

Analysis and Trends on Moving Object Detection Algorithm Techniques

A. Guzman-Pando, and M. Chacon-Murguía, *Senior Member, IEEE*

Abstract—This paper presents a survey on dynamic object detection in video sequences. Many methods have been proposed during several years that intent to solve the problems of dynamic background, jittering, illumination changes, camouflage, among other. The survey is intended to provide a recent analysis of these methods to highlight the main characteristics of each approach. The survey is based on published paper from 2013 up to date. 77 methods were found, and they are described and analyzed in this survey. In addition, the survey proposes a complete classification of the methods based on the type of technique used to achieve the detection of dynamic objects. It also evaluates the limitations of each approach and the tendencies regarding the most used techniques, color space, datasets, hardware, programming language and processing time. One of the most popular approaches correspond to advanced statistical models, and artificial neural networks with 28% and 22% respectively. The analysis reports 11% of use of GPUs, most of them found in the neural network approaches. Regarding datasets, there are authors that report the use of three datasets the most popular is Change Detection. The most used software to implement the methods is MATLAB and the RGB color space, the most employed.

Index Terms—Background subtraction methods, Moving object detection algorithms, Analysis and trends.

I. INTRODUCCIÓN

LOS algoritmos de detección de movimiento de objetos en secuencias de video son un factor importante en la implementación de diversas aplicaciones de visión por computadora tales como: sistemas de vigilancia, navegación autónoma, identificación automática de objetivos, reconocimiento de actividad humana, monitoreo, entre otras [1]–[4]. No obstante, la detección de movimiento no es una tarea sencilla debido a los cambios que se pueden suscitar en un ambiente real o no controlado. En la práctica aparecen situaciones críticas que complican este proceso. Kentaro Toyama [5] identificó 10 problemas en el campo de video vigilancia, tales como: las vibraciones de cámara, el cambio de iluminación gradual durante el día, el cambio repentino de luz, fondos dinámicos como árboles u ondulaciones en el agua, camuflaje, entre otros. Recientemente, Thierry Bouwmans [6] extendió esta lista a 13 situaciones críticas, siendo estas: ruido, vibraciones o movimientos en cámara, ajustes automáticos en

cámaras PTZ, cambios de iluminación, *bootstrapping*, camuflaje, *foreground aperture*, objetos del fondo movibles, objetos insertados al fondo, efecto fantasma, fondos dinámicos, objetos en primer plano estáticos y sombras. Estas situaciones críticas han motivado la investigación para proponer nuevos métodos que puedan lidiar con estos problemas, con el fin de obtener una detección del objeto en movimiento lo más robusta posible. Algunos métodos se basan en modelar un fondo de referencia, donde se obtiene un escenario sin objetos en movimiento o dinámicos, que luego es restado del cuadro actual para detectar a los objetos en movimiento.

Entre las técnicas pioneras y mayormente conocidas para realizar detección de movimiento se encuentran el filtro de promedio [7], el filtro de mediana [8][9] y los modelos estadísticos, que se dividen en paramétricos y no paramétricos. Dentro de los paramétricos se encuentran los modelos Gaussianos simples (GS) [10] y mezclas Gaussianas o *Gaussian mixture models* (GMM) [11][12]. Por su parte, en los modelos no paramétricos se encuentra el algoritmo *kernel density estimation* (KDE)[13]. Estas técnicas continúan siendo utilizadas debido a que no requieren una gran cantidad de parámetros, son fáciles de implementar y su consumo de recursos es bajo. No obstante, últimamente se han desarrollado nuevos métodos que pueden afrontar una mayor cantidad de situaciones críticas en los videos. El aumento de técnicas o métodos enfocados a detección de movimiento demanda un análisis para conocer cuál es la tendencia actual en esta área de investigación y establecer cuáles técnicas resultan mejores para cada situación.

En la literatura existen algunos esfuerzos para analizar algoritmos de detección de movimiento. Massimo Piccardi [14] propuso un método de clasificación de algoritmos basado en velocidad de ejecución, requerimientos de memoria y precisión. Algunos de los algoritmos considerados por Piccardi fueron los métodos: *running gaussian Average* (RGA), *temporal median filter* (TMF), GMM y KDE, entre otros. Y. Benezeth et al. [15] presentaron una evaluación y comparación (cuantitativa y cualitativa) de los métodos GS, GMM, MinMax y KDE a manera de medir su rendimiento y robustez en aplicaciones reales con video secuencias que representan diferentes retos. Norbert Buch [16] propuso una clasificación de algoritmos de detección de movimiento en sistemas de monitoreo de tráfico urbano con base en su rendimiento y su posible aplicación en tiempo real, un aporte interesante que evalúa los métodos en escenarios urbanos y carreteras.

Thierry Bouwmans, en [6] ofreció una clasificación completa de algoritmos, dividiéndolos en dos categorías principales, modelos tradicionales y modelos recientes, en ellas se da una amplia información para elegir el mejor método de detección dependiendo de los problemas a solucionar, tales

This work was supported by Tecnológico Nacional de Mexico/I.T. Chihuahua under grant 5162.19-P.

A. Guzmán-Pando, and M. I. Chacon-Murguía are with the Instituto Tecnológico de Chihuahua, Ave. Tecnológico 2909 Chihuahua, México, aguzman@itchihuahua.edu.mx., mchacon@ieeee.com.

como: cambios de iluminación, fondos dinámicos, sombras, ajustes automáticos de cámara, ruido en la imagen, etc. Recientemente, Jaya S. Kulchandani y Kruti J. Dangarwala [17] describieron y presentaron las tendencias en el área de algoritmos de detección de objetos dinámicos y proporcionaron una tabla comparativa.

Las principales contribuciones de este trabajo con respecto a las publicaciones mencionadas son las siguientes. Se presenta un análisis de la literatura actual respecto a algoritmos de detección de movimiento, detallando sus características más relevantes y proponiendo una clasificación completa del tipo de técnica que utilizan para realizar la detección de movimiento. Se describen las tendencias de diseño de algoritmos de los últimos años, sobre temas cómo; ¿Cuál es el tipo de técnica más utilizada?, ¿Qué cantidad de bases de datos son utilizadas para validar su desempeño?, ¿Qué base de datos es la más empleada? Además, se revisa que tipo de equipo de cómputo, espacio de color y lenguaje de programación es empleado, así como el tiempo de procesamiento reportado.

Para el desarrollo del análisis se realizó una búsqueda en la literatura considerando publicaciones del año 2013 a la actualidad. Decidimos considerar este periodo de tiempo para concentrarnos en las técnicas más actuales. Información sobre algoritmos anteriores a esta fecha se pueden encontrar en otros trabajos [6][16]. En total se encontraron 77 algoritmos, los cuales son presentados y analizados en este trabajo.

En la Sección II se presenta la clasificación de algoritmos por tipo de técnica y se describen las características más relevantes de cada uno de los algoritmos. La Sección III presenta un análisis general donde se identifican problemas abiertos y tendencias de cada una de las técnicas con base en el tipo de técnica más utilizada, tiempo de procesamiento, espacio de color, base de datos, hardware y lenguaje de programación. Finalmente, en la Sección IV se presentan las conclusiones.

II. CLASIFICACIÓN POR TIPO DE TÉCNICA

La clasificación de los algoritmos encontrados en la literatura fue inspirada en la clasificación por tipo de técnica propuesta por Thierry Bouwmans en [6]. La clasificación de Bouwmans, es una de las más completas encontradas en la literatura debido a que engloba una gran cantidad de técnicas, con más de 100 algoritmos, los cuales son divididos en dos categorías principales: los modelos tradicionales y los recientes, además, se proporcionan subcategorías donde se indica el tipo de técnica específica utilizada con cada método.

La Tabla I, muestra la clasificación propuesta. Se puede observar que la columna “categoría” engloba de una manera general el método utilizado para la detección de movimiento y las subcategorías muestran de una manera más específica el tipo de técnica. A continuación, se da una descripción de cada una de las categorías mostradas en la Tabla I y se resume la información más relevante de cada uno de los diferentes enfoques.

A. Modelos Básicos

Son algoritmos que usan técnicas sencillas de implementar que no requieren una gran cantidad de parámetros y su consumo de recursos es bajo, entre ellas se encuentran las técnicas basadas en promedio, en mediana y diferencia de cuadro. Entre

los modelos básicos, está el filtro de promedio recursivo también llamado aproximación por filtro de promedio o *running average filter* (RAF).

TABLA I
CLASIFICACIÓN POR TIPO DE TÉCNICA DE LOS ALGORITMOS ANALIZADOS EN EL PERIODO 2013-FEBRERO 2018

CATEGORÍA	TÉCNICA	ARTÍCULOS
Básicos (8)	Modelos basados en promedio	[18],[19],[20]
	Modelos basados en mediana	[21],[22],[23],[24]
	Modelo basado en diferencia de cuadro	[25]
Combinados (4)	Modelos Gaussianos, redes auto organizadas y un módulo de detección de cambios de iluminación	[26]
	Aprendizaje de múltiples vistas, campo aleatorio de Markov y filtro de mediana temporal	[27]
	Estimación de densidad de núcleo, flujo óptico y diferencias	[28]
	Flujo óptico, mezcla de Gaussianas y texturas	[29]
Estadísticos (2)	Modelo de mezclas Gaussianas auto-adaptivas	[30]
	Estimación de densidad de núcleo y textura	[31]
Redes Neuronales (17)	Red neuronal celular de tiempo discreto, DTCNN	[32]
	Mapas auto-organizados, SOM	[33],[34],[35],[36]
	Respuesta neuronal mixta, NeRM	[37]
	Redes neuronales convolucionales, CNN	[38],[39],[40],[41]
Modelos Estadísticos Avanzados (22)	Máquinas de Boltzman restringidas	[42],[43],[44]
	Redes neuronales tipo Weightless	[45]
	Redes neuronales tipo Hopfield	[46]
	Función de base radial	[47]
	Modelo no paramétrico mediante patrones locales binarios	[48]
	Modelos mediante extractor visual de fondo, ViBe	[49]
	Modelo no paramétrico basando en entropía por bloques	[50],[51]
	Algoritmo de estabilidad de características adaptivas, SoAF	[52]
	Mezcla de gaussianas y módulo de espacio de color	[53]
	Modelos SBM y STBM (Aquí se proponen dos métodos y son evaluados individualmente)	[54]
	Peso de la muestra para modelado de fondo, WeSamBE	[55]
	Múltiples modelos de fondo, MBSV 0	[56]
	Modelo basado en superpíxeles, MBS	[57]
	Métodos basados en algoritmo AMBER	[58]
Segmentación adaptativa basada en píxeles, PBAS	[59],[60]	
Características de luminiscencia y color, LRGB	[61]	
Mejora al modelo no paramétrico SURF	[62]	
Modelo no paramétrico basado en <i>Sift flow</i>	[63]	
Aproximación estocástica	[64]	
Aprendizaje compartido basado en mezcla de Gaussianas	[65]	
Gaussianas globalmente espaciadas e incertidumbres	[66]	
Modelo adaptivo mediante patrones binarios, SuBSENSE	[67]	
Modelo basado en corte de grafo y mezcla de Gaussianas	[68]	
Modelos Difusos (2)	Correlograma de núcleo difuso	[69]
	Diferencia de histograma de color difuso	[70]
	Análisis de componentes principales robustos, RPCA	[71]
	Análisis de componentes principales robustos y mínimos cuadrados ponderados iterativamente, IRLS	[72]
Modelos de subespacios (8)	Mejora al método análisis de componentes principales robustos en línea, OR-PCA	[73]
	Análisis de componentes principales robustos en línea a través de optimización estocástica	[74]
	Matriz de bajo rango en línea, COROLA	[75]
	Representación de bajo rango en línea, DECOLOR	[76]
	Componentes principales robustos regularizados, TVRPCA	[77]
	Descomposición basada en Superpíxeles	[78]
	Patrones locales binarios y diccionarios, PAWCS	[79]
	Análisis local de saliente	[80]
Diccionarios de aprendizaje (6)	Libro de códigos basado en superpíxeles	[81]
	Modelado a través de diccionarios de aprendizaje, BMTDL	[82]
	Mantenimiento histórico de píxeles, BREW-DLHPM	[83]
Modelos de transformación de dominio (3)	Modelo basado en codificación esparcida, K-SVD	[84]
	Onduletas para reducir variaciones de luz, BMW	[85]
	Memoria de diferencia de cuadros y onduletas	[86]
Modelos tensores robustos (2)	Enfoque basado en onduletas y patrones locales binarios	[87]
	Tensor de flujo con mezclas Gaussianas particionadas, FTSG	[88]
Otros (3)	Tensor bayesiano robusto por medio de factorización, BRTF	[89]
	Modelo basado en programación genética, IUTIS	[90]
	Modelo basado en controlador PID	[91]
	Modelo basado en unidad de codificación de árbol, CTU	[92]
77	Algoritmos en total	[93]

Este algoritmo, es relativamente acertado para modelar el fondo cuando la iluminación es constante. Sin embargo, en existencia de cambios repentinos de iluminación o fondos dinámicos, resulta en un alto número de falsos positivos. El problema principal radica en que se asume un parámetro de aprendizaje igual para todos los píxeles de la imagen.

Por lo que en [18] se propone una adaptación, en la cual el parámetro de aprendizaje es diferente para cada píxel. Otro método [19] usa dos modelos de fondo generados mediante aproximación por filtro de promedio con velocidades de adaptación distinta; baja y alta respectivamente. Además, el parámetro de ganancia de compensación se ajusta dependiendo del grado de iluminación. En [20], se combinan operaciones morfológicas con el algoritmo recursivo RAF para mejorar la detección de los objetos en movimiento. Por su parte, los modelos recursivos basados en mediana, utilizan la información actual y posterior de los cuadros de video para generar la estimación del fondo [8]. Estos modelos son conocidos como filtros de mediana debido a que el fondo calculado en cada iteración eventualmente converge a una estimación donde la mitad de los valores de los píxeles son mayores que el fondo y la otra mitad menores. El principal problema de esta técnica es que la recuperación a los cambios repentinos en el escenario de fondo es lenta. Como una mejora, en [9] se agregó una variable $1/\alpha$ como una constante de rapidez de cambio en la actualización del modelo de fondo. Algunos modelos basados en filtro de mediana recursivo encontrados en la literatura son [21]–[24].

Otro de los modelos básicos es propuesto en [25] se basa en *frame difference model* (FDM) y un método llamado *background difference model* (BDM). En FDM se calcula la diferencia de los cuadros previos y el actual. Si el valor de los píxeles es mayor o igual a un umbral entonces serán considerados como objetos en movimiento, si el valor es menor los píxeles serán considerados como parte del fondo. Por otro lado, BDM cuenta para cada píxel de la imagen actual, el número de veces que han sido considerados como fondo en n previas máscaras binarias almacenadas con el fin de lograr una actualización más rápida del modelo de fondo al sobrepasar un cierto umbral.

B. Modelos Combinados

Estos modelos son una combinación de dos o más técnicas de detección de movimiento y debido a que cada técnica se da un peso similar, resulta difícil dar una clasificación específica respecto a su técnica. Algunos de estos modelos son presentados a continuación.

Combinando características de algoritmos de modelado de fondo bien establecidos como el modelo de mezclas de Gaussianas, redes de mapas auto organizados o *self organized maps* (SOM) y métodos sensitivos a la iluminación, el método presentado en [26] maneja de una forma eficiente cambios de iluminación repentinos de fuentes de iluminación ya sea naturales o artificiales. Una ventaja significativa de esta propuesta es el hecho que no requiere inicialización o entrenamiento complejo. Sin embargo, no alcanza un procesamiento en tiempo real para videos con resolución de 320x240. En [27], se combina las características del filtro de mediana, la técnica *markov random field* (MRF) y aprendizaje por múltiples vistas. El algoritmo tiene tres pasos principales. 1) generar una imagen de referencia mediante el filtro de mediana temporal, 2) un aprendizaje con múltiples vistas es ideado para estimar la probabilidad condicional para ambos modelos; de fondo y objetos en primer plano, 3) la técnica MRF es usada para incorporar el contexto del espacio temporal a la

decisión de fondo u objeto en primer plano. El método maneja sombras, cambios de iluminación y ruido suave. Sin embargo, en situaciones como movimientos repentinos de cámara y fondos dinámicos, se obtiene un bajo desempeño. Otro método propuesto en [28] se basa en la combinación de flujo óptico *Lucas Kanade* (LK) y KDE para obtener la distribución de energía positiva y negativa de una serie de cuadros. Cuando la densidad del núcleo es más pequeña que un valor de umbral es considerada para ser parte del fondo, mientras que una densidad de núcleo grande es considerada para ser parte del modelo de objetos en movimiento. El algoritmo de flujo óptico LK es usado para disminuir ruido en la segmentación final. En [29] se describe el método *monitoring ensemble learning detection* (MELD) que integra las características de GMM, flujo óptico LK y *texture based moving object detection* (TBMOD), mantiene un bajo costo computacional y es capaz de adaptar los pesos de cada método automáticamente. MELD, permite flexibilidad ya que es posible combinar diferentes algoritmos o enfoques dependiendo de la situación. Los resultados de MELD son comparables con los del estado del arte y satisfacen lo requerimientos de aplicaciones en tiempo real.

C. Modelos Estadísticos

Los modelos estadísticos son una forma matemáticamente formalizada de realizar aproximaciones y hacer predicciones a partir de dichas aproximaciones. Se dividen en modelos paramétricos y no paramétricos.

El algoritmo auto-adaptativo presentado en [30] ajusta sus parámetros de acuerdo con las características del escenario. El método es llamado *self-adaptive* GMM (SAG) y combina dos procesos: a nivel cuadro y a nivel píxel. Es basado en el modelo de mezclas Gaussianas propuesto por Zivkovic y Van Der Heijden [94], y el método *multi dimensional Gaussian kernel density transform* (MDGKT) [95]. SAG puede adaptarse automáticamente a los cambios de iluminación ya que cuenta con un parámetro de aprendizaje adaptivo. Los autores del método son unos de los pocos que realizan las comparaciones de dos implementaciones en código distintas. Obteniendo una velocidad de 8.3 FPS en MATLAB y entre 25 a 30 FPS en C++. Por su parte C. Spampinato *et al.* [31] presentan un método no paramétrico que modela el fondo y los objetos dinámicos por medio de *textons* y el algoritmo KDE.

D. Modelos Basados en Redes Neuronales Artificiales

Son modelos matemáticos inspirados en el comportamiento de las redes neuronales biológicas. Son capaces de aprender de la experiencia, de generalizar, de extraer características a partir de entradas.

En [32] se presenta un modelo de segmentación para objetos dinámicos y estáticos mediante *discrete time cellular neural network* (DTCNN). El método involucra tres principales bloques para la realización de la segmentación dinámica: registro de fondo dinámico, detección de objetos dinámicos y mejoras a la segmentación de objetos. Una ventaja notoria del método es que no requiere ajuste manual de sus parámetros para procesar diferentes secuencias de video. El enfoque por G. Ramírez-Alonso *et al.* [33] *auto-adaptive parallel SOM architecture* (APPSA), presenta un nuevo sistema de modelado de fondo basado en dos redes SOM que se adaptan en forma paralela a diferentes velocidades. El algoritmo puede identificar

automáticamente el problema principal que afecta al escenario y a la detección de los objetos en movimiento. La identificación es lograda mediante cuatro módulos clasificados en: escenas con fondos estáticos, escenas normales, escenas dinámicas y escenas complejas.

Otro enfoque *retinotopic SOM* (RESOM) [34] está inspirado en el mecanismo de la corteza visual humana. RESOM puede adaptar sus parámetros de aprendizaje basándose en el comportamiento del escenario. Sin embargo, el método aún presenta problemas debido al camuflaje y su tiempo de procesamiento es lento. El método bio-inspirado llamado SOM-CNN es propuesto en [35], se basa en los métodos SOM y *cellular neural networks* (CNNs). El método es una mejora a la arquitectura RESOM que propone un esquema novedoso de umbralización llamado *neighbor threshold CNNs* (NTCNN).

El modelo *stacked multilayer SOM background model* (SMSOM-BM) [36] consiste en una capa de entrada y varias capas computacionales de salida para cada píxel de la imagen, donde cada capa se utiliza para modelar una parte de las características del fondo. SMSOM-BM necesita imágenes de entrenamiento para ajustar un parámetro τ de aprendizaje. En [37] se propone un método de detección de movimiento llamado *neural response mixture* (NeRM). En NeRM se extraen características para construir un modelo de mezcla de Gaussianas o GMM de respuesta neuronal mixta para capturar rasgos únicos de los objetos (personas, vehículos, animales, etc.) en entornos de vigilancia reales. Una de las desventajas de este método radica en el hecho de que requiere un extenso conjunto de entrenamiento.

M. Braham y M. Van Droogenbroeck [38] proponen un método para el modelar el fondo a través de un algoritmo basado en características espaciales aprendidas con redes neuronales convolucionales. Este método se compone principalmente por cuatro etapas: extracción de fondo de la imagen, generación del conjunto de datos, entrenamiento de la red y sustracción de fondo. El entrenamiento de la red se logra mediante una arquitectura neural muy similar a la red LeNet-5 [96] usada para la clasificación de dígitos. Otro modelo propuesto en [39] considera características de apariencia y movimiento, mediante el diseño de una arquitectura compuesta de dos redes convolucionales. La primera, *appearance network* (A-net) detecta objetos en movimiento como peatones, animales, carros, etc. La segunda, *motion network* (M-net) separa los objetos en movimiento de la diferencia entre el cuadro actual del video y el modelo del fondo. Este método fue implementado a través de la librería *tensorflow* y alcanza una velocidad de 50 FPS en GPU.

Por otra parte, el método *foreground segmentation network* (FgSegNet) [40], propone un enfoque basado en redes convolucionales que adapta una red VGG-16 pre-entrenada bajo un esquema compuesto de tres redes convolucionales *encoder-decoder* que son alimentadas con imágenes RGB a diferentes escalas. FgSegNet ha demostrado su robustez y actualmente se encuentra en la primera posición de la base de datos *Change Detection 2014*. Sin embargo, su tiempo de procesamiento es elevado, siendo 18 FPS en imágenes de resolución 320x240 usando GPU. Anterior a FgSegNet, fue propuesto el algoritmo *cascade convolutional neural networks* (Cascade CNN) [41], que usa tres redes convolucionales en cascada alimentadas con imágenes a diferentes escalas,

teniendo la primera imagen a escala original y las otras dos a escalas de 0.75 y 0.5 respectivamente. Actualmente Cascade CNN se encuentra en tercer lugar de la base de datos *Change Detection 2014*. Otro enfoque que hace uso de una red convolucional es presentado en [42], el cual utiliza las máscaras de segmentación del algoritmo *self-balanced sensitivity segmenter* (SuBSENSE) para modelar el fondo, a su vez la salida del algoritmo *flux tensor with split Gaussian models* (FTSG) es usado como un detector de movimiento a manera de tener una longitud de memoria auto-adaptiva en el modelo de fondo. Su procesamiento es en tiempo real con recursos computacionales convencionales.

Dos enfoques más, que hacen uso de redes convolucionales son [43] y [44]. El primero utiliza tres redes convolucionales atrosas con diferente dilatación para extraer información espacial de diferentes vecindarios de píxeles. El segundo, aplica redes convolucionales 3D incluyendo información temporal de la escena de video y reporta mejores resultados incluso que el algoritmo Cascade CNN. Por su parte, en [45] se propone un método de modelado de fondo a través del uso de una red *restricted boltzmann machine* (RBM). La RBM es una red neuronal, de dos capas, donde la primera capa es definida como la capa visible, la cual se alimenta de la intensidad del píxel de la imagen, y la segunda capa es llamada oculta y funciona como un detector de características. Una de las principales desventajas de RBM es que su desempeño es proporcional al número de capas ocultas, debido a que cada capa oculta representa una característica prototipo.

El método *background modeling Wisard* (BGWiS) [46] puede ser visto como un histograma del número de ocurrencias de todos los subpatrones binarios ocurridos durante el entrenamiento. En este caso de estudio se asume que el modelo de fondo de un píxel está formado por la combinación de los más frecuentes subpatrones contenidos en las RAMs. Este método fue probado en la base de datos SBI y previamente en la base de datos *Change Detection* [97], demostrando ser uno de los métodos que utiliza los mismos parámetros iniciales en todos los videos. En [47] se presenta *Gibbs-Markov random field* (GMRF) y una red *Hopfield-type neural network* (HTNN). En los resultados experimentales, el método mostró mejores resultados resolviendo situaciones de baja iluminación y cambios de iluminación repentinos. El método propuesto en [48] se basa en una red *radial basis function* (RBF) con un módulo de generación de múltiples fondos que determina el número de capas ocultas en la red. El método es adecuado para manejar escenarios dinámicos o estáticos y puede alcanzar una velocidad de 34 FPS con una resolución de 320x240.

E. Modelos Estadísticos Avanzados

Estos métodos son derivados de los modelos estadísticos convencionales como GMM o el método KDE. No obstante, los modelos estadísticos avanzados realizan combinaciones entre modelos estadísticos convencionales y/o proponen el uso de otras distribuciones paramétricas y no paramétricas, con el propósito de crear algoritmos más robustos y con mejores características.

En [49] se presenta un método robusto orientado a videos de sistemas de vigilancia en alta resolución. Utiliza el cálculo de vectores de orientación de movimiento mediante el método *motion orientation histogram* (MOH), el uso de cuadro de

diferencia y *local binary patterns* (LBP) [98]. Los objetos en movimiento se segmentan por medio de un algoritmo de bloques adaptivos particionados. El método logra buenos resultados en situaciones de inicialización del fondo con camuflaje y *bootstrapping*. Por otro lado, para mejorar al método no paramétrico *visual background extractor* (ViBe) se añaden algunas características en [50]; una estructura que indica el nivel de intermitencia (parpadeo) de cada píxel, una operación morfológica de cierre sobre la máscara binaria de la detección de movimiento y un módulo para evitar insertar objetos de interés en el modelo de fondo. El algoritmo propuesto realiza procesamiento en tiempo real a velocidades mayores a 30 FPS. Sin embargo, cuenta con dificultades en situaciones de camuflaje.

Los algoritmos ViBe y *Graph cut* son utilizados para realizar la detección de movimiento en [51] en el método *Graph Cut S/T ViBe* (GC STViBe). El grafo o *Graph* usado en este método, es conformado como un conjunto de nodos definidos por los píxeles de la imagen de entrada y un conjunto indirecto de bordes que están conectados a los nodos. Por su parte, en [52] se propone un método que utiliza la probabilidad y entropía de bloques de dimensión $W \times W$ para tomar la decisión si el bloque pertenece a objeto en movimiento o a fondo. Dan Yan et al. presentan un algoritmo llamado *stability of adaptive feature* (SoAF) [53], el cual utiliza características de color RGB, escala de gris y textura. Como descriptor de textura se realiza una mejora al método SILTP para hacerlo invariante a la rotación llamado *rotated scale invariant local ternary pattern* (R-SILTP). R-SILTP requiere menor memoria ya que solo usa un bit para codificar las diferencias entre el píxel central y su vecindario.

El método [54] selecciona el mejor espacio de color entre RGB y YCbCr mediante un módulo llamado *color space module* (CSM) que basa su criterio de decisión de color óptimo en el error cuadrático medio, luego cada píxel se modela como una mezcla de gaussianas. En [55] se proponen los métodos *spatially consistent background modeling* (SBM) y *spatiotemporal background modeling* (STBM). SBM utiliza el método GMM, como modelo inicial de fondo, luego se asume que los píxeles conectados con valores similares tienen una estimación de fondo semejante y por lo tanto un modelo de fondo espacialmente consistente puede ser obtenido. La similitud entre dos píxeles es obtenida mediante *minimax path* siendo extraído con el uso de *minimum spanning tree* (MST).

El algoritmo STBM es una extensión de SBM en el dominio temporal usando flujo óptico como base para obtener vectores de movimiento entre los píxeles actuales y los pasados. Se seleccionó el algoritmo de flujo óptico más rápido y simple. Por lo que, una mejora al método STBM podría ser probar con otros algoritmos de flujo óptico relativamente más lentos, pero con mejores desempeños. Otro enfoque es presentado por S. Jiang and X. Lu en [56], llamado *weight-sample-based method for background* (WeSamBE) y asume que para cada píxel, el modelo de fondo puede ser representado a través de una colección de N muestras de fondo previas. Cada muestra de fondo se liga a un peso. El peso actúa como una forma de caracterizar la importancia de cada muestra y se refuerzan o penalizan automáticamente de acuerdo con el grado de actividad de las muestras, lo que permite al algoritmo adaptarse rápidamente a los cambios de escenario.

En [57] se presenta un algoritmo para manejar cambios repentinos de iluminación, es llamado *multimode background subtraction version 0* (MBS V0). MBS V0 primero es entrenado a manera de generar un banco de múltiples modelos de fondo Gaussianos. Luego, para el cuadro actual del video se selecciona el mejor modelo para producir una máscara binaria (MB) para cada uno de los canales en cada espacio de color. Las MB de cada espacio de color son fusionadas y luego purgadas, para finalmente obtener una máscara de detección. En [58] se presenta una adaptación de MBS V0 que añade un módulo de segmentación basado en superpíxeles. El objetivo es la eliminación de ruido espacial considerando las estimaciones de probabilidad iniciales y la información de color de un vecindario de píxeles bajo un cuadro de superpíxeles.

Otro método que entra en la categoría de modelo estadístico avanzado es [59], el cual usa multi-modelos por píxel basados en el algoritmo no paramétrico *a fast adaptive multi-resolution background detection algorithm* (AMBER) [60] considerando un mayor número de *templates*. Por otro lado, el método *pixel-based adaptive segmenter* (PBAS) se adapta bien a cambios de iluminación, pero el procedimiento que distingue los cambios en objetos es deficiente, por lo que, en [61] se propone un método adaptivo de actualización de fondo que funciona tanto en nivel píxel como objeto. El enfoque propuesto por J. D. Romero et al. [62] resuelve los cambios de iluminación repentinos y graduales a través de un método llamado *lightness-red-green-blue* (LRGB). Este modelo compara dos imágenes y encuentra píxeles con luminosidad escalada.

En [63] se presenta un enfoque que realiza una mejora al método original *speeded up robust features* (SURF) [99]. Este usa una matriz Hessiana para encontrar puntos de interés, el determinante de la matriz es una expresión de los cambios locales alrededor de un área. La mejora del método propuesto se basa en limitar el número de puntos de características detectadas y un cálculo rápido de su orientación dominante. El algoritmo mejoró la velocidad y precisión del original SURF, al reducir la complejidad de cálculo. J. Dou y J. Li en [64] proponen un método basado en *scale-invariant feature transform* (*Sift flow*) [100] para modelar el fondo y detectar objetos en movimiento. Cada píxel es modelado con un grupo de descriptores adaptivos *Sift flow* que son calculados sobre una región rectangular alrededor del píxel. Este algoritmo obtuvo una velocidad de procesamiento en una resolución de 320x240 de 3 FPS.

En [65] se presenta un enfoque basado en aprendizaje por medio de aproximación estocástica. El método fue ideado para trabajar con cámaras PTZ de zum panorámico o cámaras estáticas que presentan variación. Dos modelos son usados en este método, uno para seguir los movimientos de la cámara y otro para detectar los objetos en movimiento. El método posee la desventaja de necesitar repetidamente ajustes a sus parámetros para obtener mejores resultados en las diferentes secuencias de video PTZ. H. Lu et al. [66] proponen un novedoso método de compartición de modelos GMM. Donde, un mecanismo de compartición es presentado para modelar fondo y objetos en primer plano. El mecanismo de compartición explota la relación espaciotemporal entre los píxeles, ya que dinámicamente para cada píxel se busca un modelo óptimo de fondo u objeto en primer plano en una región $N \times N$. El enfoque, llamado *spatially global Gaussian mixture models and*

uncertains (SGGMM) propuesto en [67] es un modelo de incertidumbre que se enfoca en mejorar y aumentar el color de fondo para obtener una detección robusta. Para la realización del método SGGMM se tienen dos modelos gaussianos: de fondo y de estimación. El método afronta problemas relativos a sombras, vibraciones y fondos dinámicos. Sin embargo, el usuario debe especificar el número de Gaussianas.

Otra técnica es el modelo estadístico no paramétrico *self-balanced sensitivity segmenter* (SuBSENSE) [68] método de segmentación a nivel píxel que depende de las características espacio temporales binarias y de información de color para detectar movimiento. La detección de movimiento se basa en un método de búsqueda de patrones *local binary similarity patterns* (LBSP), donde los píxeles vecinos semejantes al valor central tendrán un valor de 1 y aquellos que su valor difiera serán etiquetados con un 0. El método es extremadamente flexible y puede adaptarse incluso a los más difíciles escenarios. La velocidad de procesamiento de este algoritmo lo hace ideal para aplicaciones en tiempo real. En [69] se propone un método basado en grafos y una función de energía N^2cut para evaluar las relaciones espaciales en la segmentación y los cortes normalizados. El algoritmo fue probado en tiempo real en videos de alta definición tomados por una cámara GoPro obteniendo un desempeño de aproximadamente 20 imágenes por segundo.

F. Modelos Difusos

Son modelos que permiten hacer inferencias con base en reglas heurísticas.

En [70] se propone el uso de un correlograma multi-canal difuso para la detección de movimiento. Es basado en el modelo de correlograma difuso en [101]. El método no requiere de cuadros de entrenamiento para el modelo de fondo inicial y soporta *bootstrapping*. Además, mantiene todos sus parámetros constantes excepto un umbral th , y cuenta con un costo computacional elevado. D. K. Panda y S. Meher [71] proponen una técnica para el modelado de fondo a través de la combinación de *color difference histogram* (CDH) [102] y lógica difusa. El método se compone de cinco etapas principales: 1) determinar la diferencia de color de histograma CDH, 2) emplear *Fuzzy c-means* para formar un histograma de color difuso o *fuzzy color difference histogram* (FCDH), 3) inicialización del modelo de fondo, 4) detección de objetos en primer plano usando concordancia de similitudes entre histogramas, 5) finalmente el mantenimiento del fondo es logrado al comparar la concordancia de similitud del FCDH del fondo y del cuadro actual, contra un umbral específico. El método ofrece una solución a problemas debidos a la variación de iluminación y fondo dinámico. No obstante, falla en suprimir sombras y *bootstrapping*.

G. Modelos de Subespacios

Son modelos basados en una descomposición derivada de un espacio específico, algunos de los métodos que entran en esta categoría son: análisis de componentes principales, descomposición ortogonal, descomposiciones basadas en súper píxeles, matrices basadas en bajo-rango, descomposición esparcida, entre otros.

El método *robust principal component analysis* (RPCA) es un modelo que asume que los píxeles del fondo entre cuadros

adyacentes están relacionados linealmente, mientras que los píxeles pertenecientes al objeto en movimiento están esparcidos. Por lo que, una matriz de observación puede ser descompuesta en una matriz de bajo rango para el fondo y una esparcida para los objetos dinámicos. Una adaptación al modelo RPCA es presentada en [72]. Está compuesto de cinco puntos principales: 1) segmentación inicial, 2) construcción de una matriz RPCA basada en bloques (*Block-based RPCA*), 3) computo del parámetro de balance, 4) solución *Block-based RPCA* mediante multiplicadores aumentados de lagrange inexactos (IALM), 5) procesamiento final en la detección de movimiento. El método logra eliminar la influencia de sombras y presenta buenos resultados en fondos dinámicos y cambios de iluminación.

Otros métodos basados en RPCA, se presentan en [73] y [74]. El primero propone un esquema *iteratively reweighted least squares* (IRLS) en la factorización de la matriz de bajo rango y se incluyen restricciones espaciales al proceso de minimización. El segundo realiza una mejora al método *online-Robust PCA* (OR-PCA) a través de inicialización por lotes [75]. La mejora a OR-PCA resuelve los inconvenientes de los cálculos complejos y el lento tiempo de procesamiento del tradicional RPCA a través de un método recursivo. Esta mejora fue basada en el método de diccionarios de aprendizaje en línea, y algunos de los primeros cuadros son usados para el procesamiento por lotes. El método posee buenos resultados para eliminación de sombras y visualmente muestra una mejor detección respecto a los métodos evaluados.

En [76] se presenta el algoritmo *contiguous outliers representation via online low rank approximation* (COROLA). COROLA adopta una aproximación de matriz de descomposición de bajo rango en línea para poder resolver situaciones con escenarios complejos con fondos dinámicos o ruidosos. El algoritmo es basado en los enfoques *detecting contiguous outliers in the low rank representation* (DECOLOR) [77] y OR-PCA. A diferencia del método DECOLOR, el costo computacional de COROLA es independiente del número de imágenes procesadas. Por lo que, solo se incrementa el tiempo al incrementar el tamaño de resolución espacial de la imagen. A pesar de su desempeño satisfactorio COROLA comparte una desventaja con DECOLOR, ya que no cuentan con un criterio de convergencia convexo, estos probablemente caerán en mínimos locales.

En [78] se propone un algoritmo de detección de movimiento llamado *total variation regularized RPCA* (TVRPCA), en el cual, se descompone el video analizado en tres partes: fondo estático de bajo nivel, fondo dinámico esparcido y objetos en primer plano suavizados y esparcidos. El video es tratado como un tensor 3-D que incluye información temporal. TVRPCA se comparó con algoritmos como RPCA y DECOLOR, obteniendo una clara ventaja, especialmente en videos con la presencia de mal clima o fondos complejos. Algunos problemas que presenta son camuflaje y pérdidas cuando existen objetos pequeños que se mueven a gran velocidad, como automóviles que están lejanos a la cámara. En [79] se presenta una matriz de descomposición basada en superpíxeles, junto con una norma de regularización de máximos o *Max-Norm* y una estructura de restricciones esparcidas. En general el método cuenta con un buen desempeño en el manejo de fondos dinámicos y cambios de

iluminación. Pero tiene un bajo desempeño en objetos intermitentes y vibraciones de cámara.

H. Diccionarios de Aprendizaje

Los diccionarios de aprendizaje son un tipo de modelo esparcido o disperso donde la suposición básica de la representación dispersa, es que la imagen actual, se puede representar como una combinación lineal de vectores que forman un diccionario. A continuación, se presentan algunos métodos que caen dentro de esta categoría.

P. L. St-Charles *et al.* [80] proponen un método mediante el uso de un nuevo tipo de enfoque basado en *word consensus* llamado *pixel-based adaptive word consensus* (PAWCS). PAWCS es fundamentado en la caracterización y monitoreo de las diferentes representaciones de fondo mediante el uso de diccionarios locales con características de color y textura. Además, se integra el uso de características LBSP que ayudan en situaciones de variaciones de iluminación. El algoritmo fue implementado en C++ con un CPU Intel i5 trabajando a 22 cuadros por segundo.

Otro enfoque que hace uso de codificación esparcida se presenta en [81] para trabajar con videos del espectro infrarrojo. El método utiliza la información de velocidad y aceleración en vectores propios para construir un modelo saliente local que ayuda a extraer el área del objeto de interés, y un modelo de representación esparcida es usado para separar los objetos dinámicos del fondo. Los resultados de detección del método fueron robustos para situaciones de fondos dinámicos y cambios de iluminación, pero falla en detectar objetos pequeños. En [82], se presenta un método que mejora la eficiencia de detección de objetos dinámicos mediante un algoritmo esparcido de *codebook* basado en superpíxel. Los píxeles vecinos se agrupan en superpíxeles de acuerdo con sus similitudes en brillo y textura. El método logra reducir significativamente los recursos de memoria y mejorar el tiempo de procesamiento sin perder precisión en la detección de movimiento.

En [83] se propone un enfoque para modelar el fondo llamado *background modeling through dictionary learning* (BMTDL). BMTDL construye una representación de los datos a través de una combinación lineal de elementos básicos llamados *atoms* que forman un diccionario. Inicialmente, cada imagen del video se divide en una cuadrícula regular del mismo tamaño, cada cuadrilátero es llamado parche. En esencia cada parche puede tener cuatro posibles estados; *normal*, *anomaly*, *update* y *clean*. El método aprovecha la correlación espacial entre píxeles y muestra estabilidad a cambios de iluminación, vibraciones y fondos dinámicos. Sin embargo, su costo computacional está relacionado con el parámetro de frecuencia de *update* y el tamaño de los parches.

En [84] se propone una representación de fondo basada en diccionarios de aprendizaje y mantenimiento histórico de los píxeles, el método es llamado *background representation approach with dictionary learning and historical pixel maintenance* (BREW-DLHPM) y se compone básicamente de cinco bloques; 1) un muestreo temporal de los cuadros de video, 2) aplicación de diccionarios de aprendizaje sobre la imagen actual, 3) mantenimiento de fondo a nivel píxel, 4) estimación del modelo de fondo dinámico, 5) decisión para detectar objetos en movimiento. BREW-DLHPM puede trabajar en tiempo real

en resoluciones de 320x240 y 432x288, obteniendo 61.01 y 36.42 FPS respectivamente. Sin embargo, a más resolución espacial mayor será el costo computacional. C. David and V. Gui [85] proponen un método de modelado de fondo compuesto de una combinación lineal esparcida de elementos de un diccionario de aprendizaje K-SVD [103] y un conjunto de coeficientes asociados a cada posición de la imagen. Además, a manera de reducir los problemas introducidos por fondos dinámicos es añadido un modelo Gaussiano en los coeficientes. Se usan dos algoritmos de umbral. El primero elimina los errores debido a fondos dinámicos y posible ruido, y el segundo finaliza la segmentación y es calculado mediante *K-means*.

I. Modelos de Transformación de Dominio

El principio de estos modelos se basa en separar el fondo y los objetos dinámicos mediante la ayuda de una transformación de dominio. Para esto, diferentes transformaciones pueden ser usadas, tal como onduletas *Wavelet*, Transformada rápida de Fourier, *Hadamard*, entre otras.

A manera de reducir el efecto de las variaciones de luz mediante onduletas *Wavelet*. en [86] se plantea un algoritmo llamado *background wavelet model* (BWM), en él se usa una norma H2 a manera de extraer los coeficientes de energía entre el cuadro actual y el cuadro de fondo. La norma H2 es usada para medir la covarianza en estado estable o la energía de la respuesta de salida. En [87] se presenta un método que utiliza el análisis de onduletas a manera de mejorar el desempeño de un algoritmo de diferencia de cuadros basado en memoria. La idea básica es filtrar el ruido a través del uso de una memoria, permitiendo que el detector de movimiento ignore los cambios causados por el ruido. Los resultados experimentales muestran que el método logra detectar objetos dinámicos pequeños y lentos, también objetos con bajo contraste y en ambientes con cambios de iluminación. Otro método mediante transformación de dominio llamado *texture guided weighted voting* (TGMV) [88] emplea la transformación de *Wavelet* para descomponer la imagen en bandas de frecuencia, y así lograr diferenciar los objetos dinámicos pequeños y difícilmente notorios del escenario de fondo. Además, se utiliza un esquema de pesos basado en intensidad y texturas extraídas mediante LBP. El método logra una robusta segmentación de objetos dinámicos en situaciones de camuflaje en comparación a los métodos con los que se evalúa.

J. Modelos Tensores Robustos

Los modelos tensores se basan en la asunción de que no todas las relaciones en la naturaleza son lineales, pero la mayoría son diferenciables y de este modo se pueden aproximar localmente con sumas de funciones multi-lineales. Así la mayoría de las magnitudes en física se pueden expresar como tensores. Matemáticamente, un tensor es una entidad algebraica representada por un cierto número de componentes que generaliza los conceptos escalar, vector y matriz de cualquier sistema de coordenadas elegido. A continuación, se presentan los enfoques que caen dentro de esta categoría.

En [89] se presenta un sistema de detección llamado *flux tensor with split Gaussian models* (FTSG) que aprovecha los beneficios de fusionar un método basado en un tensor espacio-temporal y un modelo Gaussiano para producir mejores resultados y reducir errores debido a cambios de iluminación,

ruido o efecto halo. Ofrece tres clasificaciones en la máscara de detección: objetos en movimiento, objetos estáticos y sombras o iluminación. Una desventaja es su tiempo de procesamiento no apto para tiempo real. Q. Zhao *et al.* [90] proponen un modelo generado por medio de *bayesian robust tensor factorization* (BRTF). Se basa en el uso del tensor de bajo rango CANDECOMP/PARAFAC (CP) [104], que utiliza la información global, mientras que un tensor esparcido captura la información local, obteniendo una predicción robusta de la distribución. El método tiene la característica de que todos los parámetros del modelo pueden ser aprendidos automáticamente de los datos observados.

K. Otros

En esta clasificación, se contemplan algoritmos que no se basan en ningún modelo de los descritos anteriormente, a manera que resulta complicado incorporarlos en alguna clasificación.

En [91] se usa programación genética para combinar los resultados de los mejores algoritmos de detección de movimiento en *Change Detection* y obtener máscaras binarias con mejores desempeños. Se realizaron diferentes combinaciones y se observó que los algoritmos llamados IUTIS-3 y IUTIS-5 tuvieron los mejores resultados.

Por otro lado, Philipp Tiefenbacher *et al.* [92] Presentan un modelo basado en un controlador proporcional integral y derivativo (PID). El controlador PID regula la decisión del umbral de objetos dinámicos y la tasa de actualización del modelo de fondo. El método obtuvo buenos resultados para manejar fondos dinámicos. En [93] se propone un enfoque para modelado de fondo mediante el uso de bloques de características espacio-temporales correlacionadas y extraídas directamente de videos comprimidos en HEVC. Un video HEVC consiste de una secuencia de cuadros, los cuales están divididos en bloques que no se traslapan llamados *coding tree unit* (CTU). En un video comprimido HEVC, los coeficientes de transformación de un CTU registran cambios incrementales que ocurren entre cuadros adyacentes. Por lo que se asume que los cambios mayores corresponden a objetos dinámicos, mientras que los otros son debidos a elementos dinámicos pertenecientes al fondo. Los resultados experimentales del método propuesto muestran un buen desempeño para situaciones de fondos dinámicos. El método también posee la ventaja de que es relativamente rápido, el promedio de procesamiento oscila entre 25-30 FPS en resoluciones de video de 720×420. Por otro lado, existe un *trade-off* entre el parámetro del tamaño de bloque CTU, su desempeño en *F-Measure* y su velocidad, por lo que este parámetro deber ser elegido cuidadosamente.

III. TENDENCIAS

Esta sección plantea un análisis de las tendencias de diseño de algoritmos en los últimos años. ¿Cuál es el tipo de técnica más utilizada por los autores y sus limitaciones?, ¿Qué cantidad de base de datos son utilizadas para validar su desempeño? y ¿Qué base de datos es la más popular? Además, se analiza el tiempo de procesamiento, el tipo de hardware, espacio de color, y lenguaje de programación para implementar el algoritmo fueron los más populares dentro de los algoritmos analizados.

A. Técnicas más utilizadas y soluciones y retos

De los 77 algoritmos, las tres técnicas más utilizadas corresponden a: modelos estadísticos avanzados 28%, redes neuronales 22%, y un empate entre los modelos de subespacios y básicos 10%. Lo anterior, se observa gráficamente en la Fig. 1. Se cree que los modelos estadísticos avanzados han tomado popularidad en los últimos 5 años debido a que la gran mayoría posee parámetros auto-adaptivos. Sin embargo, uno de los inconvenientes de este tipo de técnica es que entre más robustos son los algoritmos, requieren mayor cantidad de parámetros, lo que conlleva a algoritmos más complejos y a un mayor tiempo de procesamiento para realizar los ajustes necesarios.

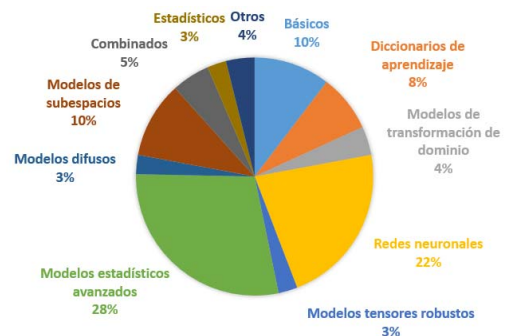


Fig. 1. Porcentaje de uso de los diferentes tipos de técnicas dentro de los 77 algoritmos encontrados en el periodo 2013-2018.

A continuación, en la tabla II se resumen los problemas abiertos y las soluciones que ofrecen los algoritmos analizados de acuerdo con su tipo de técnica. Esto con el propósito de establecer áreas de oportunidad o mejora para el desarrollo de futuros algoritmos de detección de movimiento y que el lector identifique cuál técnica que mejor se adecua de acuerdo con el tipo de aplicación que desee realizar.

TABLA II
SOLUCIONES Y RETOS POR TIPO DE TÉCNICA

Técnica	Soluciones y problemas abiertos
Modelos básicos	Son relativamente acertados para modelar el fondo cuando la iluminación es constante. Sin embargo, en existencia de cambios repentinos de iluminación o fondos dinámicos, resulta en un alto número de falsos positivos. La recuperación a los cambios repentinos en el escenario de fondo es lenta. Alcanzan un procesamiento en tiempo real para videos con resolución de 320x240. Manejan sombras, cambios de iluminación y ruido suave. Sin embargo, en situaciones como movimientos repentinos de cámara y fondos dinámicos, se obtiene un bajo desempeño.
Modelos combinados	Altamente eficientes en el manejo de fondos dinámicos y una gran variedad de problemas de video. Algunos algoritmos pueden adaptar sus parámetros de aprendizaje basándose en el comportamiento del escenario. Sin embargo, aún presentan problemas debido al camuflaje y su tiempo de procesamiento es elevado. La mayoría requieren uso de GPU para alcanzar procesamiento en tiempo real y requieren un extenso conjunto de imágenes de entrenamiento.
Modelos basados en redes neuronales artificiales	Afrontan problemas relativos a sombras, vibraciones y fondos dinámicos. Pero entre más robustos son los algoritmos, requieren mayor cantidad de parámetros y en ocasiones se necesita realizar ajustes manuales en reiteradas ocasiones para obtener mejores resultados.
Modelos estadísticos y estadísticos avanzados	Resuelven fondos dinámicos y cambios de iluminación, pero tienen problemas en suprimir sombras y poseen un costo computacional elevado.
Modelos difusos	Manejan cambios de iluminación y fondos dinámicos, pero requieren de cálculos complejos y alto tiempo de procesamiento. Además, existen problemas por camuflaje, detección de objetos dinámicos pequeños, objetos intermitentes y vibraciones de cámara.
Modelos de subespacios	Resuelven fondos dinámicos, sombras y cambios de iluminación. Sin embargo, fallan en detectar objetos pequeños.
Dictionarios de aprendizaje	Costo computacional elevado depende de la cantidad de datos en el diccionario y el tamaño de la imagen.
Modelos de transformación de dominio	Solucionan cambios de iluminación y camuflaje, pero presentan problemas en situaciones de fondos dinámicos.
Modelos tensores robustos	Manejan cambios de iluminación, ruido y objetos que se vuelven estáticos. Poseen tiempos de procesamiento no aptos para tiempo real y requieren de cálculos complejos.
Otros	Algunos requieren cuadros de entrenamiento y resultados de segmentación de otros algoritmos.

B. Bases de Datos

Hoy en día existen una gran cantidad de bases de datos de video debido a la necesidad de evaluar los algoritmos mediante el mayor número de situaciones posibles. No obstante, debido a una falta de normativa, cada autor hace referencia y reporta resultados de acuerdo a la de su preferencia utilizando inclusive más de una base de datos. En la Fig. 2, se muestra el comportamiento del número de base de datos utilizadas.

Cabe destacar que el 21% de los autores no utilizaron ninguna base de datos pública y prefieren realizar bases de datos propias con algunos videos específicos para evaluar el desempeño de su algoritmo. Por otro lado, en la Fig. 3 se muestran las bases de datos más populares reportadas por los autores. En total, 16 bases de datos fueron encontradas en las publicaciones, de las cuales *Change detection (CDnet)* [105] fue la base de datos más utilizada seguida de I2R y Wallflower. Las demás bases de datos se mantuvieron por debajo de 6 ocurrencias de uso.

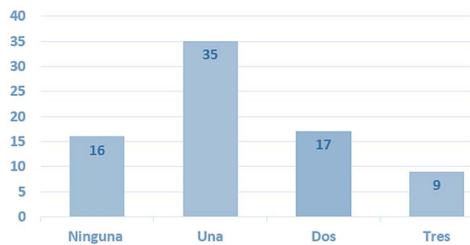


Fig. 2. Número de base de datos utilizadas por los autores de algoritmos.

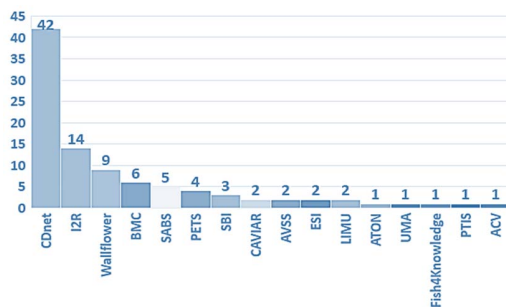


Fig. 3. Base de datos más populares.

C. Características más Relevantes

La tabla III resume las características reportadas de cada una de las publicaciones para observar cual es la tendencia en el uso del software, hardware, espacio de color y tiempos de procesamiento. Los acrónimos NE y SE hacen referencia a “No Especificado” y “Sistema Embebido” respectivamente. Los tiempos de procesamiento fueron normalizados a una resolución de 320x240 píxeles. Como se observa en la Fig. 4, el espacio de color más utilizado fue RGB seguido de escala de grises (GRAY) y HSV. Cabe destacar que existen algoritmos que hacen uso de dos o más espacios de color o implementaciones en lenguajes de programación distintos, por lo que, se consideró el número total de veces que fueron reportados dentro de los 77 algoritmos. El lenguaje de programación más popular que se reporta en las publicaciones es el lenguaje M de MATLAB, seguido de C++ y Phyton, cabe señalar que 33 autores no reportaron que tipo de código

utilizaron en su implementación. En el equipo de cómputo o hardware, 8 algoritmos hicieron uso de GPU, siendo en su mayoría modelos basados en redes neuronales, lo cual es de esperarse debido a la gran cantidad de datos requeridos para su entrenamiento.

TABLA III
CARACTERÍSTICAS DE LAS DIFERENTES TÉCNICAS

Ref.	Técnica	Espacio de color	Hardware	Lenguaje	FPS (320x240)
[60]	M. Est. Avanzados	YCbCr	CPU	C++	3517
[93]	Otros	YCbCr	CPU	C++	665
[49]	M. Est. Avanzados	NE	CPU	NE	450
[19]	Básicos	NE	CPU	NE	218
[50]	M. Est. Avanzados	RGB	CPU	C++	137
[18]	Básicos	NE	CPU	Matlab	135
[21]	Básicos	NE	CPU	C++	100
[83]	Dict. Aprendizaje	NE	CPU	NE	84
[82]	Dict. Aprendizaje	YUV	CPU	C++	67
[55]	M. Est. Avanzados	NE	CPU	NE	66
[84]	Dict. Aprendizaje	NE	CPU	NE	61
[92]	Otros	NE	CPU	NE	54
[39]	Redes neuronales	NE	GPU	Phyton	50
[61]	M. Est. Avanzados	GRAY	CPU	C++	44
[66]	M. Est. Avanzados	RGB	CPU	C++	35
[36]	Redes neuronales	HSV	GPU	NE	34
[48]	Redes neuronales	HSV	CPU	NE	34
[67]	M. Est. Avanzados	RGB	CPU	NE	33
[29]	Combinados	NE	CPU	Matlab	32
[54]	M. Est. Avanzados	RGB, YCbCr	CPU	NE	32
[53]	M. Est. Avanzados	RGB, GRAY	CPU	C++	30
[63]	M. Est. Avanzados	NE	GPU	C++	30
[68]	M. Est. Avanzados	RGB	CPU	C++	30
[80]	Dict. Aprendizaje	RGB	CPU	C++	27
[33]	Redes neuronales	HSV	CPU	Matlab	23
[52]	M. Est. Avanzados	GRAY	CPU	Java	23
[31]	Estadísticos	RGB, HSV, Lab	CPU	C++	19
[46]	Redes neuronales	RGB, HSV, Lab	CPU	C++	18
[26]	Combinados	RGB, GRAY	CPU	Matlab	17
[34]	Redes neuronales	RGB	CPU	Matlab	15
[40]	Redes neuronales	RGB	GPU	Phyton	13
[41]	Redes neuronales	RGB	GPU	Matlab	13
[55]	M. Est. Avanzados	NE	CPU	NE	12
[76]	M. de subespacios	NE	CPU	C++	12
[86]	M. Trans. Dom.	RGB	CPU	Matlab	11
[42]	Redes neuronales	RGB	CPU	NE	10
[89]	M. Tens. Robustos	RGB	CPU	Matlab	10
[35]	Redes neuronales	HSV	CPU	Matlab	9
[58]	M. Est. Avanzados	RGB, YCbCr	CPU	Matlab	9
[30]	Estadísticos	RGB	CPU	Matlab	8
[57]	M. Est. Avanzados	RGB, YCbCr	CPU	Matlab	5
[69]	M. Est. Avanzados	RGB	CPU	NE	5
[79]	M. de subespacios	NE	CPU	Matlab	5
[70]	Modelos difusos	RGB	CPU	NE	4
[77]	M. de subespacios	NE	CPU	Matlab	4
[64]	M. Est. Avanzados	RGB	CPU	NE	3
[37]	Redes neuronales	NE	SE	C++	2
[56]	M. Est. Avanzados	RGB	CPU	C++	2
[78]	M. de subespacios	GRAY	CPU	Matlab	0.3
[27]	Combinados	RGB	CPU	Matlab	0.2
[20]	Básicos	NE	CPU	Matlab	NE
[22]	Básicos	GRAY	CPU	Matlab	NE
[23]	Básicos	GRAY	CPU	Matlab	NE
[24]	Básicos	RGB	CPU	NE	NE
[25]	Básicos	NE	CPU	NE	NE
[28]	Combinados	NE	CPU	NE	NE
[32]	Redes neuronales	HSV	CPU	NE	NE
[38]	Redes neuronales	GRAY	CPU	NE	NE
[43]	Redes neuronales	RGB	GPU	NE	NE
[44]	Redes neuronales	NE	GPU	NE	NE
[45]	Redes neuronales	NE	CPU	NE	NE
[47]	Redes neuronales	NE	CPU	NE	NE
[51]	M. Est. Avanzados	Lab	CPU	NE	NE
[59]	M. Est. Avanzados	NE	CPU	NE	NE
[62]	M. Est. Avanzados	RGB	CPU	NE	NE
[65]	M. Est. Avanzados	RGB	CPU	Matlab	NE
[71]	Modelos difusos	Lab	CPU	Matlab	NE
[72]	M. de subespacios	NE	CPU	NE	NE
[73]	M. de subespacios	NE	CPU	NE	NE
[74]	M. de subespacios	NE	CPU	Matlab	NE
[75]	M. de subespacios	NE	CPU	NE	NE
[81]	Dict. Aprendizaje	Lab	CPU	Matlab	NE
[85]	Dict. Aprendizaje	GRAY	CPU	NE	NE
[87]	M. Trans. Dom.	NE	CPU	Matlab	NE
[88]	M. Trans. Dom.	GRAY	GPU	NE	NE
[90]	M. Tens. Robustos	RGB	CPU	Matlab	NE
[91]	Otros	NE	CPU	NE	NE

IV. CONCLUSIONES

Del análisis realizado a la literatura en el periodo del año 2013 a febrero 2018 mostrado en la Sección II, se encontraron 77

algoritmos, los cuales fueron resumidos y clasificados por tipo de técnica. En total, se formaron once categorías de acuerdo con el tipo de técnica, las cuales fueron: modelos básicos, combinados, estadísticos, redes neuronales, modelos estadísticos avanzados, modelos difusos, modelos de subespacios, diccionarios de aprendizaje, modelos de transformación de dominio, modelos tensores robustos y otros modelos.

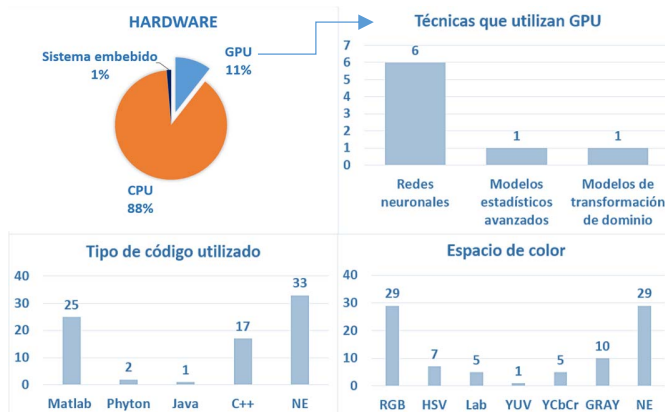


Fig. 4. Tipo de hardware, código y espacio de color más utilizados, y técnicas que utilizan GPU en su ejecución.

Las técnicas más utilizadas corresponden a los modelos estadísticos avanzados seguidos de los modelos basados en redes neuronales con 28 y 22% de uso respectivamente. Aunado a esto, los modelos estadísticos avanzados fueron una de las técnicas más veloces en tiempo de procesamiento con 11 algoritmos que pueden ejecutarse en tiempo real. Por lo que, posiblemente estas técnicas continúen siendo estudiadas los próximos años en aras de obtener algoritmos con mejores características. Por otro lado, la base de datos más reportada en las publicaciones fue *CDnet*. Sin embargo, a pesar de que *CDnet* es ampliamente reconocida por gran parte de la comunidad científica existen algunos autores que aún permanecen reacios a utilizarla.

Por su parte, se encontró que un 11% de los algoritmos utilizan GPU siendo este grupo conformado principalmente por técnicas de redes neuronales, lo cual es de esperarse debido a la gran cantidad de datos necesarios para su entrenamiento. En cuanto el tipo de código existe una tendencia por el uso del lenguaje M de MATLAB, aunque el uso de C++ está ganando popularidad últimamente, quizás debido a librerías disponibles de uso libre como Open CV, aunado a esto típicamente un programa en C++ se ejecuta más rápido que uno en MATLAB, lo cual es ideal para cumplir con las demandas de un procesamiento en tiempo real. El espacio de color preferido por los autores de algoritmos fue RGB, mostrando que los demás espacios de color continúan poco explotados. Por lo que, en un futuro sería interesante analizar si realmente este espacio representa la mejor alternativa o si simplemente es utilizado por su sentido práctico.

Respecto a los problemas abiertos de los trabajos analizados, existe una fuerte tendencia por lograr algoritmos más robustos que manejen en forma automática las diferentes circunstancias que se pueden presentar en escenarios de videos. Siendo los casos de cambios de iluminación, fondos dinámicos y

movimiento de la cámara repentinos los que representan mayor reto. Otro aspecto por trabajar es el ajuste de parámetros de los modelos, ya que en algunos modelos se debe recurrir de manera reiterada al ajuste de parámetros en forma manual, lo que es totalmente impráctico; por lo cual se debe optar por el desarrollo de modelos auto-adaptivos. Otro reto por resolver es el uso de una cantidad excesiva de imágenes de entrenamiento, sobre todo en los modelos de aprendizaje profundo. Aunado a esto algunos autores han recurrido al uso de GPUs para que sus modelos logren un procesamiento en tiempo real y puedan ser usados en aplicaciones reales.

REFERENCIAS

- [1] M. I. Chacon-Murguía, R. Sandoval-Rodríguez, y O. Arias-Enriquez, "Human gait feature extraction including a kinematic analysis toward robotic power assistance", *Int. J. Adv. Robot. Syst.*, vol. 9, pp. 1–9, 2012.
- [2] T. Kimura, M. Ohashi, K. Crailsheim, T. Schmickl, R. Odaka, y H. Ikeno, "Tracking of Multiple Honey Bees on a Flat Surface", en *Fifth International Conference on Emerging Trends in Engineering and Technology*, 2012, pp. 36–39.
- [3] R. Osorio, J. Savage, V. Lomas, y G. Lefranc, "Surveillance System Mobile Object Using Segmentation Algorithms", *IEEE Lat. Am. Trans.*, vol. 13, núm. 7, pp. 2441–2446, 2015.
- [4] D. L. Siqueira y A. M. C. Machado, "People Detection and Tracking in Low Frame Rate Dynamic Scenes", *IEEE Lat. Am. Trans.*, vol. 14, núm. 4, pp. 1966–1971, 2016.
- [5] K. Toyama, J. Krumm, B. Brumitt, y B. Meyers, "Wallflower: principles and practice of background maintenance", *Comput. Vision, 1999. Proc. Seventh IEEE Int. Conf.*, pp. 255–261, 1999.
- [6] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview", *Comput. Sci. Rev.*, vol. 11–12, núm. 1, pp. 31–66, 2014.
- [7] J. Heikkilä y O. Silven, "A real-time system for monitoring of cyclists and pedestrians", *Proc. - 2nd IEEE Int. Work. Vis. Surveillance, VS 1999*, pp. 74–81, 1999.
- [8] N. J. B. McFarlane y C. P. Schofield, "Segmentation and tracking of piglets in images", *Mach. Vis. Appl.*, vol. 8, núm. 3, pp. 187–193, 1995.
- [9] H. Al-Khateeb y M. Petrou, "Automatic Change Detection Method for an Indoor Environment", en *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, pp. 53–58.
- [10] T. S. Caetano, S. D. Olabarría, y D. A. C. Barone, "Performance evaluation of single and multiple-Gaussian models for skin color modeling", en *Conference: Brazilian Symposium of Computer Graphic and Image Processing*, 2002, pp. 275–282.
- [11] K. Kim, T. H. Chalidabhongse, D. Harwood, y L. Davis, "Real-time foreground-background segmentation using codebook model", *Real-Time Imaging*, vol. 11, núm. 3, pp. 172–185, 2005.
- [12] T. Bouwmans, F. El Baf, y B. Vachon, "Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey", *Recent Patents Comput. Sci.*, vol. 1, núm. 3, pp. 219–237, 2008.
- [13] D. Giordano, S. Palazzo, y C. Spampinato, "Kernel Density Estimation Using Joint Spatial-Color-Depth Data for Background Modeling", *2014 22nd Int. Conf. Pattern Recognit.*, pp. 4388–4393, 2014.
- [14] M. Piccardi, "Background subtraction techniques: a review", *IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583)*, vol. 4, pp. 3099–3104, 2004.
- [15] Y. Benezeth, P. M. Jodoin, B. Emile, H. Laurent, y C. Rosenberger, "Review and evaluation of commonly-implemented background subtraction algorithms", en *2008 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.
- [16] N. Buch, S. A. Velastin, y J. Orwell, "A review of computer vision techniques for the analysis of urban traffic", *IEEE Trans. Intell. Transp.*

- Syst.*, vol. 12, núm. 3, pp. 920–939, 2011.
- [17] J. S. Kulchandani y K. J. Dangarwala, “Moving object detection: Review of recent research trends”, *Pervasive Comput. (ICPC), 2015 Int. Conf.*, vol. 1, núm. c, pp. 1–5, 2015.
- [18] B. Nikolov y N. Kostov, “Motion Detection Using Adaptive Temporal Averaging Method”, *Radioengineering*, vol. 23, núm. 2, pp. 652–658, 2014.
- [19] S. Mahmoudpour y M. Kim, “Robust foreground detection in sudden illumination change”, *Electron. Lett.*, vol. 52, núm. 6, pp. 441–443, 2016.
- [20] R. Kalsotra, “Morphological based Moving Object Detection with Background Subtraction Method”, en *4th IEEE International Conference on Signal Processing, Computing and Control (ISPCC)*, 2017, pp. 305–310.
- [21] M.-H. Hung, J.-S. Pan, y C.-H. Hsieh, “A Fast Algorithm of Temporal Median Filter for Background Subtraction”, *J. Inf. Hiding Multimed. Signal Process.*, vol. 5, núm. 1, pp. 33–40, 2014.
- [22] G. M. Rao, “Object Tracking System Using Approximate Median Filter, Kalman Filter and Dynamic Template Matching”, *Int. J. Intell. Syst. Appl.*, vol. 6, núm. 5, pp. 83–89, 2014.
- [23] F. A. Díaz González y D. A. Arévalo Suárez, “Recursive median filter for background estimation and foreground segmentation in surveillance videos”, *Comput. y Sist.*, vol. 19, núm. 2, pp. 283–293, 2015.
- [24] O. El Harrouss, D. Moujahid, y H. Tairi, “Motion detection based on the combining of the background subtraction and spatial color information”, en *2015 Intelligent Systems and Computer Vision, ISCV 2015*, 2015, pp. 2–5.
- [25] W. Hossain y M. N. Das, “Moving Object Detection in Dynamic Backgrounds for Surveillance Systems”, en *IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2014, núm. 978, pp. 1476–1479.
- [26] K. K. Delibasis, T. Goudas, y I. Maglogiannis, “A novel robust approach for handling illumination changes in video segmentation”, *Eng. Appl. Artif. Intell.*, vol. 49, pp. 43–60, 2016.
- [27] K. Wang, Y. Liu, C. Gou, y F. Wang, “A Multi-view Learning Approach to Foreground Detection for Traffic Surveillance Applications”, *IEEE Trans. Veh. Technol.*, vol. 65, núm. 6, pp. 4144–4158, 2016.
- [28] F. Feiran, F. Ming, y Y. Huamin, “Temporal difference method based on positive and negative energy distribution in moving objects detection”, en *2017 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2017, pp. 1–5.
- [29] Y. T. Chan, S. J. Wang, y C. H. Tsai, “Real-time foreground detection approach based on adaptive ensemble learning with arbitrary algorithms for changing environments”, *Inf. Fusion*, vol. 39, pp. 154–167, 2018.
- [30] Z. Chen y T. Ellis, “A self-adaptive Gaussian mixture model”, *Comput. Vis. Image Underst.*, vol. 122, núm. 1, pp. 35–46, 2014.
- [31] C. Spampinato, S. Palazzo, y I. Kavasidis, “A texton-based kernel density estimation approach for background modeling under extreme conditions”, *Comput. Vis. Image Underst.*, vol. 122, pp. 74–83, 2014.
- [32] M. I. Chacon-Murguía y D. Urias-Zavala, “A DTCNN Approach on Video Analysis: Dynamic and Static Object Segmentation”, *Stud. Comput. Intell.*, vol. 551, pp. 315–336, 2014.
- [33] 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, núm. a, pp. 990–1000, 2016.
- [34] J. A. Ramírez-Quintana y M. I. Chacon-Murguía, “An Adaptive Unsupervised Neural Network Based on Perceptual Mechanism for Dynamic Object Detection in Videos with Real Scenarios”, *Neural Process. Lett.*, vol. 42, núm. 3, pp. 665–689, 2015.
- [35] J. A. Ramírez-Quintana y M. I. Chacon-Murguía, “Self-adaptive SOM-CNN neural system for dynamic object detection in normal and complex scenarios”, *Pattern Recognit.*, vol. 48, núm. 4, pp. 1133–1145, 2015.
- [36] Z. Zhao, X. Zhang, Y. Fang, y S. Member, “Stacked Multi-layer Self-Organizing Map for Background Modeling”, *IEEE Trans. IMAGE Process.*, vol. 7149, núm. c, pp. 1–10, 2015.
- [37] M. J. Shafiee, P. Siva, P. Fieguth, y A. Wong, “Embedded Motion Detection via Neural Response Mixture Background Modeling”, en *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 19–26.
- [38] M. Braham y M. Van Droogenbroeck, “Deep background subtraction with scene-specific convolutional neural networks”, en *International Conference on Systems, Signals, and Image Processing*, 2016, pp. 3–6.
- [39] B. Heo, K. Yun, y J. Y. Choi, “Appearance and motion based deep learning architecture for moving object detection in moving camera”, en *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 1827–1831.
- [40] L. A. Lim y H. Y. Keles, “Foreground Segmentation Using a Triplet Convolutional Neural Network for Multiscale Feature Encoding”, *Pattern Recognit. Lett.*, vol. 1, pp. 1–14, 2018.
- [41] Y. Wang, Z. Luo, y P. M. Jodoin, “Interactive deep learning method for segmenting moving objects”, *Pattern Recognit. Lett.*, vol. 0, pp. 1–10, 2016.
- [42] M. Babaee, D. T. Dinh, y G. Rigoll, “A Deep Convolutional Neural Network for Background Subtraction”, *Comput. Res. Repos.*, vol. abs/1702.0, pp. 1–28, 2017.
- [43] L. Yang, J. Li, S. Member, Y. Luo, y S. Member, “Deep Background Modeling Using Fully Convolutional Network”, *IEEE Trans. Intell. Transp. Syst.*, vol. 19, núm. 1, pp. 1–9, 2017.
- [44] D. Sakkos, H. Liu, J. Han, L. Shao, y L. S. Org, “End-to-end video background subtraction with 3d convolutional neural networks”, *Multimed. Tools Appl.*, vol. 77, núm. 17, pp. 23023–23041, 2017.
- [45] R. Guo y H. Qi, “Partially-sparse restricted boltzmann machine for background modeling and subtraction”, en *2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, 2013, pp. 209–214.
- [46] M. Giordano y M. De Gregorio, “Background Modeling by Weightless Neural Networks”, *New Trends Image Anal. Process. -- ICIAP 2015 Work. Lect. Notes Comput. Sci.*, vol. 9281, núm. 1, pp. 493–501, 2015.
- [47] B. N. Subudhi, S. Ghosh, y A. Ghosh, “Application of Gibbs–Markov random field and Hopfield-type neural networks for detecting moving objects from video sequences captured by static camera”, *Soft Comput.*, vol. 19, núm. 10, pp. 2769–2781, 2015.
- [48] S.-C. Huang y B.-H. Do, “Radial Basis Function Based Neural Network for Motion Detection in Dynamic Scenes.”, *IEEE Trans. Cybern.*, vol. 44, núm. 1, pp. 114–125, 2013.
- [49] S. Lee, N. Kim, K. Jeong, I. Paek, H. Hong, y J. Paik, “Multiple moving object segmentation using motion orientation histogram in adaptively partitioned blocks for high-resolution video surveillance systems”, *Optik (Stuttg.)*, vol. 126, núm. 19, pp. 2063–2069, 2015.
- [50] L. Gervasoni y R. Barbuzza, “Aplicación de un método de sustracción de fondo a partir de imágenes de vídeo-vigilancia”, en *15th Argentine Symposium on Technology*, 2014, pp. 25–36.
- [51] J. Dou y J. Li, “Moving object detection based on improved VIBE and graph cut optimization”, *Opt. - Int. J. Light Electron Opt.*, vol. 124, núm. 23, pp. 6081–6088, 2013.
- [52] O. Elharrouss, A. Abbad, D. Moujahid, y H. Tairi, “Moving object detection zone using a blockbased background model”, *IET Comput. Vis.*, vol. 12, núm. 1, pp. 86–94, 2017.
- [53] D. Yang, C. Zhao, X. Zhang, y S. Huang, “Background Modeling by Stability of Adaptive Features in Complex Scenes”, *IEEE Trans. Image Process.*, vol. 27, núm. 3, pp. 1112–1125, 2018.
- [54] A. Shahbaz y K. Jo, “Exploiting Color Spaces for the Task of Foreground Detection”, en *17th International Conference on Control, Automation and Systems (ICCAS2017)*, 2017, núm. 3, pp. 1343–1345.
- [55] M. Chen, Q. Yang, Q. Li, G. Wang, y M. Yang, “Spatiotemporal Background Subtraction Using Minimum Spanning Tree and Optical Flow”, en *Computer Vision–ECCV 2014*, 2014, pp. 521–534.
- [56] S. Jiang y X. Lu, “WeSamBE: A Weight-Sample-Based Method for

- Background Subtraction”, *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, núm. 99, pp. 1–11, 2017.
- [57] H. Sajid y S.-C. S. Cheung, “Background Subtraction for Static & Moving Camera”, en *International Conference on Image Processing (ICIP)*, 2015, pp. 4530–4534.
- [58] H. Sajid y S.-C. S. Cheung, “Universal Multimode Background Subtraction”, *IEEE Trans. Image Process.*, vol. 26, núm. 7, pp. 3249–3260, 2017.
- [59] S. Wangsiripitak y W. Rattanapitak, “CDoTS : Change Detection on Time Series Background for Video Foreground Segmentation”, en *14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) CDoTS.*, 2017, pp. 400–403.
- [60] B. W. and Piotr, “AMBER: adapting multi-resolution background extractor”, en *Image Processing (ICIP), 2013 20th IEEE International Conference on*, 2013, pp. 3417–3421.
- [61] Z. Zhong, B. Zhang, G. Lu, Y. Zhao, y Y. Xu, “An Adaptive Background Modeling Method for Foreground Segmentation”, *IEEE Trans. Intell. Transp. Syst.*, vol. 18, núm. 5, pp. 1109–1121, 2017.
- [62] J. D. Romero, M. J. Lado, y A. J. Mendez, “A Background Modeling and Foreground Detection Algorithm Using Scaling Coefficients Defined with a Color Model Called Lightness-Red-Green-Blue”, *IEEE Trans. Image Process.*, vol. 27, núm. 3, pp. 1243–1258, 2017.
- [63] J. Pan, W. Chen, y W. Peng, “A new moving objects detection method based on improved SURF algorithm”, en *2013 25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 901–906.
- [64] J. Dou y J. Li, “Modeling the background and detecting moving objects based on Sift flow”, *Optik (Stuttg.)*, vol. 125, núm. 1, pp. 435–440, 2014.
- [65] F. J. López-Rubio y E. López-Rubio, “Foreground detection for moving cameras with stochastic approximation”, *Pattern Recognit. Lett.*, vol. 68, pp. 161–168, 2015.
- [66] H. Lu, Yingying Chen, Jinqiao Wang, “Learning Sharable Models For Robust Background Subtraction”, en *IEEE International Conference on Multimedia and Expo (ICME)*, 2015, pp. 1–6.
- [67] R. Azzam, M. S. Kemouche, N. Aouf, y M. Richardson, “Efficient visual object detection with spatially global Gaussian mixture models and uncertainties”, *J. Vis. Commun. Image Represent.*, vol. 36, núm. C, pp. 90–106, 2016.
- [68] P.-L. St-Charles, G.-A. Bilodeau, y R. Bergevin, “SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity”, *IEEE Trans. Image Process.*, vol. 24, núm. 1, pp. 359–373, 2015.
- [69] M. Radolko, F. Farhadifard, E. Gutzeit, y U. F. Von Lukas, “Real time video segmentation optimization with a modified Normalized Cut”, en *9th International Symposium on Image and Signal Processing and Analysis, ISPA 2015*, 2015, pp. 31–36.
- [70] P. Chiranjeevi y S. Sengupta, “Detection of moving objects using multi-channel kernel fuzzy correlogram based background subtraction”, *IEEE Trans. Cybern.*, vol. 44, núm. 6, pp. 870–881, 2014.
- [71] D. K. Panda y S. Meher, “Detection of Moving Objects Using Fuzzy Color Difference Histogram Based Background Subtraction”, *IEEE Signal Process. Lett.*, vol. 23, núm. 1, pp. 45–49, 2016.
- [72] B. Yang y L. Zou, “Robust foreground detection using block-based RPCA”, *Optik (Stuttg.)*, vol. 126, núm. 23, pp. 4586–4590, 2015.
- [73] C. Guyon, T. Bouwmans, y E. H. Zahzah, “Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint”, *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7728 LNCS, núm. PART 1, pp. 315–320, 2013.
- [74] Y. Mo, H. Xu, y X. Ding, “Improved OR-PCA for Robust Foreground Detection”, en *Chinese Automation Congress (CAC)*, 2017, pp. 5566–5571.
- [75] J. Feng y H. Xu, “Online Robust PCA via Stochastic Optimization”, *Adv. Neural Inf. Process. Syst.*, vol. 26, pp. 1–9, 2013.
- [76] M. Shakeri y H. Zhang, “COROLA: A sequential solution to moving object detection using low-rank approximation”, *Comput. Vis. Image Underst.*, vol. 146, pp. 27–39, 2015.
- [77] X. Zhou, C. Yang, y W. Yu, “Moving object detection by detecting contiguous outliers in the low-rank representation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, núm. 3, pp. 597–610, 2013.
- [78] X. Cao, L. Yang, y X. Guo, “Total Variation Regularized RPCA for Irregularly Moving Object Detection Under Dynamic Background”, *IEEE Trans. Cybern.*, vol. 46, núm. 4, pp. 1014–1027, 2016.
- [79] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, y S. K. Jung, “Background Subtraction via Superpixel-Based Online Matrix Decomposition with Structured Foreground Constraints”, en *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 930–938.
- [80] P. L. St-Charles, G. A. Bilodeau, y R. Bergevin, “A self-adjusting approach to change detection based on background word consensus”, en *IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 2015, pp. 990–997.
- [81] B. Zhang, D. Jiao, H. Pei, Y. Gu, y Y. Liu, “Infrared moving object detection based on local saliency and sparse representation”, *Infrared Phys. Technol.*, vol. 86, pp. 187–193, 2017.
- [82] S. Fan y X. Zeng, “A superpixel-based codebook model for real time foreground detection”, *I2MTC 2017 - 2017 IEEE Int. Instrum. Meas. Technol. Conf. Proc.*, 2017.
- [83] A. Staglianò, N. Noceti, A. Verri, y F. Odone, “Online space-variant background modeling with sparse coding”, *IEEE Trans. Image Process.*, vol. 24, núm. 8, pp. 2415–2428, 2015.
- [84] P. Dong, S. Wang, Y. Xia, D. Liang, y D. D. Feng, “Foreground detection with simultaneous dictionary learning and historical pixel maintenance”, *IEEE Trans. Image Process.*, vol. 25, núm. 11, pp. 5035–5049, 2016.
- [85] C. David y V. Gui, “Sparse Coding and Gaussian Modeling of Coefficients Average for Background Subtraction”, en *2013 8th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2013, pp. 223–228.
- [86] J. Šoda, I. Kuzmanić, y I. Vujović, “Stabilising illumination variations in motion detection for surveillance applications”, *IET Image Process.*, vol. 7, núm. 7, pp. 671–678, 2013.
- [87] M. N. A. F. M. A. S. M. H. M. Ebeid, y M. F. Tolba, “Wavelet-Enhanced Detection of Small / Slow Object Movement in Complex Scenes”, en *2016 11th International Conference on Computer Engineering & Systems (ICCES)*, 2016, pp. 172–180.
- [88] S. Li, D. Florencio, Y. Zhao, C. Cook, y W. Li, “Foreground Detection in Camouflaged Scenes”, en *IEEE International Conference on Image Processing*, 2017, pp. 4247–4251.
- [89] R. Wang, F. Bunyak, G. Seetharaman, y K. Palaniappan, “Static and moving object detection using flux tensor with split gaussian models”, en *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 420–424.
- [90] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, y S. I. Amari, “Bayesian Robust Tensor Factorization for Incomplete Multiway Data”, *IEEE Trans. Neural Networks Learn. Syst.*, vol. 27, núm. 4, pp. 736–748, 2016.
- [91] S. Bianco, G. Ciocca, y R. Schettini, “How Far Can You Get By Combining Change Detection Algorithms?”, *IEEE Trans. Image Process.*, vol. 2, pp. 1–10, 2015.
- [92] P. Tiefenbacher, M. Hofmann, D. Merget, y G. Rigoll, “PID-based regulation of background dynamics for foreground segmentation”, en *IEEE International Conference on Image Processing, ICIP*, 2014, pp. 3282–3286.
- [93] B. Dey y M. K. Kundu, “Efficient Foreground Extraction from HEVC Compressed Video for Application to Real-Time Analysis of Surveillance ‘Big’ Data”, *IEEE Trans. Image Process.*, vol. 24, núm. 11, pp. 3574–3585, 2015.
- [94] Z. Zivkovic y F. Van Der Heijden, “Efficient adaptive density estimation per image pixel for the task of background subtraction”, *Pattern Recognit. Lett.*, vol. 27, núm. 7, pp. 773–780, 2006.
- [95] Z. Chen y T. Ellis, “Self-adaptive Gaussian mixture model for urban traffic monitoring system”, en *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1769–1776.

- [96] Y. LeCun, L. Bottou, Y. Bengio, y P. Haffner, "Gradient-based learning applied to document recognition", *Proc. IEEE*, vol. 86, núm. 11, pp. 2278–2323, 1998.
- [97] M. De Gregorio, I. Ciberetica, E. C. I. Cnr, M. Giordano, P. Icar, y V. P. Castellino, "Change Detection with Weightless Neural Networks", en *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 409–413.
- [98] T. Ojala, M. Pietikäinen, y T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, núm. 7, pp. 971–987, 2002.
- [99] H. Bay, T. Tuytelaars, y L. Van Gool, "SURF: Speeded up robust features", en *European Conference on Computer Vision- ECCV*, 2006, pp. 404–417.
- [100] C. Liu, J. Yuen, y A. Torralba, "Sift flow: Dense correspondence across scenes and its applications", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, núm. 5, pp. 978–994, 2011.
- [101] C. Pojala y S. Sengupta, "Detection of moving objects using fuzzy correlogram based background subtraction", en *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011, pp. 255–259.
- [102] G.-H. Liu y J.-Y. Yang, "Content-based image retrieval using color difference histogram", *Pattern Recognit.*, vol. 46, núm. 1, pp. 188–198, 2013.
- [103] M. Aharon, M. Elad, y A. M. Bruckstein, "K-Svd: Design of Dictionaries for Sparse Representation", *Spars'05*, vol. 1, pp. 9–12, 2005.
- [104] T. G. Kolda y B. W. Bader, "Tensor Decompositions and Applications", *SIAM Rev.*, vol. 51, núm. 3, pp. 455–500, 2009.
- [105] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, y P. Ishwar, "CDnet 2014: An Expanded Change Detection Benchmark Dataset", *2014 IEEE Conf. Comput. Vis. Pattern Recognit. Work.*, pp. 393–400, 2014.



Abimael Guzmán Pando nació en Chihuahua, Chihuahua, México en 1993. Obtuvo el grado de Ingeniero Electrónico, 2016, y Maestro en Ciencias en Ingeniería Electrónica, 2018, en el Instituto Tecnológico de Chihuahua, Chih., México. Actualmente, es estudiante de doctorado en el Instituto Tecnológico de Chihuahua. Sus investigaciones incluyen

áreas enfocadas en sistemas de visión por computadora, procesamiento digital de señales, reconocimiento de patrones, entre otras.



Mario Ignacio Chacón Murguía (M'86–SM'04) obtuvo el grado de Ingeniero Industrial en Electrónica, 1982, y el grado de Maestro en Ciencias en Ingeniería Electrónica, 1985 del Instituto Tecnológico de Chihuahua, México, y el grado de Doctor en Ciencias, 1998, de la Universidad Estatal de Nuevo México, EEUU. Ha desarrollado distintos proyectos

para varias compañías. Actualmente trabaja como Profesor Investigador en el Instituto Tecnológico de Chihuahua. Ha publicado más de 175 trabajos y publicado 3 libros. Su investigación actual incluye Visión por Computadora y procesamiento de imágenes y señales usando Inteligencia Computacional. El Dr. Chacón es miembro Senior de la IEEE, y miembro de las sociedades IEEE; Inteligencia computacional, Procesamiento Digital de Señales y Miembro del SNI en México.