

# Aplicación de un método de sustracción de fondo a partir de imágenes de vídeo-vigilancia

Luciano Gervasoni<sup>1,2</sup>, Juan Pablo D'amato<sup>1,2,3</sup>, Rosana Barbuzza<sup>1,2</sup>

<sup>1</sup> Facultad de Cs. de Exactas, Univ. Nac. del Centro de la Prov. de Bs.As., Paraje Arroyo Seco S/N, 7000 Tandil, Buenos Aires, Argentina.

<sup>2</sup> Instituto PLADEMA, Paraje Arroyo Seco S/N, 7000 Tandil, Buenos Aires, Argentina.

<sup>3</sup> Consejo Nacional de Investigaciones Científicas y Técnicas, Av. Rivadavia 1917, CABA, Argentina.

gervasoni.luc@gmail.com

{jpdamato,rbarbu}@exa.unicen.edu.ar

**Resumen** La sustracción de fondo es una técnica ampliamente utilizada para detectar objetos en movimiento a partir de cámaras estáticas. En el campo de la vídeo-vigilancia, es indispensable ya que permite discriminar rápidamente cuando ha surgido un evento y realizar estudios de movilidad de los objetos en escena. En los últimos años, se han propuesto muchos métodos con diferentes fortalezas y requerimientos. Uno de los métodos más novedosos y eficaces, es Visual Background Extractor (ViBe), el cual se basa en un análisis estocástico multimodelo para discriminar objetos del fondo con una alta tasa de aciertos.

En este trabajo se propone una variación de este algoritmo para su aplicación en situaciones particulares como los videos de vigilancia, especialmente para resolver problemas de píxeles intermitentes, objetos pseudo-estáticos, entre otros. Para verificar los resultados se comparan con otras técnicas del estado del arte, obteniendo en general mejores resultados de las métricas *Precision* y *Recall*, y de la correlación promedio entre ambos indicadores. Se tuvieron en cuenta los tiempos de cómputo que involucra el método propuesto.

**Keywords:** Background subtraction, foreground extraction, motion detection

## 1. Introducción

La sustracción de fondo es una etapa fundamental de una gran cantidad de aplicaciones en las que se necesita detectar movimiento, identificar y/o seguir objetos en secuencias de imágenes dinámicas. Entre estas aplicaciones se pueden nombrar la vídeo-vigilancia, como la detección y captura del flujo de movimiento, la estimación de pose, la interacción hombre-computadora o la codificación de vídeo basado en el contenido, entre otros. El proceso de sustracción es también denominada extracción de primer plano o de objetos en movimiento (*foreground extraction*), y consiste en una serie de métodos que permiten distinguir entre

zonas de fondo o estáticas (*background*), y zonas dinámicas que se corresponden con el primer plano (*foreground*).

Para lograr extraer el fondo se mantiene una representación de la escena mediante el modelado digital del fondo. Esta representación debe contener la información necesaria para en cada instante determinar, por cada nueva imagen de entrada, qué áreas de la misma se corresponden efectivamente con el fondo. Dado que el fondo no es invariante en el tiempo, es necesario realizar un modelado dinámico del fondo para adaptarlo a las variaciones que se puedan producir en el mismo. La primera vez, cualquier cambio significativo en una región de la imagen respecto del modelo de fondo será considerado *foreground*. En general, cada elemento del modelo se actualiza por cada cierto intervalo de tiempo, siempre que este haya perdido significancia.

En el contexto de la vídeo-vigilancia, existen ciertos desafíos que aún no han sido resueltos con éxito, debido a la necesidad de robustez de la detección pese a múltiples factores que se presentan en este tipo de secuencias. En la práctica, el ambiente tiende a ser complejo debido a la diversidad de alteraciones que pueden presentarse. Se pueden destacar los cambios en la iluminación, la existencia de varios objetos en movimiento, la superposición de éstos, cambios en las estructuras estáticas y la presencia de ruido en la captura de las imágenes, entre otras.

En este trabajo presentamos mejoras a un algoritmo de sustracción de fondo para su aplicación al análisis de objetos en video vigilancia. Para mejorar la tasa de detección (aciertos), se proponen y analizan algoritmos que se aplican las etapas de preparación de la imagen y de actualización del modelo. Describiremos un método para eliminar ruido que se presenta de forma intermitente y una serie de operaciones morfológicas para eliminar elementos considerados *background* que en realidad pertenecen a los objetos en movimiento. En la siguiente sección discutiremos brevemente trabajos actuales relacionados, en la sección 3 describiremos el método implementado y a continuación las mejoras propuestas. En la sección de resultados presentaremos algunos casos de estudio reconocidos como *benchmarks* de cualquier algoritmo de esta temática y las conclusiones a las que arribamos.

## 2. Trabajos Relacionados

Existe una gran cantidad de contribuciones a la sustracción de fondo han sido propuestas, así como también revisiones, comparativas y *data-sets* [8,4,3]. Entre ellas, se encuentran las técnicas que ajustan una función de densidad de probabilidad Gaussiana, o una mezcla de ellas (*Mixture of Gaussians*). Estos últimos se basan en realizar una estimación del fondo más probable, y determinar para cada valor de entrada que probabilidad tiene de pertenecer al fondo. A su vez, necesitan de parámetros que permitan actualizar el modelo, para ir ajustándolo a lo largo de una secuencia. La principal desventaja radica en que la suposición del comportamiento Gaussiano no siempre se sostiene, sobre todo en escenas complejas [7,10,11].

Por otra parte, existen sustractores de fondo basados en el análisis de texturas [5,6]. Su principal ventaja es que son robustos cuando el modelo de fondo y el objeto en movimiento presentan una gran similitud en la información de color. Sin embargo, tienen un alto costo computacional ya que requieren analizar y comparar texturas de forma reiterada en la imagen.

El análisis estocástico para modelar el fondo ha mostrado altas tasas de detección en entornos complejos, tal cual suelen ser típicos en la vídeo-vigilancia. En particular, en este trabajo nos centramos en el desarrollo de un sustractor de fondo basado en un método propuesto en la literatura [1,2,9] con distintas variantes. Entre las virtudes de estos métodos se destacan el bajo tiempo de cómputo, las altas tasas de detección y la robustez ante la existencia de ruido, las cuales son necesarias en capturas de cámaras de supervisión, pero que no son consideradas en conjunto en las restantes técnicas nombradas. Estas características son imprescindibles en el contexto de la vídeo-vigilancia. Por otro lado, tienen ciertas falencias reconocidas, como su falla en la detección cuando los objetos son similares al fondo.

La técnica base mencionada que se utiliza en este trabajo utiliza información de color para determinar la detección del fondo. El modelado de fondo está basado en muestras reales obtenidas de la secuencia, razón por la cual se considera no-paramétrica, ya que a diferencia de otras técnicas no debe realizar un estudio de ajuste u optimización de parámetros para modelar el fondo. La actualización de modelo es ejecutada aleatoriamente, reemplazando uno de los valores guardados anteriormente por el nuevo. La aleatoriedad sobre este mecanismo de actualización permite que valores introducidos para modelar el fondo tengan un tiempo de vida mayor, reduciendo gradualmente con el tiempo la probabilidad de que persistan en el modelo. Contiene una política de actualización conservativa, ingresando al modelo de fondo únicamente valores clasificados como fondo.

### 3. Sustractor de fondo sobre cámaras estáticas

El método ViBe [1,2] utilizado para discriminar el fondo de los objetos, se realiza a nivel de píxel. Para esto, para cada píxel se almacenan  $N$  muestras de frames anteriores al actual elegidos aleatoriamente, y se determina que son fondo aquellos que son estadísticamente invariante en el tiempo. En la práctica se define  $N$  como 20. Para cada muestra, puede guardarse tanto la información de color (por ej. en el espacio de colores RGB) como su valor en tono de grises, dependiendo de la imagen de entrada con la que se cuenta.

Como consecuencia de la detección o salida realizada por cada imagen o cuadro del video, se debe realizar un proceso de actualización del modelo que permita adaptar gradualmente la representación a los diferentes cambios que ocurren a lo largo de una secuencia. Esta actualización se realiza una vez realizada la sustracción de fondo en el cuadro actual y consiste en agregar en el modelo aquellos píxeles pertenecientes al fondo. Así mismo, aquellos clasificados como parte del movimiento se asegura que no se inserten en el modelo, ya que puede alterar significativamente los resultados en la detección.

A continuación se describe el algoritmo seguido de heurísticas que se aplican para mejorar los resultados conservando el bajo tiempo de cómputo que lo caracteriza. En el final del trabajo se presenta un pseudo-código de la implementación.

### 3.1. Actualización del modelo

La metodología de actualización para un píxel es estocástica y consiste en reemplazar el nuevo valor por uno de los que ya existía, según una probabilidad definida como  $1/\varphi$ . Es decir, en promedio se realizará un reemplazo en el modelo cada  $\varphi$  píxeles que se corresponden con el fondo. La decisión sobre cuál de los  $N$  valores del modelo correspondiente a un píxel reemplazar es definido mediante un mecanismo aleatorio con distribución equiprobable. Por este efecto, el módulo de actualización del modelo es eficiente, ya que no debe reemplazar un valor específico ni realizar ningún procesamiento adicional de estimación sobre el modelo.

Por otro lado, también se puede realizar una propagación espacial de cada píxel al modelo de alguno de sus vecinos con el mismo esquema probabilístico de reemplazo (se utiliza el mismo factor  $\varphi$ ). Se considera vecino a cualquier píxel conectado en las ocho posibles direcciones respecto a su posición.

El hecho de que el tiempo de vida de los distintos valores del modelo no sean ponderados según el tiempo es un factor decisivo en el mecanismo de actualización. De esta manera, se permite la persistencia de modelos que tengan un tiempo de vida relativamente mayores a otros. Resulta de utilidad en una gran diversidad de situaciones, principalmente cuando el fondo es frecuentemente superpuesto por objetos en movimiento (por ej. veredas en las que transitan continuamente personas, o calles en las que pasan muchos autos).

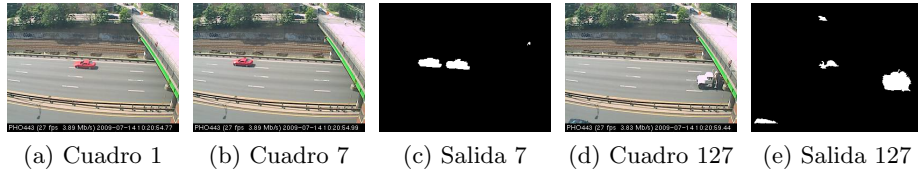
### 3.2. Inicialización del modelo

La técnica provee una inicialización rápida, de manera que puede sustraer el fondo desde el primer cuadro sin necesidad de una etapa de entrenamiento, ni de ningún tipo de ajuste de parámetros supervisado. Esto se realiza, en primer lugar, inicializando  $N/2$  valores del modelo con el valor del píxel del primer cuadro, y el resto se completa con los valores de píxeles vecinos elegidos aleatoriamente.

En segundo lugar, se modifica la probabilidad de reemplazo  $\varphi$ , utilizada para actualizar y propagar el modelo. Dicho cambio tiene efecto por una determinada cantidad de cuadros siguientes  $\kappa$ . En consecuencia, se asegura una rápida absorción de los “fantasmas” en el modelo que pueden aparecer debido a la falta de información en el primer cuadro, la cual se puede verificar en la Figura 1. Se puede observar en esta figura, que la imagen fantasma desaparece en el cuadro 127, cuando de otra forma se produciría varios cuadros más adelante.

Esta ha probado ser una forma rápida y eficaz de inicializar el modelo. A través del mecanismo de propagación de los vecinos realizada en dicha inicialización se simula la propiedad espacio-temporal presente en una secuencia. En la práctica se define  $\kappa$  en 150, y  $\varphi$  como 1, que una vez terminada la inicialización se le asigna su valor original 8.

Figura 1: Supresión de “fantasmas” en la inicialización



### 3.3. Detección del fondo

Para clasificar las zonas que se corresponden con el fondo se realiza una verificación por cada píxel  $p_t$  de la imagen de entrada. En primer lugar, se define la condición de intersección entre dos valores  $\{p_t, m_i\}$  cuando la distancia evaluada entre el valor del píxel actual y un valor del modelo  $m_i$  es menor o igual a un radio  $R$ .

$$I(p_t, m_i) = \begin{cases} 1 & \text{si } D(p_t, m_i) \leq R \\ 0 & \text{en otro caso} \end{cases} \quad (1)$$

Luego, se compara cada píxel de entrada con sus  $N$  correspondientes ejemplares del modelo, verificando la cantidad de valores que cumplen la condición de intersección. Finalmente, se utiliza la cardinalidad mínima ( $\#Min$ ) para clasificar al píxel.

$$Clasificar(p_t, m) = \begin{cases} \text{Fondo} & \text{si } \sum_{i=1}^N I(p_t, m_i) \geq \#Min \\ \text{No fondo} & \text{en otro caso} \end{cases} \quad (2)$$

Además, se debe determinar la manera de evaluar la distancia entre dos valores dados, la cual es dependiente de los valores de entrada. En este caso, se optó por utilizar el modelo de color RGB, extrayendo los componentes verde y rojo y computando la distancia euclidiana de los mismos. Como futuro trabajo, está la posibilidad de analizar otros espacios de colores como HSV, CMYL u otro.

$$D(v_1, v_2) = \sqrt{(v_{1R} - v_{2R})^2 + (v_{1G} - v_{2G})^2} \quad (3)$$

El hecho de excluir el componente azul de la distancia euclidiana se debe a dos factores fundamentales. El primero se corresponde con el objetivo de reducir los tiempos de cómputo, ya que la operación de distancia se realiza con una frecuencia elevada. Además, en las secuencias de vídeo-vigilancia el componente azul suele encontrarse más saturado que los otros componentes, por lo que los componentes verdes y rojo alcanzan para determinar correctamente la distancia entre dos valores.

Por otra parte, resulta importante notar que para el caso de imágenes en escala de grises, la comparación se realiza computando la diferencia en valor absoluto entre ambos componentes. Este caso es más eficiente, ya que involucra únicamente una comparación entre dos bytes.

El valor utilizado para el radio es sensible a la detección que realice posteriormente. Si se utiliza un radio bajo, existe una tendencia a que se aumente la cantidad de puntos *foreground* ya que se reduce el espacio sobre el cual los puntos de los modelos deben estar para que pertenezca al fondo. Aquellas perturbaciones producidas por el ruido aparecen como falsos positivos y, a su vez, las sombras también se encuentran lo suficientemente alejados de los modelos del fondo como para que se clasifiquen como estos. Por otro lado, en el contexto de un radio alto los falsos positivos se reducen considerablemente mientras que aumentan los verdaderos negativos, aquellos puntos que no deberían ser considerados fondo. La explicación subyacente radica en una probabilidad de aceptación mayor sobre cada comparación de un píxel de entrada contra los modelos del fondo.

## 4. Mejoras propuestas

En vistas de aumentar las tasas de detección se proponen ciertas heurísticas al algoritmo base. A continuación se detallan las mejoras en el proceso de actualización del modelo, la inserción de objetos en el fondo cuando sea conveniente así como también evitarlo cuando no corresponda. También se mejora la clasificación mediante un análisis de los objetos detectados. La mayoría de estas propuestas también han sido parte de la extensión al algoritmo en [2]. En esta sección realizamos una descripción de las variantes realizadas y de las cuestiones estudiadas para mejorar los resultados.

### 4.1. Parpadeo de píxeles

Frecuentemente la clasificación de un píxel de una imagen alterna entre fondo y no fondo en cuadros continuos. Cuando un determinado píxel realiza este intercambio de estado en cuadros sucesivos, se lo considera que se encuentra *parpadeando*. Dicho caso se relaciona con la existencia de fondos multi-modales, el cual presenta para un determinado píxel la posibilidad de que existan múltiples fondos. En la práctica se guarda la máscara de clasificación resultante del cuadro anterior previa a cualquier modificación de las heurísticas, para compararla frente a la actual máscara de clasificación, también sin ninguna modificación.

Se utiliza una estructura que indica el nivel de parpadeo de cada píxel. Si un determinado píxel pertenece al límite entre el objeto y el fondo, y además la máscara de clasificación para dicho píxel es diferente a la del cuadro anterior, entonces se sospecha que puede ser un caso de parpadeo y se aumenta el nivel del píxel en 15. En caso contrario, el nivel se reduce en 1. El rango de trabajo de análisis aceptado se encuentra en [0..150].

Consideramos que si un píxel contiene un nivel de parpadeo mayor o igual a 30, se procede por removerlo de la máscara que servirá de actualización en el proceso posterior. De esta manera, el nivel de parpadeo se aumenta únicamente en los bordes, impidiendo que actualicen el modelo si presentan un parpadeo constante a lo largo de los cuadros más recientes.

En esta heurística se introduce una nueva máscara que se utiliza para actualizar el modelo (definida en la sección 3.1). En consecuencia, no se actualizarán píxeles que hayan sido detectados como *foreground*, ni aquellos que demostraron *parpadear* según lo definido anteriormente. Como ventaja adicional a introducir la máscara de actualización, se permite la posibilidad de obtener un *feed-back* de la aplicación que utilice la sustracción de fondo, brindando la opción de mejorar la detección en cuadros posteriores. Por ejemplo, ante una inminente estimación de pose puede que se realice un refinamiento de siluetas de los objetos, lo cual sirve para no actualizar en el modelo píxeles que no pertenecían al fondo.

A diferencia de otra propuesta [9], no se realizan etapas de pos-procesamiento distintas para cada una de las máscaras debido al costo computacional en el que incurren, además de que no producen mejoras apreciables. Para esta heurística, resulta vital denotar que se busca una posterior actualización del modelo de fondo únicamente para aquellos píxeles pertenecientes al fondo, sea según la clasificación obtenida previamente o por información proveniente de una aplicación complementaria.

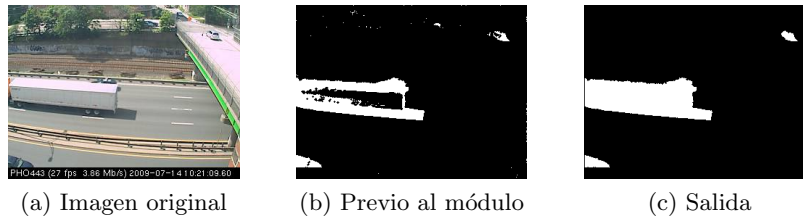
## 4.2. Análisis de objetos

En la heurística que estudia los objetos, se realizan modificaciones sobre la segmentación inicial producida en la sección 3.3. En primer lugar, se realiza la operación morfológica de *cierre* sobre la salida binaria una vez efectuada la detección. En consecuencia, se permite que componentes que se encuentran muy cercanos entre sí se terminen de conectar en el caso de que no lo estén. Se efectúa utilizando un kernel de diámetro 3. Esto permite mejorar el proceso de llenado de agujeros posterior.

Por otra parte, resulta importante descartar posibles falsos positivos para proveer una mejora en el proceso de sustraer el fondo. Éstos existen frecuentemente en forma de objetos muy pequeños aislados de cualquier otro objeto. Por lo tanto, se realiza un filtrado de aquellos objetos que contengan un área menor o igual a 10. Dicho valor ha demostrado filtrar una gran cantidad de objetos detectados que se correspondían con el fondo, mientras que los objetos de interés que se encuentran en movimiento no los filtra debido a que, en la vasta mayoría de secuencias, su tamaño es mayor al valor dado para filtrarlo.

Finalmente, se rellenan los contornos de los objetos que se detectaron como *foreground*. Resulta intuitivo analizar que si se detectó un contorno de un objeto como movimiento y, a su vez su interior se encuentra en su mayoría detectado como parte del objeto, pequeños sectores dentro del mismo que inicialmente hayan sido detectados como fondo se corresponderían con el movimiento del mismo objeto. En la Figura 2 se puede observar el efecto de aplicar la heurística.

Figura 2: Efecto del módulo “Análisis de objetos”



### 4.3. Inhibición de propagación

En el contexto de la video-vigilancia, es deseable estudiar el comportamiento de las personas cuando dejan abandonados objetos en lugares públicos (por ej. una mochila, un portafolio). En algunos casos estos comportamientos se dan de manera intencional. Es importante detectar estas situaciones para poder a continuación verificar qué es lo que está ocurriendo en ese momento y poder avisar o generar una notificación de algún tipo.

Con el objetivo de evitar insertar un objeto de interés en el modelo del fondo, aunque se encuentre estático por un lapso de tiempo en la escena, se provee una heurística para conservar la condición de *foreground* para estos objetos (durante la actualización de la sección 3.1). Correspondiente a la sección de actualización del modelo, se proporciona un mecanismo en el cual se inhibe el módulo de propagar un valor insertado en el modelo a sus vecinos, el cual es la variable que mayor peso tiene en esta acción.

En la práctica, se computa el gradiente en los bordes internos de objetos del fondo. Si el valor de este gradiente es mayor a un valor definido  $\beta$ , se inhibe la propagación del modelo correspondiente, evitando que crucen hacia objetos que no se corresponden con el fondo. De esta manera, un objeto de interés que se encuentra estático, no se incorpora al fondo de inmediato. En la práctica se define  $\beta$  como 30, que inhibe la propagación durante 180 frames.

### 4.4. Inserción de objetos en el fondo

De manera complementaria a la inhibición de propagación, hay situaciones en la que es deseable insertar objetos que permanezcan estáticos en el modelo aunque no se hayan detectado como fondo. Se dan frecuentemente en casos en los que el fondo estático cambia, por lo que se debe adaptar el modelo de fondo para que contemple estos casos.

En la práctica, cada 5 cuadros se analiza la cantidad de puntos detectados como *foreground*. Se compara la cantidad detectada en los 5 cuadros previos frente a los 5 más recientes. Si la resta de los más recientes respecto a los cuadros previos es mayor a la cantidad de píxeles que tiene la imagen ponderado por un factor de tamaño del objeto, se activa el módulo de inserción de objetos. El



factor se considera la mitad del alto de la imagen. Esto sucede, cuando hubo un cambio repentino en el que se detectó mucho movimiento en pocos cuadros.

Si está activado el módulo, se modifica la variable que representa la probabilidad de reemplazo en el proceso de actualización, definiendo  $\varphi$  como 2, y se desactiva la inhibición de propagación (sección 4.3). Este comportamiento tiene un efecto por  $\kappa$  cuadros (análogo al proceso de inicialización definida en 3.2), luego volviendo ambos a su estado original.

### Pseudo-código del algoritmo

```

Clasificar_cuadro(Input, Modelos){
  Por cada píxel p de Input{
    Salida[p.y][p.x]= Clasificar(p,Modelos[p.x][p.y]);
  }
  return Salida;
}
cMin=2, R=20, N=20,  $\varphi=1$ ,  $\beta=30$ ,  $\kappa=150$ ;
Matriz[ancho][alto][N] Modelos;
Matriz[ancho][alto] Salida,Máscara_actualización;
Matriz[ancho][alto] Segmentación_previa,Segmentación_actual;

Inicialización_modelo(Input);
Por cada Input{
  Salida = Clasificar_cuadro(Input,Modelos);
  Segmentación_actual=Salida;
  Salida=Análisis_objetos(Salida);
  Máscara_actualización = Salida |
    Parpadeo(Segmentación_previa,Segmentación_actual);
  Actualización_modelo(Máscara_actualización,Modelos, $\varphi$ );
  Segmentación_previa=Segmentación_actual;
  if ( t ==  $\kappa$  )  $\varphi=8$ ;
}

```

## 5. Resultados

En esta sección se presentan las métricas resultantes y los tiempos de cómputo necesarios para ejecutar el algoritmo sobre los distintos *data-sets* analizados. El conjunto de pruebas se realizó sobre una computadora portátil con un procesador I7 2.2 GHz bajo la carga de un típico sistema operativo. La implementación fue llevada a cabo en C++ utilizando la biblioteca OpenCV .

Los *data-sets* utilizados se corresponden con una contribución [4] en la cual brindan el *ground-truth* de cada secuencia para computar luego la tasa de detección obtenida. Las secuencias de *Highway* y *Office*, y *Street light* y *Parking*, correspondientes respectivamente a las categorías de *Baseline* e *Intermittent object motion* fueron elegidas debido a que son secuencias típicas en el contexto de la vídeo-vigilancia.

Las métricas de *Precision* y *Recall* fueron computadas y comparadas con otras técnicas ampliamente utilizadas en la sustracción de fondo. El resultado es mejor, cuanto más cercana a 1 es la métrica.

$$Precision : \frac{TP}{TP + FP} \quad Recall : \frac{TP}{TP + FN} \quad (4)$$

Entre las técnicas con las cuales se compara se encuentran los algoritmos base del desarrollo propuesto, y dos técnicas basadas en *Mixture of Gaussians* (MoG) implementadas en la funcionalidad de sustracción de fondo de la librería OpenCV. En la Tabla 1 se presentan los resultados de estas métricas tal cual han sido previamente publicados y en la Figura 3 se visualizan algunos cuadros procesados con los distintos algoritmos. En la Tabla 1, se incluye un indicador de correlación promedio entre las métricas *Precision* y *Recall*. En esta figura se escogieron frames en los cuales el algoritmo se diferenciaba más del resto de las implementaciones.

Casos	Propuesto		Vibe		Vibe+		GMM-Z		GMM-KTP	
	Pr	Rc	Pr	Rc	Pr	Rc	Pr	Rc	Pr	Rc
Highway	0.91	<b>0.97</b>	0.92	0.85	<b>0.93</b>	0.93	0.92	0.89	0.91	0.65
Office	0.94	<b>0.93</b>	<b>0.97</b>	0.79	0.92	0.70	0.93	0.51	0.99	0.36
Street light	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>	0.72	<b>0.99</b>	0.54	0.92	0.34	<b>0.99</b>	0.23
Parking	0.91	0.52	0.82	0.25	<b>0.95</b>	0.14	0.74	<b>0.69</b>	0.66	0.37
<b>Correlación</b>	0.80255		0.6165		0.5444		0.5291		0.3549	

Cuadro 1: Valores de *Precision -Recall* por cada caso y para cada método

Se puede observar, que en la mayoría de los casos se mejora la métrica *Recall* con el método propuesto, sin perjudicar la métrica *Precision*, generando una correlación más alta. Solamente el algoritmo GMM-Z mejora el caso de *Parking*, pero reduciendo drásticamente los verdaderos positivos. De todas maneras, el método falla en casos como la omisión de sombras (dataset *Highway*) o en algunos casos el rellenado de objetos (correspondiente a *Office*) debe ser selectivo. Por esto, se deben mejorar dichas situaciones.

En la Tabla 2 se presentan las métricas de tiempo de procesamiento, medida en milisegundos por cuadro (*MS*) y en cuadros por segundo (*FPS*).

Se observa, que los resultados de tiempo se encuentran siempre muy por encima de los 30 FPS (suficientes para realizar procesamiento en tiempo real). Los tiempos del método propuesto son un poco más alto que el resto de las técnicas, pero se justifica por el aumento de la precisión. En vistas a mejorar este tiempo, y pensando en trabajar a una mayor resolución de imagen, se está trabajando en una versión que funcione en placas gráficas.

Casos	Propuesto		Vibe		Vibe+		GMM	
	MS	FPS	MS	FPS	MS	FPS	MS	FPS
HighWay	7.3	137	6.2	161	8.1	123	6	167
Office	8	125	7.1	141	8.9	112	6.5	154
StreetLight	7.8	128	7	143	8.5	118	6.4	156
Parking	6.8	147	5.9	169	7.7	130	5.6	179

Cuadro 2: Tiempos de cómputo

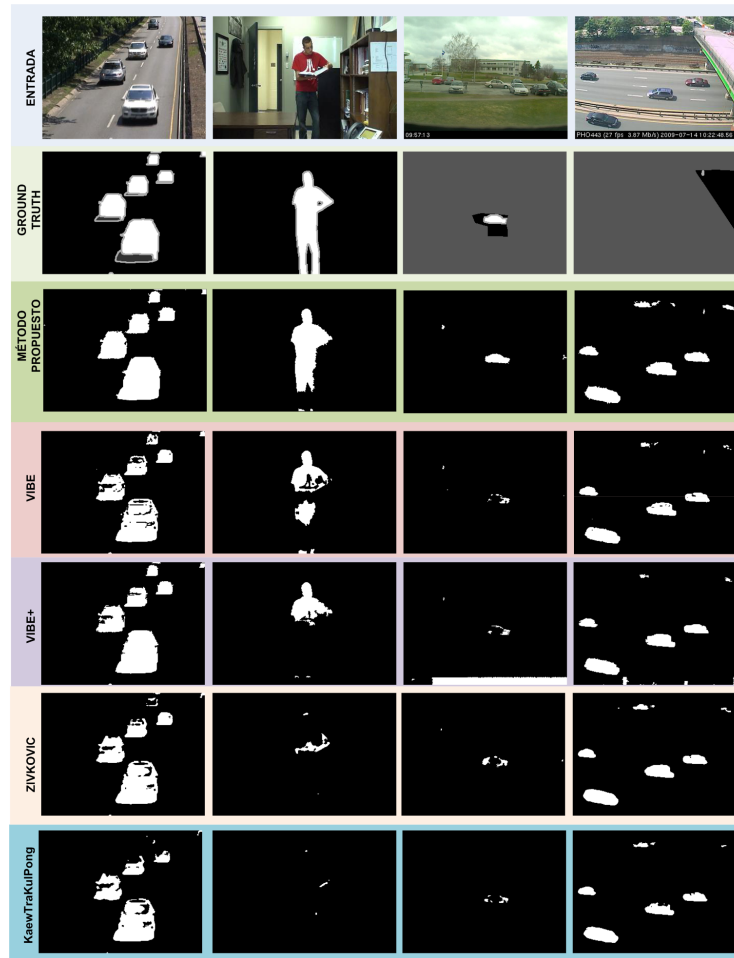


Figura 3: Comparación de los resultados aplicando distintos algoritmos.

## 6. Conclusiones

Se presentó un algoritmo de sustracción de fondo multi-modelo que utiliza reemplazo estocástico. Se proveyeron distintas técnicas y variantes que permi-

tieron aumentar la tasa de detección, basándose en ciertas características del contexto de la video-vigilancia. Los bajos tiempos de cómputo permiten realizar la sustracción de una gran cantidad de cuadros por segundo, por lo que se puede llegar a utilizar en arquitecturas embebidas (como un ARM).

El algoritmo tiene algunas dificultades para discriminar objetos similares al fondo (denominado camuflaje) por lo que se continuará estudiando y mejorando esta propuesta. Como comentario adicional, el código fuente de este algoritmo se comparte solicitándolo a los autores.

## Referencias

1. Olivier Barnich and Marc Van Droogenbroeck. Vibe: a powerful random technique to estimate the background in video sequences. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 945–948. IEEE, 2009.
2. Olivier Barnich and Marc Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6):1709–1724, 2011.
3. Yannick Benezeth, P-M Jodoin, Bruno Emile, Hélène Laurent, and Christophe Rosenberger. Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.
4. Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. Changedetection.net: A new change detection benchmark dataset. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 1–8. IEEE, 2012.
5. Marko Heikkilä and Matti Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):657–662, 2006.
6. Marko Heikkilä, Matti Pietikainen, and Janne Heikkilä. A texture-based method for detecting moving objects. In *BMVC*, pages 1–10, 2004.
7. Pakorn KaewTraKulPong and Richard Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-Based Surveillance Systems*, pages 135–144. Springer, 2002.
8. Antoine Vacavant, Thierry Chateau, Alexis Wilhelm, and Laurent Lequière. A benchmark dataset for outdoor foreground/background extraction. In *Computer Vision-ACCV 2012 Workshops*, pages 291–300. Springer, 2013.
9. Marc Van Droogenbroeck and Olivier Paquot. Background subtraction: experiments and improvements for vibe. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 32–37. IEEE, 2012.
10. Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE, 2004.
11. Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters*, 27(7):773–780, 2006.