

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



ALGORITMO DE ESTIMACIÓN DEL NÚMERO DE ELEMENTOS  
MOVILES EN VIDEOS DIGITALES ORIENTADO A LA GESTIÓN DEL  
TRÁFICO VEHICULAR

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

**Jorge Gerardo Quesada Pacora**

**ASESOR: Paul Antonio Rodriguez Valderrama**

Lima, 2015



Para mis seres queridos.

## Resumen

La gestión automatizada del tránsito es un campo de investigación que integra una variedad de tecnologías y se orienta principalmente a mejorar el flujo vehicular, haciendo uso de cámaras y otros sensores para recabar información sobre el estado del tráfico. En este contexto, las técnicas de procesamiento de imágenes digitales permiten realizar diversos análisis del entorno urbano, tales como detección, conteo y seguimiento de peatones y vehículos. Entre las capacidades que requieren los sistemas que realizan dicha gestión, la estimación del número de vehículos en circulación es una de las principales. Para obtener una estimación adecuada del número de elementos móviles en un video, primero debe realizarse una adecuada segmentación de dichos elementos. Existe una gran variedad de métodos que realizan dicha segmentación, sin embargo PCP (Principal Component Pursuit) es considerado el estado del arte para el modelado de fondo de videos digitales en donde el sensor (cámara) es estático.

El objetivo de la presente tesis es el diseño de un algoritmo que estime el número de vehículos presentes en un video digital de tránsito, que tenga como etapa de pre-procesamiento la segmentación de movimiento mediante PCP (utilizando una librería independiente) y funcione de manera semiautomática. Se busca también proponer una estrategia adecuada para dividir las etapas del algoritmo, de modo que éste pueda ser descrito como un conjunto flexible de bloques, implementable en cualquier plataforma o entorno.

El método propuesto se divide en dos bloques principales: entrenamiento (supervisado) y conteo (automático). Ambos bloques realizan la extracción del movimiento mediante PCP. El primer bloque genera los parámetros necesarios para el conteo mediante un análisis de las dimensiones de los objetos móviles. El segundo realiza el conteo mediante el uso de los parámetros proporcionados por la etapa de entrenamiento.

Para evaluar el rendimiento del algoritmo, este será implementado en el entorno de programación Matlab, y se generará una base de datos propia. Dicho rendimiento será evaluado en dos dimensiones: el número instantáneo de vehículos en escena y el número de vehículos que atraviesan una “puerta virtual” en un tiempo determinado.

En el primer capítulo de la tesis se define puntualmente el problema que se busca resolver. En el segundo capítulo se revisan los métodos y algoritmos más populares para segmentar el movimiento, haciendo especial énfasis en PCP. Las consideraciones de diseño y los detalles del algoritmo se especifican en el capítulo tres. Finalmente, se presentan los resultados obtenidos en el capítulo cuatro, seguido de las conclusiones y recomendaciones al respecto.

# Índice general

<b>Introducción</b>	<b>1</b>
<b>1. Fundamentación</b>	<b>2</b>
1.1. Presentación del asunto de estudio . . . . .	2
1.2. Objetivos . . . . .	3
1.3. Importancia y justificación . . . . .	3
1.4. Limitaciones . . . . .	4
<b>2. Segmentación del movimiento en videos digitales</b>	<b>5</b>
2.1. Estado de la investigación . . . . .	5
2.1.1. Diferenciación Temporal . . . . .	5
2.1.2. Modelado y substracción de fondo . . . . .	6
2.1.2.1. Promediado . . . . .	7
2.1.2.2. Gaussiano Unico . . . . .	7
2.1.2.3. Filtro Kalman . . . . .	8
2.1.2.4. Estimación Modal . . . . .	8
2.1.3. Mixtura de Gaussianos . . . . .	9
2.1.4. Flujo Optico . . . . .	9
2.2. Principal Component Pursuit . . . . .	10
<b>3. Diseño del Algoritmo Propuesto</b>	<b>12</b>
3.1. Base de datos utilizada para el diseño . . . . .	12
3.2. Algoritmo Propuesto . . . . .	12
3.3. Etapa de Entrenamiento . . . . .	13
3.3.1. Segmentación del movimiento . . . . .	13
3.3.1.1. Extracción de movimiento mediante PCP . . . . .	13
3.3.1.2. Segmentación Unimodal . . . . .	14
3.3.2. Extracción de características . . . . .	15

3.3.3. Estimación de parámetros para el procesamiento . . . . .	15
3.4. Etapa de Conteo . . . . .	16
3.4.1. Procesamiento espacial y temporal . . . . .	16
3.4.1.1. Procesamiento espacial . . . . .	17
3.4.1.2. Procesamiento temporal . . . . .	18
<b>4. Implementación y resultados computacionales</b>	<b>20</b>
4.1. Consideraciones de implementación . . . . .	20
4.1.1. Software de implementación . . . . .	20
4.1.2. Base de datos generada para las pruebas del algoritmo . . . . .	20
4.1.3. Características de las imágenes para la implementación . . . . .	20
4.2. Descripción de la implementación . . . . .	21
4.3. Resultados obtenidos . . . . .	22
4.3.1. Resultados del conteo instantáneo . . . . .	22
4.3.2. Resultados de la puerta virtual . . . . .	23
4.3.3. Análisis de los resultados . . . . .	23
<b>Conclusiones</b>	<b>26</b>
<b>Recomendaciones</b>	<b>27</b>
<b>Bibliografía</b>	<b>28</b>

# Introducción

La gestión automática del tránsito [1, 2] es una herramienta cuyo uso se ha difundido de forma sostenida y rápida durante los últimos años. Uno de los factores que empuja esta difusión es la constante evolución de las tecnologías implicadas, tanto a nivel de hardware como de software. En particular, las tecnologías relacionadas con la detección y conteo de objetos en movimiento (peatones y vehículos) han visto un avance significativo en el desarrollo de algoritmos en esa área.

El procesamiento de imágenes es un campo en constante desarrollo con aplicaciones en diversas áreas. Específicamente, la investigación relacionada con la segmentación de movimiento ha generado una variedad de métodos que permiten extraer los píxeles móviles de una secuencia de video digital. El estado del arte actual en dicha área es un método denominado Principal Component Pursuit (PCP) [3, 4], el cual permite realizar la mencionada segmentación con una fidelidad superior a métodos anteriores.

Existe una brecha entre el estado del arte correspondiente a los algoritmos de conteo de objetos y los métodos de segmentación de movimiento (los primeros deben, necesariamente, basarse en los segundos): a la fecha no se ha propuesto un método que estime el número de objetos en un video segmentando el movimiento mediante PCP.

Es por ello que el presente trabajo busca proponer un algoritmo de conteo vehicular que tenga como etapa de preprocesamiento una segmentación mediante PCP, y proporcione un rendimiento comparable al estado del arte actual.

# Capítulo 1

## Fundamentación

### 1.1. Presentación del asunto de estudio

Actualmente, la gestión automatizada del tránsito es un campo de investigación que integra una variedad de tecnologías y se orienta principalmente a mejorar el flujo vehicular, haciendo uso de cámaras y otros sensores para recabar información sobre el estado del tráfico. Dicha gestión es realizada mediante la implementación de variaciones en los medios convencionales (usualmente constantes) de control de tráfico, en función de la información obtenida. Algunas de estas variaciones consisten en adaptar dinámicamente los límites de velocidad [5] y la regulación de tiempo de los semáforos [6], además de habilitar temporalmente arcenes o carriles de emergencia [7] en horas de máxima demanda vehicular.

En el contexto del recabado de información sobre el estado del tráfico, el uso de cámaras de video digitales permite emplear técnicas de procesamiento de imágenes para realizar diversos análisis del entorno urbano, tales como detección, conteo y seguimiento de peatones y vehículos [8]. En particular, existe un gran número de trabajos orientados a contabilizar el número de vehículos presentes en un video [9, 10, 11, 12, 13], ya sea mediante la estimación de cuántos están presentes en una escena en cada instante, o el conteo acumulativo de aquellos que pasan por una determinada zona durante cierto tiempo (concepto de “puerta virtual”).

Para realizar adecuadamente cualquier análisis de los elementos móviles en un video, primero debe realizarse una segmentación efectiva de dichos elementos. Existe una gran variedad de métodos que realizan dicha segmentación [8, 14], sin embargo PCP (Principal Component Pursuit) es actualmente considerado el estado del arte [4] para el modelado de fondo de videos digitales en donde el sensor (cámara) es estático.

Una gran mayoría de los métodos existentes de conteo de objetos se basan en segmentaciones de movimiento mediante mixtura de gaussianos [10, 15], flujo óptico [16],

sustracción de fondo [17], y otros enfoques populares. Se observa, sin embargo, que a la fecha aún no se ha propuesto un método que estime el número de objetos en un video segmentando el movimiento mediante PCP. Dicho es el marco problemático en el cual se plantea el presente trabajo.

## 1.2. Objetivos

- Proponer un algoritmo eficiente de conteo de los vehículos presentes en una secuencia de tránsito, basado en una optimización PCP y que funcione de modo semiautomático.
- Proponer una estrategia adecuada para dividir las etapas del algoritmo de modo que éste pueda ser descrito como un conjunto flexible de bloques, implementable en cualquier plataforma o entorno.
- Implementar una aplicación en el entorno MATLAB que sea capaz de estimar el número de vehículos en una secuencia de tráfico.

## 1.3. Importancia y justificación

El conteo de vehículos en escenas de tránsito es una pieza vital en el proceso de control automatizado de éste. El rendimiento de un sistema de gestión automática del tránsito está en gran medida relacionado con la fiabilidad de la información alimentada a dicho sistema (en particular, el número de vehículos existentes en un lugar y momento determinados).

Como se mencionó anteriormente, los métodos de conteo vehicular actualmente existentes se encuentran basados en técnicas de segmentación de movimiento cuyo rendimiento es inferior al estado del arte actual para dicha aplicación (cámaras estáticas). Cabe mencionar también que la mayoría de enfoques de conteo existentes [9, 10, 11, 12, 13] realizan dicho conteo como consecuencia de un seguimiento o “tracking” de cada vehículo en escena, lo cual implica un costo computacional adicional que no es inherente al problema.

Se espera que la implementación de un algoritmo de conteo vehicular basado en una optimización PCP proporcione resultados de rendimiento comparables a aquellos que conforman el estado del arte actual. Dado que el desarrollo de implementaciones optimizadas de PCP es un área de investigación que se encuentra bastante activa [4], la idea de contar con un algoritmo de conteo vehicular basado en dicha técnica es alentadora.



## 1.4. Limitaciones

- Existen cámaras de monitoreo urbano con bases y lentes rotativas. El presente trabajo se orienta particularmente a videos con un fondo estático.
- Algunas cámaras posicionadas muy cerca del pavimento muestran una vista “horizontal” de la escena donde los vehículos se ocluyen entre sí. El tratamiento de videos con altos niveles de oclusión vehicular se encuentra fuera del alcance de esta tesis.



## Capítulo 2

# Segmentación del movimiento en videos digitales

Como se mencionó anteriormente, un paso importante previo a realizar cualquier análisis de los elementos móviles de una secuencia de video, es segmentar el movimiento mediante un método adecuado. En las últimas décadas se han desarrollado múltiples trabajos que plantean e implementan diversos enfoques para realizar dicha segmentación, mejorando cada vez la calidad de resultados y la eficiencia computacional. En [8] y [14] se categorizan varios de los métodos existentes de acuerdo a una serie de criterios. La primera sección del presente capítulo presenta una síntesis de dichas clasificaciones, resaltando las características favorables y desfavorables de cada enfoque. La segunda detallará de forma particular el método de optimización PCP (Principal Component Pursuit).

### 2.1. Estado de la investigación

#### 2.1.1. Diferenciación Temporal

La diferenciación temporal [18] es el más simple de los métodos existentes para segmentar el movimiento. Consiste en calcular la diferencia punto a punto entre dos o más cuadros consecutivos de una secuencia, detectando las regiones móviles donde el resultado es superior a un determinado valor umbral. Por ejemplo, para un cuadro  $n$ -ésimo, y un umbral  $T$  se tiene la condición de umbralización:

$$|I_n(x, y) - I_{n-1}(x, y)| > T \quad (2.1)$$

El hecho de requerir solamente la resta de los cuadros trae como consecuencia que la



Figura 2.1: (a),(b) Cuadros consecutivos de tránsito vehicular, en los cuales los vehículos de la derecha están en movimiento y los de la izquierda están detenidos en el semáforo (c) Segmentación resultante de aplicar diferenciación temporal

complejidad computacional de este enfoque sea extremadamente baja. Sin embargo, uno de los principales inconvenientes que presenta es el problema de apertura: las regiones móviles son detectadas sólo parcialmente (figura 2.1). Además, dado que sólo se toma en cuenta un número muy reducido de cuadros (usualmente dos o tres), las regiones móviles detectadas son relativas solamente a los cuadros en cuestión. Por ejemplo, para el caso particular de un video de tránsito vehicular, un vehículo que se detiene frente a un semáforo en rojo dejaría de ser detectado luego de pocos cuadros.

### 2.1.2. Modelado y substracción de fondo

La substracción de fondo [19, 8] es un método popular para extraer el movimiento en los casos en que el fondo de la secuencia de video es relativamente estático. De forma análoga al enfoque anterior, se calcula la diferencia punto a punto entre cada cuadro y un modelo de fondo determinado, umbralizando el resultado. Por lo tanto, para un modelo de fondo  $B$ , se clasificará como móviles a las regiones que cumplan con:

$$|I_n(x, y) - B(x, y)| > T \quad (2.2)$$

para el cuadro  $n$ -ésimo y un umbral  $T$ .

Al igual que en el método de diferenciación temporal, la detección de movimiento depende sólo de una resta de cuadros y posterior umbralización, lo cual implica un bajo consumo de recursos computacionales. En este caso, la calidad de los resultados y el costo computacional asociado dependen de la forma de obtener el modelo de fondo. Por ejemplo, la Figura 2.2 muestra una substracción de fondo donde (a) es un cuadro cualquiera, (b) es el modelo de fondo y (c) es el resultado de la segmentación. Observando la imagen de fondo, se puede notar que algunos autos han sido parcialmente integrados en dicho fondo (lado izquierdo de la imagen), lo cual puede ocasionar errores de segmentación en esa región (el vehículo rojo es

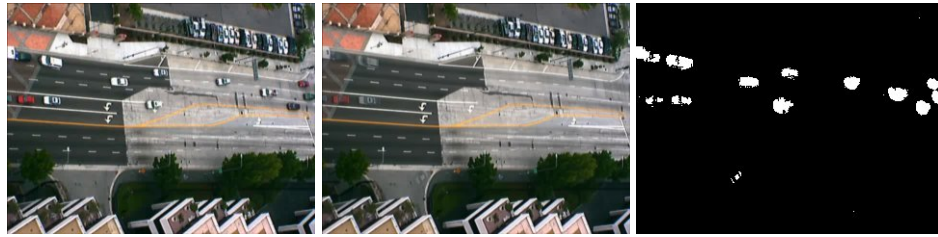


Figura 2.2: (a) Cuadro de una secuencia de tránsito (b) Imagen de fondo (c) Segmentación resultante de aplicar sustracción de fondo

sólo parcialmente detectado en la imagen segmentada).

El ejemplo anterior ilustra la importancia de escoger un método adecuado para obtener el modelo o imagen de fondo. A continuación se describen algunos de los enfoques más empleados para modelar el fondo de una secuencia.

#### 2.1.2.1. Promediado

En su forma más simple, este enfoque consiste en calcular la suma de un gran número de cuadros y promediarlos (ecuación 2.3) para obtener el modelo de fondo. En algunos casos se utiliza un factor de aprendizaje para establecer la relación entre cada nuevo cuadro y el fondo establecido.

$$B_t = \frac{\sum_{i=0}^n I_n(x, y)}{n} \quad (2.3)$$

El costo computacional es bajo, pero el principal inconveniente del algoritmo es que produce sombras o colas en la imagen segmentada debido al movimiento aparente de los vehículos. En [17] se reduce dicho problema actualizando el modelo de fondo mediante los fondos instantáneos (cuadros de donde se han removido los componentes móviles). Para un fondo instantáneo  $IB_t$  se tiene que:

$$B_t(x, y) = (1 - \alpha)B_{t-1}(x, y) + \alpha IB_t(x, y) \quad (2.4)$$

donde  $B_t(x, y)$  es el modelo de fondo actualizado y  $\alpha$  es el factor de aprendizaje.

#### 2.1.2.2. Gaussiano Unico

Este enfoque consiste en una mejora del promediado simple, en la cual además de calcular una matriz que contenga el promedio de cada píxel, se obtiene también la covarianza para cada uno de los píxeles en cuestión. Estas dos matrices (promedio y covarianza) pasan a formar el

modelo de fondo. Cada nuevo píxel es clasificado de acuerdo a su posición en una distribución gaussiana, tomando la covarianza calculada como umbral. De forma similar a la actualización del promedio del fondo mostrada en la ecuación 2.4, los autores de [20] actualizan la matriz de covarianza para cada nuevo cuadro mediante

$$\Sigma_t(x, y) = (1 - \alpha)\Sigma_{t-1}(x, y) + [\alpha(I_t(x, y) - B_t(x, y)) \times (I_t(x, y) - B_t(x, y))^T] \quad (2.5)$$

donde  $\Sigma_t(x, y)$  es la matriz de covarianza actualizada,  $B_t$  es el modelo de fondo y  $\alpha$  el factor de aprendizaje.

### 2.1.2.3. Filtro Kalman

Este método modela la distribución de intensidad de cada píxel mediante un filtro Kalman [21], el cual permite estimar el estado de un proceso que sea modelable a través de un sistema lineal (ecuación 2.1.2.3).

$$\begin{aligned} x_t &= Ax_{t-1} + Bu_{t-1} + w_{t-1} \\ z_t &= Cx_t + v_t \end{aligned} \quad (2.6)$$

Para un determinado píxel, el estado del sistema o proceso a estimar  $x_t$  es el valor del fondo en la posición de dicho píxel, el valor medido  $z_t$  es su valor medido en un instante determinado, la entrada  $u_t$  del sistema es considerada cero, y se considera que la distribución temporal para cada valor de la imagen de fondo es constante.

Como requisito para el uso del filtro Kalman, se asume que ambos parámetros de ruido ( $w_t$  y  $v_t$  son los ruidos de proceso y medición, respectivamente) son ruidos blancos con distribuciones gaussianas de moda cero. Sin embargo, los cambios drásticos en la iluminación violan este supuesto (dado que componen ruido no gaussiano), lo cual provoca errores en el resultado del filtro. En [22] se implementa un método de segmentación basado en este filtro que lidia adecuadamente con cambios de iluminación mediante la estimación global de dichos cambios y del ruido de medición.

### 2.1.2.4. Estimación Modal

Este método obtiene el histograma temporal para cada píxel (en función de una ventana de tiempo constante) y extrae su moda o valor más recurrente, formando en base a ello la imagen de fondo [36]. Este enfoque presupone que el fondo tiene una presencia dominante durante la



Figura 2.3: (a) Cuadro de una secuencia de tránsito (b) Segmentación resultante de aplicar mixtura de gaussianos

ventana de análisis para poder ser estimado correctamente. En el caso de videos de tránsito vehicular, dicho supuesto falla en el caso de que haya vehículos que se encuentren estacionados por un periodo de tiempo significativo.

### 2.1.3. Mixtura de Gaussianos

La mixtura de gaussianos [23] modela el proceso histórico  $\{X_1, \dots, X_t\}$  de cada pixel  $X_t$  como una superposición o mixtura de distribuciones gaussianas de la forma

$$P(X_t) = \sum_{k=1}^N \omega_{k,t} \mathcal{N}(X_t, \mu_{k,t}, \Sigma_{k,t}) \quad (2.7)$$

donde  $N$  es el número de gaussianos a considerar (usualmente de 3 a 5), y  $\omega_{k,t}$  es el peso del gaussiano  $k$ -ésimo en el instante  $t$ .  $\mathcal{N}(X_t, \mu_{i,t}, \Sigma_{i,t})$  es la función que representa la densidad de probabilidad gaussiana dada por:

$$\mathcal{N}(X_t, \mu, \Sigma) = \frac{1}{|\Sigma|^{1/2} 2\pi^{n/2}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \quad (2.8)$$

donde  $\mu$  y  $\Sigma$  representan la media y la covarianza de la distribución, respectivamente.

Dada la naturaleza estadística de este enfoque, su consumo de recursos computacionales es mayor a los descritos anteriormente. Puede lidiar adecuadamente con cambios de iluminación y entornos saturados de objetos, pero falla en situaciones con alta congestión de movimiento (como es el caso de un video de tránsito).

### 2.1.4. Flujo Óptico

El flujo óptico es un enfoque a partir del cual se busca estimar el campo vectorial de velocidad de una secuencia de imágenes para realizar el análisis del movimiento. Existe una

variedad de métodos para extraer dichos campos vectoriales, pero todos se basan en tres supuestos principales:

- Intensidad constante: Los objetos se desplazan por la escena, pero su valor (intensidad medida) no varía.
- Coherencia espacial: Puntos vecinos tendrán características de movimiento similares.
- Persistencia temporal: El movimiento de las regiones es gradual.

En [24] se ofrece una evaluación cualitativa de diversos métodos basados en flujo óptico, sin embargo se indica que el error relativo es de aproximadamente 7% en el mejor de los escenarios. Ello se debe a que en general, los supuestos descritos anteriormente son difíciles de satisfacer en la mayoría de casos prácticos.

## 2.2. Principal Component Pursuit

Principal Component Pursuit o PCP es un problema de optimización matemática, el cual fue originalmente formulado como una variación robusta del Análisis del Componente Principal o PCA [3]. Se busca recuperar una matriz de bajo rango  $L$ , a partir de una matriz observada de gran tamaño  $M = L + S$ , donde  $S$  es una matriz dispersa (sparse, o compuesta predominantemente por ceros) o “ruidosa”. La formulación original del problema de minimización se plantea como:

$$\arg \min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad s.t. \quad M = L + S \quad (2.9)$$

Donde  $\|L\|_*$  es la norma nuclear de  $L$  y  $\|S\|_1$  es la norma  $l_1$  de la matriz  $S$  interpretada como un vector columna.

Dependiendo del tipo de aplicación, se puede tener particular interés en el componente de bajo rango o el componente sparse. Particularmente, en el caso de la segmentación de movimiento, si cada columna de  $M$  corresponde a un cuadro de video, entonces necesariamente el componente de bajo rango  $L$  contendrá la información del fondo estático y el componente sparse  $S$  capturará los objetos en movimiento (Figura 2.4).

En su formulación original, PCP presenta dos inconvenientes principales en relación a la segmentación de movimiento: requisitos de memoria y de capacidad de procesamiento. El primero se debe a que se requiere ingresar una gran cantidad de cuadros de video antes de poder obtener una solución ( $M$  debe ser una matriz rectangular de gran tamaño), lo cual implica un alto consumo de memoria. El segundo se debe a que el cálculo de la optimización

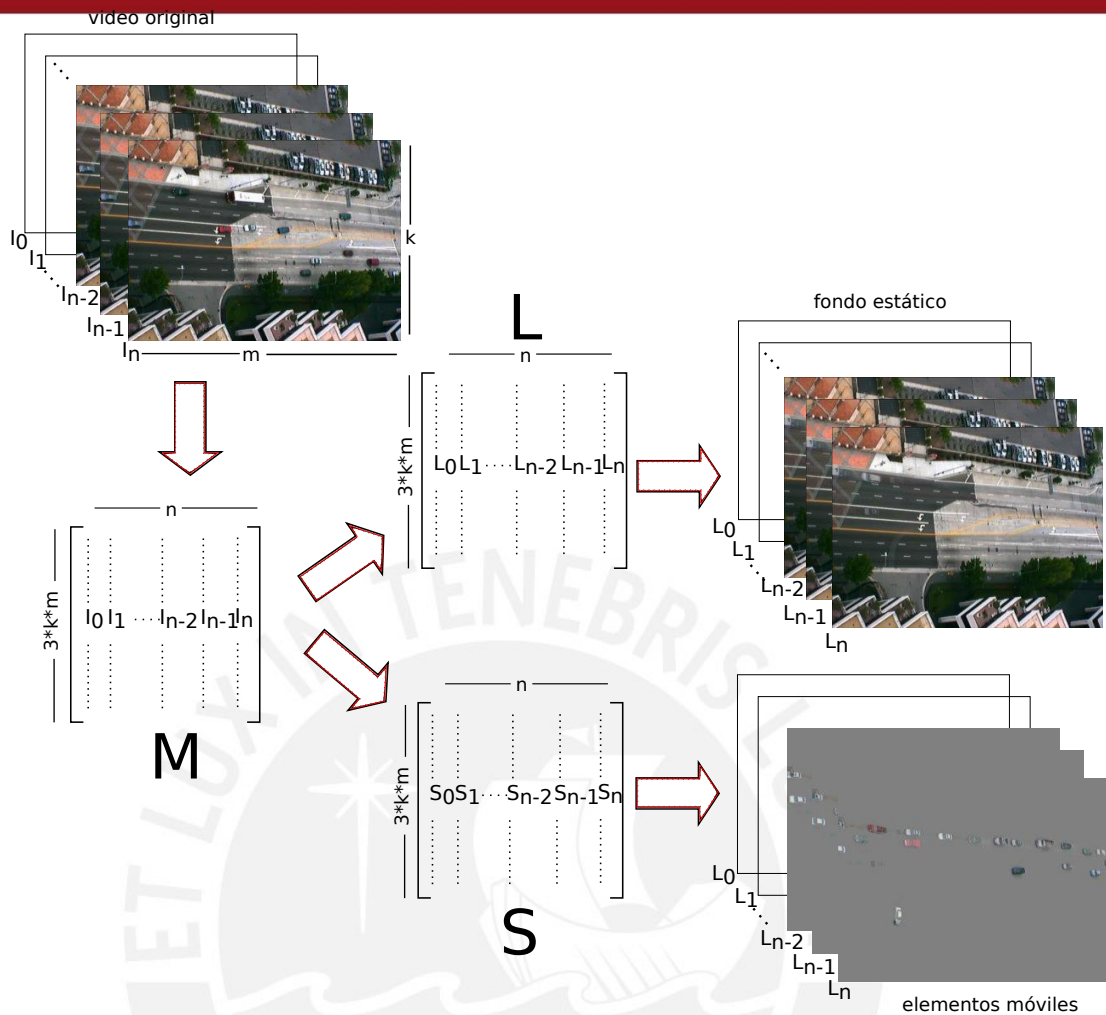


Figura 2.4: Composición de la matriz M y subsecuente descomposición en L y S

tiene un alto costo computacional, lo cual dificulta el uso del método en aplicaciones que requieran un bajo tiempo de procesamiento (tiempo real), como la video vigilancia.

Debido a ello, existe una variedad de algoritmos [25, 4] que buscan resolver PCP mediante una reformulación del problema original en diversos sub-problemas y restricciones, con el fin de mejorar su rendimiento computacional y su consumo de memoria, manteniendo la eficiencia del método. Una revisión y comparación cualitativa de estos algoritmos puede encontrarse en [4].

Recientemente, en [26] se propuso un algoritmo para resolver PCP de forma incremental, lo cual permite procesar cada cuadro de video a medida que va ingresando, en lugar de almacenar una gran cantidad de cuadros y procesarlos por bloques (enfoque típicamente usado que dificulta el procesamiento en tiempo real). Este enfoque trae como consecuencia natural una reducción significativa tanto en el consumo de memoria como en el tiempo de procesamiento por cuadro.



## Capítulo 3

# Diseño del Algoritmo Propuesto

### 3.1. Base de datos utilizada para el diseño

Para el diseño del algoritmo se empleó la base de datos Lankershim Boulevard [27], disponible libremente en línea, con videos a color de resolución 640x480 pixeles.

Como ha sido mencionado anteriormente, los videos a los cuales está orientado este algoritmo son aquellos generados por cámaras de monitoreo estáticas (o en su defecto, que hayan pasado por alguna etapa de estabilización de imagen). También cabe resaltar que el diseño del método no toma en consideración la oclusión entre vehículos, por lo que la adquisición de los videos debe realizarse desde un punto lo suficientemente elevado como para que dicha oclusión sea mínima o nula. Por último, la variación perceptual del tamaño de los vehículos desplazándose por la escena debe ser despreciable, lo cual implica que la adquisición de los videos debe ser lo más “perpendicular” a la pista posible.

### 3.2. Algoritmo Propuesto

En este capítulo se describen los bloques que componen el algoritmo propuesto. Éste se divide principalmente en dos etapas: entrenamiento y conteo. La primera generará los parámetros necesarios para el procesamiento del conteo. La segunda utilizará los datos proporcionados por la etapa anterior para efectuar el conteo continuo de los vehículos. El algoritmo 1 describe de forma general la lógica propuesta para el sistema.

**Algorithm 1:** Algoritmo de conteo

---

**Input:**  $Video = \{I_1, I_2, \dots, I_L\}$ : Secuencia de video compuesta por L imágenes  
**Input:**  $Traininglength$ : Número de imágenes necesarias para la etapa de entrenamiento  
**Result:**  $N = \{N_1, N_2, \dots, N_M\}$ : vector que contiene el número total de vehículos en escena para cada cuadro procesado

ALGORITMO DE CONTEO  
 InicializarPCP()  
**for**  $t \leftarrow 1$  **to**  $L$  **do**  
    $B_t = \text{Segmentación de Movimiento}(I_t)$   
    $D_t = \text{Extracción de Características}(B_t)$   
   **if**  $t \leq Traininglength$  **then**  
     **if**  $t == Traininglength$  **then**  
        $[Area_{min}, Area_{max}] = \text{Estimación de Parámetros}(D)$   
     **else**  
        $N = \text{Procesamiento Espacial y Temporal}(B_t, B_{t-1}, D, Area_{min}, Area_{max})$

---

### 3.3. Etapa de Entrenamiento

#### 3.3.1. Segmentación del movimiento

##### 3.3.1.1. Extracción de movimiento mediante PCP

La extracción de movimiento se realiza mediante el algoritmo PCP incremental mencionado en la sección 2.2 [26] (cabe resaltar que se utiliza la implementación de Matlab disponible en la página web del autor [28]). Tras una etapa de inicialización, el algoritmo permite obtener una imagen sparse que contiene sólo los elementos móviles para cada cuadro de video.

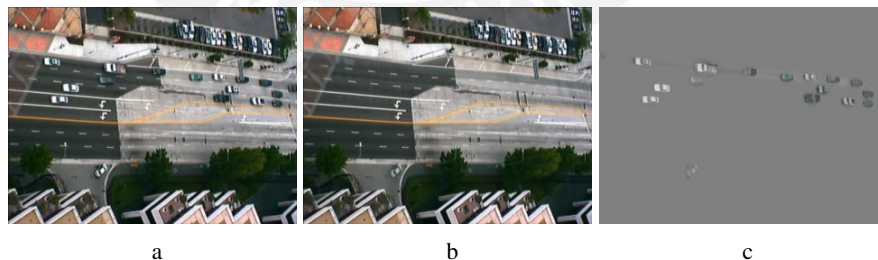


Figura 3.1: (a) Imagen original (b) Imagen de fondo estático L (c) Imagen sparse

Dada la naturaleza incremental del algoritmo, éste actualiza el modelo de fondo para cada nuevo cuadro que procesa. Ello implica que si un objeto en movimiento (en este caso, un vehículo) se detiene por un determinado período de tiempo, será gradualmente integrado en el fondo, y si empieza a moverse de nuevo, será gradualmente retirado de éste (situación que se da, por ejemplo, en zonas con semáforos). Durante estas etapas de 'transición', se generarán

colas tenues en el movimiento del objeto. Dicho inconveniente es atenuado aplicando un filtro promediador simple para limpiar la imagen sparse antes de pasar a la siguiente etapa.

### 3.3.1.2. Segmentación Unimodal

Una vez extraído el componente sparse de la imagen, el siguiente paso es obtener una máscara o imagen binaria para simplificar el procesamiento de ésta. Para ello es necesario encontrar un valor umbral a partir del cual segmentar la imagen. Dada la naturaleza de una imagen sparse, ésta presenta un valor mayormente dominante (moda), y unos pocos valores distintos. La imagen 3.2 ejemplifica esta distribución de valores.

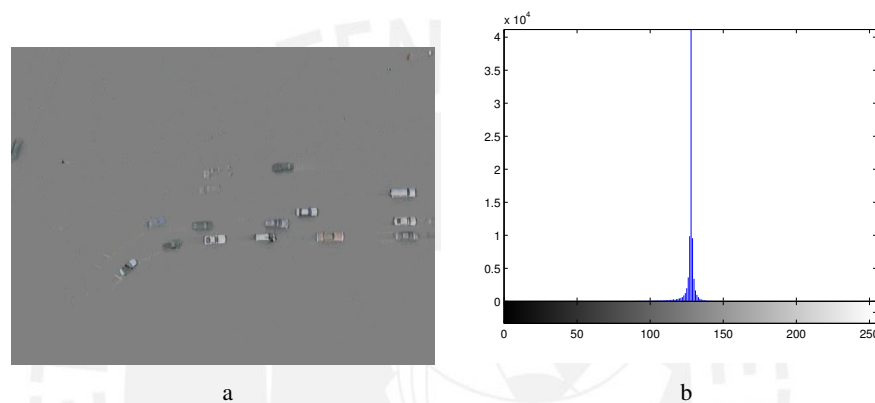


Figura 3.2: (a) Imagen sparse (b) Histograma de la imagen

En la distribución se observa que los píxeles que contienen movimiento son tan escasos con respecto a los de fondo que no generan un pico visible en el histograma, lo cual dificulta la determinación directa de un valor umbral para separar ambos grupos. En [29] se propone un método de segmentación para este tipo de distribución, denominado *segmentación unimodal*.

La segmentación unimodal definida para una distribución con moda única en cero, consiste en el trazado de una línea imaginaria entre el máximo y el mínimo valor del histograma (figura 3.3), y la búsqueda del punto del histograma que maximiza la distancia entre éste y dicha recta imaginaria. El umbral está determinado por el valor de intensidad que corresponde al punto que maximiza dicha distancia.

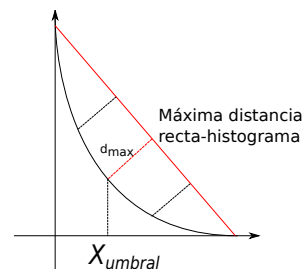


Figura 3.3: Segmentación Unimodal en una distribución con moda en cero

Esta definición funciona para distribuciones con moda en cero, sin embargo cabe resaltar que cualquier distribución unimodal puede cumplir con esta característica mediante una transformación simple.

### 3.3.2. Extracción de características

Una vez obtenida la imagen binaria, se realiza un análisis de componentes conectados para obtener ciertos parámetros de los objetos contenidos en la imagen. Este análisis consiste en agrupar los píxeles conectados entre sí mediante un barrido de la imagen, asignando una misma etiqueta a los píxeles vecinos que compartan el mismo valor.

Para cada grupo u objeto binario detectado se extraen tres características principales de interés: área, centroide y dimensiones cartesianas. El *área*  $A$  del objeto corresponde al número de píxeles que lo componen. El *centroide*  $C$  corresponde al promedio de todas las posiciones de los píxeles que integran el objeto (concepto correspondiente al centro de masa). Finalmente, las *dimensiones cartesianas*  $D$  del objeto equivalen a la máxima extensión horizontal y vertical de un objeto determinado. Una descripción gráfica de dichas características se muestra en la Figura 3.4.

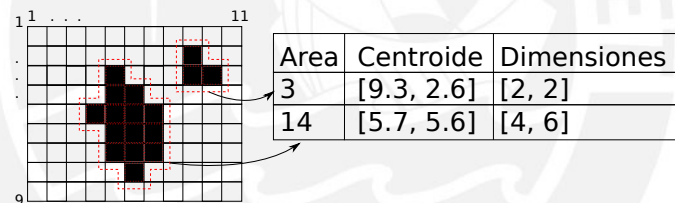


Figura 3.4: Extracción de características de interés

### 3.3.3. Estimación de parámetros para el procesamiento

Luego de haber almacenado los valores de las características correspondientes a la etapa anterior por un cierto número de cuadros (para el caso de la implementación realizada, 200), se calculan los parámetros necesarios para la etapa de procesamiento.

Dado que durante el procesamiento espacial se realiza una relajación del problema (esto se verá más adelante) mediante la cual solo se considera la existencia de vehículos de uso doméstico estándar (es decir, vehículos de tamaño mediano como autos y camionetas, descartando la existencia de buses y camiones), se necesitan cotas para limitar el tamaño en píxeles que pueden ocupar dichos vehículos en la escena.

Para obtener dichas cotas se calcula el histograma acumulado de las áreas extraídas en la etapa anterior. En un video de tránsito urbano común, dicho histograma debería presentar una

distribución semejante a la mostrada en la Figura 3.5, con tres picos distintivos correspondientes a peatones, vehículos pequeños, y vehículos grandes.

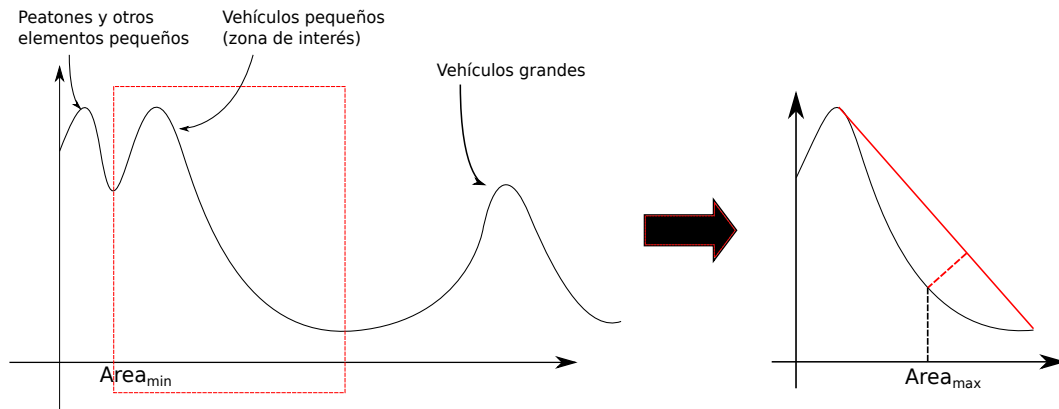


Figura 3.5: Segmentación Unimodal en una distribución con moda en cero

De acuerdo a la relajación del problema mencionada líneas arriba, solo se toma en cuenta la existencia de vehículos de uso doméstico. Dicho supuesto se representa en la Figura 3.5 en líneas rojas punteadas, es decir, se enfoca solo el conjunto de áreas correspondiente a los vehículos pequeños. La extracción de este conjunto se logra mediante una segmentación multimodal [30], la cual proporciona los límites del “pico” de interés.

### 3.4. Etapa de Conteo

De forma análoga a la etapa de entrenamiento, la etapa de conteo segmenta el movimiento mediante PCP, binariza la imagen de movimiento mediante una segmentación unimodal, y extrae las características de interés para cada cuadro. Es decir, los primeros bloques son exactamente iguales a sus contrapartes en la etapa de entrenamiento.

#### 3.4.1. Procesamiento espacial y temporal

Una vez obtenidos los parámetros de interés para un cuadro determinado, se realiza el procesamiento o conteo de los autos contenidos en el mismo. Este conteo se aplica mediante dos enfoques consecutivos: espacial y temporal. El primero toma en cuenta las características extraídas del cuadro en cuestión para generar un estimado inicial del número de vehículos en la escena. El segundo analiza las características correspondientes a cuadros previos para corregir o afinar la estimación inicial. El Algoritmo 2 describe de forma general la secuencia de procesos de esta etapa.

**Algorithm 2:** Procesamiento Espacial y Temporal

---

**Input:**  $f_t, f_{t-1}$ : imágenes analizadas en los instantes  $t$  y  $t - 1$   
**Input:**  $Data = \{A, C, D\}$ : vectores de área, centroide y dimensiones cartesianas  
**Input:**  $Area_{min}, Area_{max}$ : cotas de área provenientes de la etapa de entrenamiento  
**Result:**  $Nvec$ : vector que indica número de vehículos por objeto  
**Result:**  $T$ : número total de vehículos en escena

PROCESAMIENTO ESPACIAL  
 Inicializar  $Nvec = \text{zeros}(\text{length}(A))$   
 $Nvec = \text{ProcesamientoEspacial}(A, Area_{min}, Area_{max})$

PROCESAMIENTO TEMPORAL  
**for**  $i \leftarrow 1$  **to**  $\text{length}(Nvec)$  **do**  
   **if**  $Nvec(i) > 1$  **then**  
      $p = \text{AsociarObjetoAnterior}(Data_t, Data_{t-1})$   
      $M = \text{CriterioOrientaciondelMovimiento}(Data(i)_t, Data(p)_{t-1})$   
     **if**  $(M == \text{verdadero})$  **then**  
        $R = \text{CriterioRigidezdelObjeto}(Data(i)_t, Data(p)_{t-1}, f_t, f_{t-1})$   
       **if**  $(R == \text{verdadero})$  **then**  
          $Nvec(i) = 1$   
       **else**  
          $Nvec(i) = 2$   
     **end if**  
   **end if**  
 $T = \text{sum}(Nvec)$

---

**3.4.1.1. Procesamiento espacial**

Como su nombre lo dice, este enfoque emplea las características espaciales del cuadro de interés para estimar un conteo inicial del número de vehículos en escena. En particular, la característica de interés en este enfoque es el área de los objetos binarios detectados en la imagen. Cabe resaltar que un objeto binario determinado puede contener más de un objeto real: dos autos desplazándose muy próximos entre sí o un grupo denso de peatones pueden componer un solo objeto.

Es por ello que para simplificar el cálculo del estimado inicial se realiza una relajación del problema, en la cual solo se toma en cuenta la existencia de vehículos pequeños como automóviles y camionetas, excluyendo camiones y buses. Esta relajación conlleva a un supuesto particular: *que el número de autos en cada objeto es proporcional al área de dicho objeto*. Dicha conclusión implica que un vehículo grande como un camión será interpretado por esta lógica como un conjunto de autos más pequeños.

Para determinar si un objeto corresponde a uno o mas vehículos, se utilizarán los parámetros  $Area_{min}$  y  $Area_{max}$ , obtenidos en la etapa de entrenamiento. Si el área de un determinado objeto es inferior a  $Area_{min}$ , este será descartado. De lo contrario, el número de vehículos

contenidos en el objeto está dado por:

$$N = \frac{A_{objeto}}{Area_{max}} \quad (3.1)$$

donde el valor de  $N$  se aproxima al entero superior mas cercano. Se realiza el cálculo de  $N$  para todos los objetos detectados en la imagen antes de pasar a la siguiente etapa.

Si las cotas de área han sido adecuadamente inicializadas, este criterio detectará adecuadamente los objetos que contengan uno o varios vehículos pequeños (Figuras 3.6(a) y 3.6(b)), y fallará para aquellos que contengan vehículos grandes (Figura 3.6(c)), asignándoles un número mayor al correspondiente.

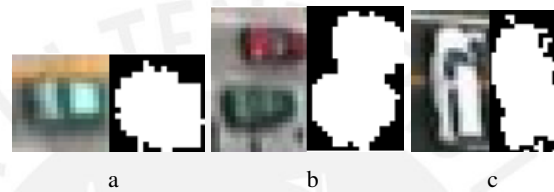


Figura 3.6: Procesamiento espacial para :(a) Objeto que contiene un vehículo pequeño ( $N = 1$ ) (b) Objeto que contiene dos vehículos pequeños ( $N = 2$ ) y (c) Objeto que contiene un vehículo grande( $N = 2$ )

#### 3.4.1.2. Procesamiento temporal

Una vez obtenido un valor inicial para el conteo, se pasa a analizar la información obtenida de cuadros anteriores para corregir los errores que hayan podido presentarse en el conteo preliminar.

Como se mencionó anteriormente, el procesamiento espacial tiende a fallar en el conteo relacionado a objetos grandes: objetos con un valor de  $N > 1$  pueden contener un número menor de vehículos al que les ha sido asignado (caso de la Figura 3.3(c)). Éstos son los objetos en los que se enfoca el procesamiento temporal.

Para realizar la corrección del valor  $N$  asignado a un determinado objeto, el primer paso es asociar dicho objeto con su contraparte en un cuadro anterior en el tiempo. Para ello es necesario se cumplan dos condiciones: la variación de área de un cuadro a otro no debe superar el 20 % (ecuación 3.2) y el desplazamiento entre cuadros debe ser menor a un determinado umbral (ecuación 3.3).

$$|A_{objetoactual} - A_{objetoprevio}| \leq 0,2 * A_{objetoactual} \quad (3.2)$$

$$|C_{objetoactual} - C_{objetoprevio}| \leq Desplazamiento_{minimo} \quad (3.3)$$



Figura 3.7: Asociación de un objeto con su contraparte en un cuadro anterior

La figura 3.7 ilustra la idea de esta asociación. Una vez verificada la correspondencia temporal entre dos objetos, se analizan dos factores para verificar que el número de vehículos  $N$  asignado sea correcto: la *orientación del movimiento* y la *rigidez*.

El análisis de la *orientación del movimiento* se basa en un supuesto simple: que el desplazamiento de un vehículo, visto desde arriba, siempre se da en la dirección de su eje más grande. Es decir, si el eje donde el desplazamiento  $\vec{Desp} = C_{objetoactual} - C_{objetoprevio}$  es máximo coincide con el eje en el que  $D$  es máximo para un determinado objeto, su movimiento es coherente con el de un único vehículo (cabe recordar que esta corrección se aplica a los objetos con  $N > 1$ ).

Este factor se evalúa debido a que una situación común que se presenta en los videos de tránsito es que vehículos que se desplazan en carriles contiguos sean interpretados como un solo objeto binario grande (Figura 3.6 (b)), además de los vehículos de mayor tamaño que por sí mismos forman objetos grandes (Figura 3.6 (c)). Los objetos que presenten una orientación de movimiento coherente pasarán al análisis de rigidez, mientras que aquellos que no lo hagan mantendrán el valor de  $N$  que se les asignó inicialmente.

El análisis de *rigidez* consiste básicamente en comprobar que la forma de un determinado objeto no haya cambiado en el tiempo. Para ello, se crea un recuadro envolvente o delimitador que encierra a cada uno de los objetos binarios (actual y previo)  $R_1$  y  $R_2$ . El objeto se considera rígido si se cumple la condición:

$$Area(Objeto_{actual} \& Objeto_{previo}) \geq 0,75 * Area(Objeto_{actual}) \quad (3.4)$$

Es decir, se evalúa la conjunción lógica punto a punto entre ambos recuadros y se calcula el área del resultado, la cual no debe diferir en más de un 25 % del área del objeto original para un objeto rígido. Si se determina que un objeto es rígido, se le asigna un  $N = 1$ , de lo contrario se le atribuye el valor  $N = 2$  (independientemente del valor que tuviese anteriormente).

Una vez realizada la corrección para todos los objetos en una imagen, el número total de vehículos en la escena se obtiene sumando los  $N$ 's de cada uno de dichos objetos.



## Capítulo 4

# Implementación y resultados computacionales

### 4.1. Consideraciones de implementación

#### 4.1.1. Software de implementación

Se utilizó el entorno MATLAB<sup>®</sup> desarrollado por MathWorks en su versión 2013a.

#### 4.1.2. Base de datos generada para las pruebas del algoritmo

Para la etapa de pruebas se adquirieron tres videos adicionales a la base de datos utilizada para el diseño, de los cuales uno corresponde a la Avenida Universitaria, y dos a una avenida contigua al domicilio del autor. De este punto en adelante se referirá a estos como “Pabellón A”, “San Martín 1” y “San Martín 2”. Todos estos videos tienen una resolución de 640x480 pixeles.

Una secuencia de imágenes correspondiente a cada uno de los videos procesados se muestra en la Figura 4.2.

#### 4.1.3. Características de las imágenes para la implementación

Dado que los videos de la base de datos de prueba fueron adquiridos de forma manual, no cumplen con algunas de las condiciones de diseño especificadas en la sección 3.1. Por ejemplo, “Pabellón A” fue adquirido a una altura adecuada pero a una distancia significativa de la Avenida Universitaria (vista no perpendicular), lo cual trae como consecuencia que si bien los objetos tienen poca o nula variación perceptual de tamaño, se produce oclusión entre vehículos de un carril y otro, y un muro oculta parcialmente uno de los carriles.

Debido a ello, se establecieron dos marcos de evaluación del algoritmo: un conteo absoluto instantáneo y un conteo acumulativo mediante el uso de una puerta virtual. El primero calcula el número instantáneo de vehículos para un número determinado de cuadros, y se evaluará su rendimiento en dos videos de la base de datos Lankershim. El segundo calcula el número de vehículos que atraviesan una “puerta virtual” colocada en una zona específica de la autopista, y se evaluará su rendimiento en ambas bases de datos (diseño y pruebas). Cabe señalar que el uso de puertas virtuales es uno de los marcos de evaluación de métodos de conteo vehicular más recurrentes en la literatura científica [10, 11].

## 4.2. Descripción de la implementación

Dado que la implementación incremental de PCP disponible en [28] procesa internamente un cuadro de video a la vez, se juzgó conveniente implementar el algoritmo de conteo dentro del bucle principal del algoritmo PCP, para procesar los cuadros sparse a medida que éstos fueran generados. Tras una etapa de inicialización de parámetros inherente al algoritmo PCP, se ejecuta la etapa de entrenamiento para los primeros 200 cuadros o iteraciones, al final de la cual se calculan los parámetros para el procesamiento y se da paso a la etapa de conteo. Ésta se ejecuta de manera continua hasta que el último cuadro o iteración ha sido procesado.

La función de procesamiento espacial recibe como entrada el vector de área de la imagen actual y las cotas de área, y entrega como resultado un vector que indica el número de vehículos de cada objeto, donde los índices para el vector de salida y el vector de área corresponden al mismo objeto.

La función de procesamiento temporal recibe como entrada los vectores de características (área, centroide y dimensiones), el vector de número de vehículos por objeto (salida del procesamiento espacial) y los factores de tolerancia de area y desplazamiento (correspondientes a las ecuaciones 3.2 y 3.3), para entregar como salida el vector corregido de número de vehículos por objeto. El número total de vehículos en escena se calcula sumando los elementos de dicho vector.

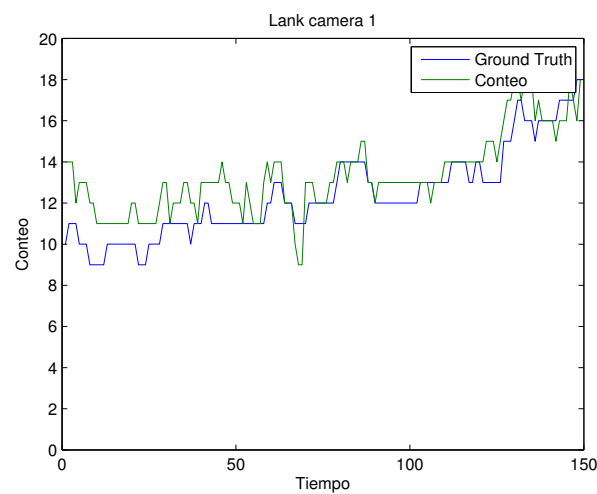
Teniendo como datos el número de vehículos por objeto y la ubicación correspondiente a cada objeto (vector de centroides), el cálculo de los vehículos que atraviesan la puerta virtual consiste simplemente en analizar los objetos que se encuentran cerca de ésta en un instante determinado, y verificar si dichos objetos se encontraban al otro lado de la puerta en un cuadro anterior. De ser así, se considera que el objeto atravesó la puerta y se incrementa el valor del contador por el número  $N$  asignado al objeto.

### 4.3. Resultados obtenidos

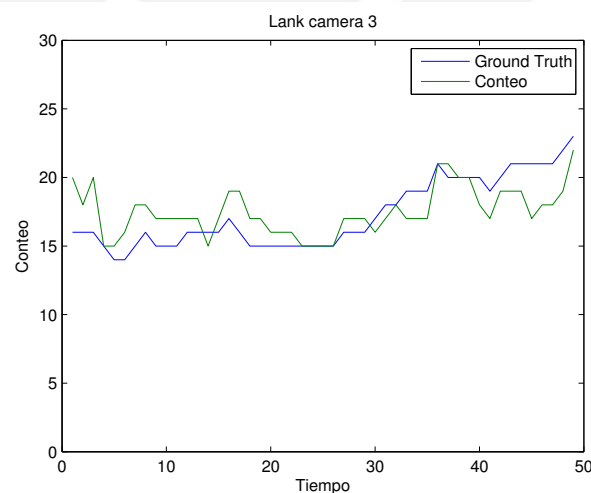
#### 4.3.1. Resultados del conteo instantáneo

La Figura 4.1 muestra los resultados del conteo instantáneo para los videos de la base de datos Lankershim en una gráfica conteo vs. tiempo. En dicha gráfica se observa el conteo generado por inspección para cada instante de tiempo (línea azul) y el conteo generado por el algoritmo propuesto (línea verde).

Una descripción estadística de dichos resultados se puede observar en la Tabla 4.1, donde se muestra el valor real del conteo, el valor generado por el algoritmo, el número de falsos positivos FP, falsos negativos FN (todos son valores acumulativos), precisión (calculada como  $(FP + FN)/ConteoReal$ ) y error máximo por cuadro.



(a)



(b)

Figura 4.1: Rendimiento de conteo instantáneo en base de datos Lankershim para (a) la cámara 1 y (b) la cámara 3

Videos	Conteo real	Conteo generado	FP (falsos positivos)	FN (falsos negativos)	Precisión	Error máximo
Lankershim 1	1846	1984	151	13	91.12 %	4
Lankershim 3	847	858	43	32	91.15 %	4

Tabla 4.1: Rendimiento de conteo instantáneo en base de datos lankershim.

### 4.3.2. Resultados de la puerta virtual

La Tabla 4.2 muestra los resultados obtenidos para el conteo de pases por puerta virtual para ambas bases de datos. Se muestra el número de pases reales (por inspección) y contabilizados (por el algoritmo), la eficiencia para cada video y la localización de la puerta virtual (los primeros cuatro videos utilizan puertas verticales y el quinto, una puerta horizontal).

Para fines evaluativos, se muestran también los resultados de eficiencia para otros trabajos que también utilizan una puerta virtual como métrica de rendimiento [10, 11, 31, 32].

Videos	Pases real	Pases contabilizados	Eficiencia	Puerta
Lankershim 1	27	27	100 %	X=400, Y=[140 230]
Lankershim 3	54	52	96.3 %	X=400, Y=[140 250]
Pabellón A	9	9	100 %	X=360, Y=[190 230]
San Martín 1	14	13	92.86 %	X=570, Y=[100 190]
San Martín 2	20	22	90.91 %	Y=240, X=[55 185]
<i>Método [11]</i>	–	–	87.1 %(*)	–
<i>Método [10]</i>	–	–	86.5 %(*)	–
<i>Método [31]</i>	–	–	95.1 %(*)	–
<i>Método [32]</i>	–	–	91.9 %(*)	–

Tabla 4.2: Rendimiento de conteo de pases por puerta virtual en ambas bases de datos. Los valores marcados con (\*) corresponden a la eficiencia promedio de las pruebas reportadas en [10, 11, 31, 32], y son mostrados para fines comparativos

### 4.3.3. Análisis de los resultados

Se han obtenido resultados alentadores para ambos marcos de evaluación. En la Figura 4.1 se observa que el conteo instantáneo “sigue” al conteo real con un error máximo de 4 vehículos por cuadro. El error generado en dicho conteo se debe principalmente al efecto de integración y separación del fondo mencionado en la sección 3.3.1.1. Cabe señalar que en promedio el error para este conteo es de 1 vehículo por cuadro, lo cual se encuentra dentro de un margen tolerable.

Respecto a la puerta virtual, el rendimiento más bajo obtenido es de 90 %, llegando a 100 % en los mejores escenarios, con un rendimiento promedio de 96.01 %. Dichos resultados

permiten al sistema propuesto competir con otros algoritmos actuales de conteo vehicular, como se observa al comparar la eficiencia obtenida con los resultados reportados por [10, 11, 31, 32].

La Tabla 4.2 muestra además los tiempos de procesamiento obtenidos para diversas resoluciones de videos, y la cantidad de cuadros procesables por segundo a dicha velocidad (FPS).

<b>Resolución de video</b>	320x240	640x480	1980x1080
<b>Tiempo de procesamiento promedio por cuadro (s)</b>	0.064	0.21	1.5
<b>FPS</b>	15.6	4.8	0.7

Tabla 4.3: Tiempos de procesamiento promedio por cuadro y FPS para diversos tamaños de video





Figura 4.2: Secuencias correspondientes a (a)..(c) Lankershim 1, (d)..(f) Lankershim 3, (g)..(i) Pabellón A, (j)..(l) San Martín 1, y (m)..(o) San Martín 2. Se señala con un círculo rojo aquellos objetos que el algoritmo interpreta como 1 vehículo, con un círculo azul aquellos que son interpretados como 2 vehículos, y con uno verde aquellos que son interpretados como 3 vehículos.

## Conclusiones

- Los resultados finales obtenidos muestran que el algoritmo propuesto proporciona un rendimiento comparable a otros métodos actuales para videos que cumplen con las condiciones de diseño propuestas. Cabe señalar que la maximización de la eficiencia obtenida para la puerta virtual depende en gran medida del criterio que se utilice para posicionar dicha puerta.
- Si bien los resultados han sido obtenidos para un número limitado de cuadros (debido a la dificultad inherente para generar un “ground truth” manual, particularmente para el caso del conteo instantáneo) y una resolución de video determinada, el algoritmo es escalable a videos de cualquier resolución y duración.

## Recomendaciones y observaciones

- Si bien el tiempo de procesamiento obtenido para videos en formato VGA o Full HD no se presta para una implementación en tiempo real, éste puede reducirse realizando una implementación del algoritmo en algún lenguaje de bajo nivel como C o C++, o implementando en algún lenguaje de programación en paralelo los módulos independientes que lo permitan.
- Una forma de incrementar la autonomía del algoritmo sería desarrollar un método que posicione la puerta virtual de forma automática en el lugar más adecuado, de modo que no se dependa de un usuario para este proceso.
- Sería importante realizar el análisis de la naturaleza de imágenes de tránsito segmentadas mediante PCP en condiciones nocturnas (donde la detección directa de los vehículos resulta menos factible), y evaluar la posibilidad de adaptar el algoritmo propuesto para funcionar en dichas condiciones.



## Bibliografía

- [1] B. Sultan, R. Meekums, and M. Brown, “The impact of active traffic management on motorway operation,” in *Road Transport Information and Control - RTIC 2008 and ITS United Kingdom Members’ Conference, IET*, pp. 1–8, May 2008.
- [2] H. Zheng, E. Nava, and Y.-C. Chiu, “Evaluating active traffic and demand management strategies for congested tourism traffic corridor,” in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 2015–2020, Oct 2011.
- [3] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?,” *J. ACM*, vol. 58, pp. 11:1–11:37, June 2011.
- [4] T. Bouwmans and E. H. Zahzah, “Robust {PCA} via principal component pursuit: A review for a comparative evaluation in video surveillance,” *Computer Vision and Image Understanding*, vol. 122, no. 0, pp. 22 – 34, 2014.
- [5] A. Csikos, I. Varga, and K. Hangos, “Freeway shockwave control using ramp metering and variable speed limits,” in *Control Automation (MED), 2013 21st Mediterranean Conference on*, pp. 1569–1574, June 2013.
- [6] L. Qi, M. Zhou, and Z. Ding, “Real-time traffic camera-light control systems for intersections subject to accidents: A petri net approach,” in *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pp. 1069–1074, Oct 2013.
- [7] M. Aron, S. Cohen, and R. Seidowsky, “Two french hard-shoulder running operations: Some comments on effectiveness and safety,” in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 230–236, Sept 2010.
- [8] N. Buch, S. Velastin, and J. Orwell, “A review of computer vision techniques for the analysis of urban traffic,” *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 920–939, Sept 2011.

- [9] J. Soh, B. T. Chun, and M. Wang, "Analysis of road image sequences for vehicle counting," in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, vol. 1, pp. 679–683 vol.1, Oct 1995.
- [10] E. Bas, A. Tekalp, and F. Salman, "Automatic vehicle counting from video for traffic flow analysis," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 392–397, June 2007.
- [11] P. Khude and S. Pawar, "Object detection, tracking and counting using enhanced bma on static background videos," in *Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference on*, pp. 1–4, Dec 2013.
- [12] I. Sina, A. Wibisono, A. Nurhadiyatna, B. Hardjono, W. Jatmiko, and P. Mursanto, "Vehicle counting and speed measurement using headlight detection," in *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pp. 149–154, Sept 2013.
- [13] G. Salvi, "An automated nighttime vehicle counting and detection system for traffic surveillance," in *Computational Science and Computational Intelligence (CSCI), 2014 International Conference on*, vol. 1, pp. 131–136, March 2014.
- [14] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, pp. 334–352, Aug 2004.
- [15] C. del Blanco, F. Jaureguizar, and N. Garcia, "An efficient multiple object detection and tracking framework for automatic counting and video surveillance applications," *Consumer Electronics, IEEE Transactions on*, vol. 58, pp. 857–862, August 2012.
- [16] H.-Y. Zhang, "Multiple moving objects detection and tracking based on optical flow in polar-log images," in *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, vol. 3, pp. 1577–1582, July 2010.
- [17] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 3, pp. 37–47, Mar 2002.
- [18] A. Lipton, H. Fujiyoshi, and R. Patil, "Moving target classification and tracking from real-time video," in *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, pp. 8–14, Oct 1998.

- [19] A. Sobral and A. Vacavant, “A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos,” *Computer Vision and Image Understanding*, vol. 122, no. 0, pp. 4 – 21, 2014.
- [20] P. Kumar, S. Ranganath, and H. Weimin, “Bayesian network based computer vision algorithm for traffic monitoring using video,” in *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, vol. 1, pp. 897–902 vol.1, Oct 2003.
- [21] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [22] S. Messelodi, C. Modena, N. Segata, and M. Zanin, “A kalman filter based background updating algorithm robust to sharp illumination changes,” in *Image Analysis and Processing â ICIAP 2005* (F. Roli and S. Vitulano, eds.), vol. 3617 of *Lecture Notes in Computer Science*, pp. 163–170, Springer Berlin Heidelberg, 2005.
- [23] C. Stauffer and W. Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, pp.–252 Vol. 2, 1999.
- [24] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [25] C. Guyon, T. Bouwmans, and E. hadi Zahzah, “Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis,” Available from: <http://www.intechopen.com/books/principal-component-analysis/robust-principal-component-analysis-for-background-subtraction-systematic-evaluation-and-comparative>, 2012.
- [26] P. Rodriguez and B. Wohlberg, “Incremental principal component pursuit for video background modeling,” *IEEE Signal Processing Letters*, 2014.
- [27] U. D. of Transportation Publication, “Lankershim boulevard dataset,” Available from: <http://ngsim-community.org/>, January 2007.
- [28] P. Rodriguez, “Incremental principal component pursuit,” Available from: <https://sites.google.com/a/istec.net/prodrig/Home/en/pubs/incpcp>, 2014.
- [29] P. L. Rosin, “Unimodal thresholding,” *Pattern Recognition*, vol. 34, no. 11, pp. 2083 – 2096, 2001.

- [30] J.-H. Chang, K.-C. Fan, and Y.-L. Chang, “Multi-modal gray-level histogram modeling and decomposition,” *Image and Vision Computing*, vol. 20, no. 3, pp. 203 – 216, 2002.
- [31] C. Bouvie, J. Scharcanski, P. Barcellos, and F. Lopes Escouto, “Tracking and counting vehicles in traffic video sequences using particle filtering,” in *Instrumentation and Measurement Technology Conference (I2MTC), 2013 IEEE International*, pp. 812–815, May 2013.
- [32] A. Sanchez, P. Suarez, A. Conci, and E. de Oliveira Nunes, “Video-based distance traffic analysis: Application to vehicle tracking and counting,” *Computing in Science Engineering*, vol. 13, pp. 38–45, May 2011.
- [33] A. Chan, Z.-S. Liang, and N. Vasconcelos, “Privacy preserving crowd monitoring: Counting people without people models or tracking,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–7, June 2008.
- [34] D. Meyer, J. Denzler, and H. Niemann, “Model based extraction of articulated objects in image sequences for gait analysis,” in *Image Processing, 1997. Proceedings, International Conference on*, vol. 3, pp. 78–81 vol.3, Oct 1997.
- [35] P. Rodriguez and B. Wohlberg, “Fast principal component pursuit via alternating minimization,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 69–73, Sept 2013.
- [36] J. Zheng, Y. Wang, N. L. Nihan, and M. E. Hallenbeck, “Extracting roadway background image: Mode-based approach,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1944, no. 1, pp. 82–88, 2006.