Información Tecnológica
Vol. 31(3), 23-38 (2020)
http://dx.doi.org/10.4067/S0718-07642020000300023

Algoritmos de rastreo de movimiento utilizando técnicas de inteligencia artificial y machine learning

Daniel Santos*, Leonardo Dallos, Paulo A. Gaona-García
Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas. Bogotá D.C. - Colombia
(correo-e: dfsantosb@correo.udistrital.edu.co, dldallosp@correo.udistrital.edu.co, pagaonag@udistrital.edu.co).

*Autor a quien debe ser dirigida la correspondencia

Recibido Sep. 10, 2019; Aceptado Nov. 6, 2019; Versión final Dic. 20, 2020, Publicado Jun. 2020

Resumen

El objetivo de este artículo es implementar un análisis de algoritmos de seguimiento basado en técnicas de visión por computador y machine learning para identificar, rastrear y clasificar diferentes elementos y patrones presentes en un video. Existen variaciones asociadas con la precisión en las que este tipo de técnicas se aplican para llevar a cabo el rastreo de objetos en movimiento, lo cual puede influir de manera significativa sobre la calidad en la captura, así como el rendimiento de procesamiento utilizado por dispositivos físicos contenedores. En este estudio se analizaron los algoritmos más usados en este tipo de rastreos: SIFT, SURF y ORB. ORB fue el algoritmo más eficiente en la detección de dichas características. Se pudo concluir que el análisis de los modelos desarrollados presentó buenos resultados bajo un ambiente controlado; sin embargo, en un ambiente no controlado se tiende a presentar errores y el nivel de precisión baja considerablemente.

Palabras clave: visión artificial; inteligencia artificial; visión por computador; procesamiento de imágenes; seguimiento de objetos

Motion tracking algorithms using AI and machine learning techniques

Abstract

The main objective of this article is to implement a tracking algorithm analysis based on computer vision techniques and machine learning to identify, track, and classify different elements and patterns present on a video. There are variations associated with the precision in which these types of techniques are applied to carry out the tracking of moving objects, which can significantly affect capture quality and performance processing used by physical devices. The most used algorithms (SIFT, SURF and ORB) for this type of tracing were analyzed. ORB was the most efficient. It was possible to conclude that the analyses of the models developed showed good results under controlled conditions. However, there were errors and the accuracy dropped considerably under uncontrolled conditions.

Keywords: artificial vision; artificial intelligence; computer vision; image processing; object tracking

INTRODUCCIÓN

Actualmente el rastreo de imágenes es de gran importancia para diversos campos tales como biología, estudios sociales, educación, seguridad, entre otros; así que desarrollar una herramienta que ayude a solucionar esto es de gran importancia; por ejemplo, en biología se desea estudiar un organismo en particular, de ahí es necesario seguirlo para ver su comportamiento. Actualmente se han realizado este tipo de estudios asociados sobre diversos campos como la vigilancia, la navegación de vehículos, los deportes, manejo de tráfico, seguridad, entre otros, (Abd-El-Hafiz et al., 2016; Arriagada y Aracena-Pizarro, 2019; Mendes et al., 2019; Grandon-Pasten et al., 2017). Cuando se está realizando la detección y seguimiento de objetos, la capacidad de los seres humanos no es del todo precisa y efectiva ya que se ve limitada a las capacidades visuales de cada persona. Por lo tanto, utilizar estrategias asociadas a la rama de visión por computador, permite automatizar este tipo de tareas con el propósito de ayudar y facilitar el seguimiento de objetos mediante su representación a partir de patrones.

En base a lo anterior se propone la implementación de un análisis de algoritmos de seguimiento basado en técnicas de visión por computador y machine learning, para identificar, rastrear y clasificar diferentes elementos y patrones presentes en un video, además se implementarán técnicas de filtración como los pasa bajos, el filtro promedio y filtros gaussianos, búsqueda de contornos, el cual se puede hacer con detector de bordes de Canny y Sobel, operaciones morfológicas como lo son la dilatación y la erosión, transformaciones invariantes a escala, rotación y traslación como lo es SIFT y algunos otros descriptores, SURF y ORB. Otros algoritmos para utilizar en el proyecto serán k-means perteneciente a aprendizaje no supervisado que servirá para encontrar grupos de características comunes de las imágenes de entrenamiento, estos grupos servirán como base para aplicar k-nearest-neighbors de aprendizaje supervisado en la etapa de clasificación.

Arriagada y Aracena-Pizarro (2019) presentan un prototipo que permite ayudar a un conductor de un vehículo a poner atención a las señales del tránsito que estén postadas en la vía, pretendiendo asistir al conductor, y por ende evitar infracciones o accidentes. En este trabajo se desarrolla un prototipo que permite captar las señales de tránsito como información existente en los caminos y calles a través de una cámara e indicar al conductor del móvil (señal audible, proyección o un visor) su resultado, mediante el empleo de técnicas de visión computacional, tales como reconocimiento de patrones, matching (homologación), transformación de distancia, detección de colores, bordes, etc., detectar las señalizaciones del tránsito que ayuden a la conducción de un vehículo. Por su parte Mendes y otros (2019) presentan un sistema de detección de posición angular de buques, utilizando técnicas de extracción de características en imágenes digitales y redes neuronales artificiales. Los resultados favorecen aplicaciones futuras en el seguimiento de buques (tracking) utilizando imágenes infrarrojas. Un sistema de Visión Computacional tiene como meta obtener, a partir de una imagen digital, información geométrica, topológica o física sobre el escenario u objetos que componen esta imagen, para realizar algún proceso decisorio. De acuerdo con su aplicación, esta información puede permitir el reconocimiento de patrones, la clasificación de objetos, el movimiento de robots, etc.

Grandon-Pasten y otros (2017) presentan un sistema que se compone de dos módulos principales. El primero realiza el procesamiento de imagen, cuyo objetivo es determinar el mapa de profundidad en un par de vistas; el segundo módulo tiene como objetivo crear el modelo 3D del objeto, para lo cual debe determinar el mapa total de todos los puntos 3D generados. Este trabajo propone una arquitectura de solución automática al problema de reconstrucción de objetos 3D, así como la organización y selección de los métodos para obtener el modelo final, centrándose en el uso exclusivo de técnicas de visión computacional.

Una de las motivaciones de la investigación es facilitar el estudio o análisis de entidades en movimiento como por ejemplo en el caso de biólogos estudiando organismos que no tienen una posición fija, en sociales el estudio de un individuo en un entorno en particular, en computación gráfica posibles lugares donde se encuentra un objeto y todo con el propósito de hacer más fácil las tareas humanas. El resto del artículo se encuentra organizado de la siguiente manera. La sección 2 aborda de manera general las técnicas que se han implementado para rastreo de objetos, así como trabajos relacionados. La sección 3 se plantea la metodología utilizada con el propósito de plantear el estudio. La sección 4 presenta el desarrollo planteado mediante la utilización de los diferentes algoritmos y técnicas de visión por computador. La sección 5 presenta los resultados obtenidos. La sección 6 presenta las discusiones del estudio respecto a otras propuestas realizadas. Finalmente, la sección 7 presenta las conclusiones y trabajo futuro.

OTROS ANTECEDENTES

El rastreo representa un reto importante porque debe garantizarse la no pérdida del foco en el objeto identificado. Se han desarrollado una gran variedad de técnicas para ello que van desde analizar las imágenes separadamente con tratamiento en bruto, hasta deducir y usar complejas fórmulas matemáticas que generalmente van acompañadas de algoritmos con capacidad de aprendizaje. La Fig. 1 refiere a una

clasificación general en la que podrían caber todas; se expondrán las generalidades de cada grupo y se detallarán someramente algunas.

De acuerdo con Ardila (2014), el grupo de métodos por seguimiento de punto se caracteriza porque el punto no es un mero pixel, sino quizás un pequeño grupo de pixeles con una característica descriptiva para su identificación; sin embargo, la pérdida de rastreo puede darse por oclusión, por falsa detección o por salto del punto de rastreo por alta velocidad, por lo que deben añadirse algoritmos predictivos y correctivos que procuren reencontrar nuevamente la referencia por técnicas como definidas en tres, a saber, i) filtro Kalman, ii) filtro de partículas, y iii) seguimiento kernel.

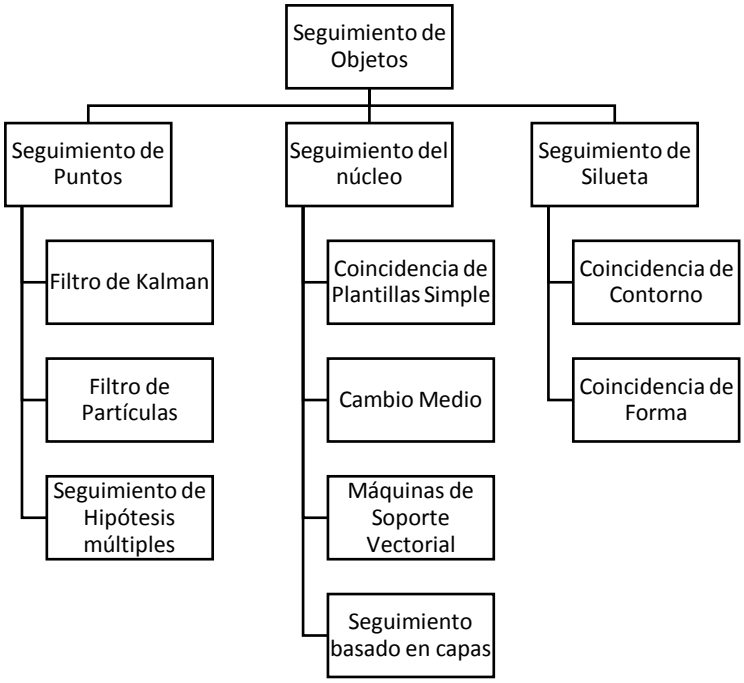

Fig. 1: Métodos para el rastreo de objetos (Athanesious y Suresh, 2014)

El filtro de Kalman ayuda a establecer un balance entre los valores predictivos y el ruido a través de ecuaciones de estado en un ciclo entre la predicción de variables de estado y la actualización de las mismas. Por su lado, el filtro de partícula utiliza unas muestras de estas con unos pesos asociados. Al iniciar el seguimiento se lanzan puntos al azar sobre la imagen, inmediatamente se le asignan valores al azar para crear un nuevo grupo de partículas que reemplazará al anterior. Esta asignación al azar hace más probable que se elijan aquellos que han caído sobre el objeto, de esta forma el bucle se repite constantemente hasta que desaparezca el objeto de escena. Por su parte, el método de seguimiento de múltiples hipótesis plantea estrategias para hacer seguimientos concurrentes a trayectorias predictivas, que pueden ir filtrándose por técnicas como el filtro de Kalman para determinar la opción adecuada. En el segundo grupo, seguimiento de kernel, se lleva a cabo calculando el objeto no estacionario que es mostrado por la región embrionaria entre los consecutivos marcos. Este embrión o kernel está parametrizado y podría permitir el rastreo de múltiples objetos. El primer método de coincidencia de plantilla extrae sectores de la imagen y la contrasta con una plantilla para identificar la posición del objeto. Esta técnica permite el seguimiento de un único objeto.

El método por cambio de medio parte del objeto identificado por segmentación. Inicialmente el fondo se separa del objeto, luego se usa la transformada de la distancia para mejorar la precisión de la representación y localización del objeto. El método máquina de soporte vectorial otorga un conjunto de valores positivos y negativos. Las muestras positivas incluyen una imagen de seguimiento y las negativas las demás cosas fuera de la trayectoria de seguimiento. Por último, en este grupo tenemos el seguimiento basado en capas en el cual la forma, los valores de movimiento como traslación y rotación incluyendo su apariencia están inmersos en cada capa en función de la intensidad. Como último grupo clasificador encontramos el rastreo por siluetas. Aquí, estructuras como el hombro, brazos o manos, que son formas compuestas no bien definidas por figuras geométricas son las aplicables a este método. Se propone detectar la región o trama de interés con base en un modelo obtenido en la trama inmediatamente anterior, pudiéndose presentar oclusiones, división de objetos y fusión de los mismos.

En el seguimiento de contorno se puede evidenciar un proceso iterativo de un contorno primario en un marco anterior a su nueva posición en el marco actual para establecer la cantidad de objeto superpuesto con el

fotograma anterior. Los parámetros de forma y movimiento determinan el estado del objeto. Como técnica final a mostrar, la técnica de forma coincidente examina formas en la actual imagen y compara con plantillas de núcleos de formas. Se usan funciones de densidad y límites de silueta, los bordes son los modelos de los objetos. Dentro del área de identificación de imágenes, se han desarrollado gran variedad de proyectos que estan asociados desde la detección de objetos en imágenes hasta el seguimiento de objetos en diferentes campos como la física, biología, sociales entre otros. Dentro de estas propuestas se han utilizado diversas técnicas para lograr rastrear y seguir objetos; a continuación, se dará una breve descripción de los trabajos realizados.

Dentro de las primeras aproximaciones a trabajos que se han realizado se encuentran propuestas asociadas al monitoreo de las vías férreas (Kate y Katti, 2016). Esta propuesta se realiza la captura de fotos de vías mientras el tren está en movimiento, y con base a estas fotos se buscan imperfecciones con el fin de mantener la seguridad del pasajero. Una de las estrategias utilizada por los autores de este proyecto, era la aplicación de un algoritmo basado en redes neuronales previamente entrenado con una serie de imágenes (patrones de tipos de materiales) para que identifique los materiales presentes en la vía y así poder reconocer posibles anomalías. Para el desarrollo de este proyecto se utilizó Redes Neuronales de Convolución profunda que permitió arrojar como resultado la detección de imperfecciones en un margen del 86,06 % a 92,11 %.

Otro trabajo relacionado lo realizaron Carracedo y otros (2002), mediante la detección del cambio volumétrico y estimación del movimiento del corazón a partir de imágenes de resonancia magnética. Como estrategia, utilizaron un modelo computacional previamente establecido basado en nodos de espacio temporales. Este planteamiento permitió utilizar las imágenes como máscara tipo mallas de identificación de patrones sobre las imágenes en cada etapa y zona de estudio. Por su parte en Ford y Siraj (2014), utilizaron estrategias mediante la transformación Wavelet, herramienta matemática que permitió llevar a cabo la detección de anomalías sobre imágenes con el propósito de detectar la inclinación de pistas (imágenes de código obtenidas con lectores ópticos en la industria aérea y el sector comercial), y corregir lecturas erróneas en la detección de caracteres ópticos (OCR). Como resultado, propusieron un algoritmo nuevo de detección de esviaje basado en la transformada wavelet.

Desde la medicina, se ha encontrado una interesante aplicación de algoritmos inteligentes para el seguimiento de células inmunes (macrófagos) en el cerebro de un ratón a partir de imágenes MR (Resonancia Magnética) (Chen, 2009). El método utilizado por los autores logró detectar manchas negras, con el propósito de asociar y seguir con el rastreo en espacios tridimensionales (voxel). Los autores utilizaron un algoritmo que prioriza la pertinencia de la célula detectada acorde a la intensidad de la misma y el tiempo medio de vida dentro del voxel. Como resultado, el estudio presentó que el uso de Máquinas de vector de soporte es bueno para detectar los macrófagos.

METODOLOGÍA

Para llevar a cabo este estudio, se ha utilizado una metodología de tipo descriptivo y para su validación de tipo experimental. Por un lado, se utilizó un método descriptivo con el propósito de caracterizar las variables de entrada requeridas para el desarrollo del proyecto actual, en donde se tiene un conjunto de imágenes del objeto a seguir, con el propósito de extraer los puntos clave de cada una estas imágenes utilizando algoritmos SIFT, SURF y ORB; después se realizará la fase de aprendizaje no supervisado donde se busca encontrar las características relevantes utilizando k-means para que después con los diferentes frames del video se pueda realizar la clasificación. Esta caracterización se tomó a partir de revisión de literatura científica y del conocimiento de expertos. Por otro lado, se trabajó con un método de investigación de tipo experimental, con el propósito de definir el modelo de clasificación adecuado de acuerdo con las variables de entrada mediante herramientas de simulación y algoritmos seleccionados para hacer el seguimiento del objeto. Esto permitió, entre otras cosas, llevar a cabo un diseño más aterrizado de acuerdo con las iteraciones realizadas. En general, e independientemente del campo de aplicación, el proceso de rastreo consta de las etapas de detección del objeto a seguir, clasificación del objeto y seguimiento del objeto. Cada etapa a su vez consta de distintas características y metodologías posibles.

Para la detección de objetos se pueden usar distintas técnicas: diferencia de frames, que requiere bajo cómputo, pero un fondo constante; sustracción de fondo, requiere baja memoria y no exige submuestreos para crear un adecuado modelo del fondo, pero se necesita un buffer constante y no soporta fondos multimodales; y flujo óptico que puede obtener información completa del movimiento pero requiere demasiado cómputo (Roohbakhsh y Yaghoobi, 2015). En la etapa de clasificación de objetos podemos encontrar las siguientes técnicas: basada en el movimiento (Agarwal, Sivakumaran y Naidu, 2016), donde se espera encontrar patrones periódicos con poco flujo residual como en el movimiento del cuerpo humano; basada en la textura analizando los gradientes de sectores de la imagen contrastados con patrones preestablecidos; basada en la forma y finalmente basada en el color (Roohbakhsh y Yaghoobi, 2015).

Por último, en la etapa de rastreo podemos usar técnicas tales como: seguimiento de puntos, fundamentado en el análisis del umbral, y sujeto a riesgos como la oclusión; seguimiento de kernel, calculando el patrón embrionario entre frames; seguimiento de silueta usado cuando las formas son figuras geométricas bien delimitadas, no útil en el cuerpo humano por sus formas compuestas y no geométricas (Chen, 2009). A partir de estas estrategias de análisis, para llevar a cabo nuestro estudio se complementaron con una serie de fases, las cuales se identifican en la Fig. 2.

Fig. 2: Metodología utilizada para experimento.

Las 5 fases estan asociadas con: 1) Fase de exploración, consiste en la recolección de datos necesarios para el entrenamiento y las pruebas del modelo de red neuronal, 2) Fase de procesamiento de los datos, en esta se lleva a cabo un análisis para la limpieza de los datos con el propósito de eliminar ruido e inconsistencias, 3) fase de definición del modelo inicial, donde se planteó un primer modelo de red neuronal, implementando una serie de pruebas y ajustes para definir el modelo de red neuronal ideal para el sistema, 4) fase de pruebas, consistió en modificar la configuración del modelo inicial y, con base en los resultados obtenidos, definir un nuevo modelo de red neuronal a partir de estrategias de aprendizaje con el propósito de optimizar un modelo ideal para llevar a cabo las predicciones. Por último, 5) fase de comparación de resultados, donde se llevaba a cabo la comparación del desempeño con trabajos relacionados del modelo de red neuronal inicial y final.
En el análisis de imágenes, es de suma importancia lograr establecer características para los distintos elementos y objetos presentes en ellas, esto implica que, para diferenciar las características de un objeto de otro, su descripción debe tener tal especificidad y particularidad que garantice inequívocamente su diferenciación entre ellos. Una vez obtenidas las características y su descripción este método de extracción debe ser susceptible de ser aplicada en otras imágenes, de esta manera, construimos patrones caracterizados usables en la identificación de objetos.

Una zona particular para identificar características se ubican en las esquinas de los objetos, porque allí se presenta gran variación en formas y colores. Un ejemplo simple puede ser el techo de una edificación, si seleccionamos una región que incluya parte del cielo y del techo (sobre la caída), muy probablemente podremos encajar esta sección en otras partes de la imagen donde distintas secciones tengan una estructura similar. No obstante, si se toma una sección de la esquina del techo, veremos que esa parte encajará muy poco con otra sección dada su perspectiva, dado que allí existe un cambio de pérdida de discontinuidad (del cielo a la pared) que hacen difícil ubicarla en otra parte de la imagen. De allí la importancia y relevancia que presentan las esquinas de los objetos.

Basados en este principio de variabilidad en las esquinas, se han desarrollado algunos algoritmos como el Harris Corner Detection (Harris y Stephens, 1988), que usando una ventana rectangular valida el cambio de gradiente en todas las direcciones, por tanto, se identifican las esquinas, funcionando incluso si la imagen se rota. Sin embargo, existe un problema con la escala, ya que, si la esquina es muy grande respecto a la ventana usada, puede no identificarse la esquina ya que esta no la cobija toda y por tanto no valida toda la variación del borde respectivo. Para responder al problema de escalado, se desarrolló el método Scale-invariant Feature Transform (SIFT), el cual se describirá a continuación.

Transformada de escalado invariante

El algoritmo Transformada de escalado invariante (*SIFT* en inglés) busca detectar y describir las características de una imagen, fue patentado por la University of British Columbia, y publicado por David Lowe (Smaoui et al., 2013).

Para subsanar el problema de escalado se usa un filtro de escala-espacio. El proceso en general es aplicar un filtro gaussiano (suavizado) (1) en varios niveles variando la desviación estándar σ

$$h(u,v) = \frac{1}{2\pi\sigma^2} \tag{1}$$

Esto permite reducir el ruido. Luego, mediante diferencias gaussianas entre imágenes se buscan puntos de interés que son detectados como máximos y mínimos locales dentro del histograma de la imagen diferencial, dado este proceso se escala la imagen y se suaviza para realizar el proceso anterior con el fin de que, para cada máximo o mínimo detectado se verifica que también lo sea en el nivel anterior y posterior comparando el pixel con sus 8 vecinos más cercanos, y a su vez comparándolo con sus 9 vecinos de los niveles adyacentes. Por la escala del objeto en la imagen y la ventana de comparación podrían escaparse puntos de interés más grandes, por lo que el proceso se repite en varias escalas de la imagen, construyendo una pirámide de escala descrita en la Fig. 3, con sus respectivas secuencias de niveles de suavizado. En últimas etapas de SIFT se tendrán puntos de interés correspondientes a escalas y niveles distintos, por tanto, como pasos culminantes deben extrapolarse los puntos hallados en escalas superiores (mayor reducción) a la escala original de partida.

Los parámetros de interés para el proceso de detección son entonces la cantidad y valor de los σ de los suavizados en cada escala de la imagen, el factor y número de escalados de la imagen. Como valores empíricos, se ha encontrado como valores óptimos 5 niveles de escala con factor de 0.5, y 4 niveles de suavizado iniciando con σ=1.6 con factor k= $\sqrt{2}$. Una vez hallados los puntos (con asociación a la escala y nivel de suavizado en que se detectó), deben generarse los descriptores para cada uno de ellos. Por cada punto se usa una ventana de 16x16 centrado en el punto para generar su descripción, cuyo análisis se realiza dentro de la imagen en escala y suavizado original de detección.

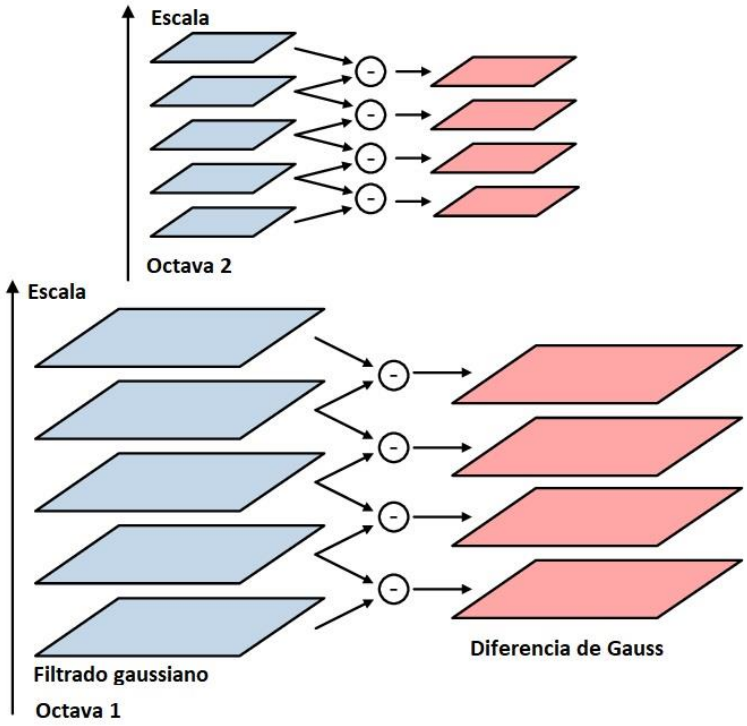

Fig. 3: Pirámide de imágenes para calcular la diferencia Gaussiana (Mentzer et al., 2014)

El descriptor parte de los gradientes horizontales (2) y verticales (3), que permite hallar la orientación (4) y magnitud (5) para cada pixel, con una división de 36 intervalos, se asigna cada uno de los pixeles en regiones acorde a su valor máximo en magnitud, asignando un peso en el intervalo igual a la distancia para clasificar en el siguiente. También se asigna a los intervalos donde el pico en el histograma supere el 80% otorgando pesos complementarios que con el dominante sumen 1. Esta asignación de pesos relativos se usa para evitar que orientaciones similares en imágenes distintas puedan tener una clasificación errónea por estar ubicado en celdas diferentes. En general, siendo n el número de divisiones, los pesos asignados en la orientación dominante y las complementarias tendrán pesos de un factor m/n, con m variando entre 1 y n como la relevancia relativa entre los mismos intervalos a considerar.

$$dx = I(x+1, y) - I(x-1, y) \tag{2}$$

$$dy = I(x, y+1) - I(x, y-1) \tag{3}$$

$$\theta = \arctan\left(\frac{dy}{dx}\right) \tag{4}$$

$$m(x, y) = \sqrt{\{dy^2 + dx^2\}} \tag{5}$$

$$\sum w_{k(x,y)} = 1 \tag{6}$$

$$\sum w_{i(x,y)} = 1 \tag{7}$$

$$\sum w_{j(x,y)} = 1 \tag{8}$$

Las ecuaciones de la 2-8 son utilizadas en el proceso para calcular los descriptores en SIFT. Igual proceso se hace con la ponderación del peso en cada celda acorde con la magnitud del gradiente en las direcciones horizontal y vertical; se tendrá una triple ponderación (interpolación trilineal) que evitan que el ruido y pequeñas variaciones entre una imagen y otra puedan alterar la correcta clasificación de los objetos. Un ejemplo de un kernel gaussiano es el siguiente:

$$\frac{1}{21}\begin{matrix} 1 & 3 & 1 \\ 3 & 5 & 3 \\ 1 & 3 & 1 \end{matrix}$$

El aporte de los pesos de cada pixel en cada intervalo (9) se pondera con una función gaussiana G(x,y), de forma tal que aquellos más cercanos al punto de interés son más relevantes que los situados en las orillas asumiendo que estos últimos son más propensos al ruido.

$$h = (h_1, h_2, ..., h_n) \tag{9}$$

$$h_k = \sum_{(x,y)} w_i(x, y) w_j(x, y) w_k(x, y) m(x, y) G(x, y) \tag{10}$$

Construido el histograma, se halla la orientación dominante de la región y se gira para alinearla con esa orientación, para construir el descriptor (12) se divide la región en 16 celdas (4x4). Se construye para cada celda el histograma (11) de igual forma a como se ha explicado anteriormente, siendo el descriptor la concatenación de los 16 histogramas por tanto dependiendo del número de orientaciones iniciales (supusimos 36 inicialmente) tendremos un total de 16x36=576 dimensiones del descriptor. En la práctica se usa regularmente una división inicial de 8 dando una dimensión final de 1x128, dándonos un vector de 128 valores por cada punto clave que se identifique.

Aunque el histograma no se afecta por cambio de iluminación, si lo es por cambio en el contraste. Para compensar se normaliza cada uno de los histogramas por la norma del conjunto (13-14).

$$H = \left(H_{\{1,1\}}, H_{\{1,2\}}, ..., H_{\{n,n\}}\right) \tag{11}$$

$$v = \left(v_1, v_2, ..., v_m\right) \tag{12}$$

$$\| v \| = \sqrt{\sum_{i=1}^{m} v_i^2} \tag{13}$$

$$v'' = \left(\frac{v_1}{\| v \|}, \frac{v_2}{\| v \|}, ..., \frac{v_3}{\| v \|}\right) \tag{14}$$

Aun minimizado el efecto del brillo y contraste para evitar cambios altos por variación alta de la iluminación se restringe los valores a valores máximos, generalmente a un valor de 0.2. (15)

$$v'' = \left(\max(v_1, 0.2)..., \max(v_m, 0.2)\right) \tag{15}$$

$$\| v \|_2 = \sqrt{\sum_{i=1}^{m} v_i^2} \tag{16}$$

Nuevamente se normaliza (16) el vector para obtener el vector final (17) de características.

$$v''' = \left(\frac{v_1}{\| v \|}, \frac{v_2}{\| v \|}, ..., \frac{v_3}{\| v \|} \right) \tag{17}$$

Características Robustas Aceleradas

De acuerdo con los resultados mostrados en el estudio de comparación de performance de distintos descriptores y la comparación de extractor de descriptores en opencv (Bradski, 2000) se muestra que SIFT produce muy buenos resultados en muchos casos sin embargo consume altos recursos debido a la cantidad de etapas que este produce, los cuales fueron descritos anteriormente. Con el propósito de superar esta deficiencia en estudio realizado por Bay (2008), presentó un algoritmo conocido como SURF. Este algoritmo está basado en el algoritmo SIFT, pero no se aproxima al laplaciano-gaussiano con diferencias gaussianas sino usando Box-filter usando la imagen integral. En una imagen integral, el valor de un pixel es la suma de todos los pixeles que están a la izquierda y sobre el punto incluyendo a este mismo.

Los puntos de interés se hallan usando la matriz Hessiana. El determinante permite medir el cambio alrededor de un punto, ya que este es máximo donde hay mayor variabilidad y también es utilizada para determinar la escala. A diferencia de SIFT no se redimensionan imágenes, sino se va aumentando el tamaño del box-filter siendo el tamaño de 9x9 como el valor inicial (escala s=1.2) y progresando a valores mayores como 15x15, 21x21, etc. Los filtros gaussianos son óptimos, pero en la práctica deben discretizarse ocasionando pérdida de información y obstaculizando la repetitividad. La orientación se consigue usando wavelet de Haar en las direcciones x e y en una región circular de radio 6s. Un gaussiano centrado en el punto de interés, ya calculado para todos los vecinos, se encuentra en la orientación dominante dentro de la suma de todos los resultados, en una ventana de 60º. El descriptor se construye a partir de una región cuadrada de tamaño 20s subdividido en 16 subregiones a quienes se les calcula el wavelet de Haar y se suavizan con filtros gaussianos. El vector descriptor es la conjunción de los descriptores de cada una de las regiones.

Algoritmo ORB

Entre las implementaciones realizadas se tienen los algoritmos de SIFT y SURF, y aunque la segunda mejora notablemente la velocidad de respuesta, una gran limitante de uso se presenta por las licencias de uso. Debido a lo anterior, en los laboratorios de OpenCv Ethan Rublee, Vincent Rabaud, Kurt Konolige y Gary R. Bradski propusieron el algoritmo ORB como una alternativa a SIFT y SURF, permitiendo no solo escapar al problema de las patentes, sino también mejorando la eficiencia y consumo de recursos con la ventaja de permitirse su uso en forma gratuita. Su diseño se basa en la mezcla de dos algoritmos para dos etapas diferenciadas. La primera de ellas enfocada en la detección (FAST), y la segunda en la descripción (BRIEF).

A) Características de la prueba de segmento acelerado – en inglés *Features from accelerated segment test (FAST):* Para la primera etapa se usa FAST como algoritmo de detección de puntos. Su funcionamiento es simple, primero se determina la intensidad del pixel a analizar, luego en una circunferencia de 16 pixeles se compara la intensidad de este pixel con los respectivos de la periferia, si el pixel de interés siempre es mayor o siempre es menor, se considera que tenemos un punto de interés. FAST no calcula orientación, por ello para ORB ha sido modificado y la orientación se determina por el vector derivado de los pesos de las intensidades respecto al centroide, y para ratificar la invarianza se calcula en los momentos respecto a (x, y) en la región circular.

B) Características Elementales Independientes Robustas Binarias – en inglés, *Binary Robust Independent Elementary Features (BRIEF):* En la segunda etapa, se generan los descriptores usando el algoritmo BRIEF (Binary Robust Independent Elementary Features) propuesto por Michael Calonder, Vincent Lepetit, Christoph Strecha, y Pascal Fua y se caracteriza por requerir muy poco cómputo y memoria de almacenamiento. BRIEF genera un descriptor básico como una cadena de bits donde en forma similar a lo realizado por el algoritmo FAST, se compara la intensidad del pixel en una versión suavizada de la imagen. Como esta cadena no es invariante a la rotación, en ORB se va rotando la región centrada en el pixel cada 12 grados, evaluando en cada ocasión el descriptor BRIEF. Los momentos hasta ahora descritos son de análisis local, es decir no

evalúan la relación entre pixeles distantes. Probando alternativas, se usó la transformada de Hough que considera las relaciones globales, estableciendo bordes y permitiendo encontrar figuras.

Transformada de Hough

Para la detección de regiones también se usó la transformada de Hough, una técnica eficiente para la detección de contornos relacionados geométricamente en una forma específica como una recta o una circunferencia, por ejemplo. Para detectar una recta se parte de la ecuación de la misma en forma polar y se discretiza los rangos de los valores en el espacio polar. Definiendo los rangos para (ρ ,θ) como (ρ_min,ρ_max) y (θ_min,θ_max), siendo este rango sobre un espacio de celdas denominado celdas de acumulación, luego, se evalúa la ecuación sobre cada punto, y si cumple la ecuación se añade un voto a la celda. Un número alto en votos implica que pertenece a la recta.

Clasificador bayesiano

Una vez obtenido los puntos de interés con sus vectores descriptivos ya sea usando el método SURF, SIFT u ORB, es preciso clasificar los mismos. De esto se encarga un clasificador bayesiano.Este clasificador se caracteriza por optar por un enfoque probabilístico de inferencia permitiendo ponderar de una hipótesis en forma cuantitativa. El aprendizaje bayesiano consiste en encontrar la hipótesis más probable, usando para ello un conjunto de entrenamiento y determinando en forma preliminar la probabilidad de una hipótesis. Estos paradigmas de clasificación supervisada basada en Bayes reciben varios nombres, como idiotaBayes, naive Bayes, simple Bayes y Bayes independiente. Naive Bayes aparece en los años 80, y se presentará una pequeña descripción del mismo, partiendo del teorema de Bayes.

Sean A y B dos sucesos aleatorios cuyas probabilidades se denotan p(A) y p(B) respectivamente y que p(B) > 0. Asumiendo que se conoce a priori las probabilidades de A y B, así como la probabilidad condicional de p(B) habiendo sucedido A. Entonces la probabilidad de P(A/B) puede calcularse acorde a la expresión. Como puede verse, este teorema requiere un conocimiento a priori, y de no tenerse, deben estimarse derivando en un costo computacional alto.

Clasificador por K-Nearest neighbors

Otro enfoque para clasificar es utilizar este método que consiste en clasificar un elemento basado en sus k vecinos más cercanos, es un método de aprendizaje supervisado. En la figura 4 se clasifica Xj con base a sus 5 vecinos más cercanos lo que da como resultado que el elemento Xj pertenece a la clase Xj.

DESARROLLO PROPUESTO

De acuerdo con las fases definidas en la metodología planteada, para validar la propuesta se desarrollaron una serie de pruebas utilizando herramientas de desarrollo sobre el marco de trabajo OpenCV ya que este framework provee un gran conjunto de ventajas como: i) Esta libre bajo licencia BSD de ahí que sea libre para uso académico y comercial; y ii) Provee un gran conjunto de rutinas que hacen posible realizar operaciones complejas sobre imágenes en segundos. Como estrategia para obtener un buen resultado en seguimiento y rastreo, se optó por implementar los algoritmos SIFT, SURF y ORB como extractores de características, k-means para clusterización y finalmente, k-nn. Finalmente se realizaron una serie de pruebas con un objeto (zanahoria) en diferentes ambientes donde cambia la luminosidad con el propósito de generar oclusión del objeto y así constatar la eficiencia del modelo creado. En la Fig. 5 se muestran algunos métodos de detección y clasificación de objetos posteriormente descritos.

En la Fig. 5 se logra identificar la secuencia general que se puede seguir con el fin de realizar seguimiento de objetos. Se muestran las principales técnicas y métodos usados, que serán ampliados y complementados con otros en las secciones siguientes. Una de las técnicas más usadas para la detección de objetos en movimiento es la de sustracción de fondo, también llamada extracción de primer plano. La usan los sistemas de seguridad de cámaras estáticas ya que permite establecer cuando ha surgido un evento y realizar estudios de movilidad sobre los mismos. El principio de funcionamiento es distinguir entre fondos (zonas estáticas) e imágenes en movimiento (primer plano), usando temporalmente representaciones del fondo mediante un modelo dinámico y adaptativo a las variaciones, como pueden ser la intensidad de luz o ingreso de nuevos actores a la escena.

Algunas apuestas al modelo son la función de probabilidad Gaussiana, que se enfocan en establecer el fondo más probable para cada entrada. Este método posee la desventaja que se satura y colapsa en escenas complejas. También encontramos métodos de sustracción de fondo basados en análisis de texturas. Son robustos cuando fondo y referencia coinciden en color, pero exigen un gran gasto computacional. Por su parte, los métodos basados en modelos estocásticos tienen altas tasas de detección, bajo consumo computacional y robustez ante el ruido, aunque pueden fallar cuando existe similitud entre la imagen y el fondo (Wang et al., 2013).

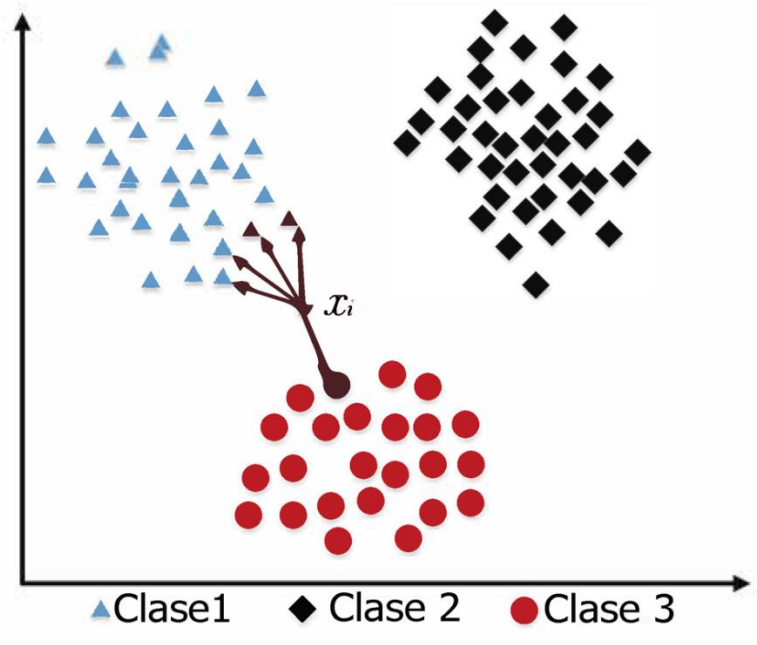

Fig. 4: Ejemplo KNN con tres clases (Chong et al., 2013).

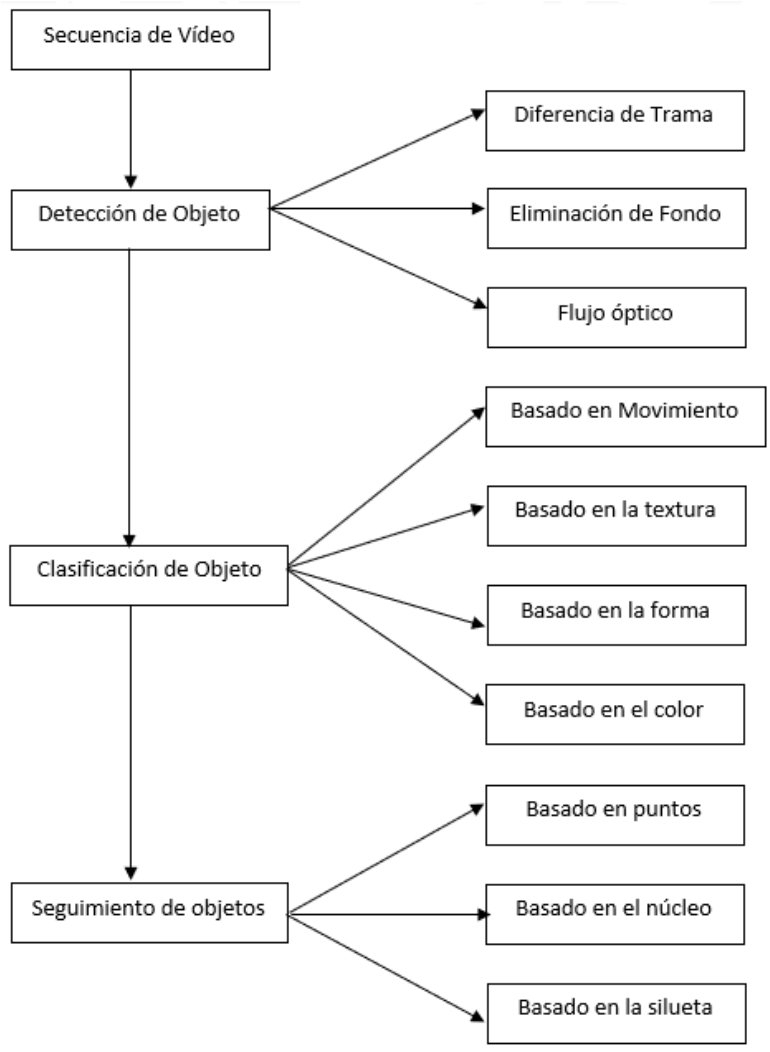

Fig. 5: Pasos básicos para el rastreo de objetos (Roohbakhsh e Yaghoobi, 2015).

La diferencia de cuadros (frame difference), es una solución bastante sencilla para la detección de objetos, que consiste básicamente en ir comparando constantemente el cuadro actual con el de referencia; seguido de esto se comparan los bits de diferencia con el fin de determinar si corresponden al fondo o al objeto a identificar. Usa bajo costo computacional pero no puede ser eficiente si la imagen y el fondo son similares, y también puede generar una selección múltiple de zonas con colores similares como parte de un mismo objeto (Ardila, 2014).

Otra forma para la detección de objetos, es la técnica de flujo óptico, cuyo principio es seguir el brillo de los pixeles para determinar los vectores de movimiento. Para ello, se considera como constante el brillo de un pixel, que por seguridad no se selecciona individualmente, sino dentro de una zona del objeto, así se evita pérdida de rastreo por oclusión. El flujo óptico puede fallar cuando los desplazamientos son grandes, ya que los pixeles registran saltos y se pierde rastreo de los mismos, algunos algoritmos pueden identificar estos saltos, pero en distancias de unos pocos pixeles.

Por último, tenemos la segmentación, que como su nombre indica, es dividir una imagen digital en distintas secciones simplificando el análisis al dividir en zonas de interés, que pueden ser tratadas separadamente. Múltiples algoritmos se pueden encontrar para la segmentación, por ejemplo, se tienen: crecimiento de regiones, conjuntos de nivel, redes neuronales, basada en modelos, por umbralización, particionamiento gráfico, entre otros.

En este último grupo tenemos la mayoría de técnicas que se emplean para situaciones complejas en los que se requiere rapidez y eficiencia, con mayor gasto computacional para unos u otros, y mayor rapidez y eficiencia para otros. Una vez detectadas estructuras, puntos, contornos o rasgos distintivos del objeto (identificación del objeto), se crean uno o varios descriptores que representarán el objeto, seguido de esto se comparan con los descriptores del conjunto de entrenamiento y se procede a su clasificación en el cual se le asigna la etiqueta en el caso de múltiples categorías o se dice si es o no es dado que sea un clasificador binario.

La clasificación por movimiento es una técnica que fija su atención en las estructuras rígidas del objeto, buscando patrones de periodicidad y rigidez, que puede verse afectado por flujo residual de lectura. Tan importante es la rigidez de la estructura que los movimientos humanos son difícilmente rastreables por esta técnica, dada la variación alta en el flujo residual. Otra forma de clasificar es la técnica basada en texturas, fundamentada en la variación del gradiente en las distintas zonas para luego superponerse a una red densa uniforme de contrastes para mejorar la precisión. Tal vez la más intuitiva conceptualmente es la clasificación por formas, que clasifica los objetos en movimiento acorde a la forma de los mismos. Parámetros como el zoom de la cámara, la relación de aspecto y el área proporcional de la imagen sirven a propósitos de puntualizar las características clasificatorias. El color también puede servir como característica clasificadora, y ofrece cierta estabilidad de variación entre los pixeles, además que es de fácil adquisición, por tanto, requieren bajo coste computacional, sin embargo, sus aplicaciones son muy concretas en ambientes que garanticen alto contraste ya sea que el análisis se haga en la escala de grises o en color.

RESULTADOS

Basados en estudio de la literatura y descripción de los algoritmos utilizados se procedió a crear el modelo. Para esto era necesario tener un conjunto de imágenes de entrenamiento para poder crear el modelo. Se determino utilizar un objeto tradicional (una zanahoria) dada su uniformidad y familiaridad para las pruebas, de la cual se tomaron 30 imágenes desde diferentes posiciones, ángulos e iluminaciones de tal forma que nuestro modelo fuera robusto a cambios de iluminación y rotación. Seguido de obtener las diferentes imágenes se procedió a calcular los puntos clave utilizando los descriptores anteriormente mencionados: SIFT, SURF y ORB. Para cada uno de los puntos clave se calculo su respectivo descriptor, SIFT otorga 128 descriptores por cada punto clave, SURF otorga 64 y finalmente ORB define 32 descriptores que con el total de puntos clave obtenidos suman un total de 21000 descriptores. Debido a que la cantidad de descriptores fue amplia, se procedió a aplicar K-Means con k = 1000 para hallar 1000 clúster que definirán los descriptores relevantes en las 30 imágenes de entrenamiento.

En la siguiente etapa se realizó una aplicación utilizando el marco de trabajo OpenCV funciones propias creadas utilizando el lenguaje C y C++ que permitieran acceder a la cámara y captura frames en tiempo real. Una vez desarrollada la aplicación se procedió a activar la cámara. El video que se obtenía de la cámara se capturo por cada frame del cual se calculaba sus respectivos puntos clave y sus descriptores utilizando SIFT, SURF y ORB. Con base a estos descriptores se hacía match con los 1000 centroides obtenidos por K-Means utilizando un clasificador de k vecinos más cercanos para la cual se medía la distancia desde los centroides hasta cada descriptor obtenido.

Para el entrenamiento se tomó como objeto una zanahoria para llevar a cabo el proceso de rastreo. Siguiendo el proceso, se llevó a cabo una prueba del modelo con las diferentes transformadas. Para llevar a cabo el experimento se adecuaron dos ambientes, a saber: 1): el elemento entrenado se encuentra solo; 2: el elemento entrenado se encuentra con otros elementos con formas y colores diferentes. Los resultados del experimento se presentan en las Fig. 6-11.

Resultados utilizando ORB

La Fig. 6 presenta resultados del primer ambiente utilizando método ORB. La Fig. 7 presenta resultados del segundo ambiente utilizando método ORB.

Fig. 6: Ambiente 1 – ORB.

Fig. 7: Ambiente 2 – ORB.

Resultados utilizando SURF

La Fig. 8 presenta resultados del primer ambiente utilizando método SURF. La Fig. 9 presenta resultados del segundo ambiente utilizando método SURF.

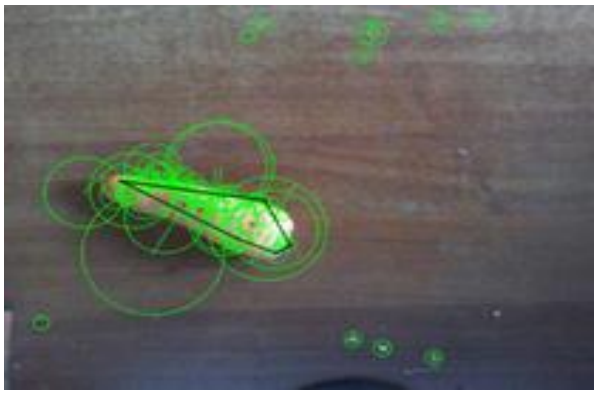

Fig. 8: Ambiente 1 – SURF.

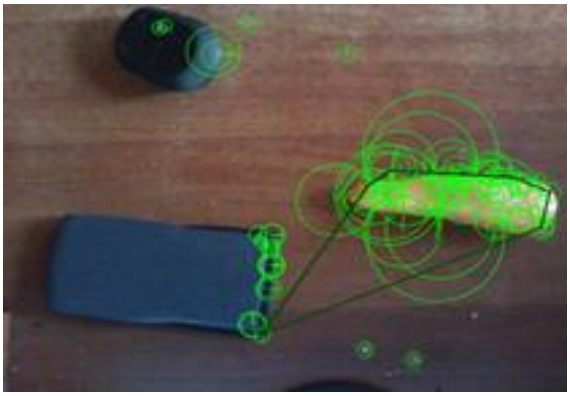

Fig. 9: Ambiente 2 – SURF.

Resultados utilizando SIFT

La Fig. 10 presenta resultados del primer ambiente utilizando método SIFT. La Fig. 11 presenta resultados del segundo ambiente utilizando método SIFT.

Fig. 10: Ambiente 1 – SIFT.

Fig. 11: Ambiente 2 – SIFT.

Los anteriores resultados fueron obtenidos con 3600 imágenes que fue el total de frames capturados en un periodo de tiempo definido por un (1) minuto.

Tabla. 1: Precisión de los Algoritmos en el trabajo.

Algoritmo	Accuracy
SIFT	0,62
SURF	0,74
ORB	0,80

Los resultados anteriores se calcularon a partir de la siguiente fórmula (18)

$$\text{Acurracy}=\frac{\text{Correct Classified}}{\text{Total number of images}} \quad (18)$$

A partir de la muestra que se definió para llevar a cabo el análisis, el método aplicado con ORB fue el algoritmo que mejor precisión presentó, seguido SIFT. Estos resultados de manera preliminar permitieron corroborar estudios similares (Rublee et al., 2011; Tafti et al., 2018; Majumdar y Mahato, 2018), donde los autores resaltaron que ORB fue uno de los algoritmos más eficientes que puede detectar una gran cantidad de características, así como el tiempo de coincidencias de imágenes para una cantidad grande de características prolongadas en el tiempo.

CONCLUSIONES

El gran número de técnicas y soluciones existentes hacen abrumador el decantarse por una alternativa en concreto, tanto para la detección de objetos como para el posterior seguimiento. Es imprescindible delimitar el alcance y los objetivos para facilitar enormemente este proceso, debido a que un gran número de objetos puede influir en el desempeño del algoritmo a utilizar. Mediante los resultados que se lograron identificar se pudo concluir que el análisis de los modelos desarrollados presentó buenos resultados en el primer ambiente de trabajo, es decir bajo un ambiente controlado; sin embargo, en un ambiente no controlado se tiende a presentar errores y el nivel de precisión baja considerablemente. Lo anterior se podría mejorar, teniendo en cuenta un conjunto mayor de entrenamiento, con el propósito de generar una mayor cantidad de iteraciones que permitan detectar aspectos dentro del análisis de una manera concreta.

REFERENCIAS

Abd-El-Hafiz, S. K. AbdElHaleem, S. H., y Radwan, A. G., *Permutation techniques based on discrete chaos and their utilization in image encryption.* https://doi.org/10.1109/ECTICon.2016.7561265 in 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 1-6 (2016).

Agarwal, V. K., Sivakumaran, N., y Naidu, V. P. S., *Six object tracking algorithms: A comparative study*. Indian Journal of Science and Technology, 9(30), pp. 1-9. (2016)

Athanesious J.J., y Suresh, P., *Systematic Survey on Object Tracking Methods in Video*, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) October 2012, pp. 242-247 (2014).

Ardila, O. F., Implementación del Voto Electrónico en Colombia: Un estudio desde la responsabilidad estatal, frente a fallas tecnológicas. Universidad Militar Nueva Granada. (2014).

Arriagada, C., y Aracena-Pizarro D., *Detección y reconocimiento de señales de tránsito utilizando matching de chamfer*. http://dx.doi.org/10.4067/S0718-33052007000200008. Ingeniare. Rev. chil. ing., Arica, v. 15, n. 2, p. 174-184. (2019).

Bay, H., Ess, A., Tuytelaars, T., y Van Gool, L., *Speeded-up robust features (SURF)*. https://doi.org/10.1016/j.cviu.2007.09.014. Computer vision and image understanding, 110(3), 346-359 (2008).

Bradski, G., y Kaehler, A., *The opencv library (2000)*. Dr. Dobb's J. Softw. Tools 2000. Available online: https://opencv.org/ (accessed on 5 June 2019).

Carracedo, J., Gómez, A., Moreno, J., Pérez, E., y Carracedo, J. D., *Votación electrónica basada en criptografía avanzada*. in II Congreso Iberoamericano de Telemática. CITA' 2002, Mérida, Venezuela, pp. 1–13 (2002).

Chen, D., *A Feasible Chaotic Encryption Scheme for Image*. In 2009 International Workshop on Chaos-Fractals Theories and Applications, pp. 172-176 (2009).

Ford, V. y Siraj A., *Applications of Machine Learning in Cyber Security*. in 27th International Conference on Computer Applications in Industry and Engineering, pp. 1-6 (2014).

Fu, C., Tang, J., Zhou, W., Liu, W., y Wang, D., *A symmetric color image encryption scheme based on chaotic maps*. in 2013 15th IEEE International Conference on Communication Technology. pp. 712-716 (2013).

Grandon-Pasten, N., Aracena-Pizarro D., y Tozzi, C., *Reconstrucción de objeto 3d a partir de imágenes calibradas*. http://dx.doi.org/10.4067/S0718-33052007000200006. Ingeniare. Rev. chil. ing., Arica, v. 15, n. 2, p. 158-168 (2017).

Harris, C., y Stephens y M.J., *A combined corner and edge detector*. In Alvey Vision Conference, pages 147–152, (1988).

Kate, N., y Katti J. V., *Security of remote voting system based on Visual Cryptography and SHA*. in 2016 International Conference on Computing Communication Control and automation (ICCUBEA) pp. 1-6. (2016).

Majumdar, J., y Mahato, A., *Comparison of SIFT & SURF Corner Detector as Features and other Machine Learning Techniques for Identification of Commonly used Leaves*. International Research Journal of Engineering and Technology (IRJET) Vol 5, No 3, pp 387-392 (2018).

Mendes, V. B., Leta, F. R., Conci, A., y Gonçalves, L. B., *Detección de Posición Angular de Embarcaciones, utilizando Técnicas de Visión Computacional y Redes Neurales Artificiales*. http://dx.doi.org/10.4067/S0718-07642010000600018 Inf. tecnol., La Serena, v. 21, n. 6, pp. 177-188 (2019).

Mentzer, N., Payá-Vayá, G., Blume, H., von Egloffstein, N., y Ritter, W., *Instruction-set extension for an ASIP-based SIFT feature extraction*. In 2014 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV) pp. 335-342 IEEE. (2014).

Roohbakhsh, D., y Yaghoobi, M., *Color Image Encryption using Hyper Chaos Chen.* International Journal of Computer Applications. https://10.5120/19303-0752. vol. 110, no. 4, pp. 9-12 (2015).

Rublee, E., Rabaud, V., Konolige, K., y Orb, G. B., *ORB: An efficient alternative to SIFT or SURF*. In ICCV, Vol. 11, No. 1, p. 2. (2011).

Smaoui, N. Karouma, A., y Zribi, M. *Adaptive Synchronization of Hyperchaotic Chen Systems with Application to Secure Communication*. International Journal of Innovate Computing, Information and Control, vol. 9(3), 1127-1144 (2013).

Tafti, A. P., Baghaie, A., y otros 5 autores, *A Comparative study on the application of SIFT, SURF, BRIEF and ORB for 3D surface reconstruction of electron microscopy images*. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 6(1), 17-30 (2018).

Wang, H., Chen, B., y otros 3 autores, *Robust adaptive fuzzy tracking control for pure-feedback stochastic nonlinear systems with input constraints*. IEEE Transactions on Cybernetics, 43(6), 2093-2104 (2013).