

## UN MÉTODO PARA LA SUSTRACCIÓN DE FONDO EN VIDEOS INESTABLES

**Rosana Barbuzza<sup>a</sup>, Juan P. D'amato<sup>a,c</sup>, Aldo Rubiales<sup>a,b</sup>, Leonardo Dominguez<sup>a,c</sup>,  
Alejandro Perez<sup>a,b</sup> y Marcelo Vénere<sup>a</sup>**

<sup>a</sup>*Instituto PLADEMA, 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, rbarbuzza@gmail.com*

<sup>b</sup>*Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina*

<sup>c</sup>*Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina*

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

**Abstract.** La sustracción de fondo es una técnica ampliamente utilizada para extraer objetos de interés en movimiento a partir de cámaras estáticas. En muchos casos, sobre todo en la vídeo-vigilancia, ocurre que las cámaras sufren de leves movimientos en la escena debido a factores como el viento o vibraciones cuando están ubicadas al aire libre, que dificultan la detección correcta de objetos. Un problema similar sucede cuando el fondo es dinámico, en donde partes de la escena que presentan movimiento constante, se corresponden con el fondo. Este es un desafío en el proceso de la sustracción de fondo que requiere en la mayoría de los casos funcionalidades adicionales para mejorar la detección de objetos en diferentes escenarios. En este artículo, se propone una variante al método tradicional de sustracción de fondo, Visual Background Extractor (ViBe), para considerar situaciones de escenas inestables. Este método basado en análisis estocástico multimodelo para discriminar objetos del fondo, resulta flexible para incorporar modificaciones que consideren la inestabilidad en el fondo. Los resultados obtenidos con los videos reales analizados, con escenas tanto en ambientes cerrados o no, muestran que se puede mejorar la tasa de confianza del método convencional, como también a bajo costo computacional lo cual permite aplicarse a casos de análisis en tiempo real necesarios para vídeo-vigilancia.

## 1 INTRODUCCIÓN

La sustracción de fondo es una de las etapas más importantes de preprocesamiento en muchas aplicaciones de *Computer Vision*, 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, la estimación de pose o flujo de movimiento, la codificación de vídeo basado en el contenido, entre otros (Brutzer et al., 2011; David Schreiber, 2009). Los objetos en movimiento, o zonas de la escena donde ocurre un cambio significativo son clasificados como *foreground*, mientras que los elementos estáticos de la escena son referidos como *background*. En la práctica esta distinción no siempre es obvia. Los cambios repentinos en la escena de objetos estáticos por movimientos del objeto mismo o de cámara generan modificaciones importantes en el fondo a considerar por el sustractor. Así por ejemplo, un caso dificultoso de clasificar se refiere a partes de la escena, que si bien presentan movimiento, se corresponden con el fondo, como el movimiento leve de las hojas de un árbol producto del viento, o el movimiento del agua que genera olas junto al reflejo de la luz sobre ellas. Estos casos presentan un desafío particular por el cual se necesita determinar el grado aceptable, en el que se cuenta con la presencia de fondo dinámico. A su vez, determinar la barrera en la que el fondo dinámico deja de pertenecer al fondo, es objeto de dificultades. Es necesario entonces, construir un proceso confiable de segmentación de regiones con movimiento relevante de manera de mejorar la performance de aplicaciones construidas sobre la base de esta etapa.

## 2 MÉTODOS DE SUSTRACCIÓN DE FONDO

Dada la importancia de la detección de cambios en las imágenes, existen trabajos relacionados en la literatura y un gran número de variantes. En general, los algoritmos de sustracción de fondo se pueden clasificar según el modelo del fondo que utilizan, y la forma en que calculan la distancia con este modelo para clasificar las escenas. El modelo de fondo generalmente se describe a nivel de píxel, (ej. modelado por el valor de intensidad promedio, por varios píxeles anteriores (Barnich y Van Droogenbroeck, 2009, 2011; Van Droogenbroeck y Paquot, 2012), o como una o múltiples distribuciones Gaussianas (Zivkovic, 2004; Yin Li, 2012)). A su vez, estos métodos utilizan parámetros que permiten actualizar el modelo del fondo para ir ajustándolo a los cambios que se producen a lo largo de una secuencia de video. Es común postprocesar la salida de la sustracción de fondo, para mejorar los resultados de los métodos mencionados. Para la evaluación se utilizan *data-sets* tradicionales (Vacavant et al., 2013; Goyette et al., 2012; Benzeth et al., 2008; Piccardi, 2004), de manera de comparar la performance entre los mismos. Los métodos típicos de postprocesamiento a la sustracción de fondo incluyen por ejemplo, la remoción de ruido, la aplicación de filtros de suavizado, entre otros, sin embargo todos ellos son muy dependientes de los parámetros utilizados para las distintas secuencias de video, y además tienen el problema de aumentar el tiempo computacional de los algoritmos.

En particular, este trabajo se centra en el desarrollo de un sustractor de fondo basado en píxel propuesto en (Barnich y Van Droogenbroeck, 2009), el cual tiene mejor comportamiento que los otros métodos mencionados, en entornos variados como suelen ser típicos en la vídeo-vigilancia. Entre las virtudes de este método 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 utilizadas hoy en día. Por otro lado, tiene cierta falencia por utilizar un modelo de fondo basado en píxel, que lo hace muy susceptible a los movimientos de cámara, o al fondo dinámico (Schick et al., 2012). Por esta razón, se propone extender el método original utilizando una representación del modelo de fondo en un nivel mayor al píxel,

Figure 1: Ejemplo de fondo dinámico



(a) Imagen original



(b) Segmentación Vibe sin postprocesamiento

particionando la imagen en bloques que consideran algunos vecinos del mismo. Esta división en bloques de píxeles cercanos conforman una grilla de modelos para el fondo. Este cambio en la representación, permite obtener mejores segmentaciones, reduciendo principalmente los falsos positivos y falsos negativos en la clasificación *foreground/background*, sin necesidad de aplicar algoritmos de postprocesamiento de la salida del sustractor.

### 3 EL MÉTODO *VISUAL BACKGROUND EXTRACTOR*

El método ViBe original guarda muestras del valor de intensidad de cada píxel, obtenidas de la historia o secuencia previa de ese píxel, en un modelo que es representativo del fondo. La actualización del modelo de cada píxel es aleatoria, con cierta probabilidad de reemplazar uno de los valores guardados, por un nuevo valor de intensidad dado en la secuencia. La aleatoriedad sobre este mecanismo de actualización permite reducir gradualmente con el tiempo, la probabilidad de que las muestras guardadas persistan en el modelo.

Si se considera  $M = \{m_1, m_2, \dots, m_N\}$  el conjunto de muestras de valores de intensidad correspondientes a un píxel  $p_t$ , cada  $m_i$  para  $i=1..N$ , consiste en un vector  $m_i = (r, g)$  que usa dos componentes del espacio RGB. Esto puede adaptarse fácilmente para extenderlo al espacio RGB, escala de grises u otro. Sin embargo, el hecho de excluir el tono azul, se debe a que en las secuencias de vídeo-vigilancia este componente se encuentra más saturado que el resto. Luego, se define la condición de intersección  $D(p_t, m_i)$  igual a 1, cuando la distancia euclídea evaluada entre el valor de intensidad del píxel actual y una muestra del modelo  $m_i$  es menor o igual a un determinado valor de radio  $R$ , de lo contrario es igual a 0, es decir:

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

La clasificación consiste en comparar cada píxel de la imagen actual con las  $N$  muestras del mismo que se encuentran almacenadas en el modelo  $M$ , y mediante un indicador determinar si es *background* o *foreground*, 1 o 0 respectivamente. Luego, la siguiente función

$$\text{Clasificar}(p_t, M) = \begin{cases} 1 & \text{si } \sum_{i=1}^N D(p_t, m_i) \geq \#Min \\ 0 & \text{en otro caso} \end{cases} \quad (2)$$

define una máscara binaria de clasificación para cada píxel del *frame* de entrada, #Min es la cantidad mínima de veces que debe ser verdadera la función distancia entre  $p_t$  y cada muestra  $m_i$  para considerarse *background*. Como consecuencia de aplicar al Ec. 2, por cada imagen del video, se genera una salida binaria correspondiente a la clasificación de cada píxel.

Luego de la detección, se actualiza el modelo cada píxel en forma estocástica, para adaptar gradualmente la representación a los diferentes cambios que ocurren a lo largo de una secuencia. Solamente aquellos píxeles procesados y clasificados como *background* pueden ingresar al modelo de fondo correspondiente. La metodología de actualización del modelo de cada píxel ya procesado *background*, consiste en reemplazar una muestra  $m_i$  de su modelo M, elegida aleatoriamente, por el nuevo valor del píxel procesado, 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 muestra  $m_i$  se reemplaza, de las N muestras del modelo, 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 mismo. También ViBe puede realizar una propagación espacial de cada píxel procesado *background* 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. Los valores recomendados son  $\varphi = 8$  y  $N=20$ ,  $R=20$  y  $\#Min=2$  (Barnich y Van Droogenbroeck, 2009, 2011).

La Figura 1 derecha, muestra la salida del algoritmo ViBe, para el *frame* mostrado a la izquierda. Uno de los problemas típicos se produce cuando un objeto comienza el movimiento para actualizar el fondo detrás del mismo, se generan por esta situación imágenes fantasmas que van desapareciendo al procesar los siguientes *frames*. Se puede observar 3 formas de personas que no están en la escena del *frame*, pero que estuvieron anteriormente detenidas y comenzaron el movimiento.

Es deseable que sólo aquellos píxeles clasificados como *background* se inserten en el modelo, ya que la inserción de píxeles que pertenecen a objetos en movimiento, o que resultan inciertos en la clasificación, pueden alterar significativamente los resultados de la detección. Esto último es muy importante ya que ha permitido la introducción de mejoras al método original logrando mayor tasa de detección. En un trabajo anterior se implementó una mejora al módulo de actualización del modelo para permitir o inhibir la propagación, de objetos *foreground*, en el modelo de fondo. Se aceleró o retrasó la incorporación de muestras al modelo para mejorar las imágenes fantasmas, y por otro se realizó un proceso de filtrado para que objetos pequeños no sean considerados falsos positivos. También, se incorporó un módulo *Fill Object*, que corrige la clasificación de aquellos píxeles clasificados según Ec. 2 como *background*, pero que en realidad es probable que pertenezcan al interior de objetos en movimiento (falso negativo) (Gervasoni et al., 2014). Como se puede ver en la Figura 2 el algoritmo permite que componentes que se encuentran muy cercanos entre sí terminen de conectar realizando una operación morfológica de cierre sobre la salida binaria. En ciertos casos resulta eficiente el rellenado como en el caso de los autos, sin embargo, para el cierre de la línea de la ruta, detectada por movimiento de cámara debería descartarse. En la Figura 3 (a), correspondiente a la segmentación de la Figura 1, no se logra mejorar los falsos positivos correspondientes al fondo dinámico (cinta en

movimiento), ni tampoco eliminar rápidamente las imágenes fantasmas. El cierre de componentes en la Figura 3 (a) del espacio entre las piernas o brazos de las personas es un problema aún muy difícil de resolver correctamente.

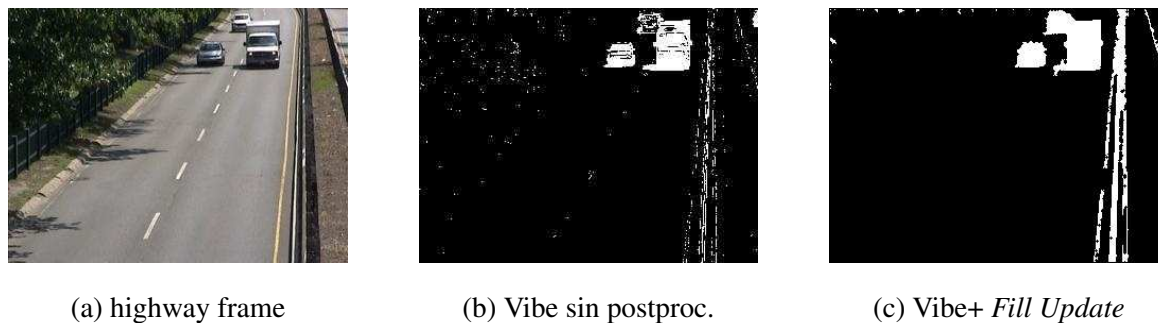
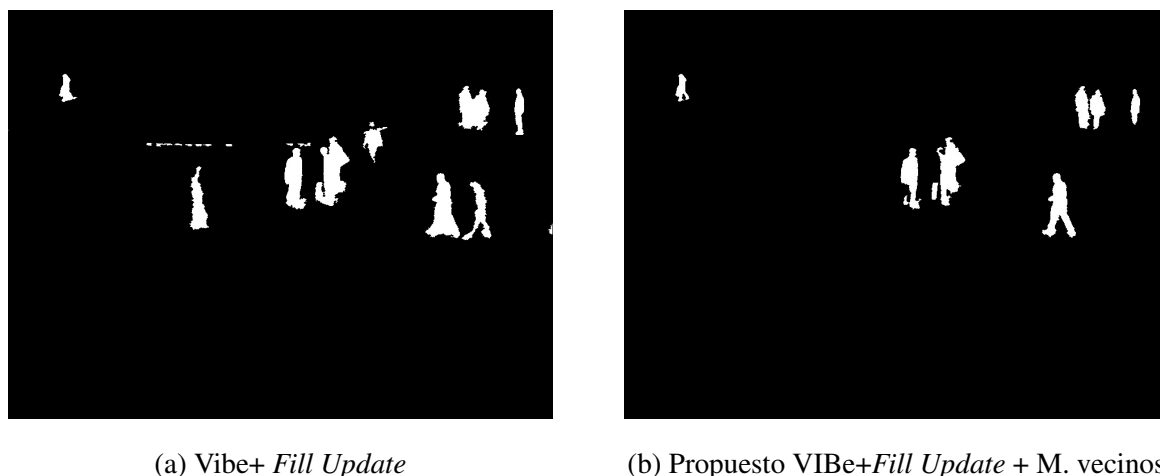
Figure 2: Efecto del módulo *Fill Update*

Figure 3: Ejemplo de fondo dinámico



#### 4 AMPLIACIÓN DEL MODELO DE FONDO DEL PÍXEL POR MOVIMIENTOS

Para mejorar las situaciones de movimientos en la imagen en donde el píxel actual se clasifica *background* en el modelo propio o en el modelo de alguno de sus vecinos, se agregó a la función de clasificación que lo verifica. Sea  $Q = \{M_0, M_1, M_{p_t}, \dots, M_k\}$  el conjunto de  $k$  modelos de los vecinos y el modelo propio del píxel  $p_t$ , se clasifica este píxel, usando la Ec. 2 con cada modelo de fondo en  $Q$ , y se suma la cantidad de veces que se el píxel clasifica como *background* en cada modelo de  $Q$ . Luego, para la clasificación final se determina si el valor de esta suma es mayor o igual que un valor definido #VEC como:

$$ClasificarconVecinos(p_t, Q) = \begin{cases} 1 & \text{si } \sum_{j=0}^k Clasificar(p_t, Q_j) \geq \#VEC \\ 0 & \text{en otro caso} \end{cases} \quad (3)$$

En este caso aumenta la probabilidad de coincidir el píxel procesado con el modelo de fondo de algunos de los vecinos o el propio. Esta consideración flexibiliza tener en cuenta modelos

de vecinos en dirección vertical, horizontal, diagonal, lo que se puede asociar a movimientos del fondo en esas direcciones frecuentemente observados en los videos inestables. El hecho de que se agrupan en bloques de modelos asociados a un píxel permite realizar una compensación, en parte del movimiento. Considerando que, en general, el movimiento es constante y leve, un punto correspondiente a un píxel tiende a moverse alrededor de su eje dentro de un radio delimitado, el cual está directamente relacionado al grado de movimiento que presente la cámara. Con los corrimientos en la imagen, es muy probable que coincida con el modelo de algún vecino, pero no con varios simultáneamente, con lo cual un valor de #VEC=1, resultó adecuado en las pruebas realizadas. En la Figura 3 (b) se muestra la segmentación con este método del mismo *frame* con #VEC= 1, tomando los 8 modelos de vecinos de alrededor de cada píxel. Se puede observar que desaparecen las figuras fantasmas más rápidamente, ya que no se encuentran en la segmentación como también que el movimiento de la cinta sobre el borde de la vereda desaparece. Además, al clasificar más píxeles como fondo en la zona de bordes, la operación de cierre resulta más apropiada, no rellena entre brazos y piernas como sucedía en la Figura 3 (a).

#### 4.1 Evaluación de la precisión del algoritmo

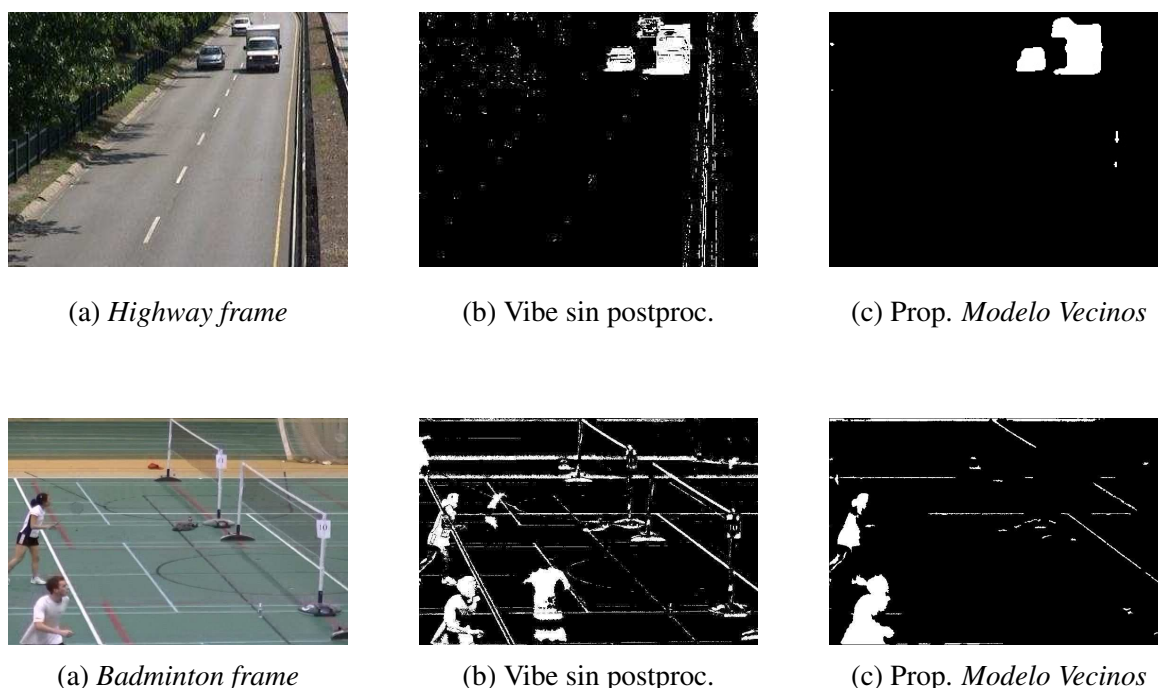
Para realizar un estudio de la eficacia del algoritmo, se utilizaron los *data-sets* provistos por (Goyette et al., 2012), en la cual se dispone del *ground-truth* para cada secuencia. Se eligieron tres secuencias bien características del contexto de video-vigilancia con diferente cantidad de *frames* cada una. Se usaron dos videos de la base *Baseline*, denominadas *Highway*, y *Office* de 1700 y 2048 *frames*, respectivamente. Además se usaron 4 ejemplos de la base *Camara Jitter: Badminton, Traffic, Sidewalk y Boulevard* de 1150, 2500, 1200 y 1570 *frames*, respectivamente, donde el movimiento de cámara es mayor. Las imágenes *ground truth* han sido rotuladas en todas las bases de videos con los siguientes valores *Static*:0, *Shadow*:50, *Non-ROI*: 85, *Unknown*: 170 y *Moving*: 255. Los rótulos *Moving* y *Static* se asocian a los píxeles con el estado de movimiento o no, respectivamente. El rótulo *Shadow* en este trabajo se considera como *Static*, para comparar con métricas de otros métodos mencionados, que tampoco incluyen la detección de sombras. Todos los píxeles cercanos a los bordes fueron rotulados como *Unknown* lo cual permite tener mejores evaluaciones en casos donde el estado *background* o *foreground* no resulta claro. El rótulo *Non-ROI* descarta zonas de la imagen que no son de interés para el análisis de las aplicaciones. En esta sección se presentan las métricas resultantes del algoritmo sobre los distintos *data-sets* analizados. El conjunto de pruebas se realizó sobre una computadora con un procesador AMD A6 GHz y una placa gráfica HP Graphics 2.7 Hz. La implementación fue realizada en Visual C++.

Casos	Propuesto		ViBe sin post.		ViBe		GMM-Z		GMM-KTP	
	Pr	Rc	Pr	Rc	Pr	Rc	Pr	Rc	Pr	Rc
Highway	0.94	0.85	0.90	0.54	0.92	0.85	0.92	0.89	0.91	0.65
Office	0.96	0.87	0.97	0.61	0.97	0.79	0.93	0.51	0.99	0.36
<b>Prom. Baseline</b>	0.95	0.86	0.93	0.57	0.94	0.82	0.92	0.70	0.95	0.50
Badminton	0.94	0.70	0.56	0.57	0.57	0.79	0.62	0.71	0.92	0.54
Boulevard	0.73	0.60	0.44	0.49	0.45	0.74	0.43	0.79	0.62	0.62
Sidewalk	0.94	0.37	0.29	0.30	0.55	0.49	0.35	0.51	0.28	0.53
Traffic	0.78	0.80	0.63	0.70	0.53	0.80	0.52	0.73	0.68	0.56
<b>Prom. C. Jitter</b>	0.84	0.61	0.48	0.51	0.52	0.70	0.48	0.68	0.62	0.56

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

Existen diferentes métricas evaluación de un método respecto de la detección correcta de *background* y *foreground*. En general, se calculan las siguientes variables, TP: número de verdaderos positivos, FN: número de falsos negativos y FP: número de falsos positivos, para obtener valores de métricas ampliamente utilizadas como *Precision* :  $TP / (TP+FP)$  y *Recall*:  $TP / (TP+FN)$ , para la sustracción de fondo (Goyette et al., 2012). En este caso el resultado es mejor, cuanto más cercana a 1 es la métrica.

Figure 4: Efecto del uso de Modelo de los vecinos



En la Tabla 1 se presentan los resultados de estas métricas comparando contra las técnicas de sustracción de fondo basadas en *Mixture of Gaussians* (MoG), y ViBe tradicional con y sin postprocesamiento. Se puede observar, que en los dos videos de *Baseline* se mejora la métrica *Precision* y *Recall* en promedio con el método propuesto. Esto indica que se corrige en promedio la cantidad de falsos positivos y falsos negativos. En todos los casos, las métricas en promedio del método propuesto, mejoran respecto a ViBe sin postprocesamiento. En particular,

si se toma la métrica *Precision* promedio, en el caso de *Camera Jitter*, mejora notablemente, un 61% respecto de ViBe con postprocesamiento, disminuyendo la cantidad de falsos positivos. Sin embargo, la métrica *Recall* en este caso, baja 12% en promedio, indicando el aumento de falsos negativos, esto ocurre principalmente cuando no rellena los objetos, por falta de cierre de los mismos. Cabe destacar que los videos de *Baseline*, presentan movimientos leves, mientras que en *Camera Jitter* el movimiento de cámara es muy pronunciado. Aún así, en ambos casos, incluso en secuencias complejas con objetos que se mueven constantemente, se acelera la absorción de píxeles fantasmas, reduciendo efectivamente los falsos positivos; como también el fondo dinámico, o los movimientos de cámara, mientras que el algoritmo base de ViBe presenta deficiencias en este aspecto. En la Figura 4, se usó #VEC=1, con 8 modelos de vecinos horizontales al píxel, se puede observar que se compensa el movimiento de cámara, ya que desaparece la línea de la carretera (falsos positivos). En el caso del video *Badminton*, el movimiento de cámara tiende a ser vertical, por lo que se usaron 8 modelos de vecinos en dirección vertical al píxel. Se puede notar que con ViBe aparece nuevamente, la figura fantasma de un jugador, como también líneas de gran espesor de la cancha que no deberían aparecer ya que es parte del fondo. En el método propuesto, las líneas se reducen notablemente, sin usar posprocesamiento de la salida. Los resultados obtenidos utilizando modelos de vecinos horizontales o verticales, fueron en todos los casos mejores según las métricas obtenidas, que si se consideraban todos los modelos de vecinos de alrededor al píxel. En todos los casos, se sugiere usar #VEC=1, es decir, generalmente coincide con el modelo propio del píxel o con el modelo de uno de sus vecinos. El tiempo de procesamiento utilizando #VEC=1 con 8 modelos de vecinos no se ve afectado notablemente, incluso en una versión mejorada a la propuesta en este trabajo, se puede utilizar técnicas de procesamiento en paralelo en una versión implementada en GPU, ya que cada etapa del procesamiento de búsqueda en los modelos de vecinos es trivialmente paralelizable; es decir, cada píxel puede ser procesado de forma independiente.

## 5 CONCLUSIONES

Se presentó un algoritmo de sustracción de fondo basado en el método ViBe extendiendo el uso del modelo de fondo al concepto de grilla de modelos de vecinos al píxel, para realizar una compensación de videos inestables por causa de movimiento presente en las cámaras, fondo dinámico, movimientos repentinos de objetos, etc. La variante no necesita un postprocesamiento de la salida como en los métodos convencionales y mejora la tasa de detección *Precision* respecto a ellos. El aumento de comparaciones de un píxel en su clasificación, aumenta la probabilidad de coincidir con el modelo de fondo de algunos de los vecinos en distintas conexiones establecidas de forma vertical, horizontal o diagonal, según los corrimientos frecuentes de cámara. Una ventaja de este procesamiento, es que no es necesario realizar un preprocesamiento de estabilización de cámara lo que representaría un elevado costo computacional, indeseable para el uso en sistemas de tiempo real. En futuros trabajos, puede automatizarse la obtención de la dirección frecuente de los movimientos de cámara procesando los primeros *frames*, utilizando por ejemplo un algoritmo tradicional de seguimiento de flujo óptico, y de esta manera usar modelos de vecinos en la dirección que indique el flujo de movimiento de cada video. Las pruebas realizadas sobre *data-set* con videos rotulados, muestran que los resultados son aceptables, y mejoran la tasa *Precision*, sin perjudicar la tasa *Recall* significativamente. También puede implementarse una versión que procese en paralelo la búsqueda en cada uno de los modelos vecinos al píxel. El algoritmo tiene algunas dificultades para discriminar objetos similares al fondo (problema denominado de 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.

## REFERENCES

- Barnich O. y Van Droogenbroeck M. Vibe: a powerful random technique to estimate the background in video sequences. *IEEE Int. Conference Acoustics, Speech and Signal Processing*, pages 945–948, 2009.
- Barnich O. y Van Droogenbroeck M. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6):1709–1724, 2011.
- Benezeth Y., Jodoin P.M., Emile B., Laurent H., y Rosenber C. Review and evaluation of commonly-implemented background sub. algorithms. *19th Int. Conf. on Pattern Rec.(ICPR)*., pages 1–4, 2008.
- Brutzer S., Höferlin B., y Heidemann G. Evaluation of background subtraction techniques for video surveillance. *Computer Vision and Pattern Recog.(CVPR)*, pages 1937–1944, 2011.
- David Schreiber M.R. Gpu-based non-parametric background subtraction for a practical surveillance system . *IEEE 12th Int.Conf. on Comp. Vision Workshops*, pages 870–877, 2009.
- Gervasoni L., Damato J., Barbuzza R., y Vénere M. Un método eficiente para la sustracción de fondo en videos usando gpu. *Enief 2014, Revista Mecánica Computacional*, pages 1721–1731, 2014.
- Goyette N., Jodoin P.M., Porikli F., Konrad J., y Ishwar P. Changedetection. net: A new change detection benchmark dataset. *Conf. on Comp. Vision and Pattern Rec. Workshops (CVPRW)*, pages 1–8, 2012.
- Piccardi M. Background subtraction techniques: a review. *Systems, man and cybernetics, 2004 IEEE international conference on*, 4:3099–3104, 2004.
- Schick A., Bauml M., y Stiefelhagen R. Improving foreground segmentations with probabilistic superpixel markov random fields. *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 27–31, 2012.
- Vacavant A., Chateau T., Wilhelm A., y Lequière L. A benchmark dataset for outdoor foreground/background extraction. *Computer Vision-ACCV 2012 Workshops*, pages 291–300, 2013.
- Van Droogenbroeck M. y Paquot O. Background subtraction: experiments and improvements for vibe. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 32–37, 2012.
- Yin Li Guijin Wang X.L. Three-level gpu accelerated gaussian mixture model for background subtraction. *Proc. SPIE 8295, Image Processing: Algorithms and Systems X; and Parallel Processing for Imaging Applications II*, 8295, 2012.
- Zivkovic Z. Improved adaptive gaussian mixture model for background subtraction. *Proceedings of the Int. Conf. on Pattern Recognition (ICPR)*, 2:28–31, 2004.