

Implementación de un Sistema para la Medición del Flujo de Tránsito y Detección de Embotellamientos en Autopistas.

Damián Oliva, Agustín Yabo, Lilián García, Sebastián I. Arroyo, and Félix Safar

Carrera de Ingeniería en Automatización y Control Industrial,
(IACI)Universidad Nacional de Quilmes,
doliva@unq.edu.ar
Roque Saenz Peña 352 (B1876BXD), Bernal, Buenos Aires

Resumen En este trabajo implementamos un sistema para la medición de propiedades del flujo de tránsito para la detección de embotellamientos en autopistas. El sistema está basado en las técnicas de ‘sustracción de fondo’ y de ‘seguimiento de características’. El sistema desarrollado permite: 1. cuantificar la fracción de llenado vehicular de distintas regiones de la autopista y 2. estimar la velocidad media del flujo de vehículos. Estudiamos la relación entre estas dos variables, y utilizando esta información proponemos un método para detectar automáticamente embotellamientos usando solo la fracción de llenado.

1. Introducción

Existen diversas tecnologías de sensores para la detección de vehículos en la red vial de caminos, con distintas capacidades en cuanto al conteo, estimación de velocidad, clasificación del tipo de vehículo, determinación de su posición relativa al camino, etc[1]. Algunos ejemplos de las tecnologías más difundidas a nivel global son: 1. el lazo inductivo: es la técnica más utilizada. El sensor está cercano al vehículo y tiene buena relación señal-ruido. Pero requiere realizar cambios físicos en la vía durante su instalación. 2. Radar: pueden medir zonas amplias de la vía y no requieren cambios físicos en la misma. Necesitan de un procesamiento más complejo de la información y no detectan vehículos lentos o detenidos. 3. Técnicas basadas en información GPS: una empresa informa sobre el estado del flujo vehicular recopilando información de teléfonos celulares. La información provista por la empresa es cualitativa y solo se tiene en zonas de con alta población de vehículos.

En este trabajo implementamos un sistema de visión artificial para la medición del flujo de tránsito. Algunas de las ventajas de esta modalidad de detección son: 1. no ser intrusiva, o sea, para instalación y mantenimiento no requiere intervenir y clausurar temporalmente la vía transitada; 2. existencia de gran cantidad de cámaras ya instaladas cuya información puede utilizarse para este fin; 3. información detallada de los tipos de vehículos que están transitando; 4. Medición de zonas amplias de la vía.

El algoritmo que proponemos estima la densidad de llenado y la velocidad media del flujo vehicular en regiones de interés de una autopista. En la sección 2 describimos las técnicas de video detección que utilizamos. En la sección 3 presentamos los resultados obtenidos en mediciones de una autopista con videos adquiridos vía web. Finalmente en la sección 4 discutimos los resultados obtenidos y las mejoras a desarrollar en el futuro.

2. Materiales y Métodos

Los videos analizados en este trabajo fueron medidos durante el mes de septiembre del año 2014 a una tasa de adquisición de 1fps y una resolución 450x800 píxeles. Fueron adquiridos automáticamente vía web [2] con un programa desarrollado a tal fin en Python-OpenCV. Los algoritmos de procesamiento que describimos a continuación, fueron implementados en Matlab (Computer Vision toolbox[11], algoritmos: SURF y FLANN) que es un software privativo y están en proceso de implementarse en Python-OpenCV que es un software libre.

En las Figuras 1A y 1B mostramos dos *frames* (cuadros o fotogramas) asociados a situaciones con la vía libre y con embotellamientos respectivamente. Para simplificar el análisis se definieron regiones de interés (ROIs) cuadrangulares definidas por un operador (Figura 1A).



Figura 1: a) Imagen de cámara PTZ en condiciones de baja densidad de llenado. b) Imagen de embotellamiento en una de las vías (alta densidad de llenado).

2.1. Método de Sustracción de Fondo

La sustracción de fondo es un método ampliamente utilizado para la detección de objetos en movimiento[3]. Este enfoque se basa en la diferencia entre el cuadro actual y una imagen de referencia (llamada ‘imagen de fondo’ o *background*) que debe actualizarse a las condiciones de luminosidad variables en el tiempo. En este trabajo, utilizamos un método heurístico desarrollado previamente[4], que detecta las coordenadas (x_i, y_i) de los píxeles asociados a los objetos en movimiento (que también denominaremos *foreground*). Para esto, realizamos un procesamiento en 3 operaciones o capas (cada ecuación corresponde a una capa de procesamiento):

$$\begin{aligned}\mu_{i,j}(t) &= \alpha \cdot I_{i,j}(t) + (1 - \alpha) \cdot \mu_{i,j}(t - 1), \\ h_{i,j}(t) &= \text{sgn}(|I_{i,j}(t) - \mu_{i,j}(t)| - U_1), \\ c_{i,j}(t) &= \text{sgn}(N_{i,j}(t)/N_{vec} - U_2).\end{aligned}$$

La primer capa, estima en forma (on-line) el valor medio $\mu_{i,j}(t)$ asociado a cada píxel (i, j) a través de un promedio móvil exponencial con constante de tiempo $1/\alpha$. La segunda operación del clasificador compara la medición actual $I_{i,j}(t)$ con $\mu_{i,j}(t)$. Si la diferencia es menor que el umbral U_1 , entonces $h_{i,j}(t) = -1$ (en caso contrario $h_{i,j}(t) = +1$). La tercer capa, calcula la cantidad de píxeles $N_{i,j}(t)$ vecinos al píxel (i, j) que tienen clase $h_{i,j}(t) = +1$. Definimos como vecinos a los píxeles dentro de una ventana cuadrada centrada en (i, j) de área N_{vec} . Si la proporción $N_{i,j}(t)/N_{vec} > U_2$, entonces $c_{i,j}(t) = +1$ y el píxel (i, j) es clasificado como foreground. En caso contrario se clasifica como background.

Es importante destacar que una ventaja de este clasificador es la velocidad de procesamiento y el bajo requisito de memoria para cada píxel.

Los parámetros $\{\alpha, U_1, U_2, N_{vec}\}$ fueron optimizados con el procedimiento descrito en [4]. Resumidamente, un supervisor indica la cantidad de vehículos que se encuentra en la región de interés en cada frame ' t ' $N_{exp}(t)$. Se usaron cuatro periodos de 2 minutos en distintas condiciones ambientales. Luego se calcula la cantidad de blobs detectados por el algoritmo $N_{pred}(t)$ y se calcula el error absoluto medio (EAM) de predicción para todos los frames. Se explora intensivamente en el espacio de los parámetros $\{\alpha, U_1, U_2, N_{vec}\}$ y se elige el conjunto de menor EAM. Los valores óptimos fueron $\{\alpha = 0,002, U_1 = 0,15, U_2 = 0,15\}$; $\{N_{vec} = 25\}$ para la mitad superior de la imagen y $\{N_{vec} = 100\}$ para la mitad inferior. Esto se hace para mitigar el efecto de disminución del tamaño de los vehículos en función de la profundidad por efectos de perspectiva.

$c_{i,j}(t)$ es una imagen binaria donde las regiones de $c = +1$ son *blobs* o 'burbujas'. En condiciones favorables (objetos suficientemente separados, sin oclusiones, luminosidad suave y homogénea) corresponden unívocamente a objetos en movimiento. En este caso, posiciones y velocidades de los blobs son las de los objetos. Como se mostrará más adelante, las condiciones no son siempre propicias y los blobs no sirven para identificar unívocamente a los vehículos.

2.2. Detección y correspondencia de características

En situaciones de embotellamiento resulta extremadamente complicado mantener la identidad de un vehículo durante varios frames debido a la fusión y separación de las burbujas foreground asociadas a cada vehículo (ver Figura 3D). Este problema nos llevó a aplicar una solución conceptualmente distinta para estimar la velocidad del flujo vehicular.[7].

Para estimar la velocidad del tráfico se usó el movimiento de 'características'[7]. Utilizamos la técnica SURF (Speed-Up Robust Features) que fue propuesta en [6] aprovechando librerías que ya incluyen su implementación. Para encontrar exitosamente ciertas características comunes entre dos frames sucesivos a pesar de desplazamiento en el espacio, rotación, cambio de escala o iluminación de las mismas de un frame a otro. El algoritmo utilizado consta de tres pasos:

1. Proceso de Detección: es el proceso para hallar un punto de interés específico de la imagen con alta repetibilidad, es decir que puede encontrarse aún cuando cambien las condiciones de visualización. SURF aproxima *kernels* Gaus-

sianosbidimensional de tamaños crecientes, para convolucionar con la imagen y generar una descripción de la misma a distintas escalas. En este trabajo usamos: 9×9 , 15×15 , 21×21 , 27×27 .

2. Proceso de Descripción: en este paso se genera un vector multidimensional que representa de forma robusta el área circundante a cada característica de interés hallada en el paso anterior. En este trabajo se utiliza un descriptor de 64 dimensiones.

3. Proceso de Correspondencia: Luego de encontrar los descriptores asociados a cada característica en la imagen $(t - 1)$ y (t) , el problema de correspondencia se reduce a aparear aquellos vectores 64D que tengan la menor distancia entre sí. Se utiliza como distancia la suma de diferencias cuadráticas y la asignación se hace típicamente mediante vecinos más cercanos, FLANN[9].

2.3. Ajuste de transformación de perspectiva y obtención de coordenadas cartesianas.

Las características detectadas tienen coordenadas (x, y) dentro de la imagen, sus velocidades no son representativas debido a efectos de perspectiva principalmente. Para poder medir la velocidad del tráfico debemos estimar la posición cartesiana de cada característica en el sistema de coordenadas ‘mundo’ (X, Y) (en metros) en función de la posición en la imagen (en píxeles). Para esto, ajustamos una transformación de perspectiva (ecuación 4) para los puntos de un cuadrilátero de calibración (ver Figura 2)

$$X = \frac{ax + by + c}{gx + hy + 1}, \quad Y = \frac{dx + ey + f}{gx + hy + 1}. \quad (1)$$

Donde $\lambda = (a, b, c, d, e, f, g)$ son los parámetros a ajustar por el método de la pseudo-inversa[12].

Es importante destacar que en este método suponemos que todos los puntos de interés se encuentran sobre el plano de la vía. Si bien esta aproximación no es cierta, despreciamos estos errores ya que la cámara se encuentra a una altura mayor (25 metros) que la máxima altura de los vehículos.

3. Resultados

3.1. Aplicación del Método de Sustracción de Fondo para Medir la Densidad de Llenado de la Vía.

Dividimos la vía analizada en 4 regiones tal que tuvieran el mismo tamaño en el marco de referencia ‘mundo’ (ver Figura 3A). Definimos la *fracción de llenado promedio* para la ROI n a tiempo t como $\rho(n, t) = A_{frg}(n, t)/A_{ROI}(n)$, donde A_{frg} y A_{ROI} corresponden al número de píxeles clasificados como foreground y área en píxeles de la ROI, respectivamente.

Aplicamos el algoritmo de sustracción de fondo en un total de 40 horas asociados a videos adquiridos durante 10 días. En la Figura 3B mostramos un ejemplo de medición para las cuatro ROIs en un periodo de 4 horas. Como se puede

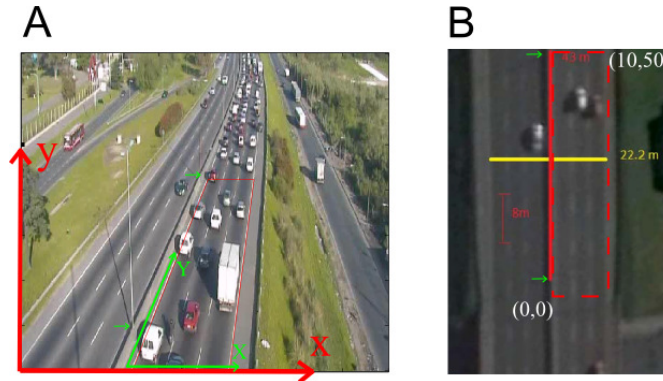


Figura2: Ajuste de transformación de perspectiva y obtención de coordenadas cartesianas sobre la autopista. A) Los ejes rojos (x, y) y verdes (X, Y) representan los sistemas de referencia ‘imagen’ y ‘mundo’ respectivamente. Los puntos de calibración utilizados para el ajuste de la transformación de perspectiva corresponden a los vértices de cuadrilátero rojo. B) Las coordenadas cartesianas de los puntos de calibración se midieron utilizando el servicio Google Maps. El rectángulo punteado representa el cuadrilátero de calibración en la trama ‘mundo’.

apreciar la fracción se mantiene baja ($\rho(n, t) < 0,2$) la mayor parte del tiempo pero aumenta sustancialmente durante un período de 30 minutos.

Puede verse como la fracción de llenado es mayor para ROIs lejanas a la cámara debido principalmente a efectos de perspectiva. Vehículos más cercanos al horizonte, tienden a ocluir más la visión del suelo. Las fluctuaciones en ρ durante embotellamientos se deben a que los vehículos están en general detenidos (el método de segmentación adapta μ restando área de vehículos, ρ disminuye) y se mueven en una suerte de ‘ondas de avance’ (ρ vuelve a subir debido a la detección del movimiento).

Las Figuras 3C y 3D muestran las regiones en movimiento en los dos tiempos indicados. Como se ve, para densidades de llenado bajas, los vehículos son separados en blobs correctamente por el algoritmo de sustracción de fondo. Sin embargo, cuando la fracción de llenado aumenta las burbujas se ‘pegan’. Esto dificulta una segmentación y seguimiento de los vehículos por separado y nos llevó a utilizar la técnica de seguimiento de características para poder estimar la velocidad media de desplazamiento de los vehículos.

3.2. Aplicación del Método de Detección y Correspondencias de Características para Medir la Velocidad Media del Flujo de Tránsito

Para aumentar la eficiencia del método, es importante definir ROIs asociados a la vía que queremos analizar evitando así que SURF detecte muchas características en regiones de poco interés para esta aplicación, como son postes, edificaciones, plantas, etc. Otro problema encontrado con frecuencia fue que el algoritmo SURF detectaba marcas pintadas en la vías (ver Figura 4A) y esto

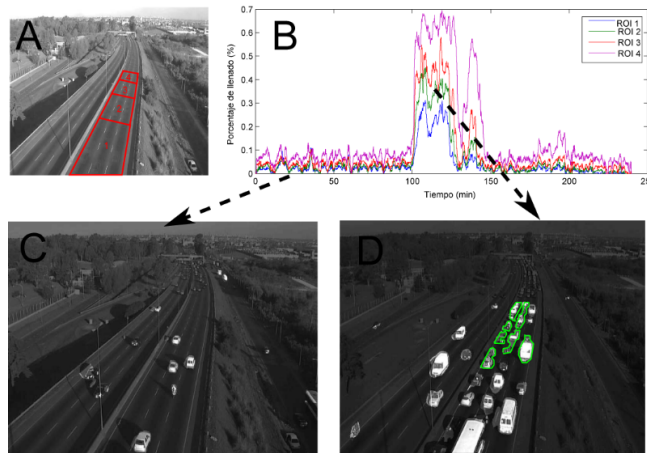


Figura 3: Sustracción de fondo para la medición de la densidad de llenado. A) ROIs en la vía de interés. B) Densidad de llenado en cada ROI en función del tiempo. C) Resultado del algoritmo de sustracción de fondo para $t=20\text{min}$. Se observa baja densidad de llenado y una buena separación de los vehículos clasificados. D) Se observa una alta densidad de llenado asociada a un embotellamiento y el pegado de los vehículos clasificados como foreground.

introducía desviaciones fuertes en la estimación de velocidad. Para resolver este problema utilizamos las características SURF que se encontraban en píxeles clasificados como foreground.

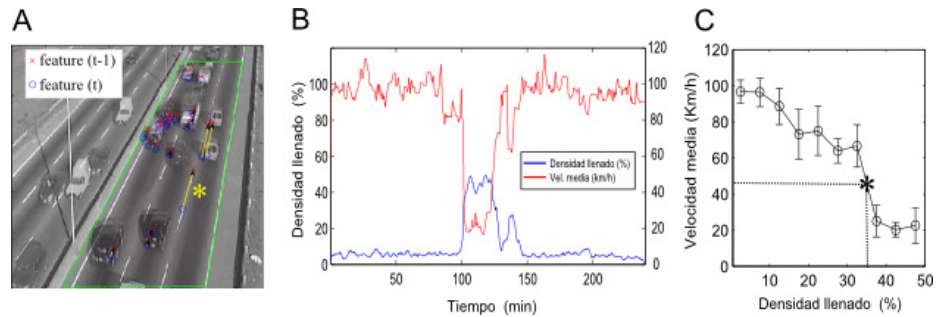


Figura 4: A) Detección y correspondencias de características para medir la velocidad media del flujo de tránsito. La imagen mostrada representa la fusión de los frames $(t-1)$ y (t) . Los círculos azules y las cruces rojas representan los features encontrados por SURF en los frames $(t-1)$ y (t) respectivamente. Representamos por flechas algunas de las asignaciones realizadas. El asterisco representa una asignación espuria que es eliminada por no estar asociada al foreground. B) Velocidad promedio de los vehículos y fracción de llenado en función del tiempo. C) Velocidad promedio en función de fracción de llenado.

3.3. Velocidad promedio de los vehículos en función de la fracción de llenado.

Luego de aplicar este método tenemos un conjunto de N_f características con correspondencias entre un frame en (t) y $(t + 1)$. Pasamos estas características al espacio cartesiano mediante la transformación de la ecuación 1 y calculamos la velocidad media del flujo como:

$$v_{prom}(t) = \frac{1}{N_f} \sum_{n=1}^{N_f} f \cdot \left(\sqrt{\Delta x_n^2 + \Delta y_n^2} \right),$$

donde f es la frecuencia de adquisición de los videos y $(\Delta x_n, \Delta y_n)$ corresponden a los desplazamientos de la característica n en ambos ejes entre $(t - 1)$ y (t) .

En la Figura 4B, mostramos la velocidad promedio y la fracción de llenado en función del tiempo para el mismo intervalo de la Figura 4B. En la Figura 4C mostramos la v_{prom} vs ρ . Como se puede apreciar, existe una relación clara entre ambas variables y para un densidad de 35 % se observa una caída abrupta en la velocidad. A partir de esta observación, definimos la condición de ‘embotellamiento de la vía’ cuando la densidad de llenado sobrepasa el 35 %. Según nuestros resultados esto es decir que los vehículos se desplazan a una velocidad menor a 30 km/h.

Definir la condición de embotellamiento en términos solo de la fracción de llenado tiene la ventaja que no se necesita asignar recursos a detectar, describir y aparear características.

Aplicamos este criterio a toda la serie de videos medidos y detectamos 4 situaciones de embotellamiento de la vía que fueron corroboradas a través de la inspección visual de los videos en alta velocidad. Además, no encontramos ninguna situación de falso positivo.

4. Discusión

En este trabajo implementamos un sistema para la medición del flujo de transporte y detección de embotellamientos en autopistas, basado en las técnicas de ‘sustracción de fondo’ y de ‘seguimiento de características’. El algoritmo fue testeado en 40 horas de mediciones a lo largo de 10 días. Si bien el algoritmo debe ser testeado con mediciones más largas (con mayor variación de condiciones meteorológicas), consideramos que los resultados presentados en este trabajo muestran que el método es una solución factible.

A futuro, nos interesa estudiar las siguientes temáticas: a) aplicación del método a mediciones en condiciones urbanas y de otras autopistas o rutas; b) extensión del método a cámaras con visión de campo amplio (por ejemplo cámaras fisheye); c) desarrollar algoritmos para medir las densidades segregadas por tipo de vehículo; d) estudiar la fusión de información obtenida por este método con la generada por el programa de información de tráfico de la empresa Google y e) extender el método a condiciones más adversas de luminosidad (noche, mal tiempo, etc).

Agradecimientos

Este trabajo se financió a través de los subsidios: Proyecto “Estrategias de automatización, control y modelado aplicadas a la resolución de problemas tecnológicos”, y al programa “Universidad y Transporte” de la Subsecretaría de Gestión y Coordinación de Políticas Universitarias, Ministerio de Educación de la Nación.

Referencias

1. Klein, Lawrence A., Milton K. Mills, and David RP Gibson.: Traffic Detector Handbook - Volume I & II. No. FHWA-HRT-06-139. (2006)
2. <http://www.ausol.com.ar/videos/camara1.html>
3. Piccardi, M.: Background subtraction techniques: a review. In Systems, Man and Cybernetics, 2004 IEEE International Conference on (Vol. 4, pp. 3099-3104). IEEE. (2004, October)
4. Stanganelli D, Oliva D, Noblia M, Safar F.: Calibración de una cámara fisheye comercial con el modelo unificado para la observación de objetos múltiples. Argencon 2014. Bariloche. IEEE Argentina.
5. D. Lowe: Distinctive Image Features from Scale-Invariant Keypoints, Kluwer Academic Publishers. (2004)
6. H. Bay, A. Ess, T. Tuytelaars y L. Van Gool: Speeded-Up Robust Features (SURF). (2008)
7. Coifman, B., Beymer, D., McLauchlan, P., & Malik, J.: A real-time computer vision system for vehicle tracking and traffic surveillance. Transportation Research Part C: Emerging Technologies, 6(4), 271-288. (1998)
8. Hsieh, Jun-Wei, et al.: Automatic traffic surveillance system for vehicle tracking and classification. Intelligent Transportation Systems, IEEE Transactions on 7.2. 175-187. (2006)
9. Muja, Marius and Lowe, David G: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. VISAPP (1), 2. (2009)
10. Lowe, David G: Distinctive image features from scale-invariant keypoints. International journal of computer vision, Springer, 60(2), 91-110 . (2004)
11. MATLAB and Computer Vision System Toolbox Release 2014, The MathWorks, Inc., Natick, Massachusetts, United States.
12. <http://xenia.media.mit.edu/~cwren/interpolator/>