Congr. Int. en Ing. Electrónica. Mem. ELECTRO, Vol. 40, pp. 169 - 174, Oct 2018, Chihuahua, Chih. México
http://electro.itchihuahua.edu.mx/memorias_electro/MemoriaElectro2018.zip
ISSN 1405-2172

DETECCIÓN DE CAMBIOS DRÁSTICOS DE ESCENA MEDIANTE FUNCIÓN PERCEPTUAL EN UN ALGORITMO DE DETECCIÓN DE MOVIMIENTO

Manuel Alberto Chávez Salcido, Mario I. Chacón-Murguía, Juan A. Ramírez Quintana
Tecnológico Nacional de México/Instituto Tecnológico de Chihuahua
División de Estudios de Posgrado e Investigación
Av. Tecnológico 2909, Tecnológico, 31200 Chihuahua, Chih.
6275211978, 614202000 ext. 112
machavezs@itchihuahua.edu.mx, mchacon@ieee.org

RESUMEN.

El presente trabajo consiste en una propuesta de mejora desarrollada sobre un algoritmo existente denominado APPSA, el cual se encuentra enfocado a la detección de objetos dinámicos a partir del análisis de secuencias de video. Dicho algoritmo presenta como principal ventaja la capacidad de adaptar sus parámetros de acuerdo a las condiciones presentes en el video bajo análisis. La propuesta de mejora se encuentra basada en una función perceptual denominada *Block Mean Hash*, mediante la cual es posible detectar cambios drásticos de escena presentes en el video analizado. Esta modificación permite generar un algoritmo más robusto al detectar de forma más eficiente y rápida un cambio drástico de escena, pudiendo entregar, por ende, mejores resultados de segmentación de los objetos en movimiento.

Palabras Clave: detección de movimiento, cambio drástico de escena, block mean hash

ABSTRACT.

This work describes the improvement of an existing algorithm called APPSA, which is focused on the detection of dynamic objects from the analysis of video sequences. The main advantage of this algorithm is the ability to adapt its parameters according to the conditions present in the video under analysis. The proposed improvement is based on a perceptual function called Block Mean Hash. Through this function, it is possible to detect drastic changes in the scene present in the analyzed video. This modification makes it possible to generate a more robust algorithm by detecting a drastic change of scene more efficiently and quickly, delivering better segmentation results for moving objects.

Keywords: motion detection, scene drastic change, block mean hash

1. INTRODUCCIÓN

La detección de movimiento consiste en detectar los cambios de posición que un objeto ha sufrido relativos a su alrededor. El movimiento puede ser detectado mediante magnetismo, vibración, radio frecuencia, dispositivos infrarrojos, y métodos ópticos como videocámaras entre otros.

Para la detección de movimiento mediante el uso de dispositivos de video, el cual es el método que aboca este trabajo, se requiere procesar las imágenes o cuadros de video entrante haciendo uso de algoritmos que generen como respuesta la salida deseada, en el caso en particular, ésta es una imagen segmentada de los objetos dinámicos.

Estos algoritmos se basan en la implementación de diversas técnicas, algunas de las más populares se basan en conseguir modelar el fondo del video, cuadro de referencia, y lo comparan con cada cuadro de video entrante. De esta manera, las diferencias existentes entre el cuadro de referencia y el cuadro de entrada mostrarían en teoría los objetos dinámicos. La metodología anterior también se le conoce como modelado de fondo.

Para abordar el problema de detección de objetos dinámicos, se han propuesto diferentes metodologías debido a que existen diversas situaciones críticas que dificultan dicha tarea, tales como ruido en la imagen, vibraciones o movimientos en cámara, uso de cámaras con ajustes automáticos, cambios de iluminación, camuflaje, efecto fantasma o *ghosts* debido la presencia de objetos dinámicos durante la inicialización de los cuadros de referencia, sombras de los objetos dinámicos, fondos dinámicos como cuerpos de agua u hojas de árbol en movimiento, entre otras. Por lo cual, es complejo desarrollar una metodología que se desempeñe correctamente en todas las situaciones anteriores. Éstas metodologías o técnicas se han desarrollado con ayuda de diversos modelos matemáticos y algoritmos de aprendizaje de máquina.

Tal como lo contempla Guzmán [1], dentro de las técnicas más simples se plantea obtener el modelo de fondo mediante algoritmos tales como filtro de promedio, filtro de mediana o variaciones de los dos último [2] Otras técnicas se basan en el uso de modelos estadísticos, que se dividen en paramétricos y no paramétricos. Los modelos paramétricos, hacen uso de funciones de densidad de probabilidad con ciertos parámetros, variando estos parámetros se obtienen diferentes funciones, con distintas varianzas o tamaños que sirven para dar una aproximación al modelo de fondo. Entre estos modelos se encuentran los modelos Gaussianos y modelos de mezclas Gaussianas [3], [4]. Mientras que los modelos no paramétricos, son modelos estadísticos cuya distribución de probabilidad no se ajusta a una distribución conocida, tales como la distribución normal, uniforme, exponencial, logarítmica, entre otras. Dentro de éstos se puede considerar el caso del algoritmo Kernel Density Estimation [5], [6]. Así también existen técnicas que realizan una combinación de modelos estadísticos paramétricos y no paramétricos que generan modelos y/o proponen el uso de

otro tipo de distribuciones estadísticas, como en el caso de los algoritmos ViBe, Sift flow, [7], [8]. También, se pueden encontrar técnicas de transformación de dominio, cuyo principio de funcionamiento se basa en separar el fondo y los objetos dinámicos mediante la ayuda de una transformación de dominio. Para esto, se acude al uso de técnicas que permite realizar diferentes transformaciones como; Wavelet, Fast Fourier Transform, Hadamard [9], [10]. Otras técnicas se apoyan en modelos basados en redes neuronales artificiales, los cuales son modelos matemáticos inspirados en el comportamiento de las redes neuronales biológicas, uno de éstos métodos es el denominado APPSA [11], el cual modela dos fondos del video que analiza a partir de dos redes neuronales tipo SOM.

A pesar de la gran variedad de soluciones propuestas, no se ha logrado un algoritmo lo suficientemente robusto para tratar todas las condiciones que se pueden presentar al segmentar objetos dinámicos en escenas de video. Uno de estos problemas es el cambio drástico de escena, el cual se presenta frecuentemente en sistemas de seguridad de circuito cerrado, en donde de forma repentina, el cuadro de video puede cambiar totalmente de escenario.

El algoritmo que se ha modificado es APPSA, cuyas características más importantes son la capacidad de auto-adaptabilidad de acuerdo a las condiciones de dinamismo presentes en la escena analizada, por lo que es capaz de entregar resultados de segmentación certeros. Sin embargo, al analizar el comportamiento de APPSA ante cambios drásticos de escena, podía presentar fallas en la detección del mismo, por lo cual los resultados de segmentación mermaban.

A continuación, se describe de forma general el funcionamiento del algoritmo APPSA y la solución propuesta a la detección de cambio drástico de escena.

2. ALGORITMO APPSA.

El algoritmo APPSA trabaja de acuerdo al diagrama presentado en la Figura 1.

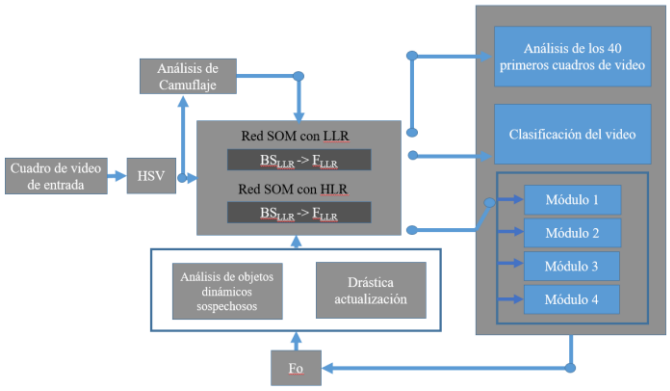

Figura 1. Diagrama de funcionamiento de APPSA.

El funcionamiento de APPSA consiste en segmentar los objetos dinámicos que se observen en una secuencia de video, la segmentación es presentada mediante una máscara binaria *Fo(x,t)*, donde *x* representa las coordenadas espaciales de un pixel en el tiempo *t*. La máscara binaria de segmentación, consiste en una imagen de iguales dimensiones a la de los cuadros del video analizado y en la cual, los objetos dinámicos se presentan en color blanco, mientras que el fondo se muestra en color negro. Para conseguir esta segmentación, APPSA se basa en 2 modelos de fondo que generan 2 posibles segmentaciones de objetos dinámicos, una segmentación por cada fondo, de las cuales sólo se considerará una como resultado final dependiendo del tipo de video bajo análisis. Estas condiciones son relativas al dinamismo del video, pudiendo ser escenas normales sin objetos dinámicos en el inicio de video, escenas con objetos dinámicos desde el inicio de análisis del video, escenas con fondos dinámicos como cuerpos de agua u hojas de árboles en movimiento en el fondo, videos captados con vibración o *jittering*, o videos provenientes de cámaras en movimiento como las llamadas cámaras PTZ. El proceso de obtención de la segmentación de objetos dinámicos se detalla a continuación.

Los cuadros de video analizados son convertidos al espacio de color HSV. Seguido de esto, los modelos de fondo se inicializan con el primer cuadro de video. En APPSA, se generan 2 modelos de fondo mediante el empleo de 2 redes neuronales de mapas auto-organizados tipo SOM, ambas redes SOM se diferencias una de la otra en la velocidad de actualización o aprendizaje con respecto al cuadro de video actual bajo análisis. Es decir, un modelo se actualiza a una velocidad mayor que el otro, dichas redes SOM presentan una conexión uno a uno entre sus neuronas y los pixeles de los modelos de fondo. De esta manera durante la actualización de cada una de las redes, el resultado de su aprendizaje refleja el modelo de fondo. Ambas redes definirán dos resultados de segmentación posibles, los cuales ayudarán a definir la segmentación final después de un análisis para determinar el tipo de dinamismo del video que se está analizando.

Posterior a la inicialización de los fondos, se realiza un análisis del histograma del canal H del primer cuadro de video para determinar si existe camuflaje en la escena, el camuflaje se da si se presenta un pico predominante en dicho histograma. Si el camuflaje existe, APPSA disminuye el valor de los umbrales que generan los dos resultados de segmentación final, esto con la finalidad de detectar objetos dinámicos que presenten colores o texturas similares al fondo de la escena.

El análisis de video determina si el mismo corresponde a escenas con bajo índice de dinamismo, presenta un dinamismo normal, cuenta con fondos dinámicos o si se trata de secuencias de video con vibraciones, o si provienen de cámaras PTZ. Cada una de las condiciones anteriores son tratadas en APPSA de una forma diferente mediante el empleo de 4 módulos. El análisis para clasificación del video consiste en obtener el promedio de los pixeles que fueron considerados como pertenecientes a objetos dinámicos durante los primeros 40 cuadros de video analizados provenientes de la segmentación dada al comparar cada cuadro de entrada con el modelo de

fondo de lenta actualización, determina el tipo de dinamismo presente en el mismo.

Una vez se ha clasificado el video, entra en función uno de los 4 módulos con los que se cuenta, los cuales realizan el cómputo de segmentación final de la forma más adecuada en función del tipo de escena.

Se cuenta también con 2 pasos posteriores al haber generado la máscara binaria, el primero es un análisis de objetos dinámicos sospechosos, que consiste en analizar las regiones de pixeles de la imagen que han sido continuamente clasificados como objetos dinámicos. Para esto, se hace uso de los puntos de interés generados mediante el algoritmo SURF, si la región detectada como objeto dinámico no contiene ningún punto de interés, la región es borrada del resultado de segmentación.

El segundo paso posterior al cálculo de la segmentación de objetos, consiste en verificar continuamente el promedio de la cantidad de pixeles que se han considerado objetos dinámicos, con la finalidad de determinar si ha ocurrido un cambio de escena drástico, lo que requeriría de una actualización también drástica de ambos modelos de fondo, además de una reclasificación de video. Para determinar el promedio de pixeles se recurre a la Ecuación (1), donde N_p es el promedio de pixeles detectados en el cuadro $Fo(x,t)$ como pertenecientes a un objeto dinámico, N y M son las dimensiones espaciales del cuadro de video.

$$N_p = \frac{\sum_{Fo(x,t)=1} 1}{MN} \quad (1)$$

APPSA basa la detección de cambio drástico de escena en un umbral de pixeles considerados objetos dinámicos a superar, dicho umbral corresponde al 65% del total de pixeles de los cuadros analizados. Si la condición anterior se ha cumplido, entonces se realiza una comparación entre los puntos de interés generados mediante el algoritmo SURF en el cuadro de video actual y los modelos de fondo, si no existe alguna correspondencia entre estos puntos de interés, entonces se clasifica como cambio drástico de escena, lo que conlleva a una actualización también drásticas de ambos modelos de fondo y una reclasificación de la escena.

Sin embargo, como se aprecia en la Figura 2, en APPSA, pueden existir cambios drásticos de escena y la cantidad de pixeles clasificados como objetos dinámicos no supera el umbral establecido, por lo cual la segmentación dada después de ese cambio drástico se ve afectada.

Para solventar el problema anterior, se propone hacer uso de una función perceptual denominada *Block Mean Hash* [12], la cual básicamente consiste en generar una huella o cadena binaria que defina la información presente en la imagen que se analiza. *Block Mean Hash* es una función de *hash* perceptual usada generalmente en aplicaciones de multimedia cuya finalidad es verificar la autenticidad o similitud de un objeto multimedia respecto a otro tomado como referencia. El funcionamiento de *Block Mean Hash* se detalla a continuación, así como su forma de empleo para verificar si existe cambio drástico de escena en el video analizado.

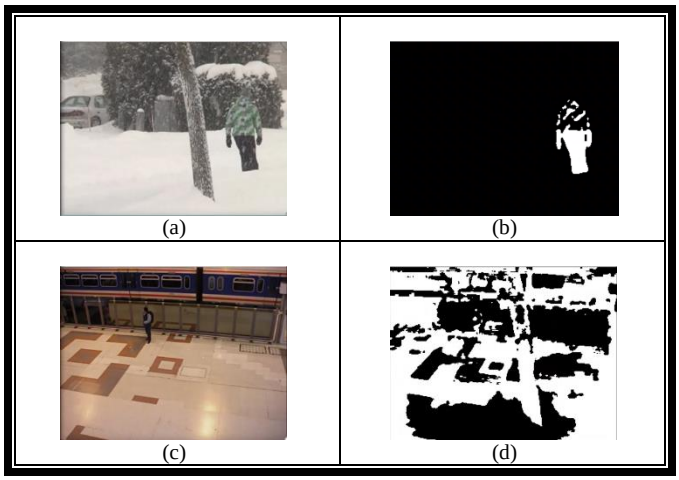

Figura 2. Cambio drástico de escena analizado en APPSA. La Figura (a) muestra el cuadro de video previo a un cambio de escena, (b) su cuadro de segmentación, (c) el nuevo cuadro de video y (d) la segmentación de este último.

3. FUNCIÓN PERCEPTUAL BLOCK MEAN HASH.

Las funciones de *hash* perceptual son un campo de investigación interdisciplinario, que involucra áreas como criptografía, marcas de agua digital y procesamiento digital de señales, entre otras. Estas funciones pueden ser vistas como huellas digitales del objeto al que se han aplicado. Dentro de este tipo de funciones, se da la clasificación de funciones *hash* codificadas y no codificadas, siendo la diferencia que, en el caso de las primeras se hace uso de una llave o clave secreta para encriptar y obtener el *hash* final, mientras que las segundas omiten el uso de esta llave.

Block Mean Hash corresponde a funciones codificadas, a pesar de ello se ha modificado la versión original para hacerla no codificada, dado que la codificación no es determinante para saber el grado de similitud entre imágenes.

Las funciones *hash* no codificadas presentan al menos las siguientes dos condiciones:

1.- Compresión. - mapea una entrada de una longitud finita de bits en una salida con una longitud fija de bits menor a la de entrada.

2-. Fácil computación. - dada una entrada, su *hash* debe ser fácilmente calculado.

Una imagen que es bit a bit idéntica a la imagen original se considera completamente auténtica (medida de autenticidad de 1.0). Mientras que, una imagen que no tiene nada en común con la imagen original, sería considerada no auténtica (medida de autenticidad de 0.0). Todas las demás imágenes serían parcialmente auténticas. Parcialmente auténtico es un concepto vagamente denotado y la medición de la autenticidad es subjetiva, por lo que cambia de un dominio a otro.

En 2006, Bian Yang, Fan Gu y Xiamu Niu propusieron un método basado en el valor medio de la función de *hash* de la imagen perceptiva [12]. En el mismo artículo se proponen cuatro métodos ligeramente diferentes entre sí. Dos de ellos incorporan una operación de rotación de imagen para mejorar la robustez del mismo, aunque esto aumenta significativamente la complejidad computacional de ambos métodos. Dado que se desea un método que sea capaz de ejecutarse de forma continua sin afectar los tiempos de procesamiento, se descartó el uso de realizar operaciones de rotación. De igual manera, en el método original, se utiliza el cifrado de los valores de *hash* utilizando una clave secreta, sin embargo, para el caso en particular, se ha omitido la parte de encriptación dado que la tarea a desempeñar no la requiere, únicamente se tiene contemplado realizar una comparación entre el *hash* obtenido de cuadros de video consecutivos.

El método implementado se describe a continuación:

1.- Se normaliza la imagen a un tamaño prestablecido de 256 por 256 pixeles mediante interpolación lineal exacta.

2.- La imagen obtenida del paso anterior se convierte a escala de grises mediante

3.- Denotando N_f como el tamaño final en bits de la cadena de *hash*, se divide la imagen en escala de grises en N_f cantidad de bloques no traslapados, todos ellos de igual tamaño, es decir, cada bloque contiene la misma cantidad de pixeles.

4.- Se obtiene la media de las intensidades de los pixeles de cada uno de los bloques en que fue dividida la imagen, con ello se obtiene un conjunto de N_f medias.

5.- Se obtiene la mediana del conjunto de medias.

6.- Se obtiene el *hash* de la imagen mediante:

$$h(A,i)=\begin{cases}0 & M_i < M_d \\ 1 & M_i \geq M_d\end{cases} \quad (2)$$

Siendo A la imagen a la que se le generará su función de *hash*, i es el i-ésimo bloque sobre el que se realiza la comparación de su media M_i respecto a la mediana M_d.

La Figura 3 muestra de forma gráfica el procedimiento anterior.

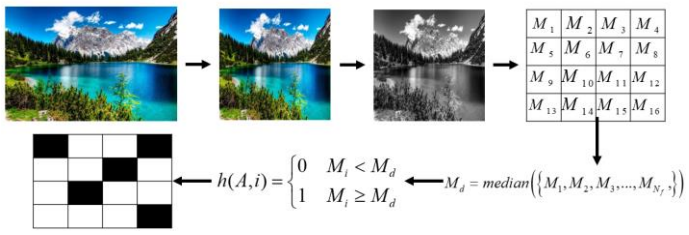

Figura 3. Procedimiento de *Block Mean Hash*.

El método anterior se aplica a cada cuadro de video de entrada y su *hash* respectivo se compara con el *hash* del cuadro anterior.

Para realizar el análisis de comparación entre el *hash* de cada imagen se recurre al uso de la distancia de Hamming, la cual es una medida de la diferencia entre dos cadenas de caracteres. La definición formal de la distancia de Hamming viene dada por:

$$D_h(h(A,i),h(B,j))=\sum_{i,j}^{N_f} h(A,i)\oplus h(B,j) \quad i\oplus j=\begin{cases}0 & i=j \\ 1 & i\neq j\end{cases} \quad (3)$$

Con base en lo anterior, la autenticidad o similitud de un cuadro de video respecto a otro, será alta cuando la distancia de Hamming sea un valor cercano a cero, si este es mayor a cero, será un indicativo de menor similitud. Sin embargo, los valores que son utilizados para determinar si un video presenta cambio drástico de escena son obtenidos mediante la diferencia entre los *hash* de cuadros consecutivos, que se puede interpretar como la derivada de la distancia de Hamming con un diferencial de tiempo igual a la unidad, definida en la Ecuación (4), en donde B es la imagen anterior a la imagen A en la secuencia de video. El hacer uso de la derivada de la distancia de Hamming se debe a que presenta mayor robustez ante cambios drásticos de escena. Los resultados obtenidos mediante esta técnica se discuten en la siguiente sección de resultados y conclusiones.

$$\frac{dD_h}{dt}=D_h(h(A,i))-D_h(h(B,i)) \quad (4)$$

4. RESULTADOS.

A continuación, se presentan los resultados obtenidos al aplicar y comparar mediante la distancia de Hamming, el *hash* de cuadros consecutivos de un video que presenta cambios drásticos de escena. Cada una de las escenas han sido generadas al concatenar un par de videos completamente diferentes uno respecto del otro, esto con la finalidad de detectar el cambio drástico que la concatenación conlleva. En cada una de las figuras que a continuación se presentan, se muestran dos gráficas, la primera ubicada en la parte superior de cada figura, corresponde a la distancia de Hamming entre las funciones de *hash* de cuadros consecutivos de la secuencia de video, mientras que la gráfica debajo de cada figura muestra la derivada de los datos graficados en la primera.

El valor de la derivada de la distancia de Hamming resulta ser más discriminante computacionalmente, ya que una distancia que represente un cambio drástico de escena, genera un pico de amplitud significativa, pero su derivada resulta tener de igual manera, un cambio drástico de valor positivo e inmediatamente un cambio drástico con un valor negativo. Esto además permite generar un método más robusto puesto que, si la detección de cambio drástico de escena se basara únicamente en la distancia de Hamming, se establecería un umbral a superar y se podría presentar el caso que se considere un cambio de escena inexistente, esto se puede apreciar en la Figura 4, la cual fue generada a partir de un video que presenta problemas de *jittering* y se observa que tiene distancias de Hamming con valores mayores a los generados por videos con escenas que presentan fondos dinámicos, Figura 5 o bajo dinamismo,

Figura 6. Sin embargo, al usar el valor de la derivada de la distancia de Hamming, se debe de cumplir 3 condiciones, la primera es que se debe de superar un umbral positivo, inmediatamente se debe de generar un valor negativo y este debe además superar un segundo umbral, de esta forma el método de cambio drástico de escena se vuelve más robusto.

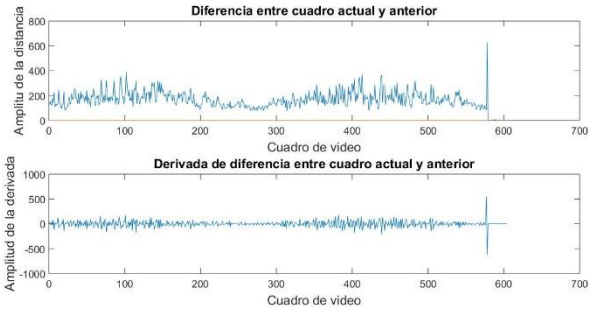

Figura 4. Distancia de Hamming y su derivada entre *hash* consecutivos en video con *jittering*.

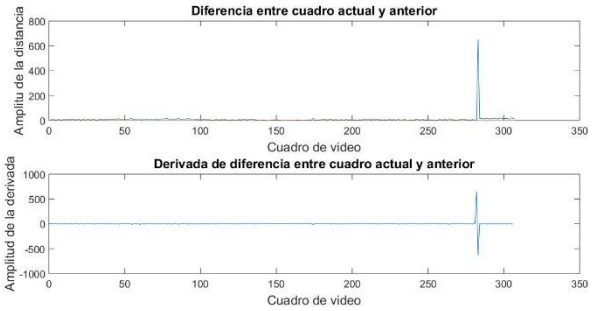

Figura 5. Distancia de Hamming y su derivada entre *hash* consecutivos en video con fondo dinámico.

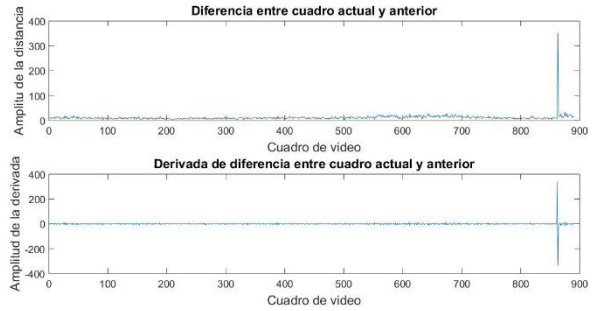

Figura 6. Distancia de Hamming y su derivada entre *hash* consecutivos en video con bajo dinamismo.

La Tabla 1, muestra los resultados al realizar 8 experimentos en los que se generaba un cambio drástico de escena. En esta tabla se muestran los valores de las amplitudes de la derivada de la distancia de Hamming obtenida al ocurrir un cambio de escena en un video con diferentes condiciones de dinamismo y de entorno, así también, se muestra el promedio de los 10 valores de la derivada de la distancia de Hamming previos al cambio de escena. Estos 10 valores fueron considerados con la finalidad de hacer evidente el gran cambio que se presenta al detectarse el cambio de escena y en lo posible tomarlos como información previa y compararla con la distancia del cuadro de video bajo análisis. Sin embargo, dado que el almacenar *S* cantidad de datos, obtener su promedio y compararlo con cada cuadro analizado conllevaría a un mayor costo de cómputo, lo que repercutiría en mayor tiempo de procesamiento del algoritmo, sólo se seleccionaron dos valores de umbral para determinar si existe un cambio drástico de escena.

Tabla 1. Características de la derivada de la distancia de Hamming

Tipo de Video	**Amplitud pico máximo positivo**	**Amplitud pico mínimo negativo**	**Media 10 valores antes de pico máximo positivo**
Fondo dinámico	647	-638	0.110
Cámara infrarroja	602	-591	-0.0395
Video con *jittering*	544	-626	-0.0089
Video con sombras	345	-335	-0.0047
Video con mal clima	389	-386	-0.0039
Video con objetos intermitentes	560	-436	4.649E-4
Video con alto dinamismo	438	-410	0.1485
Video con tomas nocturnas	567	-554	0.1891

Con base en los resultados de la Tabla 1**¡Error! No se encuentra el origen de la referencia.**, los valores de umbral de la derivada de Hamming a superarse fueron seleccionados como 200 y -200. Se desarrollaron un total de 17 experimentos, es decir 17 análisis de videos concatenados, contemplando aquéllos que contaran con fondo dinámico, videos con problemas de *jittering*, video provenientes de cámaras PTZ, videos de infrarrojos y videos con ambientes nocturnos, siendo en todos ellos detectado correctamente el cambio drástico de escena, es decir, la exactitud del método fue de 100%, por lo que se puede concluir, al añadir esta mejora en el algoritmo APPSA se ha generado un algoritmo más robusto ante escenarios donde se puede presentar este tipo de problemas, tal como sucede en el caso de cámaras de circuito cerrado en sistemas de video vigilancia. La Figura 7, muestra los resultados al aplicar la modificación propuesta al algoritmo APPSA, se aprecia que a diferencia de los resultados generados por APPSA original Figura 2; ante un cambio drástico de escena, los modelos de fondo son actualizados inmediatamente

un cuadro posterior al cambio drástico de escena, dando como resultado una mejor segmentación de los objetos dinámicos debido a que los modelos de fondo también se actualizan de forma drástica y se reclasifica el tipo de video presente. En APPSA original, algunos cambios de escena no son detectados, por lo que la segmentación final merma debido a que los modelos de fondo no se actualizan inmediatamente.

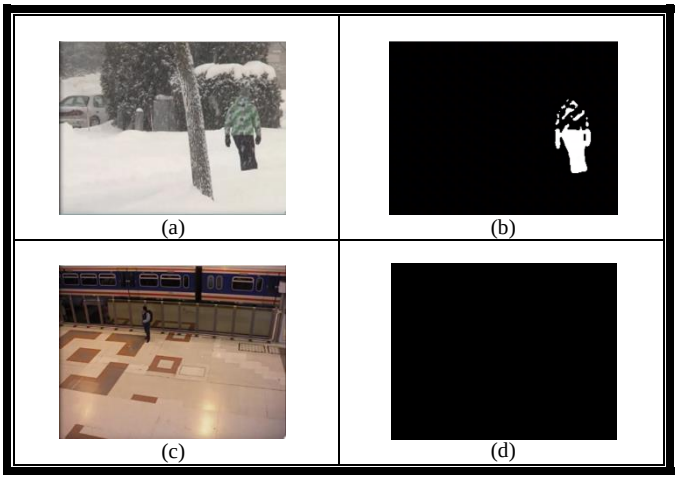

Figura 7. Cambio drástico de escena analizado mediante Block mean hash en APPSA. La Figura (a) muestra el cuadro de video previo a un cambio de escena, (b) su cuadro de segmentación, (c) el nuevo cuadro de video y (d) la segmentación de este último.

5. CONCLUSIONES.

La velocidad de procesamiento obtenida al ejecutar el algoritmo desarrollado en el lenguaje C++, mediante el empleo de la librería OpenCV, fue de 22 FPS aproximadamente en una computadora con el sistema operativo Windows 10 de 64 bits con un procesador Intel I5 6300 a 2.3 GHz y una memoria RAM de 8 Gb, mientras que el algoritmo original de APPSA implementado en el entorno de programación de Matlab y ejecutado en la misma computadora con las características anteriores, corre a aproximadamente 20 FPS. De esta manera, el tiempo de procesamiento no se ve afectado al implementar la mejora propuesta y al ejecutarse de forma continua, se genera un algoritmo más robusto en el análisis de detección de objetos dinámicos.

Así también, el algoritmo modificado se ejecutó en una tarjeta Raspberry Pi 3 con la finalidad de verificar la factibilidad de procesamiento en una computadora con bajas prestaciones de hardware, ya que esto permitiría generar sistemas de video vigilancia de bajo costo. La tarjeta Raspberry Pi 3 en que se ejecutó el algoritmo cuenta con el sistema operativo Stretch, con CPU Cortex-A53 a 1.2 GHz y 1 Gb de memoria RAM, dando como resultado una velocidad promedio de aproximadamente 3 FPS haciendo uso de entre el 25 y el 30% del CPU.

Como trabajo a futuro y basado en las gráficas de distancia de Hamming en videos con *jittering*, como se muestra en la Figura 4, se pretende realizar un análisis de los datos con la finalidad de determinar si existe una característica que permita discriminar si el video presenta *jittering* con una menor cantidad de cuadros de video analizados, esto generaría una mejor segmentación de este tipo de videos en un tiempo menor como se realiza actualmente, ya que APPSA requiere de al menos 40 cuadros para determinar el tipo de video y generar el mejor resultado de segmentación. De igual manera, se tiene contemplado hacer uso de programación multi-nucleo, con la finalidad de incrementar la cantidad de cuadros por segundo al ejecutarse en la tarjeta Raspberry Pi 3.

6. RECONOCIMIENTO.

Se agradece al Tecnológico Nacional de México/IT Chihuahua por el apoyo brindado para la realización de este trabajo bajo el proyecto 6418.18-P.

7. REFERENCIAS.

[1] A. Guzmán Pando, “Análisis y comparación de algoritmos de modelado de fondo de secuencias de video”, Tesis de maestria en ciencias, Instituto Tecnológico de Chihuahua, 2018.

[2] M.-H. Hung, J.-S. Pan, y C.-H. Hsieh, “A Fast Algorithm of Temporal Median Filter for Background Subtraction”, en *Ubiquitous International*, 2014, vol. 5, núm. 1, pp. 33–40.

[3] Z. Chen y T. Ellis, “A self-adaptive Gaussian mixture model”, en *Computer Vision and Image Understanding*, 2014, vol. 122, pp. 35–46.

[4] T. Bouwmans, F. El Baf, y B. Vachon, “Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey”, en *Recent Patents on Computer Science*, 2008, vol. 1, núm. 3, pp. 219–237.

[5] C. Spampinato, S. Palazzo, y I. Kavasidis, “A texton-based kernel density estimation approach for background modeling under extreme conditions”, en *Computer Vision and Image Understanding*, 2014, vol. 122, pp. 74–83.

[6] D. Giordano, S. Palazzo, y C. Spampinato, “Kernel Density Estimation Using Joint Spatial-Color-Depth Data for Background Modeling”, en *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 4388–4393.

[7] J. Dou y J. Li, “Modeling the background and detecting moving objects based on Sift flow”, en *Optik*, 2014, vol. 125, núm. 1, pp. 435–440.

[8] S.-I. Oh y H.-B. Kang, “A new object proposal generation method for object detection in RGB-D data”, *SAMI 2017 - IEEE 15th Int. Symp. Appl. Mach. Intell. Informatics, Proc.*, núm. 2, pp. 393–398, 2017.

[9] M. N. Al-Berr, M. A. M. Salem, H. M. Ebeid, M. F. Tolba, y A. S. Hussein, “Wavelet-enhanced detection of small/slow object movement in complex scenes”, en *Proceedings of 2016 11th International Conference on Computer Engineering and Systems, ICCES 2016*, 2017, pp. 172–180.

[10] J. Šoda, I. Kuzmanić, y I. Vujović, “Stabilising illumination variations in motion detection for surveillance applications”, en *IET Image Processing*, 2013, vol. 7, núm. 7, pp. 671–678.

[11] G. Ramírez-Alonso y M. I. Chacón-Murguía, “Auto-Adaptive Parallel SOM Architecture with a modular analysis for dynamic object segmentation in videos”, *Neurocomputing*, vol. 175, Elsevier, pp. 990–1000, 2016.

[12] B. Yang, F. Gu, y X. Niu, “Block mean value based image perceptual hashing”, en *International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2006*, 2006, pp. 167–170.