

---

# *Algoritmos de rastreo de movimiento utilizando técnicas de inteligencia artificial y machine learning.*

Cinthia Tobar Castro

October 20, 2023

## **Abstract**

Este artículo presenta un análisis de algoritmos de seguimiento basados en técnicas de visión por computador y machine learning. El objetivo es identificar, rastrear y clasificar diferentes elementos y patrones presentes en un video.

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 características.

Los resultados mostraron que los modelos desarrollados presentaron buenos resultados bajo un ambiente controlado. Sin embargo, en un ambiente no controlado se tienden 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

## **1 INTRODUCCIÓN**

El rastreo de imágenes es esencial en diversos campos como biología, estudios sociales, educación y seguridad. La necesidad de seguir objetos o patrones de interés ha llevado al desarrollo de herramientas basadas en visión por computadora y machine learning para automatizar esta tarea. Estas herramientas emplean técnicas como filtración, detección de contornos, operaciones morfológicas y transformaciones para identificar, rastrear y clasificar elementos en videos.

Ejemplos de aplicaciones incluyen un prototipo para ayudar a los conductores a identificar señales de tránsito y evitar infracciones o accidentes, utilizando técnicas de visión computacional. También se menciona un sistema que utiliza imágenes infrarrojas y redes neuronales para detectar la posición angular de buques, con aplicaciones en el seguimiento de buques.

En otro enfoque, se presenta un sistema que consta de dos módulos, uno para determinar un mapa de profundidad en imágenes y otro para crear modelos 3D de objetos. Este sistema se centra en soluciones automáticas para la reconstrucción de objetos 3D.

El objetivo principal de esta investigación es facilitar el estudio y análisis de entidades en movimiento, como organismos en biología, individuos en entornos sociales o la ubicación de objetos en computación gráfica, con el fin de simplificar tareas humanas.

El artículo se organiza en secciones que incluyen una introducción a las técnicas de rastreo de objetos, una metodología, el desarrollo de algoritmos y técnicas, resultados, discusiones, conclusiones y trabajo futuro.

## **2 OTROS ANTECEDENTES**

El rastreo de objetos representa un desafío importante, ya que implica garantizar que el objeto identificado no se pierda de vista. Se han desarrollado diversas técnicas para abordar este desafío, que van desde el análisis individual de imágenes con un procesamiento básico hasta la aplicación de complejas fórmulas matemáticas, a menudo acompañadas de algoritmos de aprendizaje automático. La

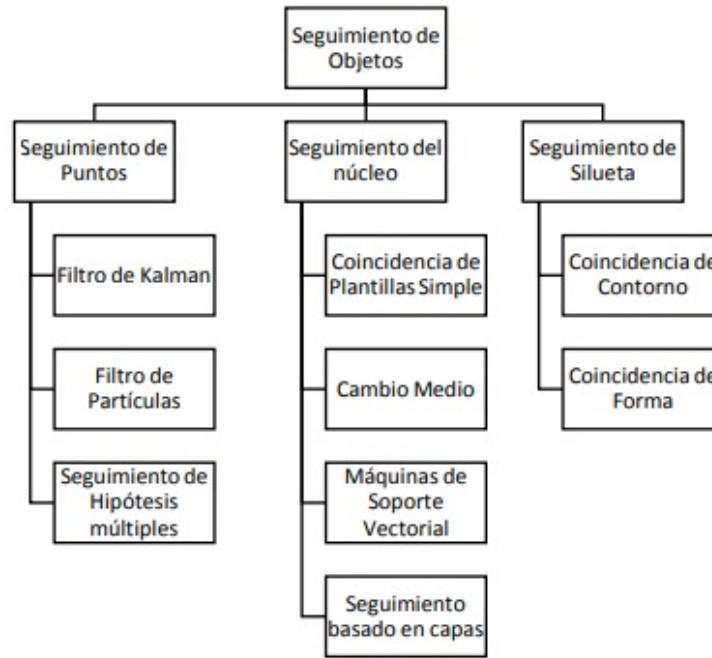


Figure 1: Métodos para el rastreo de objetos (Athanesious y Suresh, 2014)

Figura 1 presenta una clasificación general que abarca estas técnicas, y en este resumen se explorarán brevemente algunas de ellas.

Según Ardila (2014), el grupo de métodos basados en el seguimiento de puntos se distingue por considerar que el punto de interés no es simplemente un píxel, sino posiblemente un pequeño conjunto de píxeles con características descriptivas que permiten su identificación. Sin embargo, la pérdida de seguimiento puede ocurrir debido a obstáculos, detecciones incorrectas o movimientos rápidos del punto de interés. Para abordar estos desafíos, se utilizan algoritmos predictivos y correctivos que buscan reencontrar la referencia de seguimiento. Algunas de las técnicas mencionadas incluyen el filtro Kalman, el filtro de partículas y el seguimiento basado en kernel.

- El filtro de Kalman equilibra predicciones y ruido en el seguimiento de objetos utilizando ecuaciones de estado.
- El filtro de partícula utiliza muestras con pesos y genera partículas aleatorias en la imagen para seguir objetos.
- El seguimiento de múltiples hipótesis permite seguir varias trayectorias simultáneas y refinarlas.
- El seguimiento de kernel identifica objetos no estacionarios entre marcos consecutivos.
- El seguimiento basado en plantillas compara sectores de la imagen con una plantilla para rastrear un solo objeto.
- El seguimiento por cambio de medio separa el fondo del objeto y mejora la representación y localización.
- El método de máquina de soporte vectorial determina regiones de interés en función de una imagen de seguimiento.
- El seguimiento basado en capas combina forma, movimiento y apariencia para rastrear objetos.
- El rastreo por siluetas detecta regiones de interés basadas en modelos anteriores, lidiando con oclusiones y fusiones.
- Las técnicas de forma coincidente examinan formas y comparten siluetas en imágenes.



Figure 2: Metodología utilizada para experimento.

- Se mencionan proyectos relacionados con la detección y seguimiento en áreas como ferrocarriles y medicina.
- En ferrocarriles, se utiliza un algoritmo de redes neuronales para detectar materiales y anomalías.
- En medicina, se sigue el movimiento de células inmunes en el cerebro con imágenes de resonancia magnética.
- Se mencionan proyectos de detección de cambios volumétricos y estimación del movimiento cardíaco.
- También se abordan la detección de inclinaciones en pistas y corrección de lecturas erróneas en la industria aérea y comercial.

Estos párrafos describen una variedad de enfoques y aplicaciones en el seguimiento de objetos en diferentes campos.

### 3 METODOLOGÍA

El estudio se ha llevado a cabo mediante una metodología descriptiva y experimental. En el enfoque descriptivo, se caracterizaron las variables de entrada necesarias para el proyecto, que implica el seguimiento de objetos en imágenes. Se extrajeron los puntos clave de las imágenes utilizando algoritmos como SIFT, SURF y ORB. Luego, se realizó una fase de aprendizaje no supervisado utilizando k-means para identificar características relevantes y permitir la clasificación de objetos en varios fotogramas de un video. Esta caracterización se basó en revisión de literatura científica y el conocimiento de expertos en el campo.

Por otro lado, se empleó un enfoque experimental para definir el modelo de clasificación adecuado de acuerdo con las variables de entrada, utilizando herramientas de simulación y algoritmos seleccionados para el seguimiento del objeto. Este enfoque permitió un diseño más preciso basado en iteraciones y pruebas.

Independientemente del campo de aplicación, el proceso de rastreo de objetos consta de las etapas de detección, clasificación y seguimiento del objeto. Cada etapa implica diversas características y metodologías posibles.

En la etapa de detección de objetos, se pueden utilizar diferentes técnicas, como la diferencia de frames, la sustracción de fondo y el flujo óptico. Cada técnica tiene sus propias ventajas y desventajas en términos de recursos computacionales y requisitos de fondo.

En cuanto a la clasificación de objetos, se pueden aplicar enfoques basados en el movimiento, la textura, la forma y el color. Cada uno de estos enfoques se adapta a las características específicas del objeto que se está siguiendo, lo que ofrece flexibilidad en la elección de la técnica más adecuada para una aplicación particular.

El proceso de análisis de imágenes se divide en cinco fases distintas:

1. **Fase de exploración:** En esta etapa, se recolectan los datos necesarios para entrenar y probar el modelo de la red neuronal.
2. **Fase de procesamiento de datos:** Aquí, se lleva a cabo un análisis de los datos para limpiarlos, eliminando ruido e inconsistencias.
3. **Fase de definición del modelo inicial:** En esta fase, se propone un primer modelo de red neuronal y se realizan pruebas y ajustes para definir el modelo ideal para el sistema.
4. **Fase de pruebas:** Implica modificar la configuración del modelo inicial basándose en los resultados obtenidos y definir un nuevo modelo de red neuronal a través de estrategias de aprendizaje con el objetivo de optimizar las predicciones.
5. **Fase de comparación de resultados:** En esta etapa, se compara el desempeño del modelo de red neuronal inicial y el modelo final con trabajos relacionados.

En un estudio comparativo de descriptores y extractores de características en OpenCV, se destaca que el descriptor SIFT ofrece resultados sólidos en muchos casos, pero presenta una desventaja significativa debido a su alta complejidad y consumo de recursos. Para abordar este problema, se introdujo un algoritmo llamado SURF en un estudio realizado por Bay en 2008. SURF se basa en SIFT pero utiliza un enfoque alternativo, empleando un filtro de caja y la imagen integral en lugar de aproximaciones laplaciano-gaussianas.

En SURF, se localizan los puntos de interés utilizando la matriz Hessiana y se mide la variación en el entorno del punto utilizando el determinante. A diferencia de SIFT, no se redimensionan las imágenes, en su lugar, se ajusta el tamaño del filtro de caja, comenzando con un tamaño inicial de 9x9 y progresando a tamaños mayores. El uso de filtros gaussianos discretizados en SIFT puede llevar a la pérdida de información y dificultar la repetibilidad. Para determinar la orientación, SURF utiliza wavelets de Haar en direcciones x e y dentro de una región circular con radio variable y encuentra la orientación predominante en una ventana de 60 grados. El descriptor en SURF se construye subdividiendo una región cuadrada en 16 subregiones, calculando el wavelet de Haar en cada una y suavizándolo con filtros gaussianos. El descriptor final es una concatenación de los descriptores de cada subregión.

### 3.1 Transformada de escalado invariante

El algoritmo Scale-invariant Feature Transform (SIFT) es una técnica diseñada para detectar y describir las características de una imagen. Fue patentado por la University of British Columbia y publicado por David Lowe. Para abordar el problema de escalado, se emplea un filtro de escala espacio. El proceso implica aplicar un filtro gaussiano para suavizar la imagen en varios niveles, variando la desviación estándar o en cada nivel. Esto permite que el algoritmo SIFT sea invariante al escalado y pueda identificar y describir características en imágenes a diferentes escalas.

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

En el proceso de detección de puntos de interés en SIFT, se comienza reduciendo el ruido en las imágenes. Luego, se utilizan diferencias gaussianas entre imágenes para identificar máximos y mínimos locales en el histograma de las imágenes diferenciales. Este proceso se repite en múltiples escalas de la imagen, creando una pirámide de escalas.

Los parámetros importantes incluyen la cantidad y el valor del suavizado en cada escala, el factor y el número de escalas utilizadas. Valores empíricos recomendados son 5 niveles de escala con un factor de 0.5 y 4 niveles de suavizado, comenzando con  $a=1.6$  y un factor  $K=2$ .

Una vez detectados los puntos de interés, se generan descriptores para cada uno de ellos utilizando una ventana de 16x16 centrada en el punto. El análisis se realiza en la imagen en su escala y suavizado originales. Además, se asocian los puntos de interés a la escala y nivel de suavizado en los que se detectaron. En etapas finales del proceso, se extrapolan los puntos encontrados en escalas superiores a la escala original de partida.

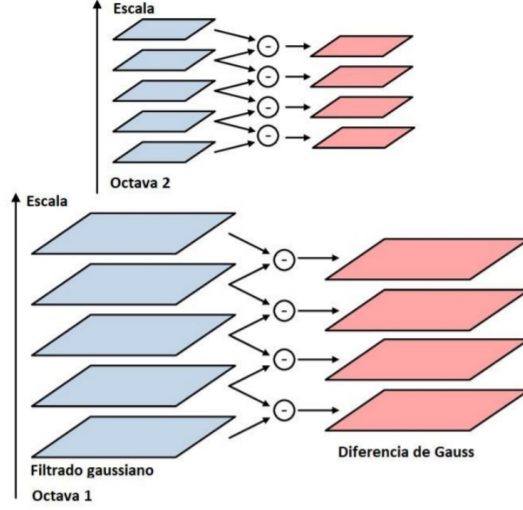


Figure 3: Fig. 3: Pirámide de imágenes para calcular la diferencia Gaussiana (Mentzer et al., 2014)

$$dx = I(x + 1, y) - I(x - 1, y) \quad (2)$$

$$dy = I(x, y + 1) - I(x, y - 1) \quad (3)$$

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

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

$$\sum wK(x, y) = 1 \quad (6)$$

$$\sum wI(x, y) = 1 \quad (7)$$

$$\sum wJ(x, y) = 1 \quad (8)$$

Las ecuaciones de la 2-8 se emplean en el proceso de cálculo de descriptores en SIFT. Estas ecuaciones también se utilizan para ponderar el peso en cada celda de acuerdo con la magnitud del gradiente en direcciones horizontal y vertical. Esto da lugar a una triple ponderación, que es una interpolación trilineal. Este enfoque ayuda a mitigar los efectos del ruido y las pequeñas variaciones entre imágenes, lo que garantiza una clasificación precisa de los objetos. Un ejemplo de un kernel gaussiano que podría usarse en este contexto es el siguiente.

$$\frac{1}{21} \begin{bmatrix} 1 & 3 & 1 \\ 3 & 5 & 3 \\ 1 & 3 & 1 \end{bmatrix} \quad (9)$$

En el proceso, se utilizan pesos para cada píxel dentro de un intervalo (9) que se ponderan mediante una función gaussiana  $G(x, y)$ . Esto significa que los píxeles más cercanos al punto de interés tienen una influencia mayor, mientras que los ubicados en los bordes tienen menos importancia debido a su propensión al ruido.

$$h = (h_1 \ h_2 \dots h_n) \quad (10)$$

$$h_k = \sum_{x,y} w_i(x,y) w_j(x,y) w_k(x,y) m(x,y) G(x,y) \quad (11)$$

Es relevante mencionar que, si bien el histograma es robusto ante cambios en la iluminación, es sensible a variaciones en el contraste. Para contrarrestar este efecto, se normaliza cada uno de los histogramas dividiéndolos por la norma del conjunto (13-14).

$$H = (H_{(1,2)}, H_{(1,2)}, \dots H_{(n,n)}) \quad (12)$$

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

$$v^n = \left( \frac{v_1}{\|v\|}, \frac{v_2}{\|v\|}, \dots, \frac{v_m}{\|v\|} \right) \quad (14)$$

$$v = (v_1, v_2, \dots, v_m) \quad (15)$$

Se limita el efecto del brillo y contraste para prevenir cambios significativos debidos a fluctuaciones en la iluminación al restringir los valores a un máximo, típicamente establecido en 0.2.(15).

$$v^n = (\max(v_1, 0.2), \dots, \max(v_m, 0.2)) \quad (16)$$

$$\|v\|_2 = \sqrt{\sum_{i=1}^m v_i^2} \quad (17)$$

Se procede a aplicar una normalización adicional (señalada como (16)) al vector con el propósito de obtener el vector de características definitivo, identificado como (17).

$$v^m = \left( \frac{v_1}{\|v\|}, \frac{v_2}{\|v\|}, \dots, \frac{v_m}{\|v\|} \right) \quad (18)$$

### Características Robustas Aceleradas

En un estudio comparativo de descriptores y extractores de características en OpenCV, se destaca que el descriptor SIFT ofrece resultados sólidos en muchos casos, pero presenta una desventaja significativa debido a su alta complejidad y consumo de recursos. Para abordar este problema, se introdujo un algoritmo llamado SURF en un estudio realizado por Bay en 2008. SURF se basa en SIFT pero utiliza un enfoque alternativo, empleando un filtro de caja y la imagen integral en lugar de aproximaciones laplaciano-gaussianas.

En SURF, se localizan los puntos de interés utilizando la matriz Hessiana y se mide la variación en el entorno del punto utilizando el determinante. A diferencia de SIFT, no se redimensionan las imágenes, en su lugar, se ajusta el tamaño del filtro de caja, comenzando con un tamaño inicial de 9x9 y progresando a tamaños mayores. El uso de filtros gaussianos discretizados en SIFT puede llevar a la pérdida

de información y dificultar la repetibilidad. Para determinar la orientación, SURF utiliza wavelets de Haar en direcciones  $x$  e  $y$  dentro de una región circular con radio variable y encuentra la orientación predominante en una ventana de 60 grados. El descriptor en SURF se construye subdividiendo una región cuadrada en 16 subregiones, calculando el wavelet de Haar en cada una y suavizándolo con filtros gaussianos. El descriptor final es una concatenación de los descriptores de cada subregión.

### Algoritmo ORB

Busca de una alternativa a los algoritmos SIFT y SURF, cuya velocidad de respuesta mejorada se ve limitada por problemas de licencias, los laboratorios de OpenCV presentaron el algoritmo ORB. ORB se diseñó como una solución que no solo evita los problemas de patentes, sino que también mejora la eficiencia y reduce el consumo de recursos, además de ser de uso gratuito. Este algoritmo combina dos etapas distintas: detección y descripción.

**A) Para la etapa de detección, se utiliza el algoritmo FAST (Features from accelerated segment test)**, que compara la intensidad de un píxel con sus píxeles circundantes para identificar puntos de interés. A diferencia de SIFT, ORB modifica la orientación utilizando el vector derivado de las intensidades en relación con el centroide y calcula la invarianza con respecto a  $(x, y)$  en una región circular.

**B) En la etapa de descripción, se generan los descriptores utilizando el algoritmo BRIEF (Binary Robust Independent Elementary Features)**, desarrollado por Michael Calonder, Vincent Lepetit, Christoph Strecha y Pascal Fua. BRIEF crea descriptores básicos en forma de cadenas de bits mediante la comparación de intensidades de píxeles en una versión suavizada de la imagen. Dado que estos descriptores no son invariables a la rotación, ORB rota la región centrada en el píxel en incrementos de 12 grados, evaluando el descriptor BRIEF en cada iteración. Hasta este punto, los enfoques han sido de análisis local, sin considerar las relaciones entre píxeles distantes. Se exploraron alternativas, como la transformada de Hough, que considera relaciones globales, permitiendo la detección de bordes y formas en las imágenes.

### Transformada de Hough

Para la detección de regiones con características geométricas específicas, se empleó la técnica de la transformada de Hough, que es eficiente en la detección de contornos relacionados con formas geométricas como líneas rectas o circunferencias. Para detectar una línea recta, se parte de su ecuación en forma polar y se discretizan los rangos de valores en el espacio polar. Se definen rangos específicos los límites para la distancia desde el origen son los límites para el ángulo de inclinación. Estos rangos se aplican en un espacio de celdas conocido como celdas de acumulación.

Luego, se evalúa la ecuación de la línea en cada punto de la imagen, y si un punto cumple con la ecuación, se agrega un voto a la celda correspondiente en el espacio de acumulación. Un alto número de votos en una celda indica que esa celda representa una línea en la imagen. Esta técnica permite identificar líneas rectas de manera eficaz en una imagen, lo que es útil en aplicaciones que requieren detección de características geométricas específicas.

### Clasificador bayesiano

En la clasificación de puntos de interés con vectores descriptivos, se utiliza un clasificador bayesiano, que se basa en un enfoque probabilístico. El objetivo es encontrar la hipótesis más probable utilizando un conjunto de entrenamiento y calculando previamente la probabilidad de una hipótesis. Este clasificador también es conocido como idiotaBayes, naive Bayes, simple Bayes o Bayes independiente. Se basa en el teorema de Bayes, que calcula la probabilidad condicional de un evento  $A$  dado un evento  $B$ , utilizando las probabilidades de  $A$  y  $B$ , así como la probabilidad condicional de  $B$  dado  $A$ . Es importante tener en cuenta que este teorema requiere un conocimiento previo, y si ese conocimiento no está disponible, debe estimarse, lo que puede resultar en un alto costo computacional.

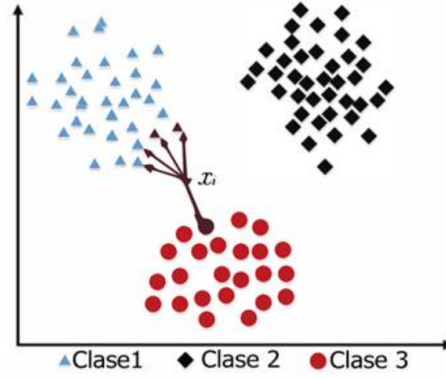


Figure 4: Ejemplo KNN con tres clases (Chong et al., 2013).

### Clasificador por K-Nearest neighbors

Otro enfoque de clasificación implica utilizar un método que se basa en la clasificación de un elemento en función de sus  $k$  vecinos más cercanos. Este método es de aprendizaje supervisado y se aprecia en la figura 4, donde se clasifica el elemento  $x_j$  en función de sus 5 vecinos más cercanos. El resultado de este enfoque indica a qué clase pertenece el elemento  $x_j$ , basándose en la mayoría de sus vecinos más cercanos.

## 4 DESARROLLO PROPUESTO

Para validar la propuesta, se llevaron a cabo pruebas siguiendo la metodología definida. Se utilizaron herramientas de desarrollo en el marco de trabajo OpenCV debido a sus ventajas, como la licencia BSD que permite su uso en contextos académicos y comerciales, así como la disponibilidad de rutinas que agilizan el procesamiento de imágenes. La estrategia para lograr un buen rendimiento en seguimiento y rastreo involucró la implementación de los algoritmos SIFT, SURF y ORB como extractores de características, el uso de  $k$ -means para la clusterización y  $k$ -nn. Además, se realizaron pruebas con un objeto (una zanahoria) en diferentes ambientes con cambios de luminosidad para evaluar la eficiencia del modelo creado.

En la Figura 4 se presentan algunos métodos de detección y clasificación de objetos que se describirán posteriormente. La Figura 5 identifica una secuencia general para llevar a cabo el seguimiento de objetos, mostrando las principales técnicas y métodos utilizados, que serán ampliados en las secciones posteriores. Una de las técnicas comunes para detectar objetos en movimiento es la sustracción de fondo, que distingue entre áreas estáticas (fondos) e imágenes en movimiento (primer plano) utilizando modelos dinámicos que se adaptan a las variaciones, como cambios en la intensidad de luz o la aparición de nuevos elementos en la escena.

Se mencionan enfoques alternativos, como el uso de funciones de probabilidad gaussiana para estimar el fondo más probable en cada entrada, aunque este método puede ser ineficaz en escenas complejas. También se exploran métodos de sustracción de fondo basados en análisis de texturas, que son robustos cuando el fondo y la referencia coinciden en color pero requieren un alto poder de procesamiento. Por otro lado, los métodos basados en modelos estocásticos ofrecen altas tasas de detección, bajo consumo computacional y resistencia al ruido, aunque pueden fallar cuando hay similitud entre la imagen y el fondo.

Existen diversas técnicas para la detección y clasificación de objetos. Una de ellas es la "diferencia de cuadros", que compara constantemente el cuadro actual con uno de referencia, analizando las diferencias para determinar si corresponden al fondo o al objeto a identificar. Aunque es computacionalmente económica, no es eficaz cuando la imagen y el fondo son similares.

Otra técnica es el "flujo óptico", que sigue el brillo de los píxeles para detectar movimientos. Sin embargo, puede fallar en desplazamientos grandes, ya que los píxeles pueden saltar y perderse en el seguimiento.





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

Finalmente, la "segmentación" divide una imagen en secciones para simplificar el análisis. Hay varios algoritmos disponibles para la segmentación, y algunos son más eficientes que otros, pero requieren más recursos computacionales. Una vez que se detectan estructuras o rasgos distintivos del objeto, se crean descriptores que representan el objeto, se comparan con descriptores de entrenamiento y se clasifican.

La clasificación puede basarse en el movimiento, las texturas, las formas o el color, cada uno con sus ventajas y desventajas en términos de costos computacionales y aplicaciones específicas.

## 5 RESULTADOS

En la creación de un modelo de seguimiento de objetos, se utilizaron imágenes de entrenamiento de una zanahoria desde varias posiciones e iluminaciones. Se calcularon puntos clave y descriptores (SIFT, SURF y ORB) para estas imágenes, generando un total de 21,000 descriptores. Luego, se aplicó el algoritmo K-Means para reducirlos a 1,000 centroides relevantes.

Se desarrolló una aplicación en OpenCV para capturar video en tiempo real desde una cámara. Cada cuadro del video se procesó para calcular puntos clave y descriptores usando SIFT, SURF y ORB, y se compararon con los 1,000 centroides utilizando un clasificador de k vecinos más cercanos. La cámara se probó con una zanahoria en diferentes condiciones, incluyendo escenarios con otros objetos.

Los resultados de estas pruebas se detallan en las Figuras 6-11.

### *Resultados utilizando ORB*

La Figura 6 muestra los resultados obtenidos en la primera situación, aplicando el método ORB, mientras que la Figura 7 exhibe los resultados obtenidos en la segunda situación utilizando el método ORB.

*Resultados utilizando SURF* La Figura 8 exhibe los resultados obtenidos en el entorno inicial



Figure 6: Ambiente 1 - ORB.



Figure 7: Ambiente 2 - ORB.

utilizando el enfoque SURF, mientras que la Figura 9 muestra los resultados obtenidos en el segundo escenario aplicando el método SURF.



Figure 8: Ambiente 1 - SURF.



Figure 9: Ambiente 2 - SURF.

### ***Resultados utilizando SIFT***

La Figura 10 ilustra los resultados obtenidos en la primera situación empleando la técnica SIFT, mientras que la Figura 11 muestra los resultados logrados en el segundo contexto utilizando el método SIFT.



Figure 10: Ambiente 1 - SIFT.

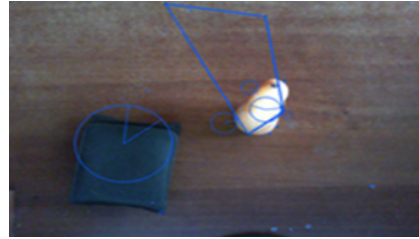


Figure 11: Ambiente 2 - SIFT.

Los resultados previos se generaron a partir de la captura de 3600 imágenes en un lapso de tiempo de un (1) minuto.

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

Los resultados previos se derivaron utilizando la fórmula (18).

$$[h]_{\text{accuracy}} = \frac{\text{Correct Clasiffiet}}{\text{Total number of images}} \quad (19)$$

De los resultados obtenidos a partir de la muestra definida para el análisis, se determinó que el método ORB demostró la mayor precisión, seguido de cerca por el método SIFT. Estos resultados iniciales respaldan hallazgos previos en estudios similares (Rublee et al., 2011; Tafti et al., 2018; Majumdar y

ALGORITMO	ACCURACY
SIF	0,62
SURF	0,74
ORB	0,80

Mahato, 2018), donde se destacó que ORB se destaca como uno de los algoritmos más eficientes para la detección de numerosas características y la coincidencia de imágenes en situaciones que involucren un gran número de características a lo largo del tiempo.

## 6 CONCLUSIONES

El vasto número de opciones y soluciones disponibles puede ser abrumador al seleccionar una específica, tanto para la detección de objetos como para su seguimiento. Para simplificar este proceso, es esencial definir claramente el alcance y los objetivos, ya que la cantidad de objetos puede influir significativamente en el rendimiento del algoritmo elegido. Los resultados obtenidos indicaron que los modelos desarrollados funcionaron bien en un entorno controlado, pero presentaron errores y una disminución notable en la precisión en un entorno no controlado. Para mejorar esto, se podría considerar un conjunto de entrenamiento más amplio, lo que permitiría realizar más iteraciones y abordar con mayor precisión aspectos en el análisis.

## 7 REFERENCIAS

@inproceedingsabd-el-hafiz2016permutation, title = Permutation Techniques Based on Discrete Chaos and Their Utilization in Image Encryption, author = Abd-El-Hafiz, S. K. and AbdElHaleem, S. H. and Radwan, A. G., booktitle = 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pages = 1-6, year = 2016, doi = 10.1109/ECTICon.2016.7561265, @inproceedingsabd-el-hafiz2016permutation, title = Permutation techniques based on discrete chaos and their utilization in image encryption, author = Abd-El-Hafiz, S. K. and AbdElHaleem, S. H. and Radwan, A. G., booktitle = 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pages = 1-6, year = 2016, doi = 10.1109/ECTICon.2016.7561265,

@articleagarwal2016six, title = Six object tracking algorithms: A comparative study, author = Agarwal, V. K. and Sivakumaran, N. and Naidu, V. P. S., journal = Indian Journal of Science and Technology, volume = 9, number = 30, pages = 1-9, year = 2016,

@phdthesisardila2014implementacion, title = Implementación del Voto Electrónico en Colombia: Un estudio desde la responsabilidad estatal, frente a fallas tecnológicas, author = Ardila, O. F., school = Universidad Militar Nueva Granada, year = 2014,

@articlearriagada2019deteccion, title = Detección y reconocimiento de señales de tránsito utilizando matching de chamfer, author = Arriagada, C. and Aracena-Pizarro, D., journal = Ingeniare. Rev. chil. ing., volume = 15, number = 2, pages = 174-184, year = 2019, doi = <http://dx.doi.org/10.4067/S0718-33052007000200008>,

@articlebay2008speeded, title = Speeded-up robust features (SURF), author = Bay, H. and Ess, A. and Tuytelaars, T. and Van Gool, L., journal = Computer vision and image understanding, volume = 110, number = 3, pages = 346-359, year = 2008, doi = <https://doi.org/10.1016/j.cviu.2007.09.014>,

@onlineopencv2000, title = The OpenCV Library, author = Bradski, G. and Kaehler, A., year = 2000, url = <https://opencv.org/>,

**Recibido:** 10 de Septiembre de 2019; **Aprobado:** 06 de Noviembre de 2019 **Autor a quien debe**

**ser dirigida la correspondencia.** Correo-e: [dfsantosb@correo.udistrital.edu.co](mailto:dfsantosb@correo.udistrital.edu.co)