

# Practical alternative to estimate flow and a vehicle speed on an avenue, using Hough Transform and Morphological Operations

Pedro Huamaní-Navarrete, Dr  
Ricardo Palma University, Peru, [phuamani@urp.edu.pe](mailto:phuamani@urp.edu.pe)

*Abstract -- In this article an estimation algorithm of vehicular flow and speed is developed, in a video AVI format for a point of Via Expresa Paseo de la República Avenue from the city of Lima, Perú. As part of digital spatial processing, Hough Transform 500x500 matrix for counted was used for the segmentation stage of the road lanes, on each of the processed frames of the AVI format video. Then, the image was binarized and morphological operators were used to label and to represent each vehicle in group of white pixels, which facilitated the obtaining of the mass center and later compared with a reference line. This allowed the isolation of the vehicles. Also, the process was developed in the MATLAB GUI software, and a total of 30 diurnal AVI format videos each one with 8 minutes of duration were used. Regarding the percentage of success of the vehicular Flow was 95%, while the speed average estimate did not exceed 80 Km/h. In addition, the number of frames processed per second was equal to 2, which allowed a capture every 675 millisecond, to process sequences of images of 25 frames per second.*

*Keywords – speed estimation, vehicular flow, morphological operators, Hough transform.*

Digital Object Identifier (DOI):  
<http://dx.doi.org/10.18687/LACCEI2019.1.1.370>  
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

# Practical alternative to estimate flow and a vehicle speed on an avenue, using Hough Transform and Morphological Operations

Pedro Huamaní-Navarrete, Dr  
Ricardo Palma University, Peru, [phuamani@urp.edu.pe](mailto:phuamani@urp.edu.pe)

*Abstract—In this article an estimation algorithm of vehicular flow and speed is developed, in a video AVI format for a point of Via Expresa Paseo de la República Avenue from the city of Lima, Perú. As part of digital spatial processing, Hough Transform 500x500 matrix for counted was used for the segmentation stage of the road lanes, on each of the processed frames of the AVI format video. Then, the image was binarized and morphological operators were used to label and to represent each vehicle in group of white pixels, which facilitated the obtaining of the mass center and later compared with a reference line. This allowed the isolation of the vehicles. Also, the process was developed in the MATLAB GUI software, and a total of 30 diurnal AVI format videos each one with 8 minutes of duration were used. Regarding the percentage of success of the vehicular Flow was 95%, while the speed average estimate did not exceed 80 Km/h. In addition, the number of frames processed per second was equal to 2, which allowed a capture every 675 millisecond, to process sequences of images of 25 frames per second.*

*Keywords—speed estimation, vehicular flow, morphological operators, Hough transform.*

## I. INTRODUCCIÓN

A lo largo de los años, la población mundial ha aumentado de manera exponencial y, en consecuencia, el uso del transporte motorizado se ha intensificado en diferentes países del mundo, lo que demuestra un alto tráfico de vehículos en las principales calles y avenidas de una gran ciudad. Es el caso de nuestra capital, Lima, donde el tráfico vehicular se vuelve insostenible en ciertos momentos de algunos días de la semana. Por este motivo, uno de los objetivos de este trabajo es la de contribuir con un medidor de flujo de vehículos pasivo, económico y accesible, de tal forma que ayude a las instituciones de transporte a determinar los lugares donde es necesario construir carreteras alternas. De esta manera se contribuiría a descongestionar el tráfico en algunos puntos críticos de la ciudad de Lima. Adicionalmente, con el resultado de esta investigación, el monitoreo del tráfico vehicular ayudaría a establecer nuevos horarios para los cambios automáticos de las luces de los semáforos durante ciertos momentos del día, y con ello se establecería una adecuada implementación de las llamadas ondas verdes automáticas. Y como también, serviría para determinar posibles vehículos que sobrepasan el límite de velocidad en una avenida particular de la ciudad de Lima. Así como estas tres situaciones anteriormente descritas, existen otras mas

donde la necesidad de conocer el flujo y la velocidad de un automóvil en determinado lugar se hace indispensable, por el gran volumen de parque automotor existente en las principales ciudades del mundo.

Por este motivo, este artículo propone la implementación de una aplicación práctica basada en un algoritmo computacional, el cual fue desarrollado en el software Matlab y con apoyo del Simulink y el Toolbox Image Processing, teniendo como principal objetivo obtener y procesar de manera espacial las imágenes contenidas en un video de formato AVI. Para esto, se recurrió al uso de una plantilla trapezoidal para lograr la segmentación de los carriles de la avenida. Esta plantilla trapezoidal se obtuvo con apoyo de una matriz de 500x500 correspondiente a la Transformada de Hough, esta a su vez estuvo conformada por vectores de distancia y ángulo. Luego, se aplicó un extractor de raíz cuadrada y transformaciones morfológicas, apertura, cierre y dilatación para lograr representar cada vehículo como un conjunto de píxeles blancos. Finalmente, para la estimación del flujo y la velocidad del vehículo, se usó una línea de referencia ubicada en cada fotograma del video capturado, el cual se utilizó para comparar con la ubicación del centro de masa de cada grupo de píxeles blancos previamente obtenidos. De esta manera, tanto la estimación del flujo vehicular como la velocidad del mismo, se logró establecer en una avenida altamente transitada como es la avenida Via Expresa en la ciudad de Lima, con orientación de Oeste a Este, es decir en la dirección del distrito Chorrillos hacia Lima. Además, la captura de video se realizó desde un puente vehicular que cruza esta importante avenida.

De esta manera, en el artículo [9] se señala el uso de un UAV como equipo de monitoreo, así como también se realiza la estimación mejorada de parámetros de flujo de tráfico con detección de vehículo, partiendo de videos aéreos propuestos. Como una de las principales herramientas utilizadas, se menciona al Rastreador Kanade – Lucas – Tomasi (KLT), Máquina de soporte vectorial (SVM), movimiento basado en puntos de interés, entre otros. Asimismo, según [10], el movimiento del vehículo es detectado y seguido a lo largo de los cuadros del video, pero utilizando el algoritmo modificado de Lucas-Kanade. Donde, la distancia viajada por el vehículo, es calculado utilizando el movimiento del centroide sobre los cuadros, para luego estimar la velocidad. La aplicación está desarrollada utilizando MATLAB and SIMULINK. También, la velocidad promedio de un carro es determinado de varios cuadros. Así como también, en [8], se hace una revisión de las

Digital Object Identifier (DOI):  
<http://dx.doi.org/10.18687/LACCEI2019.1.1.370>  
ISBN: 978-0-9993443-6-1 ISSN: 2414-6390

principales técnicas de segmentación de imágenes para el monitoreo del tránsito vehicular, del tipo espacial y estadístico. Como por ejemplo la diferenciación de frames (cuadros), la aplicación del flujo óptico, el filtro de Kalman, núcleos de densidad, filtro de Gabor, entre otros. Luego, en el trabajo de [6], se desarrollan técnicas para detectar y clasificar vehículos y peatones a partir de secuencias de imágenes, con el fin de obtener el flujo vehicular y peatonal en las intersecciones de las calles y avenidas. Para ello, utilizan filtrado homomórfico, suma de la diferencia absoluta, fronteras de objetos y descriptores de Fourier.

Asimismo, en [10], se hace una revisión de algunas técnicas de detección y seguimiento de vehículos basados en video estacionarios, y particularmente rectilíneo. Y, para ello, emplean el ROI basado en la simetría, color, sombra, textura, contorno vertical/horizontal, entre otros, así como métodos basados en Wavelets y movimiento. De igual forma, en [12], se hace mención del uso de secuencias de video públicas en internet, con el fin de abordar la creciente necesidad de monitorear las redes de transporte. Y, en [13], afirman que los métodos basados en la simetría o sombra solo funcionan con automóviles no ocluidos, y por tal razón ellos propusieron un enfoque para la detección y el conteo basado en la apariencia del parabrisa. Por ello, se empleó la Transformada de Hough para detectar regiones trapezoidales.

Además, este trabajo desarrollado tiene un enfoque de complemento o ampliación a lo alcanzado en la tesis de pregrado [10], utilizando la misma base de datos empleada. Adicionalmente, existe una variedad de alternativas que permiten la realización de la estimación, pero en la mayoría de casos no es computacionalmente viable.

## II. METODOLOGÍA

### A. Visión general

En este artículo, el proceso de estimación del flujo y la velocidad de vehículos no fue realizado en tiempo real. Sino más bien, se procedió a grabar un conjunto de archivos de video multimedia desde un puente vehicular, el cual cruza la Avenida Vía Expresa Paso de La República. Una vez realizada la grabación se procedió al cambio de formato de video con la finalidad de disminuir el tamaño de las variables. Posteriormente, se continuó con el procesamiento espacial de imágenes complementada con la Transformada de Hough, para poder generar una imagen plantilla que permitió retirar de la escena aquello que no es de interés. Luego, con el uso de otras técnicas de procesamiento espacial, se continuó con la mejora del contraste, la manipulación del histograma, y el uso de las transformaciones morfológicas con el fin de segmentar cada uno de los vehículos, ubicados sobre el asfalto.

### B. Descripción de los datos de imágenes

Los archivos fueron capturados a través de una cámara digital en formato MP4. Además, la cámara tiene una resolución por fotograma igual a 720x1280 píxeles, una codificación de 24 bits/píxel y una velocidad de fotogramas de 30 cuadros/segundos. Sin embargo, debido a la extensa resolución presentada en el formato MP4, este tuvo que convertirse, con ayuda de una aplicación online, a una resolución más baja como es la del formato AVI. 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. Después de la conversión de formato, se utilizó la biblioteca de bloques de procesamiento de imágenes y video de Simulink para leer el archivo multimedia, utilizando particularmente la sub-biblioteca FROM MULTIMEDIA FILE de la opción SOURCES, y las sub-bibliotecas VIDEO TO WORKSPACE y VIDEO VIEWER de la opción SINKS. A continuación, la figura 1 muestra el procedimiento de lectura y visualización del contenido de los archivos con formato AVI, haciendo uso del Simulink del software Matlab.

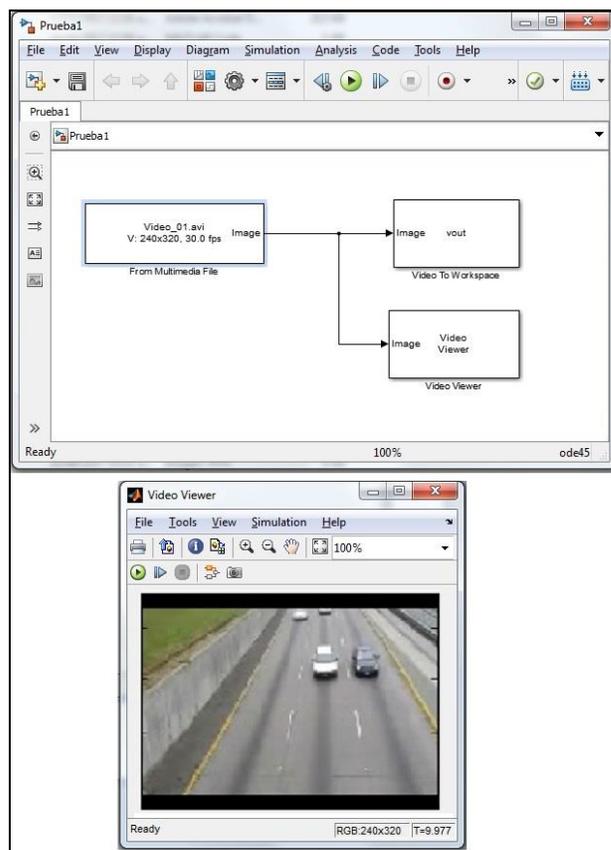


Fig. 1. Procedimiento de lectura de los archivos AVI con el Simulink [10].

### C. Primera etapa: preprocesamiento de fotogramas

En primer lugar, desde la ventana de comandos del software Matlab, se realizó la conversión de cada fotograma o cuadro de un formato de color a un formato de gris. Esto dio

origen a una variable del tipo matriz denominada  $I_{gray\_1}$ . Luego, se procedió a eliminar el ruido impulsivo utilizando un filtro no lineal del tipo mediana, para ello se ejecutaron filas y columnas por bloques de 3x3 píxeles [1]. Luego se continuó con la manipulación del histograma mediante la extracción de la raíz cuadrada, logrando que el resultado de tal operación acentúe más los efectos que se producen en el momento de aclarar más la imagen [2], [7]. Ver la siguiente ecuación:

$$I_{gray\_2} = round\left(255 * \sqrt{\frac{I_{gray\_1}}{255}}\right) \quad (01)$$

De la parte inferior de la figura 1, se observa que los vehículos se encuentran posicionados en el centro de la imagen donde a su vez se forma una imagen trapezoidal, conformada por las líneas amarillas que limitan los tres carriles de dicha avenida. Por lo tanto, se eligió la Transformada de Hough para lograr la segmentación del área de interés, separando la vía auxiliar de la avenida que se está analizando. Pues, en dicha vía no es posible encontrar a un vehículo circulando, a menos que haya un atasco de tráfico y se haya elegido como alternativa para superar este problema. O, como también es posible encontrar un vehículo estacionado, sea por fallas mecánicas o alguna otra razón particular. Por lo tanto, estas dos situaciones no pertenecen a la situación del análisis y desarrollo de este trabajo.

Los videos fueron capturados con una cámara digital colocada sobre un trípode a una distancia fija del objetivo, y dos líneas de color amarillo sobre el asfalto que limitan los tres carriles, fueron identificadas como referencias para el tema de la segmentación. Por esta razón, fue necesario identificar ese color antes de utilizar la Transformada de Hough. La identificación del color se realizó manualmente sobre la imagen en formato de color. Para ello, se realizó una búsqueda por comparación de píxeles pertenecientes a tres rangos establecidos y cercanos a la codificación de ese color en el modelo RGB. Es decir, un rango de valores de píxeles diferentes para cada canal de la imagen o fotograma en color del video AVI:  $I_{color\_R}$ ,  $I_{color\_G}$  e  $I_{color\_B}$ . Esto permitió determinar la nueva imagen de plantilla que fue etiquetada como  $I_{template\_1}$ , y en formato binario.

La siguiente expresión matemática muestra el procedimiento de binarización en cuanto a la comparación de rangos de valores de píxeles por cada canal de color.

$$I_{template\_1} = \begin{cases} 1, & \begin{cases} I_{color\_R} \in [150,175] \\ I_{color\_G} \in [150,175] \\ I_{color\_B} \in [090,110] \end{cases} \\ 0, & \begin{cases} I_{color\_R} \notin [150,175] \\ I_{color\_G} \notin [150,175] \\ I_{color\_B} \notin [090,110] \end{cases} \end{cases} \quad (02)$$

De esta manera en la figura 2 se muestra el resultado de la pre-segmentación por color, en una imagen binaria con píxeles blancos e incompletos en la línea de tendencia de interés. Este resultado se debió por el mismo hecho que los valores de píxeles sobre las líneas amarillas, no siempre pertenecerán a los intervalos definidos en la expresión matemática 2.



Fig. 2. Resultado de la binarización de un fotograma de color del archivo de video AVI.

Luego, se continuó con la aplicación de la Transformada de Hough en la imagen  $I_{template\_1}$ , dando lugar a la nueva matriz etiquetada como  $I_{template\_2}$ . Para este propósito, la aplicación de esta transformación hizo uso de la imagen segmentada previamente,  $I_{template\_1}$ , para pasar a través de ella un cierto número de rectas con diferentes pendientes y para cada píxel blanco. Por lo tanto, en primer lugar, se construyó una matriz TH con dimensión 500x500 y con todos sus elementos iguales a cero. Las filas representaron el ángulo para la variable,  $\theta$ , subdividido en 500 partes; y, las columnas representaron la distancia variable,  $\rho$ , entre el origen de las coordenadas y un píxel blanco arbitrario de la imagen  $I_{template\_1}$ . El vector de distancia también se subdividió en 500 divisiones. Por lo tanto, los intervalos e incrementos para las filas y las columnas de la matriz TH, se muestran a continuación:

$$\begin{cases} \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], \Delta\theta = \frac{\pi}{500} \\ \rho \in \left[0, \sqrt{m^2 + n^2}\right], \Delta\rho = \frac{\sqrt{m^2 + n^2}}{500} \end{cases} \quad (3)$$

Donde “m” y “n” representan the resolución de la imagen pre-segmentada  $I_{template\_1}$ .

Luego, con las filas y columnas subdivididas, se procedió a recorrer toda la imagen a través del eje de coordenada  $\theta$ , tomando en cuenta el incremento señalado en la ecuación 3. Posteriormente, se aplicó la expresión matemática 4 para determinar las distancias  $\rho$ , e incrementar de uno en uno cada celda de la matriz TH, dependiendo de la localización de  $\rho$  en los intervalos correspondientes.

$$\rho = x * \cos(\theta) + y * \sin(\theta) \quad (4)$$

De la expresión anterior, “x” e “y” representan las coordenadas de todos los píxeles blancos arbitrarios de la imagen etiquetada como  $I_{template\_1}$ .

Luego, se procedió a la búsqueda de la recta más frecuente. Y esto se logró buscando la celda de la matriz TH con el valor numérico más alto. Y, como esa celda coincide con una coordenada de los ejes  $\theta$  y  $\rho$ , se procedió a la reconstrucción de las dos líneas para formar el trapecio en la imagen etiquetada como  $I_{template\_2}$ . Ver la figura 3.

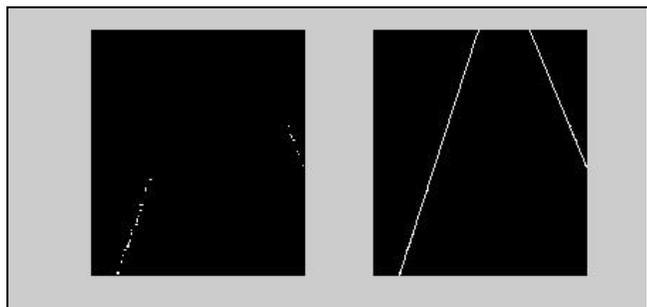


Fig. 3. Resultado de la aplicación de la Transformada de Hough.

Obtenida la imagen etiquetada como  $I_{template\_2}$ , después de aplicar la Transformada de Hough, se procedió a formar la nueva imagen plantilla, etiquetada como  $I_{template}$ , la cual toma la forma de una figura geométrica cóncava. Para esta construcción, primero se localizó un píxel blanco en el centro de la imagen  $I_{template\_2}$ , y después se procedió a llenar el interior de la figura geométrica cóncava con una operación de dilatación, con un elemento estructural tipo cuadrado y tomando como límite las líneas obtenidas de la Transformada de Hough. Ver la ecuación 5.

$$EE_{DILATE} = \left\{ \begin{matrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{matrix} \right\} \quad (5)$$

Una vez que la plantilla fue construida, se procedió a multiplicarla con la imagen  $I_{gray\_2}$ . Ver la ecuación 6.

$$I_{gray\_3} = I_{gray\_2} * I_{template} \quad (6)$$

De esta manera, en la imagen central de la figura 4, se muestra el resultado de la construcción de la plantilla con el apoyo de la Transformada de Hough y la operación morfológica dilatación, lo cual permitió llenar de píxeles blancos el área de interés. Mientras que la imagen derecha de la misma figura, muestra el resultado de multiplicar la imagen de la plantilla con la imagen original ubicada en el lado izquierdo de dicha figura.

De esta forma, teniendo en cuenta las normas de tránsito de la ciudad de Lima, la circulación de vehículos en la avenida

Via Expresa Paseo de la República no debe ser superior a 80 Km/h. Además, asumiendo que cada imagen del video capturado presenta aproximadamente 15 metros de longitud de fondo, decidimos trabajar con una frecuencia suficiente de 2 cuadros / segundo. El supuesto de 15 metros de longitud del fondo, lo apoyamos porque las líneas blancas discontinuas que dividen dos carriles de vehículos presentan una longitud de 3.0 metros. Y, la separación entre dos líneas blancas discontinuas y seguidas es de 5.0 metros. Siempre que sea un área urbana [3]. Por lo tanto, usar 25 cuadros / segundo equivale a capturas de cuadros cada 40 milisegundos, y con la información aproximada de 15 metros de longitud, se alcanza la relación presentada en la ecuación 07. Y con una regla de tres simples, obtenes el resultado del tiempo requerido para la captura de imágenes.

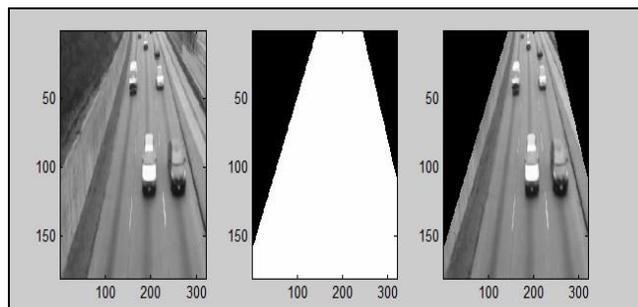


Fig. 4. Resultado de la construcción de la plantilla para la segmentación de los fotogramas del archivo de video AVI.

$$\begin{aligned} 25 \text{ frames} &\rightarrow 1 \text{ sec.} \\ 01 \text{ frame} &\rightarrow 40 \text{ mSec.} \end{aligned} \quad (7)$$

$$time = \frac{15 * 3600}{80000} * 10^{-3} = 675 \text{ mSec.}$$

De esta manera, sería necesario capturar las imágenes a 675 milisegundos para poder apreciar el vehículo de interés, al menos una vez en el cuadro capturado y considerando que no excede la velocidad límite de dicha ruta.

El video se adquirió en una ubicación al aire libre, por lo que cada cuadro presenta una distribución segmentada de la intensidad de los píxeles, debido a la influencia de la luz natural en función del tiempo de captura. Por este motivo, decidimos usar un filtro de sombrero de copa blanco sobre la imagen gris para resaltar los detalles en zonas de bajo contraste o poca iluminación. Ver las ecuaciones 9, 10 y 11.

Además, el elemento estructural (EETOP-HAT), representa a una línea vertical construida con 10 píxeles. De esta manera, la figura 5 representa el efecto del filtrado Top-Hat.

$$EE_{TOP-HAT} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{array} \right\}_{10 \times 1} \quad (8)$$

$$DI_{gray\_3} = \left\{ z \mid (EE_{TOP-HAT})_z \cap I_{gray\_3}^c \neq \Phi \right\} \quad (9)$$

$$I_{gray\_4} = \left\{ z \mid (\hat{EE}_{TOP-HAT})_z \cap DI_{gray\_3} \neq \Phi \right\} \quad (10)$$

$$I_{gray\_filt} = I_{gray\_3} - I_{gray\_4} \quad (11)$$

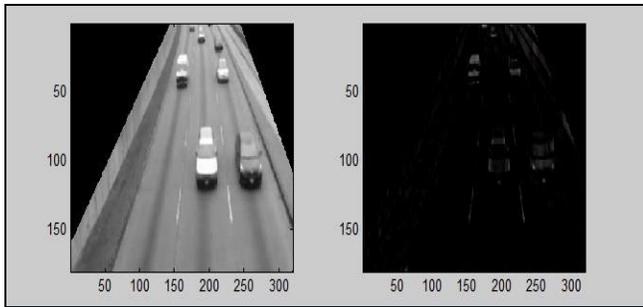


Fig. 5. Resultado del filtrado Top-Hat en una imagen segmentada.

Luego, se realizó la binarización segmentada en cada cuadro. Primero, a la subimagen formada por las primeras 100 filas y luego a la subimagen con el número restante de filas. Ver la siguiente ecuación.

$$I_{bin} = \begin{cases} I_{gray\_filt}(1:100,:) > 0.1 & , I_{bin} = 1 \\ I_{gray\_filt}(101:end,:) > 0.2 & , I_{bin} = 1 \end{cases} \quad (12)$$

Una vez realizada la binarización segmentada, se continuó con la aplicación de 03 tipos de transformaciones morfológicas: apertura, cierre y dilatación, todas con diferentes elementos estructurantes (EE). Esta determinación se realizó después de realizar pruebas de prueba y error, así como también por la característica de la imagen procesada que contiene trazos verticales y horizontales en cada vehículo. El procedimiento de aplicación de las operaciones morfológicas, en secuencia, tiene una base similar a la utilizada en la tesis [4] y [5].

De esta manera, las ecuaciones (13), (14) y (15) muestran los tres elementos estructurales utilizados. Después de estas operaciones, se dio lugar a una nueva imagen binaria llamada  $I_{bin\_transf}$ . Ver la figura 6.

$$EE_{CLOSE} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{array} \right\}_{20 \times 1} \quad (13)$$

$$EE_{OPEN} = \{ 1, 1, 1, 1, \dots, 1 \}_{1 \times 20} \quad (14)$$

$$EE_{DILATE} = \left\{ \begin{array}{c} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{array} \right\}_{20 \times 1} \quad (15)$$

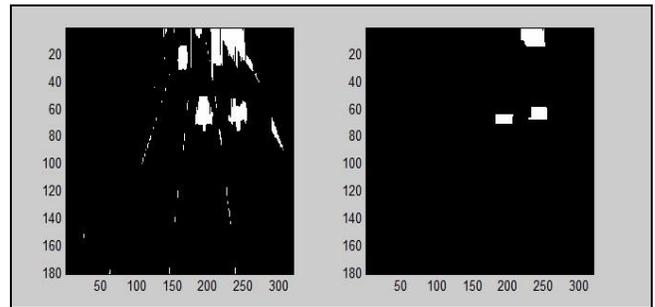


Fig. 6. Resultado de la transformación morfológica sobre una imagen binaria.

#### D. Segunda etapa: estimación del flujo vehicular

Para determinar la estimación del flujo del vehículo, se trazó una línea de referencia en cada cuadro para tener una guía en el momento del conteo del vehículo. Esto significa que un número de objetos blancos que cruzaron la línea de referencia se contaron en función de un tiempo determinado y de una cantidad fija de imágenes.

#### E. Tercera etapa: estimación de la velocidad vehicular

Adicionalmente, la línea de referencia se situó en la fila 60 de cada fotograma capturado para realizar el recuento de píxeles blancos acumulados en la línea. El tamaño promedio del vehículo representado como un objeto blanco consistió en una matriz con una resolución de 25x20 píxeles. Por lo tanto, la imagen binaria,  $I_{bin\_transf}$ , fue etiquetada usando un vecindario de 4, dando lugar a una nueva matriz llamada  $I_{bin\_label}$ . Luego, en esta última matriz, se calculó el centro de masa de cada área blanca etiquetada y utilizando las filas y columnas promedio. Ver el siguiente pseudocódigo.

```

for k = 1 to N
  if  $I_{bin\_label}(i, j) = k$  then
     $x \leftarrow i$ 
     $y \leftarrow j$ 
  end if
end for

```

También, ver las siguientes ecuaciones:

$$xx = \text{round} \left( \frac{\sum_{j=1}^N x_j}{N} \right) \quad (16)$$

$$yy = \text{round} \left( \frac{\sum_{j=1}^N y_j}{N} \right) \quad (17)$$

Donde, N representa el número etiquetado de áreas blancas, en cada uno de los cuadros analizados.

Y, utilizando un condicional, se procedió a observar el aumento de la variable de cuenta definida como Count\_1, siempre que el vector del centro de masa "xx" sea menor que la última fila del cuadro y mayor que la línea de referencia. A continuación, se muestra el siguiente pseudocódigo.

```

N ← number_labeled_white_zones
END ← [100, 100, 100, ..., 100]1x100
LINE ← 60
Count1 ← 0
for j = 1 to N
  if  $xx(j) \geq \text{LINE}$  AND  $xx(j) \leq \text{END}(j)$  then
    Count_1 ← Count_1 + 1
    END(j) ← xx(j)
  else
    END(j) = 100
  end if
end for

```

Y, con la variable "Count\_1", se procedió a obtener el flujo vehicular para cada video AVI. Por lo tanto, si cada video tenía una duración de 8 minutos, la división se realizó entre 480 segundos.

$$FLOW = \frac{\text{Count}_1}{480} \left[ \frac{\text{cars}}{\text{sec.}} \right] \quad (18)$$

Por otro lado, para la estimación de la velocidad promedio, se utilizaron los centros de masa de los cuadros procesados a partir de la determinación de una diferencia para dos cuadros consecutivos, resultado de un promedio de 13

cuadros que equivale a 0.5 segundos. Hecha la diferencia de los centros de masa, se usó una regla de tres simple tomando como referencia la frecuencia de cuadros por segundo en el video capturado, y el parámetro "Dist" en cm/cuadro. Aproximadamente, este parámetro representa un número de centímetros por cada cuadro en relación con los 4 píxeles de viaje de un vehículo cuando se procesan dos cuadros consecutivos.

$$\text{Dist [cm/frame]} \rightarrow 1/25 \text{ sec.}$$

$$\text{speed} \rightarrow 1 \text{ sec.}$$

$$\text{speed [Km/h]} = \frac{\text{Dist[cm/frame]} * 3600}{1/25[\text{sec/frame}] * 100000} \quad (19)$$

### III. RESULTADOS EXPERIMENTALES

Con respecto a la estimación del flujo vehicular, se realizó la simulación en un conjunto de 30 videos digitales durante el día, con una duración de 8 minutos cada uno. Por lo tanto, cada video se procesó por separado y los resultados se obtuvieron con márgenes de error inferiores al 5.0%. Además, todos los videos fueron capturados durante el día para los cinco días de la semana, de lunes a viernes.

La Tabla I muestra el resultado de la estimación del flujo vehicular para 12 videos AVI con una duración de 08 minutos cada uno; Donde se puede ver la comparación de resultados obtenidos tanto por el flujo estimado como por el flujo real.

TABLE I. RESULTADOS DE EL FLUJO VEHICULAR PARA UN TOTAL DE 12 VIDEOS AVI.

	VIDEOS	Flujo aproximado (cars/sec)	Flujo verdadero (cars/sec.)
01	Video_01.avi	2.30	2.30
02	Video_02.avi	1.80	1.80
03	Video_03.avi	1.70	1.60
04	Video_04.avi	1.80	1.70
05	Video_05.avi	1.20	1.20
06	Video_06.avi	2.20	2.20
07	Video_07.avi	2.20	2.20
08	Video_08.avi	2.20	2.30
09	Video_09.avi	1.90	1.90
10	Video_10.avi	2.00	2.00
11	Video_11.avi	2.10	2.20
12	Video_12.avi	2.10	2.00

Luego, se continuó con la obtención de la suma de la diferencia al cuadrado entre los flujos aproximado y verdadero. Inmediatamente, se obtuvo el complemento y se

tomó el porcentaje para calcular la tasa de éxito en términos de la estimación del flujo vehicular.

$$Dif\_flow = \sum (App\_Flow - True\_Flow)^2$$

$$success\_rate(\%) = (1 - Dif\_flow) \times 100 \quad (20)$$

$$success\_rate(\%) = 95.00\%$$

Con respecto a la estimación de la velocidad del vehículo, el algoritmo desarrollado obtiene la velocidad promedio aproximada cada medio segundo, lo que equivale a procesar aproximadamente 12 cuadros consecutivos. De esta manera, la Tabla II muestra los resultados de la estimación de la velocidad del vehículo según el día de la semana de la semana.

TABLE II. RESULTADOS DE LA ESTIMACIÓN DE LA VELOCIDAD VEHICULAR EN 12 VIDEOS AVI

	VIDEOS	Día de semana laborable.	Velocidad promedio estimada.
01	Video_01.avi and Video_08.avi	Lunes	78.2 Km/h
02	Video_02.avi, Video_04.avi and Video_09.avi	Martes	71.8 Km/h
03	Video_03.avi, Video_05.avi and Video_10.avi	Miércoles	77.7 Km/h
04	Video_06.avi and Video_11.avi	Jueves	75.3 Km/h
05	Video_07.avi and Video_12.avi	Viernes	70.0 Km/h

Además, el algoritmo de estimación del flujo y la velocidad se desarrolló en la GUI de Matlab, para facilitar su uso después de capturar el video AVI. La Figura 7 muestra la interfaz de dos botones, uno para el conteo de vehículos y el otro para limpiar los resultados.

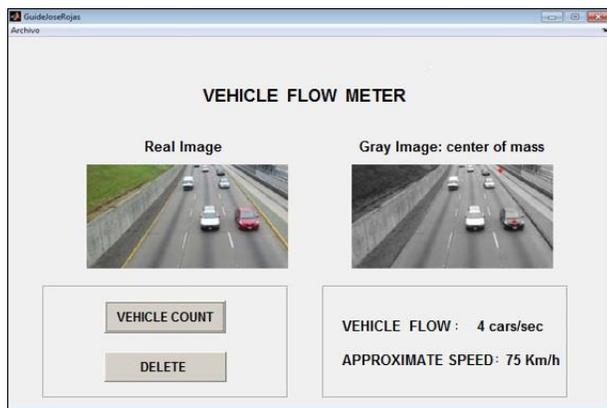


Fig. 7. Interface Matlab GUI mostrando la velocidad y el flujo de manera estimada [10].

## DISCUSIONES

Este trabajo de investigación se centra en un lugar específico en la ciudad de Lima-Perú, donde aún no se han realizado pruebas de medición de flujo y velocidad vehicular a través de algoritmos de procesamiento de imágenes digitales. Además, los trabajos de investigación de la referencia bibliográfica coinciden en el uso de operadores morfológicos para la etapa de segmentación sin embargo utilizando diferentes elementos estructurantes, así como el empleo de la Transformada de Hough para determinar regiones diferentes. Por lo tanto, en este artículo no es posible comparar la efectividad del algoritmo propuesto con otros existentes en la bibliografía, por el simple hecho de usar diferentes escenarios.

## CONCLUSIONES

En relación con la determinación de la velocidad promedio estimada para los vehículos que circulan en un punto de la avenida Vía Expresa de la ciudad de Lima, se concluye que este cálculo presentó ciertos errores, especialmente cuando el algoritmo de segmentación por etiquetado mostró algunos casos de autos fantasmas.

Además, se concluye que el tránsito es mayor en el primer y último día laborable de la semana, es decir, los lunes y viernes, esto se debe a muchas razones diarias y habituales en los conductores, de manera similar como se obtuvo en [10].

Además, no se pudo verificar el grado de error obtenido al obtener la velocidad promedio, ya que era imposible conocer el valor verdadero de la velocidad porque no se utilizó un radar para esta medición. Pues los resultados de la estimación de la velocidad fueron muy próximos a los obtenidos en [10].

## REFERENCIAS

- [1] E. Cuevas, D. Zaldivar y M. Pérez, *Procesamiento Digital de Imágenes usando Matlab y Simulink*. México: Editorial Alfaomega, 2010.
- [2] G. Pajares y J. De la Cruz. *Visión por Computador Imágenes Digitales y Aplicaciones*, 2ª ed. España: Alfaomega Ra-Ma, 2008.
- [3] Manual de Dispositivos “Marcas en el pavimento y bordes de pavimento” [En línea]. Disponible en: <https://www.mtc.gob.pe>
- [4] V. Cabello Hernández, “Reconocimiento Inteligente de Flujo Vehicular basado en el Procesamiento de Imágenes”, tesis de Ingeniero Electrónico, Universidad Michoacana de San Nicolás de Hidalgo-México, 2010 [En línea]. Disponible en: <http://bibliotecavirtual.dgb.umich.mx:8083/jspui/bitstream/123456789/5046/1/RECONOCIMIENTOINTELIGENTEDEFUJOVEHICULARBASADOENELPROCESAMIENTODEIMANES.pdf> [Accedido: 01-May-2017]
- [5] A. Chávez, “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 del Perú, 2007 [En línea]. Disponible en: [http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/224/CHAVEZ\\_CORDOVA\\_ALEX\\_ALGORITMO\\_INGRESO\\_VEH% C3% 8DCULOS.pdf?sequence=1](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/224/CHAVEZ_CORDOVA_ALEX_ALGORITMO_INGRESO_VEH% C3% 8DCULOS.pdf?sequence=1) [Accedido: 05-Jun-2017]

- [6] A. Granados y J. Marin, “Detección de Flujo Vehicular basado en visión artificial”, *Scientia et Technica*, Año XIII, no. 35. pp. 163-166, agosto 2007.
- [7] A. Alcain e C. Dos Santos, *Fundamentos do Processamento de Sinais de Voz e Imagen*. Rio de Janeiro, Brasil: Interciência, 2011.
- [8] N. Leal, E. Leal y J. Branch, “Sistemas de monitoreo de tránsito vehicular basados en técnicas de segmentación de imágenes”, artículo de la Universidad Nacional de Colombia, Medellín, 2010 [En línea]. Disponible en: <http://www.bdigital.unal.edu.co/28799/1/26655-93566-1-PB.pdf>
- [9] Najiya K. V. and Archana M, “UAV Video Processing for Traffic Surveillance with Enhanced Vehicle Detection”, Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018), pp. 662-668.
- [10] G. Wang, D. Xiao and J. Gu, “Review on vehicle detection based on video for traffic surveillance”, in IEEE International Conference on Automation and Logistics, Qingdao-China, September 2008, pp. 2961-2966.
- [11] J. Rojas, “Técnica matemática de medición de flujo vehicular por visión artificial”, Tesis de Pregrado, Universidad Tecnológica del Perú, Lima, Perú, 2015.
- [12] P. Loureiro, R. Rosseti and R. Braga, “Video Processing Techniques for Traffic Information Acquisition Using Uncontrolled Video Streams”, Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, MO, USA, October 2009, pp 127-133.
- [13] P. Huy and L. Byung-Ryong, “Front-view car detection and counting with occlusion in dense traffic flow”, *International Journal of Control, Automation and Systems*, vol. 13, no. 5, pp. 1150-1160, October 2015.