

Taller No.4 Dense Optical Flow

Niño Rivera María Carolina, Machado Manuel
{maria.nino02}@usa.edu.co, {manuel.machado01}@usa.edu.co}
 Profesor: Rudas Jorge

I. INTRODUCCIÓN

En el área de visión por computador uno de los aspectos más importantes es el seguimiento de objetos, para esto se utiliza las técnicas y métodos de rastreo. El seguimiento de objetos es la tarea de identificar automáticamente objetos en un video e interpretarlos como un conjunto de trayectorias con alta precisión. A menudo, hay una indicación alrededor del objeto que se está rastreando, por ejemplo, un cuadrado circundante que sigue al objeto y muestra al usuario dónde se encuentra el objeto en la pantalla [11]. Pero este rastreo presenta algunas dificultades, como son las distracciones de fondo. Los fondos que están demasiado ocupados, tienen el mismo color que el objeto o están demasiado desordenados pueden dificultar el seguimiento de los resultados de un objeto pequeño o de un color claro [11]. Otro aspecto que puede ayudar a confundir a los algoritmos es la variedad de tamaños y relaciones de aspecto, debido a que pueden creer que el objeto tiene una escala mayor o menor que su tamaño real. Además se presentan problemas con la ocultación que ocurre cuando varios objetos se acercan demasiado (se fusionan) y se superponen [11]. Dentro del seguimiento de objetos se estudian los siguientes conceptos [12]:

1. Flujo óptico denso: Estos algoritmos ayudan a estimar el vector de movimiento de cada píxel en un cuadro de video.
2. Flujo óptico disperso: Estos algoritmos, como el rastreador de características Kanade-Lucas-Tomashi (KLT), rastrean la ubicación de algunos puntos característicos en una imagen.
3. Filtrado de Kalman: un algoritmo de procesamiento de señales muy popular que se utiliza para predecir la ubicación de un objeto en movimiento en función de la información de movimiento anterior. Una de las primeras aplicaciones de este algoritmo fue la guía de misiles, además cabe destacar que la computadora de a bordo que guió el descenso del módulo lunar Apolo 11 a la luna tenía un filtro Kalman.
4. Meanshift y Camshift: Estos son algoritmos para localizar los máximos de una función de densidad. También se utilizan para el seguimiento.
5. Rastreadores de un solo objeto: En esta clase de rastreadores, el primer cuadro se marca con un rectángulo para indicar la ubicación del objeto que queremos rastrear. A continuación, se realiza un seguimiento del objeto

El presente documento corresponde a un informe de práctica de laboratorio de Inteligencia Artificial presentado en la Universidad Sergio Arboleda durante el periodo 2022-2.

en fotogramas posteriores utilizando el algoritmo de seguimiento. En la mayoría de las aplicaciones de la vida real, estos rastreadores se usan junto con un detector de objetos.

6. Seguimiento de objetos múltiples con reidentificación: En los casos en que tenemos un detector de objetos rápido, tiene sentido detectar varios objetos en cada cuadro y luego ejecutar un algoritmo de búsqueda de seguimiento que identifique qué rectángulo en un cuadro corresponde a un rectángulo en el cuadro siguiente .

II. FLUJO ÓPTICO

El flujo óptico es el patrón de movimiento aparente de los objetos de la imagen entre dos fotogramas consecutivos causado por el movimiento del objeto o la cámara. Es un campo vectorial 2D donde cada vector es un vector de desplazamiento que muestra el movimiento de los puntos del primer cuadro al segundo. Considere la imagen 1 donde se aprecia el cambio entre frames de una imagen [1]. El uso del flujo óptico se puede aplicar a áreas como: Estructura del movimiento, comprensión de vídeo y estabilización de vídeo. Presenta dos puntos bases [1].

1. Las intensidades de píxel de un objeto no cambian entre fotogramas consecutivos.
2. Los píxeles vecinos tienen movimiento similar.

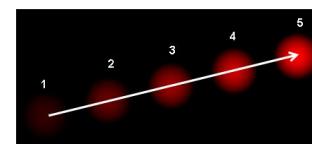


Figura 1. Imagen que muestra un ejemplo del flujo óptico donde el objeto se desplaza en 5 cuadros consecutivos
 [1]

El flujo óptico se encuentra comúnmente en editores de vídeo para compresión, estabilización, cámara lenta, etc. Además, el flujo óptico encuentra su aplicación en tareas de reconocimiento de acciones y sistemas de seguimiento en tiempo real. Hay dos tipos de flujo óptico, y el primero se llama flujo óptico disperso. Calcula el vector de movimiento para el conjunto específico de objetos (por ejemplo, esquinas detectadas en la imagen). Por lo tanto, requiere un preprocesamiento para extraer características de la imagen, que serán la base para el cálculo del flujo óptico. OpenCV proporciona algunas implementaciones de algoritmos para resolver la tarea Flujo óptico disperso [2]:

- Pirámide Lucas-Kanade.
- RLOF escaso.

Usar solo un conjunto de características escaso significa que no se tiene la información de movimiento sobre los píxeles que no están contenidos en él. Esta restricción se puede eliminar utilizando algoritmos de flujo óptico denso que calculan un vector de movimiento para cada píxel de la imagen. Algunos algoritmos de flujo óptico denso ya están implementados en OpenCV [2]:

- Pirámide Densa Lucas-Kanade.
- Farneback.
- PCAFlujo.
- Flujo simple.
- RLOF.
- Flujo profundo.
- DualTVL1

III. FLUJO ÓPTICO DISPERSO: MÉTODO LUCAS-KANADE

El método de Lucas-Kanade (1981) desarrolla una técnica diferencial ampliamente usada para la detección del movimiento y estimación del flujo óptico, se usa comúnmente para un conjunto de características dispersas, el principal concepto es la suposición de que el flujo es esencialmente constante en las proximidades espaciales a un píxel en la imagen bajo consideración. Bajo este planteamiento, la base del mismo consiste en determinar tanto variaciones espaciales en las direcciones x e y de la escena como variaciones temporales. Se introduce así una tercera dimensión en el tratamiento de las imágenes, la cual es temporal [4]. La idea principal de este método se basa en una suposición de constancia de movimiento local, donde los píxeles cercanos tienen la misma dirección de desplazamiento [2].

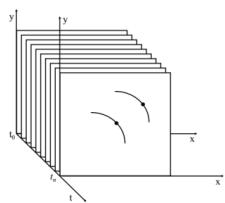


Figura 2. Representación de una secuencia temporal (t) de imágenes con dimensiones espaciales x e y

[4]

Los algoritmos de flujo óptico realmente sufren de movimientos bruscos debido a las limitaciones del algoritmo. El enfoque común en la práctica es utilizar un truco de escala múltiple. Para eso se necesita crear una llamada pirámide de imágenes, donde cada imagen siguiente será más grande que la anterior por algún factor de escala (por ejemplo, el factor de escala es 2). En cuanto a la ventana de tamaño fijo, el movimiento brusco en la imagen de tamaño pequeño es más notorio que en la grande. Los vectores de desplazamiento encontrados en las imágenes pequeñas se utilizarán en las siguientes etapas piramidales más grandes para lograr mejores resultados [2]. Una pirámide de imágenes es un conjunto de imágenes con tamaño decreciente o creciente, se les llama de este modo

por la forma de pirámide que se origina al organizarlas en forma consecutiva una sobre la otra. Entre los métodos más conocidos para la generación de las pirámides de imagen se tiene: Pirámides Gaussianas y Pirámides Laplacianas. [3]

El método Lucas-Kanade toma un parche de 3×3 alrededor del punto. Así que todos los 9 puntos tienen la misma moción. Dentro de esta explicación el concepto más simple se basa en que dados algunos puntos a rastrear, se reciben los vectores de flujo óptico de esos puntos. Este método falla cuando hay movimientos grandes, para tal fin se utilizan las pirámides, en donde, los movimientos pequeños son removidos y los movimientos grandes se convierten en movimientos pequeños [1].

Uno de los limitantes de este método radica en que solo rastrea los puntos iniciales, es decir, si se introduce otro elemento en el vídeo no lo rastreará ni tomara ningún punto de interés, para mejorar esto se han realizado las siguientes modificaciones.

1. Definir una variable de intervalo “self.detect_interval = 5” en este ejemplo se va a realizar cada 5 iteraciones.
2. Se realiza la operación “cv.calcOpticalFlowPyrLK” dos veces la primera con la imagen anterior y la actual con los puntos iniciales detectados o p0. La segunda vez con la imagen actual, la anterior y los puntos claves recuperados al calcular la primera función. Esto con el fin de retrocederse y encontrar la distancia que hay entre el punto actual y como se encontraba inicialmente.
3. diferencia de distancia de los puntos p0 y p0r: Es decir encontrando la distancia de cambio, si la distancia es menor significa que es un buen punto clave por lo tanto lo conservará, caso contrario lo eliminará.
4. Por medio del operador modulo % se valida que cada intervalo definido ingrese y busque nuevos puntos claves por medio del método “goodFeaturesToTrack”

IV. DENSE OPTICAL FLOW

Los algoritmos de flujo óptico denso calculan el vector de movimiento para el conjunto de características escasas, por lo que el enfoque común aquí es usar el detector de esquina Shi-Tomasi. Se utiliza para encontrar esquinas en la imagen y luego calcular el vector de movimiento de las esquinas entre dos fotogramas consecutivos [2]. Una diferencia importante en este método es la conversión que se realiza de la imagen a HSV con el fin de lograr una mejor separación y detección de los colores. En este espacio de color, cada color trabaja con 3 componentes básicas: matiz (Hue), saturación (Saturation) y luminancia (Value). El matiz hace referencia al valor de cromaticidad o clase de color. La saturación se refiere a las longitudes de onda que se suman a la frecuencia del color y determinan la cantidad de blanco que contiene un color. La luminancia se corresponde con la apreciación subjetiva de claridad y oscuridad [7].

La salida del algoritmo de flujo óptico denso se puede codificar como el esquema de color HSV. Usando la cv2.cartToPolarfunción, se puede convertir las coordenadas de desplazamiento(dx, dy)en coordenadas polares como magnitud

y ángulo para cada píxel. Debido a esto se puede codificar el ángulo y la magnitud como Matiz y Valor respectivamente, mientras que la Saturación permanece constante. Para mostrar el flujo óptico correctamente, es necesario convertir el HSV al formato BGR [2].

Para continuar con un Lucas-Kanade, OpenCV permite usar este método no solo para tareas escasas sino también para el cálculo de flujo óptico denso. La técnica principal aquí es usar la salida del algoritmo Sparse y hacer la interpolación en toda la imagen para obtener el vector de movimiento para cada píxel [2].

IV-A. Algoritmo de Farneback

Este algoritmo nace en 2003 y se utiliza para calcular el flujo óptico de cada píxel de la imagen. La idea principal de este método es aproximar algunos vecinos de cada píxel con un polinomio. En términos generales, en el método de Lucas-Kanade se usa la aproximación lineal ya que solo se tiene la expansión de Taylor de primer orden. Ahora, se aumenta la precisión de la aproximación con valores de segundo orden. La idea lleva a observar las diferencias en los polinomios aproximados causados por desplazamientos de objetos. El algoritmo OpenCV Farneback requiere una imagen de entrada unidimensional, por lo que se debe convertir la imagen BRG en escala de grises [2].

V. MEAN SHIFT

En 1975, Fakunaga y Hostetter propusieron un algoritmo llamado Mean-shift (MS) object-Tracking Approach. Es básicamente una maximización iterativa de la expectativa: un algoritmo de agrupamiento ejecutado dentro de las regiones de búsqueda locales [9]. El algoritmo de desplazamiento Mean Shift es en realidad un concepto estadístico más amplio relacionado con la agrupación. Como la mayoría de los otros algoritmos de agrupamiento, el algoritmo intenta buscar lugares en el conjunto de datos con una alta concentración de puntos de datos o agrupaciones. El algoritmo coloca un kernel en cada punto de datos y los suma para hacer una estimación de la densidad del kernel (KDE). El KDE tendrá picos y valles, que corresponden a lugares con una densidad de puntos de datos alta y baja, respectivamente [8]. El algoritmo toma una copia de los puntos de datos y desplaza los puntos en una pequeña cantidad a su pico KDE más cercano en una sola iteración. El algoritmo continuará pasando por iteraciones de desplazamiento de los puntos hasta que los puntos ya no se muevan mucho. Esto puede ayudar a simplificar los datos y hacer que los grupos sean más pronunciados [8]. Los siguientes pasos se iteran para rastrear el objeto usando el algoritmo Mean-Shift [9]:

1. Seleccione un tamaño de ventana de búsqueda y la posición inicial de la ventana de búsqueda.
2. Estime la posición media en la ventana de búsqueda.
3. Centre la ventana de búsqueda en la posición media estimada en el Paso 2.
4. Repita los pasos 2 y 3 hasta que la posición media se mueva menos que un umbral preestablecido. Es decir, hasta que se logre la convergencia.

VI. CAMSHIFT

El algoritmo fue propuesto por primera vez por Fukunaga en la década de 1970. Es un tipo de agrupación de densidad, que utiliza operaciones iterativas continuas para buscar la posición del punto de muestra con la mayor densidad de probabilidad en el espacio de características. La dirección de búsqueda siempre cambia hacia la dirección donde la densidad de puntos de muestra aumenta más y la tendencia del objeto en movimiento se analiza y juzga de acuerdo con estos datos, para rastrear el objeto en movimiento [10].

La desventaja del algoritmo de Mean Shift es que el rectángulo de detección no se puede cambiar de forma adaptativa con el tamaño del objeto. Cuando la distancia entre el objeto y la cámara cambia o hay una ligera perturbación desde el exterior, es fácil perder el rastro o no seguir el rastro del objeto. El algoritmo Camshift se creó para resolver las deficiencias del algoritmo de desplazamiento medio. El algoritmo de seguimiento Camshift utiliza el algoritmo de seguimiento de objeto de desplazamiento medio para cada fotograma de la secuencia de video. El resultado del seguimiento de cada imagen determina el valor inicial de la iteración de seguimiento en la siguiente imagen y la ventana de seguimiento cambiará automáticamente a medida que cambie el objeto [10].

Los pasos de implementación de este algoritmo son [10]:

1. Inicializar la ventana de búsqueda: Seleccione el objeto en movimiento a rastrear y calibre la ventana de búsqueda inicial, que contiene únicamente el objeto de seguimiento completo. Simultáneamente, la imagen debe convertirse al espacio de color HSV (Tono, Saturación, Valor) porque el modelo RGB (Rojo, Verde, Azul) es más sensible a los cambios de iluminación, lo que reducirá el impacto de la iluminación.
2. Determine la distribución de probabilidad de color de la ventana: Extraiga el canal de tono en la secuencia de imágenes de video de la ventana seleccionada para obtener el modelo de histograma de objeto y la tabla de búsqueda de probabilidad de color. Consulte el modelo de histograma de color del objeto para cada píxel en la secuencia de video subsiguiente para determinar la probabilidad de que el píxel sea el píxel del objeto. Finalmente, vuelva a proyectar la imagen.
3. Ejecute el algoritmo de desplazamiento medio para obtener el tamaño, la posición y el ángulo de la nueva ventana de búsqueda.
4. Utilice el valor del paso 3 para reiniciar el tamaño y la posición de la ventana de búsqueda en el siguiente fotograma del vídeo. Utilice la nueva ventana de búsqueda y repita a partir del paso 2.

VII. RESULTADOS TALLER RASTREO PRIMERA PARTE

Para el taller de rastreo se utilizaron los algoritmos de Dense Optical Flow, Mean Shift, Camshift y Lucas Kanade y Lucas Kanade mejorado. Se trabajaron en el video utilizado en el tutorial de OpenCV “vtest.avi” [13]. El taller se dividió en dos partes, la primera corresponde a la ejecución e identificación

manual de los puntos clave hallados al utilizar el algoritmo Dense Optical Flow. Una vez seleccionados estos puntos se pasaron a los siguientes algoritmos. La segunda parte se trato de correr los algoritmos pero esta vez dejando que sea el propio algoritmo de flujo óptico a través la función “goodFeaturesToTrack” que encuentre los puntos claves y pasandolos al resto de algoritmos. Se identifican los resultados de los diferentes algoritmos a continuación.

VII-A. Dense Optical Flow

El primer paso de este taller se utilizó para encontrar los puntos claves de los objetos en movimiento, de manera manual se encuentran los centroides de estos puntos, la figura 3 muestra el resultado en la transformación de las imágenes en HSV.



Figura 3. Imagen que muestra la ejecución del algoritmo Dense Optical Flow

VII-B. Mean Shift

Al seleccionar los puntos manuales en la imagen 3, se utilizaron 3 puntos claves para este algoritmo, que corresponden con las primeras personas que aparecen. En la figura 4 se aprecia la trayectoria de desplazamiento que hizo la ventana creada, se utilizó la cintura de las personas como punto inicial en los puntos 310, 115 con un tamaño de 20*20. La figura 4(a) muestra como se marca la primera posición, en la figura 4(b) se aprecia como debido a una obstrucción en este caso por un cartel y al pasar otra persona la ventana de rastreo cambia de posición y comienza a seguirla, al final en la imagen 4(c) se aprecia como la persona ha desaparecido de escena y la ventana ya no puede seguir a nadie más. En la figura 5 se aprecian los resultados obtenidos en las coordenadas 380, 165 con un tamaño de 40*40. Este seguimiento fue mejor que el anterior pudo rastrear más personas, de igual manera que en el anterior, la oclusión del letrero provocaba un cambio en el seguimiento, excepto en la figura 5(d) en la que la persona se movió un poco más arriba y logró hacer un seguimiento más largo a esta persona. Por último al utlizar el punto con coordenadas 140,145 de tamaño de 40*40 los resultados no fueron muy buenos, la figura 6 muestra como el algoritmo luego de ubicar el centroide en el cuerpo de esa persona, no realiza el tracking y al no pasar nadie más por este punto, no se vuelve a mover. Para la tercera ejecución se utilizó los puntos 140,145 con un tamaño de ventana de 40*40, al igual que en el caso anterior los resultados no fueron buenos, en este caso la ventana se creció pero no realizó ningún seguimiento, el resultado de esto se muestra en la figura 9



(a) Punto Inicial

(b) Cambio de objeto de seguimiento



(c) Punto final de seguimiento

Figura 4. Resultados MeanShift coordenadas manuales 310,115



(a) Punto Inicial

(b) Cambio de objeto de seguimiento



(c) Nuevo Cambio de seguimiento (d) Punto donde realizó el seguimiento más largo

Figura 5. Resultados MeanShift coordenadas 380,165



Figura 6. Resultados MeanShift coordenadas manuales 140,145

VII-C. Camshift

Para este ejercicio se utilizó las mismas posiciones encontradas en el método Dense Optical Flow. Los resultados encontrado en las posición inicial 310, 115 se evidencia en



(a) Punto Inicial

(b) Cambio de tamaño ventana



(a) Punto Inicial

(b) Trayectoria de cambio



(c) Seguimiento a una persona

Figura 7. Resultados Camshift coordenadas manuales 310,115



a) Punto Inicial

(b) Cambio de tamaño ventana

Figura 11. Resultados Lucas Kanade Mejorado Puntos Manuales



(a) Punto Inicial

(b) Cambio de objeto de seguimiento y de tamaño de la ventana



Figura 9. Resultados Camshift coordenadas manuales 140,145

VIII. LUCAS KANADE

Para la ejecución de este algoritmo se utilizaron 5 puntos para tratar de cubrir más puntos clave iniciales, los cuales fueron [310, 115], [380, 165], [140, 145], [435, 175], [460, 185]. De los cuales los 3 primeros fueron utilizados en los resultados anteriores. Como se aprecia en la figura 10(a) el punto con coordenadas 140,145 resultó también ser un punto que no aportaba valor como se demostró previamente. Sin embargo en un momento una de las personas pasa por esta zona y se genera un rastreo, los resultados se observan en la figura 10(b).



Figura 8. Resultados Camshift coordenadas 380,165

IX. LUCAS KANADE MEJORADO

En este ejercicio se trabajaron los 5 puntos que se trabajaron en el método de flujo óptico. Se apreció que se realizaron algunos seguimientos pero no muchos, en este caso los puntos claves seleccionados se quedaron casi estáticos. En la figura 11(a) se muestra los primeros segundos de ejecución y en la figura 11(b) un momento donde se tenían 5 puntos clave y se realizó el seguimiento pequeño de la mujer en chaqueta negra donde justo estaba el punto clave.

X. RESULTADOS TALLER RASTREO SEGUNDA PARTE

Para la segunda parte del taller se utiliza la función “cv.goodFeaturesToTrack”. Se trata de la extracción de puntos clave por medio del detector de esquinas de Shi-Tomasi. Es una mejora del algoritmo de Harris en donde se toman los puntajes para determinar si se trata de una esquina. Entonces, para un píxel, si esta puntuación R es mayor que cierto umbral, ese píxel se considera una esquina. Similar al Harris Corner Detector si trazamos esto en el espacio $\lambda_1 - \lambda_2$, obtendremos el gráfico mostrado en la figura 12 [5]:

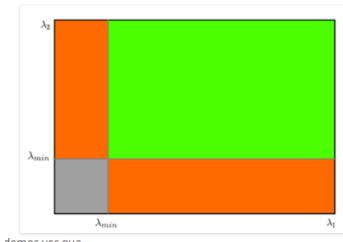


Figura 12. Gráfico del umbral

[5]

En este gráfico se visualizan las siguientes características:

1. Solo cuando λ_1 y λ_2 están por encima de un valor mínimo, λ_{min} , se considera como una esquina (región verde)
2. Cuando λ_1 o λ_2 están por debajo de un valor mínimo, λ_{min} , se considera como un borde (región naranja)
3. Cuando tanto λ_1 como λ_2 están por debajo de un valor mínimo, λ_{min} , se considera como una región plana (región gris)

X-A. Optical Flow

Al utilizar la función el sistema reconoció 35 puntos claves, aunque muchos de ellos se presentaron en partes que se encontraban quietas, las personas que aparecen luego que el sistema inicia no se pudieron rastrear, y algunos puntos perdieron el rastreo posteriormente al pasar otra persona utilizó este punto abandonado para rastrearlo. En la imagen 13 Se aprecia con el círculo rojo los puntos estáticos y en color negro el punto abandonado.



Figura 13. Resultado de la ejecución con goodFeaturesToTrack en el método de Flujo Óptico

X-B. Lucas Kanade Mejorado

Con las ventajas que representa este método, se pudieron rastrear personas que entraban luego de la detección de puntos

clave inicial, al igual que en el anterior detecto punto quietos como el auto y algunas imperfecciones en el suelo, estos puntos permanecieron activos durante todo el video. La imagen 14 muestra como se aumentaban los puntos claves permitiendo el rastreo de más personas.



Figura 14. Resultado de la ejecución con goodFeaturesToTrack en el método de Lucas Kanade Mejorado

X-C. Meanshift

Al utilizar los 35 puntos encontrados en el flujo óptico se encontró que la mayoría de puntos no generaba un rastreo debido a dos factores el primero cuando eran en partes fijas y el segundo debido a cambios de tamaños o perspectiva de tamaño en el video, sin embargo en los siguientes tres puntos se logró realizar un buen seguimiento, siendo similares a los puntos 310,115 y 140,145

1. 167, 140
2. 187, 123
3. 320, 102

X-D. Camshift

Debido a que este método permite una redimensión de la ventana la mayoría de los puntos salvo los que estaban quietos y en lugares muy alejados donde transitan las personas se redimensionan de una forma que permanecía la mayor parte del tiempo en el centro, el mejor punto encontrado para este método fue el punto 167,140.

XI. CONCLUSIONES

Para poder realizar un buen seguimiento debemos partir desde la base de encontrar buenos puntos clave para esto también se debe utilizar la teoría de cambio del flujo óptico debido a que es necesario tener la información de la dirección del cambio que presenta el objeto al cuál queremos realizar un seguimiento. Cada método presenta ventajas y desventajas que pueden ser aprovechadas dependiendo el caso que se desea estudiar, para el ejercicio notamos que el mejor rastreo realizado lo hizo con el método de Lucas Kanade mejorado. Además, se ha notado que la utilización del método de Dense Optical Flow mejora la adquisición de los puntos iniciales en los métodos de Meanshift y Camshift, ya que el seguimiento lo realizó mejor con los puntos ingresados manualmente. También se notó que al utilizar los puntos con Dense Optical Flow redujo los puntos estáticos.

REFERENCIAS

- [1] *Unipython Flujo Óptico Lucas-Kanade y Gunner Farneback 2018* <https://unipython.com/flujo-optico-lucas-kanade-y-gunner-farneback/>. Recuperado el 24 de Febrero de 2023.
- [2] *LearnOpenCV Flujo Óptico Lucas-Kanade y Gunner Farneback Maxim Kuklin 2021/01/04* <https://learnopencv.com/optical-flow-in-opencv/>. Recuperado el 26 de Febrero de 2023.
- [3] Pirámides de Imágenes con OpenCV <http://acodigo.blogspot.com/2017/02/piramides-de-imagenes-con-opencv.html>. Recuperado el 02 de Octubre de 2022.
- [4] Vaz Desarrollo de una aplicación informática para el análisis de imágenes térmicas. *Sepulcre Vaz, Alejandro* <https://eprints.ucm.es/id/eprint/48850/1/042.pdf>. Recuperado el 24 de Febrero de 2023.
- [5] *TheAILearner TAG ARCHIVES: CV2.GOODFEATURESTOTRACK()* 2020 <https://theailearner.com/tag/cv2-goodfeaturestotrack/>. Recuperado el 24 de Febrero de 2023.
- [6] *cv.calcOpticalFlowPyrLK* MATLAB File Help 2023 <http://amroamroamro.github.io/mexopencv/matlab/cv.calcOpticalFlowPyrLK.html>. Recuperado el 24 de Febrero de 2023.
- [7] *Evaluación del estado de maduración de la piña en su variedad perolera mediante técnicas de visión artificial* Angel, Luis 2012 https://www.researchgate.net/figure/IMAGEN-DE-LA-PINA-EL-ESPACIO-RGB-Y-EN-EL-ESPACIO-HSV_fig3_266854636. Recuperado el 24 de Febrero de 2023.
- [8] *Mean Shift and Cam Shift Object Tracking* Udell, Andrew 2021/02/10 <https://towardsdatascience.com/mean-shift-and-cam-shift-object-tracking-f1c2c515b6bc>. Recuperado el 27 de Febrero de 2023.
- [9] *Object Tracking in Video Using Mean Shift Algorithm: A Review* Shivhare, Aparna; Dr Choudhary, Vinita 2015, ISSN: 0975-9646 <https://ijcsit.com/docs/Volume%206/vol6issue04/ijcsit20150604106.pdf>. Recuperado el 27 de Febrero de 2023.
- [10] *Improved Camshift object tracking algorithm in occluded scenes based on AKAZE and Kalman Pei, Lili; Zhang, He, Vinita; Yang Bo 2021/10/21, ISSN: 2145-2159* <https://link.springer.com/article/10.1007/s11042-021-11673-7>. Recuperado el 27 de Febrero de 2023.
- [11] *Object Tracking in Computer Vision (2023 Guide)* Klinger, Nico 2023 <https://viso.ai/deep-learning/object-tracking/>. Recuperado el 27 de Febrero de 2023.
- [12] *The Complete Guide to Object Tracking – OpenCV, DeepSort, FairMOT* Learn Open CV 2023 <https://learnopencv.com/the-complete-guide-to-object-tracking-in-computer-vision/>. Recuperado el 27 de Febrero de 2023.
- [13] *Optical Flow OpenCV 2023* https://docs.opencv.org/4.x/d4/dee/tutorial_optical_flow.html. Recuperado el 27 de Febrero de 2023.