

**Universidad de Pamplona**  
**Facultad de Ingenierías y Arquitectura**  
**Programa de Ingeniería de Sistemas**

**Tema:**

**Desarrollo de un sistema de visión artificial para el seguimiento de objetos en el espacio  
y determinación de sus trayectorias**

**Autor:**

**Andrés David Cohen Gamarra**

**Pamplona, Norte de Santander**

**Junio 2020**

**Universidad de Pamplona**

**Facultad de Ingenierías y Arquitectura**

**Programa de Ingeniería de Sistemas**

**Trabajo de grado presentado para optar al título de Ingeniera de Sistemas.**

**Tema:**

**Desarrollo de un sistema de visión artificial para el seguimiento de objetos en el espacio  
y determinación de sus trayectorias**

**Autor:**

**Andrés David Cohen Gamarra**

**Director:**

**José Orlando Maldonado Bautista**

**Doctor en ingeniería informática**

**Pamplona, Norte de Santander.**

**Junio 2020.**

## Resumen

La visión artificial es una disciplina que está siendo ampliamente estudiada en las últimas décadas. Los sistemas de visión artificial han sido utilizados con gran éxito en procesos industriales tales como automatización, inspección o control de calidad, en sistemas de vigilancia o de seguridad, en procesos médicos, etc. Sus objetivos y ventajas van desde la mejora en la velocidad de los procesos a la automatización de tareas que no pueden hacerse de otro modo. Por tal motivo se presenta el desarrollo de un sistema para el seguimiento de objetos, ubicación y generación de trayectorias en un espacio tridimensional en tiempo real. Este sistema está desarrollado en el lenguaje de programación Python, utilizando técnicas de visión por computador e inteligencia artificial, se estudiaron diferentes herramientas, técnicas y algoritmos de los cuales se seleccionaron y adaptaron los mejores algoritmos.

A partir de los métodos base, se desarrollaron los algoritmos que dieron la solución a los requerimientos establecidos. Estos algoritmos están embebidos en un prototipo funcional que facilita su operación mediante una interfaz gráfica que permite la configuración del sistema de cámaras de seguimiento. Todos los algoritmos están totalmente documentados. Esto con fines didácticos. El sistema fue evaluado en diferentes escenarios controlados donde se determinó el error de seguimiento, ubicación y generación de trayectorias.

**Abstract**

Artificial vision is a discipline that is being studied in recent decades. Artificial vision systems have been used with great success in industrial processes such as automation, inspection or quality control, surveillance or security systems, medical processes, etc. Its objectives and advantages range from the improvement in the speed of the processes to the automation of tasks that cannot be done otherwise. For this reason, the development of a system for object tracking, location and trajectory generation in a three-dimensional space in real time is presented. This system is developed in the python programming language, using computer vision techniques and artificial intelligence, studying different tools, techniques and algorithms from which the best algorithms are selected and adapted.

From the base of methods, develop the algorithms that specified the solution to the established requirements. These algorithms are embedded in a functional prototype that facilitates their operation through a graphical interface that allows the configuration of the tracking camera system. All algorithms are fully documented with didactic fines. The system was evaluated in different controlled scenarios where the tracking error, location and trajectory generation is determined.

## **Agradecimientos**

Agradecerle primeramente a Dios, quien con su bendición llena siempre mi vida. Gracias a mis padres: Omaira del Carmen gamarra y Donaldo Antonio Cohen, por ser los principales promotores de mis sueños, por confiar y creer en mí, por los consejos, valores y principios que me han inculcado.

A todos esos amigos ya que esto no hubiese sido posible sin la cooperación desinteresada de todas y cada una de las personas que me acompañaron en el recorrido laborioso de este trabajo y muchas de las cuales han sido un soporte muy fuerte en momentos de angustia y desesperación.

Agradecerle a mi tutor José Orlando Maldonado Bautista quien estuvo siempre presente guiándome con paciencia en cada paso dado disponiendo de su tiempo para la elaboración de ese proyecto.

Y, por último, pero no menos importante a todos mis profesores por haber compartido sus conocimientos a lo largo de la preparación de la profesión.

# Tabla de contenidos

1	Introducción .....	11
1.1	Planteamiento del problema y justificación .....	11
1.2	Delimitación .....	12
1.2.1	Objetivo General:.....	12
1.2.2	Objetivos Específicos.....	12
1.3	Acotaciones .....	12
1.4	Metodología.....	13
1.4.1	Tipo de investigación.....	13
2	Marco teórico y estado del arte .....	14
2.1	Marco conceptual.....	14
2.2	Análisis de video.....	20
2.2.1	Seguimiento de objetos .....	21
2.2.2	Clasificación de objetos .....	22
2.3	Características de los algoritmos de seguimiento .....	24
2.3.1	Región de interés .....	24
2.3.2	Estimación del movimiento.....	25
2.4	Visión estereoscópica.....	25
2.4.1	Modelo de cámara: .....	26
2.4.2	Longitud focal .....	27
2.4.3	Distorsión de la lente .....	28
2.4.4	Calibración con OpenCV .....	29
2.4.5	Visión estéreo.....	30
2.4.6	Triangulación .....	30
2.4.7	Geometría Epipolar .....	32
2.4.8	Matrices Esenciales y Fundamentales .....	32
2.5	Rectificación Estéreo .....	33
2.5.1	Algoritmo Hartley .....	34
2.5.2	Algoritmo de Bouguet .....	34

2.6	Métricas estándar más utilizadas en modelos de regresión. ....	34
2.7	Modelo Visión estéreo .....	36
2.8	Estado del arte.....	38
2.8.1	Internacional: .....	38
2.8.2	Nacional: .....	39
2.8.3	Regional: .....	41
3	Análisis y evaluación de los algoritmos .....	43
3.1	Retos y dificultades.....	43
3.2	Análisis de los algoritmos de seguimiento planteados y estudiados.....	44
3.3	Experimentación de los algoritmos seleccionados .....	49
3.4	Flujograma del Algoritmo para la calibración de las cámaras .....	58
3.5	Calibración .....	59
3.5.1	Calibración de distorsión .....	60
3.5.2	Calibración de la cámara estéreo.....	60
3.5.3	Cálculo del mapa de disparidad .....	61
3.6	Calculando la distancia desde las cámaras al objeto.....	62
3.7	Reconstrucción de la escena 3D .....	63
3.8	Seguimiento de los objetos.....	63
3.9	Integración de los algoritmos .....	64
3.10	Reconstrucción del seguimiento de un objeto.....	67
3.11	Resultados .....	68
4	Descripción del prototipo de visión artificial para el seguimiento de objetos desarrollado ..	69
4.1	Descripción detallada del prototipo de visión artificial creado.....	69
4.2	Arquitectura del sistema realizado .....	73
4.3	¿Por qué utilizar Opencv?.....	75
4.4	Algoritmos utilizados .....	76
4.5	Pruebas Finales.....	77
4.5.1	Resultados.....	81
4.6	Error del modelo final con la métrica MAE .....	82
4.7	Conclusiones .....	83

5 Bibliografía .....	84
----------------------	----



## Tabla de Figuras

Figura 1: Representación de un punto en una imagen.....	16
Figura 2: Intensidad de una imagen.....	16
Figura 3: Representación de un gradiente.....	17
Figura 4: Esquema representativo para el análisis de un video) .....	18
Figura 5: Representación del método-kernel .....	18
Figura 6: Representación del método-silueta.....	19
Figura 7: Representación del método punto.....	19
Figura 8: cuadro de diferentes texturas.....	20
Figura 9: Reconocimiento de objetos a través del Color .....	20
Figura 10: Reconocimiento de objetos a través de formas .....	21
Figura 11: Repetitividad y Distinción de una imagen .....	21
Figura 12: Funcionamiento de una cámara. Modelo pinHole.....	24
Figura 13: Como se proyecta un objeto. Modelo pinhole.....	25
Figura 14: Distorsión radial de las cámaras .....	26
Figura 15: Distorsión tangencial de las cámaras. ....	27
Figura 16: Tablero de ajedrez para calibración de cámaras.....	27
Figura 17: Triangulación de cámaras hacia un objeto .....	28
Figura 18: Triangulación de dos cámaras .....	29
Figura 19: Prueba del algoritmo CSRT con pelotas de colores .....	48
Figura 20: Prueba del algoritmo de color con pelotas de colores .....	48
Figura 21: Prueba del algoritmo CSRT con pelota de ping pong .....	49
Figura 22: Prueba del algoritmo de color con pelota de ping pong.....	50
Figura 23: Prueba del algoritmo CSRT en escenario de pérdida parcial de objetos .....	52
Figura 24: Prueba del algoritmo de color con pérdida parcial del objeto.....	52
Figura 25: Prueba del algoritmo de color con objetos a la distancia.....	52
Figura 26: Prueba del algoritmo CSRT con escenario de sombras .....	54
Figura 27: flujograma algoritmo para la calibración de cámaras .....	54
Figura 28: Captura de cámaras sin calibración .....	56
Figura 29: captura de cámaras calibrada.....	56
Figura 30: Regresión lineal para obtener ecuación de distancia .....	56
Figura 31: Ecuación de línea recta .....	57
Figura 32: Flujograma del algoritmo resultante .....	58
Figura 33: Control inicial de parámetros de ejecución .....	59
Figura 34: Medición y seguimiento de objeto detectado .....	60
Figura 35: Distancia real 87 cm .....	60
Figura 36: Frames representativos del seguimiento hechos al objeto .....	61
Figura 37: Grafica de los puntos 3D seguidos del objeto .....	61
Figura 38: Errores de la reconstrucción 3D .....	62
Figura 39: Grafica de los puntos 3D del eje X obtenidos .....	71
Figura 40: Grafica de los puntos 3D del eje Y obtenidos .....	72
Figura 41: Grafica de los puntos 3D del eje Z obtenidos .....	72



## **1 Introducción**

### **1.1 Planteamiento del problema y justificación**

El seguimiento de objetos es una tarea que se requiere en diversos problemas de aplicación de la visión artificial (Análisis de flujo vehicular, análisis del movimiento corporal, comportamiento de grupos de personas, sistemas de video vigilancia, etc.). Aunque hay un conjunto de algoritmos, técnicas y operaciones básicas que se encuentran disponibles tanto en las herramientas de desarrollo especializadas en visión artificial y tratamiento de imágenes, como en la documentación disponible en la red, los algoritmos completamente funcionales no son de dominio público. No es fácil encontrar un algoritmo para el seguimiento y detección de trayectorias implementado para su rápida adaptación y uso. Los artículos de divulgación y artículos científicos en el mejor de los casos revelan los esquemas o diagramas de sus propuestas, pero no el código. Por lo tanto y dada la necesidad de dar solución a algunos requerimientos de grupos de investigación que se han acercado al Grupo de Investigación en Ciencias Computacionales – CICOM para la solución de problemas o implementación de sistemas que requieren de este tipo de algoritmos, el presente proyecto busca adaptar un conjunto de algoritmos genéricos y probados, que permitan realizar actividades de seguimiento de objetos en el espacio, la ubicación de los mismos y la generación de trayectorias de recorridos. Además, dichos algoritmos deben ser puestos en funcionamiento de tal forma que se permita la fácil configuración del sistema de cámaras utilizado y la escena bajo análisis.

## **1.2 Delimitación**

### **1.2.1 Objetivo General:**

- Desarrollar un sistema de visión artificial que permita realizar el seguimiento de objetos en el espacio tridimensional, determinando en cada instante de tiempo su ubicación espacial y trayectoria recorrida.

### **1.2.2 Objetivos Específicos**

- Analizar los diferentes algoritmos de seguimiento de objetos propuestos en la bibliografía disponible para seleccionar los más apropiados para el desarrollo del presente proyecto.
- Estudiar los modelos y algoritmos de visión estereoscópica para la determinación de la ubicación espacial de objetos.
- Adaptar los algoritmos que permitan la detección, ubicación y seguimiento de objetos en una escena 3D.
- Diseñar e implementar el sistema acoplado a una interfaz gráfica que permita configurar el conjunto de cámaras y el escenario sobre el cual se realizará el seguimiento.
- Evaluar el desempeño del sistema mediante la utilización de métricas para la predicción del error en el seguimiento.
- Crear un documento donde se especifique el porqué de la selección de algoritmos seleccionados de detección, ubicación y seguimiento de objetos.

## **1.3 Acotaciones**

- El sistema desarrollado será un prototipo funcional, que implemente los requerimientos descritos en los objetivos.
- Los algoritmos serán probados sobre objeto simples de fácil detección, por ejemplo, pelotas de pin-pong.
- Los algoritmos se probarán sobre escenarios controlados, con ciertas condiciones de iluminación y dimensiones fijas.

## **1.4 Metodología**

El proyecto consta de 7 fases, la primera fase de este proyecto es la revisión y selección de algoritmos de seguimiento. En la cual se estudian los diferentes algoritmos bases para dar comienzo a la selección del más adecuado. La segunda fase consiste en la revisión y selección de algoritmos para la visión estereoscópica y sincronización del sistema de cámaras. Se estudian las diferentes metodologías que utiliza la visión estereoscópica, se selecciona una de estas y se da comienzo a la tercera fase del proyecto que es la revisión y selección de algoritmos para visión estereoscópica y sincronización del sistema de cámaras. En esta fase se selecciona el algoritmo de visión estereoscópica más adecuado para dar cumplimiento con los requisitos descritos en los objetivos. La cuarta fase se basa en la realización de pruebas y refinamiento de algoritmos. En la cual se hacen diferentes pruebas las cuales sirven como guía para modificar algunos errores de los códigos seleccionados, los cuales serán parte del proyecto. La quinta fase de este proyecto es el diseño de interfaz gráfica e implementación de prototipo completo. En esta fase se acoplan los algoritmos de seguimiento y visión estereoscópica, se crea un GUI, en el cual será embebido el código para facilitar su uso. La sexta fase del proyecto es el desarrollo de pruebas y evaluación de resultados. En esta fase se desarrollan diferentes pruebas en un ambiente controlado para verificar la fiabilidad del seguimiento de uno o varios objetos y a la distancia en la que se encuentra este. Por último, la séptima y última fase consiste en la creación de la documentación del proyecto.

### **1.4.1 Tipo de investigación**

Para este proyecto utilizaremos la metodología cuantitativa, según el teórico (“El Proceso de Investigación, Carlos Sabino,” n.d.) nos dice que la metodología cuantitativa es: “En cuanto a los elementos que es necesario operacional izar pueden dividirse en dos grandes campos que requieren un tratamiento diferenciado por su propia naturaleza: el universo y las variables” ya que nos permita analizar y medir de forma clara y sencilla las fases de cada actividad propuesta, tomando muestras de los resultados obtenidos después de cada prueba, se obtendrá archivos de estas actividades los cuales servirán para realizar y soportar nuestro documento final.

## 2 Marco teórico y estado del arte

### 2.1 Marco conceptual

- **Visión artificial**

La visión artificial engloba a cualquier proceso óptico mediante el cual un sistema inteligente es capaz de extraer información de un entorno para su interpretación mediante el uso de la computadora. En un principio los sistemas de visión artificial estaban estrechamente basados en la visión humana, pero debido a la falta de entendimiento de los procesos que tienen lugar en el cerebro a la hora de interpretar los datos del sistema visual, estos sistemas artificiales resultaron básicamente imprácticos. El cerebro humano, en combinación con el sistema visual del que dispone nuestro cuerpo, permite un reconocimiento y una interpretación de su entorno mucho más flexible y adaptable a cambios que cualquier sistema de visión artificial. Sin embargo, los sistemas de visión artificial son capaces de procesar cantidades de datos mucho mayores y en menos tiempo si se trata de tareas repetitivas. La infalibilidad de la precisión matemática permite un estudio y análisis mucho más detallado y extenso por parte de un sistema basado en la visión artificial. Esta característica permite la creación de sistemas muy diferentes en cuanto a componentes y capacidades, según la tarea a implementar.(Ochoa et al. 2015). El propósito de la visión artificial es programar un computador para que "entienda" una escena o las características, también conocida como Visión por Computador. Las técnicas y métodos a utilizar en la solución del problema son diversas y varían dependiendo de la naturaleza del problema a tratar, por ejemplo, utilizando técnicas de color, formas, texturas, etc.(Guerrero Hernández, Pajares Martinsanz, and Guijarro Mata-García, n.d.). Algunos campos en el cual la visión artificial es aplicable se mostrarán a continuación:

**Astronomía:** Se puede hacer observaciones del cosmos (estrellas, explosiones, a planetas, superficies, etc.) utilizando imágenes telescópicas o satelitales (Cuervo, Andrés, and Salazar 2013).

**Biología:** Se puede identificar determinadas características en las especies vegetales o animales de acuerdo a la textura, color, tamaño, forma, etc. (Du, Wang, and Zhang 2007).

**Inspección y control de calidad:** Los procesos industriales requieren la automatización para verificar que los productos manufacturados, frutas, vegetales cumplan con ciertos criterios de calidad previamente establecidos, evitando los defectos de producción, impurezas no deseadas, mala apariencia de los productos (Fuente López, Trespaderne, and Universidad de Valladolid 2012).

**Geología:** Se puede detectar movimientos de terrenos (por deslaves por ejemplo), formaciones y mediciones geológicas, etc.(Revollo Sarmiento et al., n.d.).

**Medicina:** Se puede trabajar con las imágenes provenientes de ecografías, radiografías, mamografías, resonancia magnética, tomografía, etc. para el diagnóstico de ciertas patologías (Sebastian Barberis Maestrando Juan Pablo Graffigna Director Roberto Isoardi Co-Director, n.d.) .

**Meteorología:** Se emplea para el monitoreo y prevención de fenómenos meteorológicos y las condiciones ambientales (como lluvia, viento, nieve, huracán, tornado, tormenta, inundaciones, sequías, heladas) del planeta a través de imágenes satelitales (Avendaño Pérez et al., n.d.).

**Microscopía:** Utilizado principalmente para la observación celular y microorganismos; identificar, contar y reconocer objetos de una determinada muestra utilizando imágenes microscópicas.

**Modelado y visualización en 3D:** Se puede extraer la estructura geométrica y semántica de imágenes para reconstruir y crear automáticamente modelos 3D y sistemas de visualización interactiva (Snavely et al. 2010).

**Reconocimiento y clasificación:** Se puede reconocer objetos inmersos en las imágenes y su posterior clasificación, como por ejemplo el reconocimiento de rostros y la clasificación por edades, identificación de placas de vehículos, etc.

**Robótica:** Un sistema de visión permite la navegación o el guiado automático de máquinas (Jiang, Zhao, and Si 2010).

**Seguridad:** A través de secuencias de imágenes se puede realizar vigilancia para detectar la presencia y movimiento de cuerpos extraños, reconocimiento dactilar y ocular.

**Teledetección:** Mediante el uso de imágenes aéreas o satelitales se puede detectar cambios en una zona de terreno en diferentes lapsos como, por ejemplo, detección de zonas erosionadas, deforestadas, incendios, inundaciones, presencia de edificaciones o infraestructuras como carreteras y puentes (J. D. Falcón and Ambrosio 2004).

**Agricultura:** Se puede implementar soluciones tecnológicas en el agro para minimizar la aplicación de químicos e incrementar la producción. Por ejemplo: detección y control automática de maleza y plagas, identificación de tipos de plantas, etc.

- **Técnicas más representativas**

**Umbralización:** También conocida como thresholding consiste en eliminar los valores superiores o inferiores (poniendo estos en cero). (EDMANS 2006).

**Binarización:** Es una variante de la umbralización y consiste en dejar a cero todos los pixeles menores de un umbral y a uno a aquellos que son iguales o mayores, quedando construida la imagen final.(EDMANS 2006).

**Suavizado:** Consiste en sustituir la media aritmética de los puntos que tiene alrededor (incluido el mismo). Donde los resultados que se producen son equivalentes a una difuminación de la imagen produciendo un suavizado de los gradientes en toda la imagen. (EDMANS 2006).

**Realzado:** Se usan para compensar imperfecciones de la imagen. Imperfecciones como partículas de polvo, ruido, cuadros entrelazados (generalmente, provenientes de capturas de una televisión) y falta de nitidez.(EDMANS 2006).

**Detección de puntos de interés a través de regiones:** Las Regiones de Interés son ampliamente usadas en el procesamiento digital de imágenes para la detección de anomalías u otros tipos de procesamiento. Las anomalías son pequeños objetos que son diferentes de la gran mayoría.(Rodríguez Guillén 2014). El reconocimiento de regiones se refiere a las técnicas cuyo objetivo es detectar puntos o regiones más claras u oscuras de la imagen. Hay dos clases principales de detectores de regiones, métodos diferenciales y métodos basados en extremos locales.

**Detección del contorno:** La detección de contorno es parte de un proceso de aislamiento (segmentación), que consiste en la identificación de objetos dentro de una imagen. Como es usual, hay varias posibles definiciones de un contorno, siendo cada una aplicable en distintas circunstancias. (toronto, n.d.).

**Reconocimiento de objetos:** El reconocimiento de objetos sirve para buscar y reconocer patrones aprendidos en la imagen. Esta funcionalidad puede utilizarse por separado o combinada con todas las demás funciones mencionadas, por lo que tiene diversos campos de aplicación.(SIMENS, n.d.).

- Pick&Place.
- Control de calidad en la fabricación.
- Control de posición en sistemas de alimentación.



- **Seguimiento de objetos (Tracking).**

Es un término inglés, traducible como “seguimiento”, que se utiliza para hablar de todas aquellas acciones con las que se siguen y se miden índices utilizados para cuantificar el comportamiento de determinadas variables en distintos momentos del tiempo. Es un concepto muy ligado al marketing, sobre todo al digital. (NeoAttack, n.d.).

- **Visión estereoscópica.**

Es aquella que integra dos imágenes por medio del cerebro, el ser humano es capaz de integrar esas dos imágenes en una sola, y crear una imagen tridimensional. Se produce cuando cada uno de los ojos acomoda el cristalino para enfocar correctamente un objeto, así ambos ejes ópticos convergen sobre el objeto que se mira. Para este proceso se utiliza la visión binocular y la estereoscopia (CEFOCCA-UNSJ 2016).

La tecnología de Visión por Computador proporciona al sistema, las sensaciones realistas del mundo. Cuyo objetivo es la sensación tridimensional que se consigue mediante la obtención de imágenes estereoscópicas generadas por dos cámaras, las cuales adquieren dos imágenes distintas de la misma escena. La visión estereoscópica constituye un procedimiento para la obtención de la forma de los objetos en la escena. En este caso la forma se determina a través de la distancia de los objetos con relación a un sistema de referencia. Por lo cual se trata de un método para la obtención de la tercera dimensión. Para la obtención de la estructura de la escena, existen dos tipos de métodos: activos y pasivos.

Los métodos pasivos son aquellos que intervienen externamente sobre la escena, bien iluminándola o bien enviando un haz energético, tales como sensores de ultrasonidos, luz estructurada, triangulación, telémetro de tiempo de vuelo, etc. Por el contrario, los pasivos no actúan sobre la escena, siendo la visión estereoscópica, el principal método que constituye el objetivo final de estudio en este trabajo. (Guerrero Hernández, Pajares Martinsanz, and Guijarro Mata-García, n.d.).

- **Kernel**

Un kernel o máscara de convolución puede entenderse como una matriz de coeficientes que al ser aplicada en un pixel objetivo (considérese el pixel como un punto  $(x, y)$  del plano) obteniendo una transformación en el pixel objetivo como en sus vecinos (entendiendo la transformación como la imagen  $G(x,y)$  en todos ellos). Si bien la forma y tamaño del kernel es variable, usualmente se trabaja con máscaras de convolución cuadradas.

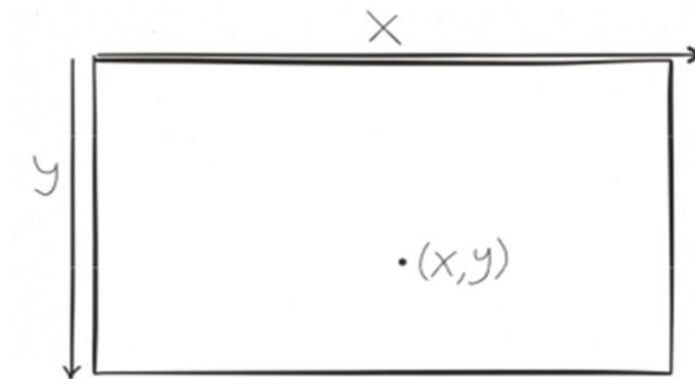
- **silueta**

Es el proceso de separar o enmascarar una parte de una imagen (como el fondo) para que no se muestre.

- **Punto**

Una imagen se representa como una matriz con valores que son interpretados por las maquinas convirtiendo esos valores en colores que pueden ser observado por una persona, por ende, los valores dentro de la matriz son puntos que componen a la imagen. Cada punto tiene características que permiten identificarse unos de otros, estas características son.

- **Posición:** Es la ubicación de un punto en la imagen o frame respecto al origen de la misma. En un plano bidimensional la posición se representa con las coordenadas  $x, y$  como en la **Figura 1**.



*Figura 1: Representación de un punto en una imagen*

- **Intensidad:** Corresponde al valor de la información almacenada por un punto, generalmente se puede entender de una forma numérica, en las imágenes que se encuentran en escalas de grises este valor se puede entender como intensidad. En la **Figura 2** se da un punto de interés (POI) el cual se ubica en el centro de la imagen.

100	100	0
80	POI 80	150
0	150	150

Figura 2: Intensidad de una imagen

**Fuente:**(Leonardo and Fajardo 2015a)

- **Gradiente:** Esta depende de la comparación con otros puntos. Este indica la pendiente entre valores que tiene un punto y otro, esto con el propósito de determinar los cambios más significativos que definen los bordes, límites u otros puntos que pueden ser de interés en una imagen, un ejemplo de este se puede ver en la **Figura 3**.

100	100	0
80	20 20 80	150
0	80 70 70	150

Figura 3: Representación de un gradiente

**Fuente:**(Leonardo and Fajardo 2015a)

- **Laplaciano:** Esta característica depende de los puntos adyacentes que se están analizando, en este caso el laplaciano indica una rápida variación en los valores de los puntos lo que indica la presencia de puntos de interés como bordes o límites.
- **Curvatura:** Se define como la velocidad de cambio de dirección entre puntos de una función, determinando si esta función es cóncava o convexa

## 2.2 Análisis de video

El análisis de un video se puede representar como el análisis y procesamiento de imágenes, ya que un video no es más que una secuencia de imágenes que se denominan frames, que van lo suficientemente rápido para que el ojo humano pueda percibir la continuidad del contenido, este forma parte de la visión por computador en donde el objetivo principal es obtener un mayor entendimiento sobre lo que se está observando en cada imagen.

El primer paso para el análisis de un video es capturar un frame y detectar las regiones de interés para posteriormente hacer un análisis sobre ella. En la región de interés estará el objeto el cual se clasifica para su posterior procesamiento, es por ello por lo que se analiza el frame en su totalidad a fin de detectar el objeto de interés. Una vez que se detecta el objeto el próximo paso es la clasificación de este para determinar si se debe realizar el seguimiento de este.

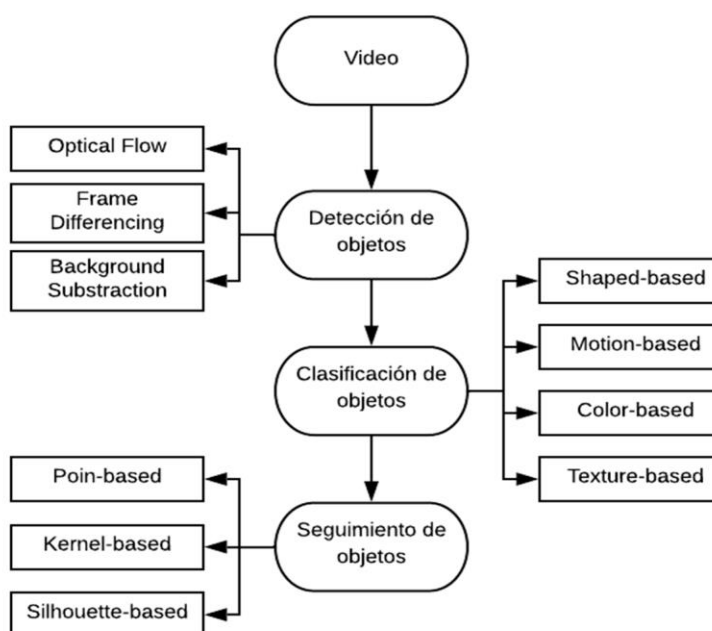
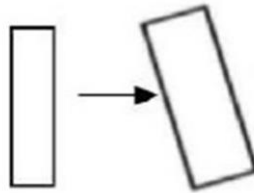


Figura 4: Esquema representativo para el análisis de un video)

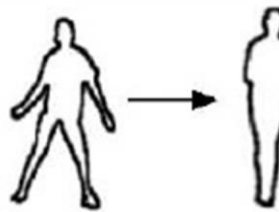
### 2.2.1 Seguimiento de objetos

**Método kernel:** Este abarca todos los métodos en donde el movimiento del objeto es representado por una región embriónica la cual se computa de un frame al siguiente, como se muestra en la **Figura 5**.



*Figura 5: Representación del método-kernel*

**Método silueta:** Este método está enfocado en hacer el seguimiento de aquellos objetos cuya forma geométrica es compleja, como pueden ser manos, dedos, entre otras, los cuales no pueden ser definidos por una forma geométrica simple. Un ejemplo de este método se muestra en la **Figura 6**.



*Figura 6: Representación del método-silueta*

**Fuente:**(Guerrero Hernández, Pajares Martinsanz, and Guijarro Mata-García, n.d.)

**Método punto:** Representa a los objetos en movimiento con un conjunto de puntos característicos y actualiza su estado en el tiempo, como se muestra en la **Figura 7**

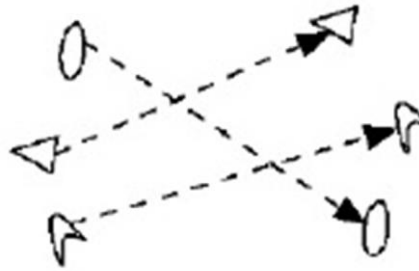


Figura 7: Representación del método punto

**Autor:**(Guerrero Hernández, Pajares Martinsanz, and Guijarro Mata-García, n.d.)

### 2.2.2 Clasificación de objetos

**Basado en textura:** Esta es una característica que consiste en transformar la textura de un objeto en la forma en la cual se utilizara para diferenciarlo de su entorno y de otros objetos presentes en una imagen, Este método presenta un grado alto de dificultad cuando existe más de un objeto con la misma textura. Técnicamente se representa como la intensidad que varía en la superficie que se rastrea y por lo general se analiza la suavidad y regularidad de aparición. Algunos tipos de texturas se presentan en la **Figura 8**.

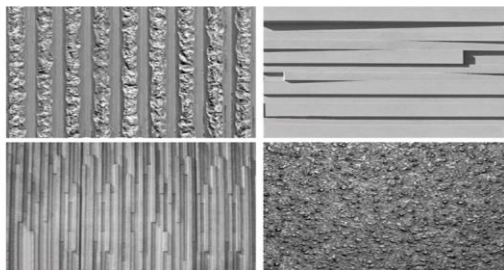


Figura 8: cuadro de diferentes texturas

**Fuente:**(Pastorelli 2011)

**Basados en color:** El método para clasificar un objeto por el color, es el método más fácil de implementar y comprender , pero también es el más difícil de utilizar ya que presenta graves

problemas con la luminosidad o propiedad reflectante de un objeto, otro problema que se puede presentar es la situación en la cual existen dos objetos con colores iguales, la cual causa una confusión al momento del seguimiento, en otras ocasiones se pueden sobreponer dos objetos del mismo color lo cual dificulta el seguimiento del objeto que se sigue. Las imágenes utilizan diferentes formatos de color, pero uno de los más comunes es el formato RGB.



*Figura 9:  
Reconocimiento de objetos a  
través del Color*

**Basado en forma:** Este método se basa en la búsqueda de la silueta del objeto y su modelo asociado dentro de la imagen actual.



*Figura 10: Reconocimiento de objetos a través de formas*

**Basado en movimiento:** Este tipo de técnicas principalmente utiliza la información del movimiento. Dentro de este tipo de seguimiento se encuentran dos métodos comúnmente utilizados, los cuales se presentan a continuación.

**Discontinuidad de flujo óptico:** Esta técnica se basa en el desplazamiento de un píxel, representándolo en un vector de movimiento entre el píxel en una imagen y el píxel correspondiente de la otra.

**Detección de cambios:** Se basa en la detección de los píxeles de un objeto y el fondo de este. Donde se asume que el fondo es estacionario y que los cambios entre imágenes consecutivas son debido al movimiento.

## 2.3 Características de los algoritmos de seguimiento

### 2.3.1 Región de interés

Al igual que un punto puede ser objeto de seguimiento en una serie de imágenes es importante conocer que se puede lograr también con una región, es decir obtener una subárea de la imagen permite conocer mayor información relevante, el propósito de esta depende de la aplicación, en este caso se utilizara para hacer seguimiento de un objeto. Para hacer el seguimiento de un objeto a través de regiones es necesario que el objeto tenga una característica que permita encontrarlo en la secuencia de imágenes (Augusto Betancur Cruz 2016). Las características que se tienen en cuenta son:

**Repetitividad:** Esta característica se refiere a que un objeto puede encontrarse en dos o más imágenes bajo condiciones diferentes. Un ejemplo de esta característica se puede observar en la figura 11

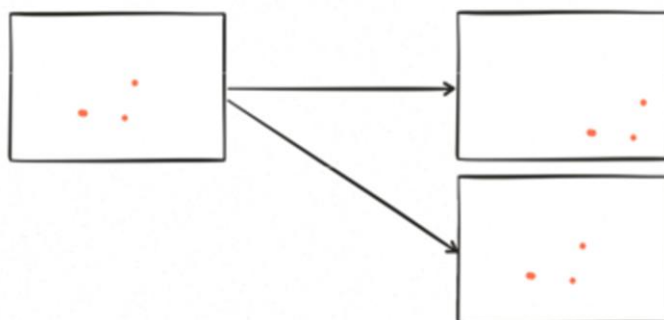


Figura 11: Repetitividad y Distinción de una imagen

**Distinción:** La región seleccionada debe tener la información suficiente para que pueda ser distinguible de forma única. En este tipo de casos se suele utilizar el apoyo de puntos de interés, o el reconocimiento de objetos. En la **Figura 11** se nota que existe una diferencia entre la información que se dio y la posición de los puntos respecto al origen, pero no entre ellos mismos.

**Localidad:** La información se encuentra contenida en una imagen lo cual reduce la probabilidad de oclusión.



**Cantidad:** Las referencias tomadas y existentes dentro de la región de interés deben ser suficientes para que al hacer una comparación entre dos regiones candidatas estas sean identificables.

**Precisión:** La información que se está detectando debe estar precisamente localizada.

El uso de regiones de interés es una gran solución a los problemas que se da en los puntos de interés ya que estos no cuentan con una gran cantidad de información como lo hace esta, al tener más información permite que disminuya la probabilidad de errores de identificación del objeto, pero este también propone diferentes problemáticas como lo son el barrido de la región de interés en una imagen diferente, desaparición, rotación etc.

### 2.3.2 Estimación del movimiento.

Generalmente, la mayoría de trackers asumen que el objetivo se va a encontrar en un área próxima a su localización previa. Por ello, la búsqueda se realiza en torno al área espacial que ocupa el objetivo en el cuadro anterior siendo todas las direcciones de desplazamiento igual de probables. Este método ofrece en general buenos resultados ya que permite adaptarse a cambios de dirección del movimiento del objeto, aunque resulta insuficiente cuando el movimiento es muy rápido. Para solventar esta limitación se usan otras técnicas en las que la búsqueda se centra en zonas estimadas mediante predicción en base al movimiento previo del objeto (Augusto Betancur Cruz 2016).

### 2.3.3 Modelo de actualización

Mediante la actualización de la apariencia, la búsqueda del objetivo en el nuevo cuadro se realiza en base a las características del modelo de apariencia en su última posición detectada. Esto permite que la apariencia del objeto evolucione a lo largo de la secuencia. Sin embargo, algunos algoritmos no realizan esta actualización de tal manera que se mantiene el mismo modelo para la búsqueda durante toda la secuencia. Otras alternativas utilizan un sistema de actualización mediante predicción de la apariencia pensados para secuencias largas (Augusto Betancur Cruz 2016).

## 2.4 Visión estereoscópica

La visión estereoscópica es el conjunto de técnicas que intentan recuperar información tridimensional a partir de dos o más vistas de una escena. (Fernández López 2016). Existen cuatro pasos fundamentales para poner en práctica la visión estereoscópica (Uribe Sáenz de Cámara, n.d.):

- La calibración de las cámaras. Los parámetros intrínsecos e intrínsecos involucrados en la geometría estereoscópica.

- Rectificación. Geometría epipolar para simplificar todo lo que sea posible la búsqueda que se realiza para resolver el problema de la correspondencia. aprendizaje profundo.
- Correspondencia entre puntos de ambas imágenes, que proporcionan un mapa de disparidades.
- Reconstrucción de la escena 3D. El cálculo de las profundidades a partir de las disparidades.

#### 2.4.1 Modelo de cámara:

Las cámaras capturan los rayos de luz de nuestro entorno. En principio, una cámara funciona como nuestro ojo, los rayos reflejados de nuestro entorno vienen a nuestro ojo y se recogen en nuestra retina. La "cámara estenopeica" es el modelo más simple (Freeman and Seitz, n.d.). Es un buen modelo simplificado para comprender cómo funciona una cámara. En el modelo, todos los rayos de luz son de una superficie detenida. Solo detectará los rayos que penetran a través del orificio y siguen un área de la cámara que está configurada al revés (Uribe Sáenz de Cámara, n.d.). La **Figura 12** explica este principio.

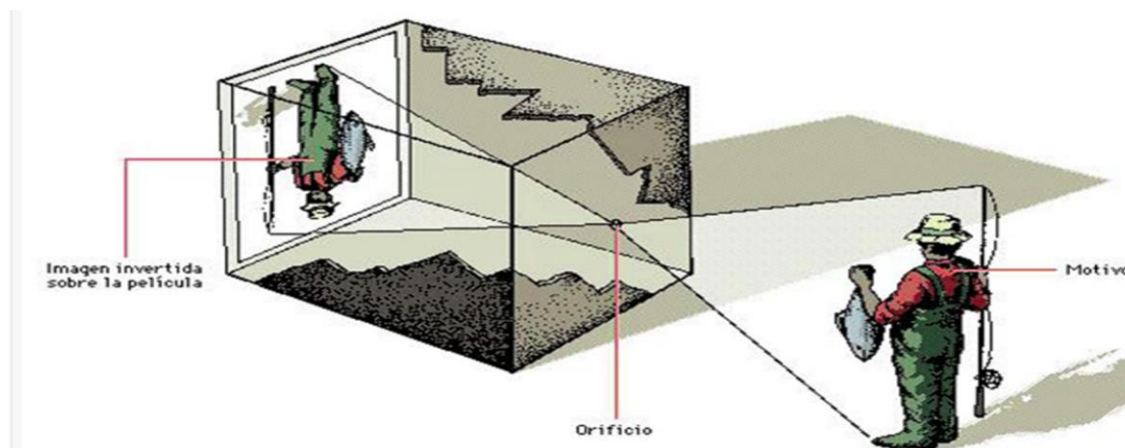


Figura 12: Funcionamiento de una cámara. Modelo pinHole

Este principio es muy simple, pero no es una buena forma de obtener suficiente luz para capturar una exposición rápida. Es por eso que las lentes se usan para recoger rayos de luz en un solo lugar. El problema de esta lente es que tiene dos tipos diferentes de distorsión:

- La distorsión radial.
- La distorsión tangencial.

La distorsión radial proviene de la forma de la lente y de la distorsión tangencial proviene de la geometría de la cámara. Las imágenes pueden analizarse luego usando matemática para ser corregido. (Julián Aguirre de Mata, n.d.). El proceso de calibración permite un modelo de la

geometría de la cámara y un modelo de distorsión de la forma de la lente. Estos modelos forman los parámetros intrínsecos de una cámara.

### 2.4.2 Longitud focal

El tamaño relativo de la imagen que se proyecta sobre la superficie de la cámara depende de la longitud focal. En el modelo de agujero de alfiler es la longitud focal, la distancia entre el orificio y el área donde se proyecta la imagen.

El teorema de Tales da entonces:

$$-x = F * \left(\frac{X}{Z}\right)$$

Con:

- x: imagen del objeto (el signo menos proviene de invertir la imagen).
- X: tamaño del objeto.
- Z: distancia del agujero al objeto.
- F: distancia focal, distancia desde el agujero a la imagen.

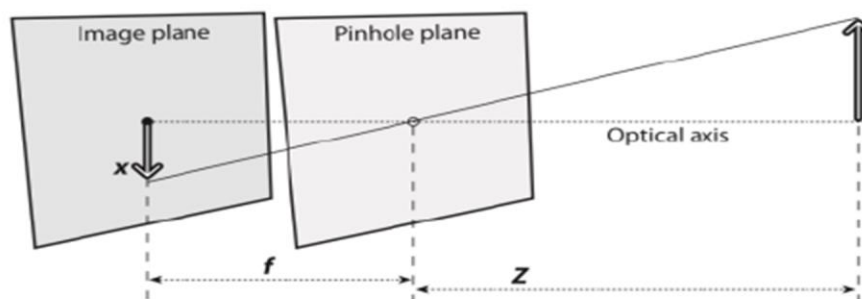


Figura 13: Como se proyecta un objeto. Modelo pinhole

**Fuente:**(Gary, bandski , adrian 2008)

Como la lente no está perfectamente centrada, se introducen dos parámetros,  $C_x$  y  $C_y$  para cada desplazamiento horizontal y vertical de la lente. La distancia focal en el eje X y en el eje Y también es diferente porque el área de la imagen es rectangular. Esto conlleva a obtener las siguientes fórmulas para la posición del objeto en la superficie.(Gary, bandski , adrian 2008)

$$x_{screen} = f_x\left(\frac{X}{Z}\right) + c_x$$

$$y_{screen} = f_y\left(\frac{Y}{Z}\right) + c_y$$

Los puntos proyectados del mundo real en la escena pueden explicarse de la siguiente manera donde M es la matriz intrínseca.

$$q = MQ$$

Dónde:

$$q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

### 2.4.3 Distorsión de la lente

Teóricamente, es posible construir una lente que no cause distorsión con una sola lente parabólica, sin embargo, es mucho más fácil hacer una lente esférica. Hay dos tipos de distorsión. La distorsión radial proveniente de la forma de la lente y la distorsión tangencial proveniente de la lente en el proceso de montaje de la cámara. (Torrontegi and Azpiazu n.d.)

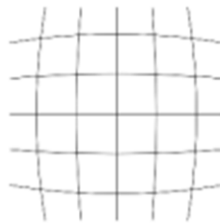


Figura 14: Distorsión radial de las cámaras

Fuente: wiki pedía

No hay distorsión radial en el centro óptico, pero este se hace más grande al acercarse desde los bordes, permitiendo una distorsión pequeña, hacer un desarrollo de Taylor es suficiente para corregir la distorsión. El resultado es la siguiente fórmula.

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

X e Y son las coordenadas del punto original en el área de la imagen y. También hay una distorsión tangencial porque la lente no es perfecta está construido en paralelo con la superficie de la imagen. Para corregir esto se introducen dos parámetros adicionales,  $p_1$  y  $p_2$ .

$$x_{corrected} = x + [2p_1 y + p_2(r^2 + 2x^2)]$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 x]$$

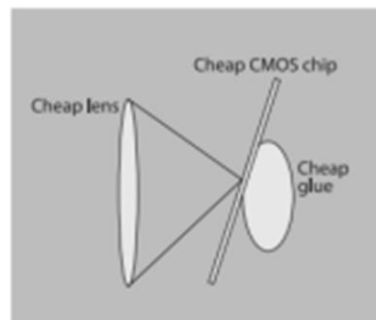
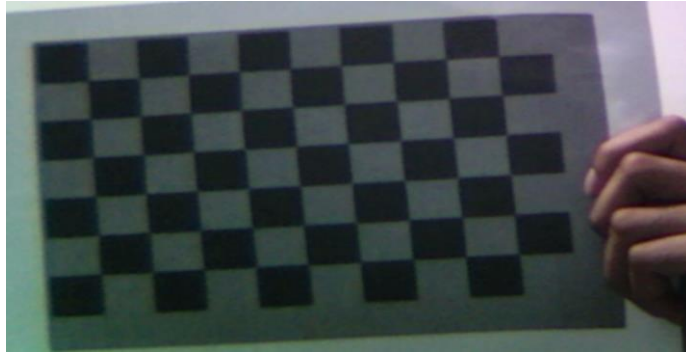


Figura 15: Distorsión tangencial de las cámaras.

Fuente: (Gary, bandski , adrian 2008)

#### 2.4.4 Calibración con OpenCV

La biblioteca OpenCV nos permite establecer los parámetros intrínsecos basados en parámetros específicos. Para calcular funciones, este proceso se llama calibración. Esto se podrá realizar con diferentes dimensionamientos de un tablero de ajedrez.



*Figura 16: Tablero de ajedrez para calibración de cámaras*

#### **2.4.5 Visión estéreo**

La visión estéreo permite detectar la profundidad en una imagen, tomando medidas en esta, permitiendo hacer localizaciones en 3D. Esta debe incluir puntos que coincidan entre ambas cámaras. Ya que al obtener la distancia entre la cámara y el punto al manejar la geometría se puede simplificar el cálculo.

La visión estéreo realiza estos cuatro pasos:

- Eliminación de la distorsión radial y tangencial por cálculos matemáticos. Obtienes imágenes sin distorsiones.
- Rectificar el ángulo y el espaciado de las imágenes. Esta etapa permite tantas imágenes coplanarias en el eje Y, de modo que aliviana la búsqueda de correspondencias ya que solo necesita buscar en un solo eje (en el eje X).
- Encuentre la misma característica en las imágenes derecha e izquierda. Esto te da uno mapa de disparidad que muestra las diferencias entre las imágenes en el eje x.
- El último paso es la triangulación. Transforme el mapa de disparidad en distancias por triangulación.

#### **2.4.6 Triangulación**

En el paso final, la triangulación supone que ambas imágenes de proyección Coplanar son la fila de píxeles horizontales de la imagen izquierda con el correspondiente de la fila horizontal de píxeles de la imagen derecha alineada.

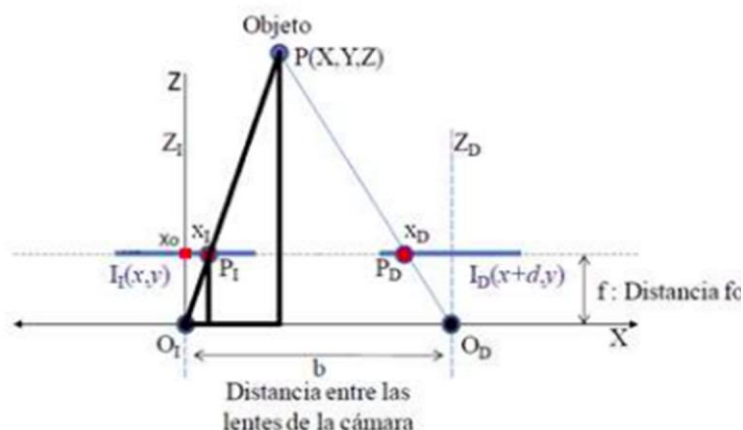


Figura 17: Triangulación de cámaras hacia un objeto

El punto P está en el entorno y estará en la imagen izquierda y derecha mostrada en  $P_L$  y  $P_D$ , con las coordenadas correspondientes  $X_L$  y  $X_D$ . Este nos permite introducir un nuevo tamaño que disparidad:

$$d = X_L - X_D$$

Cuanto más se disminuye el punto P, se puede ver cuanto menor es el tamaño d. La disparidad es, por lo tanto, inversamente proporcional a distancia.

Con la siguiente fórmula se puede calcular la distancia:

Se puede ver que existe una relación no lineal entre la disparidad y la distancia. Si la disparidad es cercana a 0, las pequeñas diferencias de disparidad serán demasiado grandes. Esto es viceversa si la disparidad es grande las diferencias de disparidad no conducen a grandes diferencias en la distancia. A partir de ese se puede concluir que la visión estéreo tiene una resolución de alta profundidad, solo para objetos cerca de la cámara.

Este método solo funciona si la configuración de la cámara estéreo es ideal. Sin embargo, esto no es el caso en realidad. Es por eso que la imagen izquierda y derecha serán alineadas matemáticamente en paralelo. Por supuesto, las cámaras tienen que ser al menos aproximadas físicamente.

Antes de explicar el método de alineación matemática de las imágenes, tenemos que entender primero la geometría epipolar.

### 2.4.7 Geometría Epipolar

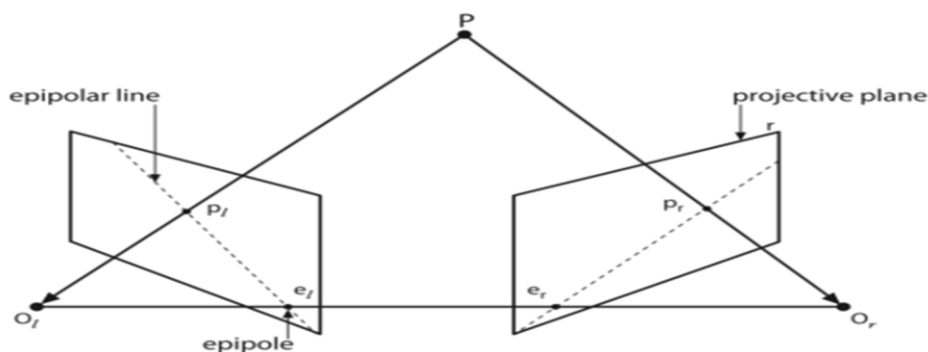


Figura 18: Triangulación de dos cámaras

La imagen de arriba nos muestra el modelo de una cámara estéreo no perfecta. Hay dos modelos de cámara estenopeica. Al cruzar la línea de centros de ingeniería ( $O_l$ ,  $O_r$ ) y los puntos epipolares ( $e_l$  y  $e_r$ ) que surgen con los planos de activación. Las líneas ( $p_l, e_l$ ) y ( $p_r, e_r$ ) se llaman líneas epipolares. (Unrweiller, Frederic, vujasinovic 2017). La imagen de todos los puntos posibles de un punto. En un nivel de proyección, la línea epipolar está en el otro plano de imagen y pasa por el punto epipolar y el punto buscado. Esto permite la búsqueda de una sola dimensión en lugar de a un plano completo.

Para que pueda resumir los siguientes puntos:

- Cada punto 3D en la vista de una cámara está incluido en el plan epipolar.
- Una característica en un plano debe estar en las líneas epipolares apropiadas de otro nivel (condición epipolar).
- Una búsqueda bidimensional de una característica correspondiente se convierte en una búsqueda unidimensional convertida, si se conoce la geometría epipolar.
- Se mantiene el orden de los puntos, es decir que dos puntos A y B en el mismo orden se puede encontrar en las líneas epipolares de un plano, como en el otro nivel.

### 2.4.8 Matrices Esenciales y Fundamentales

Para comprender cómo se calculan las líneas epipolares, primero se debe considerar lo esencial de las matrices fundamentales (correspondientes a las matrices E y F). La matriz esencial E contiene la información, como las dos cámaras juntas están dispuestas físicamente describe la ubicación de la segunda cámara en relación con la primera cámara a través de parámetros de traslación y rotación. Estos parámetros están en la Matriz, no se puede leer directamente porque se usa para la configuración. La matriz F contiene la información de la



matriz esencial E, para la disposición física de las cámaras y la información sobre los parámetros intrínsecos de las cámaras.(Unrweiller, Frederic, vujasinovic 2017).

La relación entre el punto proyectado en la imagen izquierda ( $p_l$  y  $e_l$ ) de la derecha. Se define como la imagen  $p_r$ .

$$p_r^T E p_l = 0$$

Se podría pensar que esta fórmula es el vínculo entre la izquierda y la izquierda. Pero esta fórmula es la ecuación de una línea recta que define completamente la relación entre los puntos, uno debe por lo tanto tener en cuenta los parámetros intrínsecos.

Recordando que  $q = M p$ , con la matriz intrínseca M. Al reemplazar en la ecuación anterior, obtenemos:

$$p_r^T (M_l^{-1})^T E M_l^{-1} q_l = 0$$

Se reemplaza por:

$$F = (M_l^{-1})^T E M_l^{-1}$$

Y así obtener:

$$p_r^T F q_l = 0$$

## 2.5 Rectificación Estéreo

Hasta ahora, hemos tratado el tema de la "calibración estéreo". Se trataba de la descripción de la disposición geométrica de ambas cámaras. La tarea de la rectificación consiste en proyectar las dos imágenes para que se encuentren exactamente en el mismo plano y alinear con precisión las filas de píxeles para que las líneas epipolares se vuelvan horizontales. Para encontrar la correspondencia de un punto más al azar en las dos imágenes.(Parra Escamilla 2012). Como resultado del proceso de alinear ambas imágenes, uno obtiene 8 términos, 4 por cada cámara:

- Un vector de distorsión
- Una matriz de rotación R rectificada, que debe aplicarse a la imagen.
- Una cámara rectificada matriz M rectificada.
- Una matriz de cámara no rectificada M.

OpenCV nos permite calcular estos términos con dos algoritmos: el Hartley algoritmo y el algoritmo de Bouguet.

### 2.5.1 Algoritmo Hartley

El algoritmo Hartley busca los mismos puntos en las dos imágenes. Él intenta minimizar las disparidades y encontrar las homografías epipolares. Este método no necesita los parámetros intrínsecos.

Una ventaja de este método es que la calibración solo es posible gracias a la observación de puntos en la escena. Una gran desventaja que tiene es que no escala la imagen, solo tiene la información de la distancia relativa. No puede medir exactamente un objeto lejos de las cámaras.

### 2.5.2 Algoritmo de Bouguet

El algoritmo de Bouguet utiliza la matriz de rotación calculada y el vector de traducción al girar ambos niveles proyectados media vuelta para que estén en el mismo nivel. Esto hace que los rayos principales sean paralelos y los planos coplanarios, pero aún no alineado en filas. En este proyecto se utiliza el algoritmo Bouguet

## 2.6 Métricas estándar más utilizadas en modelos de regresión.

- **Root Mean Squared Error**

### Sintaxis RMSE(X, F):

X es el resultado eventual de datos de las series de tiempo.

F es el pronóstico de datos de series de tiempo.

El error cuadrático medio (RMSE) es la desviación estándar de los residuos (errores de predicción). Los residuos son una medida de qué tan lejos están los puntos de datos de la línea de regresión; RMSE es una medida de la dispersión de estos residuos. En otras palabras, le dice qué tan concentrados están los datos alrededor de la línea de mejor ajuste. El error cuadrático medio se usa comúnmente en climatología, predicción y análisis de regresión para verificar resultados experimentales(Anthony G, n.d.).

La fórmula sería la siguiente:

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^n (A_t - F_t)^2}{n}}$$

Dónde:

$A_t$  son las observaciones

$F_t$  pronostica los valores de una variable.

$n$  el número de observaciones disponibles para el análisis.

- **Mean Absolute Error**

**Sintaxis MAE(X, Y)**

X son los datos originales de la muestra de series de tiempo.

Y son los datos de las series de tiempo pronosticadas.

El error absoluto medio es una medida de evaluación del modelo utilizado con los modelos de regresión. (Sammut and Webb 2010) El error absoluto medio de un modelo con respecto a un conjunto de prueba es la media de los valores absolutos de los errores de predicción individuales en todas las instancias del conjunto de prueba. Cada error de predicción es la diferencia entre el valor verdadero y el valor predicho para la instancia (Ait-Amir, Pougnet, and El Hami 2015).

La fórmula sería la siguiente:

$$\text{MAE} = \frac{\sum_{t=1}^n |A_t - F_t|}{n}$$

**Dónde:**

$A_t$  Son las observaciones actuales de las series de tiempo

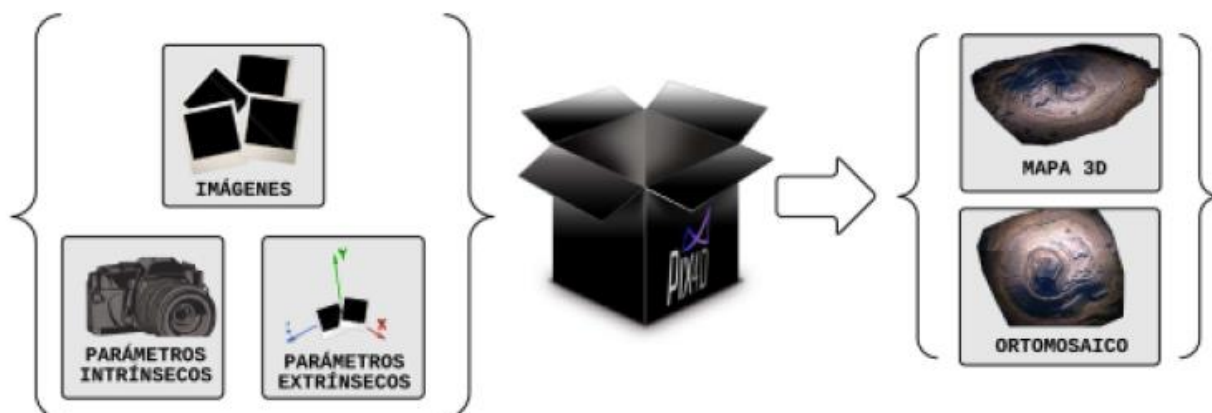
$F_t$  Es la serie de tiempo estimada o pronosticada

$n$  Es el número de puntos de datos no faltantes

## 2.7 Modelo Visión estéreo

- **Modelo de caja negra**

El modelo de caja negra o también conocido por esquema pipeline que lleva a la generación de mapas 3d son esquemas capaces de representar a cualquier sistema con estructuras parametrizadas. Es decir que obtenemos unas salidas a partir de unas entradas dadas. En la Figura 4 se presenta este modelo.



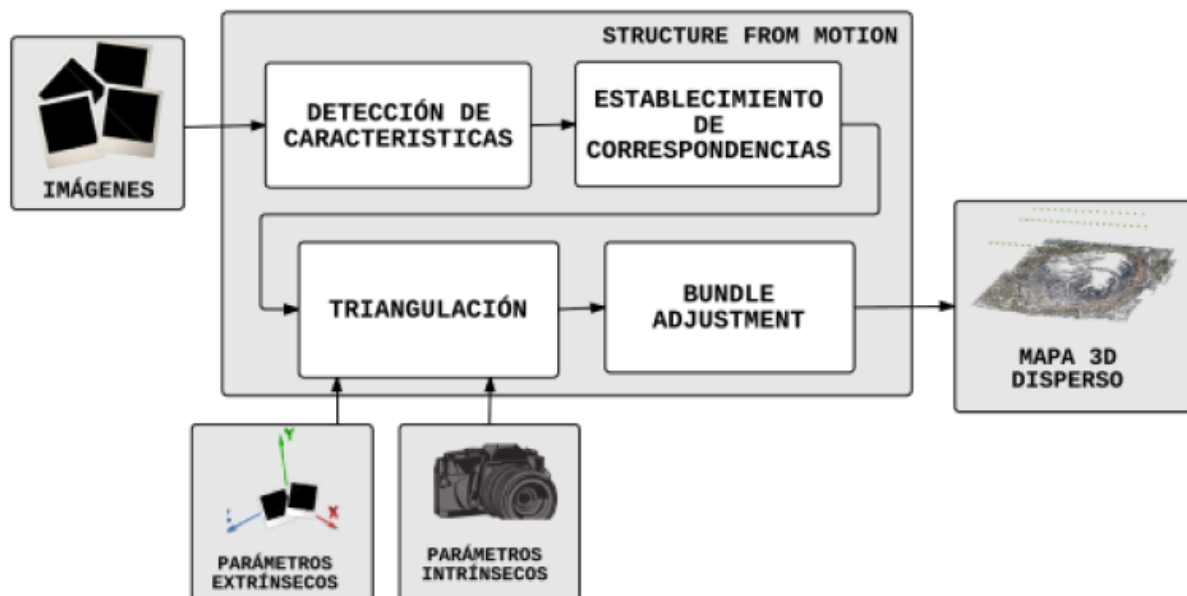
A parte de las imágenes para la calibración, se necesita información del posicionamiento de las cámaras y los respectivos parámetros extrínsecos e intrínsecos.

- **Modelo de mapa dispersos**

El modelo de mapas dispersos es un conjunto de algoritmos el cual permite resolver la reconstrucción 3D a partir de imágenes, este lo hace resolviendo las siguientes incógnitas.

- Geometría de la escena: se establece la correspondencia entre varias imágenes, ¿conocerlos puntos 3D correspondientes a esos puntos?.
- Establecer la correspondencia: Dado el punto 2D de una imagen, ¿Cómo se podría localizar ese punto en las otras imágenes
- Geometría de la cámara: Ya establecidos los puntos correspondientes, ¿Cuáles son los parámetros extrínsecos e intrínsecos asociados a la imagen?

Este conjunto de algoritmos establece la correspondencia entre diferentes imágenes con un objetivo. Obtener las coordenadas 3D mediante un conjunto de algoritmos de triangulación, una vez obtenida la estructura del modelo 3D esta es sometida a un proceso de optimización y minimización de error de reproyección. El modelo es presentado a continuación.



## 2.8 Estado del arte

### 2.8.1 Internacional:

**Atención visual basada en un espacio perceptual conjunto de color y brillo para la mejora de algoritmos de seguimiento de objetos en secuencias de vídeo.**(Fernandez-Carbajales Cañete 2018). En este trabajo se propone un nuevo modelo de atención visual que fusiona color y brillo, lo cual permite una caracterización más discriminativa y robusta de los objetos, incluso si son visualmente muy similares, que puede ser explotada en aplicaciones como el seguimiento de objetos. Los resultados experimentales con secuencias de vídeo reales muestran que el modelo de atención visual propuesto, mejora significativamente la tarea del seguimiento de objetos en secuencias de vídeo al encontrar características visuales más discriminantes, especialmente cuando se trata de objetos con una alta similitud visual, obteniendo resultados más precisos en este ámbito de aplicación que otros modelos de atención visual ampliamente conocidos que procesan el color y el brillo por separado.

**Advanced tracking and image registration techniques for intraoperative radiation therapy.**(Marinetti Carrillo 2017).El objetivo principal de este trabajo es la evaluación de alternativas tecnológicas y procedimentales para la mejora de la práctica de la RIO y resolver los problemas de incertidumbre descritos anteriormente. El rastreador óptico de 8 cámaras OptiTrack se evaluó en términos de Sensibilidad de calibración, precisión, oclusiones de cámara y detección de herramientas en una configuración clínica factible que cubre una amplia gama de aplicaciones y escenarios IGS.

**Diseño y construcción de un dispositivo para realizar reconstrucción tridimensional basado en un sistema estéreo de dos cámaras.**(Portella Coaguila 2015). Se desarrollará un sistema estéreo de dos cámaras, que mediante algoritmos de reconstrucción tridimensional basados en un lenguaje de programación C++, permita detectar las características reales del objeto a reconstruir. Los resultados obtenidos en el archivo Calib\_Results.m, proporciona los parámetros intrínsecos y extrínsecos para la cámara izquierda o derecha. De la misma manera, se realiza el mismo procedimiento para la cámara restante.

**Reconstrucción 3d a partir de imágenes rayos x de frutos con semilla.**(López Yovera 2018) Este trabajo plantea como objetivo la aplicación de 2 técnicas de Visión Estéreo para la reconstrucción 3d de 2 imágenes rayos x del fruto de la palta con semilla; esta investigación surge debido a que el tiempo de congelación resulta ser mucho más complejo según la variación de forma del producto, y a pesar que existen modelos matemáticos con el que podemos apoyarnos para determinar resultados de congelamiento estos aún siguen siendo aproximaciones, ya que la forma geométrica como esfera, elipses, cilindros no son formas precisas que se puedan comparar con otros tipos de frutos como la palta con semilla, ya que esta fruta es solo una

aproximación a figuras geométricas, como sabemos tiene variación de forma lo que hace que el cálculo de enfriamiento no sea el adecuado; el propósito es entonces reconstruir la forma real del fruto de la palta en tres dimensiones permitiendo de esta manera que los cálculos de enfriamiento del fruto sea el adecuado en las industrias que practican este tipo de procesos. Para realizar esta investigación, se ha estudiado las bases teóricas y los trabajos previos relacionados a la reconstrucción 3d mediante visión estéreo usando el método 1: Técnica tradicional y método 2: Técnica de pistas. Se utilizaron 2 imágenes rayos x del fruto de la palta como muestra para la puesta a prueba de nuestros métodos implementados, los indicadores utilizados para la evaluación de cada método fueron el tiempo de procesamiento del mapa de disparidad, tiempo de reconstrucción del fruto de la palta con semilla, porcentaje de consumo de memoria en procesar el mapa de disparidad, porcentaje de precisión y error en la reconstrucción 3d del fruto de la palta con semilla. En esta investigación se desarrollan dos métodos para la reconstrucción 3d del fruto de la palta con semilla , y se determinó que el método que procesa de manera más rápida el mapa de disparidad mediante el ingreso de 2 imágenes rayos x cuya dimensión es (648x864 pixeles) es el Método 1 que permite preparar el mapa de disparidad para poder reconstruirlo basándose en la técnica tradicional, obteniendo un tiempo de 18.2825 segundos, en comparación con el método 2 que permite preparar el mapa de disparidad para poder reconstruirlo basándose en la técnica de pistas ,que tarda un tiempo de 19.451 segundos en procesar el mapa de disparidad, tardando 1.1685 segundos más que el método 1.

**Road Scene Interpretation For Autonomous Navigation Fusing Stereo Vision and Digital Maps.**(Fernández López 2016). Conseguir una imagen a priori de la forma de la carretera basado en la posición actual del vehículo y la información de las calles proporcionada por el mapa. Las imágenes de detección de carreteras se comparan con algunos de los algoritmos clasificados en el punto de referencia KITTI. Para concluir El capítulo, una discusión de algunas situaciones donde nuestro algoritmo no detectan correctamente la carretera.

**Sistema de reconocimiento y seguimiento de objetos en tiempo real a través de visión artificial.**(Sánchez Asmat 2016). Determinar las características adecuadas de un sistema de reconocimiento y seguimiento de objetos en tiempo real a través de técnicas de visión artificial y procesamiento de imágenes. Los resultados obtenidos con cada una de las pruebas eran similares, por lo tanto, la decisión final fue por esta cámara porque era con la que se contaba inicialmente de propiedad propia del tesista, mientras que los otros modelos eran temporales por préstamo.

### **2.8.2 Nacional:**

**Detección de vehículos empleando visión por computador en el marco del proyecto ComVidas.**(Romero and Viveros Albornoz 2018). Desarrollar un sistema de detección de vehículos basado en visión por computador empleando un cámara monocular implementado en el sistema ADAS de ComVidas. Los resultados del DataSet 1 arroja un balance positivo

teniendo en cuenta las condiciones de iluminación de las imágenes, hubo 15 aciertos de 20 posibles y un total de 5 falsos-positivos dando un porcentaje de acierto del 85% para el dataset 1.

**Detección y conteo de personas en espacios cerrados utilizando estrategias basadas en visión artificial.**(Felipe et al. 2017). Diseñar, implementar y evaluar un sistema de detección y conteo de personas en imágenes interiores de video vigilancia utilizando estrategias basadas en visión artificial. Obtener de las curvas de aprendizaje permitió verificar y evaluar cuatro aspectos fundamentales de cada modelo construido Magnitud de puntaje de entrenamiento correcta, Magnitud de puntaje de validación alto, Sobreajuste de modelos, Sobre entrenamiento de modelos.

**Seguimiento de Puntos en Imágenes Diagnósticas.**(Leonardo and Fajardo 2015b). Adaptar y mejorar varios modelos que permita seguir uno o más puntos ubicados en un espacio matricial cartesiano, que está representado en secuencias de imágenes con el fin de analizar exámenes diagnósticos. Para el caso del filtro de Kalman se ve una mejoría generalizada en la implementación de los métodos, con excepción de la tercera versión en la cual se adapta un barrido en estrella que permite que la misma implementación pueda identificar y validar la medida del error con la predicción hecha. KLT que no presentan problemas en el procesamiento de las imágenes de las pruebas base, pero si presento problemas en el procesamiento de las imágenes médicas en donde hubo una afectación de los resultados debido al brillo de las imágenes.

**Learning to Track: Online Multi-Object Tracking by Decision Making.**(Xiang et al. 2016).El seguimiento de objetos múltiples (MOT) en línea tiene aplicaciones amplias en escenarios de análisis de video críticos para el tiempo, como la navegación con robots y la conducción autónoma. En el seguimiento por detección, un desafío importante de la MOT en línea es cómo asociar de manera sólida detecciones de objetos ruidosos en un nuevo cuadro de video con objetos previamente rastreados. En este trabajo, formulamos el problema de MOT en línea como toma de decisiones en los procesos de decisión de Markov (MDP), donde la vida útil de un objeto se modela con un MDP. Aprender una función de similitud para la asociación de datos es equivalente a aprender una política para el MDP, y el aprendizaje de políticas se aborda en una forma de aprendizaje reforzado que se beneficia tanto de las ventajas del aprendizaje fuera de línea como del aprendizaje en línea para la asociación de datos. Además, nuestro marco puede, naturalmente, manejar el nacimiento muerte y la aparición desaparición de objetivos tratándolos como transiciones de estado en el MDP al tiempo que aprovecha los métodos de seguimiento de objetos únicos en línea existentes. Realizamos experimentos en el MOT Benchmark para verificar la efectividad de nuestro método. La mesa que realiza entrenamientos y pruebas en escenarios similares es beneficiosa. Por ejemplo, el rastreador capacitado en ADLRundle-6 logra el mejor rendimiento en ADL-Rundle-8. Segundo, los rastreadores



entrenados en las cinco secuencias de entrenamiento se desempeñan razonablemente bien en todas las secuencias de prueba.

### **2.8.3 Regional:**

**Generación de mapas de disparidad, a partir del uso de Imágenes rectificadas** .(Beainy et al. 2013).Dentro del desarrollo de este proyecto, se seleccionaron imágenes proporcionado por el repositorio Middlebury College, con el propósito de la selección de estas imágenes es para verificar el comportamiento de algoritmos para aislar los problemas inherentes a los mismos, tal que en el siguiente paso de su aplicación al mundo real, solo existen esos problemas de esta naturaleza, como diferentes niveles de intensidad en las dos imágenes, debido al resplandor y otros efectos de iluminación de ambientes no estructurados, distorsión radial en lente o desplazamientos verticales de las cámaras. Este artículo realiza una evaluación de algoritmos para calcular el mapa de disparidad en sistemas de visión estéreo, una idea para el desarrollo futuro de este proyecto consiste en la implementación de algoritmos en sistemas embebidos, para observar el comportamiento del sistema frente a un escenario en tiempo real ejecución.

**Restricciones en la visión estéreo humana para tele-operación.**(Camposeco et al. 2011). Este artículo introdujo un sistema para presentar imágenes en tiempo real dicópticamente a los humanos. El sistema consta de un par de cámaras y un HMD. La distancia de referencia entre cámaras se puede ajustar a voluntad y los usuarios pueden usar el dispositivo completo fuerte mientras mantiene sus manos libres para la manipulación. El sistema se caracteriza principalmente por su simplicidad, portabilidad, y confiabilidad. Se evaluaron algunas limitaciones de la visión estéreo humana. A través de un conjunto de pruebas que involucran rivalidad de color, color selectivo supresión y efectos de la variación de la distancia inter-ocular.

**Computer Vision in AUVs: Automatic Roto-Rectification of Stereo Images.**(Zelasco et al. 2000). Se han tenido experiencias con pares estereoscópicos sintéticos, así como con imágenes de escenas reales. La amabilidad de la solución propuesta mejora en escenas en las que sus características se dispersan en toda la imagen, excepto una pequeña región en los extremos que no son considerados por la estrategia de control utilizada al obtener el entorno de búsqueda. Esto es debido a Las características de la última etapa del algoritmo, para que sea necesario no considerar puntos homólogos que están más cerca, en lugar del desenfocado de más alto nivel. Este requisito coincide con la conveniencia de que el grupo mínimo de homólogos los puntos a ser rectificadas también se encuentran distribuidos en Toda la imagen.

Correspondería en el futuro comparando resultados de la búsqueda exhaustiva de todos los homólogos puntos en imágenes no roto-rectificadas versus resultados obtenido aplicando nuestra secuencia: búsqueda de un mínimo grupo de puntos para la roto-rectificación, la roto-rectificación misma, y finalmente búsqueda de puntos homólogos con algoritmos que requieren la condición epipolar.

**Seguimiento de objetos 3D basado en información visual para vehículos aéreos no tripulados.** (Mondragón et al. 2011). En este artículo se basa un método para seguir objetos voladores solo en la información visual proporcionada por un color adaptativo. Se ha presentado un algoritmo de seguimiento. El método ha sido validado experimentalmente en una prueba real utilizando un UAV de viento rotativo en escenarios al aire libre. El método propuesto ha demostrado ser robusto contra El clima, el viento y las variaciones de luz realizan un seguimiento correcto. ing y siguiendo la acción del objetivo. El algoritmo propuesto permite rastrear y sigue un objeto en movimiento en un espacio tridimensional, sin ningún tipo de restricción dinámica. La tarea de control se realiza en tal una forma de lograr un control tridimensional, lo que permite tener un cambio de altitud tanto en el objetivo como en el seguidor vehículo.

**Una comparación de esquemas basados en Kalman para localización y seguimiento en sistemas de sensores.** (Patiño, Espinosa, and Correa 2010). En este artículo, se estudió el desempeño de tres Modelos basados en EKF para el seguimiento de objetivos en WSN. se demostró que el seguimiento en WSN se puede resolver con P, PV o modelos de estimación PVA EKF dependiendo de la aplicación requisitos Señalamos el impacto del valor medio de la matriz de covarianza de datos en la efectividad de cada modelo. Los mejores resultados estadísticos se obtuvieron para el modelo PVA con el valor  $q$  más bajo; este es el modelo que intenta reunir La mayoría de la información del proceso con la estimación de la velocidad y aceleración, Además, estudiamos el impacto del número de balizas en el rendimiento de los modelos de estimación, y mostramos la mejora con un mayor número de puntos de referencia. Como trabajo futuro, nos estamos centrando en impacto del número de balizas y su ubicación en el conjunto seguimiento del rendimiento de la estimación.

### 3 Análisis y evaluación de los algoritmos

#### 3.1 Retos y dificultades

Al seguir un objeto en una secuencia de imágenes (video), este tiende a abrir una gran cantidad de dificultades asociadas que se deben tener en cuenta, las más comunes que se pueden encontrar son las siguientes.

##### 3.1.1 Cambios de iluminación

El cambio de iluminación en una secuencia de video afecta los valores de los píxeles de la imagen, Debido al cambio producido por los píxeles los métodos que se basan en la obtención de características en función de los valores de los píxeles son poco efectivos ante este tipo de problema.

##### 3.1.2 Oclusión

Se puede denominar oclusión a todo acto en el cual el objetivo se vea totalmente cubierto a través de la acción del entorno o de otro objeto.

##### 3.1.3 Deformación

Esta cualidad depende de las características físicas del objeto, dependiendo de estas cualidades es el grado de deformaciones que puede experimentar. Según las características se pueden clasificar en tres tipos: objetos flexibles, objetos articulares y objetos rígidos, siendo los dos primeros los que presentan mayor dificultad.

##### 3.1.4 Ruido

Habitualmente proviene del proceso de captación de la señal, aunque también puede aparecer en las etapas de codificación y cuantificación o transmisión.

##### 3.1.5 Cambios de perspectiva

Los objetos pueden experimentar rotaciones, cuando el objeto experimenta rotaciones cambia parcialmente su apariencia por lo cual se dificultará enormemente su seguimiento.

##### 3.1.6 Emborronado

Este se produce a partir de la limitada resolución temporal que cuentan algunos dispositivos ya que en algunos cuadros de secuencia el objeto se mueve muy rápido.

### 3.2 Análisis de los algoritmos de seguimiento planteados y estudiados.

Esta sección se hace un respectivo análisis de los algoritmos de seguimiento para determinar las ventajas y desventajas de estos, posteriormente se diseñará una tabla donde se tendrá en cuenta los retos y dificultades anteriormente mencionados para hacer una evaluación rápida de los algoritmos de seguimiento, con el fin de dar solución a uno de los objetivos planteados.

**Tracking based color:** El color ha sido ampliamente utilizado en sistemas de seguimiento en tiempo real. Ya que estos ofrecen varias ventajas significativas sobre señales geométricas como la simplicidad computacional, la robustez bajo oclusión parcial, rotación, escala y cambios de resolución. Aunque los métodos de color resultaron ser eficiente en una variedad de aplicaciones de visión, hay varios problemas asociados con estos métodos en los que la constancia del color es uno de los más importantes.

#### Ventajas

- El sistema rastrea objetos de colores con buena precisión.
- Permite rastrear objetos en video en vivo.

#### Desventajas

- Baja precisión en imágenes de cámara de baja resolución.
- Baja precisión en condiciones de poca luz.

**BOOSTING Tracker:** Este rastreador se basa en una versión en línea de AdaBoost, el algoritmo que el detector facial basado en cascada HAAR usa internamente. Este clasificador debe ser entrenado en tiempo de ejecución con ejemplos positivos y negativos del objeto. El cuadro delimitador inicial suministrado por el usuario (o por otro algoritmo de detección de objetos) se toma como un ejemplo positivo para el objeto, y muchos parches de imagen fuera del cuadro delimitador se tratan como fondo. Dado un nuevo marco, el clasificador se ejecuta en cada píxel en la vecindad de la ubicación anterior y se registra la puntuación del clasificador.

La nueva ubicación del objeto es aquella donde la puntuación es máxima. Entonces ahora tenemos un ejemplo positivo más para el clasificador. A medida que entran más cuadros, el clasificador se actualiza con estos datos adicionales.

#### Ventajas

- El rastreador se utiliza para hacer comparaciones.
- El rastreador utiliza herencia.

#### Desventajas

- El rastreador es muy lento.
- El rastreador no funciona muy bien.

**MIL Tracker:** Este rastreador es similar en idea al rastreador BOOSTING descrito anteriormente. La gran diferencia es que, en lugar de considerar solo la ubicación actual del objeto como un ejemplo positivo, se ve en un pequeño vecindario alrededor de la ubicación actual para generar varios ejemplos positivos potenciales. Puede estar pensando que es una mala idea porque en la mayoría de estos ejemplos” positivos. Él objeto no está centrado. Aquí es donde el aprendizaje de instancias múltiples (MIL) viene a rescatar. En MIL, no se especifican ejemplos positivos y negativos, sino “bolsas” positivas y negativas. La colección de imágenes en la bolsa positiva no son todos ejemplos positivos. En cambio, ¡solo una imagen en la bolsa positiva debe ser un ejemplo positivo! Una bolsa positiva contiene el parche centrado en la ubicación actual del objeto y también parches en un pequeño vecindario a su alrededor. Incluso si la ubicación actual del objeto rastreado no es precisa, cuando las muestras de la vecindad de la ubicación actual se colocan en la bolsa positiva, existe una buena posibilidad de que esta bolsa contenga al menos una imagen en la que el objeto esté bien centrado. Página del proyecto MIL tiene más información para las personas que les gusta profundizar en el funcionamiento interno del rastreador MIL.

#### **Ventaja**

- El rendimiento es bastante bueno. No se desliza tanto como el rastreador BOOSTING y hace un trabajo razonable bajo oclusión parcial. Si está utilizando OpenCV 3.0, este podría ser el mejor rastreador disponible para usted. Pero si está utilizando una versión superior, considere KCF.

#### **Desventajas**

- La falla de seguimiento no se informa de manera confiable. No se recupera de la oclusión completa.

**KCF Tracker:** KFC significa filtros de correlación kernelized . Este rastreador se basa en las ideas presentadas en los dos rastreadores anteriores. Este rastreador utiliza el hecho de que las múltiples muestras positivas utilizadas en el rastreador MIL tienen grandes regiones superpuestas. Estos datos superpuestos conducen a algunas buenas propiedades matemáticas que este rastreador aprovecha para hacer que el seguimiento sea más rápido y preciso al mismo tiempo.

#### **Ventaja**

- La precisión y la velocidad son mejores que MIL e informa de un error de seguimiento mejor que BOOSTING y MIL. Si está utilizando OpenCV 3.1 y superior, le recomiendo usar esto para la mayoría de las aplicaciones.

#### **Desventajas**

- No maneja bien la oclusión completa.

**MedianFlow Tracker:** Internamente, este rastreador rastrea el objeto tanto en dirección hacia adelante como hacia atrás en el tiempo y mide las discrepancias entre estas dos trayectorias. La minimización de este error ForwardBackward les permite detectar de manera confiable fallas de seguimiento y seleccionar trayectorias confiables en secuencias de video.

En mis pruebas, descubrí que este rastreador funciona mejor cuando el movimiento es predecible y pequeño. A diferencia de otros rastreadores que continúan incluso cuando el seguimiento ha fallado claramente, este rastreador sabe cuándo ha fallado el seguimiento.

#### **Ventaja**

- Hace un buen trabajo al informar fallas.

#### **Desventajas**

- El modelo fallara si se encuentra con un salto de movimiento muy grande.

**TLD Tracker:** TLD significa seguimiento, aprendizaje y detección. Como su nombre indica, este rastreador descompone la tarea de seguimiento a largo plazo en tres componentes: seguimiento, aprendizaje y detección (a corto plazo). Del artículo del autor, “El rastreador sigue el objeto de cuadro a cuadro. El detector localiza todas las apariencias que se han observado hasta ahora y corrige el rastreador si es necesario. El aprendizaje estima los errores del detector y lo actualiza para evitar estos errores en el futuro”. Esta salida de este rastreador tiende a saltar un poco. Por ejemplo, si está rastreando a un peatón y hay otros peatones en la escena, este rastreador a veces puede rastrear temporalmente a un peatón diferente al que tenía previsto rastrear. En el lado positivo, esta pista parece rastrear un objeto a mayor escala, movimiento y oclusión.

#### **Ventaja**

- Funciona mejor bajo oclusión en múltiples cuadros. Además, rastrea mejor los cambios de escala.

#### **Desventajas**

- El modelo es muy propenso a fallas.

**CSRT Tracker:** En el Filtro de correlación discriminativa con canal y confiabilidad espacial (DCF-CSR), utilizamos el mapa de confiabilidad espacial para ajustar el soporte del filtro a la parte de la región seleccionada desde el marco para el seguimiento. Esto asegura la ampliación y localización de la región seleccionada y un seguimiento mejorado de las regiones u objetos no rectangulares. Utiliza solo 2 características estándar (HoGs y Colornames). También funciona a

fps comparativamente más bajos (25 fps) pero ofrece una mayor precisión para el seguimiento de objetos.

#### **Ventaja**

- Es más preciso que KCF.

#### **Desventajas**

- Más lento que KFC.

**MOSSE Tracker:** La suma mínima de salida de error cuadrado (MOSSE) utiliza correlación adaptativa para el seguimiento de objetos que produce filtros de correlación estables cuando se inicializa usando un solo cuadro. El rastreador MOSSE es robusto a las variaciones de iluminación, escala, pose y deformaciones no rígidas. También detecta la oclusión en función de la relación pico a lóbulo lateral, lo que permite que el rastreador haga una pausa y reanude donde se quedó cuando el objeto vuelve a aparecer. El rastreador MOSSE también funciona a un fps más alto (450 fps e incluso más). Para agregar a los aspectos positivos, también es muy fácil de implementar, es tan preciso como otros rastreadores complejos y mucho más rápido.

#### **Ventaja**

- Es muy, muy rápido.

#### **Desventajas**

- No es tan preciso como CSRT o KCF.

#### **Motion detection**

##### **Ventajas**

- Es un método bastante robusto a los cambios de iluminación.
- Es un método bastante robusto a los cambios de movimiento.
- Consigue estabilizarse pasado un tiempo.

##### **Desventajas**

- Si el objeto o la persona en movimiento se quedan quietos, no se detecta.
- No es capaz de detectar siluetas.

**SSD single shot detection:** Es un algoritmo popular en la detección de objetos. Generalmente es más rápido que Faster RCNN

##### **Ventajas**

- Es un método bastante ágil.
- Es un método eficiente basado en el aprendizaje profundo.
- Modelo fácil de entrenar.
- Modelo de fácil integración.

### Desventajas

- Pocas etiquetas en la clase MobileNets.

A continuación, se presentarán el conjunto de tablas donde se evalúan los criterios descritos anteriormente en tres categorías de puntos:

Buena 1.0

Regular 0.5

Malo 0.0

Criterios	Color	Movimiento	Región de interés	Boosting	MIL
Claridad	0.5	1.0	0.5	1.0	0.5
Fácil implementación	1.0	1.0	1.0	0.5	0.5
Ruido	1.0	1.0	0.0	0.5	1.0
Cambios de perspectiva	1.0	0.0	1.0	0.0	1.0
Sobreposición	1.0	0.5	0.5	0.0	0.5
Seguimiento	0.5	1.0	1.0	0.5	0.5
Rapidez	1.0	1.0	1.0	0.5	1.0
<b>Resultado</b>	<b>0.85</b>	<b>0.6</b>	<b>0.7</b>	<b>0.4</b>	<b>0.7</b>

*Tabla 1 Resultados de los algoritmos estudiados*

Criterios	KCF	Median flow	TLD	CSRT	Mosse	SSD
Claridad	1.0	1.0	1.0	1.0	1.0	1.0
Fácil implementación	0.5	0.5	0.5	1.0	0.5	0.5
Ruido	1.0	1.0	1.0	1.0	1.0	0.5
Cambios de perspectiva	1.0	0.5	0.5	1.0	0.5	0.5
Sobreposición	0.5	0.5	0.5	1.0	0.5	0.5
Seguimiento	1.0	0.5	0.5	1.0	0.5	1.0
Rapidez	1.0	0.5	0.5	0.5	1.0	0.5
<b>Resultado</b>	<b>0.86</b>	<b>0.6</b>	<b>0.6</b>	<b>0.9</b>	<b>0.7</b>	<b>0.6</b>

*Tabla 2 Resultados de los algoritmos estudiados*



### 3.3 Experimentación de los algoritmos seleccionados

- **Seguimiento múltiple de objetos. (normal):**

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
<b>Prueba 1</b>	100%	3	Amarillo, Azul, Rojo	9	0%
<b>Prueba 2</b>	100%	4	Verde, Blanco, Azul, Rojo	9	0%
<b>Prueba 3</b>	100%	3	Verde, Azul, Blanco	Variado 1 obj. 5 2 obj. 9	0%
<b>Prueba 4</b>	100%	2	Azul, Rojo	9	0%
<b>Prueba 5</b>	100%	5	Verde, Blanco, Amarillo, Rojo, Azul	Variado 2 obj. 5 3 obj. 9	0%
<b>Prueba 6</b>	100%	2	Azul, Amarillo	9	0%
<b>Prueba 7</b>	100%	3	Rojo, Azul, Amarillo	Variado	0%

Tabla 3 Algoritmo CSRT Seguimiento múltiple de objetos. (Normal)

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
<b>Prueba 1</b>	100%	3	Amarillo, Azul, Rojo	9	0%
<b>Prueba 2</b>	100%	4	Verde, Blanco, Azul, Rojo	9	0%
<b>Prueba 3</b>	66.6%	3	Verde, Azul, Blanco	Variado 1 obj 5 2 obj 9	33.3%
<b>Prueba 4</b>	100%	2	Azul, Rojo	9	0%
<b>Prueba 5</b>	60%	5	Verde, Blanco, Amarillo, Rojo, Azul	Variado 2 obj 5 3 obj 9	40%
<b>Prueba 6</b>	100%	2	Azul, Amarillo	9	0%
<b>Prueba 7</b>	100%	3	Rojo, Azul, Amarillo	9	0%

Tabla 4 Algoritmo de color Seguimiento múltiple de objetos. (Normal)

Después de las 7 pruebas realizadas del algoritmo de Color y CSRT los resultados obtenidos fueron óptimos para el caso del algoritmo CSRT el cual sigue el objeto el 100 % de las veces y el algoritmo de detección de color sigue el objeto un 89.53 % de las veces. En la **Figura 19** se puede observar los resultados obtenidos en el algoritmo CSRT y en la **Figura 20** se puede observar los resultados obtenidos en el algoritmo de color. El porcentaje de error fue calculado a través del error relativo porcentual.

Lugar: Habitación.	Descripción: El video se graba en una habitación controlada con un fondo blanco
Iluminación: iluminación artificial, situados en el techo de la habitación,	Altura de la cámara: La cámara se posiciono a 50 cm sobre el suelo.

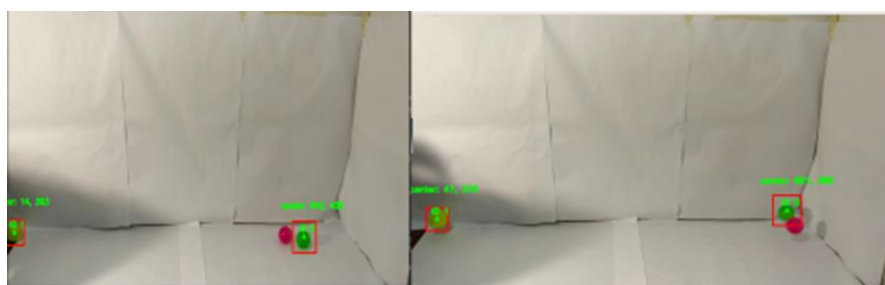


Figura 19: Prueba del algoritmo CSRT con pelotas de colores



Figura 20: Prueba del algoritmo de color con pelotas de colores

- **Seguimiento de un objeto (prueba ping pong):**

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
<b>Prueba 1</b>	100%	1	Blanco	5	0%
<b>Prueba 2</b>	100%	1	Blanco	5	0%
<b>Prueba 3</b>	100%	1	Blanco	5	0%
<b>Prueba 4</b>	100%	1	Blanco	5	0%
<b>Prueba 5</b>	100%	1	Blanco	5	0%
<b>Prueba 6</b>	100%	1	Blanco	5	0%
<b>Prueba 7</b>	100%	1	Blanco	5	0%

Tabla 5 Algoritmo CSRT Seguimiento de un objeto (Prueba ping pong)

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
Prueba 1	0%	1	Blanco	5	100%
Prueba 2	0%	1	Blanco	5	100%
Prueba 3	0%	1	Blanco	5	100%
Prueba 4	0%	1	Blanco	5	100%
Prueba 5	0%	1	Blanco	5	100%
Prueba 6	0%	1	Blanco	5	100%
Prueba 7	0%	1	Blanco	5	100%

Tabla 6 Algoritmo de color Seguimiento de un objeto (Prueba ping pong)

Con estas respectivas pruebas se pretende probar el código ante una experiencia más exigente, donde se batea la pelota contra la pared en diferentes ocasiones como se muestra en la **Figura 21**. al igual que en la situación anterior los resultados obtenidos fueron excelente cumpliéndose con un seguimiento del 100% lo que demuestra la solidez del seguimiento, La prueba del algoritmo de color tuvo un error del 100% ya que no siguió el objeto y se puede ver el resultado en la **Figura 22**. El porcentaje de error fue calculado a través del error relativo porcentual.

Lugar: Habitación.	Descripción: El video se graba en una habitación controlada con un fondo blanco
Iluminación: iluminación artificial, situados en el techo de la habitación,	Altura de la cámara: la cámara se posiciono a 50 cm sobre el suelo.



Figura 21: Prueba del algoritmo CSRT con pelota de ping pong

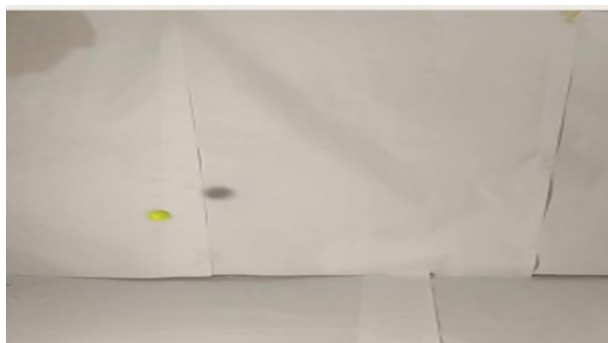


Figura 22: Prueba del algoritmo de color con pelota de ping pong

- **Seguimiento de múltiples objetos donde un objeto presenta oclusión parcial por un periodo corto de tiempo:**

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
Prueba 1	100%	2	Rojo, Azul	5	0%
Prueba 2	100%	2	Amarillo, Verde	9	0%
Prueba 3	100%	4	Azul, Verde, Rojo, Blanco	Variado 2 obj. 5 2 obj. 9	0%
Prueba 4	100%	4	Azul, Amarillo, Verde, Rojo	9	0%
Prueba 5	80%	4	Verde, Azul, Blanco, Amarillo	9	20%
Prueba 6	100%	2	Amarillo, Blanco	9	0%
Prueba 7	100%	2	Azul, Blanco	9	0%

Tabla 7 Algoritmo CSRT Seguimiento de múltiples objetos donde un objeto presenta oclusión parcial por un periodo corto de tiempo

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
Prueba 1	100%	2	Rojo, Azul	5	0%
Prueba 2	100%	2	Amarillo, Verde	9	0%
Prueba 3	50%	4	Azul, Verde, Rojo, Blanco	Variado 2 obj. 5 2 obj. 9	50%
Prueba 4	100%	4	Azul, Amarillo, Verde, Rojo	9	0%
Prueba 5	100%	4	Verde, Azul, Blanco, Amarillo	9	0%
Prueba 6	100%	2	Amarillo, Blanco	9	0%
Prueba 7	100%	2	Azul, Blanco	9	0%

Tabla 8 Algoritmo de color Seguimiento de múltiples objetos donde un objeto presenta oclusión parcial por un periodo corto de tiempo

En esta prueba se trata de observar el seguimiento del algoritmo CSRT y el algoritmo de Color cuando se presenta una oclusión parcial del objeto, El algoritmo CSRT Tubo buenos resultados gracias al soporte espacial del filtro de la aplicación, esta es capaz de seguir los objetos, aunque estos se crucen. Este actúa de manera eficiente en la mayoría de las situaciones, si bien es verdad que hay un pequeño porcentaje de error de 2.85% en las que este no acierta y pierde el rastro del objeto, se puede observar el seguimiento en la **Figura 23**, El seguimiento a Color tuvo un resultado irregular ya que se perdió el seguimiento al ir alejándose con un porcentaje de error de 7.14%, se muestra el resultado en la **Figura 24** y la **Figura 25**. El porcentaje de error fue calculado a través del error relativo porcentual.

Lugar: Campo principal de la universidad de pamplona.	Descripción: El video se graba en el campus de la universidad de Pamplona, en el bloque "EC".
Iluminación: iluminación natural.	Altura de la cámara: La cámara se posiciono a 60 cm sobre el suelo.

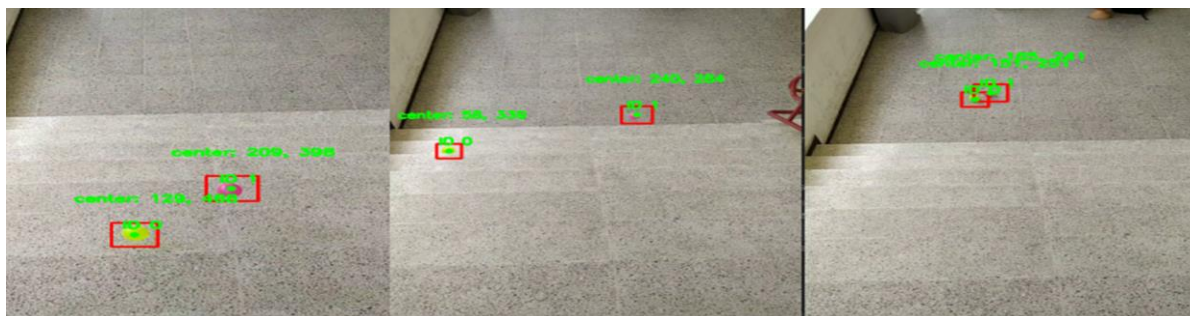


Figura 23: Prueba del algoritmo CSRT en escenario de pérdida parcial de objetos



Figura 24: Prueba del algoritmo de color con pérdida parcial del objeto



Figura 25: Prueba del algoritmo de color con objetos a la distancia

- **seguimiento de múltiples objetos (carrera de objetos):**

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
Prueba 1	100%	2	Rojo, Azul	9	0%
Prueba 2	100%	2	Amarillo, Blanco	9	0%
Prueba 3	100%	3	Rojo, Azul, Blanco	9	0%
Prueba 4	100%	3	Amarillo, Blanco, Verde	Variado 1 obj. 5 2 obj. 9	0%
Prueba 5	100%	5	Verde, Azul, Blanco, Rojo, Amarillo	9	0%
Prueba 6	100%	5	Verde, Azul, Blanco, Rojo, Amarillo	9	0%
Prueba 7	100%	2	Rojo, Amarillo	9	0%

Tabla 9 Algoritmo CSRT seguimiento de múltiples objetos (Carrera de objetos)

Número de pruebas	Precisión de pronóstico de seguimiento	Numero de objetos	Color objetos	Diámetro de objetos en (CM)	Error
Prueba 1	100%	2	Rojo, Azul	9	0%
Prueba 2	100%	2	Amarillo, Blanco	9	0%
Prueba 3	100%	3	Rojo, Azul, Blanco	9	0%
Prueba 4	66.6%	3	Amarillo, Blanco, Verde	Variado 1 obj 5 2 obj 9	33.3%
Prueba 5	100%	5	Verde, Azul, Blanco, Rojo, Amarillo	9	0%
Prueba 6	100%	5	Verde, zul,	9	0%



			Blanco, Rojo, Amarillo		
Prueba 7	100%	2	Rojo, Amarillo	5	0%

Tabla 10 Algoritmo de color seguimiento de múltiples objetos (Carrera de objetos)

Esta prueba se ejecuta en un ambiente no controlado, donde hay fluctuaciones de luminosidad, esta con el fin de añadir una cantidad de complejidad. Los resultados del algoritmo de seguimiento CSRT se observa en la **Figura 26** obteniendo un porcentaje de error del 0% mientras que el algoritmo de color obtuvo un porcentaje de error de 4.7 %. El porcentaje de error fue calculado a través del error relativo porcentual.

Lugar: Campo principal de la universidad de pamplona.	Descripción: El video se graba en el campus de la universidad de Pamplona, en un ambiente no controlado.
Iluminación: iluminación natural.	Altura de la cámara: La cámara se posiciono a 60 cm sobre el suelo.

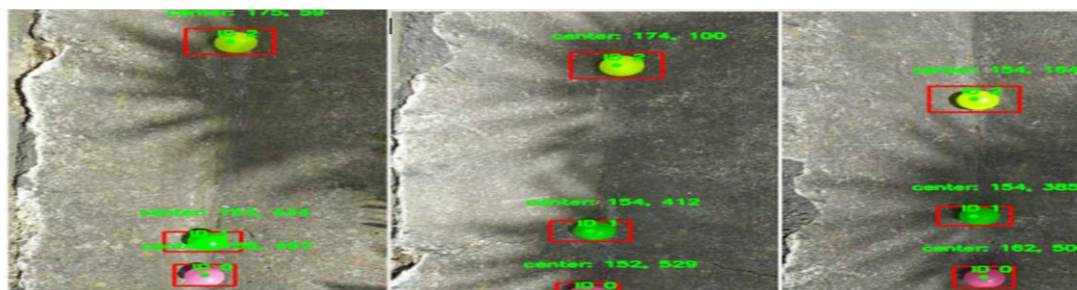


Figura 26: Prueba del algoritmo CSRT con escenario de sombras

### 3.4 Conclusión.

	Algoritmo de color	Algoritmo CSRT
Seguimiento múltiple de objetos. (Normal)	10.47%	0%
Seguimiento de un objeto (Prueba ping pong)	100%	0%
Seguimiento de múltiples objetos donde un objeto presenta oclusión parcial por un periodo corto de tiempo	7.14%	2.85%
color seguimiento de múltiples objetos (Carrera de objetos)	4.7%	0%

De las diferentes pruebas realizadas se puede concluir que el algoritmo que presenta más inconvenientes al momento de realizar un seguimiento, es el algoritmo de color el cual tuvo problemas en cada una de las pruebas , mientras que el algoritmo CSRT solo tuvo problemas en una sola prueba, por lo tanto el algoritmo de seguimiento que utilizamos es el CSRT

### 3.5 Flujograma del Algoritmo para la calibración de las cámaras

El programa para tomar fotos del tablero de ajedrez para una calibración posterior detecta las esquinas del tablero de ajedrez (que esta codificado inicialmente para reconocer un tablero de 9x6) en ambas cámaras, se abren dos ventanas con la imagen reconocida para cada cámara. El usuario podrá, ya sea guardar o eliminar las imágenes mostradas. Las fotos guardadas se utilizarán más tarde para la calibración de las cámaras.

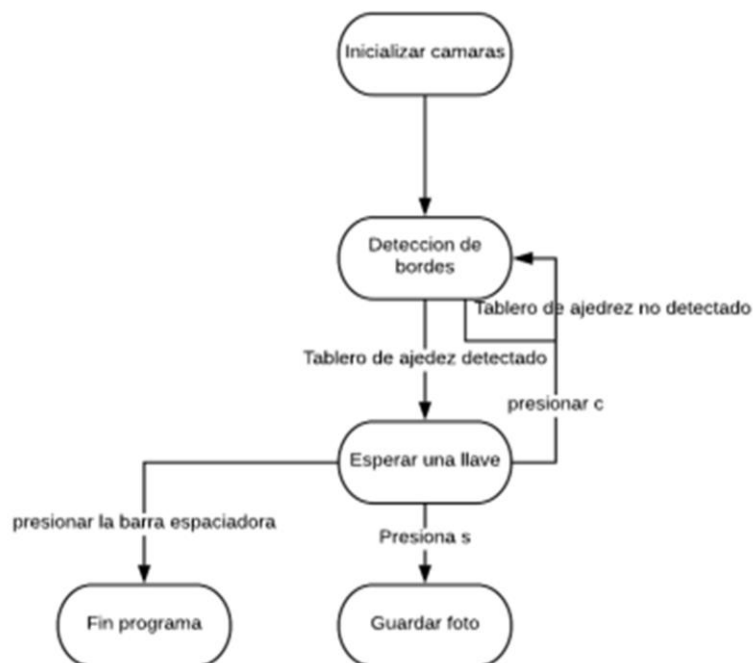


Figura 27: flujograma algoritmo para la calibración de cámaras

### 3.6 Calibración

OpenCV recomienda al menos 10 fotos por cada cámara para obtener una buena calibración. Tenemos 39 para obtener buenos resultados para cada cámara. Para calibrar las cámaras, el código Python busca las esquinas del tablero de ajedrez en cada una.

Para cada imagen con la función de OpenCV: `cv2.findChessboardCorners` se encuentra la posición de las esquinas para cada imagen se almacena en un vector de imagen y los puntos de objeto para la escena 3D se almacenan en otro vector. Tomando los `imgpoints` y `objpoints` obtenidos se utiliza la función `cv2.calibeCamera ()` para obtener como salida la matriz de la cámara, el coeficiente de distorsión, los vectores de rotación y traslación.

### 3.6.1 Calibración de distorsión

Para la corrección de la distorsión de las cámaras, se toman los *Imgpoints* y los puntos de la posición de las esquinas del tablero de ajedrez almacenados. La función `cv2.calibrateCamera()` se usa para crear nuevas matrices de cámara (The Camera Matrix que describe la proyección de un punto del mundo 3D en una imagen 2D), la distorsión, los coeficientes, los vectores de rotación y traslación se obtienen para cada cámara para posteriormente eliminar la distorsión de cualquier cámara. Se utiliza la función `cv2.getOptimalNewCameraMatrix()` para aumentar la precisión.

### 3.6.2 Calibración de la cámara estéreo

Para la calibración estéreo, se utiliza la función `cv2.StereoCalibrate()`, que calcula la transformación entre ambas cámaras (una cámara sirve como referencia para la otra). La función `cv2.stereoRectify()` permite que las líneas epipolares de ambas cámaras estén activas para traer

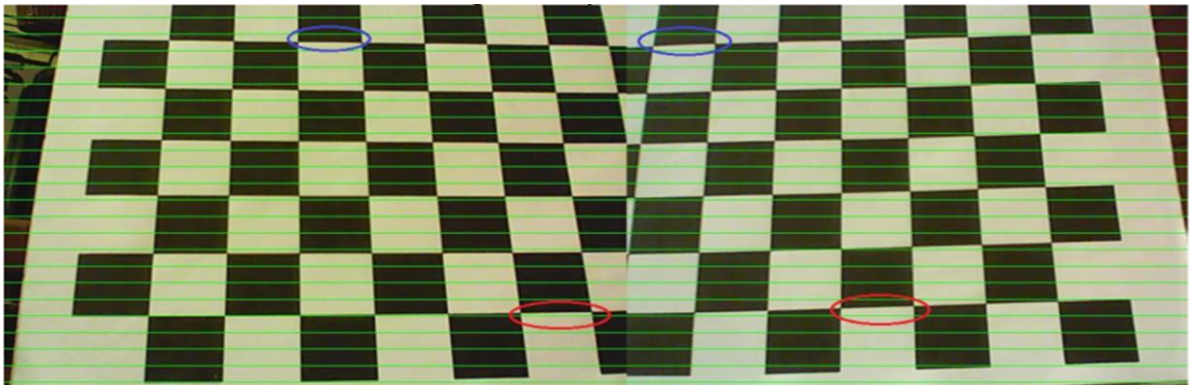


Figura 28: Captura de cámaras sin calibración

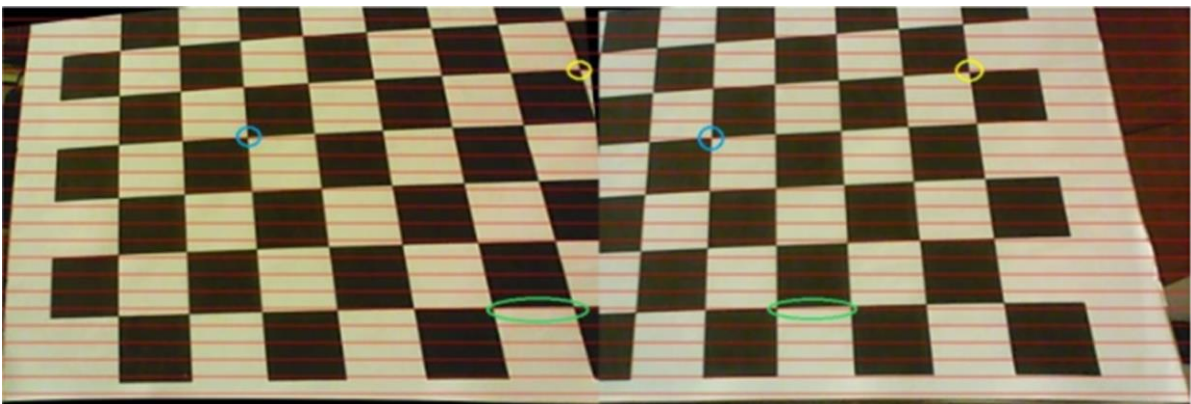


Figura 29: captura de cámaras calibrada

el mismo nivel. Esta transformación facilita el trabajo para la función de disparidad.

### 3.6.3 Cálculo del mapa de disparidad

Para calcular el mapa de disparidad, se crea un objeto StereoSGBM con la función `cv2.StereoSGBM_create()`. Esta clase utiliza un algoritmo de coincidencia semi-global. Para una coincidencia estéreo entre las imágenes de la derecha cámara derecha y la izquierda para llegar.

El objeto SGBM generado compara los bloques de una imagen con los bloques de la imagen coincidente. Si la calibración estéreo se ha hecho bien, el cálculo de la tarjeta de disparidad se vuelve más eficiente. Tomemos el ejemplo anterior con los bloques de la cuarta fila para explicar cómo se hace la tarjeta de disparidad. En la ilustración inferior, se debe comparar el bloque (4,7) de la cuarta fila, la séptima columna de la imagen base con todos los demás bloques (4, i) de la cuarta fila (misma línea epipolar) de la imagen coincidente. Cuanto mayor sea la alineación entre la referencia del Bloque con el bloque de coincidencia es más probable que tenga que ser el mismo punto en el entorno. El programa utiliza imágenes rectificadas en blanco y negro para calcular el mapa de disparidad.

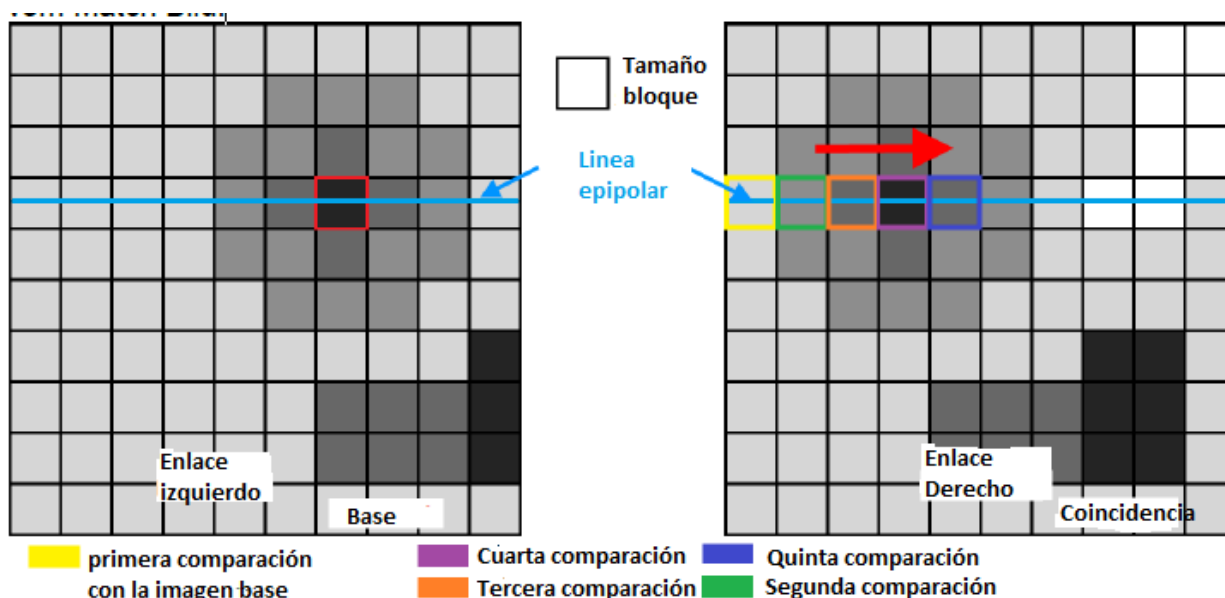


Ilustración 1 coincidencia de bloque con StereoSGBM

### 3.7 Calculando la distancia desde las cámaras al objeto

Después de que se haya generado el mapa de disparidad, se debe determinar la distancia. El trabajo es encontrar la relación entre el valor de disparidad y la distancia. Para hacerlo hay que determinar una regresión lineal.

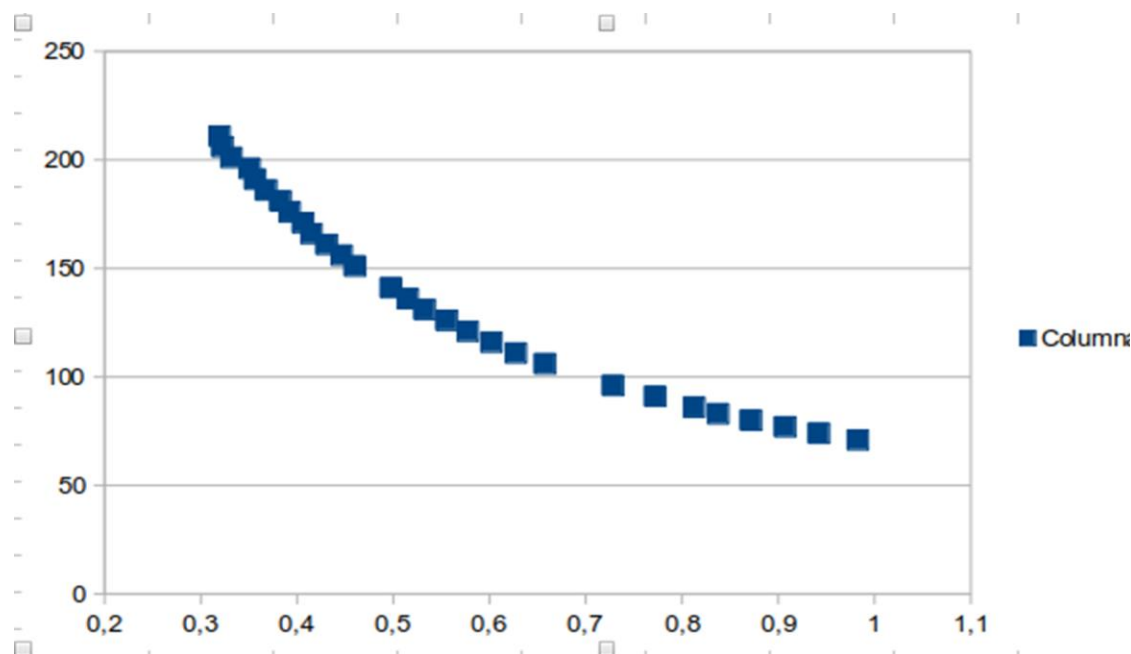


Figura 30: Regresión lineal para obtener ecuación de distancia

Ecuación de línea recta en el programa Python.

$$\text{Distance} = -469.67245074286 * (\text{average}) ** 3 + 1207.65108848869 * (\text{average}) ** 2 - 1135.03367515956 * (\text{average}) + 462.73187741405$$

Figura 31: Ecuación de línea recta

Para obtener esta ecuación de regresión en línea recta, se utilizó la capa de empaque openpyxl para guardar los valores de la disparidad en un archivo de Excel. La medición de distancia solo es válida desde una distancia de 67 cm a 203 cm para obtener buenos resultados. La precisión de la medición también depende de la calidad de la calibración. Con nuestras cámaras estéreo están a una distancia de un objeto con una precisión de +/- 3 cm

### 3.8 Reconstrucción de la escena 3D

Para la reconstrucción de la escena 3D se establecen las coordenadas 3D a partir de las proyecciones 2D de la matriz rectificada, donde las imágenes son coplanares, de la **Figura 17** se obtuvieron las coordenadas X, Y a través de semejanza de triángulos.

Dónde:

$$(1) \quad \frac{x}{z} = \frac{x_1 - x_o}{f}$$

Despejando a X se obtiene:

$$(2) \quad X = \frac{z}{f}(x_1 - x_o)$$

Despejando ha Y se obtiene:

$$(3) \quad Y = \frac{z}{f}(y_1 - y_o)$$

La ecuación (2) representa la coordenada X del punto P en las coordenadas 3D y la ecuación (3) representa la coordenada Y del punto P en las coordenadas 3D.

Z hace referencia a la distancia obtenida anteriormente, la f es la distancia focal de las cámaras la cual es igual para ambas y esta viene dada en unidades de pixel,  $y_1$  es la coordenada Y del centroide en la imagen izquierda e  $y_o$  es la coordenada Y del punto principal de la imagen,  $x_o$  es la coordenada X del punto principal de la imagen,  $x_1$  es la coordenada del objeto en el eje X de la cámara izquierda.

Se debe tener cuidado con las unidades de cada parámetro esto con el fin de que las unidades correspondan a la distancia del objeto. Teniendo las fórmulas (2), (3) ya se puede calcular las coordenadas de todos los puntos de la escena tridimensional, es decir, podemos conocer la posición del objeto en cada instante de tiempo, teniendo entonces el seguimiento del objeto y las coordenadas de este completamente definidas se puede representar cada punto en un gráfico.

### 3.9 Seguimiento de los objetos

Para el algoritmo de seguimiento se utilizó el Filtro de correlación discriminativa con canal y confiabilidad espacial (DCF-CSR), utilizamos el mapa de confiabilidad espacial para ajustar el soporte del filtro a la parte de la región seleccionada desde el marco para el seguimiento. Esto asegura la ampliación y localización de la región seleccionada y un seguimiento mejorado de las regiones u objetos no rectangulares. Utiliza solo 2 características estándar (HoGs y Colornames). También funciona a fps comparativamente más bajos (25 fps) pero ofrece una mayor precisión para el seguimiento de objetos.

### 3.10 Integración de los algoritmos

Luego de analizar los métodos seleccionados se realizaron las pruebas en la cuales se tuvieron en cuenta las complicaciones que se presentaron durante el desarrollo del trabajo de grado, como la claridad, ruido, sobre-posición, seguimiento y rapidez. Dando como resultado la integración del algoritmo de seguimiento CSRT con el algoritmo de visión estereoscópica para dar solución a la problemática planteada en este proyecto, dando como resultado un código que permite observar el seguimiento de objetos en el espacio y detección de su trayectoria ,eficiente, rápido y preciso de error ,en cuanto al seguimiento, este es capaz de seguir el objeto mientras no presente pérdida total del objeto ,el flujo grama del algoritmo se presenta en la **Figura 32**.

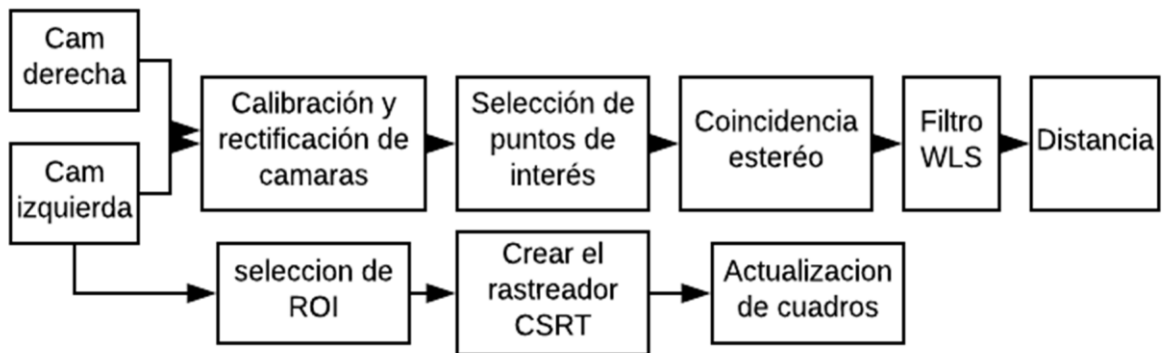


Figura 32: Flujograma del algoritmo resultante

A continuación, se muestra en la **Figura 33** el panel de control, al principio los valores predeterminados se pueden considerar los mejores parámetros escogidos, pero el usuario puede modificarlos a su disposición



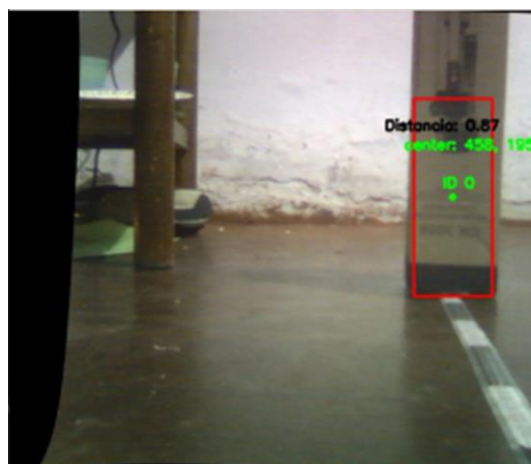
Control									
Disparidad Minima									
2									
0	1	2	3	4	5	6	7	8	9
Tamaño de los bloques									
3									
0	10	20	30	40					
Ratio de unicidad									
10									
0	10	20	30	40					
Lambda									
8000									
4000	5000	6000	7000	8000	9				
Sigma									
1.8									
0.0	0.4	0.8	1.2	1.6	2.0	2.4	2.8	3.2	3.6
iniciar									

Figura 33: Control inicial de parámetros de ejecución

El primer parámetro se escogió en dos ya que solo se está comparando la imagen recogida de la cámara derecha y la izquierda. El tamaño de bloques hace referencia al tamaño de la imagen que se va a comparar con la otra y el ratio de unicidad es el margen en porcentaje por el cual el mejor (mínimo) valor de función de costo calculado debe "ganar" el segundo mejor valor para considerar la coincidencia encontrada correcta. Normalmente, un valor dentro del rango 5-15 es lo suficientemente bueno.

Los otros dos valores lambda y sigma son para modificar el filtro WLS que afectan el suavizado de los bordes.

Al ejecutar el programa nos abrirá una ventana donde al seleccionar el objeto determinará las coordenadas X, Y, Z y el seguimiento de este, mostrará y guardará las coordenadas en un archivo Excel para su posterior graficación. Un ejemplo se puede ver en la Figura 34:



*Figura 34: Medición y seguimiento de objeto detectado*



*Figura 35: Distancia real 87 cm*

### 3.11 Reconstrucción del seguimiento de un objeto

Para validar la generación de trayectorias del algoritmo resultante, se realizó una prueba donde se establece el seguimiento de un objeto. En la **Figura 36** se puede observar el movimiento que hizo el objeto en ciertos instantes de tiempo. Y en la **Figura 37** se establece el resultado de la graficación de las coordenadas 3D obtenidas.



Figura 36: Frames representativos del seguimiento hechos al objeto

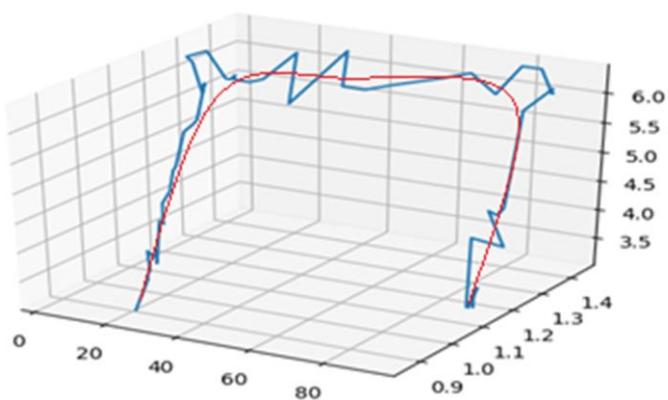
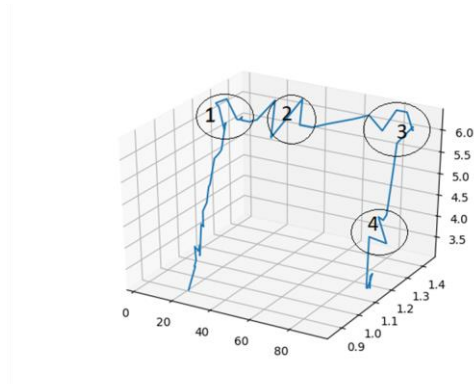


Figura 37: Grafica de los puntos 3D seguidos del objeto

### 3.12 Resultados

El seguimiento realizado al objeto ayuda en gran medida a la reconstrucción de la escena 3D sin embargo se producen errores de trayectoria del objeto seguido, al hacer dicha reconstrucción, los errores están enmarcados en círculos enumerados en la **Figura 38**.



*Figura 38: Errores de la reconstrucción 3D*

Se puede notar que en el círculo que tiene el número 1 que es el momento exacto donde se realiza el giro se produce un error mínimo, en el círculo 2 donde se terminó el giro también hubo un error, el círculo con el número 3 fue donde se realizó otro giro y el círculo con el número 4 debió ser una línea recta y no presentar esa punta mostrada, pero eso se debió al error estimado de 2 cm. Por lo tanto, el resultado de este trabajo consta de un seguimiento de objetos muy eficaz y con bajo coste computacional, en cuanto a la precisión de los puntos 3D se tienen en cuenta los siguientes factores:

1. La calibración estereoscópica, ya que esta ayuda a disminuir el error
2. La velocidad con la cual el objeto se mueve.
3. El uso del filtro WLS, que ayuda a tener una disparidad del objeto más aproximada a la real ya que no abra tanto ruido en la imagen.

#### 4 Descripción detallada del prototipo de visión artificial para el seguimiento de objetos desarrollado

Iniciamos haciendo la corrección de la distorsión de las cámaras, las imágenes utilizadas son las que fueron tomadas por el programa "photo\_booth.py". Esta calibración se basa en `Imgpoints` y los puntos de la posición de las esquinas del tablero de ajedrez. La función `cv2.calibrateCamera ()` se usa para crear nuevas matrices de cámara (describe la proyección de un punto del mundo 3D en una imagen 2D), la distorsión los coeficientes, los vectores de rotación y traslación se obtienen para cada cámara los cuales se necesitan más tarde para eliminar la distorsión de cualquier cámara. Para optimizar las matrices de las cámaras se utiliza la función `cv2.getOptimalNewCameraMatrix ()` (aumentando la precisión).

Se prosigue a realizar la calibración estéreo, para ello se utiliza la función `cv2.StereoCalibrate ()`, que calcula la transformación entre ambas cámaras (una cámara sirve como referencia para la otra). La función `cv2.stereoRectify ()` permite que las líneas epipolares de ambas cámaras estén activas para traer el mismo nivel. Esta transformación facilita el trabajo para la función de disparidad creada, con esta característica se obtiene la matriz esencial y la matriz fundamental necesarias para la función de disparidad.

La función `cv2.initUndistortRectifyMap ()` devuelve una imagen que no tiene distorsión. Estas fotos luego se utilizan en el cálculo del mapa de disparidad.

Se prosigue a crear la instancia que será la que siga a nuestro objeto, para ello se utiliza la función `cv2.MultiTracker_create()`. Esta clase se usa para rastrear múltiples objetos usando el algoritmo de rastreador especificado. El `MultiTracker` es una implementación del seguimiento de múltiples objetos. Procesa los objetos rastreados independientemente sin ninguna optimización entre los objetos rastreados.

Al terminar el proceso anterior se comienza a iniciar las cámaras, para ello se utiliza la función `cv2.VideoCapture()` que es el llamado inicial a las cámaras, después de hacer el llamado de las cámaras se comienza a capturar la información capturando los frames con la función `read()`.

El paso siguiente es calcular la tarjeta de disparidad, para encontrar la disparidad, las coordenadas del bloque de coincidencia se convierten en el bloque de referencia rastreado, entonces se le aplica el valor absoluto al resultado, y cuanto mayor sea este valor más cerca de la cámara estéreo está el objeto. El programa utiliza imágenes en blanco y negro calibradas para calcular el mapa de disparidad, También es posible trabajar con imágenes BGR, pero eso tomaría más tiempo para una computadora media. El cálculo de la tarjeta se realiza con un método de `OpenCV`, `cv2.StereoSGBM_create ()`. `compute ()`.

después de inicializar la tarjeta de disparidad el paso que sigue es aplicar el filtro WLS (mínimos cuadrados ponderados), ya que el resultado que da al aplicar la tarjeta de disparidad no es malo, pero sigue siendo muy difícil obtener los bordes de los objetos debido al ruido, por lo que se utiliza un filtro WLS. En primer lugar, los parámetros del filtro se establecen en la inicialización.

Lambda generalmente se establece en 8000, cuanto mayor es este valor, las formas de la tarjeta de disparidad aumentan, la tarjeta de disparidad adjunta los formularios de la imagen de referencia, la cual tenemos establecida en 8000 porque hubo mejores resultados con este valor. Sigma describe qué tan fuerte es nuestro filtro en los bordes de los objetos para ser más preciso.

Se crea otro objeto estéreo usando `cv2.ximgproc.createRightMatcher()`. Estas dos instancias se utilizan en el filtro WLS para uno crear la tarjeta de disparidad.

Se crea una instancia del filtro utilizando la función `cv2.ximgproc.createDisparityWLSFilter()` Producida.

Para aplicar la instancia del filtro WLS, se llama al siguiente método `cv2.ximgproc.createRightMatcher().filter()`, entonces los valores de nuestro filtro serán normalizados con `cv2.normalize()`.

Se utilizó `cv2.ColorMap_Ocean` para obtener una mejor visualización con `cv2.applyColorMap()`. Cuanto más oscuro sea el color, más lejos estará nuestro objeto de la cámara estéreo.

La próxima instancia es seleccionar el cuadro delimitador, que dará comienzo al seguimiento de nuestro objeto esto se hace a través de la función `cv2.selectROI()` que crea una ventana y permite al usuario seleccionar un ROI con el ratón.

Al terminar la selección del objeto se crea un nuevo rastreador de objetos para el cuadro delimitador seleccionado y proseguir a agréguelo al rastreador de objetos múltiples creado anteriormente.

Al tener ya definida la región delimitadora se prosigue a recorrer los cuadros delimitadores para dibujarlos e ir actualizando las coordenadas del centroide, para dibujar los cuadros primero tenemos que saber las coordenadas de este, para ello se utiliza la función `box.astype()` y lo dibujamos con la función `cv2.rectangle()` después de dibujarlo se utiliza la función `ct.update()` para ingresar las coordenadas de ese cuadro delimitador a la función `centroide` la cual retorna el punto medio del cuadro delimitador

Por último, se comienza a medir la distancia a la que está un objeto, una vez que se ha generado el mapa de disparidad, se debe determinar la distancia. El trabajo es encontrar la relación entre el valor de disparidad y la distancia. Esto se hizo posible gracias a que se tomaron valores de disparidad experimentalmente al elegir puntos en varios lugares y hacer una regresión lineal, al tener la ecuación de la distancia se envían las coordenadas (x,y) del centroide a la función `coords_mouse_disp()` la cual retorna la distancia

Teniendo la distancia del objeto Z se pasa a calcular las coordenadas X, Y con las fórmulas (2) y (3) mostradas anteriormente para cada instante de tiempo, para poder realizar la reconstrucción de la escena 3D del objeto.

Donde  $X_{ccc}$  son las coordenadas del objeto del eje X y  $Y_{ccc}$  son las coordenadas del objeto del eje Y.

Para exportar los datos obtenidos para realizar la reconstrucción 3D se utilizó la librería `openpyxl`, primeramente importamos la librería, se prosigue con la creación del objeto `workbook`, especificamos la ruta donde guardaremos el archivo, en este caso se trabajó en mi pc y se utilizó la ruta que convenía más en ese momento, al utilizar el programa en otro computador se deberá cambiar la ruta donde se guardara dicho archivo, después de tener la ruta establecida se crea el objeto `load_workbook` y como parámetro se le pasa la ruta creada anteriormente y por ultimo seleccionamos el archivo.

Para guardar los datos en el archivo se utiliza la instrucción `sheet.append`, se utiliza el for anidado para ir pasando dato por dato del diccionario donde los datos fueron almacenados.

La última instrucción para guardar el archivo con los respectivos cambios es `wb.save(filepath)`

Quedará un archivo en la ruta seleccionada con el respectivo nombre puesto

Dentro de este se almacenarán los datos de cada objeto seguido con la siguiente correspondencia de columnas, la primera columna representara al eje X, la segunda columna representara al eje Y, la tercera y última columna representara al eje Z. el seguimiento de cada objeto está separado por una serie de columnas en 0. Así se es más fácil diferenciar el seguimiento del objeto uno con los demás objetos seguidos.

0	0	0
19	1,36	0,92
19,41	1,39	0,94
17,57	1,24	0,84
18,4	0,84	0,88
19,21	1,37	0,93
19	1,36	0,92
19,21	1,13	0,93
17,35	1,24	0,84
17,56	1,47	0,85
19,41	1,14	0,94
19,03	1,34	0,91
17,78	1,25	0,85
17,57	1,24	0,84
18,82	1,09	0,9
18,19	1,28	0,87
18,19	1,28	0,87
19,66	1,39	0,94
17,78	1,25	0,85
19,66	1,39	0,94
18,19	1,28	0,87
0	0	0
91,27	61,45	2,26
92,08	61,99	2,28

*Ilustración 2: Datos recogidos del seguimiento 3D del objeto*



#### 4.1 Arquitectura del sistema realizado

Este proyecto se subdivide en diferentes módulos los cuales ejecutan diferentes tareas, cada una tienen una función específica la cual es la estimación de distancias de objetos mediante el uso de un sistema de visión estéreo. La Ilustración 2 muestra un diagrama de bloques del sistema, en el que se da una idea de cómo interactúan los diferentes procesos del mismo.

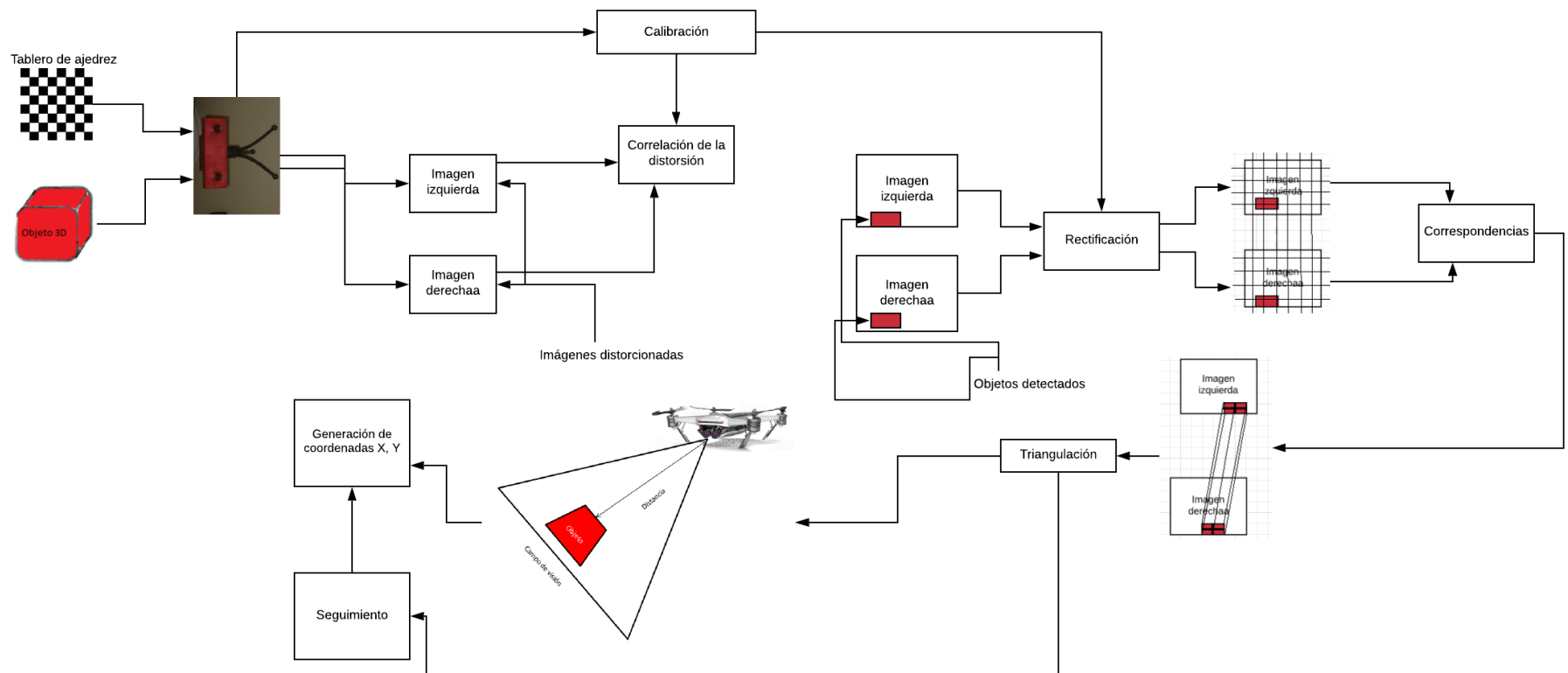


Ilustración 3 Arquitectura de tareas que se realizan durante todo el proceso

### **Disposición de las cámaras**

El primer paso para seguir el diagrama de bloques es la disposición en la que se pondrán las cámaras. Siendo esta una aplicación con ejes ópticos paralelos, las cámaras deberán quedar con sus ejes ópticos lo más paralelo posible. De tal forma que la escena pueda ser grabada por ambas cámaras, y determinar la distancia entre estas ya que es un parámetro que se necesitara para determinar la distancia.

Determinada la disposición de las cámaras se prosigue a la calibración estereoscópica de las cámaras.

### **Correlación de la distorsión.**

Se tomaron fotos de un patrón de un tablero de ajedrez como lo muestra el diagrama de bloques, este patrón fue obtenido por ambas cámaras y se encuentran los puntos principales del tablero, es decir las esquinas interiores del tablero, lo que hace es relacionar cada punto real en ambas imágenes

Una vez identificados los puntos principales en ambas imágenes, se podrán obtener los parámetros extrínsecos e intrínsecos de las cámaras y los vectores de distorsión.

### **Rectificación estéreo**

Para la rectificación estéreo se utilizan los parámetros extrínsecos e intrínsecos y los vectores de distorsión obtenidos anteriormente para proyectar las dos imágenes y se encuentren exactamente en el mismo plano alineando con precisión las filas de píxeles para que las líneas epipolares se vuelvan horizontales. Para encontrar la correspondencia de un punto más al azar en las dos imágenes

### **Búsqueda de correspondencia**

Este paso es un paso anterior a la estimación de distancias toma las imágenes rectificadas y busca la correspondencia de puntos entre ellas, obteniendo un mapa de disparidad. El objeto SGBM generado compara los bloques de una imagen de referencia con los bloques de la imagen coincidente. Si la calibración estéreo se ha hizo bien, por ejemplo, un bloque de la fila cuatro de la imagen de referencia debe compararse con todos los bloques de la imagen coincidente que están solo en la cuarta fila. De esta manera, el cálculo de la tarjeta de disparidad se vuelve más eficiente.

El programa utiliza imágenes rectificadas en blanco y negro para calcular el mapa de disparidad.

Después de calcular la tarjeta de disparidad se aplicó el filtro WLS (mínimos cuadrados ponderados), ya que el resultado que da la tarjeta de disparidad no es malo, pero sigue siendo muy difícil obtener los bordes de los objetos debido al ruido, por lo que se utiliza un filtro WLS.

### **Seguimiento. Selección del cuadro delimitante**

Se selecciona el objeto al que se desea seguir por un cuadro delimitador y se prosigue a agréguelo al rastreador de objetos múltiples creado anteriormente.

Al tener ya definida la región delimitadora se prosigue a recorrer los cuadros delimitadores para dibujarlos e ir actualizando las coordenadas del centroide, se utiliza una clase centroide para ingresar las coordenadas de ese cuadro delimitador el cual retorna el punto medio del cuadro delimitador

### **Triangulación.**

Se comienza a medir la distancia a la que está un objeto, una vez que se ha generado el mapa de disparidad, se debe determinar la distancia. Se encontró la relación entre el valor de disparidad y la distancia. Esto se hizo posible gracias a que se tomaron valores de disparidad experimentalmente al elegir puntos en varios lugares y hacer una regresión lineal, al tener la ecuación de la distancia se envían las coordenadas (x,y) del centroide a la función `coords_mouse_disp()` la cual retorna la distancia.

### **Generación de coordenadas X, Y**

Ya teniendo la distancia calculada se aplicaron las fórmulas que aparecen en el índice 3.7, ya que se tienen todos los parámetros correspondientes al punto principal ( $X_o, Y_o$ ) y las coordenadas del objeto de interés en cada instante de tiempo de la cámara izquierda ( $X_i, Y_i$ )

## **4.2 ¿Por qué utilizar Opencv?**

Se utiliza OpenCV en este proyecto ya que el proyecto tiene como meta el proveer un marco de desarrollo fácil de utilizar y altamente eficiente, ya que la librería puede tomar ventaja del procesamiento multi-núcleo es una razón de utilizarla. Otra ventaja que ofrece OpenCV es que es multiplataforma, Existiendo versiones para GNU/Linux, Mac OSX y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos, reconocimiento facial, calibración de cámaras, visión estéreo y visión robótica.

Las OpenCV además, cuentan a su disposición con dos funciones que tienen gran importancia para el correcto funcionamiento del proyecto:

**cv2.StereoSGBM\_create (). compute ().**

Esta clase se utiliza para el cálculo de la tarjeta de disparidad.

**cv2.MultiTracker\_create()**

Esta clase se usa para rastrear múltiples objetos usando el algoritmo de rastreador especificado

### 4.3 Algoritmos utilizados

Basado en la información de la Tabla 1. El porcentaje de efectividad de los algoritmos varía significativamente, se puede notar que los códigos que presentaron una mejor efectividad al momento de realizar estas pruebas fueron:

- CSRT tracker
- Color tracker
- KCF tracker

Además de realizar esta prueba, se estudiaron las ventajas y desventajas de todos los algoritmos propuestos en las Tabla 1 y Tabla 2, dicho estudio fue una clave para la selección de los algoritmos que pasarían a la prueba de seguimiento. De los cuales después de ser estudiados y analizando las ventajas y desventajas que proporcionaría cada uno de los algoritmos, se decidió que los más convenientes para el desarrollo del proyecto serían los siguientes:

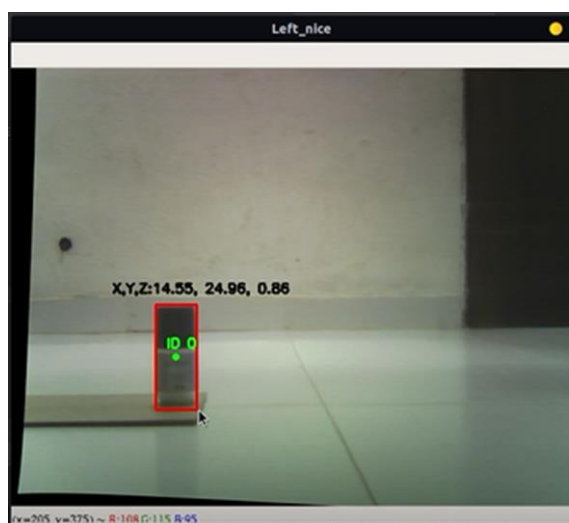
- CSRT tracker
- Color tracker

Ya que al hacer una comparativa entre el algoritmo KCF contra el algoritmo CSRT tanto de las ventajas y desventajas que pueden ofrecer y analizar el resultado del estudio que se hicieron en las tablas anteriormente mencionadas, el algoritmo CSRT tiene mejores resultados que el algoritmo KCF.

Por otra parte, para el modelo de visión estéreo se escogió el modelo de caja negra ya que este se basa en la generación de mapas 3D con estructuras parametrizadas donde hay unas entradas y estas generan una salida, es decir no se precisa conocer ni definir los detalles de su funcionamiento interno

#### 4.4 Pruebas Finales

Para validar el funcionamiento del código se realizó una prueba básica para verificar el funcionamiento de este en condiciones variantes. La prueba consta de realizar el seguimiento de un objeto determinando sus distancias cuando el objeto está en movimiento en condiciones lumínicas normales y el objeto no se sale de la región captada por la cámara.



*Ilustración 4: Prueba de calidad del software resultante del escenario (INICIO)*



*Ilustración 5: Prueba de calidad del software resultante del escenario (FINAL)*

Al finalizar el seguimiento de los objetos graficamos los datos obtenidos que quedaron guardados en un Excel en este caso llamado “demo.xlsx” y graficamos, el resultado que se obtuvo es mostrado en la Figura 39, Figura 40, Figura 41.

Este grafica representa el movimiento que hizo el objeto rastreado en el eje x, el seguimiento hecho tiene una variación de 3 a 2 centímetros respectivamente de la medida original que está dada por medio de la línea roja graficada en la respectiva Figura

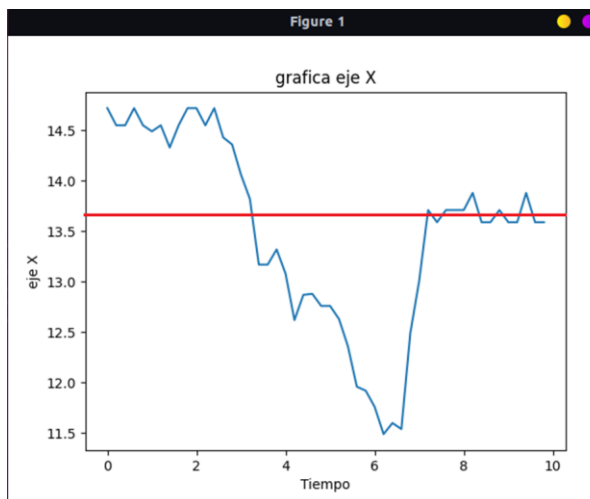


Figura 39: Grafica de los puntos 3D del eje X obtenidos

Este grafica representa el movimiento que hizo el objeto rastreado en el eje Y, se puede apreciar que el objeto mientras se va alejando va aumentando su trayectoria en el mismo eje, esto debido a que el objeto se va alejando.

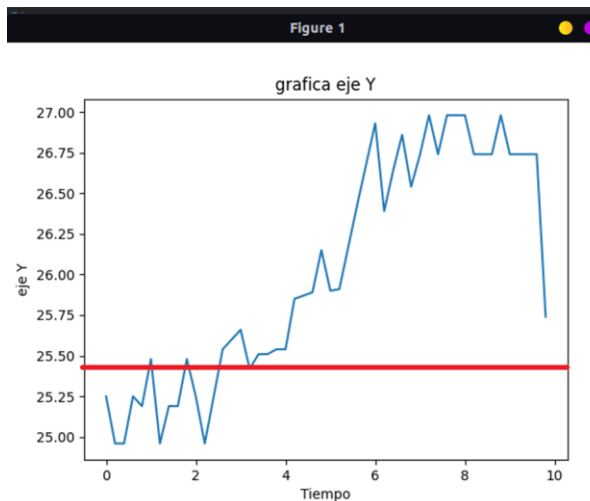


Figura 40: Grafica de los puntos 3D del eje Y obtenidos

Esta grafica representa la distancia del objeto al que se le realizo el seguimiento, se puede notar que esta distancia se va incrementando ya que se estaba moviendo el objeto.

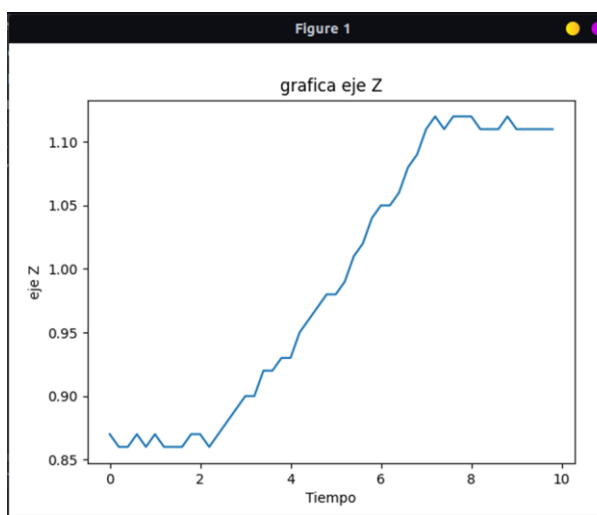


Figura 41: Grafica de los puntos 3D del eje Z obtenidos

Tiempo	Actual	Modelo	Error	Valor del error Absoluto	Error cuadrado
$t$	$A_t$	$F_t$	$A_t - F_t$	$ A_t - F_t $	$(A_t - F_t)^2$
0	25,25	25,25	0,000	0,000	0,000
1	25,32	25,5	-0,180	0,180	0,032
2	25,35	25	0,350	0,350	0,123
3	25,49	25,49	0,000	0,000	0,000
4	25,52	25,52	0,000	0,000	0,000
5	25,28	26,23	-0,950	0,950	0,902
6	26,32	26,89	-0,570	0,570	0,325
7	26,5	26,5	0,000	0,000	0,000
8	26,8	26,89	-0,090	0,090	0,008
9	26,89	26,89	0,000	0,000	0,000
10	26,89	25,75	1,140	1,140	1,300
	Total		-0,300	3,280	2,690
	$n$	11			
	Métrica	% de error			
	MAE	0,298			
	MSE	0,245			
	RMSE	0,495			

Ilustración 6 Porcentaje de error calculado por las métricas estudiadas en el eje Y

Tiempo	Actual	Modelo	Error	Valor del error Absoluto	Error cuadrado
$t$	$A_t$	$F_t$	$A_t - F_t$	$ A_t - F_t $	$(A_t - F_t)^2$
0	13,600	14,800	-1,200	1,200	1,440
1	13,600	14,400	-0,800	0,800	0,640
2	13,600	13,300	0,300	0,300	0,090
3	13,600	14,000	-0,400	0,400	0,160
4	13,600	13,400	0,200	0,200	0,040
5	14,000	12,280	1,720	1,720	2,958
6	14,000	11,500	2,500	2,500	6,250
7	14,000	13,600	0,400	0,400	0,160
8	14,800	13,700	1,100	1,100	1,210
9	13,600	13,700	-0,100	0,100	0,010
10	13,600	13,500	0,100	0,100	0,010
	Total		3,820	8,820	12,968
	n	11			
	Métrica	% de error			
	<u>MAE</u>	0,802			
	<u>MSE</u>	1,179			
	<u>RMSE</u>	1,086			

Ilustración 7 Porcentaje de error calculado por las métricas estudiadas en el eje X



Tiempo	Actual	Modelo	Error	Valor del error Absoluto	Error cuadrado
$t$	$A_t$	$F_t$	$A_t - F_t$	$ A_t - F_t $	$(A_t - F_t)^2$
0	0,860	0,860	0,000	0,000	0,000
1	0,860	0,870	-0,010	0,010	0,000
2	0,680	0,860	-0,180	0,180	0,032
3	0,900	0,900	0,000	0,000	0,000
4	0,950	0,950	0,000	0,000	0,000
5	1,000	1,000	0,000	0,000	0,000
6	1,050	1,050	0,000	0,000	0,000
7	1,100	1,120	-0,020	0,020	0,000
8	1,100	1,100	0,000	0,000	0,000
9	1,100	1,100	0,000	0,000	0,000
10	1,100	1,100	0,000	0,000	0,000
Total			-0,210	0,210	0,033
$n$		11			
Métrica		% de error			
<u>MAE</u>		0,019			
<u>MSE</u>		0,003			
<u>RMSE</u>		0,055			

Ilustración 8 Porcentaje de error calculado por las métricas estudiadas en el eje Z

#### 4.4.1 Resultados

- Las dimensiones están dadas en centímetros, la variación que se presenta en la Figura 40 se debe a que el objeto se va alejando y el objeto de interés se va haciendo más pequeño y debido a eso el centroide queda colocado a una altura mayor. Por otra parte se realizó una comparación de las métricas estudiadas para verificar cuanto porcentaje de error calculaba cada modelo y si un modelo tenía una diferencia mayor a los otros. Obteniendo estos resultados :

Porcentaje de error para el eje X:

- MAD 0,802
- MSE 1,179
- RMSE 1,086

Porcentaje de error para el eje Y

- MAD 0,298
- MSE 0,245

- RMSE 0,495

Porcentaje de error para el eje Z

- MAD 0,019
- MSE 0,003
- RMSE 0,055

Donde el mayor porcentaje de error fue del modelo MSE en el eje X y podemos observar que el eje que produce menos porcentaje de error es el eje Z el cual es el eje en el cual se está observando la distancia medida del objeto seguido.

#### 4.5 Error del modelo final con la métrica MAE

Métrica MAE	Eje X	Eje Y	Eje Z
Prueba 1	0,349	0,748	0,021
Prueba 2	0,325	0,823	0,012
Prueba 3	0,298	0,802	0,019
Prueba 4	0,273	0,866	0,034
Total error	0,31125	0,80975	0,0215

El porcentaje de error calculado para las pruebas finales tienen un valor mínimo, por lo tanto se puede concluir que los valores calculados por el modelo se aproximan a los valores del mundo real.

Se puede usar el modelo con el cual se sienta más cómodo, ya que los tres hacen lo mismo. El criterio MAE es un criterio de medida similar al RMSE y al MSE. Sin embargo, es más robusto ya que es menos sensible a los valores extremos que MSE Y RMSE. Todas las mediciones de distancia (MSE, RMSE y MAE) son equivalentes y ayudan a cuantificar la precisión de la solución aproximada en comparación con los datos simulados. Un valor pequeño para estos criterios significa que el modelo estimado está cerca del valor real.

## 4.6 Conclusiones

- Al hacer el respectivo análisis de los códigos de seguimiento para su elección se tuvieron en cuenta parámetros como, calidad de seguimiento, velocidad de ejecución, oclusión, entre otros. Se realizaron diferentes tipos de pruebas para validar el funcionamiento del mismo y medir el porcentaje de error de seguimiento que presenta este en diversos escenarios. Dichos resultados de las pruebas están disponibles en el capítulo 4.3, se estudiaron las ventajas y desventajas de cada algoritmo de seguimiento propuesto, el cual fue realizado en el capítulo 4.2, todo esto con el fin de seleccionar el algoritmo de seguimiento más adecuado y con menos porcentaje de error para posteriormente hacer la integración de los algoritmos.
- El estudio de los algoritmos de visión artificial nos llevó al algoritmo de hartley y el algoritmo de bouguet, el cual, al hacer una comparación entre estos dos algoritmos, se eligió el algoritmo bouguet por las ventajas que tiene sobre el algoritmo hartley, esto debido a que experimentalmente casi nunca los ejes ópticos quedan alineados o paralelos entre sí. Por lo que es una fuente que produce un pequeño porcentaje de error y esta ayuda de gran manera a la solución, además un factor clave es colocar las cámaras en una base estable para la adquisición del video y de las imágenes de calibración.
- Al hacer la respectiva integración de los algoritmos seleccionados, se tuvo que realizar diferentes pruebas para identificar los problemas que se presentan al momento de la unión de dichos algoritmos, esto con el fin de dar solución a los objetivos propuestos en este documento, esta unión dio como resultado un código de fácil implementación que permite observar el seguimiento de objetos en el espacio, detectando las coordenadas X, Y, Z del objeto, al momento de detectar las coordenadas mencionadas este tiene un porcentaje de error para cada coordenada, para la coordenada X el error total es de un 0,31125 %, para el eje Y es de 0,80875 % y para el eje Z es de 0,0215 % de error, este porcentaje se cumple mientras el objeto no presente una oclusión completa, las coordenadas quedaran guardadas en un documento Excel en la ruta especificada.
- Se diseñó una GUI para el algoritmo desarrollado, que permite un manejo sencillo del software el cual permite la modificación de parámetros afectando tanto las cámaras como el escenario en sí. Consta de 5 parámetros base que al momento de ejecutar el algoritmo estos parámetros estarán ajustados en los mejores valores que se tuvieron al momento de realizar pruebas y consultar en la documentación que se encuentra disponible en la red. Estos parámetros se podrán modificar si el usuario a si lo quiere en cualquier momento.

- Al realizar diferentes pruebas nos percatamos que la luminosidad afecta mínimamente los valores escogidos, ya que este crea un espacio entre los valores, en cuanto a la precisión de los puntos 3D se tienen en cuenta los siguientes factores: La calibración estereoscópica, ya que esta ayuda a disminuir el error y el uso del filtro WLS, que ayuda a tener una disparidad del objeto más aproximada a la real ya que no abra tanto ruido en la imagen. Al visualizar la generación de la escena 3D se observa que la silueta generada se asemeja a la trayectoria recorrida del objeto.

## 5 Bibliografía

- Ait-Amir, Bouzid, Philippe Pougnet, and Abdelkhalak El Hami. 2015. “Meta-Model Development.” *Embedded Mechatronic Systems 2*: 151–79. <https://doi.org/10.1016/b978-1-78548-014-0.50006-2>.
- Anthony G, BArnston. n.d. “Correspondence among the Correlation, RMSE, and Heidke Forecast Verification; Refinement of the Heidke Score.”
- Augusto Betancur Cruz, César. 2016. “UNIVERSIDAD AUTÓNOMA DE MADRID ESCUELA POLITÉCNICA SUPERIOR Seguimiento de Objetos Mediante Constricción Espacial de Puntos TILDE.”
- Avendaño Pérez, Jonathan, Jaime Alberto, Parra Plazas, and Jhon Fredy Bayona. n.d. “Universidad Antonio Nariño-Revista Facultades de Ingeniería.”
- Beainy, Camilo Ernesto Pardo, Edgar Andres Gutierrez Caceres, Fabian Rolando Jimenez Lopez, and Luis Fredy Sosa Quintero. 2013. “Disparity Map Generation, from the Use of Rectified Images.” In *Symposium of Signals, Images and Artificial Vision - 2013, STSIVA 2013*. <https://doi.org/10.1109/STSIVA.2013.6644908>.
- Camposeco, Federico, Cecilia Aviles, Benjamin Careaga, Leopoldo Spindola, and Ramiro Velazquez. 2011. “Constraints on Human Stereo Vision for Tele-Operation.” In *IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, 2011 IEEE*, 1–6. IEEE. <https://doi.org/10.1109/LARC.2011.6086805>.
- CEFOCCA-UNSJ. 2016. “Documento de Cátedra FOTOGRAFOMETRIA,” 1–15.
- Cuervo, Luisa Fernanda Gutiérrez, Felipe Calvo Salcedo Andrés, and Edwin Andrés Quintero Salazar. 2013. “Caracterización Del Ruido Presente En Las Imágenes Astronómicas Del Observatorio de La Universidad Tecnológica de Pereira.” *Scientia Et Technica* 18 (3): 575–82. <https://doi.org/10.22517/23447214.8663>.

- Du, Ji Xiang, Xiao Feng Wang, and Guo Jun Zhang. 2007. "Leaf Shape Based Plant Species Recognition." *Applied Mathematics and Computation* 185 (2): 883–93. <https://doi.org/10.1016/j.amc.2006.07.072>.
- EDMANS, grupo investigacion. 2006. "Técnicas y Algoritmos Básicos de Visión Artificial." "El Proceso de Investigación, Carlos Sabino." n.d.
- Felipe, Andrés, Luna Camacho, Nicolás Esteban, Rodríguez Menjura, Julián Armando, and Quiroga Sepúlveda. 2017. "Detección y Conteo de Personas En Espacios Cerrados Utilizando Estrategias Basadas En Visión Artificial." BOGOTÁ D.C.
- Fernandez-Carbajales Cañete, Víctor. 2018. "Atención Visual Basada En Un Espacio Perceptual Conjunto de Color y Brillo Para La Mejora de Algoritmos de Seguimiento de Objetos En Secuencias de Vídeo," 1.
- Fernández López, Carlos. 2016. "Road Scene Interpretation for Autonomous Navigation Fusing Stereo Vision and Digital Maps," 1.
- Freeman, William, and Steve Seitz. n.d. "Hoy • Modelo de Cámara • Calibración de Cámaras • Referencias."
- Fuente López, Eusebio de la., Félix Miguel. Trespaderne, and Universidad de Valladolid. 2012. *Visión Artificial Industrial: Procesamiento de Imágenes Para Inspección Automática y Robótica*. Universidad de Valladolid, Secretariado de Publicaciones e Intercambio Editorial.
- Gary, bandski , adrian, Kaehler. 2008. "Learning Opencv 3-O'Reilly." 2008. [https://e-archivo.uc3m.es/bitstream/handle/10016/10118/PFC\\_Javier\\_Yanez\\_Garcia.pdf?sequence=4](https://e-archivo.uc3m.es/bitstream/handle/10016/10118/PFC_Javier_Yanez_Garcia.pdf?sequence=4).
- Guerrero Hernández, José Miguel, Gonzalo Pajares Martinsanz, and María Guijarro Mata-García. n.d. "TÉCNICAS DE PROCESAMIENTO DE IMÁGENES ESTEREOSCÓPICAS."
- J. D. Falcón, J. González, and G. Ambrosio. 2004. "Detección de Olivos En Imágenes de Satélite de Alta Resolución. | Asociación Española de Teledetección." 2004. <http://www.aet.org.es/?q=revista21-1>.
- Jiang, Guo Quan, Cui Jun Zhao, and Yong Sheng Si. 2010. "A Machine Vision Based Crop Rows Detection for Agricultural Robots." In *2010 International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR 2010*, 114–18. <https://doi.org/10.1109/ICWAPR.2010.5576422>.
- Julián Aguirre de Mata, Autor. n.d. "UNIVERSIDAD POLITÉCNICA DE MADRID Calibración Geométrica de Cámaras No Métricas. Estudio de Metodologías y Modelos Matemáticos de Distorsión TESIS DOCTORAL."
- Leonardo, Jorge, and Hernández Fajardo. 2015a. "CIS1330IS08."
- . 2015b. "Seguimiento de Puntos En Imágenes Diagnósticas." BOGOTÁ, D.C. <http://pegasus.javeriana.edu.co/~CIS1330IS08>.

- López Yovera, Armando. 2018. "Reconstrucción 3D a Partir de Imágenes Rayos x de Frutos Con Semilla." *Repositorio Institucional USS*.
- Marinetti Carrillo, Eugenio. 2017. "Advanced Tracking and Image Registration Techniques for Intraoperative Radiation Therapy," 1.
- Mondragón, Iván F., Pascual Campoy, Miguel A. Olivares-Mendez, and Carol Martinez. 2011. "3D Object Following Based on Visual Information for Unmanned Aerial Vehicles." In *2011 IEEE 9th Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, LARC 2011 - Conference Proceedings*. <https://doi.org/10.1109/LARC.2011.6086794>.
- NeoAttack. n.d. "Tracking." <https://neoattack.com/neowiki/tracking/>.
- Ochoa, Jiménez, Magaly Gabriela, Ing Paz Arias, and Henry Patricio. 2015. "UNIVERSIDAD NACIONAL DE LOJA Área de La Energía, Las Industrias y Los Recursos Naturales No Renovables."
- Parra Escamilla, Alejandra. 2012. "Análisis de Movimiento En 3 Dimensiones En Video de Alta Velocidad Para Evaluación de Sistemas Dinámicos."
- Pastorelli, Giuliano. 2011. "Centro de Interpretación de La Naturaleza / Capilla Vallejo Arquitectos." <https://www.archdaily.pe/>. 2011.
- Patiño, Julián Alberto, Jairo J. Espinosa, and Rosa E. Correa. 2010. "A Comparison of Kalman-Based Schemes for Localization and Tracking in Sensor Systems." In *2010 IEEE Latin American Conference on Communications, LATINCOM 2010 - Conference Proceedings*. <https://doi.org/10.1109/LATINCOM.2010.5640986>.
- Portella Coaguila, Ana Gaby. 2015. "Diseño y Construcción de Un Dispositivo Para Realizar Reconstrucción Tridimensional Basado En Un Sistema Estéreo de Dos Cámaras." *Reponame:UNSA-Institucional*.
- Revollo Sarmiento, Natalia V, Marina P Cippolletti, Claudio Delrieux, Gerardo M Perillo, and M Cintia Piccolo. n.d. "Segmentación y Medición de Zonas Costeras En Secuencias de Videos."
- Rodríguez Guillén, Reinier. 2014. "Detección de Regiones de Interés En Imágenes de La Prueba de Papanicolaou," June.
- Romero, Cristian Felipe, and Juan Gabriel Viveros Albornoz. 2018. "DETECCIÓN DE VEHÍCULOS EMPLEANDO VISIÓN POR COMPUTADOR EN EL MARCO DEL PROYECTO COMVIDAS." Cali.
- Sammut, Claude, and Geoffrey I. Webb, eds. 2010. *Encyclopedia of Machine Learning*. *Encyclopedia of Machine Learning*. Boston, MA: Springer US. <https://doi.org/10.1007/978-0-387-30164-8>.
- Sánchez Asmat, José Alfredo. 2016. "Sistema de Reconocimiento y Seguimiento de Objetos En Tiempo Real a Través de Visión Artificial." *Instacron:UNITRU*.
- Sebastian Barberis Maestrando Juan Pablo Graffigna Director Roberto Isoardi Co-Director, Luis.

n.d. “APLICACIÓN DE UN SISTEMA AUTOMÁTICO DE PROCESAMIENTO DE IMÁGENES MÉDICAS BASADA EN ESTÁNDARES.”

SIMENS. n.d. “Detección de Objetos - Sistemas de Identificación - Siemens.”

Snavely, Noah, Ian Simon, Michael Goesele, Richard Szeliski, and Steven M. Seitz. 2010. “Scene Reconstruction and Visualization from Community Photo Collections.” In *Proceedings of the IEEE*, 98:1370–90. <https://doi.org/10.1109/JPROC.2010.2049330>.

toronto. n.d. “Detección de Contorno.” <https://www.cs.toronto.edu/~dmac/images/ProjectFiles/sraf/srafdoc/deteccion.html>.

Torrentegi, Unai Mujika, and Jon Azpiazu. n.d. “Tesis de Máster Reconstrucción Densa de Modelos Tridimensionales Utilizando Visión Artificial.” Accessed April 22, 2020. [www.ccia-kzaa.ehu.es](http://www.ccia-kzaa.ehu.es).

Unrweiller, Frederic, vujasinovic, Stephane. 2017. “Inhaltsverzeichnis.” *Stereo-Vision*, 1–30. <https://github.com/LearnTechWithUs/Stereo-Vision>.

Uribe Sáenz de Cámara, Alumno. n.d. “Fecha: 12 de Julio de 2018 DOCUMENTO-MEMORIA, METODOLOGÍA, ASPECTOS ECONÓMICOS Y CONCLUSIONES.”

Xiang, Y, A Alahi, S Savarese - Proceedings of the IEEE, and Undefined 2015. 2016. “Learning to Track: Online Multi-Object Tracking by Decision Making.” *Cv-Foundation.Org*, 9. [https://www.cv-foundation.org/openaccess/content\\_iccv\\_2015/html/Xiang\\_Learning\\_to\\_Track\\_ICCV\\_2015\\_paper.html](https://www.cv-foundation.org/openaccess/content_iccv_2015/html/Xiang_Learning_to_Track_ICCV_2015_paper.html).

Zelasco, José F., Diego Amin Dagum, Judith Donayo, and Teresa Arcomano. 2000. “Computer Vision in AUVs: Automatic Roto-Rectification of Stereo Images.” *Oceans Conference Record (IEEE)* 3: 2169–76. <https://doi.org/10.1109/OCEANS.2000.882255>.