

## **Detección y rastreo de peatones empleando vectores de características de histogramas de gradientes orientados y patrones binarios locales en una máquina de soporte vectorial**

Raymundo Rodríguez Romero<sup>1</sup>, Luis Pastor Sánchez Fernández<sup>1</sup>,  
Edgardo Manuel Felipe Riverón<sup>2</sup>, José Juan Carbajal Hernández<sup>1</sup>

<sup>1</sup> Instituto Politécnico Nacional, Centro de Investigación en Computación,  
Ciudad de México, México  
ing.m.rodriguez@gmail.com, {edgardo,lsanchez}@cic.ipn.mx, carbajalito@hotmail.com

**Resumen.** La detección e identificación de peatones, así como su rastreo, es un problema abierto en el área de la visión artificial que se ha atacado de formas muy diversas. Este trabajo presenta una propuesta para la detección de peatones utilizando técnicas de análisis de imágenes para la extracción de características de un peatón basadas en histograma de gradientes orientados y patrones binarios locales los cuales, son extraídos de una región de interés previamente detectada, combinados e ingresados a una máquina de soporte vectorial que funge como clasificador entre muestras de peatones y no peatones. Un rastreador basado en el algoritmo *MeanShift* es implementado por su eficiencia en el procesamiento del rastreo y la identificación de las nuevas ubicaciones de cada uno de los objetivos detectados.

**Palabras clave:** detección de objetos, gradientes orientados, patrones binarios, MeanShift, soporte vectorial.

### **Detection and Tracking of Pedestrians using Feature Vectors of Oriented Gradient Histograms and Local Binary Patterns using a Support Vector Machine**

**Abstract.** The detection and identification of pedestrians, as well as their tracking, is an open problem in the area of artificial vision that has been worked in many different ways. This paper presents a proposal for the detection of pedestrians using image analysis techniques for the extraction of characteristics of a pedestrian based on histogram of oriented gradients and local binary patterns which are extracted from a region of interest previously detected, combined and entered to a vector support machine that functions as a classifier between samples of pedestrians and non-pedestrians. A tracker based on the MeanShift algorithm is implemented for its efficiency in tracking processing and identifying the new locations of each of the detected targets.

**Keywords:** object detection, oriented gradients, binary patterns, MeanShift, vector support.

## 1. Introducción

La detección y el rastreo de objetos han representado un reto en el área de la visión artificial desde hace varias décadas, siendo actualmente de los más estudiados gracias al desarrollo de hardware más potente que permite un mejor desempeño en las tareas necesarias para llevar a cabo la detección de los objetos de interés y su rastreo.

En este trabajo se aborda una propuesta de solución a la detección de objetos que excluye aquellos que no correspondan con la descripción de un peatón y posteriormente llevar a cabo el rastreo del objetivo utilizando una caja de contenido.

La detección de objetos en una toma aleatoria ha sido estudiada empleando diferentes algoritmos, generalmente sustrayendo el fondo con diversas técnicas de análisis de imágenes y modelos de color, por ejemplo, Leykin [9] presentó una propuesta en el 2010 empleando un modelo basado en la temperatura del color para la sustracción del fondo y la detección de los objetos visibles. Por otro lado, Zheng et al. [1] trabajó en el 2017 una propuesta de actualización continua empleando un modelo de color RBG y aplicando promedios sobre cada pixel, de forma tal que, al evaluar una imagen con un objeto móvil, dicho modelo genere una diferencia probabilística sobre los pixeles en el fotograma evaluado. En el 2016 Lacabex et al. [5] realizó la detección de objetos en su sistema empleando una resta directamente entre el fondo y los fotogramas actuales en un modelo RGB, aplicando posteriormente dos umbrales en cada canal de color: uno bajo y uno alto; tras lo que aplicó una operación de correlación entre ambos.

Existen otros trabajos que no realizan una búsqueda de objetos en la imagen, sino que evalúan directamente la existencia de peatones, como en el caso de Yi et al. [2] donde se empleó una búsqueda por contorno y formas a través de un algoritmo de máquina de estados algorítmica (ASM) [11] y un modelo probabilístico basado en una distribución Gaussiana.

Lim y Kim presentaron en el 2013 una detección de peatones basada en la extracción de características empleando el histograma de gradientes orientados [6] (*Histogram of Oriented Gradient*, HOG por sus siglas en inglés) para posteriormente realizar la clasificación utilizando el algoritmo AdaBoost propuesto por Viola y Jones [10]. Por su parte, Li et al. [7] realizaron un trabajo con algunas similitudes al de Lim et al., sin embargo, la principal diferencia fue el uso de una máquina de soporte vectorial (*Support Vector Machine*, SVM por sus siglas en inglés) las cuales se basan en la construcción de hiperplanos que separan las muestras en clases linealmente separables utilizando los llamados vectores de soporte.

Existen diversas propuestas para realizar el rastreo de un objetivo; entre aquellas que se basan en peatones tenemos ejemplos como: el trabajo de Chen et al. [8] en el cual el rastreo se realiza en cada fotograma con la detección continua de objetivos y la diferencia entre ambos para determinar las nuevas ubicaciones a través de una detección de orientación de movimiento. Por otra parte, el trabajo de Li et al. [7] también incluye un rastreador basado en una combinación entre los algoritmos de rastreo Camshift [13] y el filtro Kalman.

En nuestra propuesta, se aplica una separación de los canales de color empleando los canales de intensidad, luminiscencia y los tres canales RGB extrayendo la información útil de cada canal, realizando un umbralado y sumándola posteriormente para obtener los objetos existentes en la toma.

Además, se empleó una clasificación entre peatones y no peatones utilizando un clasificador SMV, con la diferencia de que es alimentado por vectores de características HOG y patrones locales binarios (*Local Binary Patterns*, LBP por sus siglas en inglés) [4] los cuales describen las características de textura en una imagen. Finalmente se utilizó un algoritmo *MeanShift* [12] para llevar a cabo el rastreo de los peatones encontrados, basado en la búsqueda por ventana de histograma de color que describan a los objetivos y su centroide.

## 2. Metodología

El modelo de solución se basa en diversas etapas secuenciales que corresponden a los procesos necesarios para segmentar las características, de forma que el sistema pueda detectar y rastrear a los objetivos en los fotogramas procesados mostrado en la Fig. 1.

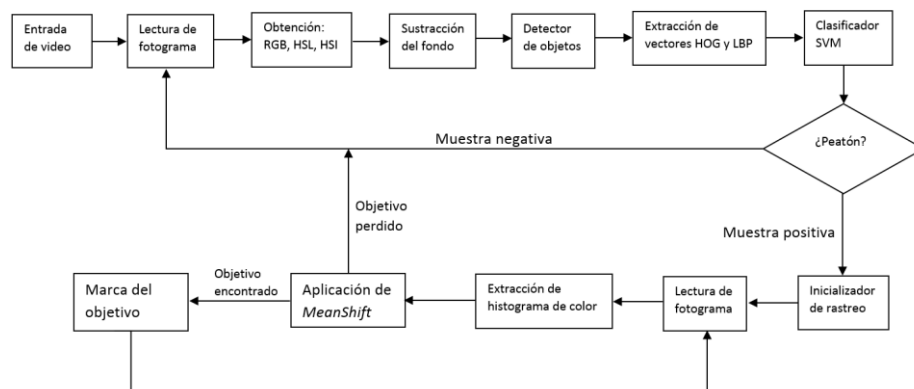


Fig. 1. Diagrama de flujo del algoritmo de rastreo.

### 2.1. Detección de peatones

Empleando la lectura de los fotogramas de video se obtienen los canales HSL, HSI y RGB (Fig. 2) mediante los cuales se realizan diversas operaciones de análisis de imágenes y de morfología matemática para la detección de peatones.

El modelo propone tener siempre la información sobre el fondo de la imagen, de forma que se pueda optimizar el proceso de detección de objetos.

Para identificar diferentes características en la toma de acuerdo a los cambios de luz y color de los objetos presentes en la toma extraemos los canales de luminiscencia, intensidad, RGB y su conversión a escala de grises.



**Fig. 2.** a) Fondo RGB. b) Fondo a escala de grises. c) Fondo en canal de luminiscencia. d) Fondo en canal de intensidad.

Se emplea una operación para obtener la media del fondo en cada uno de los canales propuestos cuando no se han detectado objetivos para el rastreo (1), lo que permite la reducción del ruido generado por cambios de iluminación:

$$\mu_t = \frac{\sum_{j=0}^n F_j}{n}, \quad (1)$$

donde  $\mu_t$  representa la media del fondo a través del tiempo,  $\sum_{j=0}^{j-1} F_j$  representa la sumatoria del conjunto de fotogramas analizados donde no se ha detectado ningún objeto y  $n$  es el número total de fotogramas analizados que cumplen esta condición.

Para realizar la detección de los objetos en el fotograma actual se sustrae la media del fondo a la imagen actual, aplicamos valor absoluto al resultado (2) y finalmente se umbrala el contenido de la imagen obteniéndose una imagen binaria (3):

$$T_j = |F_j - \mu_t|, \quad (2)$$

$$T_j = \{255 \text{ Si } p(x, y) \geq u \text{ 0 Si } p(x, y) < u, \quad (3)$$

donde  $F_j$  es el valor del fotograma actual,  $\mu_t$  define el promedio del fondo,  $T_j$  es el valor del fotograma resultado de la resta, mientras que  $p(x, y)$  representa cada uno de los pixeles presentes en  $T_j$ .

Aplicando la Ec. (3), obtenemos objetos más cercanos a los presentes dentro de la escena en cada uno de los canales de color procesado, los cuales serán almacenados en un único fotograma realizando la unión lógica de las 4 imágenes binarias como se observa en la Fig. 3, donde  $A_j$ ,  $B_j$ ,  $C_j$ , y  $D_j$  son los planos binarios y  $U_j$  es la unión de todos ellos (4).

$$U_i = A_j \cup B_j \cup C_j \cup D_j. \quad (4)$$

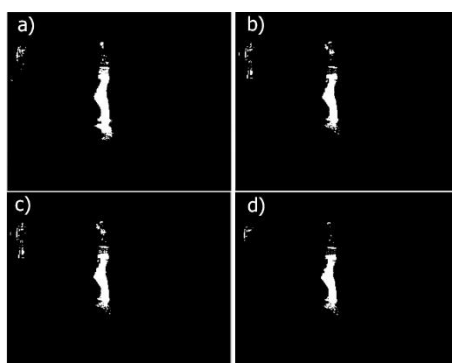
Para eliminar las partículas que no son parte de los objetivos detectados sino pequeños segmentos de ruido presentes en la toma, utilizamos un operador morfológico de apertura (7),

$$A_D = A \oplus B, \quad (5)$$

$$A_E = A \ominus B, \quad (6)$$

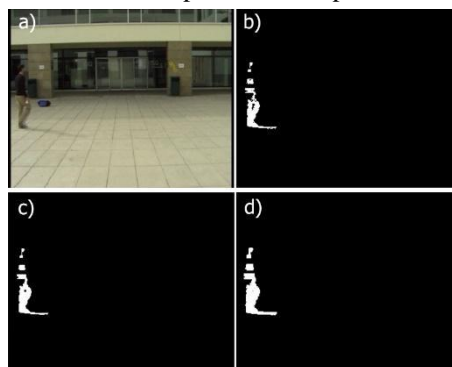
$$A \circ B = (A \oplus B) \ominus B, \quad (7)$$

donde  $A$  representa al conjunto de píxeles de la imagen procesada,  $B$  es el elemento de estructura seleccionado (un cuadrado de 3x3 píxeles), La Ec. (5) muestra la operación de dilatación del elemento de estructura sobre la imagen binaria y la Ec. (6) representa la operación de erosión del elemento de estructura sobre la imagen dilatada.



**Fig. 3.** a) Objetos obtenidos desde RGB. b) Objetos obtenidos desde la escala de grises. c) Objetos obtenidos desde el canal de luminiscencia. d) Objetos obtenidos desde el canal de intensidad.

Finalmente, la operación de dilatación extra es aplicada con el fin de ampliar los objetos detectados y tener una mejor percepción de ellos durante la selección de la región de interés, el resultado de estas operaciones se puede observar en la Fig. 4.



**Fig. 4.** a) Fotograma actual b) Unión de imágenes binarias c) Apertura morfológica. d) Dilatación morfológica con elemento de estructura cuadrado de 3x3 píxeles.

Con los datos obtenidos tras las operaciones morfológicas se tiene una descripción de los objetos detectados, lo cual permite obtener regiones de interés (*Region of interest*, ROI, por sus siglas en inglés) sobre las cuales se determina si han sido o no identificados los peatones entrantes a la toma. Las ROI se utilizan para analizar los

objetos detectados y extraer sus características mediante vectores dados por el histograma de gradientes orientados y los patrones binarios locales.

### 2.1.1. Extracción de características HOG

El histograma de gradientes orientados se basa en el supuesto de que toda imagen tendrá los mismos cambios en su iluminación sin importar si ésta es clara u oscura, además, tiene un funcionamiento eficiente cuando se evalúan imágenes con cambio de escala; gracias a que otorga una descripción de los gradientes dentro de la imagen y la dirección de estos.

Dada una imagen en escala de grises dividida en celdas (para este caso contempladas como 8x8) se determina el gradiente computacional descrito por las Ecs. (8) y (9):

$$G_x(x, y) = [-1, 0, 1] \cdot [I(x-1, y), I(x, y), I(x+1, y)]^T, \quad (8)$$

$$G_y(x, y) = [-1, 0, 1] \cdot [I(x, y-1), I(x, y), I(x, y+1)]^T. \quad (9)$$

La magnitud de cada pixel está definida por la ec. (10):

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \quad (10)$$

mientras su dirección está dada de la siguiente forma (11):

$$\varphi(x, y) = \arctan \left[ \frac{G_y(x, y)}{G_x(x, y)} \right]. \quad (11)$$

Convencionalmente se utilizan 9 contenedores que dispondrán de todos los ángulos entre 0 y 180 grados. Cuando el ángulo de un pixel entra dentro del rango del contenedor, la magnitud de este es sumada al mismo y contemplada como 0 para todos los demás, tal y como lo describe la Ec. (12):

$$hog_i = \{G(x, y)\varphi(x, y) \in contenedor(i) \mid 0 \notin contenedor(i)\}. \quad (12)$$

Agregamos cada uno de los valores obtenidos a los vectores de características de cada una de las celdas dentro de toda la imagen. Para poder tener una descripción correcta de todos los cambios de iluminación y contraste, es necesario normalizar localmente, por lo que debemos agrupar las celdas en grupos de mayor tamaño. Dado lo anterior, el descriptor por bloques está dado por la concatenación de los histogramas normalizados generados en cada celda. Estos bloques generalmente suelen sobreponerse lo que significa que cada una de las celdas podrá ser considerada en más de una ocasión para el descriptor final (Ec. 13):

$$hog_B = \frac{hog_B}{\sqrt{\|hog_B\|^2 + \epsilon}}, \quad (13)$$

donde  $hog_B$  es el descriptor de cada bloque y  $\epsilon$  representa un pequeño valor constante (teóricamente no importa cuál) agregado para evitar problemas numéricos en la raíz cuadrada.

Entonces, el descriptor final estará dado por la concatenación de cada uno de los bloques dentro de los 9 contenedores.

### 2.1.2. Extracción de características LBP

Por su parte, LBP describe las características de textura encontradas en un objeto analizando los píxeles de forma local y otorga valores de peso en aquellos que tengan una relación de valor muy cercana y a su vez aleja otros que no tengan ninguna relación o un parentesco.

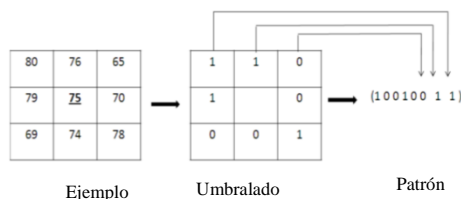


Fig. 5. Cálculo de LBP.

Acorde al ejemplo anterior (Fig. 5) podemos determinar el valor del pixel central con una máscara de 3x3 realizando un barrido con sus 8 vecinos otorgando un peso binario a cada uno de ellos, con lo que para el caso anterior 10010011 estaría definido como  $LBP = (10010011)_2 = 147$ .

De esta forma podemos definir la textura de una imagen como la distribución conjunta de sus niveles de gris en una área local definida por la equidistancia entre el pixel central ( $R$ ) y los vecinos contemplados ( $P$ ) (Fig. 6) que está dada por la Ec. (14).

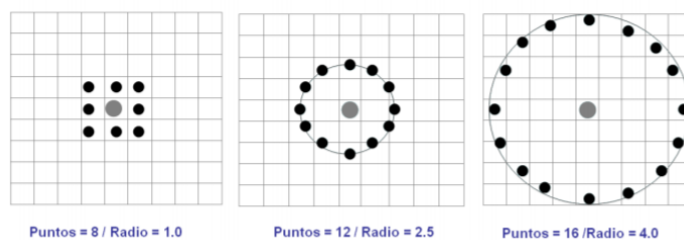


Fig. 6. Simetría circular para conjuntos de vecinos [3].

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p, \quad (14)$$

donde  $s$  es una transformación definida por el signo derivado de la diferencia entre  $g_c$  y  $g_p$  los cuales representan al pixel central evaluado y a cada uno de los píxeles vecinos contemplados para la transformación LBP respectivamente, mientras que  $2^p$  es el peso asignado a cada uno de los vecinos.

Una vez obtenida la nueva imagen definida por la transformación LBP se aplican mallas en las direcciones  $X$  y  $Y$ , y se divide la imagen en secciones dentro de las cuales obtendremos los histogramas LBP que serán utilizados como entradas para las SVM.

### 2.1.3. Clasificación utilizando SVM

Las máquinas de soporte vectorial (SVM) son clasificadores que permiten trabajar con muestras linealmente separables, como en nuestro caso se tienen dos tipos de

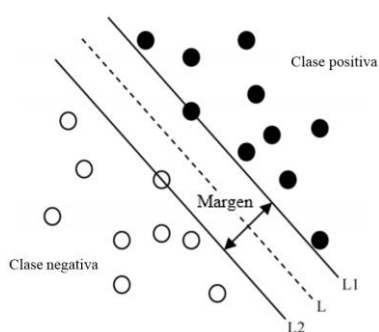
objetos (peatón o no peatón). Este tipo de clasificador nos permite separar el problema muy fácilmente con las muestras adecuadas.

El clasificador está dado por la relación de las muestras etiquetadas (Ec. 15):

$$\psi_{1:n} = \{(h_1, y_1), (h_2, y_2), \dots, (h_n, y_n)\}, \tag{15}$$

donde  $(h_i, y_i) \in \psi_{1:n}$  y  $y_i \in \{-1, 1\}$ ; el conjunto  $\psi_{1:n}$  es linealmente separable si existe un vector que satisfaga la Ec. (16):

$$y_i(w \otimes h_i + b) \geq 1. \tag{16}$$



**Fig. 7.** Clasificador máquina de soporte vectorial de dos clases [7].

Dado lo anterior y como se observa en la Fig. 7 se puede construir una matriz de confusión (Tabla 1) que describa el problema de clasificación en las muestras evaluadas por la red neuronal una vez realizado el entrenamiento con muestras positivas.

**Tabla 1.** Matriz de confusión de la SVM para la detección de peatones.

Resultado de la clasificación	Muestra – No peatón	Muestra – Peatón
No peatón	Verdadero Negativo	Falso Positivo
Peatón	Falso Negativo	Verdadero Positivo

Para el entrenamiento de la SVM se utilizan los vectores de características descritos anteriormente, realizando una concatenación entre ambos descriptores para obtener un solo vector de características, el cual contiene toda la información sobre las muestras ingresadas a la red neuronal. La Fig. 8 muestra las ROI identificadas en el objeto, las cuales nos permiten identificar las zonas para el análisis de la SVM.





**Fig. 8.** a) Objetos encontrados para el análisis de las ROI. b) Peatón detectado con los descriptores HOG y LBP.

## 2.2. Rastreo de peatones

Para realizar el rastreo, una vez encontrados los objetivos que cumplen con las características de ser un peatón, se utiliza un algoritmo *MeanShift*, que se basa en la estimación del kernel de densidad en el histograma de color, el cual está conformado por los niveles obtenidos en el canal de matiz (*Hue* o *H* por su sigla en inglés) de la imagen, para este caso se optó por emplear 64 contenedores para optimizar la velocidad de procesamiento.

Se emplea un fotograma anterior para determinar la posición en el fotograma actual a través de un análisis de la densidad del kernel, el cual se basa en la obtención de los valores de distancia (ponderados) en un área de análisis circular a partir de una posición central determinada aleatoriamente en un área de búsqueda en una posición cercana a la original. De esta forma determina el centroide de la densidad actual, lo que muestra el desplazamiento del objetivo entre un fotograma y otro.

Se obtiene la nueva ubicación del objetivo aplicando iteraciones hasta encontrar el máximo local dentro de la nueva imagen y asignando la nueva posición dada por la Ec. (17),

$$\Delta P = \frac{\sum K(P - P_{ant})w(P)(P - P_{ant})}{\sum K(P - P_{ant})w(P)}, \quad (17)$$

donde,  $P_{ant}$  y  $P$  representan la posición anterior y actual respectivamente y  $w$  describe el cambio en la región de interés del candidato y el objetivo generado por los histogramas de color de la Ec. (18),

$$w(P) = \sqrt{\frac{h_{obj}(I(P))}{h_c(I(P))}}, \quad (18)$$

donde  $h_{obj}$  y  $h_c$  son los histogramas de color del objetivo y del candidato, respectivamente.

La Fig. 9 muestra cuando un objetivo es detectado dentro de la ventana de búsqueda, la cual actualiza la posición de la caja de contenido y avanza al siguiente fotograma para verificar que no existan nuevos objetivos dentro de la toma.



Fig. 9. Rastreo de un objetivo con cambios de iluminación.

### 3. Resultados experimentales

Las distintas secciones del algoritmo fueron probadas por separado y de forma completa con la base de datos EPFL *Multi-camera pedestrian video* [], ya que dispone de diferentes entornos para la comprobación de su funcionamiento tanto en iluminación como en texturas y tipos de fondo. Cada escena fue grabada con 3 o 4 cámaras con diferentes ubicaciones de forma paralela, a una altura de 2 metros desde el suelo; los peatones son personal del laboratorio de visión por computadora de la Escuela Politécnica Federal de Lausana, mismo que realizó las grabaciones y compartió la base de datos.

El sistema de rastreo desarrollado es capaz de detectar objetos en cualquier tipo de entorno ya que tiene gran robustez ante cambios de iluminación gracias a que el uso de los canales de luminiscencia e intensidad proporcionan las características de luz ambiental en los fotogramas analizados, mientras que la integración RGB brinda la información necesaria para actualizar el cambio de color para cada región de interés encontrada. La Fig. 10 muestra el detector de objetos aplicado en dos áreas diferentes, una solo iluminada por luz artificial, mientras que en la otra se dispone únicamente de iluminación natural. En ambos casos los objetos presentes son extraídos de forma exitosa.

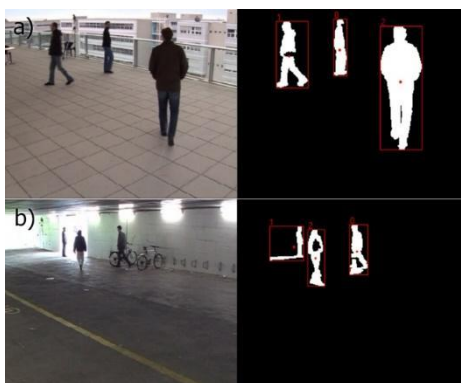


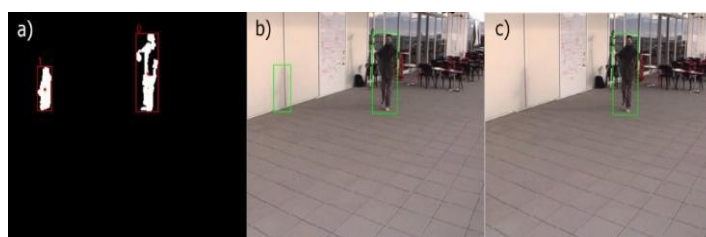
Fig. 10. a) Detección en entorno con luz natural. b) Detección en entorno con luz artificial.

El uso de descriptores de características tales como HOG y LBP es muy común en el área de la visión artificial, ya que los alcances para sus respectivas aplicaciones

pueden ser muy amplios. La combinación de ambos descriptores como entrada para un clasificador tipo SVM (Tabla 2) mejoró en gran medida la detección de falsos positivos en comparación con el uso de un solo descriptor como entrada al clasificador como se observa en la Fig. 11.

**Tabla 2.** Precisión del clasificador SVM con diferentes características de entrada.

Extractor de características	Precisión	Falsos negativos	Falsos positivos
HOG	89%	5%	6%
LBP	78%	14%	8%
HOG y LBP	92%	6%	2%



**Fig. 11.** a) Detección de objetos. b) Falso positivo detectado con HOG. c) Detección de objetivo con HOG y LBP.

La precisión del algoritmo se calculó realizando una detección de peatones en un conjunto de 100 tomas en las cuales ya se tenía identificada la cantidad de peatones presentes en cada toma, se considera correcto solo el caso donde se identifica un peatón correctamente, un falso negativo es el caso donde un peatón no es identificado como tal y un falso positivo es el caso en que un objeto es identificado como peatón (comúnmente dado en sombras).

El algoritmo *MeanShift* tiene muy buen funcionamiento cuando los objetivos se encuentran en entornos contrastantes; sin embargo, tiende a perderlos cuando estos entran en zonas donde la iluminación puede confundirlo con el fondo. En casos extremos incluso se puede perder definitivamente, lo que originaría una nueva búsqueda desde la etapa de detección.

La Fig. 12 muestra la secuencia de rastreo de un objetivo con un fondo iluminado por luz natural, la precisión obtenida en cada una de las secuencias puede verse reflejada en la Tabla 3.



**Fig. 12.** Rastreo de un objetivo único.

**Tabla 3.** Porcentaje de rastreo exitoso en escenas con un solo objetivo.

Secuencia	Precisión
1	93%
2	92%
3	95%

En la Fig. 13 se aprecia el rastreo de varios peatones, apreciando el rastreo correcto cuando se encuentran distanciados y la pérdida por oclusión en otros casos, la precisión obtenida en cada secuencia para este caso puede ser consultada en la Tabla 4.

Se probaron 3 tipos de escenas diferentes de 300 fotogramas para determinar la precisión del rastreo, contemplando el seguimiento desde la entrada hasta la salida de cada objetivo sin la existencia de una nueva detección.



**Fig. 13.** Rastreo multiobjetivo.

**Tabla 4.** Porcentaje de rastreo exitoso en escenas con varios objetivos

Secuencia	Precisión
1	92%
2	90%
3	93%

Este tipo de rastreo permite la ubicación de los objetivos de forma eficiente; sin embargo, al ser dependiente del histograma de color, las oclusiones o encuentros entre objetivos tienden a afectar en gran medida la nueva ubicación de las posiciones, ocasionando pérdida o confusión entre objetivos.

#### 4. Conclusiones y trabajo futuro

La detección de objetos empleando una metodología simbólica en combinación con técnicas de análisis de imágenes, tuvo como resultado un sistema de detección de objetos eficaz y robusto ante cambios de iluminación, lo que permite utilizar el algoritmo en cualquier tipo de fondo, ya sea con iluminación natural o artificial.

Por otro lado, los descriptores de imágenes basados en HOG y LBP permiten la invariancia al nivel de gris y al escalamiento, lo que los ha llevado a ser utilizados

principalmente en la detección de rostros o tipos de texturas; la propuesta ayuda a mejorar la precisión para determinar la existencia de un peatón sobre cualquier tipo de iluminación y excluir objetos que no deben ser rastreados (como automóviles, animales, sombras, etc.). Debido al nivel de confianza que garantiza el clasificador SVM se puede optar por utilizar un algoritmo de rastreo simple que no consume grandes recursos computacionales como lo es *MeanShift*.

*MeanShift* tiene grandes ventajas, tales como un procesamiento rápido basado en el histograma de color y por lo tanto aprovecha toda la información que brinda el modelo RGB. También la búsqueda se basa en un entorno local, lo cual facilita la ubicación de la nueva posición. A pesar de esto, el rastreo tiene puntos a mejorar como la pérdida por oclusión o la confusión de objetivos al confundir dos objetivos con un histograma de color muy parecido.

El sistema puede mejorarse con pruebas de otros algoritmos de rastreo y la inclusión de bibliotecas para el reconocimiento, con el fin de no perder aquellos objetivos que hayan sido detectados al menos una vez anteriormente.

## Referencias

1. Zheng, L., Ruan, X., Chen, Y., Huang, M.: Shadow removal for pedestrian detection and tracking in indoor environments. *Multimedia Tools and Applications* 76(18), 18321–18337 (2017)
2. Yi, X., Mingtao, P., Jianguan, Z., Meng, M., Yunde, J.: Tracking Pedestrian with Incrementally Learned Representation and Classification Model. *Journal of Information Science and Engineering* 30(4), 1035–1052 (2014)
3. Rami, H., Hamri, M., Masmoudi, L.: Objects Tracking in Images Sequence using Local Binary Pattern (LBP). *International Journal of Computer Applications* 63(20), 19–23 (2013)
4. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distribution. *Pattern Recognition* 29(1), 51–59 (1996)
5. Lacabex, B., Cuesta-Infante, A., Montemayor, A., Pantrigo, J.: Lightweight tracking-by-detection system for multiple pedestrian targets. *Integrated Computer-Aided Engineering* 23(3), 299–311 (2016)
6. Lim, J., Kim, W.: Detecting and tracking of multiple pedestrians using motion, color information and the AdaBoost algorithm. *Multimedia Tools and Applications* 65(1), 161–179 (2013)
7. Li, D., Xu, L., Goodman, E., Xu, Y., Wu, Y.: Integrating a statistical background- foreground extraction algorithm and SVM classifier for pedestrian detection and tracking. *Integrated Computer-Aided Engineering* 20(3), 201–216 (2013)
8. Chen, Y., Chen, N., Zhou, Y., Wu, K., Zhang, W.: Pedestrian Detection and Tracking for Counting Applications in Metro Station. *Discrete Dynamics in Nature and Society*, pp. 1–11 (2014)
9. Leykin, A., Hammoud, R.: Pedestrian tracking by fusion of thermal-visible surveillance videos. *Machine Vision and Applications* 21(4), 587–595 (2010)
10. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. *CVPR*, pp. 511–518 (2001)
11. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. In: *Proceedings of International Conference of Computer Vision*, pp. 734–741 (2003)
12. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(5), 564–577 (2003)

*Raymundo Rodríguez Romero, Luis Pastor Sánchez Fernandez, et al.*

13. Bradski, G.: Computer Vision Face Tracking as a Component of a Perceptual User Interface. In: IEEE Workshop Applications of Computer Vision, pp. 214–219 (1998)
14. Baqué, P., Fua, P.: Multi-camera pedestrians video – CVLAB. Lausana, Suiza: Multi-camera pedestrians video. <https://cvlab.epfl.ch/data/data-pom-index-php/> (Recuperado el 10 de agosto del 2019)