

SISTEMA BASADO EN PROCESAMIENTO DE IMAGENES PARA LA DETECCION DE INCIDENTES DE TRAFICO

Takeshi Asahi, Aldo Cipriano

Pontificia Universidad Católica de Chile, Casilla 306, Santiago 22, Santiago, Chile

Fax: 56-2-5522563; e-mail: aciprian@ing.puc.cl.

RESUMEN

En los últimos años, se ha visto disminuida la seguridad en las vías de circulación debido al aumento de la densidad del tráfico. Para enfrentar este problema se ha requerido de la automatización de funciones como el control de tráfico y monitoreo. El presente trabajo comienza con una revisión de diferentes sistemas que realizan detección automática de incidentes de tráfico. El objetivo es seleccionar un algoritmo que utilice procesamiento de imágenes, para luego medir su eficiencia. El algoritmo fue implementado en un sistema que funciona en ambiente Windows, donde se capturan imágenes de una cámara de video. Las imágenes de la vía son analizadas en forma automática de manera de generar mensajes de alarma en caso de detectar incidentes. El sistema fue evaluado con grabaciones adquiridas desde la Unidad Operativa de Control de Tráfico (UOCT) de Santiago, obteniendo resultados preliminares bastante satisfactorios.

1. INTRODUCCION

El problema del aumento de la congestión vehicular que experimentan las grandes ciudades ha sido atacado desde diferentes puntos de vista, por ejemplo, con una adecuada sincronización de semáforos e instalando en las vías letreros de mensajes variables que aconsejen a los conductores para que eviten las rutas de mayor congestión. Sin embargo, esto no es suficiente, requiriéndose de un mejor control de tráfico y monitoreo de las principales vías de circulación. Para ello es útil contar con mediciones precisas de variables como el flujo vehicular, y en general, la ocupación de las pistas, lo que hace necesario mejorar los instrumentos de medición existentes, como las espiras magnéticas, y crear nuevos sensores. Para esto último se presta la utilización de técnicas de procesamiento de imágenes, que explotan la naturaleza bidimensional de la dinámica del flujo vehicular (Iñigo, 1985).

Por otra parte, investigaciones realizadas durante estos últimos años han comprobado que un significativo número de accidentes ocurren en las vías de alta velocidad, y que son causados por vehículos detenidos o que se desplazan a velocidades muy bajas y que obstruyen las vías (Roberts, 1994). Para evitarlos sería útil suministrar a los conductores información más precisa acerca del estado de las vías, de modo que éstos pudieran tomar las precauciones necesarias. Aún

si se dispone de cámaras de video que capturen imágenes de las vías, para un supervisor humano sería imposible analizar tal cantidad de imágenes y avisar a tiempo a los conductores, por lo cual es necesario automatizar el proceso de detección de incidentes en las diferentes vías de circulación.

El objetivo de este trabajo es justamente describir un sistema que detecta en forma automática incidentes en calles o carreteras, utilizando procesamiento de imágenes, entendiéndose por incidente cualquier evento que impide una circulación expedita. El sistema desarrollado despliega en tiempo real imágenes dinámicas de la vía, captadas por una cámara, analiza estas imágenes y genera mensajes de alarma al detectarse incidentes.

El trabajo comienza con una comparación de diferentes algoritmos conocidos de detección de incidentes, para luego seleccionar el más apropiado considerando principalmente su flexibilidad de aplicación para diferentes casos. A continuación se presenta la programación del algoritmo escogido, en lenguaje C++, bajo ambiente Windows, y se presentan los resultados de su evaluación utilizando imágenes adquiridas desde la Unidad Operativa de Control de Tráfico (UOCT) de Santiago.

2. DETECCION DE INCIDENTES

2.1. Procesamiento de imágenes aplicado a tráfico

El procesamiento de imágenes permite extraer la información sobre las condiciones del tráfico, analizando imágenes de video filmadas en las vías, con lo cual se pueden lograr diversos objetivos (Iñigo, 1985):

- Obtener parámetros numéricos tales como el número de vehículos, la velocidad y el largo de éstos, así como la ocupación de la pista, a medida que el tráfico cruza una determinada línea en la respectiva vía. Este método tiene su análogo convencional que es el de utilizar como detectores espiras magnéticas bajo el pavimento.
- Vigilar automáticamente una porción de hasta 1 [Km] de carretera, rastreando vehículos en forma individual o detectando maniobras de cambio de pista.
- Detectar automáticamente la presencia de incidentes en calles o carreteras, determinando si existen atochamientos o accidentes.
- Rastrear vehículos, monitoreando su movimiento en una intersección, típicamente en una área cuadrada de 30 [m] de ancho.
- Clasificar vehículos, registrando sus formas y haciéndolas calzar con patrones para obtener un cierto código de clasificación.
- Identificar las matrículas de los vehículos.

Como se puede apreciar, la flexibilidad en el uso del procesamiento de imágenes es mucho mayor que la ofrecida por los métodos convencionales, como las espiras magnéticas. Estas, si bien son ampliamente utilizadas hoy en día por su bajo costo, se instalan en forma fija a unos 10 [cm] bajo el pavimento, y por ello no son posibles de desplazar a otras posiciones.

Otra ventaja ofrecida por el procesamiento de imágenes es que en muchas ciudades ya existen cámaras instaladas en las principales vías y arterias, las que se utilizan para una inspección visual desde una sala de control centralizada.

2.2. Algoritmos para la detección de incidentes

Como ya se mencionó con anterioridad, los avances en la tecnología digital están permitiendo investigar seriamente en métodos para detección de incidentes. Los primeros desarrollos en detección de incidentes a través de imágenes de las vías no introdujeron mayores innovaciones en cuanto a los métodos tradicionales de detección, ya que el procesamiento de imagen se realizaba emulando una espira magnética, creando líneas virtuales de detección. La detección de incidentes se hacia analizando los cambios estadísticos en las tendencias de las variables.

Nuevos desarrollos han explotado la idea de que a través de la imagen se puede extraer datos de tráfico no solamente de un punto específico (como lo hacen las espiras magnéticas), sino que también de un área en el espacio. A continuación se describen algunos métodos y sistemas que realizan la detección de incidentes en base a procesamiento de imágenes.

CCATS. (Camera & Computer Aided Traffic Sensor). Este sistema ha sido desarrollado y comercializado por la compañía belga Devlonics Control NV (Cypers, 1990), durante los últimos 12 años. El sistema opera en forma distribuida, procesa localmente el área de interés dentro de la escena para obtener una imagen binaria de los vehículos presentes, y determina dirección, velocidad y clasificación de cada vehículo. Además, el sistema provee la información necesaria para detectar incidentes en forma automática. Ya existe una aplicación que utiliza CCATS para detección de atochamientos, para lo cual se comparan parámetros de tráfico, calculados en tiempo real. Se han iniciado estudios destinados a emplear variantes de CCATS para la detección de otros incidentes en las vías de circulación.

AUTOSCOPE. Los métodos de detección se basan en los algoritmos de McMaster (Panos, 1993), los cuales emplean datos de volumen, ocupación y velocidad promediados durante 30 segundos. El primer método algorítmico desarrollado fue SPIES (Speed Profile Incident Evaluation System), que hace uso de dos medidores de velocidad situados en cada pista, a la mayor separación posible (60-90 [m]). Los incidentes se detectan comparando velocidades y volúmenes de tráfico aguas arriba y aguas abajo de los detectores. La alarma se genera si esta diferencia es mayor que un umbral dado. Un segundo algoritmo, denominado AUTOSCOPE Incident Detection Algorithm (AIDA), combina las fortalezas de los algoritmos de McMaster y de SPIES, pero también utiliza la variación temporal de los parámetros de tráfico. Este algoritmo utiliza la información acerca de los cambios repentinos en el flujo del tráfico tales como reducciones abruptas de velocidad o incrementos de ocupación para determinar el nivel de congestión.

Algoritmo FAST-Q. El algoritmo FAST_Q (Fourier Analysis for Slow Traffic Queues) (Rourke, 1991), emplea un método basado en una descripción macroscópica cuasi cualitativa de la escena de tráfico, que utiliza la reducción tanto espacial como temporal de los datos. La reducción espacial se realiza al determinar una ventana, consistente en una sola línea de pixeles, paralela a

la dirección del movimiento vehicular, a lo largo de una pista de tráfico que está siendo monitoreada. La reducción en el dominio del tiempo se realiza seleccionando la ventana a procesar por algunos segundos.

IMPACTS. El algoritmo utilizado por IMPACTS (Image Processing for Automatic Computer Traffic Surveillance) realiza las siguientes funciones para la detección de incidentes: determinación de las celdas, cálculo de la referencia, determinación de la referencia inicial, extracción de las imágenes, determinación de los parámetros de cada celda, determinación de los objetos, y determinación de parámetros adicionales. Parte del método es descrito en detalle en el capítulo siguiente.

Al comparar los diferentes algoritmos para detectar incidentes, se aprecia que FAST_Q e IMPACTS son más fáciles de implementar, debido a que los otros métodos requieren de información histórica para ajustar sus parámetros. Los algoritmos basados en estadísticas requieren muchas veces modelos de comportamiento del flujo vehicular, los cuales, por lo general, no se ajustan bien para todas las condiciones de tráfico existentes. Las soluciones del tipo mixto, que extraen parámetros de tráfico usuales a través de imágenes y detectan incidentes estadísticamente, constituyen un paso intermedio en el desarrollo de estos sistemas. El presente trabajo se concentrará en el sistema IMPACTS, debido a que es un algoritmo que realiza el análisis de la imagen en forma similar a como lo haría un supervisor experto. Se caracteriza por utilizar mejor el conocimiento espacial (2-dimensiones) que el algoritmo FAST_Q (1-dimensión).

3. SISTEMA DESARROLLADO

3.1. Algoritmo implementado

El algoritmo implementado se basa en el utilizado por IMPACTS (Hoose, 1992), y consiste en los pasos descritos a continuación:

Determinación del tamaño y posición de las celdas. Primeramente se definen celdas rectangulares de una altura tal que cubren el ancho de una pista, y el largo promedio de un vehículo particular, como se puede apreciar en la Figura 1.

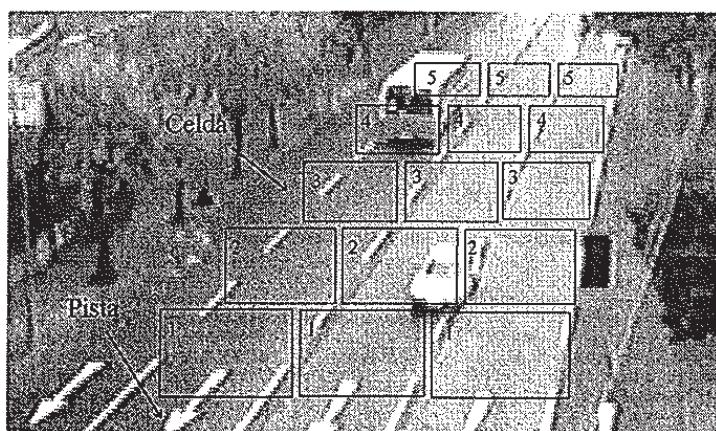


Figura 1 : Ejemplo de celdas rectangulares

Determinación de la referencia inicial. Para determinar la referencia en cada celda sin la presencia de vehículos, se utiliza el método de WADS (Hilbert, 1978), mediante el cual se obtienen 10 muestras de la celda con una separación temporal de 0,4 segundos. Las casillas que se comparan son las que se encuentran a 2 segundos entre sí, esto es las celdas 1 y 6, 2 y 7, ... 5 y 10. A continuación se calculan desviaciones para cada par de celdas, de acuerdo a:

$$S_k = \sum_{i=1, j=1}^{n_x, n_y} |P_{k+5}(i, j) - P_k(i, j)| \quad (1)$$

En esta expresión S_k es la desviación total entre el par de celdas k y $k+5$, n_x y n_y la dimensión de las celdas en pixeles y $P_k(i, j)$ es el nivel de gris del pixel (i, j) de la muestra k .

La imagen de referencia de cada celda se escoge como el promedio entre las imágenes k y $k+5$ cuya desviación sea mínima.

Cálculo de la referencia. Para actualizar en forma automática y permanente la imagen de referencia de cada celda, se utiliza una atenuación exponencial para los niveles de gris de cada pixel de la celda, según la fórmula indicada a continuación:

$$R_k(i, j) = w * P_k(i, j) + (1-w) * R_{k-1}(i, j) \quad (2)$$

En la ecuación anterior $R_k(i, j)$ es el nivel de gris de la referencia en el tiempo k , del pixel de la celda con coordenadas (i, j) , $P_k(i, j)$ el nivel de gris del pixel (i, j) obtenido en el tiempo k , R_{k-1} el nivel de gris de la referencia en el tiempo $k-1$, y w es un factor de ponderación. De esta forma los grises de la referencia seguirán la tendencia a oscurecer o a aclararse según la hora del día. En este caso tanto P como R son niveles de grises entre 0 y 255, y w varía entre 0 y 1, aunque generalmente es bajo (cercano a 0,2) para no incorporar distorsión por el ruido de la nueva imagen.

La referencia se actualiza si no hay vehículos presentes en la celda, pero en caso que esto no ocurra durante las últimas cinco veces, ésta se modifica mediante la utilización de un parámetro Bias. Este Bias se determina como el cambio del nivel de gris promedio de un sector de la pantalla donde no circulan vehículos, al restar el promedio de gris de la muestra actual con el de cinco muestras atrás. El Bias se suma al nivel de gris de cada pixel de la celda que se está actualizando.

Obtención de las imágenes y determinación de los parámetros de cada celda. Primeramente se obtienen dos imágenes separadas entre sí por 0,5 segundos. Para cada celda se calculan los parámetros r_k , que permite comparar la celda con la referencia, y d_k , que representa la diferencia entre dos mediciones consecutivas:

$$r_k = \sum_{i=1, j=1}^{n_x, n_y} |P_k(i, j) - R_k(i, j)| \quad (3)$$

$$d_k = \sum_{i=1, j=1}^{n_x, n_y} |P'_k(i, j) - P_k(i, j)| \quad (4)$$

con R_k la referencia en el tiempo k , P_k el nivel de gris de la primera muestra correspondiente al tiempo k , y P'_k el nivel de gris de la segunda muestra efectuada 0,5 [seg] después.

Determinación del estado de cada celda. Los parámetros r_k y d_k se comparan con umbrales r_θ y d_θ , obtenidos como:

$$r_\theta = n_x * n_y * I_{r\theta} \quad (5)$$

$$d_\theta = n_x * n_y * I_{d\theta} \quad (6)$$

donde $I_{r\theta}$ e $I_{d\theta}$ son los niveles de gris promedio que se utilizan como umbrales de la comparación con la referencia y diferencia respectivamente. La Tabla 1 muestra como se define el estado de cada celda utilizando estos umbrales.

Tabla 1
Estados de una celda

	$r_k < r_\theta$	$r_k \geq r_\theta$
$d_k < d_\theta$	N	D
$d_k \geq d_\theta$	A	M

En la Tabla 1 "N" significa que no existe vehículo en la casilla, "D" indica que existe un vehículo en la celda y está detenido, "M" si los vehículos en la celda se están desplazando, y "A" si es necesario analizar con más detalle pues no es posible determinar en forma adecuada la situación.

Estado de cada pista. Para caracterizar el estado de cada pista, se determinan dos parámetros que son la Densidad Movimiento (D_M) y la Densidad de Fila (D_S), definidos según las siguientes relaciones:

$$D_M = \frac{N_M}{N_C} \quad (7)$$

$$D_S = \frac{N_D}{N_C} \quad (8)$$

En estas expresiones N_C es el número total de celdas en una pista, N_M el número de celdas de tipo M en una pista, y N_D el número de celdas de tipo D en una pista.

A partir de estos valores y utilizando la Tabla 2, se determina el estado de cada pista en forma individual.

Mensajes desplegados. A continuación se describen los mensajes desplegados, que dependen de la alarma generada luego del análisis de los datos históricos de las seis últimas mediciones.

- **Taco en la vía:** se genera una alarma si todas las pistas han presentado estado de DETENCION durante las cuatro últimas mediciones.
- **Taco en la pista:** se genera señal de aviso si una pista determinada ha tenido el estado de DETENCION durante las cuatro últimas mediciones.
- **Possible taco en la pista:** se despliega un aviso si durante las cuatro últimas mediciones ha ocurrido DETENCION o TRAFICO DENSO, pero en al menos dos de ellas, ha habido una DETENCION. El último estado de la pista tiene que ser DETENCION.
- **Vehículo detenido:** se revisan las celdas que han estado en DETENCION durante las seis últimas mediciones. Si ocurriera que existe más de una celda en la misma pista con estas características, se privilegia aquella celda que se encuentra aguas abajo.

Tabla 2
Estados de la pista

Condiciones	Estado
$D_S \geq 0,8$	DETENCION
$D_S < 0,8$ y $D_S + D_M \geq 0,8$	TRAFICO DENSO
$D_S + D_M < 0,8$	NORMAL

Para evitar que el mensaje se genere en cada medición, sólo se despliega el mensaje si en la medición anterior no hubo mensaje o bien si el mensaje fue diferente.

Modificación del algoritmo. Al determinar el estado de cada celda en días con sol, se generaron errores causados por la sombra de los vehículos y árboles cercanos en el pavimento. Por esta razón el algoritmo fue modificado considerando el hecho que la imagen original es en colores y utilizando la representación YIC, donde Y es la *luminancia* (la intensidad del gris utilizada inicialmente) y los parámetros I y C la *crominancia*.

La transformación RGB (Red, Green, Blue) a YIC es la siguiente (Russ, 1995):

$$\begin{bmatrix} Y \\ I \\ C \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,275 & -0,321 \\ 0,212 & -0,523 & 0,311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (9)$$

Cabe destacar que al utilizar solamente la información de I y C, se disminuye el problema de las sombras sobre el pavimento, pero el algoritmo no puede discriminar bien entre el pavimento y los vehículos blancos, grises y negros. Por ello, para calcular la distancia entre dos pixeles se utiliza también el parámetro de la luminancia, pero en un factor reducido, definiéndose la diferencia entre dos pixeles correspondientes como:

$$d = \sqrt{(I_1 - I_2)^2 + (C_1 - C_2)^2 + \alpha * (Y_1 - Y_2)^2} \quad (10)$$

donde I_i indica a la componente I , C_i a la componente C e Y_i a la luminancia, $i=1$ si los parámetros del pixel pertenecen a la imagen 1 e $i=2$ si pertenecen a la imagen 2. α es un ponderador de la luminosidad, que en este caso corresponde a un valor de 0,06, obtenido por prueba y error.

Con la ecuación (10) se obtiene la diferencia entre dos píxeles para determinar los parámetros d_k y $d_{k'}$ correspondientes a cada medición.

3.2. Hardware y Software

Para implementar el algoritmo, se utilizó un computador PC compatible con procesador PENTIUM de 100 MHz y una tarjeta de video *Comet* de Matrox Electronic Systems Ltd. La programación se realizó en lenguaje C++, utilizando el compilador Borland C++ versión 4.02, que corre bajo sistema operativo Windows 3.11. Este computador requiere de 20 segundos para ejecutar el algoritmo descrito anteriormente.

4. ANALISIS DE RESULTADOS

Las pruebas se realizaron utilizando cintas de video grabadas en diferentes lugares de Santiago por la Unidad Operativa de Control de Tráfico (UOCT) de Santiago, una de cuyas funciones es la supervisión de intersecciones importantes, como la de Avenida Costanera con el puente de Avenida el Cerro, analizando imágenes de diferentes cámaras situadas en esos lugares. La grabación fue realizada entre las 16:30 y las 18:20 de un día de semana y con sol. Otra grabación utilizada, corresponde a la intersección de Avenida Bellavista con Purísima entre las 7:15 y las 9:15 A.M. Esta grabación presenta muchas dificultades para el algoritmo de detección debido a que la cámara filma en dirección hacia el oriente, por lo cual recibe los reflejos de la luz del sol.

Para evaluar el desempeño del algoritmo se utilizaron dos indicadores:

La Razón de Detección (DR). Esta razón indica el porcentaje de veces que los mensajes de alarma se despliegan debidamente y es calculada como:

$$DR = \frac{\text{Ciclos con eventos significativos correctamente identificados}}{\text{Número total de eventos significativos}} * 100 \quad (11)$$

La Razón de Falsa Alarma (FAR), definida como:

$$FAR = \frac{\text{Ciclos en que el aviso no esta debidamente establecido}}{\text{Ciclos en que el aviso esta debidamente establecido}} * 100 \quad (12)$$

Se entiende por aviso incorrecto a un aviso falso (incidente que nunca existió), o a un evento importante que debió ser informado y no se detectó.

Tiempo medio de detección (MTTD). Corresponde al tiempo promedio que el algoritmo demora en detectar los diversos incidentes. El algoritmo empleado requiere cuatro ciclos para discriminar entre un taco o solamente una detención momentánea. Como los ciclos son cada 20 segundos, el tiempo medio de detección corresponde en este caso a 80 segundos. El detectar un *possible taco* también demora 80 segundos (cuatro ciclos). El tiempo medio para detectar un vehículo detenido es de 120 segundos, pues el algoritmo analiza los seis últimos ciclos antes de enviar el aviso de detención.

Con las imágenes de la Avenida Costanera, se obtuvieron 327 muestras cada 20 segundos, con las cuales se generó la Tabla 3, en la que se indican los incidentes detectados.

Tabla 3
Resultados de Avenida Costanera

Muestra	Hora	Tipo de incidente	Comentario
93	17:02:13	Detención en pista 3, celda 1	No existe
94	17:02:33	Possible taco en pista 1	No detectado
94	17:02:33	Possible taco en pistas 2 y 3	Detectado
176	17:29:53	Detención en pista 1, celda 3	No detectado
177 → 228	17:30:13 17:47:13	- Detención en pista 1, celda 3	Detectado

El incidente detectado en la muestra 93 corresponde a una gran fila de vehículos que avanzan en forma muy lenta, por lo cual no constituye un incidente real.

En la muestra 94 no fue detectado un *possible taco* en la pista 1 aunque si fue detectado para el resto de las pistas. Este *possible taco* es despejado rápidamente.

La detención en la pista 1, celda 3, fue detectada con una muestra de retraso.

A partir de la tabla anterior se obtuvo una Razón de Detección (DR) de 94.55% y una Razón de Falsa Alarma (FAR) de 0.92%. Estos indicadores pueden ser comparados con los obtenidos al evaluar IMPACTS, (Hoose, 1992): DR = 96,7% y FAR = 0,032%. Cabe destacar que IMPACTS presenta una confiabilidad muy superior, pero por otra parte ciertos parámetros del algoritmo desarrollado en este trabajo no han sido ajustados con la suficiente precisión.

Pruebas realizadas con otras grabaciones, no permitieron obtener resultados satisfactorios. Este es el caso de la grabación en Avenida Bellavista, en la cual el sistema opera en forma adecuada hasta que sale el sol, aproximadamente a las 7:25. En ese momento el pavimento comienza a recibir luz directa del sol, la cual hace cambiar rápidamente su luminosidad, haciendo que el sistema detecte la presencia de un vehículo falso. Esto lleva a la no actualización de las celdas y,

finalmente, al desajuste del sistema. Se puede mencionar que en los últimos 40 minutos, cuando el sol deja de estar frente a la cámara, el algoritmo no se desajusta.

Una posible solución al problema anterior consiste en reinicializar las referencias, procedimiento en cierto modo similar al de la determinación de las referencias iniciales a través de la utilización de diez muestras seguidas.

5. CONCLUSIONES

Mediante el uso de procesamiento de imágenes es posible detectar diversos incidentes que ocurren en una vía, como por ejemplo un vehículo detenido o congestión. Pruebas realizadas para medir su desempeño, indican una Razón de Detección (DR) de 94,5% y una Razón de Falsa Alarma (FAR) de 0,9%, índices que son razonables pero en todo caso inferiores a los que se obtienen con el sistema IMPACTS. La gran debilidad del algoritmo está en la actualización de las referencias, pues ante cambios ambientales rápidos el programa genera detecciones erróneas ya que confunde el cambio ambiental con un vehículo detenido. Por consiguiente, se requiere robustecer esta parte del algoritmo.

Existen otras mediciones que se podrían incorporar al sistema, entre las que se puede mencionar la determinación del largo de las colas y la identificación de bloques y oleadas de vehículos. Para ello se hace uso de la información bidimensional de la imagen, la cual no puede ser obtenida en forma directa por técnicas convencionales como las espiras magnéticas.

REFERENCIAS

- Iñigo, Rafael M. (1985). Traffic monitoring and control using machine vision: a survey. *IEEE Transactions on Industrial Electronics*, IE-32, N° 3, 177-185.
- Nicholas P. Roberts, Simon A. Webb y Graham Coe. (1994). Incidents on motorways. *Traffic Engineering + Control*, 550-554.
- L. Cypers, G. Kolacny, J. M. Poncelet, P. Vervenne, F. Lemaire y F. De Jaegere. (1990). CCATS: The image processing-based traffic sensor. *Traffic Engineering + Control*, 371-375.
- Panos G. Michalopoulos, Richard D. Jacobson, Graig A. Anderson y Thomas B. Debruycker. (1993). Automatic incident detection through video image processing. *Traffic Engineering + Control*, 66-75.
- A. Rourke y M. G. Bell. (1991). Queue detection and congestion monitoring using image processing. *Traffic Engineering + Control*, 412-421.