

Comparación de algoritmos de detección de cambios en videos

Santos Canul, U.¹, Bolón Méndez, G.², Moreno Sabido, M.³, Hernández López, F.⁴, Castillo Atoche, A.⁵

Fecha de recepción: 29 de junio de 2017 – Fecha de aprobación: 29 de agosto de 2017

RESUMEN

El presente trabajo se centra en la implementación y comparación de cuatro algoritmos de detección de cambios en videos: PBAS, LBAdaptiveSOM, DPWrenGABGS y MixtureOfGaussianV1BGS. Mediante la utilización de las librerías de OpenCV, se realizaron comparaciones de estos algoritmos usando las bases de datos CDNET y BMC con el fin de obtener las características y el desempeño de cada uno de ellos. Los videos de estas bases de datos contienen diversos retos (cambios de iluminación, sombras, movimiento de la cámara, entre otros), los cuales hacen difícil la detección de objetos y personas en movimiento. De acuerdo con los resultados obtenidos en las pruebas, el algoritmo PBAS es mejor que los otros algoritmos aquí analizados.

Palabras Clave: Detección de Cambios, Visión Artificial, Modelo de Fondo.

Comparison of video change detection algorithms

ABSTRACT

The present work focuses on the comparison of four video change detection algorithms: PBAS, LBAdaptiveSOM, DPWrenGABGS and MixtureOfGaussianV1BGS. Comparisons of these algorithms are made using the CDNET and BMC databases with OpenCV libraries, in order to obtain the characteristics and performance of each one. The videos in these databases contain several challenges (illumination changes, shadows, camera movement, among others), which make it difficult to detect moving objects and people. According to the experiments, the PBAS algorithm is better than the other algorithms analyzed in this work.

Keywords: Change Detection, Artificial Vision, Background Model.

¹ Estudiante de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Mérida

² Estudiante de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Mérida.

³ Profesor-Investigador de Tiempo Completo del Instituto Tecnológico de Mérida.

⁴ Profesor-Investigador. CONACYT – Centro de Investigación en Matemáticas A.C., CIMAT Unidad Mérida.

⁵ Profesor-Investigador en el área de Sistemas Embebidos y Procesamiento en Tiempo Real.

Nota: Este artículo de investigación es parte de Ingeniería–Revista Académica de la Facultad de Ingeniería, Universidad Autónoma de Yucatán, Vol. 21, No. 2, 2017, ISSN: 2448-8364.

INTRODUCCIÓN

La detección de cambios (CD) es un paso crucial en muchos sistemas de visión computacional; ha sido ampliamente estudiada desde 1990, y se ha utilizado en aplicaciones como la video vigilancia, el reconocimiento de objetos, la detección de intrusos, el seguimiento de objetos, el conteo de personas, entre otras cosas.

Muchos algoritmos han sido diseñados para la CD; en general comparten el mismo esquema: son evaluados en una sola base de datos y su buen desempeño depende de los parámetros con los que fueron entrenados o inicializados, sin embargo, dichos parámetros podrían no ser apropiados al evaluar los algoritmos en diferentes bases de datos.

El resultado de un algoritmo de CD es una imagen binaria, en la cual, regiones con valor 1 representan objetos en movimiento, y regiones con valor 0 representan el fondo, o viceversa. Una forma de obtener dicha imagen binaria es aplicando un umbral sobre la resta entre la imagen actual y un modelo de

fondo. En la Figura 1 se muestra un esquema general de la CD, y los pasos son descritos a continuación:

- *Inicialización del modelo de fondo*: En este paso se obtiene la parte estática de la escena o, en un sentido más general, todo lo que puede ser considerado como fondo dadas las características de la escena observada. Para calcular el modelo, una opción sencilla es tomar el promedio de un conjunto de los primeros N frames del video.
- *Detección de cambios*: En los siguientes frames, se procesa una comparación entre el frame actual y el modelo de fondo.
- *Actualización del modelo de fondo*: Durante este proceso, las imágenes se analizan con el fin de actualizar el modelo de fondo, el cual fue aprendido en la etapa de inicialización, con respecto a una tasa de aprendizaje. Un objeto que no se mueva durante mucho tiempo se debe integrar en el fondo¹.

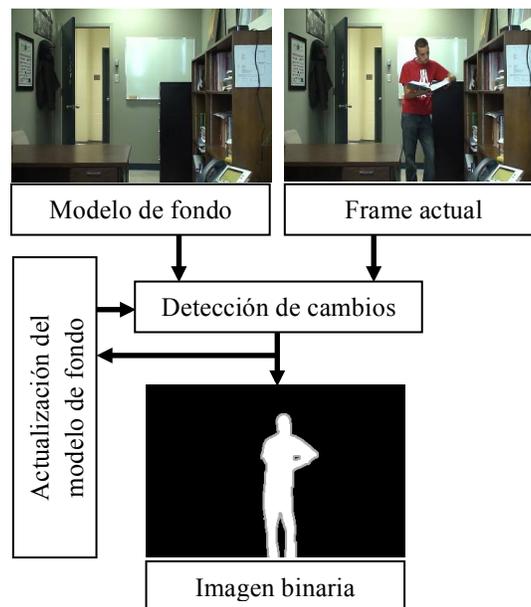


Figura 1. Esquema general de la detección de cambios.

¹ Background Subtraction, disponible en:

http://docs.opencv.org/trunk/d1/dc5/tutorial_background_subtraction.html.

En la actualidad existen bases de datos sobre la CD, las cuales permiten indagar cual método convendría utilizar dependiendo de las necesidades, sin embargo, estas no ofrecen una amplia gama de las situaciones en las que una cámara estaría expuesta.

Con este trabajo de investigación se pretende obtener una base comparativa de los mejores métodos para la detección de cambios, lo cual permitirá tener un mayor número de opciones a elegir. En este trabajo se compararon los siguientes algoritmos: PBAS, LBAdaptiveSOM, DPWrenGABGS y MixtureOfGaussianV1BGS, usando las librerías de OpenCV². Se realizaron las comparaciones usando las bases de datos CDNET y BMC, con el fin de obtener las características y el desempeño de cada uno de ellos.

Segmentación Adaptativa Basada en Píxeles (PBAS): La idea general de este método es realizar el proceso de comparación o resta entre la imagen actual y el modelo de fondo, de manera tal que sea comparada pixel a pixel en los canales RGB; esto hace que los pequeños cambios como los de iluminación, no sean tomados como un objeto en movimiento (Martin *et al.*, 2012). A continuación, se describen algunos de los procesos principales de este método.

El bloque de decisiones, como su nombre indica, al analizar una imagen con el modelo de fondo, compara los píxeles de la imagen actual uno por uno con su respectivo pixel en el modelo de fondo; entonces, por cada pixel, este bloque decide si el pixel pertenece al fondo $B(x_i)$, o es parte de un objeto en movimiento; esta decisión se realiza aplicando un valor de umbral $R(x_i)$ por cada píxel.

Luego, el modelo de fondo tiene que ser actualizado cada determinado tiempo, con el fin de agregar los cambios graduales del entorno en el fondo, por ejemplo, el movimiento de los árboles ocasionados por el viento. En este modelo, esta actualización depende de un parámetro de aprendizaje $T(x_i)$ por cada píxel.

La idea esencial y novedosa del enfoque PBAS, es que los valores de los umbrales por pixel, cambian dinámicamente con base en una estimación de la dinámica del fondo (que tan rápido y que tanto cambia el fondo). El objetivo de todos los métodos de segmentación de fondo es llegar a una decisión binaria si un píxel pertenece al primer plano o al fondo. Este proceso de decisión toma la imagen de entrada y la compara de alguna manera con un modelo de fondo. En este caso, el modelo de fondo $B(x_i)$ se define por un conjunto N de valores de pixel, observados recientemente:

$$B(x_i) = \{B_1(x_i), \dots, B_k(x_i), \dots, B_N(x_i)\} \quad (1)$$

Un píxel x_i se decide que pertenece al fondo, si su valor de píxel $I(x_i)$ está más cerca que un cierto umbral de decisión $R(x_i)$ a al menos un número $\#_{\min}$ de los valores de fondo

N . Por lo tanto, el resultado que define si un pixel pertenece a un objeto en movimiento, se calcula de la siguiente manera:

$$F(x_i) = \begin{cases} 1 & \#dist_{IBR} < \#_{\min} \\ 0 & \text{Otro caso,} \end{cases} \quad (2)$$

con $\#dist_{IBR} = \#\{dist(I(x_i), B_k(x_i)) < R(x_i)\}$.

² OpenCV (Open Source Computer Vision Library), disponible en: <http://opencv.org/>

En la parte de actualización del modelo de fondo, se crea un arreglo $D(x_i) = \{D_1(x_i), \dots, D_N(x_i)\}$ de distancias mínimas calculadas entre valor actual y el modelo del fondo como sigue:

$$d_m(x_i) = \min_k \text{dist}(I(x_i), B_k(x_i)) \quad (3)$$

$$R(x_i) = \begin{cases} R(x_i)(1 - R_{i/d}), & \text{si } R(x_i) > \bar{d}_m(x_i)R_s \\ R(x_i)(1 + R_{i/d}), & \text{otro caso,} \end{cases} \quad (4)$$

donde $R_{i/d}$ y R_s son parámetros constantes. La idea es que $R(x_i)$ se incremente en regiones dinámicas y se decremente en regiones estáticas.

$$T(x_i) = \begin{cases} T(x_i) + \frac{T_i}{\bar{d}_m(x_i)}, & \text{si } F(x_i) = 1 \\ T(x_i) - \frac{T_d}{\bar{d}_m(x_i)}, & \text{si } F(x_i) = 0, \end{cases} \quad (5)$$

donde T_i y T_d son parámetros constantes. Los valores que toma $T(x_i)$ quedan delimitados en el rango $T_l < T < T_u$, con T_l y T_u también parámetros constantes.

Redes neuronales para la sustracción de fondo (LBAdaptiveSOM): Este método propone realizar el proceso de modelado de fondo utilizando mecanismos de atención visual. La idea general es hacer un modelo de fondo, de tal manera que aprenda de forma auto-organizada, muchas variaciones del fondo, a partir de videos tomados de cámaras estáticas (Lucia *et al.*, 2008).

Para cada pixel a color, se construye un mapa neuronal de $n \times n$ vectores de pesos (con $n = 9$), entonces para una imagen I con N_F el número de filas y N_C el número de columnas, habrá un mapa neuronal A con $n \times N_F$ filas y $n \times N_C$ columnas, el cual representa el modelo de fondo inicial.

El método tiene una etapa de entrenamiento, en donde se utiliza un número K_F de frames para que la red neuronal aprenda; este

Entonces, esta distancia mínima se asigna al $D_k(x_i)$, por lo tanto, se genera una historia de distancias mínimas y el promedio $\bar{d}_m(x_i) = (1/N) \sum_k D_k(x_i)$ es una medida de que tan dinámico es el fondo. Con esta medida, el umbral $R(x_i)$ se actualiza con la siguiente relación:

Luego, el parámetro de aprendizaje es actualizado de la siguiente manera:

número debe elegirse con cuidado, ya que depende de los primeros frames del video que de preferencia no contengan objetos en movimiento.

Sea $C = (c_1, c_2, \dots, c_{n^2})$ el modelo para un pixel (x, y) , en la etapa de actualización del modelo de fondo, primero se busca el vector c_m que está más cercano en distancia Euclidiana al vector p_t del pixel en el frame actual; ambos vectores se trabajan en el espacio de color HSV. La distancia queda definida de la siguiente manera:

$$d(c_m, p_t) = \min_{i=1, \dots, n^2} d(c_i, p_t) \leq \epsilon \quad (6)$$

donde

$$\epsilon = \begin{cases} \epsilon_1, & \text{si } 0 \leq t \leq K_F \\ \epsilon_2, & \text{si } t > K_F, \end{cases}$$

con ϵ_1 y ϵ_2 dos constantes de valor pequeño. Luego, se actualizan los valores que están más cercanos a la posición (\bar{x}, \bar{y}) del vector c_m en el modelo de fondo con la siguiente ecuación:

$$A_t(i, j) = (1 - \alpha_{i,j}(t))A_{t-1}(i, j) + \alpha_{i,j}(t)p_t(x, y) \quad (7)$$

para $i = \bar{x} - \lfloor n/2 \rfloor, \dots, \bar{x} + \lfloor n/2 \rfloor$, $j = \bar{y} - \lfloor n/2 \rfloor, \dots, \bar{y} + \lfloor n/2 \rfloor$ y $\alpha_{i,j}(t) = \alpha(t)w_{i,j}$, donde $w_{i,j}$ son pesos Gaussianos en la vecindad de $n \times n$ del vector c_m . El factor de aprendizaje $\alpha(t)$ se define como

$$\alpha(t) = \begin{cases} \alpha_1 - t \frac{\alpha_1 - \alpha_2}{K_F}, & \text{si } 0 \leq t \leq K_F \\ \alpha_2, & \text{si } t > K_F, \end{cases} \quad (8)$$

donde α_1 y α_2 son constantes predefinidas tal que $\alpha_2 \leq \alpha_1$.

Mezcla de Gaussianas para el modelo de fondo (MixtureOfGaussianVIBGS): Este método modela a cada píxel de fondo mediante una mezcla de K_G distribuciones Gaussianas (K_G es un número pequeño de 3 a 5); por cada Gaussiana, se obtiene un valor binario (fondo u objeto), entre los cuales se determina cual es el más factible; en caso de que los valores resultantes de alguna Gaussiana no sean factibles, esta se reemplaza por otra nueva Gaussiana. Este método es muy preciso, pero contraproducente en la capacidad computacional, ya que entre más Gaussianas se agreguen, el algoritmo tarda más tiempo (KaewTraKulPong *et al.*, 2011).

El método realiza el proceso de modelado de fondo de la siguiente manera: la probabilidad de que un determinado píxel tenga un valor

de x_t en el tiempo t se puede escribir como:

$$p(x_t) = \sum_{k=1}^{K_G} w_k \eta(x_t; \mu_k, \Sigma_k) \quad (9)$$

donde w_k es el parámetro de peso del k^{th} componente Gaussiano, $\eta(x_t; \mu_k, \Sigma_k)$ es la distribución normal del k^{th} componente, donde μ_k es la media y $\Sigma_k = \sigma_k^2 I$ es la matriz de covarianza del k^{th} componente, con I la matriz identidad.

Las distribuciones K_G se ordenan con base al valor de aptitud Σ_k/σ_k y las primeras distribuciones B se utilizan como modelo de fondo de la escena, donde B se estima como:

$$B = \operatorname{argmin}_b \left(\sum_{j=1}^b w_j > T \right) \quad (10)$$

con T un umbral que representa la fracción mínima del modelo de fondo. Entonces, un píxel es clasificado como objeto en movimiento, si está a más de 2.5 veces a la desviación estándar de cualquiera de las B distribuciones Gaussianas. El primer componente Gaussiano que coincida con el valor de prueba se actualizará mediante las siguientes ecuaciones:

$$\hat{w}_k^{t+1} = (1 - \alpha) \hat{w}_k^t + \alpha \hat{p}(\omega_k | x_{t+1}) \quad (11)$$

$$\hat{\mu}_k^{t+1} = (1 - \alpha) \hat{\mu}_k^t + \rho x_{t+1} \quad (12)$$

$$\hat{\Sigma}_k^{t+1} = (1 - \alpha) \hat{\Sigma}_k^t + \rho (x_{t+1} - \hat{\mu}_k^{t+1})(x_{t+1} - \hat{\mu}_k^{t+1})^T \quad (13)$$

$$\rho = \alpha \eta(x_{t+1}; \hat{\mu}_k^t, \hat{\Sigma}_k^t) \quad (14)$$

$$\hat{p}(\omega_k | x_{t+1}) = \begin{cases} 1 & \text{si } \omega_k \text{ es la primera} \\ & \text{coincidencia del} \\ & \text{componente gaussiano} \\ 0 & \text{de otra manera.} \end{cases} \quad (15)$$

Donde ω_k es el k^{th} componente Gaussiano y

α es un factor constante de aprendizaje. Si

ninguna de las distribuciones K_G coincide con el valor de píxel, el componente menos probable se reemplaza por una distribución con el valor actual como su media, una varianza inicialmente alta y un parámetro de peso bajo.

Detector de personas PFinder (DPWrenGABS): Este método construye el modelo de fondo localizando a las personas y realizando un modelo 2-D de las mismas; esto hace que solamente las personas sean tomadas como objetos en movimiento (Wren et al., 1997).

A diferencia de los demás métodos, éste utiliza un sistema especializado en detectar personas llamado “Pfinder” (Person Finder), el cual es un sistema en tiempo real para rastrear a la gente y para interpretar su

$$P(\vec{o}) = \frac{1}{z} \exp[-(\vec{o} - \vec{\mu})^T \Sigma^{-1} (\vec{o} - \vec{\mu})] \quad (15)$$

con $z = (2\pi)^{3/2} |\Sigma|^{1/2}$. Además, tiene una etapa de aprendizaje, en donde se requiere que el video no contenga personas. Después de modelar la escena, Pfinder encuentra variaciones en los nuevos frames a partir de este modelo, usando la distancia de Mahalanobis. Si hay un cambio con suficiente tamaño, entonces Pfinder construye un componente conectado modelando a una persona. El componente conectado se define como:

$$s_k(x, y) = \begin{cases} 1 & (x, y) \in k \\ 0 & \text{otro caso} \end{cases} \quad (17)$$

donde k representa el índice del componente conectado. Luego, para actualizar el modelo, se aplica la siguiente relación:

$$\vec{\mu}_t = \alpha \vec{y} + (1 - \alpha) \vec{\mu}_{t-1} \quad (18)$$

con \vec{y} el vector del píxel en el frame actual y α un factor de aprendizaje. Esto le permite al método ser robusto a cambios de iluminación no tan drásticos y a problemas cuando un objeto estático es desplazado (una maleta, una silla, un libro, etc.).

comportamiento. Funciona a 10Hz en una computadora SGI Indy estándar, y ha funcionado de manera confiable en miles de personas en muchas ubicaciones físicas diferentes. El sistema utiliza un modelo estadístico multi-clase de color y forma para obtener una representación 2-D de cabeza y manos en una amplia gama de condiciones de visión.

Este método asocia un vector de promedios $\vec{\mu}$ y una matriz de covariancia Σ a una región de píxeles o componente conectado en cada frame, entonces la probabilidad de que un píxel \vec{o} pertenezca a dicho componente se calcula de la siguiente manera:

En los siguientes apartados se describen las configuraciones que se realizaron a los cuatro métodos descritos anteriormente para realizar las pruebas con las bases de datos ChangeDetection y BMC, describiendo cada una de estas bases de datos. Luego, se presentan los resultados de forma cualitativa y cuantitativa. Finalmente, se presentan las conclusiones.

METODOLOGÍA

Como primer paso, se configuraron los programas de cada método en una PC con sistema operativo Windows 7.0, procesador Intel i5 con 8 GB de RAM, usando Visual Studio C++ y la librería OpenCV versión 3.1.

El código fuente de *PBAS*³, fue configurado para probarse en la base de datos del ChangeDetection del 2012. Por otro lado, los códigos fuente de *DPWrenGABS*, *LBadaptiveSom* y *MixtureOfGaussian-VIBGS*⁴, fueron configurados para la base de datos del BMC.

³ The Pixel-Based Adaptive Segmenter, disponible en:
<https://sites.google.com/site/pbassegmenter/download-1>

⁴ Disponibles en: <https://github.com/andrewssobral/bgslibrary>

En cada uno de los programas, se agregaron instrucciones para leer correctamente los directorios de los videos de las correspondientes bases de datos. También se agregaron instrucciones para guardar de forma adecuada las imágenes binarias resultantes de cada método, con el fin de utilizar un programa que evalué los resultados, y de esta manera obtener medidas cuantitativas de que tan precisos son cada uno de los métodos. A continuación, se describen cada una de las bases de datos.

Esta base de datos está compuesta por videos reales y sintéticos que representan escenarios urbanos. Los videos fueron adquiridos a partir de cámaras estáticas. Cuenta con videos CGI (imágenes generadas por computadora), con los que es posible tener un conjunto de datos necesarios para la evaluación y comparación de los algoritmos. La base de datos puede ser descargada en la página de los autores⁵. En la Tabla 1 se muestran las categorías, el número de videos y la cantidad de frames que se procesaron para cada uno de los métodos.

Background Modelling Challenge (BMC):

Tabla 1. Base de datos BMC.

Categoría	Videos	Frames
<i>learningmode</i>	5	7,460
<i>syntheticvideos</i>	5	7,463
Total	10	14,923

ChangeDetection.net (CDNET): Base de datos con videos realistas, capturados por cámara sin CGI. Estos videos han sido seleccionados con la intención de cubrir varios desafíos y representar escenas muy comunes, tanto en interiores como en exteriores⁶. La base de datos del 2012 del CDNET incluye los siguientes desafíos: fondo dinámico, trepidación de la cámara, movimiento de objetos intermitente, las sombras y videos tomados de cámaras térmicas. Por otro lado, la base de datos del 2014 de CDNET incluye todos los videos de

2012, además de otros adicionales con las siguientes dificultades: condiciones climáticas difíciles, velocidad de fotogramas baja, la adquisición por la noche, capturas de cámaras de video vigilancia “PTZ” y la turbulencia del aire. Cada conjunto de datos va acompañado de una segmentación exacta y de la anotación de las áreas de cambio y movimiento para cada trama de video. En la Tabla 2 se puede apreciar la cantidad de categorías, videos y frames que se procesaron para cada método.

Tabla 2. Base de datos CDNET.

Categoría	Videos	Frames
<i>badWeather</i>	4	20,872
<i>Baseline</i>	4	6,021
<i>cameraJitter</i>	4	6,392
<i>dynamicBackground</i>	6	18,822
<i>intermittentObjectMotion</i>	6	18,607
<i>lowFramerate</i>	4	9,372
<i>nightVideos</i>	6	14,789
<i>PTZ</i>	4	8,602
<i>Shadow</i>	6	16,907

<i>Thermal</i>	5	21,065
<i>Turbulence</i>	4	15,672
Total	53	157,121

⁵ <http://bmc.iut-auvergne.com/>

⁶ ChangeDetection.net, disponible en: <http://changedetection.net/>

RESULTADOS

En la Figura 2, se muestra una comparación cualitativa entre los cuatro métodos presentados en este trabajo, usando uno de los videos de la categoría *CameraJitter* de la base de datos CDNET. Este video se caracteriza por la constante vibración de las cámaras, lo cual provoca que, en los

resultados de cada método, se observen errores en el fondo. Esto es un punto muy importante por resaltar, ya que cada método construye el modelo de maneras distintas. El Ground Truth sirve de referencia para saber qué tan acertado es el método, como se muestra en la Figura 2.

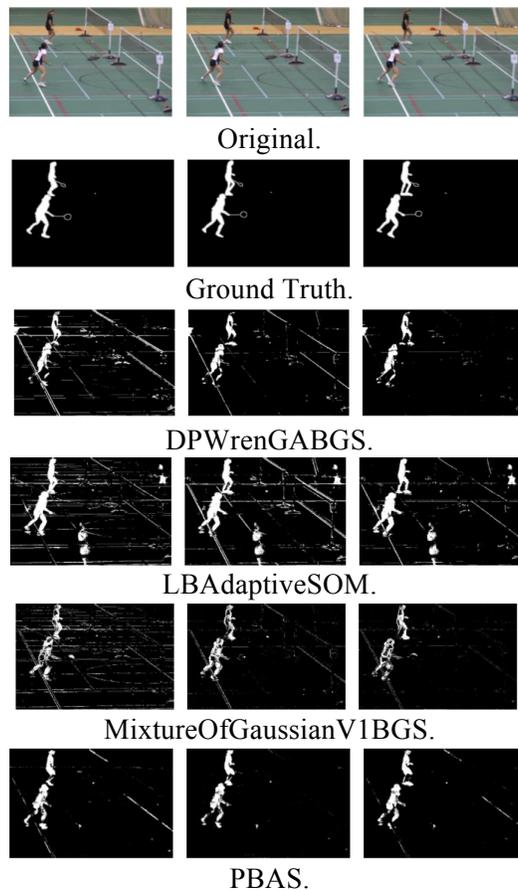


Figura 2. Resultados de los métodos usando uno de los videos de la categoría *CameraJitter*.

El método DPWrenGABGS detecta a las personas, pero éste genera unas líneas a lo largo de la escena, así como también detecta la cancha de bádminton; esto se debe a que este método se centra más en la detección de personas, sin embargo, este método no es completamente robusto a situaciones como lo

es una cámara en vibración.

El método LBAdaptiveSOM detecta a las personas casi a la perfección, sin embargo, éste tiene las líneas en la escena más notorias con respecto a los otros métodos; esto se debe a que este método tiene un mapa neuronal

para los datos del modelo de fondo; esto hace que detecte de manera fiable a las personas, pero a consecuencia tiene datos extras generados por la vibración de la cámara, lo cual genera muchas detecciones erróneas, y esto se puede notar en la secuencia de frames del método.

El método MixtureOfGaussian detecta a las personas, pero con huecos, así como también algunas líneas y parte de la cancha; esto se debe a la forma de realizar la máscara binaria, los pixeles dentro del contorno de las personas son parecidos al del fondo ocasionando que los confunda con el fondo; esto indica que el método no es robusto en este tipo de situaciones.

El método PBAS detecta a las personas y genera algunas líneas en la escena, sin embargo, las personas tienen secciones que no detecta, cuando estas deberían estar detectadas; esto se debe al problema de la vibración de la cámara, ya que al evitar incluir las líneas divisoras de la cancha de bádminon como parte de lo que se mueve, ocasiona que las secciones de las personas en contacto con estas líneas se vuelvan parte del fondo cuando no debería ser así; esto indica que puede afrontar situaciones como éstas, pero no en su totalidad.

A continuación, se muestran las siete medidas que se obtuvieron utilizando el código disponible⁷. Dicho código, mide cuantos verdaderos positivos (TP), falsos positivos

(FP), verdaderos negativos (TN) y falsos negativos (FN) tienen cada uno de los resultados de los métodos con respecto al Ground Truth en cada una de las bases de datos. Cada medida está definida de la siguiente manera:

- Exhaustividad:
 $RE = TP / (TP + FN)$
- Especificidad:
 $SP = TN / (TN + FP)$
- Taza de falsos positivos:
 $FPR = FP / (FP + TN)$
- Taza de falsos negativos:
 $FNR = FN / (TP + FN)$
- Porcentaje de clasificaciones incorrectas:
 $PWC = 100 * (FN + FP) / (TP + FN + FP + TN)$
- Precisión:
 $PR = TP / (TP + FP)$
- F-Measure:
 $FM = (2 * PR * RE) / (PR + RE)$

Los resultados de los algoritmos usando la base de datos CDNET se pueden observar en la Tabla 3. Los resultados usando la base de datos BMC se pueden observar en la Tabla 4, y los resultados mostrados en la Tabla 5, son tomados de la evaluación realizada por (Andrews *et al.*, 2014). En estas tablas, tres de los métodos han sido renombrados, quedando de la siguiente manera: DPWrenGABGS (DP), MixtureOfGaussian (Mix) y LBAdaptiveSOM (LBA).

Tabla 3. Resultados generales– CDNET.

Met.	RE	SP	FPR	FNR	PWC	FM	PR
DP	0.55	0.94	0.05	0.44	6.58	0.40	0.38
Mix	0.25	0.98	0.01	0.74	4.59	0.34	0.21
LBA	0.78	0.86	0.13	0.21	13.5	0.30	0.37
PBAS	0.72	0.95	0.04	0.27	5.00	0.64	0.62

Tomando en cuenta que el algoritmo con un valor alto en las medidas *RE*, *FM* y *PR* tiene mejor desempeño, y observando los valores de la Tabla 3 y la gráfica de la Figura 3, se puede notar que el algoritmo con mejor

desempeño es el PBAS, el cual tiene los mejores resultados en las medidas *FM* y *PR*, mientras que el método con el peor desempeño es el MixtureOfGaussian.

De la misma manera en que se evaluaron los algoritmos en el CDNET, aquí también se compararon los algoritmos con base en las medidas *RE*, *FM* y *PR*. Como se puede observar en la Tabla 4 y la Tabla 5, hay una diferencia significativa entre estas medidas;

esto se debe a que los resultados mostrados en la Tabla 4, fueron obtenidos usando una de las dos categorías del BMC (la categoría *learnig*).

⁷ <http://wordpress-jodoin.dmi.usherb.ca/code/>

Tabla 4. Resultados generales– BMC.

et.	RE	SP	FPR	FNR	PWC	FM	PR
DP	0.71	0.994	0.0053	0.28	0.68	0.65	0.63
Mix	0.40	0.994	0.0052	0.59	0.84	0.39	0.37
LBA	0.85	0.990	0.0091	0.14	0.98	0.59	0.65
PBAS	0.91	0.998	0.0018	0.08	0.23	0.74	0.81

Tabla 5.1 Resultados generales – BMC 2014. Obtenida de (Andrews *et al.*, 2014).

Met.	RE	PR	FM
DP	0.795	0.922	0.853
Mix	0.793	0.912	0.847
LBA	0.838	0.907	0.867
PBAS	0.923	0.852	0.885

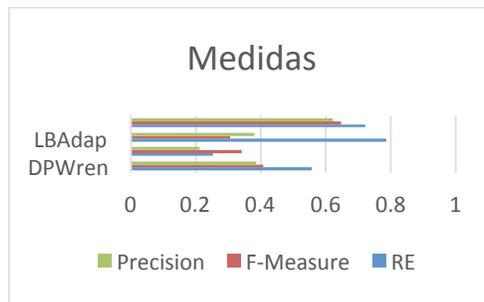


Figura 3. Resultados de las medidas generales actuales CDNET.

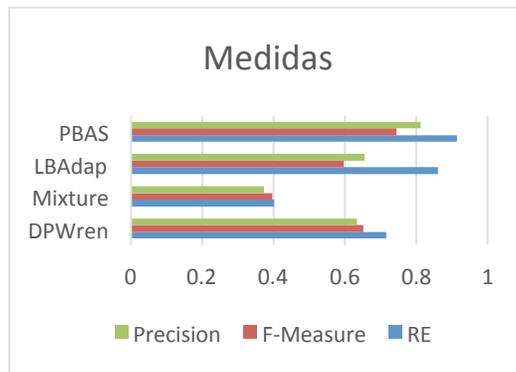


Figura 4. Resultados de las medidas generales actuales BMC.

En la Figura 4, se muestra una gráfica de los resultados de la Tabla 4, en donde se puede observar que el método PBAS cuenta con el mejor desempeño.

CONCLUSIONES

Toda la información obtenida demuestra que el método más óptimo para las necesidades generales es el PBAS, ya que en las medidas generales sobresalió más en comparación a los demás métodos mencionados, mientras que muy al contrario, el método MixtureOfGaussians demostró que no funciona adecuadamente en la mayoría de las categorías analizadas.

Por otra parte, los métodos LBAadaptiveSOM y DPWrenGABGS son muy parecidos en

cuanto a resultados, diferenciándolos en gran medida por el porcentaje de fallos, ya que el método LB cuenta con el mayor número de fallos; a pesar de que estos tres métodos no sobresalieron como el PBAS, no quiere decir que sean malos, únicamente denota que los métodos de modelado de fondo de éstas necesitan ajustes o modificaciones que les permitan desempeñarse mejor en entornos difíciles.

También se pudo observar que entre las categorías, el PTZ es un problema el cual no ha sido abordado por ninguno de estos métodos, ya que el entorno es un conjunto de diferentes problemáticas a resolver; algunas de ellas son el movimiento constante de la cámara, sombras y cambios de iluminación.

REFERENCIAS

Sobral, A., & Vacavant, A. (2014). A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122, 4-21.

Maddalena, L., & Petrosino, A. (2008). A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Transactions on Image Processing*, 17(7), 1168-1177.

Hofmann, M., Tiefenbacher, P., & Rigoll, G. (2012, June). Background segmentation with feedback: The pixel-based adaptive segmenter. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012 IEEE Computer Society Conference on (pp. 38-43). IEEE.

KaewTraKulPong, P., & Bowden, R. (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. *Video-based surveillance systems*, 1, 135-144.

Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7), 780-785.

Este documento debe citarse como: Santos Canul, U., Bolón Méndez, G., Moreno Sabido, M., Hernández López, F., Castillo Atoche, A., (2017). **Comparación de algoritmos de detección de cambios en videos**. *Ingeniería, Revista Académica de la FI-UADY*, 21-2, pp. 11-21, ISSN 2448-8364.