

CENTRO DE INGENIERÍA Y DESARROLLO INDUSTRIAL

Seguimiento y estimación de posición de  
objetos en el plano utilizando cámaras Pan  
& Tilt

**TESIS**

QUE PARA OBTENER EL GRADO ACADÉMICO DE:

MAESTRO EN CIENCIA Y TECNOLOGÍA CON ESPECIALIDAD EN  
MECATRÓNICA

PRESENTA:

Ing. Edrei Reyes Santos.

ASESORES

Dr. Hugo Jiménez Hernández.

Dr. Leonardo Barriga Rodríguez.

SANTIAGO DE QUERÉTARO, QUERÉTARO,  
31 de agosto de 2016

---

# Dedicatoria

---

Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre mi corazón y mi agradecimiento.

Papá y Mamá

---

# Agradecimientos

---

A CIDESI, por su espacio y material prestado para el desarrollo de esta tesis.

A Conacyt, quien me ofreció su beca de maestría para mi manutención a lo largo del proyecto de investigación.

A mis asesores Dr. Hugo Jiménez Hernández, Dr. Leonardo Barriga Rodríguez y revisores Dra. Diana Margarita Córdova Esparza, MCyT. Juan Manuel García Huerta, por su paciencia, tiempo y dedicación este proyecto, así como a mi formación académica.

---

# Índice general

---

<b>1. Introducción.</b>	<b>1</b>
1.1. Antecedentes y estado del arte . . . . .	2
1.1.1. Antecedentes . . . . .	2
1.1.2. Estado del arte . . . . .	3
1.2. Planteamiento del problema . . . . .	5
1.3. Hipótesis . . . . .	6
1.4. Objetivos . . . . .	7
1.4.1. Objetivo general . . . . .	7
1.4.2. Objetivos particulares . . . . .	7
1.5. Justificación . . . . .	8
1.6. Alcances y limitaciones . . . . .	9
1.6.1. Alcances . . . . .	9
1.6.2. Limitaciones . . . . .	9
<b>2. Fundamento teórico.</b>	<b>11</b>
2.1. Visión por computadora . . . . .	11
2.2. Seguimiento de objetos . . . . .	11
2.3. Taxonomía de técnicas para seguimiento de objetos . . . . .	13
2.3.1. Correspondencia de puntos . . . . .	15
2.3.2. Seguimiento de núcleo . . . . .	15
2.3.3. Seguimiento de contornos o siluetas . . . . .	16

2.4. Cámaras Pan & Tilt (PT) . . . . .	20
2.5. Sistemas Multicámara . . . . .	21
2.5.1. Campo de visión activo . . . . .	22
2.5.2. Factores que afectan a los sistemas multicámara . . . . .	23
2.6. Detección de movimiento por diferencia de imágenes . . . . .	24
2.7. Proceso de filtrado . . . . .	26
2.7.1. Filtrado en el dominio espacial . . . . .	27
2.7.2. Filtrado en el dominio de la frecuencia . . . . .	28
2.8. Detección de bordes en una imagen . . . . .	28
2.9. Transformación distancia . . . . .	31
2.10. Contornos activos (Snakes) . . . . .	33
2.11. Método de Newton-Raphson . . . . .	38
2.12. Transformación de coordenadas . . . . .	40
2.12.1. Coordenadas polares en el plano . . . . .	40
2.12.2. Coordenadas cilíndricas en el espacio . . . . .	42
2.12.3. Coordenadas esféricas en el espacio . . . . .	44
2.13. Inversa de una Matriz . . . . .	45
2.13.1. Inversas bilaterales . . . . .	46
2.13.2. Pseudoinversa . . . . .	47
<b>3. Desarrollo Experimental</b>	<b>49</b>
3.1. Metodología . . . . .	49
3.2. Adquisición de imágenes . . . . .	50
3.3. Detección de movimiento . . . . .	51
3.4. Pre-procesamiento . . . . .	52
3.5. Cálculo de centroide del objeto de interés . . . . .	53
3.5.1. Transformada maxima extendida . . . . .	54
3.5.2. Etiquetado . . . . .	55
3.5.3. Algoritmo para cálculo de centroide . . . . .	56

3.6. Imposición del contorno activo (snake) . . . . .	57
3.6.1. Transformada distancia . . . . .	57
3.7. Seguimiento de objetos . . . . .	61
3.7.1. Centrado del objeto en la imagen . . . . .	62
3.8. Estimación de coordenadas bidimensionales del objeto . . . . .	65
<b>4. Resultados</b>	<b>69</b>
<b>5. Conclusiones y trabajo futuro</b>	<b>83</b>
5.1. Conclusiones . . . . .	83
5.2. Trabajos futuros . . . . .	85
<b>A. Validación de trayectorias</b>	<b>95</b>
<b>B. CAD del espacio de pruebas</b>	<b>104</b>
<b>C. Tablas de error y ECM</b>	<b>105</b>
<b>D. Hoja de datos Cámara Pan &amp; Tilt</b>	<b>107</b>

---

# Índice de figuras

---

1.1. Interfaz de software, para cálculo de distancia. . . . .	2
2.1. Centro de control de un sistema de vigilancia. . . . .	11
2.2. Estación de monitoreo de tráfico vehicular [1]. . . . .	12
2.3. Detección y seguimiento de objetos en un sistema multicámara [2]. . . . .	14
2.4. Taxonomía de técnicas para seguimiento de objetos [3]. . . . .	14
2.5. Seguimiento de objetos por correspondencia de puntos [3]. . . . .	15
2.6. Seguimiento de objetos usando una plantilla rectangular [4]. . . . .	16
2.7. Seguimiento de objetos por contorno [5]. . . . .	16
2.8. Seguimiento de objetos [5] . . . . .	17
2.9. Seguimiento de objetos a través de coincidencia de formas [6]. . . . .	18
2.10. Contornos deformables para seguimiento de objetos no rígidos [5]. . . . .	19
2.11. Cámara PTZ/IP. . . . .	20
2.12. Cámaras IP/PT en red. . . . .	21
2.13. Campo de visión de un arreglo multicámara. . . . .	22
2.14. Campo de visión activo de una cámara Pan & Tilt. . . . .	23
2.15. Campo de visión activo de un sistema multicámara Pan & Tilt. . . . .	23
2.16. Detección de movimiento por resta de imágenes. . . . .	25
2.17. Detección de movimiento con ruido en la imagen resultante. . . . .	25
2.18. filtrado de una imagen. . . . .	27
2.19. Filtrado en el dominio espacial [30]. . . . .	27
2.20. Filtrado en el dominio de la frecuencia . . . . .	28

2.21. Derivada en el eje x . . . . .	29
2.22. Derivada en el eje y . . . . .	30
2.23. Operador de Roberts para detección de bordes. . . . .	30
2.24. Operadores de Prewitt, Sobel y Frei-Chen. . . . .	30
2.25. Explicación grafica de la transformada distancia. . . . .	31
2.26. Transformada distancia . . . . .	31
2.27. Centroide de una figura geométrica. . . . .	33
2.28. Contorno deformable (Snake). . . . .	34
2.29. Minimización de la energía por algoritmo voraz. . . . .	38
2.30. Método de Newton-Raphson. . . . .	39
2.31. Representación de un punto en el plano cartesiano. . . . .	41
2.32. Representación polar de un punto en el plano cartesiano. . . . .	41
2.33. Correcta definición de la función arco tangente. . . . .	42
2.34. Coordenadas cilíndricas. . . . .	43
2.35. Coordenadas esféricas. . . . .	44
3.1. Metodología para el desarrollo del proyecto. . . . .	49
3.2. Adquisición de imagen desde MATLAB con comando CGI. . . . .	50
3.3. Detección de movimiento por diferencia de imágenes . . . . .	52
3.4. Detección de movimiento con ruido . . . . .	52
3.5. Filtro gaussiano aplicado a la detección de movimiento . . . . .	53
3.6. Imagen segmentada utilizando la transformada máxima extendida . . . . .	55
3.7. El centroide calculado se muestra en color amarillo . . . . .	56
3.8. Imposición del contorno activo . . . . .	57
3.9. Contorno activo en convergencia con el contorno de interés . . . . .	61
3.10. Objeto centrado en la imagen, usando algoritmo 7 . . . . .	65
3.11. Planteamiento para la estimación de posición del objeto . . . . .	66
4.1. Arreglo de cámaras propuesto. . . . .	69

4.2. Marco referencia con coordenadas locales. . . . .	70
4.3. Trayectoria en cuadro y en diente de sierra. . . . .	70
4.4. Ejemplo de trayectorias usadas durante la experimentación. . . . .	71
4.5. Trayectoria diagonal. . . . .	72
4.6. Gráfico de error en el eje $X$ de la trayectoria diagonal. . . . .	73
4.7. Gráfico de error en el eje $Y$ de la trayectoria diagonal. . . . .	73
4.8. Desviación estándar del error en el eje $X$ de la trayectoria diagonal. . . . .	74
4.9. Desviación estándar del error en el eje $Y$ de la trayectoria diagonal. . . . .	74
4.10. Trayectoria horizontal. . . . .	75
4.11. Trayectoria en "d". . . . .	76
4.12. Trayectoria en "V". . . . .	76
4.13. Trayectoria en sin forma geométrica. . . . .	77
4.14. Trayectoria en "m". . . . .	78
4.15. Trayectoria escalón. . . . .	78
4.16. Dispersión del error en el eje $x$ para las 30 trayectorias. . . . .	79
4.17. Dispersión del error en el eje $x$ para las 30 trayectorias. . . . .	79
4.18. Distribución de frecuencia del error en $x$ . . . . .	80
4.19. Distribución de frecuencia del error en $y$ . . . . .	80
4.20. Dispersión del error por pares de coordenadas. . . . .	81
4.21. Desviación estándar del error por pares de coordenadas. . . . .	81
A.1. Trayectoria en forma de casa. . . . .	95
A.2. líneas cruzadas. . . . .	95
A.3. Trayectoria en cuadro. . . . .	96
A.4. Diente de sierra. . . . .	96
A.5. escalones. . . . .	97
A.6. Trayectoria en forma de G. . . . .	97
A.7. Trayectoria en forma de herradura. . . . .	97
A.8. Hexágono. . . . .	98

A.9. Trayectoria en forma de L. . . . .	98
A.10. Medio hexágono. . . . .	98
A.11. Onda cuadrada. . . . .	99
A.12. Trayectoria en forma de pentágono. . . . .	99
A.13. Trayectoria en forma de raíz. . . . .	99
A.14. Rectángulo. . . . .	100
A.15. Rombo. . . . .	100
A.16. Trayectoria sin forma definida. . . . .	100
A.17. Trayectoria irregular. . . . .	101
A.18. Ruta de objeto en forma de S. . . . .	101
A.19. Trapecio. . . . .	101
A.20. Trayectoria con forma de triángulo. . . . .	102
A.21. Trayectoria A.20 invertida. . . . .	102
A.22. Ruta de objeto en forma de U. . . . .	102
A.23. Trayectoria en forma de V. . . . .	103
A.24. Línea recta vertical. . . . .	103
B.1. Espacio de pruebas sin fondo neutro. . . . .	104
B.2. Espacio de pruebas con fondo neutro. . . . .	104

---

# Índice de tablas

---

3.1. Rotación. . . . .	62
3.2. Ángulo de giro ( $\theta$ y $\phi$ ) . . . . .	62
4.1. Datos necesarios para la estimación de la posición . . . . .	71
4.2. Error de la trayectoria diagonal y su ECM . . . . .	75
4.3. Error de la trayectoria horizontal, y su ECM . . . . .	75
4.4. Tabla de Error y ECM. Trayectoria en "d" . . . . .	76
4.5. Error y ECM. Trayectoria en "V" . . . . .	77
4.6. Error y ECM. Trayectoria irregular . . . . .	77
4.7. Error y ECM. Trayectoria en "m" . . . . .	78
4.8. Error de la trayectoria escalón, y su ECM . . . . .	78

# INTRODUCCIÓN.

---

Las ciudades en crecimiento hacen difícil controlar y analizar la información de movimiento o desplazamiento de objetos y personas para la toma de decisiones. Para hacer frente a esta situación, la tecnología toma un papel importante como una herramienta para establecer estrategias, identificar y manejar circunstancias que representan información relevante. Los sensores son cada vez más confiables y económicos, lo que hace factible implementar soluciones en ciudades saturadas. Mientras que la tecnología se vuelve más asequible, el desarrollo la infraestructura requerida para analizar grandes cantidades de datos en un periodo de tiempo corto, se convierte en un reto.

Una tarea típica en vigilancia y monitoreo, es el seguimiento de objetos, no obstante esta tarea se ve afectada por diferentes factores como: perspectiva de la cámara, afectaciones de color, ausencia de sincronía en la transmisión etc.; lo cual hace imposible el seguimiento de objetos en escenarios exteriores.

Algunos trabajos como los de A. Yilmaz, J. Fernández Gutiérrez y D. P. Chau, F. [7, 8, 9] proponen estrategias para un seguimiento de objetos fiable en escenarios al aire libre; aun así se observa que no utilizan cámaras con rotación horizontal y vertical. Este trabajo propone una estrategia para el seguimiento de objetos y la estimación de posición en dos dimensiones utilizando un conjunto de cámaras-PT (Pan & Tilt), la base de este enfoque es realizar seguimiento dinámico del objeto de interés con cada una de las cámaras en el arreglo.

Los parámetros Pan & Tilt de cada cámara se aprecian como componentes vectoriales en coordenadas polares. La distancia entre una cámara específica y el objeto de interés es representada por el radio, de tal forma que si se utilizan los mismos parámetros para todas las cámaras en el arreglo, se genera un sistema lineal, donde los valores desconocidos

representan la ubicación del objeto en el plano utilizando coordenadas locales. Más allá de este punto, el proceso se considera como una tarea de optimización cuyo objetivo es reducir el error de estimación de la distancia entre el arreglo de cámaras-PT (Pan & Tilt) y el objeto.

Finalmente se realiza un análisis experimental que consiste en la validación de las trayectorias artificiales creadas con anterioridad, en un escenario controlado, comparando la posición real del objeto con la estimada por el enfoque propuesto.

## 1.1. Antecedentes y estado del arte

### 1.1.1. Antecedentes

En el año 2014 en el centro de ingeniería y desarrollo industrial, en Querétaro, México [10] desarrolló un sistema de visión capaz de detectar un objeto en un escenario utilizando la técnica de marcas pasivas, es decir se utilizó un color en particular como una manera de distinguir al objeto en las imágenes obtenidas por la cámara. Este sistema calcula la distancia en centímetros que existe entre la cámara y el objeto de interés basándose en la altura en pixeles de los objetos, ya que ésta varía dependiendo de la distancia a la que se encuentren los objetos, además de seguir al objeto en un campo de visión dinámico, gracias a que utiliza cámaras Pan & Tilt, lo que le permite seguir al objeto mientras éste se encuentre en su rango de movimiento.

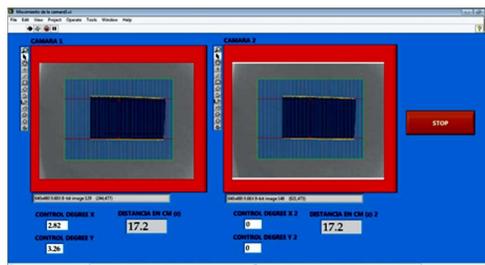


Figura 1.1: Interfaz de software, para cálculo de distancia.

### 1.1.2. Estado del arte

Diferentes aplicaciones de visión por computadora se han implementado para la detección, localización y seguimiento de objetos. En el año 2000, N. Paragios y R. Deriche, [11] propusieron un nuevo marco variacional para la detección y seguimiento de múltiples objetos en movimiento en secuencias de imágenes. La detección de movimiento se realizó utilizando un marco estadístico que se aproxima a la función de diferencia de densidad entre tramas observadas utilizando un modelo de mezcla. Esta función se minimiza utilizando contornos activos, lo que permite a su vez dar seguimiento a los objetos en movimiento. Este enfoque hace uso de una cámara fija permitiendo discriminar de forma relativamente sencilla el movimiento de fondo y el movimiento del objeto de interés.

A. K. Mishra, B. Ni, S. Winkler, y A. Kassim [12], implementaron un sistema de vigilancia 3D, en el año 2007 utilizando múltiples cámaras que rodean la escena, para la identificación de los seres humanos en la escena y luego la identificación de sus posturas, utilizando cámaras calibradas, colocadas en una posición fija. La detección de objetos y la interpretación se realizó en tres dimensiones, usando la información de profundidad, para separar a las personas del fondo e identificar su postura por la combinación con las plantillas del modelo 3D.

Por su parte R. Mohedano, Jaureguizar, L. Salgado y N. García, en 2008 [13] desarrollaron un método robusto para el seguimiento y posicionamiento tridimensional de personas, semi solapando imágenes de varias cámaras que observaban el mismo escenario, convirtiéndolo en un sistema capaz de responder a posibles oclusiones, las regiones en movimiento se dividieron en segmentos de forma independiente en cada cámara por medio de una estrategia de modelado de fondo basado en filtros de Gabor, la detección de personas se llevó a cabo a través de una estrategia de correlación con una plantilla, y por último el posicionamiento de las personas se realizó mediante la posición de la cabeza y no con el centroide del objeto.

También en 2009, S. Robla, J. y E. Sarabia [14] presentaron una herramienta para el seguimiento de derrames de petróleo, utilizando imágenes de satélite. Esta herramienta se

basa en contornos activos que se adaptan a las manchas de petróleo para determinar la forma y la localización del derrame de petróleo, incluso proporcionar una estimación de la deformación y el desplazamiento que podría sufrir en el tiempo, para rastrear los residuos vertidos en el mar. De igual manera en 2010, C. Losada, M. y M. Marrón [15] propusieron un sistema multicámara para segmentación, y localización de múltiples robots en un espacio controlado, utilizando de forma complementaria múltiples sensores colocados en los robots, para comparar las trayectorias estimadas. Este sistema, utilizó contornos activos para la segmentación de la escena, y seguimiento de los robots con cámaras fijas. Finalmente proporciona la ubicación de los robots en coordenadas de imagen.

En el año 2011, H. L. Paredes [16] implementó un sistema para la detección de objetos en movimiento en sistemas que usen cámaras móviles alineando las imágenes para posteriormente detectar las regiones de movimiento mediante el módulo de segmentación reduciendo el problema de detectar movimiento al de cámara estática. Las regiones de movimiento detectadas por el segmentador son analizadas por un algoritmo de seguimiento que trata de establecer una coherencia espacial y temporal de las distintas regiones detectadas para validarlas como objetos y reducir los errores que puedan aparecer por el ruido en el alineamiento.

En 2011, A. Utasi y C. Benedek [17] propusieron un enfoque probabilístico para la localización de múltiples personas utilizando múltiples cámaras calibradas, la característica del enfoque propuesto a nivel de píxeles se basa en las propiedades físicas del proceso de formación de la imagen 2-D y puede localizar con precisión la posición de las piernas en el plano de apoyo y estimar la altura de las personas.

Por su parte, Y. Lee y A. Yilmaz [18] en 2013 aportaron al estado del arte un sistema de posicionamiento de objetos en tiempo real que combina la detección y seguimiento, con el posicionamiento 3D en un múltiples cámaras sin movimiento de traslación. Se consideraron restricciones geométricas definidas entre varias cámaras, para el seguimiento de un objeto que contiene un marcador para su identificación.

En el año 2014, L. Maddalena, A. Petrosino, y F. Russo [9] desarrollaron un sistema que,

basándose en la información recogida por varias cámaras, es capaz de realizar conteo de personas. Esta información se recopila en una estructura sinérgica, que explota la homografía asociada a cada vista de la escena, apoyada por un mecanismo para el seguimiento de las personas y de sus identidades a lo largo del tiempo.

En 2014, S. J. You, P. y H. Truong [19] difundieron un sistema en tiempo real para el seguimiento de un objeto en movimiento rápido utilizando múltiples cámaras web adyacentes. En este sistema se utilizaron tres cámaras, que dentro del sistema de comunicación, desempeñan el rol de clientes, además se tiene un equipo de adquisición de imágenes, catalogado como servidor, quien a partir de la información de los equipos cliente, calcula las posiciones relativas del objeto en cada captura de imágenes, obteniendo así la velocidad de objetos en tiempo real. En 2014, S. Ballesta, G. Reymond, M. Pozzobon, y J. Duhamel [20] presentaron un sistema de seguimiento a grupos de primates, empleando una herramienta multicámara para evitar posibles oclusiones, y entregando la posición tridimensional de los primates, haciendo uso de collares de colores como marcas activas para detectar el movimiento en la escena; dotando al sistema con la capacidad de realizar el seguimiento de los primates en tiempo real.

Por su parte en el año 2015, Z. Jin y B. Bhanu, [7] propusieron seguimiento de múltiples cámaras con simulación de multitudes integrada, con el fin de explorar la posibilidad de hacer que la información homográfica sea mucho mas útil y para investigar la influencia de la simulación en el rendimiento de sistemas dedicados al seguimiento de objetos.

## 1.2. Planteamiento del problema

Los parámetros Pan & Tilt, de un conjunto de cámaras, proporcionan información que localmente ayuda en la estimación y ubicación de un objeto, utilizando técnicas de visión por computadora que trabajen de manera complementaria con dichos parámetros. Este planteamiento resulta útil si se tiene un método eficiente de seguimiento y detección del objeto, debido a que estos métodos forman parte esencial, en el proceso de estimación de

posición en un escenario.

Por otra parte la precisión y exactitud en la estimación de posición dependerá de las características físicas de las cámaras, pues tanto la velocidad como el paso mínimo de los motores; por mencionar algunas características, representan un factor influyente en el desempeño del procedimiento mencionado.

De acuerdo con la bibliografía consultada en el estado del arte, existen sistemas de visión que proporcionan información como la ubicación bidimensional del objeto y a su vez la trayectoria del mismo, pero no utilizan cámaras con movimiento horizontal y vertical, y en los trabajos en donde se utilizan, solamente se hace uso de una cámara Pan & Tilt dentro de todo el arreglo de cámaras, descartando la flexibilidad de utilizar un arreglo compuesto solamente de cámaras Pan & Tilt (PT).

En este trabajo se pretende mostrar un método en donde sólo se empleen estos parámetros para la estimación de posición en el plano de un objeto. En una primera aproximación el error de calibración es descartado, pero en situaciones donde se requiera repetibilidad y exactitud será necesario este proceso, para éste caso sólo se aborda el problema de estimar la posición a partir de los parámetros de giro de una cámara PT. El método propuesto presenta un enfoque que resulta viable en situaciones donde la geometría del escenario es desconocida y se tiene una referencia global, con lo que se conoce la posición física de cada una de las cámaras.

### 1.3. Hipótesis

Si se poseen diferentes proyecciones de un mismo objeto en un mismo instante, obtenidas mediante un conjunto de cámaras con la misma referencia y con conocimiento previo de la ubicación física de las mismas, es posible determinar la posición bidimensional  $(x, y)$  a partir de los parámetros PT  $(\theta, \phi)$  del arreglo de cámaras que observan al objeto.

## 1.4. Objetivos

### 1.4.1. Objetivo general

El objetivo general de este proyecto es el seguimiento de un objeto dentro del campo de visión dinámico de un arreglo de cámaras PT y determinar sus coordenadas bidimensionales a partir de los parámetros PT  $(\theta, \phi)$  del arreglo de cámaras que observa al objeto.

### 1.4.2. Objetivos particulares

- Detectar al objeto por diferencia de imágenes utilizando una cámara en un entorno controlado, procediendo a extender este mismo método de detección a todo el arreglo de cámaras.
- Realizar un arreglo de cámaras sobre una mesa óptica graduada, utilizando las perforaciones de la mesa como un marco de referencia, y determinando el punto de origen, así como los ejes  $xy$
- Sincronizar el arreglo de cámaras, a través de un sistema de comunicación compuesto, por un router y un equipo de computo.
- Calcular el centroide del objeto por contornos activos adaptables al objeto de interés, haciendo uso de los puntos del contorno activo para el cálculo de dicho centroide.
- Centrar el objeto en el campo de visión del arreglo de cámaras, mediante la ejecución de comandos CGI que dictaminan el movimiento horizontal y vertical de cada cámara.
- Realizar un modelo matemático, que parte de los parámetros PT del arreglo de cámaras que observan al objeto, para determinar las coordenadas en 2D del objeto.

## 1.5. Justificación

La detección de movimiento, localización y seguimiento de objetos son cuestiones importantes en aplicaciones de visión por computadora; puesto que estas tareas representan un reto en escenas al aire libre o en lugares concurridos, debido a que las condiciones de iluminación no pueden ser controladas, existe movimiento de fondo irrelevante y se presentan oclusiones causadas por diversos objetos en movimiento o estáticos [21]. Debido a esto los algoritmos de visión generalmente siguen un enfoque secuencial, partiendo de la segmentación del escenario, ya que proporciona características útiles para interpretar la información obtenida a partir de imágenes del mismo, hasta llegar al análisis a nivel del objeto de interés, una mejora directa a este enfoque es utilizar imágenes de múltiples cámaras-PT colocadas en puntos diferentes con lo cual el sistema se vuelve capaz hacer frente a las distintas condiciones mencionadas anteriormente.

Se puede comprender entonces que la detección de objetos, localización y seguimiento de los mismos, es muy importante en el campo de la visión artificial [19], sin embargo el realizar estas tareas con un arreglo de cámaras PT, representa un reto al momento de determinar que objetos pertenecen al fondo de una escena y separarlo del movimiento del objeto de interés, debido a que la rotación de las cámaras proporciona la impresión de que todo esta en movimiento en el escenario, aun los objetos estáticos, por lo que los métodos mas comunes para el modelado de fondo en un escenario, quedarían completamente inhabilitados ante esta situación. Es por eso que en este trabajo se propone la detección de movimiento utilizando la diferencia de dos imágenes, además de la implementación de un modelo físico de fondo neutro, que aísle, el movimiento externo al escenario.

A partir de lo señalado anteriormente con respecto a la propuesta de utilizar múltiples cámaras podemos concluir que es mucho más versátil realizar el seguimiento y determinar las coordenadas bidimensionales  $(x, y)$  de un objeto en una forma no invasiva, empleando redundancia de sensores (cámaras-PT) que proporcionan proyecciones bidimensionales y utilizando una referencia global para todos los sensores.

## 1.6. Alcances y limitaciones

### 1.6.1. Alcances

- Seguimiento de objetos con más de una cámara, es decir obtener varias perspectivas de un mismo objeto a partir de un arreglo de cámaras para tolerancia a oclusiones.
- Estimación de coordenadas cartesianas del objeto, partiendo de los parámetros PT de las cámaras que lo observan.
- Comunicación entre cámaras y sistema de cómputo, a través de un router, previamente configurado.
- Actualización de parámetros PT de las cámaras para focalizar al objeto de interés, mediante la implementación de un algoritmo que minimice la diferencia entre el centroide del objeto y el centro de imagen.

### 1.6.2. Limitaciones

- El sistema de visión solo es capaz de seguir un objeto a la vez, con lo que si se tiene mas de un objeto en movimiento en el escenario, deberá existir un criterio de discriminación que determine que objeto tiene mayor relevancia para el sistema propuesto.
- Los parámetros PT garantizan que se puede ver el objeto mientras este se encuentre dentro del campo de visión dinámico del arreglo de cámaras, es decir la cobertura del desplazamiento del objeto se ve limitada por el rango de movimiento de cada una de las cámaras que lo observan.
- Las variaciones de intensidad de luz en el entorno pueden afectar de manera importante al algoritmo de seguimiento, esto se debe a que si existen cambios de iluminación abruptos, se obtendrían imágenes en donde se pierde información necesaria para la detección del objeto.

- La latencia en la captura de imágenes, causada por retrasos en el funcionamiento del software o hardware, puede impedir la fluidez en el seguimiento a un objeto en movimiento.
- La detección y seguimiento de objetos con el arreglo de cámaras se realiza en línea y no en tiempo real, es decir existe una ligera diferencia temporal entre el movimiento real del objeto y el movimiento de las cámaras que lo siguen.

# FUNDAMENTO TEÓRICO.

---

## 2.1. Visión por computadora

Por muchos años el ser humano ha inspirado sus avances tecnológicos dentro del campo de la visión artificial, en las propiedades, habilidades y características que tiene este sentido tan vital. El término “visión por computadora” dentro del campo de la Inteligencia Artificial puede considerarse como el conjunto de todas aquellas técnicas y modelos que nos permitan el procesamiento, análisis y explicación de cualquier tipo de información obtenida a través de imágenes digitales [22]. El presente trabajo de tesis se enfoca principalmente en la detección y seguimiento de objetos orientado a vigilancia y seguridad (figura 2.1), por lo que se hace una breve descripción la importancia y el uso que tiene la visión por computadora en este rubro.



Figura 2.1: Centro de control de un sistema de vigilancia.

## 2.2. Seguimiento de objetos

El seguimiento de objetos es una actividad significativa en el campo de visión por computadora o también conocida como visión artificial. El auge de las computadoras de prominente capacidad, la disponibilidad de cámaras de vídeo de bajo costo y alta calidad, y

la ascendente necesidad de un análisis automatizado de vídeo ha dado como resultado un gran interés en los algoritmos de seguimiento de objetos. Existen tres pasos clave en el análisis de vídeo: detección de objetos de interés en movimiento, el seguimiento de esta clase de objetos en cada captura, y el examinar estos objetos en un periodo de desplazamiento con la finalidad de reconocer su comportamiento. De esta forma, el uso de visión artificial para seguimiento de objetos es pertinente en las tareas de [1]:

- Movimiento basado en el reconocimiento, es decir, la identificación humana, detección automática de objetos, etc.
- Vigilancia automatizada, que se encarga de detectar actividades sospechosas y dar seguimiento a la misma (figura 2.2).
- Indexación de vídeo, cuyo trabajo es la anotación automática y recuperación de los videos en bases de datos multimedia;
- Interacción persona-ordenador, en donde se realiza el reconocimiento de gestos, o de retina para la entrada de datos.
- Monitoreo de tráfico, se refiere a el almacenamiento en tiempo real de las estadísticas de tráfico para conseguir un flujo directo.
- Navegación de vehículo, es decir, la planificación de una ruta basada en video y evasión de obstáculos.



Figura 2.2: Estación de monitoreo de tráfico vehicular [1].

En su forma más simple, el seguimiento de objetos se puede definir como la estimación de la trayectoria de un objeto en el plano de la imagen mientras se mueve alrededor de una escena. En otras palabras, un rastreador asigna etiquetas consistentes a los objetos localizados en diferentes marcos de un vídeo. Además, dependiendo del dominio de seguimiento, un rastreador puede proporcionar también información centrada en el objeto, como la orientación, área, o la forma de un objeto.

Los primeros sistemas de video vigilancia se establecieron en lugares privados como bancos e instalaciones militares debido a la importancia de la seguridad en las mismas. Sin embargo lugares como parques, cruces de carreteras, comercios, etc., han incluido este tipo de sistemas correspondiendo a la preocupación de la sociedad en determinar un cierto grado de garantía durante su uso. Aunque el nivel de seguridad exigida en estos entornos varía considerablemente. No se establece la misma escala de riesgo para un parque en las afueras que en un cruce situado en las cercanías de un colegio [2].

No obstante en los sistemas mencionados anteriormente, aquellos cuya necesidad de seguridad tiene una gran importancia, siendo crítico el número de cámaras, y por lo tanto de información que han de procesar estos operadores humanos, es muy grande [2]. A partir de lo anterior se puede comprender la relevancia del seguimiento de objetos aplicado a el área de vigilancia y seguridad; ya que es de suma importancia en los sistemas de seguridad en las empresas, poder detectar a tiempo actividades inusuales en sus instalaciones y dar continuidad a las mismas (figura 2.3), de manera que se tenga información acerca de la trayectoria del objeto de interés, para la toma de decisiones.

### **2.3. Taxonomía de técnicas para seguimiento de objetos**

Una vez que se ha comprendido la definición e importancia de la detección y seguimiento de objetos en sistemas de vigilancia y seguridad, es importante señalar que existen diferentes técnicas para el seguimiento de objetos en el campo de la visión por computadora, esto debido a necesidades particulares en los escenarios donde se desempeñan estos sistemas, además de

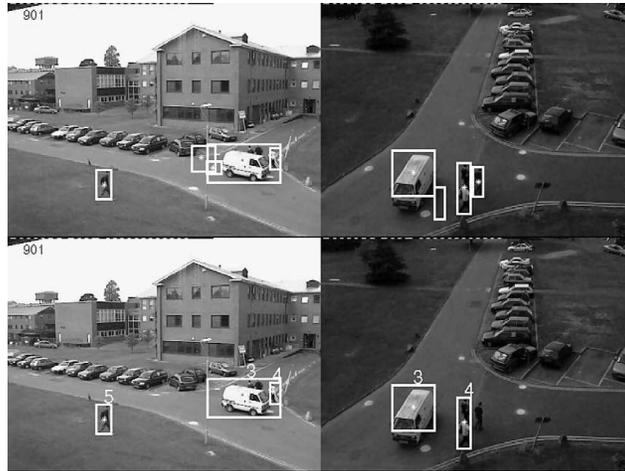


Figura 2.3: Detección y seguimiento de objetos en un sistema multicámara [2].

que cada aplicación va enfocada a utilizar o adaptarse al hardware disponible y a la información que debe de entregar durante su desempeño. Para continuar se muestra una clasificación de las técnicas para seguimiento de objetos propuesta por Chau, Duc Phu y Bremond [3] en 2013, cabe señalar que esta taxonomía se centra en las metodologías para el seguimiento de objetos en general y no aborda algunos aspectos como el seguimiento de objetos específicos, por ejemplo, seguimiento de personas que utilizan la cinemática humana como la base de su aplicación.

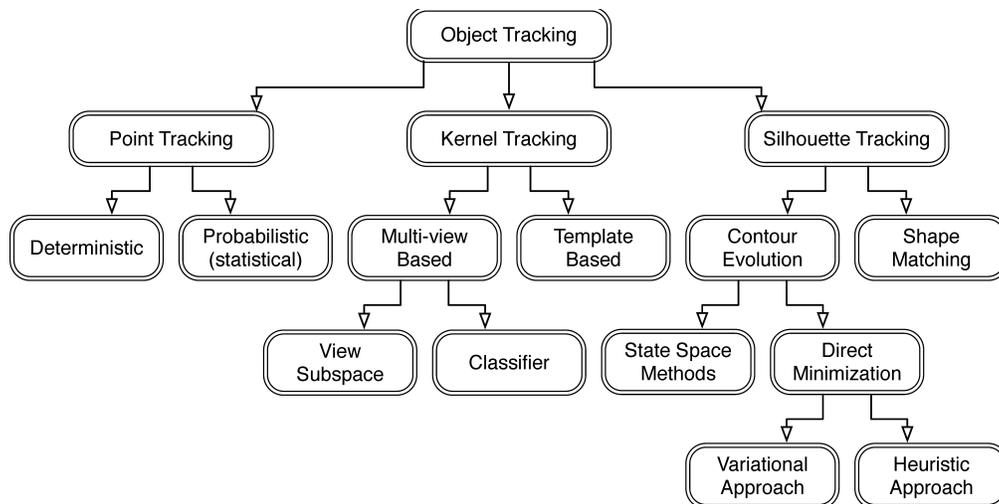


Figura 2.4: Taxonomía de técnicas para seguimiento de objetos [3].

La figura 2.4 muestra la taxonomía expuesta anteriormente y se tiene que de forma general

el seguimiento de objetos está dividido en tres clasificaciones principales:

### 2.3.1. Correspondencia de puntos

Los objetos son detectados en imágenes consecutivas, y están representados por puntos como se muestra en la figura 2.5; la asociación de éstos se basa en el estado previo del objeto, referente a la posición y el movimiento. Este enfoque requiere un mecanismo externo para detectar objetos en cada fotograma. El seguimiento se formula como la correspondencia de los objetos detectados y representados por puntos a través en una secuencia de imágenes, pero la correspondencia se vuelve complicada en presencia de oclusiones, pues se pierde la detección de objetos en las imágenes. Los métodos de correspondencia de puntos se dividen principalmente en dos categorías, deterministas y estadísticos [3].

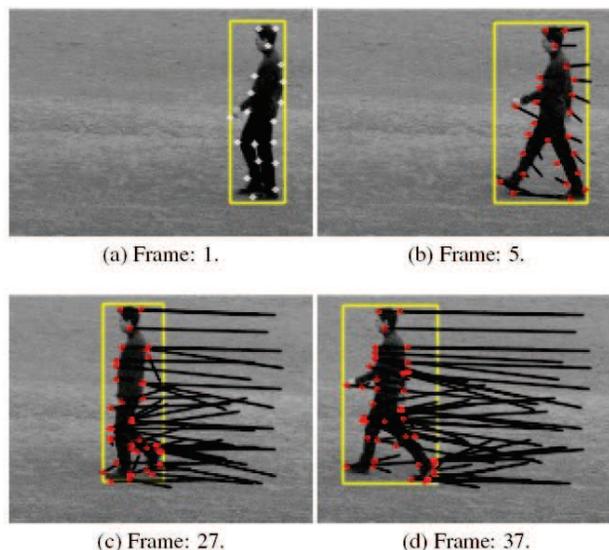


Figura 2.5: Seguimiento de objetos por correspondencia de puntos [3].

### 2.3.2. Seguimiento de núcleo

El termino núcleo se refiere a la forma del objeto y la apariencia, por ejemplo el núcleo puede ser una plantilla rectangular (figura 2.6), o una forma elíptica, con un histograma, el seguimiento se realiza mediante el cálculo de movimiento del núcleo, en imágenes consecutivas.

Este movimiento generalmente se presenta en forma de una transformación paramétrica, como la traslación, rotación y afín.



Figura 2.6: Seguimiento de objetos usando una plantilla rectangular [4].

### 2.3.3. Seguimiento de contornos o siluetas

Esta técnica utiliza conocimiento apriori de la escena y se realiza mediante la estimación de la zona en donde se encuentra el objeto de interés en cada captura de imagen como se observa en la figura 2.7. Los objetos pueden tener formas complejas, por ejemplo: las manos, la cabeza y los hombros, que no pueden ser bien descritos por formas geométricas simples. Los métodos basados en contornos (imagen 2.8) proporcionan una descripción de la forma exacta para estos objetos cuyo objetivo es encontrar la zona del objeto en cada imagen por medio de un modelo de objeto generado usando los fotogramas anteriores [3].

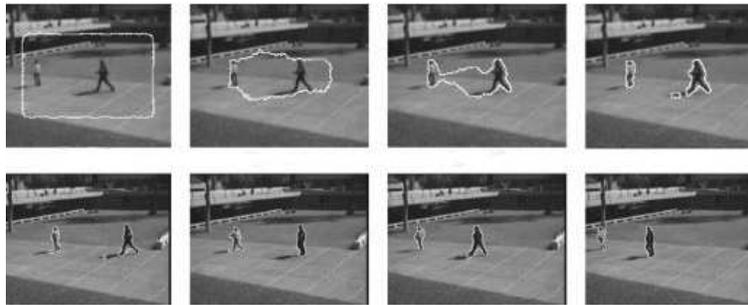


Figura 2.7: Seguimiento de objetos por contorno [5].

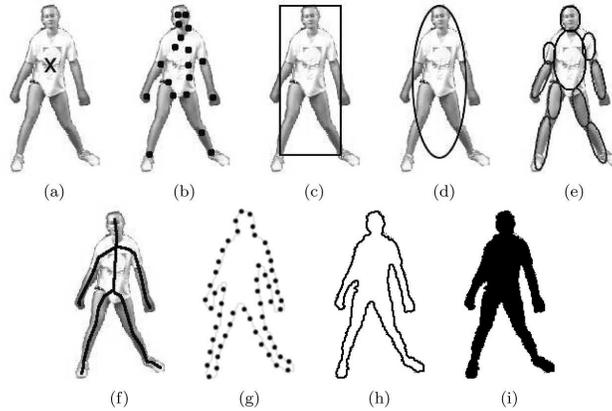


Figura 2.8: Seguimiento de objetos [5]

Seguimiento de objetos(a-b) Correspondencia de puntos, (c-f) Seguimiento de núcleos, (g-i) Contornos o siluetas .

El presente proyecto de tesis realiza el seguimiento de objetos utilizando la técnica de contornos, por lo que se omite la definición de cada una de las herramientas que se derivan del seguimiento de objetos por correspondencia de puntos y los métodos de núcleos, poniendo especial atención en el uso de contornos para la detección y seguimiento de objetos, definiendo cada uno de los métodos que se derivan de esta técnica, según la taxonomía expuesta, hasta llegar a los enfoques de tipo variacional.

**Coincidencia de formas** Se realiza de forma similar al seguimiento basado en plantillas, es decir, se realiza una búsqueda calculando la similitud entre la silueta de un objeto y su modelo asociado, en el frame actual, donde el modelo del objeto se obtiene de un frame anterior, realizando esto de manera continua en una secuencia de imágenes se obtiene el seguimiento de objetos no rígidos de forma implícita. Sin embargo, esta actualización entre la toma anterior y la actual debe de realizarse de forma constante (figura 2.9) para poder hacer frente a problemas de perspectiva, cambios en las condiciones de iluminación, así como movimiento de objetos no rígidos [6].

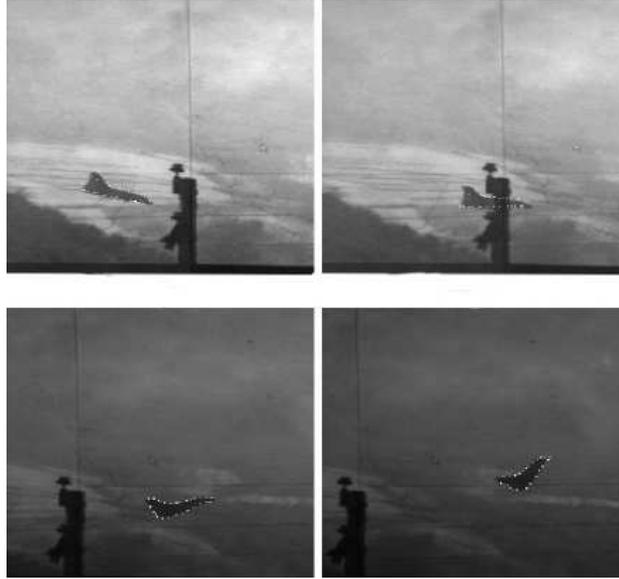


Figura 2.9: Seguimiento de objetos a través de coincidencia de formas [6].

**Evolución de contornos** En contraste con la coincidencia de formas, los contornos evolutivos son colocados de forma inicial cerca del contorno de interés, adaptándose de manera iterativa en cada captura de imagen hasta tomar la forma del objeto a seguir. El seguimiento de objetos con contornos evolutivos se puede realizar utilizando dos enfoques diferentes, el primero usa modelos de espacio de estados, para adaptarse al contorno de interés y detectar el movimiento, el segundo provoca directamente la evolución del contorno, minimizando la energía del mismo, utilizando técnicas de minimización gradiente descendente [1].

**Modelo de espacio de estados:** Los estados están definidos en términos de la forma y parámetros de movimiento del contorno. Los estados se actualizan en cada instante de tiempo, de tal manera que la probabilidad de que el contorno dependa de estos estados se maximiza, en consecuencia el estado posterior, depende del actual y del estado previo, los cuales se definen generalmente en términos de la distancia entre el contorno y los bordes de interés.

**Minimización de energía:** Existe una analogía entre los métodos de segmentación y el seguimiento de contornos, debido a que la segmentación así como el seguimiento de contornos

consisten en minimizar una energía funcional, utilizando los llamados algoritmos codiciosos, o por descenso de gradiente. Los métodos de contornos evolutivos por minimización de energía se pueden dividir en dos enfoques: el enfoque heurístico que consiste en utilizar una serie de reglas o herramientas para lograr que el contorno deformable se adapte al contorno de interés y el enfoque variacional que consiste en utilizar la derivada como un espacio funcional en el que se puedan encontrar los valores que minimizan la energía funcional del contorno deformable (imagen 2.10), tal que este converge en el contorno de interés [5]. Para el sistema

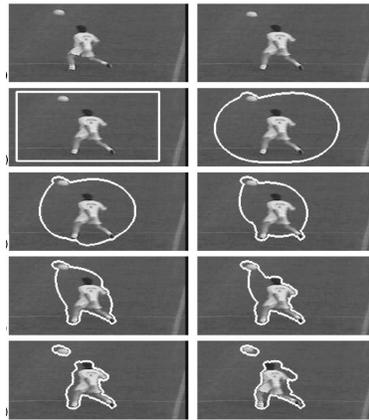


Figura 2.10: Contornos deformables para seguimiento de objetos no rígidos [5].

de visión desarrollado en este trabajo, se utilizan contornos deformables en los que se minimiza la energía utilizando el enfoque variacional propuesto por Michael Kass, Andrew Witkin y Dimitri Terzopoulos [23], además de otras herramientas para el procesamiento de imágenes que permiten la detección del objeto y la actualización de los parámetros Pan & Tilt de cada una de las cámaras en el arreglo, haciendo posible realizar el seguimiento de objetos.

El sustento teórico de cada uno de los métodos utilizados durante el desarrollo experimental, desde la detección de movimiento, seguimiento y estimación de la posición en dos dimensiones del objeto se explican en la siguientes secciones.

## 2.4. Cámaras Pan & Tilt (PT)

Las cámaras Pan & Tilt (figura 2.11), también conocidas como activas o PTZ/IP por su capacidad de giro horizontal, vertical y zoom, además de tener un protocolo de comunicación TCP/IP, que permite el control y transferencia de información vía internet. Debido a la tecnología que gira entorno a ellas, se han vuelto de gran utilidad en el campo de la seguridad y vigilancia y en sistemas de visión por computadora para el seguimiento de objetos, puesto que proporcionan un campo de visión mucho más amplio que las cámaras de tipo estáticas. Esta capacidad la adquieren gracias a que cuentan con tres grados de libertad de los cuales dos son para movimiento y uno para enfoque. Dentro de los dos grados de libertad que proporcionan un movimiento semiesférico a estas cámaras se encuentra el Pan o giro horizontal y el Tilt o giro vertical. No obstante se dice que el espacio cubierto por el giro de Pan & Tilt es semiesférico porque aun cuando existen cámaras de este tipo que proporcionen un giro de 360° en ambos sentidos (horizontal y vertical), cuentan con una zona ciega que depende de la estructura y diseño de cada una de ellas.



Figura 2.11: Cámara PTZ/IP.

Las cámaras activas cuentan generalmente con un paquete para desarrolladores de software, disponible para su descarga en la web de la empresa en la que se adquirieron. Este paquete hace posible crear interfaces distintas de las de fábrica para la intercomunicación Cámara-PC, además de habilitar la manipulación de estos equipos a través de una lista de comandos predefinidos por el fabricante, lo que en conjunto con la comunicación a través de IP (imagen 2.12), provee versatilidad para el desarrollo de software aplicado a visión por computadora.

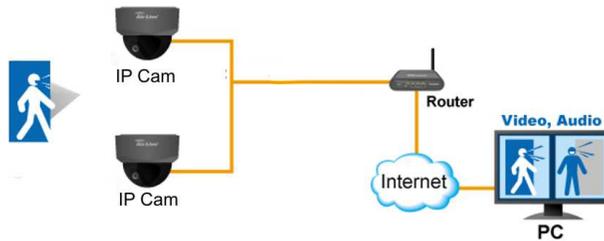


Figura 2.12: Cámaras IP/PT en red.

## 2.5. Sistemas Multicámara

En el ámbito de la investigación los sistemas multicámara, se han utilizado para diversos propósitos, por ejemplo la reconstrucción tridimensional de objetos a partir de imágenes bidimensionales [24], o fusión de múltiples vistas para seguimiento tridimensional[25].

Es posible asegurar que en toda las aplicaciones en las que se utiliza un sistema multicámara, éste tiene la función de capturar diferentes perspectivas de una misma escena, lo que proporciona mayor información para la toma de decisiones a los algoritmos implementados, además de que permite tener una región de interés mucho más amplia que la que se tiene con una sola cámara, como se muestra en la figura 2.13, esto habilita al sistema de visión para hacer frente a posibles oclusiones sin que se pierda información vital sobre el escenario, convirtiéndose en un atributo de suma importancia en escenarios reales en donde el movimiento de los objetos no se controla y donde existen también objetos estáticos los cuales pueden provocar que una o mas cámaras pierdan de vista al objeto de interés.

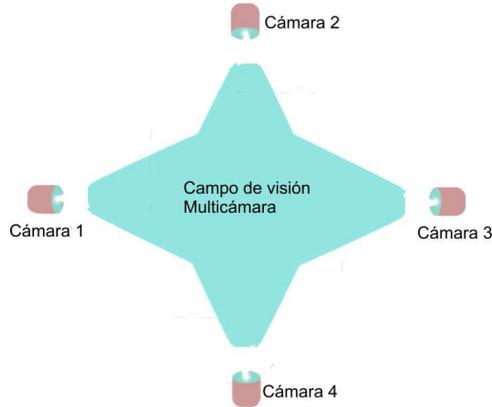


Figura 2.13: Campo de visión de un arreglo multicámara.

### 2.5.1. Campo de visión activo

Como se mencionó anteriormente el aporte de las cámaras Pan & Tilt, es su capacidad de giro vertical y horizontal. Ahora se aborda esta capacidad desde el punto de vista de un sistema multicámara debido a que esta característica representa una ventaja importante cuando se habla de campos de visión solapados de dos o más cámaras.

En la figura 2.13 se puede observar que cada cámara en particular posee un ángulo de visión que proyectado a la distancia indica el área visible para dicha cámara y si se traslapa el área visible de dos o mas cámaras el campo de visión se amplía, haciendo posible cubrir un escenario mas grande.

Si a cada cámara se le dota con la capacidad de girar verticalmente sobre su propio eje, la zona cubierta por el arreglo de cámaras se convierte en un campo de visión activo como se muestra en la imagen 2.15, denominado así por la variabilidad en cuanto a la zona a cubrir la cual depende de la dirección en la que se desplaza el objeto observado (figura ??).

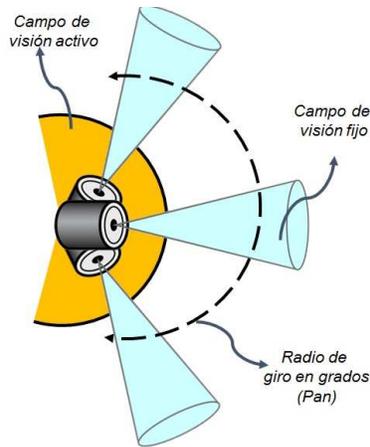


Figura 2.14: Campo de visión activo de una cámara Pan & Tilt.

Todo lo mencionado anteriormente hace de los sistemas multicámara una de las estrategias más empleadas en aplicaciones de visión, cuando se trabaja en escenarios en los que se necesita mantener siempre en cuadro al objeto de interés sin importar la dinámica del espacio en el que se desplaza dicho objeto, mayormente cuando se trata de sistemas de vigilancia y seguridad en donde perder de vista a una persona por la falta de información, puede ocasionar pérdidas económicas.

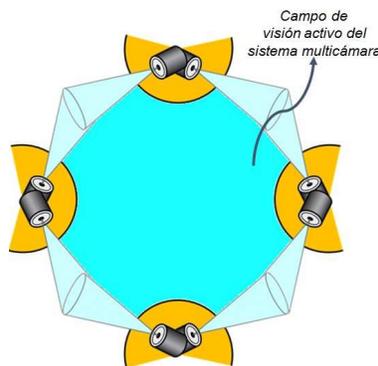


Figura 2.15: Campo de visión activo de un sistema multicámara Pan & Tilt.

### 2.5.2. Factores que afectan a los sistemas multicámara

Si bien los sistemas multicámara aportan propiedades importantes a un sistema de visión, como por ejemplo la tolerancia a oclusiones, o el seguimiento de objetos en un área mucho

mas amplia que con un una sola cámara; también pueden presentar ciertos inconvenientes en su operación en conjunto o de forma individual, puesto que existen factores que afectan a dichos sistemas, como por ejemplo el medio de comunicación que presenta una velocidad de transmisión determinada por el fabricante, pero que depende de el hardware utilizado para intercomunicación Cámaras-Pc.

Otro factor que afecta a los sistemas multicámara es la velocidad de conexión de cada una de las cámaras con el router, o computadora a la que se encuentran ligadas, así mismo la velocidad de giro de los motores en el caso de las cámaras PT puede representar una propiedad influyente en el desempeño de un arreglo multicámara. El protocolo de comunicación utilizado es un aspecto de suma importancia, pues la configuración de este determina en gran manera la velocidad de transferencia de información entre los equipos conectados a una red.

Todos los aspectos mencionados forman parte vital en el funcionamiento de un sistema multicámara, porque determinan en gran manera el tiempo de latencia existente, lo que es de suma importancia en el seguimiento de objetos, pues es necesario que dicha tarea se realice por lo menos en línea y de ser posible en tiempo real.

## **2.6. Detección de movimiento por diferencia de imágenes**

Las operaciones lógicas y aritméticas que se realizan de forma regular entre imágenes son la suma, resta, multiplicación, división, AND, OR y NOT; dichas operaciones son utilizadas como herramientas o preprocesos para diferentes propósitos, por ejemplo: clasificación de objetos [26], normalización de imágenes [27], segmentación de imágenes [28] etc. En el presente trabajo se hace uso de la operación resta (también conocida como diferencia) de imágenes, para la detección de diferencias encontradas en un mismo escenario, es decir la detección de movimiento en una escena, debido a que los objetos estáticos de un escenario no presentan cambios en imágenes tomadas desde una misma perspectiva. La detección de movimiento mediante la diferencia de imágenes como se ilustra en la imagen 2.16 se realiza de la siguiente

manera:

Si se tiene una función bidimensional  $I_1(i, j)$  que representa una imagen digital de un escenario con condiciones lumínicas controladas, donde  $i$  y  $j$  son las coordenadas en el plano digital capturado, la intensidad o nivel de gris está identificado como  $I$ . Esta función puede restarse a la función  $I_2(x, y)$  con las mismas propiedades, pero con distintos valores de intensidad o niveles de gris por representar una imagen digital del mismo escenario capturada en un instante diferente. Por lo tanto si el resultado de la resta o diferencia de estas funciones, es distinto de cero en cualquier subconjunto de coordenadas  $i, j$  del plano digital resultante  $I_3(i, j)$  se dice que existe movimiento en esa zona del escenario.

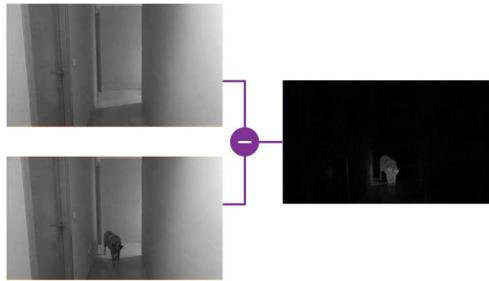


Figura 2.16: Detección de movimiento por resta de imágenes.

En ocasiones en la imagen resultante se puede obtener falsas detecciones de movimiento en zonas muy pequeñas o también conocidas como ruido, provocado por pequeños cambios lumínicos imperceptibles a la vista humana, un ejemplo de esto se muestra en la figura 2.17. Una solución efectiva para este tipo de situaciones es el uso de técnicas de filtrado de imágenes para mejorar ciertas características de las mismas que posibiliten efectuar operaciones.



Figura 2.17: Detección de movimiento con ruido en la imagen resultante.

## 2.7. Proceso de filtrado

El objetivo fundamental del filtrado es obtener imágenes más adecuadas para una aplicación específica, mejorando ciertas características que permitan efectuar operaciones en ellas con resultados mucho más favorables que con imágenes sin filtrar [29]. Algunos objetivos que se persiguen con la aplicación de filtros son:

- Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.
- Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- Realzar bordes: destacar los bordes que se localizan en una imagen.
- Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad.

De manera clara se puede definir el filtrado digital como una convolución de la imagen original con la función filtro; de este modo como una imagen es una función bidimensional, la operación de convolución entre el filtro y la imagen se define:

$$f(t) = f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(x)f_2(t-x)dx \quad (2.1)$$

Como en una imagen es una entrada finita y discreta representada por la función de intensidad de luminancia  $I(i, j)$ , donde  $i, j$  representan las coordenadas bidimensionales de cada píxel que la compone; se obtendrá entonces una imagen de salida  $I'(i, j)$  alterada por un sistema de filtrado  $F$ , cuya función de respuesta es  $f(m, n)$  siendo  $m, n$  las coordenadas matriciales de la función de filtrado. El proceso de convolución puede describirse como [30]:

$$I'(i, j) = f(m, n)I(i, j) = \sum_{m=-k}^k \sum_{n=-l}^l f(m, n)I(i-m, j-n) \quad (2.2)$$



Figura 2.18: filtrado de una imagen.

Los filtros son considerados como operaciones que se aplican a los píxeles de una imagen digital para optimizarla (figura 2.18), enfatizar cierta información o conseguir un efecto especial en ella. El proceso de filtrado puede llevarse a cabo en los dominios de frecuencia y/o espacio.

### 2.7.1. Filtrado en el dominio espacial

Esta formado por aquellas técnicas que actúan directamente sobre los valores de los píxeles de imagen (imagen 2.19), y tienen la siguiente estructura:

$S(i, j) = F(I(i, j))$  donde  $S$  representa la imagen resultante,  $I$  la imagen original y  $F$  el filtro o transformación.

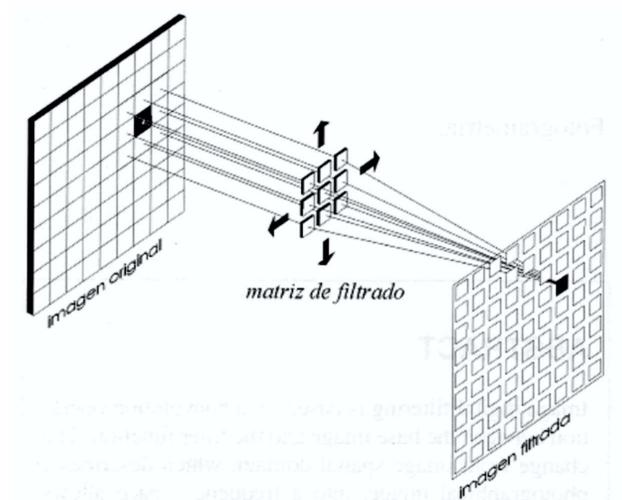


Figura 2.19: Filtrado en el dominio espacial [30].

Los filtros espaciales más comunes son:

- Mediana.

- Realce de bordes por desplazamiento y diferencia.
- Filtro de Laplace.
- Gradiente direccional.

### 2.7.2. Filtrado en el dominio de la frecuencia

En el área de visión por computadora, el dominio de la frecuencia es en el que una imagen se interpreta como la suma de señales periódicas con diferentes frecuencias. Por ejemplo, la transformada de Fourier de una imagen es la suma de exponenciales complejos de distintas magnitudes, frecuencias y fases [31]. Este tipo de transformaciones o filtros cuentan algunas variantes por ejemplo:

- Pasa bajo.
- Pasa alto.
- Pasa banda.
- Muesca.

El proceso de filtrado frecuencial se muestra en la figura 2.20.

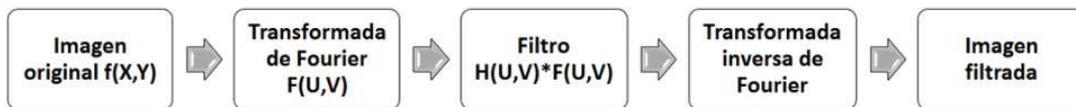


Figura 2.20: Filtrado en el dominio de la frecuencia

## 2.8. Detección de bordes en una imagen

Los bordes de una imagen pueden definirse como la transición significativa de niveles de grises entre dos regiones, éstos representan una fuente de información muy importante sobre

las fronteras de los objetos, por lo que puede ser utilizados para la segmentación de imágenes o reconocimiento de objetos. La mayoría de técnicas para detectar bordes se basan en operadores locales, o aproximaciones discretas de la primera y segunda derivada de los niveles de grises en la imagen.

**Gradiente de imagen** Las variaciones locales de una señal continua se obtienen utilizando la primera derivada; obteniendo valores mayores si las variaciones aumentan con respecto al tiempo. En el caso de las imágenes la derivada es un vector que apunta en la dirección de la máxima variación de niveles de grises, por lo que la derivada tendra valores mayores en los bordes de un objeto, debido a que es ahí donde varía de forma mas notoria el nivel de intensidad o de gris. Este vector se denomina gradiente y se define como [32]:

$$\nabla I = \begin{pmatrix} \frac{\delta(I)}{\delta x} \\ \frac{\delta(I)}{\delta y} \end{pmatrix} \quad (2.3)$$

Como una imagen es un conjunto de puntos discretos, la derivada o gradiente de imagen debe de realizarse en el dominio espacial, por lo que los métodos mas comunes utilizan mascarar o kernels, que recorren la imagen de manera vertical y horizontal, proporcionando las diferencias entre los niveles de grises. Las máscaras se obtienen a partir de la derivada discreta de una imagen (figuras 2.21 y 2.22) como se muestra a continuación.

$$\frac{\delta(I)}{\delta x} \approx \nabla I_x = I(x, y) - I(x - 1, y) \quad (2.4)$$

-1	1
----	---

Figura 2.21: Derivada en el eje x

Mascara para obtener la derivada con respecto al eje x de una imagen.

$$\frac{\delta(I)}{\delta x} \approx \nabla I_x = I(x, y) - I(x, y - 1) \quad (2.5)$$

-1
1

Figura 2.22: Derivada en el eje y

Mascara para obtener la derivada con respecto al eje y de una imagen.

Los operadores mas utilizados, basados en la derivada de una imagen, para la detección de bordes en una imagen son:

- Operador de Roberts

Gradiente fila	Gradiente columna																		
<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>-1</td><td>0</td></tr> </table>	0	0	0	0	0	1	0	-1	0	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	-1	0	0	0	1	0	0	0	0
0	0	0																	
0	0	1																	
0	-1	0																	
-1	0	0																	
0	1	0																	
0	0	0																	

Figura 2.23: Operador de Roberts para detección de bordes.

- Los operadores Prewitt, Sobel y Frei-Chen de formulan de manera conjunta a partir de la figura 2.24, en donde el operador de Prewitt ( $K = 1$ ) involucra a los vecinos de las filas y columnas adyacentes haciéndolo mas robusto con respecto al ruido y el operador de Sobel ( $K = 2$ ) es mas sensible a los bordes diagonales. Finalmente el operador de Frei-Chen ( $K = \sqrt{2}$ ) tiene el mismo gradiente para los bordes verticales, horizontales y diagonales.

	Gradiente columna																						
<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;"><math>\frac{1}{2+K}</math></td> <td> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>K</td><td>0</td><td>-K</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table> </td> </tr> </table>	$\frac{1}{2+K}$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>K</td><td>0</td><td>-K</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	K	0	-K	1	0	-1	<table style="margin-left: auto; margin-right: auto;"> <tr> <td style="padding-right: 10px;"><math>\frac{1}{2+K}</math></td> <td> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>-K</td><td>-1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>K</td><td>1</td></tr> </table> </td> </tr> </table>	$\frac{1}{2+K}$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>-K</td><td>-1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>K</td><td>1</td></tr> </table>	-1	-K	-1	0	1	0	1	K	1
$\frac{1}{2+K}$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>1</td><td>0</td><td>-1</td></tr> <tr><td>K</td><td>0</td><td>-K</td></tr> <tr><td>1</td><td>0</td><td>-1</td></tr> </table>	1	0	-1	K	0	-K	1	0	-1													
1	0	-1																					
K	0	-K																					
1	0	-1																					
$\frac{1}{2+K}$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>-1</td><td>-K</td><td>-1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>K</td><td>1</td></tr> </table>	-1	-K	-1	0	1	0	1	K	1													
-1	-K	-1																					
0	1	0																					
1	K	1																					

Figura 2.24: Operadores de Prewitt, Sobel y Frei-Chen.

## 2.9. Transformación distancia

La transformada distancia es un operador que normalmente sólo se aplica a las imágenes binarias. El resultado de la transformada es una imagen en escala de grises (figura 2.26) donde las intensidades de cada píxel dentro de regiones de primer plano se cambian para mostrar la distancia más corta que existe entre dicho píxel y el cambio de blanco a negro o viceversa más cercano en la imagen original (imagen binaria), como se ilustra en la imagen 2.25 [33].

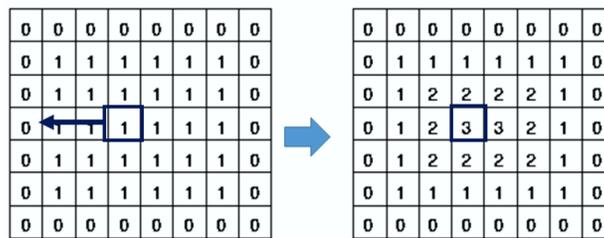
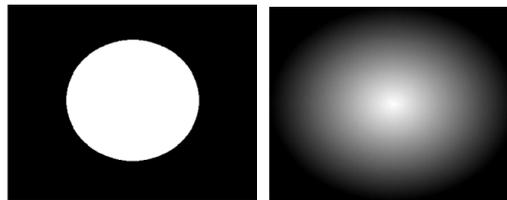


Figura 2.25: Explicación grafica de la transformada distancia.



(a) Imagen original (b) Imagen resultante

Figura 2.26: Transformada distancia

Hay diferentes maneras de obtener la transformación distancia, dependiendo de la métrica que se utiliza para determinar la distancia entre los píxeles, por ejemplo la distancia euclidiana o la distancia de tablero de ajedrez. Una forma intuitiva, pero extremadamente ineficiente de hacerlo es llevar a cabo múltiples erosiones sucesivas con un elemento estructurante adecuado hasta que todas las regiones del primer plano de la imagen se han erosionado. Si cada píxel se etiqueta con el número de erosiones que tuvieron que ser realizadas antes de que desapareciera, entonces se obtiene la transformación mencionada.

No obstante, el elemento de estructuración real que se debe utilizar depende de qué métrica se ha elegido. En resumen la transformada distancia puede ser calculada eficientemente a través de algoritmos encontrados en la literatura, como el propuesto por Rosenfeld y Pfaltz en 1968.

Si una imagen  $A$ , se expresa como un conjunto y se sabe que  $A$  está compuesta por píxeles con coordenadas  $(i, j)$  es posible definir la transformación distancia como una métrica que cumple con las siguientes propiedades:

Sea  $A$  el conjunto de todos los pares de números enteros  $(i, j)$ . La función  $f$  de  $A \times A$  en los enteros no negativos es:

- a) Definida positiva si  $f(x, y) = 0$  si y solo si  $x = y$ .
- b) Simétrica si  $f(x, y) = f(y, x)$  para todo  $x$  en  $A$ .
- c) Triangular si  $f(x, z) \leq f(x, y) + f(y, z)$  para todo  $x, y, z$  en  $A$ .

Si la función  $f$  satisface los requerimientos anteriores entonces se le llama transformada distancia. **Cálculo de centroide por transformación distancia** El centroide de una figura bidimensional se refiere al punto en el cual líneas rectas dibujadas sobre la figura correspondiente se intersectan unas con otras de tal manera que dividen la figura en dos partes iguales en los momentos equivalentes así mismo, la definición puede ser ampliada y se vuelve aplicable un objeto  $n$ -dimensional. Si se establece físicamente, un centroide se refiere al centro del objeto geométrico, por lo tanto, al calcular el centroide de una figura en particular, sólo el área de la figura geométrica se toma en cuenta y por este motivo, el centroide también se denomina como centro geométrico.

Cuando se requiere calcular el centroide de un objeto que no es geoméricamente regular, dichos cálculos se vuelven más complicados. Una alternativa usada en visión por computadora es el cálculo de centroide a través la transformada distancia, ya que como se ha descrito anteriormente ésta proporciona una imagen en escala de grises, donde cada píxel contiene un valor de intensidad que representa la distancia más corta entre dicho píxel y el borde

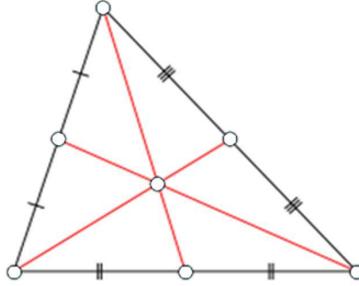


Figura 2.27: Centroide de una figura geométrica.

más cercano, lo que significa que el píxel con mayor intensidad en la imagen es el que se encuentra más cercano al centroide de la figura. Existe la posibilidad de que dos píxeles o más tengan el valor de intensidad más alto de una figura, con lo que se tendría el conflicto de determinar que píxel tomar como el centroide de la figura, cabe mencionar que hay distintas opciones para solucionar este problema, no obstante en este trabajo se toma la transformada máxima extendida como una solución a este problema. Una vez que se tiene el píxel que será considerado como el centroide del objeto o figura, se obtienen la coordenadas de dicho píxel en la imagen, para poder proporcionar el centroide del objeto en coordenadas  $(X, Y)$ .

## 2.10. Contornos activos (Snakes)

Propuestos en 1987 por Kass, Witkin y Terzopoulos[23], se basan en el estudio del movimiento de un contorno abierto o cerrado sobre una imagen a la que trata de adaptarse. Su adaptabilidad a diferentes problemas es tan variada que se han utilizado en muchos campos, por ejemplo en la medicina para la reconstrucción 3-D de imágenes cerebrales [35, 36, 37], en el manejo automático de automóviles [38], en el reconocimiento de características faciales [39, 40, 41] para el reconocimiento de firmas manuscritas [42], en el procesamiento de imágenes satelitales [43, 44], llegando incluso a ser usados en la agricultura[45, 46] . A este contorno se asocia una función de energía que posee dos componentes llamadas energía interna y externa. Estas energías se deben a ciertas características que posee el contorno. Así como la energía interna controla la elasticidad y la

flexibilidad, la externa se debe a la influencia que ejerce la imagen sobre la que esta posicionado el contorno. La forma discreta de este se representa por varios puntos llamados "puntos de control" que en conjunto forman una curva spline (figura 2.28) con la cual se representa este contorno.

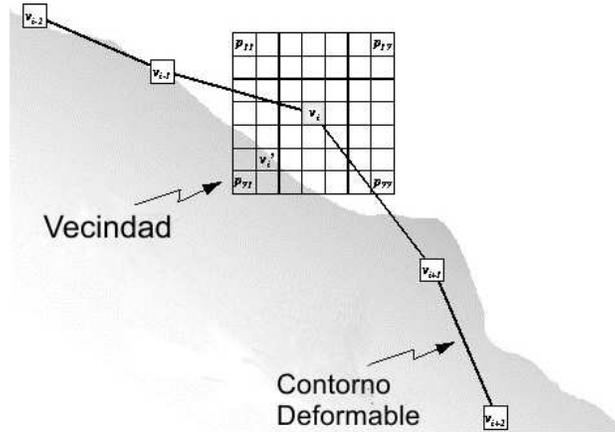


Figura 2.28: Contorno deformable (Snake).

La energía interna a su vez también posee dos términos, la primera y la segunda derivada. La primera describe la distancia entre los puntos (elasticidad) y la segunda la curvatura (flexibilidad), con lo que busca lograr un equilibrio de mínima energía para el cual la posición actual del contorno esté sobre la forma u objeto que se busca, además la manera de serpentear que tiene la curva al momento de buscar este equilibrio es lo que hace que también sean conocidos como Snakes.

**Formulación** La energía del contorno activo según el modelo original viene dada por la fórmula [23]:

$$E_{\text{contorno}} = \int_0^1 [E_{\text{int}}(v(x)) + E_{\text{ext}}(v(x)) + E_{\text{rest}}(v(x))] \quad (2.6)$$

Donde:

- $E_{\text{int}}$  = Energía interna.
- $E_{\text{ext}}$  = Energía Externa o de imagen.
- $E_{\text{rest}}$  = Energía de restricciones.

**Energía interna:** Dentro de la ecuación que modela los contornos activos el primer término que interviene es la energía interna, este término se compone de dos expresiones; la primera es una derivada de primer orden, la cual tiene por objetivo mantener la distancia entre los puntos de control y la otra es una segunda derivada que mantiene la curvatura del contorno.

- **Primera derivada:** Con este término se controla que los puntos no se junten ni se separen en demasía. Su representación formal viene dada por:

$$P_{derivada} = \|v_i - v_{i-1}\|^2 \quad (2.7)$$

Donde  $v_i$  y  $v_{i-1}$  representan dos puntos del contorno y la distancia entre ellos está dada por su diferencia al cuadrado. Sin embargo es posible que los puntos del contorno tiendan a aglomerarse, debido a las características de la imagen, o al algoritmo de minimización, por lo tanto una solución propuesta por Williams y Shah [47] para este problema es:

$$P_{derivada} = (\bar{d} - \|v_i - v_{i-1}\|)^2 \quad (2.8)$$

Donde  $\bar{d}$  es la distancia promedio entre los puntos en cuyo caso discreto, la ecuación queda descrita como sigue:

$$P_{derivada} = \bar{d} - \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (2.9)$$

- **Segunda derivada:** La segunda derivada representa la variación de la primera derivada con respecto a los puntos de control, esto quiere decir que representa los cambios en las distancias entre los puntos de control con respecto a la posición de estos en la curva. Con esta nueva información podemos manejar la curvatura del contorno activo, por ejemplo en un valor alto de este término el contorno tiende a tener una curvatura más suave y menos angulosa. La segunda derivada viene dada por la ecuación:

$$S_{derivada} = \|v_{i+1} - 2v_i + v_{i-1}\|^2 \quad (2.10)$$

Escrita de forma cartesiana:

$$S_{derivada} = (x_{i+1} - 2x_i + x_{i-1})^2 (y_{i+1} - 2y_i + y_{i-1})^2 \quad (2.11)$$

Uniendo los términos la energía interna queda descrita como:

$$E_{int} = \alpha(P_{derivada}) + \beta(S_{derivada}) \quad (2.12)$$

$$E_{int} = \alpha(\|v_i - v_{i-1}\|^2) + \beta(\|v_{i+1} - 2v_i + v_{i-1}\|^2) \quad (2.13)$$

Los valores  $\alpha$  y  $\beta$  son los encargados de dar una ponderación dentro de la ecuación a la elasticidad y flexibilidad de la curva respectivamente. Así para  $\alpha = 0$ ,  $\beta = 0$  la fuerza interna sería nula, con lo cual los puntos del contorno no tendrían nada que los ligue entre sí y actuarían dentro de la imagen solo como un conjunto de puntos independientes unos de otros. Por otro lado para  $\alpha > 0$  y  $\beta = 0$  aparece un factor de estiramiento en la curva es decir, los puntos tenderían a estar más juntos unos de otros y ya no actuarían por sí solos, el factor  $\beta = 0$  ocasiona que no existan restricciones en cuanto a la curvatura del contorno, apareciendo por ejemplo esquinas pronunciadas a lo largo de la curva. Por ultimo para  $\alpha > 0$  y  $\beta > 0$  el contorno será una curva con bordes continuos y curvatura suave [48].

**Energía Externa:** representa la información que proviene de la imagen. A cada punto de ésta por donde se mueva el contorno activo se le asocia una función de energía, que se define por términos que atraen al contorno a terminaciones, líneas y bordes. Para que el contorno sea atraído hacia estas características la imagen debe ser previamente tratada con un preproceso. La energía externa se define como:

$$E_{ext} = E_{linea} + E_{borde} + E_{term} \quad (2.14)$$

- Para la detección de la líneas Kass propone utilizar la función de la imagen por si misma. Esta energía se representa por la intensidad del píxel y se define:

$$E_{linea} = -I(x, y) \quad (2.15)$$

El problema de esta energía es que podría llevar el contorno hacia puntos de ruido que pueden colapsar la curva hacia un contorno falso.

- La energía de borde,  $E_{borde}$ , busca atraer el contorno hacia bordes de la imagen. Para esto Kass propone utilizar una ecuación que lleva el contorno hacia zonas con mayor nivel de gradiente, con esto el contorno tiene una menor sensibilidad al ruido que la energía de línea, por lo que la  $E_{linea}$  se describe como:

$$E_{linea} = -|\nabla I(x, y)| \quad (2.16)$$

- La energía de terminación,  $E_{term}$ , atrae el contorno hacia terminaciones de líneas y esquinas presentes en la imagen. Para esto Kass propone utilizar la curvatura de las líneas en una imagen  $C$  suavizada mediante un filtro de Gauss, de esta manera:

$$E_{term} = \frac{\partial \theta}{\partial n} \quad (2.17)$$

Donde:  $\theta = \arctan\left(\frac{c_y}{c_x}\right)$  y  $n = (-\sin(\theta), \cos(\theta))$

**Minimización de la energía:** La minimización de la energía del contorno es fundamental para lograr la solución buscada. Los métodos para realizar esto se dividen en 4 grandes grupos:

- Basados en análisis numérico: su idea es lograr la minimización de ecuaciones Euler-Lagrange y es el método propuesto en el modelo original [23].
- Métodos estocásticos: permiten encontrar un mínimo global sin caer en mínimos locales, el método estocástico más utilizado es el enfriamiento simulado o simulated annealing
- Métodos probabilísticos: se basan en el movimiento de los puntos hacia un mínimo global mediante cambios probabilísticos, lo que conduce a una solución más rápida.
- Métodos de fuerza bruta: dentro de este grupo destacan dos tipos; los de programación dinámica [49] y los algoritmos voraces [47]. Un ejemplo de minimización de energía por algoritmo voraz, se muestra en la figura 2.29.

Cada uno de los métodos mencionados anteriormente presenta ventajas que dependen de la aplicación que se está dando al algoritmo de contornos activos, y de las necesidades del

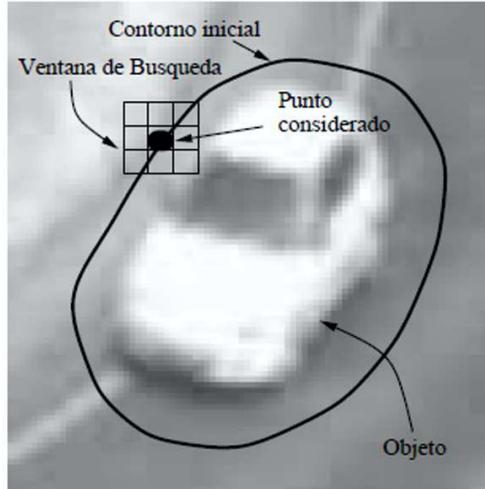


Figura 2.29: Minimización de la energía por algoritmo voraz.

sistema implementado. Por lo tanto, no es posible determinar cual de ellos es el mejor sin hacerlo de forma parcial, es decir, dependiendo del contexto de aplicación de cada uno de ellos. En el presente trabajo de tesis, se usa un método de minimización de la energía del tipo basado en análisis numérico, llamado método de Newton-Raphson. Una de las razones por las que se usa este método es su relativa facilidad en cuanto a la implementación del algoritmo en algunos software de programación como MATLAB.

## 2.11. Método de Newton-Raphson

El método de Newton-Raphson, es de segundo orden de convergencia [50], cuando se trata de raíces reales no repetidas, fue propuesto por Isaac Newton en 1669 [51]. Consiste en llevar la ecuación  $f(x) = 0$  a la forma  $x = g(x)$  de modo que  $g'(x) = 0$  y se utiliza de manera eficiente para encontrar aproximaciones de los ceros o raíces de una función real, su deducción se presenta enseguida [52]. En la figura 2.30 se observa la gráfica de la ecuación  $f(x)$ , y en donde cruza con el eje  $x$  se encuentra una raíz real  $\bar{x}$ , si se supone un valor inicial  $x_0$  y se sitúa en el eje horizontal, trazando una tangente a la curva en el punto  $(x_0, f(x_0))$  y a partir de ese punto se continúa hasta su intersección con el eje  $x$ . El punto de corte  $x_1$  es una nueva aproximación a  $\bar{x}$  (reemplazando la curva por su tangente en el punto  $(x_0, f(x_0))$ ), si el proceso

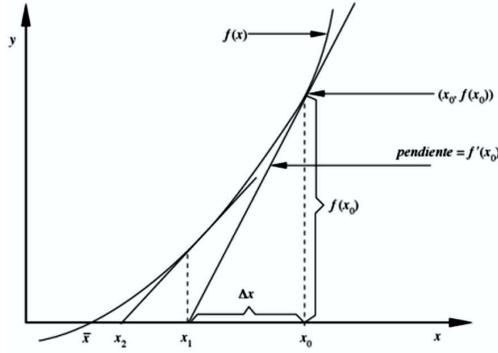


Figura 2.30: Método de Newton-Raphson.

se repite comenzando con  $x_1$  y así sucesivamente hasta encontrar un valor  $x_i$  que satisfaga:

$$|f(x)| \leq \varepsilon_1, |x_{i+1} - x_i| < \varepsilon \quad (2.18)$$

Si lo anterior no se cumpliera en un máximo de iteraciones, debe de reiniciarse el proceso con un nuevo valor de  $x_0$ . Partiendo de lo mencionado anteriormente la ecuación central del algoritmo de Newton-Raphson se obtiene como sigue:

$$x_1 = x_0 - \Delta x$$

Así que:

$$\Delta x = \frac{f(x_0)}{f'(x_0)}$$

y sustituyendo

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Quedando de forma general:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = g(x_i) \quad (2.19)$$

Se dice que este método es de orden dos porque  $g'(\bar{x}) = 0$  y  $g''(\bar{x}) \neq 0$  [52].

## 2.12. Transformación de coordenadas

Durante el análisis de conjuntos y funciones es fundamental utilizar un sistema para representar de forma gráfica puntos específicos. Aunque generalmente se acostumbra utilizar la estructura del espacio afín o del espacio vectorial  $\mathbb{R}^n$ , junto con el sistema de representación cartesiano mediante tuplas de números en el caso del plano, o mediante ternas en el caso del espacio, ya que identificamos con un sistema de coordenadas ortogonal [53]. No obstante existen otras formas de identificar puntos que pueden resultar más útiles, por ejemplo el sistema de de coordenadas cartesianas es muy utilizado para representar la superficie de la tierra en el plano, como en los barcos que utilizan un sistema de radar bidimensional, que coloca los puntos del plano en círculos centrados en el origen de coordenadas. Los aviones o los submarinos hacen uso de un sistema de radar tridimensional, los cuales se basan en sistemas de coordenadas polares o cilíndricas y esféricas que se describen a continuación.

### 2.12.1. Coordenadas polares en el plano

Partiendo de la representación cartesiana del plano, usando pares ordenados de números para simbolizar un punto, se tiene que la distancia  $x$  de este punto a dos ejes ortogonales (ejes de coordenadas), está implícita en las coordenadas (pares ordenados de números) que se asignan al punto, siendo el primer elemento del par la distancia del punto al eje horizontal, y el segundo elemento la distancia al eje vertical.

De este modo, cada punto en el plano cartesiano está unívocamente determinado por sus dos coordenadas, como se muestra en la figura 2.31.

Se puede identificar a cada punto del plano cartesiano con otros dos números de la manera que muestra la imagen 2.32, donde el primero es la distancia que lo separa del origen de coordenadas, denotado como  $r$ , y el segundo es el ángulo  $t$  que forma el segmento que une a  $P$  con el origen en sentido positivo del eje horizontal.

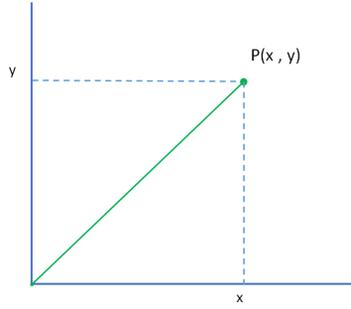


Figura 2.31: Representación de un punto en el plano cartesiano.

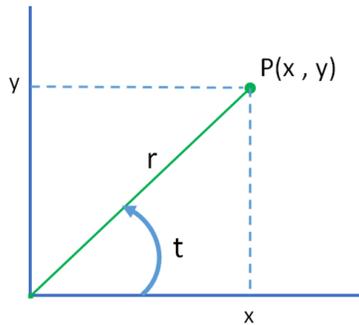


Figura 2.32: Representación polar de un punto en el plano cartesiano.

Esta forma de identificar puntos en el plano cartesiano es conocida como coordenadas polares  $(r, t)$ , donde:

- $r$  es el modulo de  $P$
- $t$  es el argumento de  $P$

De igual manera un par de coordenadas polares pueden transformarse en coordenadas cartesianas y viceversa utilizando las siguientes relaciones:

- Coordenadas polares a cartesianas.

$$x = r \cos(t) \tag{2.20}$$

$$y = r \sin(t)$$

- Coordenadas cartesianas a polares.

$$r = \sqrt{x^2 + y^2} \tag{2.21}$$

$$t = \arctan\left(\frac{y}{x}\right)$$

Para que la función arco tangente esté bien definida (a un número real le corresponda un único ángulo), debe escogerse un intervalo de longitud  $\pi$  en el que definir la imagen. Usualmente se define la función arcotangente de  $\mathbb{R}$  en el intervalo  $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ , es decir  $\arctan : \mathbb{R} \rightarrow \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ . En este caso para un punto  $P$  que esté en el segundo o tercer cuadrante del plano la función  $\arctan\left(\frac{y}{x}\right)$  nos dará un ángulo entre  $-\frac{\pi}{2}$  y  $\frac{\pi}{2}$  y el verdadero argumento de  $P$  será  $t = \alpha + \pi$ . Y si  $P$  está en el cuarto cuadrante, el argumento de  $P$  será  $\alpha + 2\pi$ , es decir:

$$t = \begin{cases} \arctan\left(\frac{y}{x}\right) & \text{si } x \geq 0, y \geq 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & \text{si } x < 0 \\ \arctan\left(\frac{y}{x}\right) + 2\pi & \text{si } x \geq 0, y < 0 \end{cases}$$

Lo anteriormente expresado se observa claramente en la figura 2.33.

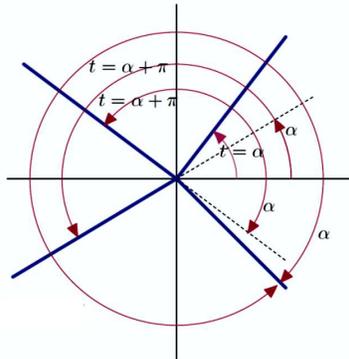


Figura 2.33: Correcta definición de la función arco tangente.

### 2.12.2. Coordenadas cilíndricas en el espacio

En el espacio tridimensional partimos de la representación cartesiana del espacio mediante ternas ordenadas de números, que representan la distancia del punto a tres ejes ortogonales,

llamados ejes de coordenadas. De este modo cada punto del espacio está unívocamente determinado por sus tres coordenadas  $P(x, y, z)$ . Pero también podemos identificar cada punto del espacio por otros tres números (figura 2.34): dos números  $r$  y  $t$  son las coordenadas polares en el plano horizontal de la proyección de  $P$  sobre este plano,  $P'_0 = (x, y, 0)$ , y el tercero es la altura de  $P$  sobre el plano horizontal, la coordenada  $z$ . La terna  $(r, t, z)$  se denomina coordenadas cilíndricas de  $P$ .

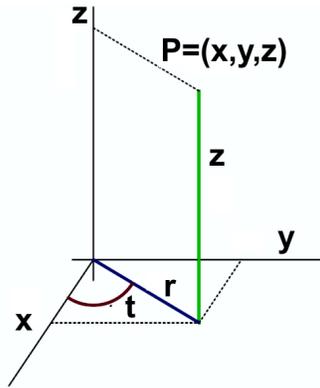


Figura 2.34: Coordenadas cilíndricas.

De esta forma si se quiere pasar de coordenadas cilíndricas a espaciales o viceversa se utilizan las siguientes relaciones:

- Coordenadas cilíndricas a cartesianas.

$$x = r \cos(t) \tag{2.22}$$

$$y = r \sin(t)$$

$$z = z$$

- Coordenadas cartesianas a cilíndricas.

$$r = \sqrt{x^2 + y^2} \tag{2.23}$$

$$t = \arctan\left(\frac{y}{x}\right)$$

$$z = z$$

Usando las mismas condiciones para el arco tangente que en las coordenadas polares.

### 2.12.3. Coordenadas esféricas en el espacio

Cada punto del espacio tridimensional se puede identificar también mediante otros tres números (dos ángulos y una distancia).  $\varphi$  es el ángulo que forma el vector  $P$  con el plano horizontal (latitud),  $\theta$  es el ángulo que forma el vector  $P$  con el plano  $y = 0$  (longitud) y  $\rho$  es la distancia de  $P$  al origen de coordenadas. La terna  $(\varphi, \theta, \rho)$  se denomina coordenadas esféricas de  $P$ .

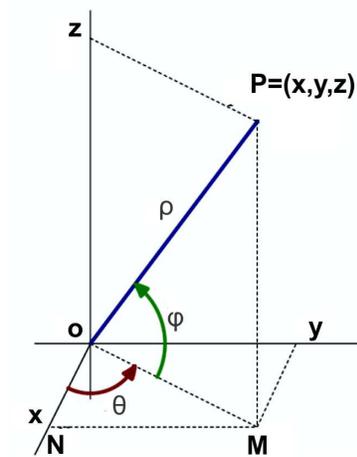


Figura 2.35: Coordenadas esféricas.

En la figura 2.35 se muestra como se representan las coordenadas esféricas en el espacio, donde:

- $\varphi \in [-\pi/2, \pi/2]$
- $\theta \in [0, 2\pi]$
- $\rho \geq 0$

Aplicando un poco de trigonometría a los triángulos  $OPM$  y  $OMN$ , tenemos:

- Coordenadas esféricas a cartesianas.

$$z = \rho \sin \varphi \tag{2.24}$$

$$x = \rho \cos \varphi \cos \theta$$

$$y = \rho \cos \varphi \sin \theta$$

- Coordenadas cartesianas a esféricas.

$$\rho = \sqrt{x^2 + y^2 + z^2} \tag{2.25}$$

$$\varphi = \arcsin \left( \frac{z}{\rho} \right)$$

$$\theta = \begin{cases} \arctan \left( \frac{y}{x} \right) & \text{si } x \geq 0, y \geq 0 \\ \arctan \left( \frac{y}{x} \right) + \pi & \text{si } x < 0 \\ \arctan \left( \frac{y}{x} \right) + 2\pi & \text{si } x \geq 0, y < 0 \end{cases}$$

## 2.13. Inversa de una Matriz

Dado un número  $a$  su inverso recíproco  $x$  es el número tal que resuelva la ecuación:

$$xa = ax = 1$$

Por cierta analogía con los números y tomando en cuenta que el producto de matrices no es conmutativo, se puede conceptualizar a la matriz inversa  $X$  de una matriz dada  $A$  como la matriz que resuelve:

$$XA = I$$

$$AX = I$$

**Definición:** Sea  $A$  una matriz  $m \times n$ .

- Cualquier matriz  $L_{(n,m)}$  para la cual  $L_{(n,m)}A_{(m,n)} = I_{(n,n)}$  se llama inversa izquierda de  $A$

- b) Cualquier matriz  $R_{(n,m)}$  para la cual  $A_{(m,n)}R_{(n,m)} = I_{(m,m)}$  se llama inversa derecha de  $A$
- c) Cualquier matriz  $A_{n,m}^{-1}$  para la cual  $A_{(n,m)}^{-1}A_{(m,n)} = I_{(n,n)}$  y  $A_{(m,n)}A_{(n,m)}^{-1} = I_{(m,m)}$  se llama inversa bilateral de  $A$

Si la matriz  $A$  es de  $m \times n$  entonces cualquier inversa derecha  $R$  o izquierda  $L$  de  $A$  y por lo tanto la inversa bilateral  $A_{-1}$  deben ser de  $n \times m$ . Se puede determinar cual de estos dos tipos de inversas tiene una matriz  $A$  mediante la solución de su conjunto de ecuaciones lineales correspondientes, en donde los elementos de las inversas están considerados como incógnitas [54]. Cuando  $A$  es una matriz de  $n \times n$ , la inversas derecha e izquierda existen y son matrices cuadradas (igual número de filas y columnas) del mismo orden, es muy importante señalar que un análisis a fondo de la existencia de inversas bilaterales, lleva a la conclusión de que solo las matrices cuadradas tienen inversas derecha e izquierda, por lo tanto en el inciso c de la definición anterior se debe suponer que  $m = n$ .

### 2.13.1. Inversas bilaterales

Sea  $A$  una matriz si existen una inversa derecha  $R$  y una inversa izquierda  $L$  de  $A$  entonces son iguales y son una inversa bilateral  $A_{-1}$ . Por lo tanto, cualesquiera dos inversas bilaterales de  $A$  son idénticas.

**Demostración:** Sean  $R$  y  $L$  dos inversas de  $A$  respectivamente, entonces:

$$R = IR = (LA)R = L(AR) = LI = L$$

De esta manera  $R = L \equiv A_{-1}$ , y ésta es una inversa bilateral, luego supongamos que  $X$  y  $Y$  son dos inversas bilaterales de  $A$ . A partir de:

$$X = IX = (YA)X = YI = Y$$

Se obtiene la unicidad, es así que la propiedad de tener inversa bilateral es tan importante que las matrices cuadradas con tales inversas, reciben el nombre de matrices no singulares,

por lo que una matriz singular sería una matriz que no tiene inversa bilateral. A pesar de que una matriz cuadrada singular  $A$  no tenga inversa bilateral, puede obtenerse una inversa utilizando algunos métodos como:

- Por definición  $AB = I = BA$  donde  $B$  es la inversa de la matriz  $A$  e  $I$  es la matriz identidad.
- Gauss-Jordan.
- Gauss.
- Regla de Cramer.

La inversa de una matriz que no es cuadrada sino rectangular (diferente número de filas y columnas), no puede calcularse utilizando los métodos mencionados anteriormente, una alternativa para encontrar la inversa de una matriz rectangular es la pseudoinversa, que se describe a continuación.

### 2.13.2. Pseudoinversa

Definimos como pseudoinversa a la matriz inversa por la derecha o por la izquierda (no ambas) de una matriz  $A$  que se denota como  $A^\dagger$  [55], es decir si una matriz  $A^\dagger$  que multiplicada por la derecha por  $A$ , da la matriz diagonal  $I$  (matriz identidad) con todos los elementos diagonales iguales a 1, se dice que  $A^\dagger$  es la pseudo inversa de  $A$  [56].

- Matriz pseudoinversa por la izquierda:

Recordemos que:

$$L_{(n,m)}A_{(m,n)} = I_{(n,n)}$$

La matriz  $L_{(n,m)}$  puede calcularse construyendo una matriz  $n \times n$  tal que:

$$A^\dagger A = M$$

Si el determinante de  $M \neq 0$ , entonces existe  $M^{-1}$  y puede escribirse:

$$M^{-1}A^\dagger A = M^{-1}M$$

o lo que es igual

$$M^{-1}A^\dagger A = I = LA$$

de donde se obtiene

$$L = M^{-1}A^\dagger$$

- Matriz pseudoinversa por la derecha:

Si se tiene que  $A_{(m,n)R_{(n,m)}} = I_{(,m)}$ , entonces  $AA^\dagger = M$ , si  $M \neq 0$ , entonces existe  $M^{-1}$  y puede escribirse  $AA^\dagger M^{-1} = I = AR$ , por lo tanto:

$$R = A^\dagger M^{-1}$$

# DESARROLLO EXPERIMENTAL

En este capítulo se explica de manera concisa la metodología utilizada (ver figura 3.1), para la implementación de un sistema de visión por computadora, capaz de detectar, seguir y estimar la posición bidimensional de un objeto, también se expone el desarrollo de cada una de las etapas, las cuales se explican a detalle en las secciones posteriores. La infraestructura utilizada para el desarrollo del sistema de visión y la experimentación se encuentra descrita detalladamente en la sección de apéndices.

## 3.1. Metodología

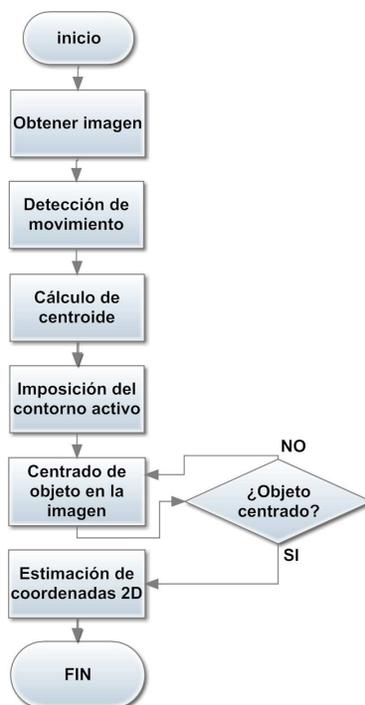


Figura 3.1: Metodología para el desarrollo del proyecto.

## 3.2. Adquisición de imágenes

El proceso de captura de imágenes desde una plataforma distinta a la de fábrica es posible gracias al uso de comandos CGI (Common Gateway Interface), que permite ver al sistema como un modelo cliente servidor; donde el proceso se efectúa en el servidor (cámara) y la velocidad de adquisición depende de la capacidad de ejecutar ordenes CGI, de cada una de las cámaras, pero es importante señalar que este proceso añade latencia a la topología de red utilizada. El acceso a los comandos CGI se obtiene gracias al paquete para desarrolladores de software; el cual es una herramienta, que permite a los programadores crear aplicaciones para un sistema concreto, y que permita la comunicación del dispositivo, con la computadora o algún sistema embebido. Las cámaras FOSSCAM modelo FI8918W empleadas para el desarrollo del proyecto, cuentan con un paquete para desarrolladores, que incluye lo siguiente:

- Firmware.
- Documentación de comandos CGI para manipulación de las cámaras.
- Herramienta de configuración.

Para realizar la adquisición de imágenes (fig. 3.2) se utiliza el comando CGI con la estructura que se muestra.

`http://ip:puerto/snapshot.cgi?user=usuario&pwd=contraseña`

Donde el texto en marcado en color hace referencia a la información que debe sustituir el usuario, de acuerdo a la configuración que haya definido para cada una de las cámaras.

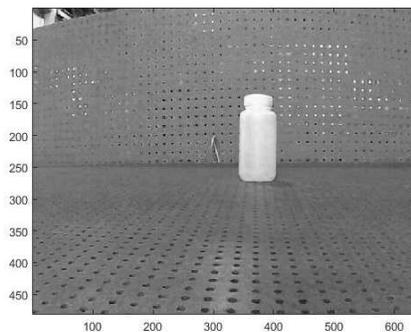


Figura 3.2: Adquisición de imagen desde MATLAB con comando CGI.

### 3.3. Detección de movimiento

En esta sección se explica consistentemente el procedimiento empleado para la detección de movimiento, el cual se lleva a cabo mediante la diferencia de dos imágenes.

Una vez que se realiza la adquisición de una primera imagen de la forma en la que se mencionó anteriormente, se compara mediante la operación resta con una segunda adquisición, en la cual los objetos que tienen movimiento presentan un desplazamiento en los píxeles de la imagen. Como la mayoría de ocasiones este desplazamiento es imperceptible a simple vista, es imposible seleccionar manualmente el objeto a seguir en una escena, sin embargo al desplegar el resultado de la resta en una tercera imagen, se puede apreciar claramente los objetos que presentan un cambio de posición en la escena. Aun así sería complicado, sin la intervención humana, escoger el objeto de interés de forma automática, por lo que para dotar con esta habilidad al sistema desarrollado, este trabajo limita su desempeño a un entorno controlado, donde mediante un fondo impuesto manualmente, se aísla al objetivo, para facilitar su detección.

El algoritmo 1 expresa detalladamente la detección de objetos por diferencia de imágenes y la imagen 3.3, muestra dicha detección.

---

**Algoritmo 1** Detección de movimiento por diferencia de imágenes

---

**Entrada:** Imagen  $f_1(x, y)$ , Imagen  $f_2(x, y)$ , Resolución de imagen  $(x \times y)$ .

**Salida:** Imagen  $f_3(x, y)$ , Detección de movimiento.

```
1:  $f_3 = 0$ .
2: while ( $f_3 == 0$ ) do
3:   for  $i=1:m$  do
4:     for  $i=1:n$  do
5:        $f_3(x, y) = |f_1(x, y) - f_2(x, y)|$ 
6:     end for
7:   end for
8: end while
```

---



Figura 3.3: Detección de movimiento por diferencia de imágenes

### 3.4. Pre-procesamiento

Después de obtener la detección de movimiento en una escena través de la diferencia de imágenes, aunque se trate de un entorno controlado en donde solamente haya un objeto en movimiento, se tiene la posibilidad de que exista ruido en la imagen resultante, por lo que debe de ejecutarse una etapa de filtrado a la imagen con la que se va a trabajar <sup>1</sup>.



Figura 3.4: Detección de movimiento con ruido

Existen diferentes técnicas para el filtrado de imágenes, algunas de ellas mencionadas en el capítulo anterior, no obstante para este trabajo se optó por aplicar un filtro gaussiano a la

---

<sup>1</sup>La figura 3.4 se encuentra en blanco y negro para poder apreciar fácilmente el ruido en la detección de movimiento

proyección digital que resulta de la detección de movimiento, debido a que proporciona una imagen donde el valor máximo aparece en el píxel central y disminuye hacia los extremos, reduciendo de manera efectiva los valores anómalos de intensidad de imagen a cantidades muy cercanas a cero, con lo que, lo mas destacado en la imagen sería el objeto de interés. Se presenta a continuación el algoritmo, para aplicar un filtro gaussiano a la imagen resultante de la detección de movimiento <sup>2</sup>.

---

**Algoritmo 2** Filtro gaussiano aplicado a una imagen

---

**Entrada:** Imagen  $f_3(x, y)$ ,  $\sigma$ .

**Salida:** Imagen resultante  $R(x, y)$  filtrada.

$$g = \frac{1}{\sqrt{2\pi\sigma_x^2\sigma_y^2}} e^{-\frac{x^2 + y^2}{2\sigma^2}}.$$

$$2: G(x, y) = \frac{g(x, y)}{\min_{x,y}(g(x, y))}$$

$$R(x, y) = f_3(x, y) * G(x, y)$$


---

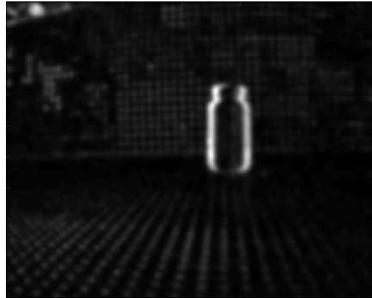


Figura 3.5: Filtro gaussiano aplicado a la detección de movimiento

### 3.5. Cálculo de centroide del objeto de interés

Para calcular el centroide del objeto de interés, después de llevar a cabo los procesos anteriores, se utiliza la imagen filtrada en escala de grises, y se obtiene su inversa, es decir calculando la imagen complemento, puesto que lo que se requiere resaltar es el objeto de interés, asignando así valores intensidad mínimos a los píxeles pertenecientes al fondo, y

---

<sup>2</sup>La imagen 3.5 se encuentra en blanco y negro para resaltar el resultado del filtrado de imagen

valores máximos al objeto de interés, además de que prepara a la imagen para utilizar la transformada máxima extendida en favor del cálculo del centroide.

### 3.5.1. Transformada máxima extendida

La transformada máxima ( $HMAX$ ), pertenece a la clase de operadores basados en componentes conexas y se utiliza para suprimir todos los máximos cuya altura es menor o igual al del valor de umbral  $h$  dado, con lo que se pueden extraer los máximos locales en una imagen en escala de grises, no obstante entrega como resultado una imagen binaria. Esto se consigue realizando la reconstrucción por dilatación de  $f$  desde  $(f - h)$ [57].

$$HMAX_h(f) = R_f^\delta(f - h) \quad (3.1)$$

Donde  $R_f^\delta(f - h)$  es la reconstrucción morfológica por dilatación de la imagen desde  $(f - h)$ . La transformada máxima encuentra los picos de intensidad que son  $N$  valores más altos que el fondo, en otras palabras es un buscador de máximos robusto y se lleva a cabo utilizando la transformada máxima seguida del reconocimiento de máximos regionales. Por lo tanto se dice que la transformada máxima es extendida si un máximo original, es decir un máximo regional de la imagen no reconstruida, todavía pertenece a un máximo regional de la imagen reconstruida. La transformada máxima extendida ( $EMAX$ ) está definida como la máxima regional de la correspondiente transformación ( $HMAX$ ):

$$EMAX_h = RMAX[HMAX_h(f)] \quad (3.2)$$

Para los propósitos de este proyecto, la transformada máxima extendida se utiliza para segmentar la imagen, es decir separar al objeto que nos interesa del fondo de la escena, obteniendo como resultado una imagen binaria, cabe señalar que para obtener este resultado es esencial entregar a la transformada máxima el complemento de la imagen filtrada. El resultado de la segmentación usando  $EMAX$  se observa en la figura 3.6.



Figura 3.6: Imagen segmentada utilizando la transformada máxima extendida

### 3.5.2. Etiquetado

Una vez que se tiene la imagen segmentada, se procede al etiquetado de la misma, para identificar todos los píxeles pertenecientes a un mismo elemento con un determinado valor de intensidad. El etiquetado de una imagen se realiza a través una lista de objetos, la cual se obtiene utilizando la técnica de componentes conexas. Básicamente, las componentes conexas permiten asociar elementos o partes de imágenes entre sí para luego realizar operaciones a partir de la composición de cada imagen en partes. Una componente conexa no es más que un conjunto de puntos o píxeles de las imágenes que se han agrupado a partir de cierta característica que los identifica.

Esta agrupación permite, por ejemplo, discriminar partes de una imagen, encontrar relaciones entre los elementos de una misma componente o incluso relaciones entre distintas componentes. EL numero de objetos en la lista utilizada se determina en base la cantidad de objetos que se obtienen después de etiquetar la imagen; aunque para este casó particular existe un solo objeto en la imagen.

Esto no solo nos permite saber si por diversas circunstancias existiera más de un objetivo en la imagen, además habilita la alternativa de obtener las coordenadas de todos los píxeles que pertenecen a un objeto particular.

### 3.5.3. Algoritmo para cálculo de centroide

El siguiente algoritmo, detalla el procedimiento para calcular el centroide de un objeto (imagen 3.7) utilizando la transformada máxima extendida, como una herramienta de segmentación y el etiquetado como una manera de obtener una lista de coordenadas de los píxeles pertenecientes a dicho objeto.

---

**Algoritmo 3** Cálculo de centroide

---

**Entrada:** Imagen filtrada  $R(x, y)$ .

**Salida:** Centroide  $c(x, y)$  de un objeto en coordenadas de imagen.

```
for i=1:x do
    for i=1:y do
3:      $R^c(x, y) = |255 - R(x, y)|$ 
    end for
end for
6: Transformada máxima extendida  $(x, y) = \text{Maximos}_{regionales}[HMAX_h(R^c(x, y))]$ 
   Etiquetar Transformada máxima extendida  $(x, y)$ .
   Obtener coordenadas  $(x, y)$  de los píxeles donde la Transformada máxima extendida
    $(x, y) == 1$ .
9: Centroide  $c = (\bar{x}, \bar{y})$ .
```

---



Figura 3.7: El centroide calculado se muestra en color amarillo

## 3.6. Imposición del contorno activo (snake)

En el capítulo de fundamento teórico, se explicó detalladamente el concepto y la definición matemática de un contorno activo, por lo que en esta sección solo se hace referencia a éste como una curva deformable, capaz de adaptarse a la forma de un objeto, pero recordemos que una de sus características es que debe de ser colocado cerca del contorno de objetivo, como se ilustra en la figura 3.8, por lo que necesita coordenadas específicas para colocar el centro del contorno, y a partir de esa posición colocar los puntos de control dentro de un radio determinado. Para asegurar que el contorno activo esté ubicado siempre cerca del contorno de interés, se utilizan las coordenadas del centroide del objeto al que se requiere converja la curva deformable, de esta manera se garantiza no solo la ubicación correcta del snake, sino que se esté dando seguimiento al objeto deseado.

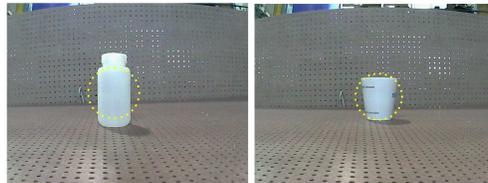


Figura 3.8: Imposición del contorno activo

### 3.6.1. Transformada distancia

Con el objetivo de determinar de forma automática la distancia (radio) de cada punto del snake hacia el centro del mismo, se hace uso de la transformada distancia la cual es un operador que normalmente sólo se aplica a las imágenes binarias. El resultado de la transformada es una imagen en escala de grises donde las intensidades de cada píxel dentro de regiones de primer plano se cambian para mostrar la distancia mas corta que existe entre dicho píxel y el cambio de blanco a negro o viceversa mas cercano en la imagen original (imagen binaria) [33]. El algoritmo para calcular la transformada distancia es:

---

**Algoritmo 4** Transformada distancia

---

**Entrada:** Imagen binaria complemento  $R_{bin}^c(x, y)$ .

**Salida:** Transformada distancia  $T_{dist}(x, y)$ .

p= píxeles (x,y).

Recorrido en orden video de píxeles  $(p) \in R_{bin}^c$

**if**  $R_{bin}^c(p) == 1$  **then**

4:  $T_{dist}(p) = 1 + Min \{R_{bin}^c(q) : q \in N_G^-(p)\}$ .

**end if**

Recorrido en orden video inverso de píxeles  $(p) \in R_{bin}^c$

**if**  $R_{bin}^c(p) \neq 0$  **then**

8:  $T_{dist}(p) = Min \{R_{bin}^c(p), 1 + Min \{R_{bin}^c(q) : q \in N_G^+(p)\}\}$

**end if**

---

Al obtener la transformada distancia del objeto al que se adaptará el snake impuesto, se puede determinar la distancia  $r$  de los puntos del contorno activo hacia el centro del mismo, multiplicando el máximo valor  $D$  de la transformada, por un escalar  $E$  como sigue:

$$D = Max(Transformada\ distancia) \quad r = D * E \quad (3.3)$$

El algoritmo 5 describe de forma detallada como se implementó el método de contornos activos en este trabajo.

---

**Algoritmo 5** Contornos activos

---

**Entrada:** Imagen  $f_2(x, y)$ , número de puntos de contorno.

**Salida:** Convergencia del contorno deformable hacia el contorno de interés.

Inicializar  $\alpha, \beta, \omega_1, \omega_2, \omega_3$ .

Obtener  $\nabla f_2(x, y)$  /\* Gradiente de imagen o primera derivada \*/

Obtener  $\nabla^2 f_2(x, y)$  /\* Gradiente de imagen o primera derivada \*/

$$E_{int} = (\alpha * \nabla f_2(x, y))^2 + (\beta * \nabla^2 f_2(x, y))^2$$

5: Inicializar  $\sigma_1, \sigma_2, \sigma_3$ .

$$G_1(x, y) = \frac{g_1(x, y)}{\min_{x,y}(g_1(x, y))} : g_1 = e^{-\frac{x^2 + y^2}{2\sigma_1^2}}$$

$$G_2(x, y) = \frac{g_2(x, y)}{\min_{x,y}(g_2(x, y))} : g_2 = e^{-\frac{x^2 + y^2}{2\sigma_2^2}}$$

$$G_3(x, y) = \frac{g_3(x, y)}{\min_{x,y}(g_3(x, y))} : g_3 = e^{-\frac{x^2 + y^2}{2\sigma_3^2}}$$

$$GI(x, y) = G_1(x, y) * R(x, y) /* R(x, y) imagen filtrada de detección de movimiento */$$

10:  $GJ(x, y) = G_2(x, y) * \nabla f_2(x, y)$

$$Gk(x, y) = G_3(x, y) * \nabla f_2(x, y)$$

$$E_{ext} = \omega_1 * GI(x, y) - \omega_2 * GJ(x, y)^2 - \omega_3 * Gk(x, y) * R(x, y)$$

$$E_{rest} = 0$$

$$E_{contorno} = E_{int} + E_{ext} + E_{rest}$$

15:  $c = (\bar{x}, \bar{y})$  /\* Centroide \*/

$$r = D * E /* radio */$$

Inicializar puntos  $p$ .

$$K = 0 : 1 : p /* Vector */$$

$$fi = (2 * \pi / p) * K /* radio y angulo */$$

20:  $x_c = \bar{x} + r * \cos(fi)$  /\* coordenadas del snake en x \*/

$$y_c = \bar{y} + r * \sin(fi) /* coordenadas del snake en y */$$

Desplegar contorno activo con radio  $r$  y centro en  $(\bar{x}, \bar{y})$

Convergencia del snake al contorno de interés minimizando energía con el método Newton-Raphson.

---

Para que el contorno activo se adapte al objeto de interés, se debe minimizar su energía funcional, es decir encontrar la posición en la imagen donde la diferencia entre cada punto del snake y el contorno objetivo sea mínima. El método empleado en este proyecto para la etapa de minimización de energía es el método conocido como Newton-Raphson, que es del tipo numérico y se caracteriza por tener un bajo coste computacional.

El algoritmo 6 implementado y adaptado a las necesidades de esta tesis se presenta a continuación.

---

**Algoritmo 6** Newton-Raphson

---

**Entrada:**  $E_{contorno}$ , coordenadas  $(x_c, y_c)$  del snake , N°de iteraciones  $n$ ,  $\varepsilon$ ,  $l$ ,  $k$ .

**Salida:** coordenadas  $(x_c, y_c)$  del snake una vez que se adaptó al contorno objetivo.

```

Obtener  $\nabla f_2(x, y)$  /* Gradiente de imagen o primera derivada */
 $X = (x_c, y_c)$  /* Condición inicial */
 $\varepsilon_{minimo}$  /* Error minimo */
 $IT$  /* Contador de iteraciones */
while (( $IT < n$ ) && ( $\|\varepsilon\|_l > \varepsilon_{minimo}$ )) do
6:    $X = X - k * \frac{\delta x}{\delta y}$ 
       $\varepsilon = \text{interpolar}(X, E_{contorno})$ 
      if ( $\|X\| \neq 0$ ) then
           $\varepsilon = [1 \ 1]$ 
      end if
       $IT = IT + 1$ 
12:  Desplegar en  $R(x, y)$  las nuevas coordenadas  $(x_c, y_c)$  del snake
end while

```

---

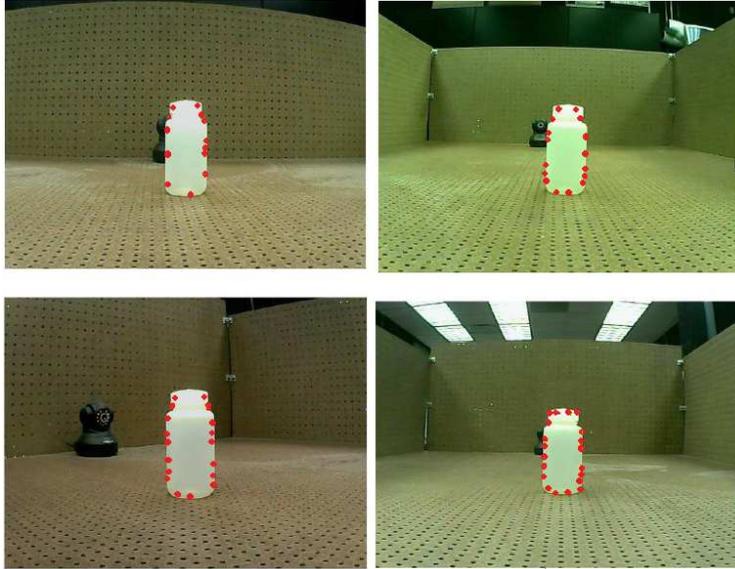


Figura 3.9: Contorno activo en convergencia con el contorno de interés

Es importante señalar que cuando el contorno activo converge hacia el contorno del objeto (fig. 3.9), se vuelve a calcular el centroide del objeto, pero esta ocasión utilizando las coordenadas de los puntos del contorno activo; ésto se realiza para asegurar que el objeto que se va a centrar en la imagen es el correcto, evitando así falsos positivos. El centroide obtenido de las coordenadas de cada elemento del contorno activo se estima como se muestra en la ecuación 3.4.

$$c = (\bar{x}_{snake}, \bar{y}_{snake}). \quad (3.4)$$

### 3.7. Seguimiento de objetos

El seguimiento de objetos es indispensable en muchos sistemas de visión por computadora, y de suma importancia en sistemas de video vigilancia, sin embargo como se menciona en el capítulo uno, la mayoría de sistemas de visión realizan el seguimiento de objetos con cámaras sin movimiento de rotación, por lo que el campo de visión para esta tarea es mucho más limitado; de esta forma cuando se utilizan cámaras Pan & Tilt, se obtiene un campo de visión dinámico, dentro del rango de rotación de la cámara.

Aunque existen distintas técnicas para el seguimiento de objetos, en el presente trabajo de tesis, se aprovecha ampliamente que se debe de calcular el centroide del objeto para imponer el contorno activo siempre cerca del contorno objetivo, para centrar al objeto en la imagen de manera continua, lo que hace posible el seguimiento de objetos.

### 3.7.1. Centrado del objeto en la imagen

La capacidad de hacer coincidir el centroide de un objeto con el centro de imagen de manera automática mediante el uso de algoritmos y del paquete para desarrolladores, es una característica que solo está presente en las cámaras con movimiento de rotación, debido a que cuentan con motores que permiten el giro vertical y horizontal de las mismas. Además de que es posible manipular de manera aceptable la rotación a través de comandos específicos predefinidos por el fabricante; estos comandos tienen también la facultad de determinar el ángulo de giro de cada motor en las cámaras, propiedad que eventualmente se traduce en centrado del objeto en la imagen.

La estructura del comando CGI para controlar el movimiento de rotación de las cámaras e indicar el ángulo de giro, es la siguiente:

`http://ip:puerto/decoder_control.cgi?command=dirección&degree=ángulo&user=usuario  
&pwd=contraseña`

Donde el texto en marcado en color gris hace referencia a la información que debe sustituir el usuario, de acuerdo a la configuración que haya definido previamente y el texto en color azul, debe sustituirse por lo siguiente:

Valor	Dirección
0	Arriba
2	Abajo
4	Izquierda
6	Derecha

Tabla 3.1: Rotación.

Ángulo de giro en grados	
Horizontal	0 - 300
Vertical	0 - 120

Tabla 3.2: Ángulo de giro ( $\theta$  y  $\phi$ )

Las etapas de la metodología desarrolladas hasta el momento, son suficientes para llevar a cabo el proceso de centrar al objeto en la imagen, o como se menciona en el algoritmo 7, manipular el movimiento horizontal y vertical de la cámara hasta lograr que la diferencia entre el centroide del objeto y el centro de imagen sea menor o igual que un error determinado por el usuario, logrando de esta manera el seguimiento del objeto dentro del rango de rotación de la cámara. Un ejemplo de esta etapa se muestra en la figura 3.10.

---

**Algoritmo 7** Centrado de objeto en la imagen

---

**Entrada:** Centroide del snake  $(\bar{x}_c, \bar{y}_c)$ , centro de imagen  $(x, y)$ , IP de la cámara.

**Salida:** Objeto de interés centrado en la imagen, ángulos  $\theta$  y  $\phi$ .

```
1:  $(\bar{x}_c, \bar{y}_c) = 0$      $\theta = 0$      $\phi = 0$      $i = 0$ .
2: while  $(|(\bar{x}_c, \bar{y}_c) - (x, y)| > \varepsilon_{predefinido})$  do
3:   Realizar detección de movimiento, posterior filtrar la imagen. Algoritmos 1 y 2.
4:   Calcular centroide ( algoritmo 3).
5:   Obtener transformada distancia con algoritmo 4 y radio  $r$  con ecuación 3.3.
6:   Crear contorno activo (algoritmo 5) e imponerlo con radio  $r$  y coordenadas  $(x_c, y_c)$ .
7:   Adaptar contorno activo a contorno de interés con el método de Newton-Raphson.
8:    $(\bar{x}_c, \bar{y}_c) =$  centroide del snake.
9:   if  $(\bar{x}_c > x)$  then
10:    Girar cámara a la izquierda con un ángulo  $\Delta\theta$  /* Centrado horizontal*/.
11:     $\Delta\theta = -\Delta\theta$ .
12:   else
13:    Girar cámara a la derecha con un ángulo  $\Delta\theta$ .
14:   end if
15:   if  $(\bar{y}_c > y)$  then
16:    Girar cámara hacia abajo con un ángulo  $\Delta\phi$  /* Centrado vertical*/.
17:   else
18:    Girar cámara hacia arriba con un ángulo  $\Delta\phi$ .
19:     $\Delta\phi = -\Delta\phi$ 
20:   end if
21:    $i = i + 1$      $\theta = \sum_{i=1}^n \Delta\theta$      $\phi = \sum_{i=1}^n \Delta\phi$ 
22: end while
```

---



Figura 3.10: Objeto centrado en la imagen, usando algoritmo 7

### 3.8. Estimación de coordenadas bidimensionales del objeto

Obtener  $\theta$  y  $\phi$  al final del algoritmo, como variables de salida es uno de los pasos de suma importancia para el aporte de este proyecto de tesis, ya que como se plantea en el objetivo general, se requiere dar seguimiento a un objeto en movimiento con un arreglo de cámaras Pan & Tilt, sin embargo también se necesita que el sistema propuesto sea capaz de proporcionar las coordenadas bidimensionales del objeto a partir de los ángulos  $\theta$  y  $\phi$  del arreglo de cámaras que observan al objeto.

Una vez que se tienen las variables de salida mencionadas anteriormente, se puede plantear la posición espacial relativa del objeto con respecto a cada cámara, utilizando el sistema de representación esférica como se muestra en la ecuación 3.5.

$$\underbrace{P(x, y, z)}_{\text{coordenadas cartesianas}} = \underbrace{(r, \theta, \phi)}_{\text{Coordenadas esféricas}} \quad (3.5)$$

El propósito del enfoque propuesto es obtener coordenadas bidimensionales, por lo que se usa la representación polar, derivada de la representación esférica como sigue:

$$\underbrace{P(x, y)}_{\text{coordenadas cartesianas}} = \underbrace{(r, \theta)}_{\text{Coordenadas polares}} \quad (3.6)$$

Por lo tanto, la posición en dos dimensiones con respecto a un arreglo de  $n$  cámaras, se expresa:

$$P_1(x, y) = (r_1, \theta_1) \tag{3.7}$$

$$P_2(x, y) = (r_2, \theta_2)$$

$$P_3(x, y) = (r_3, \theta_3)$$

⋮

$$P_n(x, y) = (r_n, \theta_n)$$

Para este caso  $n = 4$  entonces:

$$P_1(x, y) = (r_1, \theta_1) \tag{3.8}$$

$$P_2(x, y) = (r_2, \theta_2)$$

$$P_3(x, y) = (r_3, \theta_3)$$

$$P_4(x, y) = (r_4, \theta_4)$$

Donde  $r$  es la distancia de cada cámara al objeto, y  $\theta$  el ángulo total de giro horizontal, necesario para lograr que la diferencia entre el centroide y el centro de la imagen sea menor o igual que un error determinado. Si se define el punto inicial de cada coordenada como la posición de cada cámara en el arreglo, y todas las coordenadas pertenecen a un mismo punto visto desde diferentes perspectivas (cámaras), como lo ilustra la figura 3.11 se pueden definir líneas rectas que idealmente se intersectan todas en un mismo punto.

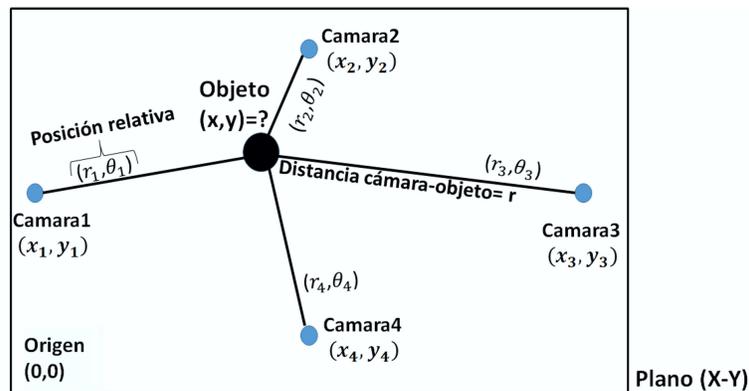


Figura 3.11: Planteamiento para la estimación de posición del objeto

El planteamiento para estimar la posición del objeto a partir de los pares de coordenadas dados en la ecuación 3.8 se obtiene como sigue:

Utilizando las formulas descritas en la sección 2.12, se realiza la transformación de coordenadas polares a cartesianas:

$$\begin{aligned}x &= r * \sin(\theta) \\y &= r * \cos(\theta)\end{aligned}\tag{3.9}$$

Despejando r:

$$\begin{aligned}r &= \frac{x}{\sin(\theta)} \\r &= \frac{y}{\cos(\theta)}\end{aligned}$$

Igualando:

$$\frac{x}{\sin(\theta)} = \frac{y}{\cos(\theta)}$$

Despejando

$$\begin{aligned}x &= y \frac{\sin(\theta)}{\cos(\theta)} = y * \tan(\theta) \\-x + y * \tan(\theta) &= 0\end{aligned}$$

De esta forma obtenemos una ecuación que describe a una recta que parte del origen a un punto particular, en este caso la ubicación del objeto. No obstante las cámaras no están ubicadas en el origen del plano (figura 3.11), por lo que esta ecuación debe de adaptarse a la ecuación de la recta que permite trasladar el origen a un punto dado  $(x, y)$ . Así tenemos:

$$\tan(\theta)(y - P_y) = (x - P_x)\tag{3.10}$$

Desarrollando:

$$x - \tan(\theta)y = P_x - \tan(\theta)P_y\tag{3.11}$$

Donde  $\tan(\theta)$  es la pendiente de la recta,  $P_x$  y  $P_y$  son la posición en  $x$  e  $y$  de la cámara respectivamente y  $x$  e  $y$  son las coordenadas del objeto.

Realizando el mismo procedimiento para todas las cámaras, se puede formular el sistema de ecuaciones que se describe en 3.12:

$$\begin{aligned}
x_1 - \tan(\theta_1)y_1 &= P_{1x} - \tan(\theta_1)P_{1y} \\
x_2 - \tan(\theta_2)y_2 &= P_{2x} - \tan(\theta_2)P_{2y} \\
x_3 - \tan(\theta_3)y_3 &= P_{3x} - \tan(\theta_3)P_{3y} \\
x_4 - \tan(\theta_4)y_4 &= P_{4x} - \tan(\theta_4)P_{4y}
\end{aligned} \tag{3.12}$$

Representado en forma matricial:

$$\begin{pmatrix} 1 & -\tan(\theta_1) \\ 1 & -\tan(\theta_2) \\ 1 & -\tan(\theta_3) \\ 1 & -\tan(\theta_4) \end{pmatrix}_{4 \times 2} \begin{pmatrix} x \\ y \end{pmatrix}_{2 \times 1} = \begin{pmatrix} P_{1x} - \tan(\theta_1)P_{1y} \\ P_{2x} - \tan(\theta_2)P_{2y} \\ P_{3x} - \tan(\theta_3)P_{3y} \\ P_{4x} - \tan(\theta_4)P_{4y} \end{pmatrix}_{4 \times 1} \tag{3.13}$$

Finalmente, la estimación de la posición bidimensional del objeto, utilizando un arreglo de 4 cámaras que observan y centran al objetivo desde diferentes perspectivas; se obtiene de la siguiente manera:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 & -\tan(\theta_1) \\ 1 & -\tan(\theta_2) \\ 1 & -\tan(\theta_3) \\ 1 & -\tan(\theta_4) \end{pmatrix}^{\dagger} \begin{pmatrix} P_{1x} - \tan(\theta_1)P_{1y} \\ P_{2x} - \tan(\theta_2)P_{2y} \\ P_{3x} - \tan(\theta_3)P_{3y} \\ P_{4x} - \tan(\theta_4)P_{4y} \end{pmatrix} \tag{3.14}$$

# RESULTADOS

---

Todas las aplicaciones de visión desarrolladas a pesar de ser diseñadas de manera minuciosa, o de contar con herramientas que lo ayuden a cumplir mejor con su tarea, cuentan con un margen de diferencia entre el objetivo real y el alcanzado; evidentemente el enfoque propuesto en este trabajo no es la excepción.

Por lo anterior, en este capítulo se presenta la adaptación del sistema para el seguimiento y la estimación de la posición 2D de un objeto a partir de los parámetros Pan & Tilt de un arreglo de cámaras que lo observan, además se muestran los resultados experimentales obtenidos durante la implementación de la aplicación, y las conclusiones a las que se llegó luego de todos los procesos y procedimientos realizados.

Como parte de la implementación del trabajo propuesto, se diseñó un arreglo de cámaras que permitiera observar al objeto a través del escenario controlado, como se muestra en la figura 4.1.

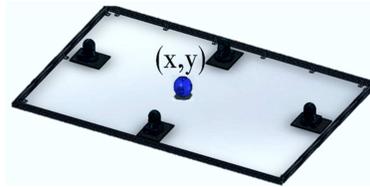


Figura 4.1: Arreglo de cámaras propuesto.

La orientación inicial de las cámaras y su ubicación se determinó en base a las necesidades del escenario es decir, ubicar las cámaras de forma que se aproveche al máximo su capacidad de rotación para cubrir de forma conjunta el área donde puede desplazarse el objeto a seguir y hacer el sistema tolerante a la oclusión de alguna de las cámaras.

El espacio de trabajo también cuenta con perforaciones que tiene una pulgada de distancia entre ellas, con las que se obtiene un marco de referencia para las coordenadas locales (figura

4.2 ), que se utiliza, para conocer la posición de cada cámara en el escenario, sino y diseñar las trayectorias con las que se valida el enfoque propuesto.

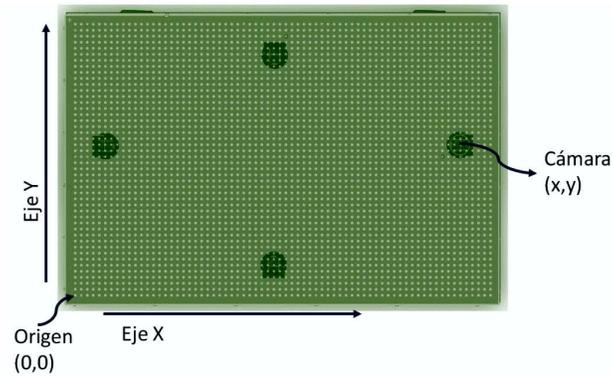


Figura 4.2: Marco referencia con coordenadas locales.

Algunos ejemplos de estas trayectorias se muestran en las figuras 4.3 y 4.4 , en donde los círculos rojos representan las cámaras y el grid azul el marco de referencia. El proceso de validación consiste en comparar la posición real del objeto en coordenadas locales, con la posición estimada.

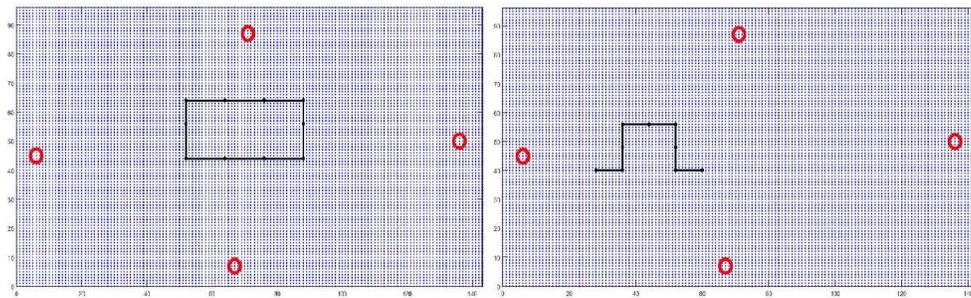


Figura 4.3: Trayectoria en cuadro y en diente de sierra.

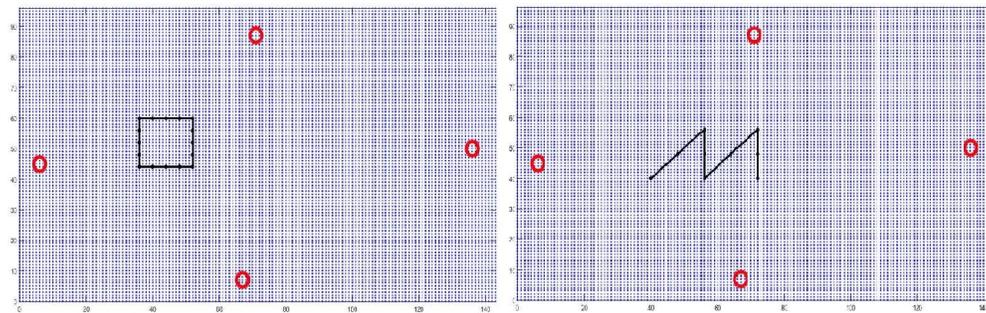


Figura 4.4: Ejemplo de trayectorias usadas durante la experimentación.

La tabla 4.1 muestra un ejemplo de la información necesaria para la estimación de la posición del objeto mediante la metodología descrita en el capítulo anterior. Es muy importante señalar que al ángulo  $\phi$  se le asigna un valor de cero debido a que se estima la posición en el plano. Sin embargo, para el proceso de centrado del objeto en la imagen, su valor es distinto de cero.

Posición física de la cámara: (6,45)	
$\theta$	$\phi$
27	0
4	0
5	0
6	0
1	0

Tabla 4.1: Datos necesarios para la estimación de la posición

Utilizando las herramientas mencionadas anteriormente y la metodología propuesta, se obtienen los siguientes resultados.

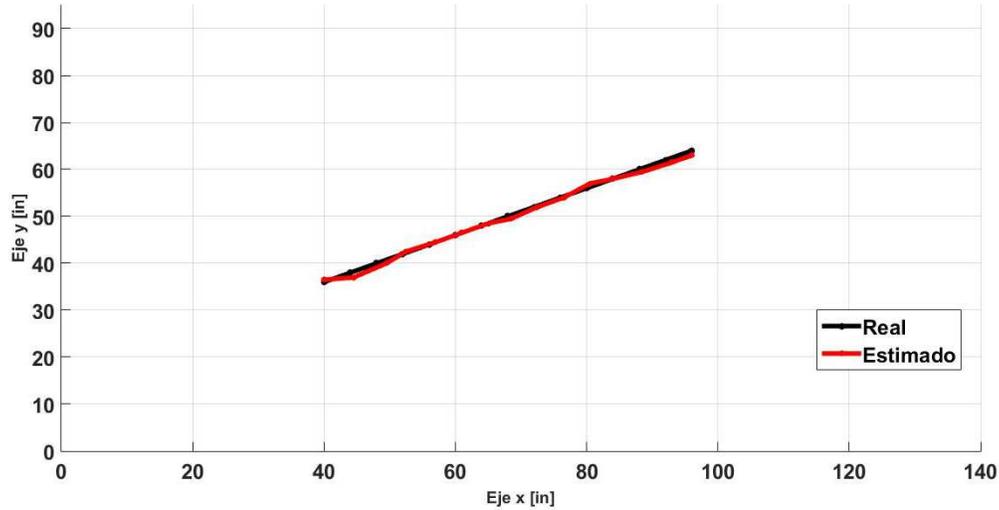


Figura 4.5: Trayectoria diagonal.

La gráfica 4.5 ejemplifica la ruta diagonal (color negro) del objeto moviéndose a través del espacio de pruebas, con coordenadas locales establecidas, y la línea color rojo representa la trayectoria compuesta por las posiciones estimadas, utilizando el enfoque propuesto. Cabe señalar que las dimensiones de los ejes así como las coordenadas de las trayectorias están dadas en pulgadas, de esta forma al calcular el error se le asigna esta unidad de medida. El error para cada trayectoria se calcula usando la ecuación 4.1:

$$\varepsilon_x = x_{i\text{ real}} - x_{i\text{ estimada}} \tag{4.1}$$

$$\varepsilon_y = y_{i\text{ real}} - y_{i\text{ estimada}}$$

Una vez calculado el error para todos los puntos de la trayectoria en ambos ejes, se puede realizar un gráfico (figuras 4.6 y 4.7) que nos permita tener una idea más clara del comportamiento de este parámetro.

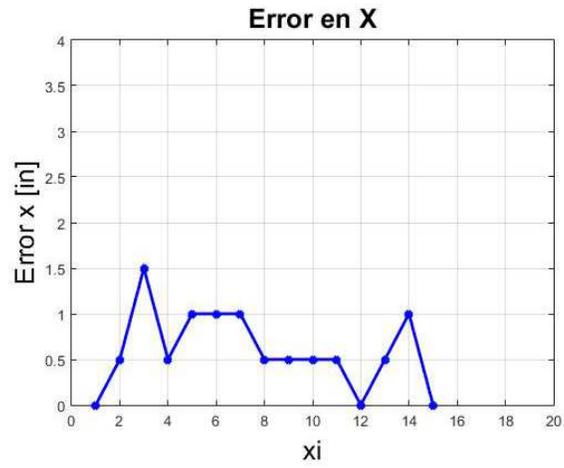


Figura 4.6: Gráfico de error en el eje X de la trayectoria diagonal.

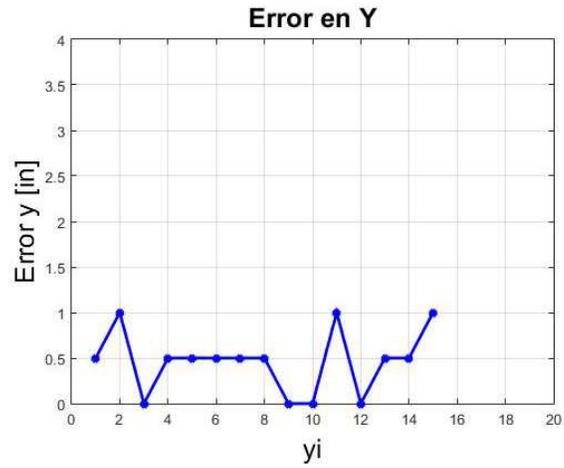


Figura 4.7: Gráfico de error en el eje Y de la trayectoria diagonal.

De igual manera, se puede obtener la desviación estándar del error en los dos ejes (ver figuras 4.8 y 4.9).



Figura 4.8: Desviación estándar del error en el eje  $X$  de la trayectoria diagonal.



Figura 4.9: Desviación estándar del error en el eje  $Y$  de la trayectoria diagonal.

Calculando el promedio con los datos anteriores, se produce la cifra que representa al error de la trayectoria estimada en el eje  $x$  y en el eje  $y$ , así también utilizando un poco de estadística, se calcula la desviación estándar o RMSE. Estos datos se muestran en la tabla 4.2.

	$\varepsilon$	ECM
x	0.6	0.41
y	0.46	0.33

Tabla 4.2: Error de la trayectoria diagonal y su ECM

Para el proceso de validación de resultados, se diseñaron 30 trayectorias distintas, desplazando el objeto por el escenario, según la trayectoria en cuestión. Realizando los procedimientos anteriores para cada una de las trayectorias diseñadas, se adquirió la información sobre la experimentación que se muestra en lo que resta de este capítulo.

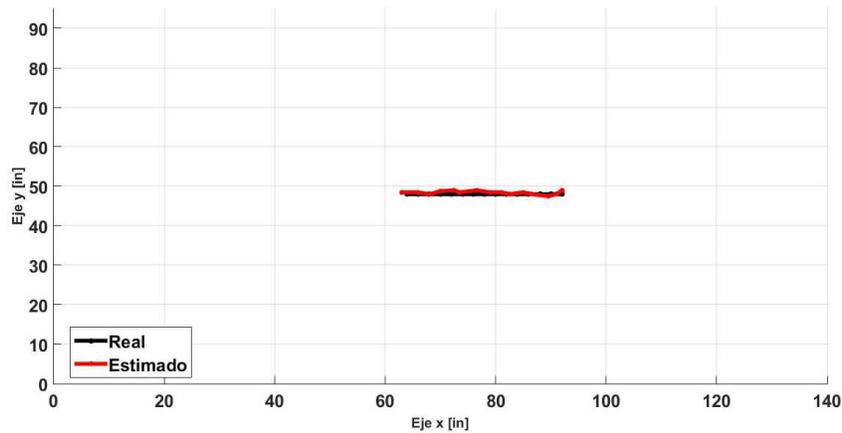


Figura 4.10: Trayectoria horizontal.

	$\varepsilon$	ECM
x	0.6	0.45
y	0.48	0.35

Tabla 4.3: Error de la trayectoria horizontal, y su ECM

Se observa en la tabla 4.3 que el error entre la trayectoria estimada y la real es pequeño, lo cual se confirma de manera visual en la figura 4.10, donde la línea color rojo guarda un alto grado de solapamiento con la línea en color negro.

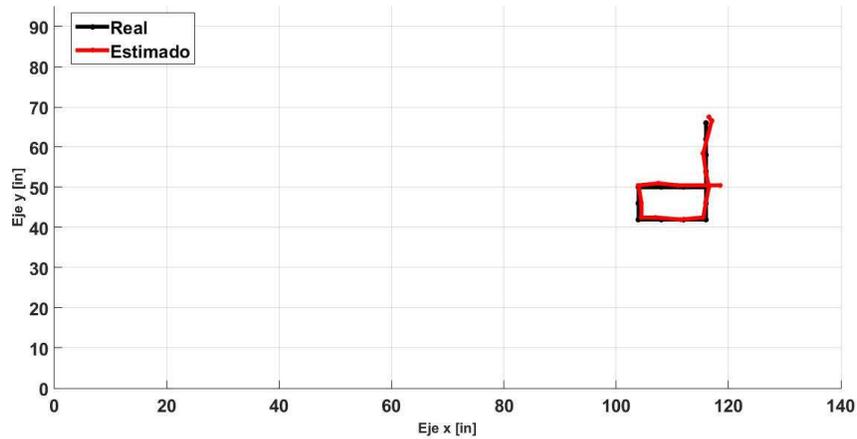


Figura 4.11: Trayectoria en "d".

	$\varepsilon$	ECM
x	0.63	0.59
y	0.76	1.06

Tabla 4.4: Tabla de Error y ECM. Trayectoria en "d"

Los resultados que se observan en la imagen 4.11 muestran que existen casos como los de una ruta que pasa dos veces por un mismo punto donde se obtienen una estimación de posición ligeramente distinta y esto se debe al error acumulado en las estimaciones anteriores, ya que el ángulo  $\theta$  para ese punto se tiene sumando todos los valores anteriores de  $\theta$  desde el punto de inicio.

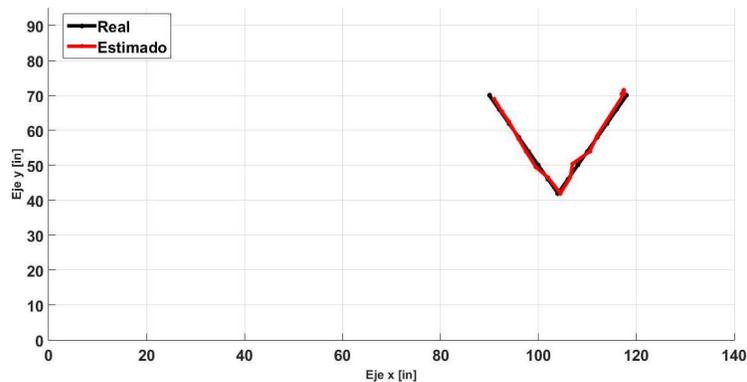


Figura 4.12: Trayectoria en "V".

	$\varepsilon$	ECM
x	0.46	0.57
y	0.43	0.50

Tabla 4.5: Error y ECM. Trayectoria en "V"

La diferencia entre la posición estimada y la real según se observa en la figura 4.12 se mantiene cercana a media pulgada (Tabla 4.5) siendo aceptable, considerando que el objeto no es un punto en el plano sino un cuerpo, por lo que muy probablemente la magnitud del error se mantiene dentro del diámetro que ocupa el cuerpo en el plano.

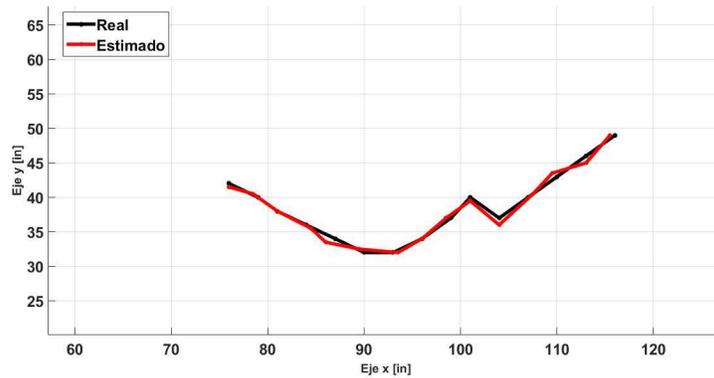


Figura 4.13: Trayectoria en sin forma geométrica.

	$\varepsilon$	ECM
x	0.33	0.44
y	0.41	0.51

Tabla 4.6: Error y ECM. Trayectoria irregular

Las trayectorias diseñadas, tienen formas variadas (ver imágenes 4.14 y 4.15), e incluso formas irregulares como la que se muestra en la figura 4.13.

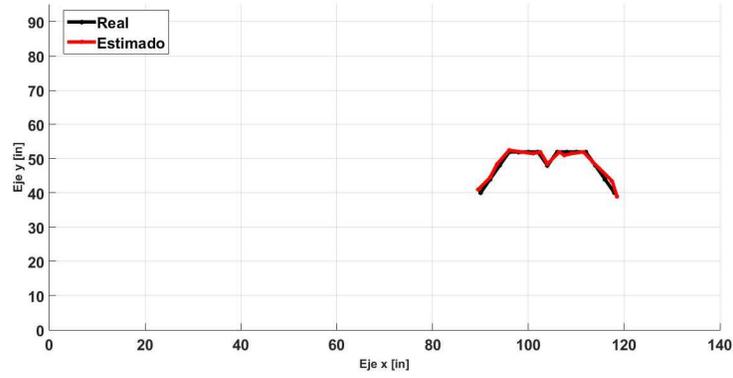


Figura 4.14: Trayectoria en "m".

	$\varepsilon$	ECM
x	0.53	0.65
y	0.50	0.61

Tabla 4.7: Error y ECM. Trayectoria en "m"

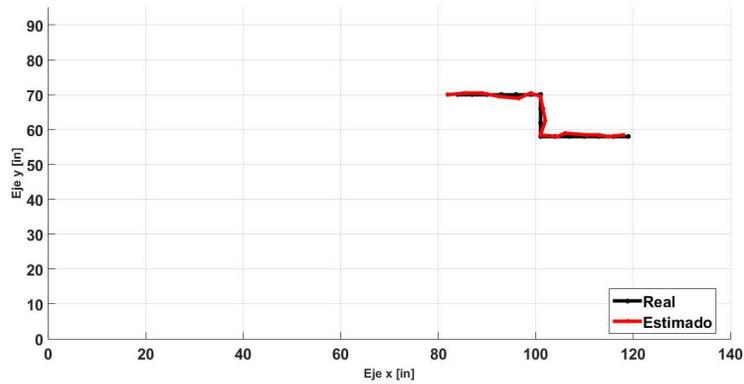


Figura 4.15: Trayectoria escalón.

	$\varepsilon$	ECM
x	0.68	0.88
y	0.43	0.53

Tabla 4.8: Error de la trayectoria escalón, y su ECM

Otro dato interesante que proporciona una idea clara sobre el comportamiento del error en todas las trayectorias diseñadas es el gráfico de dispersión del error, ya que nos permite analizar de forma visual el parámetro mencionado.

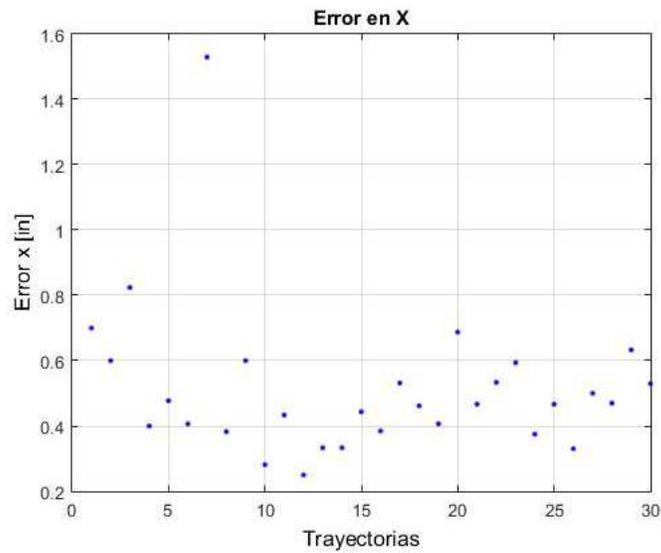


Figura 4.16: Dispersión del error en el eje x para las 30 trayectorias.

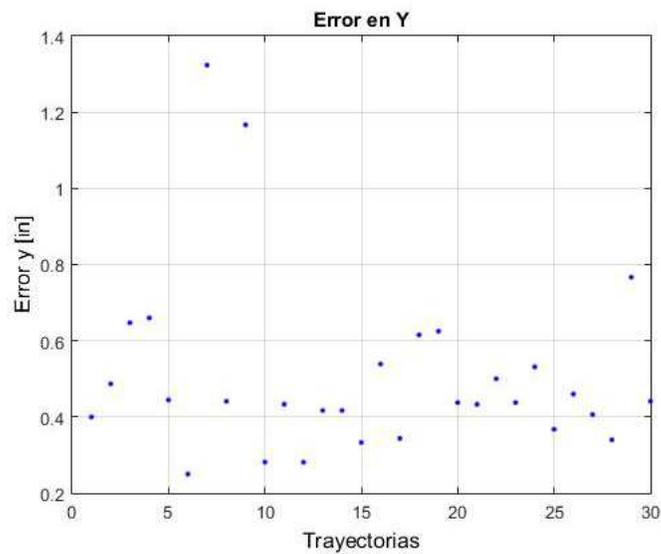


Figura 4.17: Dispersión del error en el eje y para las 30 trayectorias.

En los gráficos anteriores se observa que la diferencia entre las trayectorias reales y las estimadas, es similar no solamente entre trayectorias, además si comparamos las figuras

4.16 y 4.17 se tiene que el error es similar en los dos ejes, para la mayoría de los desplazamientos del objeto a través del escenario.

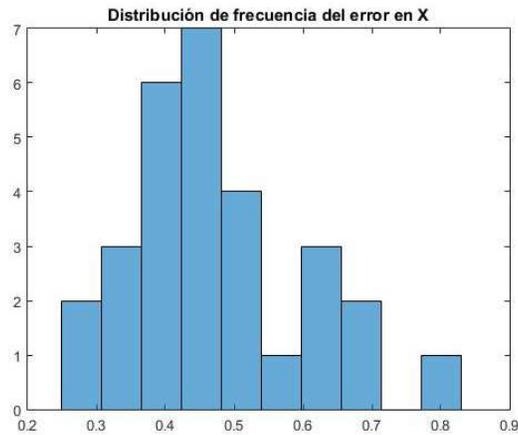


Figura 4.18: Distribución de frecuencia del error en x.

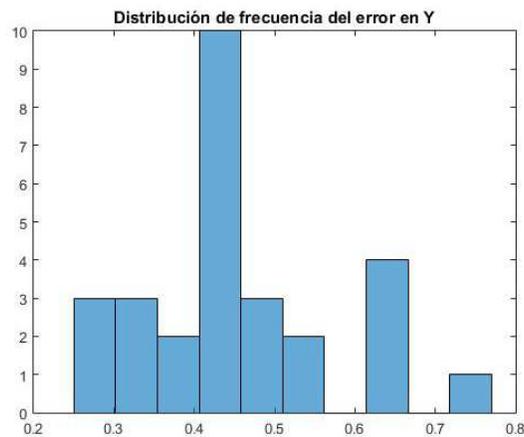


Figura 4.19: Distribución de frecuencia del error en y.

Eliminando los puntos aberrantes de los gráficos 4.16, 4.17 y realizando una distribución de frecuencia para ambos ejes (gráficos 4.18 y 4.19), se observa que el error tiene la tendencia a una distribución normal, o gaussiana, los que permitiría caracterizar al error en futuros análisis.

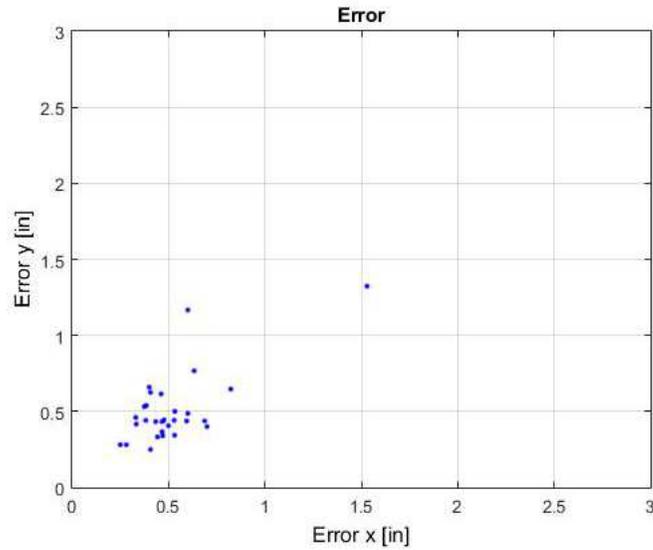


Figura 4.20: Dispersión del error por pares de coordenadas.

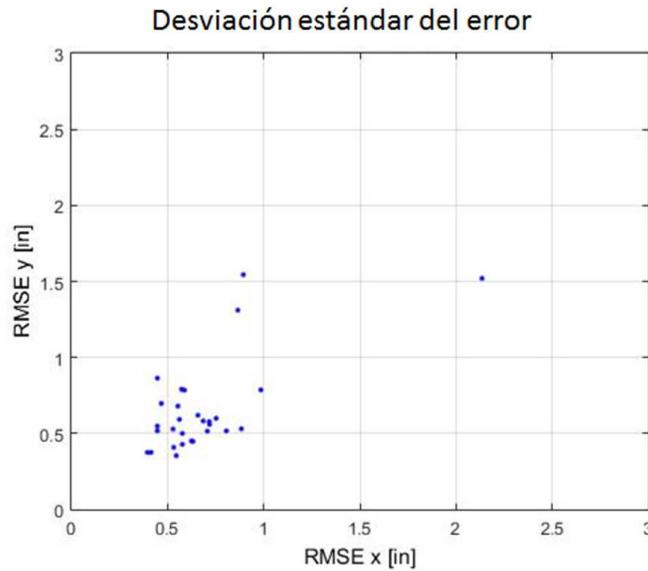


Figura 4.21: Desviación estándar del error por pares de coordenadas.

Para finalizar la sección de resultados, se muestran los gráficos de dispersión 4.20 y 4.21 que describen el error por pares de coordenadas para las 30 trayectorias en conjunto, de esta forma observamos la fusión de los dos ejes y sus respectivos errores, teniendo un panorama general de la diferencia entre lo real y los estimado por el enfoque propuesto, con lo cual es mucho más sencillo determinar la calidad de los resultados.

Durante la experimentación se hicieron algunas observaciones por ejemplo, que es importante considerar una mayor altura para el modelo de fondo neutro (paredes del espacio de pruebas), para evitar falsas detecciones o intromisiones durante el seguimiento de objetos; esto se debe a que parte de la región de interés cubierta por alguna cámara de arreglo incluye áreas externas al entorno controlado, por encontrarse a una distancia lejana de la pared frontal de la misma. El papel que juega la distancia entre el objeto y la cámara es muy importante, ya que si este se encuentra demasiado lejos de alguna de las cámaras sale de su campo de visión efectivo; de igual manera si se encuentra a una distancia muy pequeña es prácticamente imposible centrar el objeto. Debido a que en su primera aproximación el enfoque propuesto no considera la calibración de las cámaras utilizadas; los efectos que tiene la distorsión asociada a cada una de ellas, aumenta de forma proporcional a la distancia que se encuentra el objeto, influyendo en el error de estimación, por lo que la calibración de cámaras es uno de los primeros rubros mencionados en la sección de trabajos futuros.

De igual manera en la sección de trabajos futuros se incluye la aplicación de la metodología presentada, para la estimación de posición en tres dimensiones. En esta sección se describen de manera sencilla algunas consideraciones que se deben realizar para cumplir este objetivo, como el uso de patrones calibrados en las mismas unidades que se manejan en el entorno controlado (para este caso pulgadas), con el fin de tener una referencia formal al ubicar al objeto en una posición tridimensional.

También es necesario colocar las cámaras de tal forma que se puedan tener perspectivas variadas del objeto, sin alterar el funcionamiento efectivo del sistema propuesto. El diseño de trayectorias tridimensionales para la experimentación y la validación de resultados es una consideración importante a realizar para la estimación de posición en tres dimensiones de un objeto. Finalmente, se hace la observación de la importancia que tiene la velocidad de motores de las cámaras, debido a que cuanto mayor sea la rapidez con la que giran dichos motores, menor será el tiempo que tome centrar al objeto en la imagen.

# CONCLUSIONES Y TRABAJO FUTURO

---

## 5.1. Conclusiones

En este trabajo se desarrolló un sistema de visión por computadora, utilizando cámaras Pan & Tilt para la estimación de la posición bidimensional y el seguimiento de objetos. La experimentación se realizó en un espacio de pruebas controlado, aprovechando ampliamente las características de la infraestructura para beneficio del experimento, por ejemplo, las perforaciones de la mesa óptica graduada, donde a cada perforación se le asocia una coordenada  $x, y$  en el plano cartesiano, lo que permite conocer de una manera mas simple la ubicación física de objeto y la posición de las cámaras con respecto al origen en el marco de referencia.

El enfoque presentado ofrece la posibilidad al usuario de manipular las cámaras para el seguimiento de un objeto, por medio de los comandos CGI presentados en capítulos anteriores, de tal manera que el desarrollo de tareas resulte fácil de implementar. Es decir, los comandos CGI permiten personalizar el desempeño de las cámaras, incluso en solicitudes de acceso, implementando niveles de operación con permisos especiales y limitados.

Dado que el trabajo desarrollado es multicámara, es posible hacer frente a oclusiones en el escenario, o a zonas difíciles cubrir por alguna cámara en particular. Sin embargo es recomendable determinar la mejor ubicación de cada cámara para cubrir de la manera más óptima posible el entorno de operación.

De manera formal se presenta un enfoque unificado para la estimación de posición bidimensional y seguimiento de objetos, basado en contornos activos para la detección de objetos, se utiliza la transformada máxima extendida para calcular el centroide del objeto a seguir y compararlo con el centro de imagen, de esta forma se utilizan también los

parámetros Pan & Tilt de las cámaras para centrar el objeto en el campo de visión del arreglo de cámaras y se implementa la pseudoinversa como una forma de solucionar el sistema de ecuaciones generado por los parámetros mencionados anteriormente.

Los datos experimentales son consistentes con respecto a la aseveración inicial, que consiste en comprobar que es posible estimar la posición bidimensional de un objeto, a partir de los parámetros Pan & Tilt de un arreglo de cámaras que lo observan desde diferentes perspectivas en un mismo instante y teniendo conocimiento previo de la ubicación física de cada una de las cámaras en coordenadas locales.

Por otra parte, uno de los requerimientos mínimos del algoritmo es que el objeto sea detectado por al menos tres cámaras a fin de estimar con eficacia su ubicación con respecto al arreglo de cámaras. Esta información se vuelve de suma importancia ya que a partir de ella y mediante el uso de la metodología descrita en este documento es posible dar cumplimiento al objetivo principal de este trabajo.

Finalmente, a través de este trabajo se proporcionan las bases para una evolución del algoritmo, en otras palabras, una vez que se ha probado la hipótesis inicial se puede ahora proceder a la retroalimentación para obtener mejores resultados y acceder a la aplicación de esta metodología para la estimación de la posición tridimensional, además de que se abre paso a la posibilidad de llevar este trabajo a la implementación en escenarios exteriores, donde existe un mayor número de variables no controladas ante las cuales el sistema debe de conservar su eficacia sin permitir un aumento en el error de estimación.

Todo esto se traduce en una amplia gama de posibles aplicaciones, por ejemplo: el seguimiento de objetos y estimación de la posición en robots móviles, la detección y el seguimiento de objetos en una banda transportadora, estimación de coordenadas en el mundo real, para superposición de objetos con realidad aumentada, así como fructuosas aplicaciones en el campo de video vigilancia y seguridad.

## 5.2. Trabajos futuros

Como prosecución a esta tesis se proponen los siguientes puntos:

- Una vez que se ha comprobado que es posible realizar la estimación de la posición a partir de los parámetros Pan & Tilt de un arreglo de cámaras que observa a un objeto desde diferentes perspectivas, sería muy interesante realizar la estimación de la posición en coordenadas tridimensionales dentro de un entorno controlado, ubicando tanto el cuerpo como el arreglo de cámaras en posiciones espaciales.
- El uso de zoom en cada cámara es un factor de gran importancia, que introduce otras variables al desarrollo de la aplicación, por lo que analizar trabajos en el estado del arte que cuenten con esta característica es muy importante, así como estudiar sus implicaciones y verificar sus efectos en el sistema desarrollado.
- Por otra parte un trabajo futuro para obtener mayor exactitud al estimar la posición, es la calibración de cámaras de forma individual, ya que esto permite conocer la relación entre el giro de la cámara y los píxeles de imagen con mayor precisión.
- A su vez la calibración de las cámaras en conjunto no solo da un mayor conocimiento de la relación giro de motor-píxeles, sino que también proporciona información sobre la geometría del escenario en relación con el mundo real, de esta forma se eliminan las posibles distorsiones del arreglo de cámaras con respecto a lo mencionado anteriormente.
- La migración del sistema desarrollado a entornos exteriores como oficinas, laboratorios, almacenes, etc. representa un reto debido a que no se tiene tanto control sobre las condiciones lumínicas, y a que debe encontrarse la manera de discriminar el movimiento de fondo del ocasionado por la rotación de las cámaras, no obstante el encontrar una alternativa o solución a dicha problemática es un trabajo a futuro muy interesante que se traduce en un excelente aporte al estado del arte.
- Otro trabajo futuro de gran interés es la aplicación de la metodología propuesta a seguimiento de personas, pues se puede aprovechar al máximo la técnica de contornos

activos, para hacer frente al cambio de forma en la silueta de los seres humanos mientras se desplazan de un punto a otro, con lo que se reduce el trabajo de cada cámara al seguimiento del contorno activo.

---

# Bibliografía

---

- [1] A. Yilmaz, O. Javed, y M. Shah, “Object tracking: A survey,” *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.
- [2] J. Fernández Gutiérrez, “Sistema de video vigilancia semántico basado en movimiento. aplicación a la seguridad y control de tráfico,” Master’s thesis, Universidad de Valladolid. Escuela Técnica Superior de Ingenieros de Telecomunicación, Julio 2013.
- [3] D. P. Chau, F. Bremond, y M. Thonnat, “Object tracking in videos: Approaches and issues,” *arXiv preprint arXiv:1304.5212*, no. 1, p. 10, 2013.
- [4] K. Kim y L. S. Davis, “Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering,” in *European Conference on Computer Vision*. Springer, 2006, pp. 98–109.
- [5] R. Goldenberg, R. Kimmel, E. Rivlin, y M. Rudzsky, “Fast geodesic active contours,” *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1467–1475, 2001.
- [6] Q. Xiaoping, Z. Qiheng, O. Yimin, y M. Jiaguang, “A method for object tracking using shape matching,” in *2006 IEEE Workshop on Signal Processing Systems Design and Implementation*, Oct 2006, pp. 372–376.
- [7] Z. Jin y B. Bhanu, “Analysis-by-synthesis: Pedestrian tracking with crowd simulation models in a multi-camera video network,” *Computer Vision and Image Understanding*, vol. 134, pp. 48–63, 2015.

- [8] M. N. Ali, M. Abdullah-Al-Wadud, y S.-L. Lee, “Multiple object tracking with partial occlusion handling using salient feature points,” *Information Sciences*, vol. 278, pp. 448–465, 2014.
- [9] L. Maddalena, A. Petrosino, y F. Russo, “People counting by learning their appearance in a multi-view camera environment,” *Pattern Recognition Letters*, vol. 36, pp. 125–134, 2014.
- [10] E. Reyes, “Detección y medición de distancia de un objeto utilizando dos cámaras móviles,” *Centro de ingeniería y desarrollo industrial*, vol. 1, pp. 1–37, 2014.
- [11] N. Paragios y R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 3, pp. 266–280, 2000.
- [12] A. K. Mishra, B. Ni, S. Winkler, y A. Kassim, “3d surveillance system using multiple cameras,” in *Electronic Imaging 2007*. International Society for Optics and Photonics, 2007, pp. 649 103–649 103.
- [13] R. Mohedano, C. R. Del-Blanco, F. Jaureguizar, L. Salgado, y N. García, “Robust 3d people tracking and positioning system in a semi-overlapped multi-camera environment,” in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 2656–2659.
- [14] S. Robla, J. Llata, C. Torre, y E. Sarabia, “An approach for tracking oil slicks by using active contours on satellite images,” in *OCEANS 2009-EUROPE*. IEEE, 2009, pp. 1–8.
- [15] C. Losada, M. Mazo, S. Palazuelos, D. Pizarro, y M. Marrón, “Multi-camera sensor system for 3d segmentation and localization of multiple mobile robots,” *Sensors*, vol. 10, no. 4, pp. 3261–3279, 2010.
- [16] H. L. Paredes, “Detección y seguimiento de objetos con cámaras en movimiento,” *Proyecto fin de carrera, Universidad Autónoma de Madrid*, 2011.

- [17] A. Utasi y C. Benedek, “A 3-d marked point process model for multi-view people detection,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3385–3392.
- [18] Y. Lee y A. Yilmaz, “Real-time object detection, tracking, and 3d positioning in a multiple camera setup,” in *The ISPRS Workshop on Image Sequence Analysis*, vol. 55, 2013, p. 56.
- [19] S. J. You, P. H. Truong, S. H. Ji, S. M. Lee, C. E. Lee, y Y. J. Cho, “A cooperative multi-camera system for tracking a fast moving object,” in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2014 IEEE 4th Annual International Conference on*. IEEE, 2014, pp. 141–145.
- [20] S. Ballesta, G. Reymond, M. Pozzobon, y J. Duhamel, “A real-time 3d video tracking system for monitoring primate groups,” *Journal of neuroscience methods*, vol. 234, pp. 147–152, 2014.
- [21] C. Benedek, “3d people surveillance on range data sequences of a rotating lidar,” *Pattern Recognition Letters*, vol. 50, pp. 149–158, 2014.
- [22] M. G. Chirinos Colunga, “Visión artificial: Percepción de profundidad,” Master’s thesis, Instituto Tecnológico de Mérida, Noviembre 2004.
- [23] M. Kass, A. Witkin, y D. Terzopoulos, “Snakes: Active contour models,” *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [24] R. Kouskouridas y A. Gasteratos, “Location assignment of recognized objects via a multi-camera system,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 4, no. 3, pp. 1–18, 2011.
- [25] A. Tyagi, G. Potamianos, J. W. Davis, y S. M. Chu, “Fusion of multiple camera views for kernel-based 3d tracking,” in *Motion and Video Computing, 2007. WMVC '07. IEEE Workshop on*, Feb 2007, pp. 1–1.

- [26] E. Santillán y C. A. Cruz, “Detección y clasificación de objetos dentro de un salón de clases empleando técnicas de procesamiento digital de imágenes,” *Universidad Autónoma Metropolitana, México, Tesis de Maestría*, 2008.
- [27] M. A. CHAPA MARTELL, “Implementacion de operaciones aritmeticas en fpga, rapida y de bajo costo aplicado a la normalizacion de imagen en huellas dactilares,” Ph.D. dissertation, 2010.
- [28] F. G. R. Samaniego y R. L. Q. Rojas, “Generación automática de operaciones de segmentación de imágenes basadas en algoritmos evolutivos multi-objetivo.”
- [29] J. Elizondo y L. Maestre, “Fundamentos de procesamiento de imágenes,” *Documentación Universidad Autónoma de Baja California, Unidad Tijuana*, 2002.
- [30] A. A. . A. F. Pinilla, C., “Filtrado de imágenes en el dominio de la frecuencia,” *Revista de la Asociación Española de Teledetección*, 1997.
- [31] R. C. González, R. E. Woods, F. D. Rodríguez, y L. Rosso, *Tratamiento digital de imágenes*. Addison-Wesley New York, 1996, vol. 3.
- [32] B. Aldalur y M. Santamaría, “Realce de imágenes: filtrado espacial,” *Revista de teledetección*, vol. 17, pp. 31–42, 2002.
- [33] A. K. Jain, *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989.
- [34] A. Rosenfeld y J. L. Pfaltz, “Distance functions on digital pictures,” *Pattern recognition*, vol. 1, no. 1, pp. 33–61, 1968.
- [35] M. Á. Cazorla, P. Cáceres, F. Escolano, D. Gallardo, y R. Rizo, “Detección automática con snakes y representación 3d sobre imágenes cerebrales,” *VI CAEPIA*, pp. 331–340, 1995.
- [36] S. R. Gunn y M. S. Nixon, “Global and local active contours for head boundary extraction,” *International Journal of Computer Vision*, vol. 30, no. 1, pp. 43–54, 1998.

- [37] T. F. Cootes, A. Hill, C. J. Taylor, y J. Haslam, “Use of active shape models for locating structures in medical images,” *Image and vision computing*, vol. 12, no. 6, pp. 355–365, 1994.
- [38] J. Denzler y H. Niemann, “3d data driven prediction for active contour models with application to car tracking.” in *MVA*, 1996, pp. 204–207.
- [39] J. Deutscher, A. Blake, y I. Reid, “Articulated body motion capture by annealed particle filtering,” in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2. IEEE, 2000, pp. 126–133.
- [40] F. Leymarie y M. D. Levine, “Tracking deformable objects in the plane using an active contour model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 617–634, 1993.
- [41] A. Lanitis, C. J. Taylor, y T. F. Cootes, “Automatic face identification system using flexible appearance models,” *Image and vision computing*, vol. 13, no. 5, pp. 393–401, 1995.
- [42] J. F. Vélez, n. Sánchez, A. B. Moreno, y J. L. Esteban, “Verificación offline de firmas manuscritas: Una propuesta basada en snakes paramétricos,” *Simposio de Reconocimiento de Formas y Análisis de Imágenes*, 2005.
- [43] M. Della Rocca, M. Fiani, A. Fortunato, y P. Pistillo, “Active contour model to detect linear features in satellite images,” *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. B3, pp. 446–450, 2004.
- [44] M. Rochery, I. H. Jermyn, y J. Zerubia, “Higher order active contours and their application to the detection of line networks in satellite imagery,” in *Proc. IEEE Workshop Variational, Geometric and Level Set Methods in Computer Vision*, 2003.
- [45] P. Horvath, I. H. Jermyn, Z. Kato, y J. Zerubia, “A higher-order active contour model

- of a ‘gas of circles’ and its application to tree crown extraction,” *Pattern Recognition*, vol. 42, no. 5, pp. 699–709, 2009.
- [46] P. Horváth, I. Jermyn, J. Zerubia, y Z. Kato, “A higher-order active contour model for tree detection,” in *18th International Conference on Pattern Recognition (ICPR’06)*, vol. 2. IEEE, 2006, pp. 130–133.
- [47] D. J. Williams y M. Shah, “A fast algorithm for active contours,” in *Computer Vision, 1990. Proceedings, Third International Conference on*. IEEE, 1990, pp. 592–595.
- [48] J. M. L. VALENZUELA, “Comparación entre contornos activos y transformada de hough, para la detección de círculos en imágenes digitales,” 2007.
- [49] A. A. Amini, T. E. Weymouth, y R. C. Jain, “Using dynamic programming for solving variational problems in vision,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 12, no. 9, pp. 855–867, 1990.
- [50] J. E. B. BOLÍVAR, A. J. B. ARANGO, y M. B. ARBELÁEZ, “El metodo de newton-raphson-la alternativa del ingeniero para resolver sistemas de ecuaciones no lineales,” *Scientia Et Technica*, vol. 11, no. 27, pp. 221–224, 2005.
- [51] T. J. Ypma, “Historical development of the newton-raphson method,” *SIAM review*, vol. 37, no. 4, pp. 531–551, 1995.
- [52] D. S. C. Federico y N. H. Antonio, *Métodos numéricos aplicados a la ingeniería*. Grupo Editorial Patria, 2014.
- [53] P. P. Beatriz, “Análisis de varias variables reales,” Universidad de Cantabria, Noviembre 2008, departamento de Matemáticas, Material de clase.
- [54] E. Kurmyshev, *Fundamentos de métodos matemáticos para física e ingeniería*. Editorial Limusa, 2003.
- [55] J. M. C. Villalba, *Introducción al álgebra lineal*. ESIC Editorial, 2004.

- [56] C. A. López, “Cátedra de matemática y elementos de matemática,” Universidad nacional de la Plata, departamento de Matemáticas, Material de clase.
- [57] P. Soille, *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

# Apéndices

# VALIDACIÓN DE TRAYECTORIAS

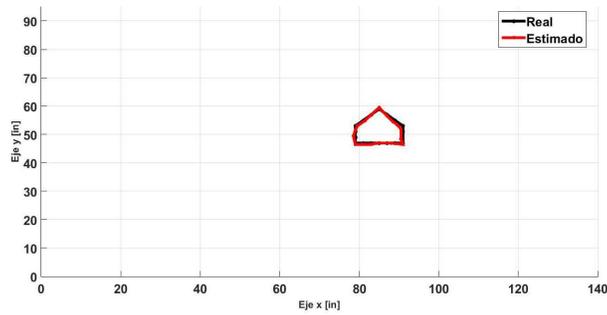


Figura A.1: Trayectoria en forma de casa.

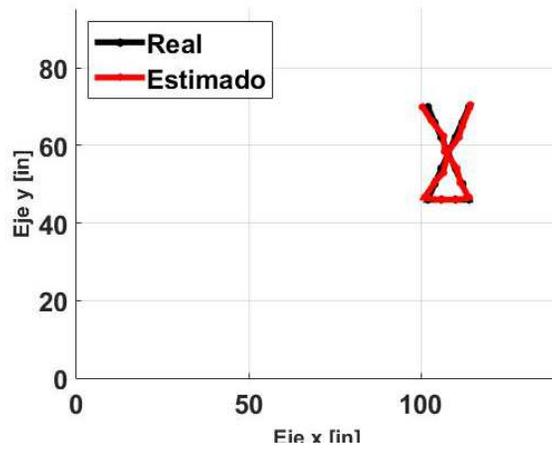


Figura A.2: líneas cruzadas.

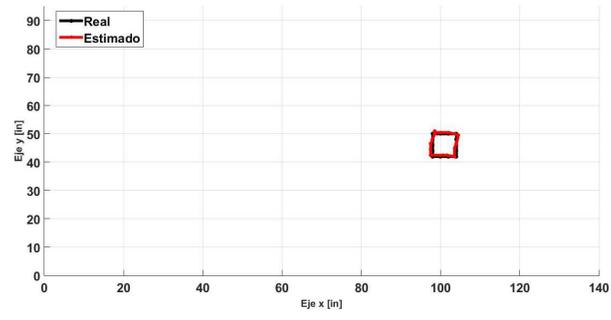


Figura A.3: Trayectoria en cuadro.

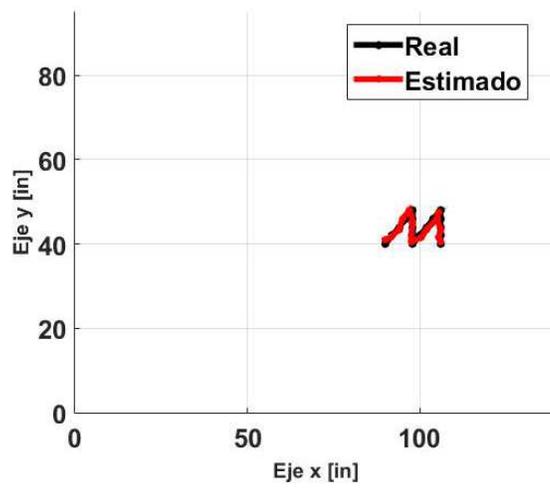


Figura A.4: Diente de sierra.

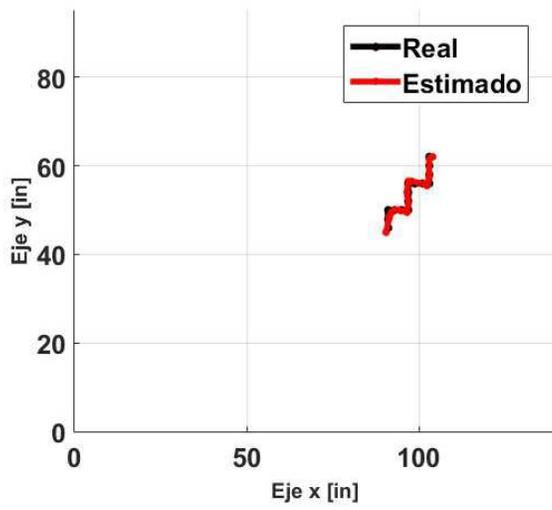


Figura A.5: escalones.

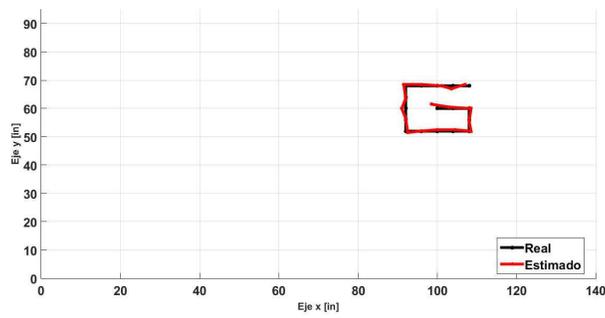


Figura A.6: Trayectoria en forma de G.

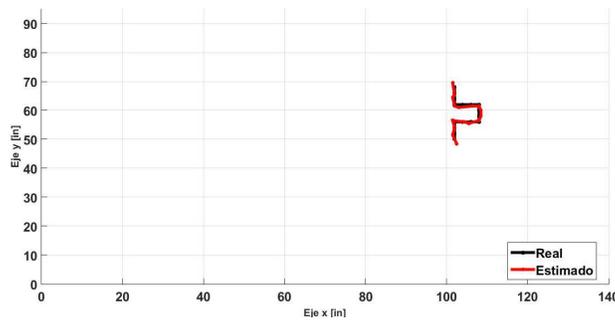


Figura A.7: Trayectoria en forma de herradura.

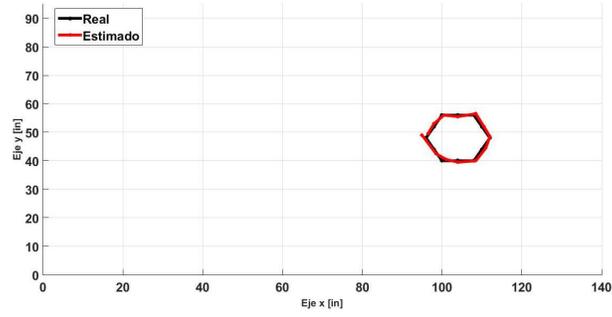


Figura A.8: Hexágono.

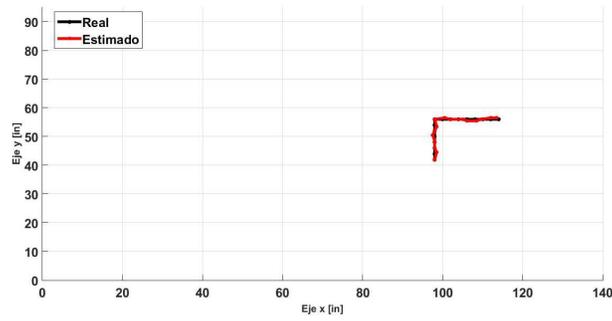


Figura A.9: Trayectoria en forma de L.

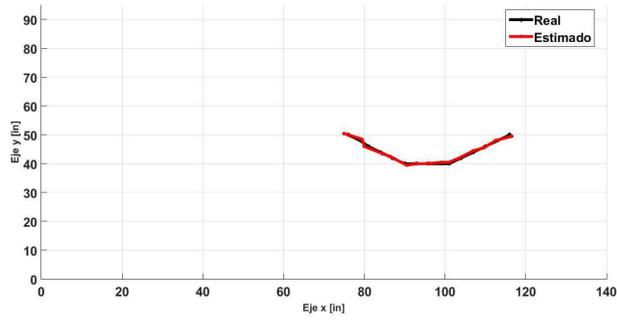


Figura A.10: Medio hexágono.

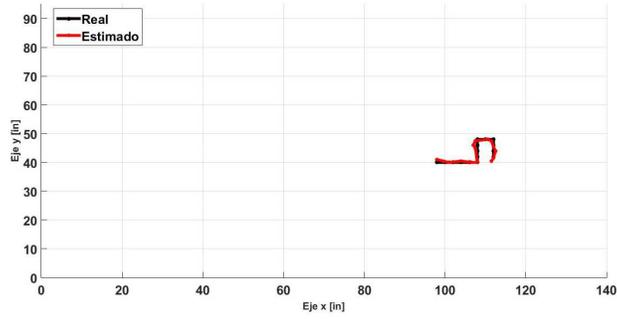


Figura A.11: Onda cuadrada.

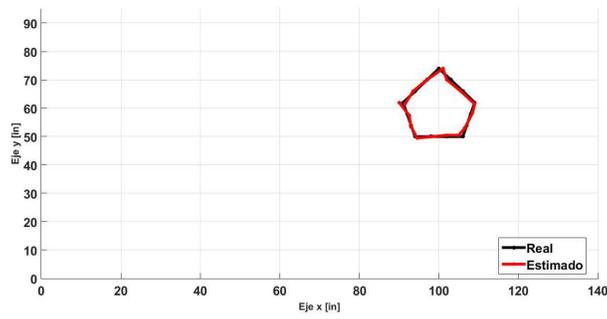


Figura A.12: Trayectoria en forma de pentágono.

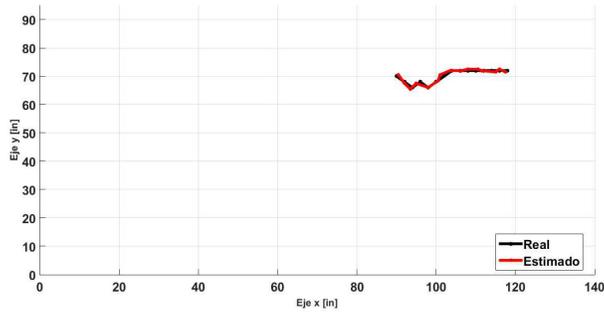


Figura A.13: Trayectoria en forma de raíz.

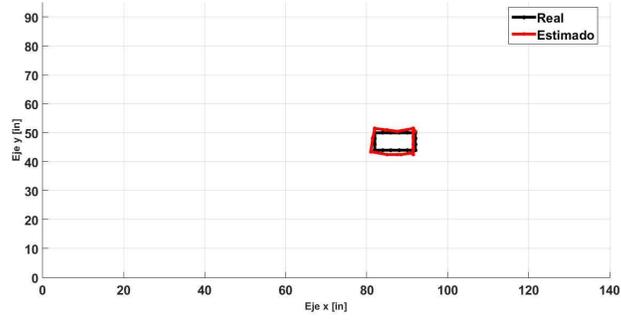


Figura A.14: Rectángulo.

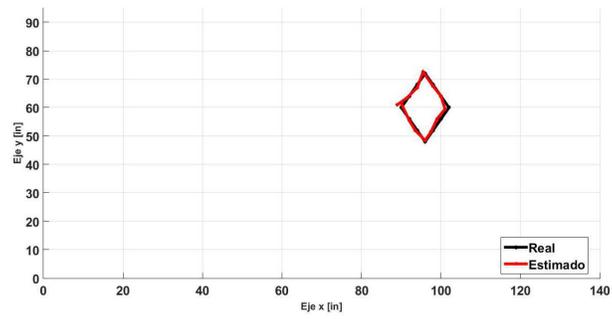


Figura A.15: Rombo.

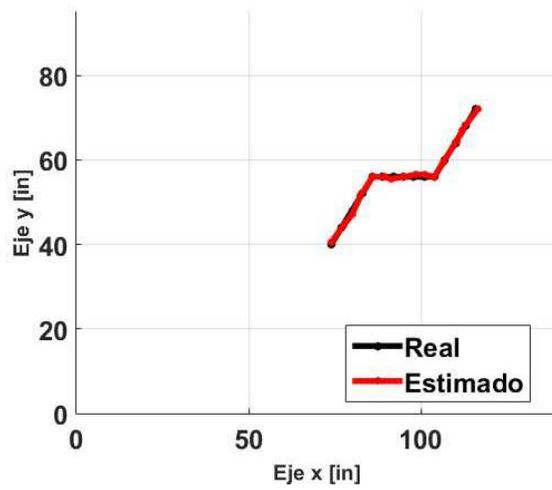


Figura A.16: Trayectoria sin forma definida.

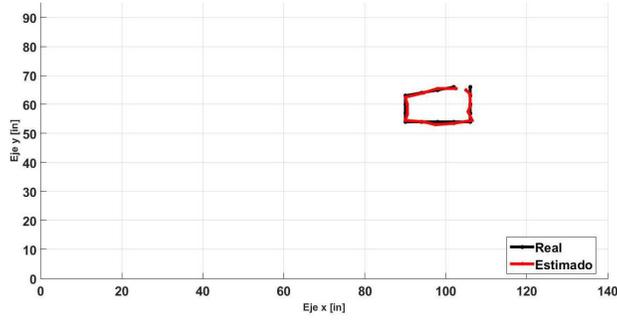


Figura A.17: Trayectoria irregular.

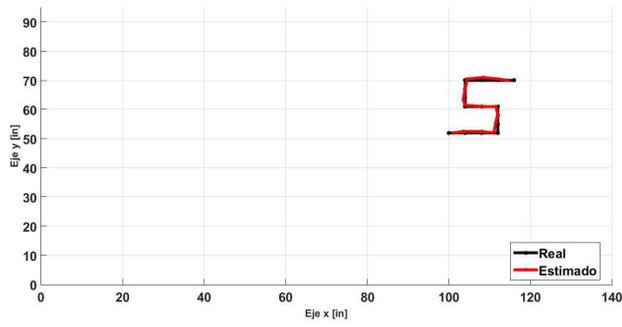


Figura A.18: Ruta de objeto en forma de S.

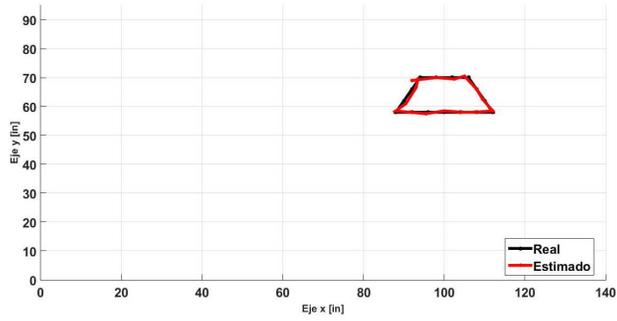


Figura A.19: Trapecio.

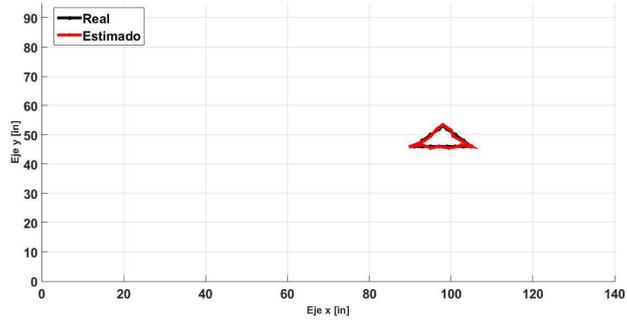


Figura A.20: Trayectoria con forma de triángulo.

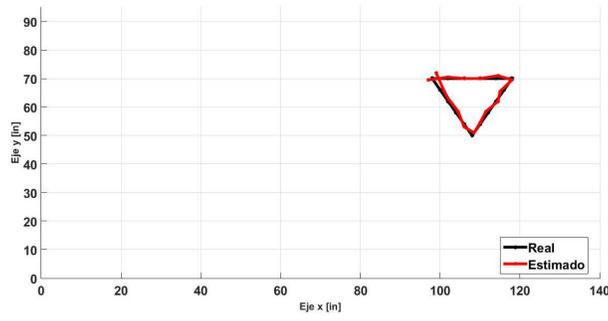


Figura A.21: Trayectoria A.20 invertida.

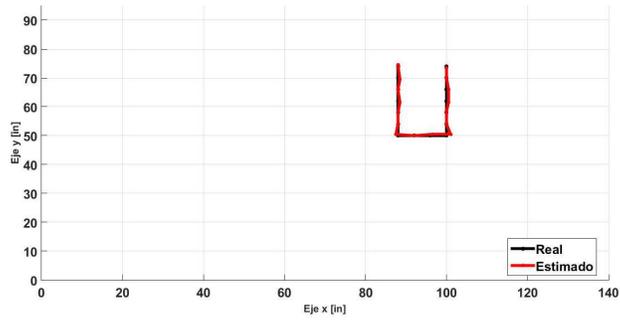


Figura A.22: Ruta de objeto en forma de U.

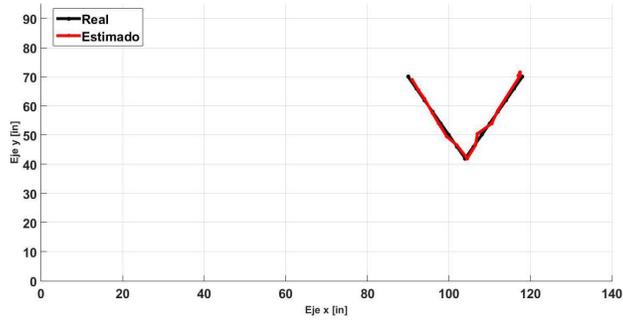


Figura A.23: Trayectoria en forma de V.

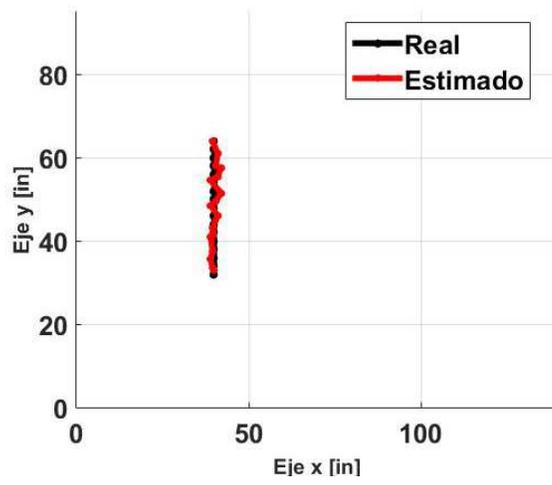


Figura A.24: Línea recta vertical.

# CAD DEL ESPACIO DE PRUEBAS

---

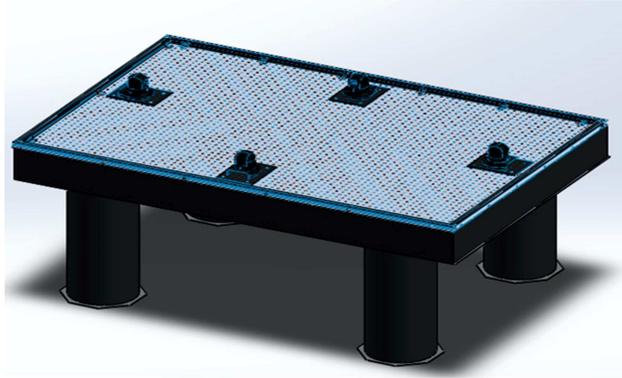


Figura B.1: Espacio de pruebas sin fondo neutro.

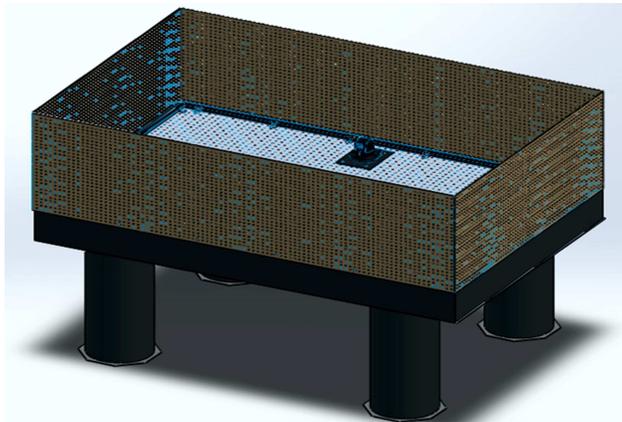


Figura B.2: Espacio de pruebas con fondo neutro.

# TABLAS DE ERROR Y ECM

TRAYECTORIA	$\varepsilon_x$	$\varepsilon_y$	$ECM_x$	$ECM_y$
CASA	0.48	0.44	0.72	0.58
CRUZADA	0.59	0.44	0.72	0.56
CUADRADO	0.40	0.66	0.45	0.86
D	0.63	0.77	0.59	1.06
DIAGONAL	0.60	0.46	0.41	0.33
DIENTE SIERRA	0.38	0.44	0.53	0.53
ESCALERA	0.43	0.43	0.56	0.59
ESCALÓN	0.69	0.44	0.88	0.53
G	0.53	0.44	0.69	0.58
HERRADURA	0.38	0.53	0.47	0.70
HEXÁGONO	0.46	0.62	0.59	0.78
HORIZONTAL	0.60	0.49	0.45	0.35
L	0.28	0.28	0.41	0.38
M	0.53	0.50	0.66	0.62
MEDIO HEXÁGONO	0.44	0.33	0.53	0.41
ONDA CUADRADA	0.60	1.17	0.89	1.54

TRAYECTORIA	$\varepsilon_x$	$\varepsilon_y$	$ECM_x$	$ECM_y$
PENTÁGONO	0.53	0.34	0.63	0.45
RAÍZ	0.47	0.37	0.58	0.43
RECTÁNGULO	1.53	1.32	2.14	1.52
ROMBO	0.38	0.54	0.55	0.68
RUTA 1	0.33	0.42	0.45	0.52
RUTA 2	0.33	0.42	0.45	0.52
RUTA 3	0.33	0.46	0.45	0.55
S	0.47	0.34	0.63	0.45
TRAPECIO	0.50	0.41	0.71	0.52
TRIÁNGULO	0.41	0.25	0.54	0.35
TRIANGULO INVERTIDO	0.41	0.63	0.57	0.79
U	0.25	0.28	0.40	0.38
V	0.47	0.43	0.58	0.50
VERTICAL	0.82	0.65	0.99	0.79

# HOJA DE DATOS CÁMARA PAN & TILT

---

- FOSCAM modelo FI8918W
- Powerful high-speed video protocol processor
- High Definition Color CMOS Sensor
- 300K Pixels
- IR night vision (Range: 8m)
- Pan 300 degree, tilt 120 degree
- Optimized MJPEG video compression for transmission
- Multi-level users' management and passwords definition
- Embedded Web Server for users to visit by IE
- Wi-Fi compliant with wireless standards IEEE 802.11b/g/n
- Embedded FOSCAM domain name
- Supports Dynamic IP (DDNS) and UPnP LAN and Internet (ADSL, Cable Modem)
- Motion and Sound detection activates alarm
- Supports image snapshot
- Supports multiple network protocols:  
HTTP/TCP/IP/UDP/STMP/DDNS/SNTP/DHCP/FTP
- Supports WEP/WPA/WPA2 encryption

- Supports WPS(Wi-Fi Protected Set-up)
- Supports Daylight Saving Time
- Supports MSN
- Supports Gmail as sender on mail service settings
- Supports audio on Firefox, Google Chrome and Safari
- Providing Phone APPs for Android and iPhone users
- Providing Software development kit