. This article presents the development of an algorithm for detecting and tracking vehicles in real time from video. The algorithm allows to count vehicles on the road and to estimate traffic volume or average speed.

## INTRODUCCIÓN

Las ciudades de gran tamaño se asocian con congestiones de tráfico vehicular, ya sea porque la infraestructura no creció o porque no se desarrolló de manera adecuada con respecto al aumento en el número de vehículos. En general, la opción para mejorar la movilidad en una ciudad es optimizar la manera como se usan los recursos de infraestructura con que se cuenta. Para ello, es indispensable adquirir y almacenar de manera adecuada la información en tiempo real, la cual permita actualizar el modelo de tráfico que se pretenda optimizar. Bajo esta premisa, se desarrollan proyectos como el de semaforización inteligente. Pero el semáforo es sólo el dispositivo que entrega la información final: antes de llegar a éste, se debe definir una metodología y contar con los medios que permitan la captura de datos que continuamente alimenten el modelo.

La ecuación (1) se utiliza para modelar el tráfico en función de la cantidad de vehículos en la vía y de su velocidad promedio. Este modelo da información de la cantidad de vehículos por longitud de un segmento de la vía.

$$\text{Densidad (veh / km)} = \frac{\text{Volumen(veh / h)}}{\text{Velocidad(km / h)}} \quad (1)$$

Este modelo puede ser calculado y actualizado en tiempo real al usar los medios tecnológicos necesarios. En el caso de las ciudades colombianas, actualmente la información del tráfico es recolectada manualmente y se utiliza para el diseño de vías y la implementación de normas para mejorar la movilidad, como por ejemplo el “pico y placa”.

En los últimos años, se ha empleado la visión por computador para capturar la información del tráfico vehicular de forma automática [1]. Las soluciones basadas en visión tienen varias ventajas: permiten guardar un registro de la historia del tráfico en la vía, su instalación es sencilla comparada con otras alternativas que requieren sensores debajo de la vía y, además, brindan la posibilidad de monitorear más de un carril simultáneamente [2].

Existen diferentes enfoques para la detección y seguimiento de vehículos. En [3] los vehículos son detectados a partir de los bordes extraídos de la imagen. En [4] una diferencia entre cuadros consecutivos del video es computada para localizar los vehículos en movimiento y el seguimiento es realizado al utilizar momentos estadísticos de los objetos encontrados. Sin embargo, estos dos métodos no permiten obtener con precisión la localización del vehículo, lo que dificulta hacer el seguimiento. Otros métodos detectan los vehículos a partir de un modelo del fondo de la escena, con lo que se consigue una localización más precisa de los mismos [5] [6]. En [5] el modelo del fondo es obtenido por medio de la ponderación de cuadros del video y el seguimiento es realizado modelando cada vehículo como un rectángulo. En [6] se utiliza una combinación de funciones Gaussianas para modelar el valor temporal de los píxeles pertenecientes al fondo. Otra opción para realizar el seguimiento utilizando la geometría de la escena se explora en [7], este método requiere una calibración de la cámara. Estos algoritmos de detección y seguimiento pueden verse afectados por el bajo contraste entre los vehículos y la escena.

Con el fin de mejorar la detección de vehículos, en este trabajo se combinan la segmentación de objetos a partir de un modelo del fondo, basado en [8] y [9], y una proyección de los bordes de la imagen para localizar con precisión los vehículos de la escena. Para que el sistema de conteo pueda funcionar en tiempo real, el algoritmo de seguimiento utiliza los momentos de área de los contornos de los vehículos encontrados, los cuales pueden ser computados eficientemente.

En el presente trabajo, se describe el desarrollo de un conjunto de algoritmos que permiten la detección y seguimiento de vehículos utilizando visión por computador, con el fin de realizar el conteo automático y, con éste, estimar el volumen de vehículos en la vía.

Este artículo está organizado como sigue. En la Sección 2 se hace una descripción general del algoritmo de conteo automático. En las Secciones 3 y 4 se presentan de manera detallada respectivamente los algo-

ritmos de detección del primer plano y de detección y seguimiento de vehículos. Los resultados obtenidos son descritos en la Sección 5. Finalmente, en la Sección 6 se realiza la conclusión del trabajo realizado.

### DESCRIPCIÓN GENERAL DEL ALGORITMO

Los componentes principales del algoritmo de detección y conteo de vehículos se ilustran en la Figura 1 y se describen a continuación.

Primero, se define una región de interés (RDI) en el video para disminuir la cantidad de información a procesar y, por ende, el tiempo de proceso; adicional a esto, se hace una corrección de perspectiva para reducir el error producto de una inclinación muy grande en el ángulo de captura de la cámara. Luego, para realizar la detección de los vehículos en la RDI, se optó por una estrategia de estimación del primer plano, pixeles pertenecientes a los vehículos, a partir de la comparación con un modelo del fondo de la escena.

El modelo del fondo es estimado periódicamente y este modelo es utilizado para encontrar los pixeles que no forman parte del fondo; después, los pixeles seleccionados son clasificados como vehículos u otro tipo de objeto móvil, de acuerdo a sus características geométricas. Las propiedades de cada vehículo en la escena son determinadas a partir de los momentos centrales del conjunto de pixeles que lo conforman, calculados por filas y por columnas. Posteriormente, la localización del vehículo es encontrada a partir del primer momento (centro de masa) y el tamaño del

mismo es determinado en función del segundo momento (varianza).

Los dos primeros momentos centrales de cada objeto de interés son utilizados para realizar el seguimiento cuadro a cuadro dentro de la RDI, con el fin de no contar múltiples veces un mismo vehículo. El algoritmo se implementó usando las bibliotecas de OpenCV [10], disponibles en [11].

### DETECCIÓN DEL PRIMER PLANO

La estimación del primer plano consiste en determinar qué pixeles corresponden a objetos móviles en la escena. Puede realizarse mediante la comparación de un modelo del fondo con cada cuadro del video. En [8] y [9] se presenta el método *FGDStatmodel* que realiza una clasificación bayesiana de los pixeles de la imagen, como pertenecientes al fondo o al primer plano, en función de su comportamiento temporal. No obstante, debido al bajo contraste que puede existir entre el color de los vehículos y la vía (usualmente pavimento ó concreto), esta estrategia basada en la intensidad de los pixeles puede presentar errores en la estimación del primer plano. En la Figura 2(C) se muestra un error en la segmentación debido al bajo contraste: uno de los vehículos es segmentado incorrectamente; este error podría producir un doble conteo del vehículo.

El algoritmo de detección propuesto incorpora al modelo obtenido por *FGDStatmodel* la información de los bordes de la escena. La Figura 2(D) muestra un ejemplo de la información aportada por los bordes.

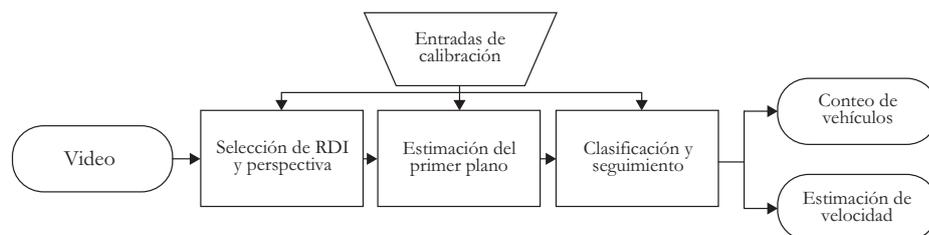


Figura 1. Diagrama en bloques del algoritmo.



Figura 2. (A) Cuadro del video, (B) primer plano obtenido por el algoritmo propuesto, (C) primer plano mediante el algoritmo *FGDSstatmodel* y (D) bordes del primer plano utilizando fondo por *Media Aumentada*.

En la Figura 2(B) se muestra el resultado del algoritmo desarrollado. Esta modificación permite detectar vehículos que presentan bajo contraste con el fondo y superar cambios homogéneos en la iluminación; en la Figura 2 se ve cómo, independiente del color del carro, 2(A) es detectado en 2(D).

#### USO DE LA MEDIA AUMENTADA

Con el objetivo de unir los segmentos que pertenecen a un mismo vehículo y quedaron desligados después de restar el modelo del fondo usando *FGDSstatmodel*, se desarrolló el algoritmo de *Media Aumentada*, el cual encuentra una imagen del fondo que converge rápidamente a vías. A continuación se describe en detalle este algoritmo.

Sean  $I_1(x, y, n)$ ,  $I_2(x, y, n)$  y  $I_3(x, y, n)$  las componentes de rojo, verde y azul, del cuadro de video  $n$ , donde  $x, y$  representan la coordenada en píxeles dentro de dicho cuadro. Sea  $W_i(x, y, n)$  el peso asociado a cada pixel del cuadro de video  $I_i(x, y, n)$ ; dicho peso asigna un mayor valor a los píxeles con color similar al color del pavimento esperado, como se muestra en la ecuación (2).

$$W_i(x, y, n) = \begin{cases} 5 & t_i^L < I_i(x, y, n) < t_i^H \text{ y } \delta(I_i(x, y, n)) = 1 \\ 1 & \text{otro caso} \end{cases}, \quad i=1,2,3 \quad (2)$$

Los valores  $t_i^H$  y  $t_i^L$  representan los valores mínimos y máximos de color en el plano  $i$  del pavimento observado y segmentado manualmente en los videos de entrenamiento. Para separar el conjunto de grises que se asemeja al pavimento, se define una dispersión de los

colores entre cada uno de los planos menor al 10% del espacio. La dispersión máxima observada en el conjunto de entrenamiento fue 20, es decir, el 7,8% del espacio de representación (256 con 8 bits por plano de color) y este fue el valor con el que se trabajó.  $\delta(I_i(x, y, n))$  se define en la ecuación (3).

$$\delta(I_i(x, y, n)) = \begin{cases} 1 & |I_i(x, y, n) - I_j(x, y, n)| < 20, \quad \forall i \neq j \in \{1,2,3\} \\ 0 & \text{otro caso} \end{cases} \quad (3)$$

El valor ponderado de los píxeles de la imagen, hasta el cuadro  $n$ , se muestra en la ecuación (4)

$$I_i^{paa}(x, y, n) = I_i^{paa}(x, y, n-1) + W_i(x, y, n) \cdot I_i(x, y, n), \quad i=1,2,3 \quad (4)$$

El peso acumulado de ponderación de cada componente de color se muestra en la ecuación (5)

$$m_i(x, y, n) = \sum_{k=1}^n W_i(x, y, k) = m_i(x, y, n-1) + W_i(x, y, n), \quad i=1,2,3 \quad (5)$$

El modelo del fondo,  $F_i^{MA}(x, y)$ , es obtenido al hacer el promedio de los píxeles, como se muestra en la ecuación (6); se calcula para  $n$  mayor o igual a 1, de forma que el denominador nunca sea cero.

$$F_i^{MA}(x, y) = \frac{I_i^{paa}(x, y, n)}{m_i(x, y, n)}, \quad i=1,2,3 \quad (6)$$

La convergencia de este modelo es más rápida que la convergencia de los algoritmos presentados en [8], [9] y [7], y con un menor tiempo de compilación para el mismo número de cuadros. En la Figura 3 se ilustran las diferencias entre el algoritmo desarrollado, media aumentada y otros modelos de fondo.

*FGDStatmodel* MODIFICADO

El segundo modelo del fondo es obtenido de acuerdo a [8], sin embargo, en nuestro algoritmo el modelo sólo es actualizado cuando se cumplen las siguientes dos condiciones:

*Condición 1.* El porcentaje de píxeles que contienen movimiento (variación temporal superior a un umbral específico) es inferior a 0,5%, que es un valor obtenido mediante observación sobre los videos de entrenamiento.

*Condición 2.* El porcentaje de píxeles de los bordes del primer plano (diferencia entre las imágenes de bordes del fondo y los bordes del cuadro actual) es inferior a 0,2%, que es un valor obtenido mediante observación sobre los videos de entrenamiento.

Por medio de estas dos condiciones se busca que el fondo sólo sea actualizado si la variación temporal de los píxeles de la escena es baja, característica ideal de los píxeles del fondo, y no se presentan objetos en el primer plano, para evitar el efecto la inclusión de vehículos estáticos.

La Figura 3 muestra distintas imágenes de fondos obtenidos por los dos algoritmos usados y por el algoritmo *GaussianBGModel* [7]

ESTIMACIÓN DEL PRIMER PLANO

El primer plano de la escena es estimado al incorporar la información de bordes de la escena a la información del primer plano, obtenida por *FGDStatmodel*, como sigue:

- i. Se detectan los bordes de la imagen del fondo por media aumentada,  $F_i^{MA}(x,y)$  en la ecuación (2), y de la imagen del cuadro actual del video.
- ii. Se aplica una AND lógica entre el negativo de la imagen de bordes de  $F_i^{MA}(x,y)$  y la imagen de los bordes del cuadro actual del video.
- iii. Se aplica una OR lógica entre la imagen de bordes encontrada y el primer plano estimado por *FGDStatmodel*.

Dado que el modelo del fondo debe ser adaptativo para lidiar con los cambios de iluminación y posibles



Figura 3. (A) Primer cuadro del video de muestra, (B) cuadro del video sin vehículos, (C) fondo por media aumentada, (D) fondo por media, (E) fondo por *GaussianBGModel* y (F) fondo por *FGDStatmodel* [8] y [9].

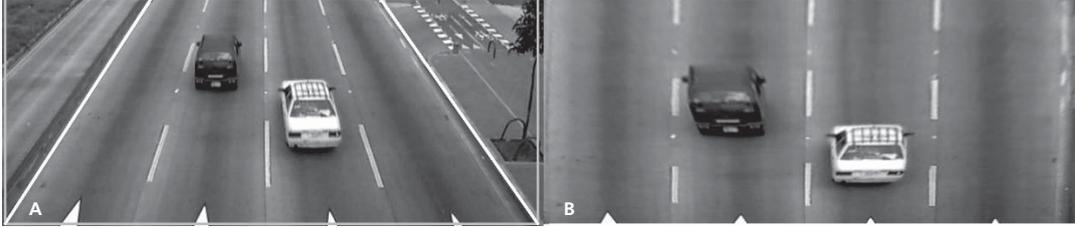


Figura 4. (A) Imagen que requiere corrección de perspectiva, (B) corrección de la perspectiva de la imagen A.

errores en modelos calculados previamente, los modelos FGDStatmodel Modificado y Media Aumentada son actualizados de forma periódica.

#### DETECCIÓN Y SEGUIMIENTO DE VEHÍCULOS

La información del primer plano es transformada en función de la perspectiva de la escena, con el fin minimizar la variación del tamaño de los objetos dentro de la región de interés, como se ilustra en la Figura 4. Adicionalmente, esta normalización permite que la información del número de carriles de la vía sea utilizada para determinar el ancho en pixeles de cada carril y el área, en pixeles, mínima y máxima de un vehículo.

#### ALGORITMO DE DETECCIÓN

Los conjuntos de pixeles conectados, del primer plano transformado, son denominados objetos de interés y para cada uno de ellos se obtiene su contorno y se le asocia un mínimo rectángulo (MR), alineado con los ejes que lo contiene. Los objetos de interés con área inferior a un umbral específico dependiente de la resolución del video y el número de carriles de la vía son eliminados.

Posteriormente, con el fin de unir o separar objetos que puedan estar mal segmentados debido a errores en la estimación del primer plano se hacen dos operaciones. Primero, los MR cercanos se evalúan y los que cumplen la condición que se explica a continuación son combinados en un nuevo MR que los contiene.

Sea  $r_i$  un MR con centro  $(x_i, y_i)$ , alto  $h_i$  y ancho  $w_i$ , entonces dos MR,  $r_i$  y  $r_j$ , son combinados en un nuevo MR si se sobreponen en más de un 80%, tal como se muestra en la ecuación (6).

$$|x_i - x_j| \leq 0.8 \left( \frac{h_i + h_j}{2} \right) \quad y \quad |y_i - y_j| \leq 0.8 \left( \frac{w_i + w_j}{2} \right) \quad (6)$$

Segundo, los MR que presentan un ancho superior al 130% del carril son divididos. Ambos umbrales se determinaron a partir de la observación sobre los videos del conjunto de entrenamiento.

#### EXTRACCIÓN DE CARACTERÍSTICAS

Para identificar los vehículos de un cuadro al otro se utilizan como características los momentos asociados al conjunto de pixeles que define cada MR. Sea  $O$  el objeto de interés del primer plano asociado a uno de los MR detectados, el  $p, q$ -ésimo momento de área de  $O$  se define en la ecuación (7), donde  $I(x, y)$  es la imagen binaria del primer plano. El centro de masa de  $O$  es la coordenada  $(x_c, y_c)$ , definida en la ecuación (8).

$$m_{p,q} = \sum_{(x,y) \in O} x^p y^q I(x,y) \quad (7)$$

$$x_c = \frac{m_{1,0}}{m_{0,0}}, \quad y_c = \frac{m_{0,1}}{m_{0,0}} \quad (8)$$

La dispersión de  $O$  con respecto a  $(x_c, y_c)$  se define en función de su segundo momento central, como se muestra en la ecuación (9); aquí  $\sigma_x$  y  $\sigma_y$  son las dispersiones horizontal y vertical.

$$\sigma_x = 4 \sqrt{\frac{m_{2,0}}{m_{0,0}} - (x_c)^2}, \quad \sigma_y = 4 \sqrt{\frac{m_{0,2}}{m_{0,0}} - (y_c)^2} \quad (9)$$

Cuadro a cuadro, cada  $O$  asociado a un MR es actualizado, es decir, se actualiza el centro de masa en  $(x_c, y_c)$ , su ancho  $w = \sigma_x$  y su alto  $b = \sigma_y$ .

ALGORITMO DE SEGUIMIENTO

Los MR detectados en cuadros consecutivos se relacionan para evitar contar múltiples veces un mismo vehículo. Sea  $N_n$  el número de MR detectados en el cuadro  $n$ , para cada MR, denotado como  $r_i$ , se define una estructura  $E_i[n]$ . Esta estructura contiene: el centro de masa, el tamaño, la velocidad instantánea y una bandera que permite saber si el MR permanece en escena.

Sean  $\{E_j[n-1]\}$  y  $\{E_k[n]\}$  los conjuntos de estructuras definidas para los MR detectados en los cuadros  $n-1$  y  $n$  del video. Sea  $D(j, k)$  la matriz de  $N_{n-1} \times N_n$ , que contiene en la posición  $(j, k)$  la distancia euclidiana entre el centro de masa almacenado en  $E_j[n-1]$  y el almacenado  $E_k[n]$ . Las  $N_{n-1}$  estructuras  $\{E_j[n-1]\}$  son relacionadas con las  $N_n$  estructuras  $\{E_k[n]\}$ , como sigue:

- i. Se encuentra  $(j', k') = \arg \{ \min [D(j, k)] \}$ .
- ii. Si  $D(j', k')$  es inferior al tamaño almacenado en  $E_{k'}[n]$  (desplazamiento máximo de un MR) se continua al paso *iii*, de lo contrario se prosigue al paso *iv*.
- iii. Se asocian las estructuras  $E_{j'}[n-1]$  y  $E_{k'}[n]$ : la velocidad en  $E_{k'}[n]$  es asignada como  $D(j', k')$ , y se activan las banderas de seguimiento de  $E_{j'}[n-1]$  y  $E_{k'}[n]$ .
- iv. Los valores de la fila  $j'$  y la columna  $k'$  son asignados como  $\max [D(j, k)]$ .

El procedimiento  $\min \{N_{n-1}, N_n\}$  se repite veces. Las estructuras de  $\{E_j[n-1]\}$  cuyas banderas no son activadas se borran (vehículos que salen de la RDI), mien-

tras que las de las estructuras de  $\{E_k[n]\}$  no activadas se toman como nuevos objetos de interés (vehículos que entran a la RDI).

RESULTADOS

Se tomaron videos en diferentes ubicaciones para entrenar y evaluar el algoritmo, aprovechando edificios contiguos a las vías y puentes peatonales. Así se obtuvieron muestras con distintos tipos de tráfico vehicular (autopistas y vías semaforizadas de alta y baja velocidad). En la Figura 5 se ilustran algunas de las perspectivas utilizadas para la evaluación del algoritmo.

Para evaluar su efectividad se hizo un conteo manual (cada video lo revisaron al menos dos personas para evitar errores) y luego se comparó con los conteos realizados por el algoritmo. Esta evaluación se hizo con 15 videos, cada uno de 3 a 10 minutos, seleccionados en forma aleatoria de entre 30 videos del conjunto de prueba (diferente al conjunto de entrenamiento). El número de vehículos contado manualmente en los 15 videos fue 2106, mientras que el algoritmo registró 2103, para una efectividad de 99,85%.

Cuando se lograba una ubicación ideal de la cámara, como la ilustrada en la Figura 5(D), el conteo realizado por el algoritmo fue perfecto, 100% de efectividad comparado con el conteo manual. En casos donde la imagen capturada tenía problemas de perspectiva y traslape, debido al ángulo de captura de la cámara, como se observa en la Figura 5 (A), el algoritmo dio un error en el conteo del 3,5% comparado con el conteo manual.



Figura 5. Ejemplos de perspectivas usadas.

## CONCLUSIONES

El algoritmo de estimación del primer plano presentado funcionó mejor que los algoritmos propuestos en [7], [8] y [9] (midiendo la cantidad de vehículos detectados); además, utiliza menos recursos computacionales. Estas dos características son fundamentales si se quiere automatizar la adquisición de variables de tráfico, pues la primera garantiza la exactitud del conteo y la segunda, además de permitir la viabilidad computacional de la solución propuesta, reduce los costos de la implementación real.

El desarrollo del conteo automático presentado combina dos estrategias de estimación del primer plano para realizar un proceso robusto de detección y seguimiento de los vehículos presentes en la vía; adicionalmente, es capaz de funcionar bajo diferentes perspectivas y volúmenes de tráfico. El algoritmo puede lograr una efectividad de conteo mayor al 99% en tiempo real, siempre que se garantice una ubicación de la cámara que minimice el traslape entre vehículos.

Con la información entregada por el algoritmo es posible determinar otras variables de interés, tales como la ocupación por carril o el tipo de servicio: particular o público. Así mismo, el algoritmo desarrollado permite clasificar los vehículos observados por tamaño, discriminando tres grupos: motos, automóviles y vehículos grandes. El algoritmo también permite obtener imágenes instantáneas de cada vehículo con las cuales se está construyendo una base de datos para entrenar un clasificador de vehículos.

Por último, el algoritmo desarrollado es capaz de detectar peatones, razón por la cual actualmente nos encontramos realizando pruebas para tratar de estimar flujos de peatones; otra variable importante en el tráfico urbano y que resulta más difícil de estimar que el tráfico vehicular.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] **K.S. Jie and M. Liu.**  
“Computer Vision Based Real-time Information Acquisition for Transport Traffic”. *IEEE International Conference on Information Acquisition*, 2005.
- [2] **J. M. Pardillo y V. Sánchez.**  
*Apuntes de Ingeniería de Tráfico*. E.T.S. Ingenieros de Caminos, Canales y Puertos, Universidad Politécnica de Madrid, Madrid, 2003.
- [3] **J. Soh, B. Chun and M. Wang.**  
“Analysis of Road Image Sequences for Vehicle Counting”. *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 1, 1995, pp. 679-683.
- [4] **S. H. Rhee, S. Han, P. K. Kim, M. B. Ahmad and J. A. Park.**  
“Vehicle Tracking Using Image Processing Techniques”. *Rough Sets and Current Trends in Computing*, Vol. 3066, 2004, Springer, Lecture Notes in Computer Science, pp. 671-678.
- [5] **E. Bas, M. Tekalp and F. S. Salman.**  
“Automatic Vehicle Counting from Video for Traffic Flow Analysis”. *IEEE Intelligent Vehicles Symposium*, 2007, pp. 392-397.
- [6] **S. Gupte, O. Masoud, R.F.K. Martin and N.P. Papanikolopoulos.**  
“Detection and Classification of Vehicles”. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, No. 1, March 2002, pp. 37-47.
- [7] **P. KaewTraKulPong and R. Bowden.**  
“An Improved Adaptive Background Mixture Model for Real time Tracking with Shadow Detection”. *Proceedings of 2nd European Workshop on Advanced Video Based Surveillance*, Vol. 5308, 2001.

[8] **L. Li, W. Huang, I. Gu and Q. Tian.**

“Foreground Object Detection from Videos Containing Complex Background”. *Proceedings of the eleventh ACM International Conference on Multimedia*, 2003, pp. 2–10.

[9] **P. L. Rosin**

“Thresholding for Change Detection”. *Proceedings of the Sixth International Conference on Computer Vision*, 1998.

[10] **C.C.C. Pang, W.W.L. Lam and N.H.C. Yung.**

“A Method for Vehicle Count in the Presence of Multiple-Vehicle Occlusions in Traffic Images”. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 8, No. 3, 2007, pp. 441-459.

[11] **G. Bradsky and A. Kaehler.**

*Learning OpenCV*, O’Reilly Media, Inc., 1st edition, 2008.