

DETECCIÓN Y SEGUIMIENTO DINÁMICO DE LA PUPILA PARA MANIPULACIÓN DE GUI'S



**CARLOS MARIO EMBUS CASAS
JADIR CAMILO PAZ MARTÍNEZ**

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE INVESTIGACIÓN DE CONTROL
POPAYÁN
2013

DETECCIÓN Y SEGUIMIENTO DINÁMICO DE LA PUPILA PARA MANIPULACIÓN DE GUI'S

Carlos Mario Embus Casas
Jadir Camilo Paz Martínez

Trabajo de Grado para Optar al Título de
Ingeniero en Automática Industrial

Director:
Elena Muñoz España
Ingeniera en Electrónica y Telecomunicaciones

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE INVESTIGACIÓN DE CONTROL
POPAYÁN
2013

TABLA DE CONTENIDO

INTRODUCCIÓN	1
Capítulo 1. CONCEPTOS GENERALES	3
1.1 Antecedentes.....	3
1.2 El ojo humano.....	5
1.3 La cámara web como sensor.....	7
1.4 Visión artificial.....	8
Capítulo 2. FILTRO DE PARTÍCULAS.....	11
2.1 Fundamentos del filtro de partículas.....	11
2.2 Filtro de partículas monomodal.....	14
2.2.1 Modelos involucrados.....	14
2.2.2 Funcionamiento del Filtro de partículas.....	16
2.3 Filtro de partículas multimodal.....	24
2.3.1 Algoritmo de las K-Medias [12].....	26
2.3.2 Recalcular pesos.....	28
Capítulo 3. ALGORITMO INTELIGENTE.....	32
3.1 Segmentación por forma.....	32
3.2 Búsqueda Dispersa.....	34
3.2.1 Método de Combinación.....	36
3.2.2 Método de mejora.....	37
3.3 Hibridación del Filtro de Partículas con la Búsqueda Dispersa.....	38
Capítulo 4. SISTEMA DE COORDENADAS INCREMENTALES E INTERFAZ	41
4.1 Sistema de Coordenadas Incrementales.....	41
4.2 Interfaz.....	44
Capítulo 5. PLATAFORMA EXPERIMENTAL, PRUEBAS Y RESULTADOS	46
5.1 Creación de la cámara IR-LED	46
5.2 Disposición Hardware del sistema de seguimiento ocular.....	47
5.3 Descripción Informática.....	48

5.4	Requisitos de Desarrollo	49
5.5	Implementación del Software.	49
5.6	Pruebas y resultados.....	51
5.6.1	Pruebas de detección.....	51
5.6.2	Pruebas de seguimiento y precisión.....	52
5.6.3	Pruebas de posicionamiento.....	58
Capítulo 6.	CONCLUSIONES Y TRABAJOS FUTUROS	61
BIBLIOGRAFÍA.....		64

LISTA DE IMÁGENES

<i>Imagen 1.1 Vista frontal del ojo (adoptado de [22]).</i>	6
<i>Imagen 1.2 Sección del ojo humano (adoptado de [22]).</i>	6
<i>Imagen 1.3 Medición del movimiento mediante electro-oculografía (adoptado de [25]).</i>	7
<i>Imagen 1.4 a) Pupila brillante, b) Pupila oscura (adoptado de [26]).</i>	8
<i>Imagen 1.5 Histogramas de color en HSV.</i>	10
<i>Imagen 2.1 Imagen del modelo de la pupila del ojo</i>	16
<i>Imagen 2.2 Representación de una partícula.</i>	17
<i>Imagen 2.3 Inicialización de las partículas.</i>	17
<i>Imagen 2.4 Representación de las partículas de mayor peso.</i>	21
<i>Imagen 2.5 Resultado del seguimiento monomodal.</i>	23
<i>Imagen 2.6 Representación de las zonas de interés mediante una fdp.</i>	25
<i>Imagen 2.7 Resultado del seguimiento multimodal.</i>	31
<i>Imagen 3.1 Imágenes procesadas. a) Imagen Original en escala de gris, b) Imagen resultado de aplicar el filtro de Canny.</i>	33
<i>Imagen 3.2 Segmentación por forma. a) Imagen original, b) Imagen con forma circular (color rojo) encontrada con la Transformada de Hough.</i>	33
<i>Imagen 3.3 Funcionamiento del filtro de partículas híbrido.</i>	40
<i>Imagen 5.1 Fuente de luz infrarroja.</i>	47
<i>Imagen 5.2 Disposición hardware de la aplicación.</i>	48
<i>Imagen 5.3. Interfaz de la aplicación.</i>	50
<i>Imagen 5.4 Secuencia de imágenes para búsqueda.</i>	51
<i>Imagen 5.5 Secuencia de vídeo para medir precisión.</i>	54

LISTA DE FIGURAS

<i>Figura 2.1 Etapas del Filtro de Partículas.</i>	13
<i>Figura 2.2 Representación gráfica de un filtro de partículas (adoptado de [6]).</i>	14
<i>Figura 2.3 Ejemplo de creación de la ruleta.</i>	20
<i>Figura 2.4 Diagrama de Bloques del Filtro de Partículas</i>	22
<i>Figura 2.5 Diagrama funcional del K-Medias Básico.</i>	27
<i>Figura 2.6 Campana de Gauss para el cálculo de peso.</i>	28
<i>Figura 2.7 Diagrama de Bloques del Filtro de Partículas</i>	29
<i>Figura 3.1 Diagrama de estados de la Búsqueda Dispersa</i>	35
<i>Figura 3.2 Diagrama de Combinación de soluciones.</i>	36
<i>Figura 3.3 Combinación de dos soluciones.</i>	37
<i>Figura 3.4 Representación de los 8-vecinos.</i>	38
<i>Figura 3.5 Diagrama de bloques del Filtro de Partículas Híbrido.</i>	39
<i>Figura 4.1 Sistema de coordenadas.</i>	41
<i>Figura 4.2 Diagrama del Sistema de Coordenadas Incrementales [Adoptado de 13]</i>	42
<i>Figura 4.3 Ángulo entre dos rectas</i>	43
<i>Figura 5.1 Modelo del ciclo de vida en espiral.</i>	49
<i>Figura 5.2 Distancia entre las pupilas.</i>	53
<i>Figura 5.3 Resultados de precisión</i>	54
<i>Figura 5.4 Resultados de precisión</i>	55
<i>Figura 5.5 Resultados de Los filtros frente a cambios de velocidad.</i>	56
<i>Figura 5.6 Respuesta de los filtros frente a cambios de velocidad.</i>	57
<i>Figura 5.7 Posicionamiento del cursor en la pantalla.</i>	58
<i>Figura 5.8 Posicionamiento del cursor en la pantalla, prueba 2.</i>	59

LISTA DE TABLAS

<i>Tabla 4.1 Tabla de ajuste del vector M.</i>	43
<i>Tabla 5.1 Datos de detección de los objetivos.</i>	52
<i>Tabla 5.2 Rango de fotogramas y velocidades.</i>	57

RESUMEN

El presente trabajo está dirigido a la creación de una herramienta informática para la detección y seguimiento de la pupila del ojo, que permita fijar la mirada del usuario en una pantalla en la cual pueda manipular una interfaz gráfica de usuario.

La técnica utilizada para la detección y el seguimiento dinámico de la pupila consiste en el filtro de partículas básico, luego de detallar sus etapas se evalúa su funcionamiento en una secuencia de imágenes. Posteriormente, se mejora esta solución básica del filtro implementando una técnica para realizar seguimiento multimodal, es decir de varios objetos, que da paso al desarrollo del algoritmo de búsqueda dispersa, con el que se optimiza el filtro y se mejoran los resultados. La hibridación entre filtro y la técnica de optimización se implementa de tal manera que se obtengan los mejores resultados en seguimiento y detección con el menor coste computacional posible, representando esto el principal aporte del presente trabajo.

De otro lado, se hace un estudio de la técnica de coordenadas incrementales para realizar el posicionamiento de la mirada en la pantalla del usuario. Finalmente se presentan todos los resultados obtenidos con la aplicación desarrollada y el análisis de los mismos.

INTRODUCCIÓN

Día a día se hace necesario el uso de la tecnología informática, ya que el mundo actual gira en torno a la computación y el acceso a medios virtuales; no solo para facilitar las tareas, sino para expandir los conocimientos, entretenerse, entre otros aspectos.

El manejo de un computador se convierte entonces, en un evento fundamental para acceder a dicha tecnología, usando sus periféricos de entrada como son el teclado o el mouse, se puede manipular la máquina y apoderarse de los beneficios que esta brinda. Para ello se necesita de por lo menos una extremidad funcional del cuerpo, por lo que personas con discapacidades motoras, como por ejemplo personas tetrapléjicas, resultan marginadas en cierta forma.

En este sentido, la presente investigación se enfoca en el desarrollo de una herramienta que permita la interacción de personas con un ordenador mediante el seguimiento ocular, siendo esto un importante avance tecnológico que amplía el tipo de usuarios y mejora la calidad de vida de los mismos, sin grandes costos en dispositivos tecnológicos y sin necesidad de intervenciones médicas en este tipo de usuarios con discapacidad. Además, las oportunidades que brinda una herramienta como esta pueden ser diversas, como el entretenimiento, la comunicación, el acceso a información, estudios, trabajo, etc.

Para lograr el desarrollo de la herramienta, se dispuso de un sistema de captura de imágenes conformado por una cámara web, una lente vari-focal y una fuente de luz infrarroja, obteniendo así las imágenes adecuadas para ejecutar el seguimiento ocular. Este último se realiza mediante la técnica bayesiana del filtro de partículas, la cual fue adaptada al propósito de este trabajo mediante la inserción de un algoritmo de clasificación, como lo es el K-Medias y un algoritmo inteligente, como lo es la búsqueda dispersa. La hibridación entre estos algoritmos permite un seguimiento multimodal que detecta las pupilas de los ojos de forma eficaz y posibilita la ubicación de la mirada en la pantalla de usuario, siendo esto el principal aporte con el desarrollo de este trabajo.

Por otra parte, este trabajo está organizado en VI capítulos y dos anexos, cuyo contenido es el siguiente; En el primer capítulo se realiza una revisión de algunos trabajos y técnicas relacionadas con el tema de esta investigación, donde se resaltan los aportes tomados de cada uno y las diferencias establecidas con ellos. En el segundo capítulo se realiza una

descripción detallada del Filtro de Partículas, en su forma monomodal y multimodal, las diferentes etapas que lo componen y su funcionamiento en una secuencia de imágenes.

En el tercer capítulo se describe en forma detallada el algoritmo inteligente utilizado para mejorar la respuesta del Filtro de Partículas, las etapas que lo componen, así como también el funcionamiento dentro del Filtro y los resultados obtenidos.

En el cuarto capítulo se hace mención de la técnica de Coordenadas Incrementales, detallando su funcionamiento e interacción con el Filtro para posicionar la mirada en la pantalla. También se describe la interfaz utilizada para validar el funcionamiento de la técnica.

El quinto capítulo se refiere a la implementación llevada a cabo para conseguir el seguimiento dinámico de la pupila, los experimentos realizados y los resultados obtenidos. Por último, en el sexto capítulo se presentan las conclusiones obtenidas durante el desarrollo de este trabajo.

En el primer anexo se realiza una descripción del funcionamiento de la aplicación, detallando los requerimientos de máquina para un buen desempeño, los pasos a seguir para instalar el programa en un ordenador y las limitaciones de movimiento del usuario. En el segundo se realiza la descripción de las funciones y procedimientos desarrollados mediante el pseudocódigo.

Capítulo 1.

CONCEPTOS GENERALES

En este capítulo se realiza una revisión de temas relacionados con el presente trabajo, incluyendo algunos conceptos básicos asociados al estudio, como son el ojo humano y sus partes, la visión artificial y la cámara web cómo sensor.

1.1 Antecedentes.

Durante los últimos años se ha despertado un especial interés en el estudio de los algoritmos de seguimiento por parte de la comunidad investigadora, esto gracias a sus múltiples aplicaciones y prestaciones, algunos de sus usos más importantes son: supervisión o vigilancia automática, interacción hombre-computador, monitorización del tráfico, navegación de vehículos y reconocimiento basado en movimiento.

Una técnica de seguimiento que ha recibido una atención considerable es la familia de los filtros de partículas propuesto por Gordon, Salmond y Smith en 1993[1], como lo menciona Salih y compañía en [2], entre los años 2006 – 2010, más del 50% de los trabajos registrados que presentaban como tema principal el seguimiento dinámico en secuencia de imágenes se realizaron utilizando el filtro de partículas, convirtiéndola en una técnica popular para el seguimiento visual.

En [3] se presenta una aplicación para la predicción de la trayectoria de un móvil utilizando el filtro de partículas, este trabajo permite observar el potencial del filtro para la predicción de un sistema, pero también deja en evidencia la importancia de configurar adecuadamente los modelos involucrados, principalmente el modelo del objeto a seguir, puesto que de ellos depende la eficiencia del mismo. También se remarca en [4] la importancia de definir adecuadamente la estructura del filtro, en este trabajo se presentan sugerencias del modo adecuado de establecer un filtro de partículas para su implementación. Otro aspecto importante del filtro es su flexibilidad, como se menciona [5], muchos algoritmos auxiliares y técnicas, cómo la inteligencia de enjambre aplicado en el mencionado trabajo, permiten

mejorar el funcionamiento del filtro y adaptarlo a los requerimientos de uso. También se ha profundizado los estudios en hibridación del filtro, es decir, añadirle una etapa de optimización al filtro convencional para mejorar su respuesta, por ejemplo, en [6] se agrega una etapa metaheurística, para mejorar las soluciones en la fase de exploración, pero este trabajo va orientado a encontrar la solución de una función objetivo que permita encontrar el mejor camino al problema del vendedor viajero (TPD en inglés). En [7] se proponen una variedad de métodos con los que se puede realizar una hibridación con el filtro de partículas, tales como algoritmos genéticos, meméticos [8], metaheurísticas poblacionales y trayectorias, entre otros, destacando que dependiendo de la complejidad del problema a solucionar, se debe escoger el método adecuado.

El seguimiento de objetos en una secuencia de imágenes, es sin duda una de las mayores aplicaciones del filtro de partículas, muchos trabajos se han realizado en este tema [9] y la mayoría resalta la eficiencia y fácil implementación de esta técnica en la visión artificial. En el trabajo realizado por Eva Burgueño [10], la autora realiza la implementación de un filtro de partículas convencional que responde a modelos de funciones gaussianas unimodales, utilizando la segmentación basada en umbralizado y sustracción de fondo para la fase de seguimiento de objetos. De este trabajo se rescata la estructura del filtro unimodal, que se usa como base para la implementación multimodal en esta investigación, pero difiere en la forma del seguimiento ya que la propuesta realizada involucra la comparación de histogramas como se realiza en [11] y no el conteo de píxeles como lo trabaja la autora.

El principal problema que presenta el enfoque basado en estrategias clásicas, cómo el filtro convencional mencionado anteriormente, es que no constituye un enfoque multimodal, es decir, no permite sostener a lo largo del tiempo el seguimiento de múltiples objetos, sin embargo es posible obtener una solución eficaz, como es mencionado en [12] y [13], adaptando un clasificador al filtro, de manera que pueda mantener múltiples hipótesis en el espacio de búsqueda, también la combinación de filtros convencionales permite obtener un seguimiento de varios objetos [14], pero así mismo el coste computacional se eleva. En este trabajo se propone el seguimiento de dos objetos correspondientes a las dos pupilas del usuario, siendo necesario el uso de un filtro de partículas multimodal, como el estudiado en [13], y que es adaptado para este trabajo utilizando un proceso de *clustering* para conformar los grupos que exploran las diferentes hipótesis.

Durante los últimos años el seguimiento ocular o *eye tracking*, por sus siglas en inglés, ha tomado gran fuerza en los estudios de visión artificial, debido a las múltiples aplicaciones que brinda, como el análisis expresiones faciales, vigilancia de conductores al volante, interacción humano-computador, por ejemplo, [15], [16], [17], utilizan el rastreo de los ojos para determinar la orientación de la cabeza mediante la resta sucesiva de fotogramas, con el propósito de manipular algunas aplicaciones de un ordenador. En [18] se utiliza un procesamiento de imágenes mediante histogramas, para segmentar el ojo mediante la comparación con plantillas creadas con anterioridad. Lo rescatable de estos trabajos, radica

en la forma de detección de los ojos, donde se implementa un sistema de emisores de luz infrarroja y sensores o cámaras sensibles a este tipo de luz, que utilizan la propiedad del ojo humano para reflejarla y así lograr determinar su posición. Este sistema es muy eficiente, de bajo costo y no invasivo para el usuario, características destacables que permiten a este método ser adoptado para el desarrollo de esta investigación.

Es posible también, utilizar el filtro de partículas para el seguimiento ocular como lo realiza Yali Li y compañía en [19], utilizando una plantilla del ojo como modelo para las partículas y así determinar su ubicación y orientación. Este enfoque que se le da al filtro se utiliza como base para este trabajo, debido a que se propone la utilización del filtro para la detección de las pupilas de los ojos del usuario.

Dentro de las aplicaciones encontradas similares a la obtenida en esta investigación, se destaca la desarrollada en [20], que utiliza el procesamiento digital de imágenes para encontrar la pupila del ojo del usuario, mediante la binarización de los fotogramas y la búsqueda de formas circulares. Una vez encontrada la ubicación de la pupila, se determina la orientación con respecto al centro de la misma, para identificar acciones como, mirar a la derecha o izquierda, arriba o abajo y así posicionar el cursor del ordenador. A diferencia de este trabajo, la propuesta presentada involucra el seguimiento múltiple y no de un solo ojo como lo desarrolla el autor, además de esto, el posicionamiento se realiza mediante coordenadas incrementales [16], lo que permite más movilidad del cursor y variación de velocidad dependiendo de la distancia que deba recorrer, sin mencionar que para el seguimiento se utiliza el filtro de partículas multimodal y no la resta sucesiva de imágenes.

1.2 El ojo humano.

El ojo humano es un órgano foto-receptor que detecta la luz, es un sistema óptico positivo o convergente que forma una imagen invertida del mundo externo sobre la capa sensible de la retina, situada al fondo del globo ocular [21]. El ojo se convierte en la base del sentido de la vista, uno de los cinco sentidos que permiten la comprensión del mundo que nos rodea. El globo ocular mide aproximadamente unos 2,5 cm. de diámetro, y está recubierto por una membrana compuesta por varias capas. En la *Imagen 1.1* se puede observar un ojo con sus partes básicas visibles.

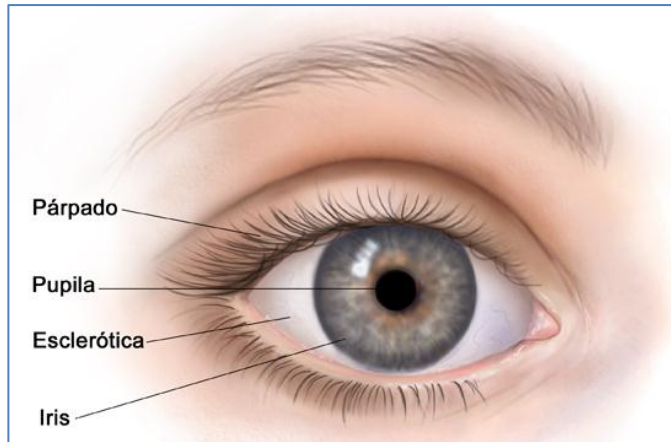


Imagen 1.1 Vista frontal del ojo (adoptado de [22]).

La zona óptica del ojo está compuesta principalmente por la córnea, el iris, la pupila y el cristalino, como lo ilustra la *Imagen 1.2*, así como también sus otros componentes. La córnea es un material transparente que permite el paso de la luz hacia el interior del ojo el cual sirve de protección al iris y al cristalino debido a que posee propiedades ópticas de refracción y funciona como una lente fija. El iris es una membrana coloreada y circular que posee una apertura interior variable de color negro que es denominada pupila, la cual comunica la cámara anterior del ojo con la posterior. El iris está constantemente activo para permitir que la pupila se dilate (midriasis) o se contraiga (miosis), esto con la finalidad de controlar la cantidad de luz que llega a la retina. A su vez, el cristalino está situado delante del humor vítreo y detrás del iris, tiene forma de lente biconvexa, se encarga de refractar la luz para poder proyectarla en la retina. Su función principal es permitir enfocar objetos que se encuentran a diferentes distancias [23]. La retina se encuentra en la parte trasera del globo ocular y es la que contiene los receptores sensibles a la luz, denominados fotorreceptores. Estos últimos se encargan de convertir la luz en impulsos eléctricos, que van a los centros encargados de la visión en el cerebro.

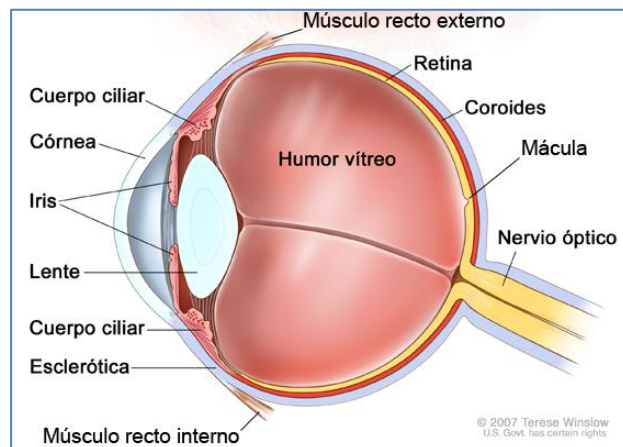


Imagen 1.2 Sección del ojo humano (adoptado de [22]).

1.3 La cámara web como sensor.

El dispositivo encargado de seguir los movimientos del ojo es llamado *Eye Tracker* (Seguidor de ojos). Se pueden distinguir dos técnicas de monitorización del movimiento de este órgano, una que mide la posición del ojo con respecto a la cabeza y otra que mide la posición del ojo en el espacio, conocida como punto de mirada [23].

Existen cuatro técnicas básicas de medición del movimiento ocular mencionadas en [23], Electro-oculografía, lentes de contacto, foto o video oculografía y detección por video basada en la pupila y la reflexión en la córnea. Cada una de las técnicas nombradas tiene sus particularidades.

La electro-oculografía (EOG) descrita en [24], es una técnica que mide diferencias de potencial mediante unos electrodos, normalmente cinco, colocados alrededor de la cavidad ocular del sujeto que registran los movimientos del ojo y establecen la posición relativa con respecto a la cabeza (ver *Imagen 1.3*), esta técnica no permite medir la posición en la que el ojo entra la mirada, por lo que no se aborda más el tema.



Imagen 1.3 Medición del movimiento mediante electro-oculografía (adoptado de [25])

La técnica de lentes de contacto, según Turégano en su libro [23], es una de las más precisas para medir el movimiento del ojo, consiste en un lente que contiene un hilo de metal enrollado en una vuelta, utilizado para medir las variaciones del campo magnético. Este método mide la posición de los ojos respecto a la cabeza y tampoco es funcional para establecer el punto de mirada, además es un método intrusivo que genera incomodidades al usuario para realizar la medición.

La video-oculografía o foto-oculografía representa una gran variedad de técnicas bajo esta denominación, las cuales se encargan de extraer características de los ojos bajo la rotación y

traslación, como son la forma de la pupila, la unión entre el iris y la esclerótica, las reflexiones córneas, etc. Pero, ninguna de estas técnicas se encarga de brindar una medición precisa del punto de interés.

Por último, la técnica de detección por video basada en la pupila y la reflexión de la córnea, a diferencia de las técnicas mencionadas anteriormente, ésta si permite obtener una medición precisa del punto de interés o punto de mirada del ojo en el espacio. Para esto, se necesita que la cabeza del sujeto esté quieta, ya sea mediante un soporte en el mentón que mantenga la cabeza fija o mediante un dispositivo integrado en la misma cabeza (*Head Mounted System*), de manera que la posición del ojo pueda coincidir con el punto de interés.

En esta técnica, se presentan dos tipos de seguimiento basados en luz infrarroja: los de pupila oscura y los de pupila brillante. Cuando el haz de luz que ilumina el ojo es coaxial al camino de visión, el ojo actúa como reflector, la luz se refleja en la retina y produce una pupila brillante en la captura de la imagen (*Imagen 1.4 a*). En cambio, si existe un desplazamiento entre la luz y el camino de visión, la pupila será capturada con un color muy oscuro: pupila oscura (*Imagen 1.4 b*).

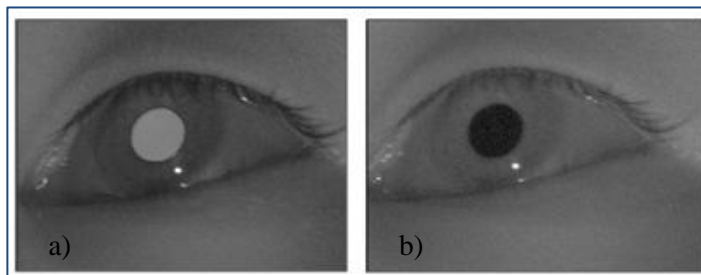


Imagen 1.4 a) Pupila brillante, b) Pupila oscura (adoptado de [26]).

De acuerdo a las características de cada método presentado anteriormente, para el desarrollo de este trabajo se decidió utilizar la detección por video basada en la pupila y la reflexión de la córnea, con luz infrarroja para destacar la pupila oscura.

1.4 **Visión artificial.**

La visión artificial estudia los diferentes procesos que intervienen en la visión humana, con el fin de comprenderla y desarrollar dispositivos o aplicaciones con propiedades similares, en el intento de reproducir este comportamiento humano, dentro de la visión artificial se definen tradicionalmente cinco etapas principales:

- La primera etapa es puramente sensorial, consiste en la captura o adquisición de las imágenes digitales mediante algún tipo de sensor para su posterior almacenamiento.

- En la segunda etapa, se realiza un tratamiento digital de las imágenes mediante filtros o transformaciones geométricas con el fin de eliminar partes indeseables o resaltar zonas de interés y así facilitar etapas posteriores.
- La tercera etapa consiste en segmentar las zonas importantes de la escena mediante sus atributos, de esta manera se consigue aislar los elementos de interés que la componen.
- En cuarto lugar, se extraen las características relevantes del objeto, con el fin de realizar una adecuada clasificación.
- Por último, se presenta una etapa de clasificación o reconocimiento, que pretende distinguir los objetos segmentados de acuerdo a las características extraídas en la etapa anterior.

Dentro de los problemas clásicos de la visión artificial, se encuentra el seguimiento de objetos en secuencias de imágenes, llamado en inglés *tracking*, este problema se puede abordar resolviendo tres etapas importantes, la detección de los objetos de interés, seguimiento de los objetos de un cuadro al siguiente y por último, el análisis del movimiento de los objetos para identificar patrones de comportamiento.

Seleccionar las características correctas del objetivo a detectar, juega un papel crítico en cualquier aplicación de visión artificial, en general, se busca una propiedad distintiva del mismo, que permita diferenciarlo del resto de objetos de una escena, como son el color, la forma o la textura, o combinaciones de ellas. Dentro de la etapa de adquisición de las imágenes, un punto muy importante radica en la selección del espacio de color con que se censa el mundo real, dependiendo de los requisitos para el procesamiento, es posible establecer qué espacio de color se debe utilizar para resaltar características de los objetos en la escena. Un objeto puede ser una entidad del mundo que tenga un color determinado, por eso al pensar en características la primera y más sencilla es el color [9].

Existen diferentes espacios de color con los que se puede representar una imagen, el más conocido de ellos es el RGB (rojo, verde, azul) con los que se representan los colores en el procesamiento de imágenes, sin embargo, tiene la desventaja que si una misma escena varía en su brillo, los tres componentes de color se ven afectados. Otro de los modelos de color muy utilizados en el procesamiento es el HSV (matiz, saturación, brillo) debido a que su descomposición de canales permite procesar imágenes cuando el brillo es un factor clave, aislando este canal y operando con los otros dos. Por último, en el procesamiento digital de imágenes es muy utilizada la escala de grises, representando la imagen con sombras que varían en intensidad de este color, donde cada pixel posee un valor equivalente a una graduación de gris, pasando por negros profundos hasta llegar al blanco.

Una imagen, en cualquier espacio de color, para el procesamiento digital comúnmente se representa utiliza un histograma que contiene las características de color, los histogramas de color son la representación gráfica de la luminosidad de la imagen, donde la abscisa representa la luminosidad dividida en 32, 64, 128 o 256 tonos (*o bins*) y la ordenada indica la frecuencia con la que se repite el tono. En la *Imagen 1.5*, se puede observar un ejemplo de histogramas de color en el espacio HSV.

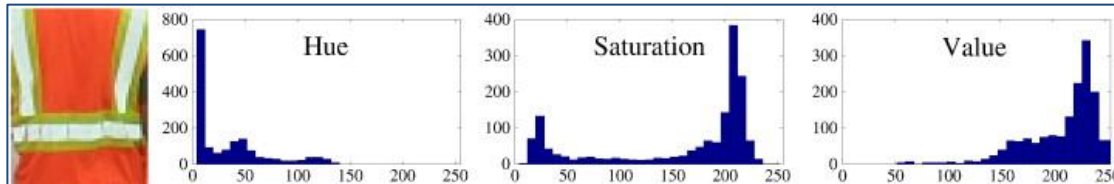


Imagen 1.5 Histogramas de color en HSV

Capítulo 2.

FILTRO DE PARTÍCULAS

En este capítulo se aborda teóricamente el funcionamiento del filtro de partículas como técnica de estimación dinámica (localización y seguimiento) dentro de un marco probabilístico, para ello se inicia definiendo el filtrado bayesiano en su forma recursiva y su aproximación mediante el método de Monte Carlo, el cual se emplea hoy en día como base de muchas técnicas para la estimación bayesiana. Posteriormente, se definen las etapas del filtro de partículas monomodal y su funcionamiento en una secuencia de imágenes, para terminar con el algoritmo de un filtro de partículas multimodal aplicado a una secuencia de imágenes.

2.1 Fundamentos del filtro de partículas.

Hablar de una estimación dinámica, se refiere a la habilidad de estimar continuamente la posición de un objeto, o en otros términos, estimar algún valor que cambia con respecto al tiempo y mantener dicha estimación.

Una de las principales formas de solucionar el problema de seguimiento utilizando métodos probabilísticos es el filtrado bayesiano [1] que se encarga de proporcionar una estimación precisa del estado real del objetivo, dadas unas observaciones relacionadas estadísticamente con él. Usando el teorema de Bayes (Ecuación 2.1), es posible relacionar las variables observadas con el estado a estimar, utilizando distribuciones de probabilidad condicionadas del estado con respecto a las observaciones y viceversa, esto es, estimar $p(x_k|z_{1:k})$, a partir de $p(z_{1:k}|x_k)$, donde x_k representa el estado del sistema en el instante de tiempo k y $z_{1:k}$ las observaciones realizadas sobre dicho sistema hasta el instante k , mediante:

$$p(x_k|z_{1:k}) = \frac{p(z_{1:k}|x_k)p(x_k)}{p(z_{1:k})} \quad (2.1)$$

Sin embargo, la estimación del estado se vuelve más compleja, a medida que k aumenta, siendo esto el gran problema del filtrado bayesiano y por lo cual es muy difícil su implementación, computacionalmente hablando. Para alivianar la complejidad del cálculo de la distribución de probabilidad, se hace necesario una formulación recursiva, donde simplemente la estimación del estado actual, dependa únicamente del estado anterior, a esto se le llama filtro bayesiano recursivo [27] y su aproximación es la base para el filtro de partículas [10]. Caracterizando lo anterior dentro de un marco probabilístico, se tiene que, el estado actual, x_k depende del estado anterior x_{k-1} , es decir $p(x_k|x_{k-1})$, representando la evolución del sistema. Por otro lado, las observaciones del sistema se trabajan como un canal sin memoria, esto es, z_k depende únicamente del estado en ese instante x_k , es decir $p(z_k|x_k)$, esto representa el proceso de observación.

Como se mencionó anteriormente, la base del filtro de partículas radica en la aproximación del filtrado bayesiano, ya que analíticamente es complejo obtener la secuencia de distribuciones de probabilidad para definir el estado. Para esta aproximación, el método más conocido y utilizado es el método de Monte Carlo [28], que mediante muestras de la distribución de probabilidad actual, puede estimar, mantener y actualizar la función de densidad a *posteriori*.

Monte Carlo, recibe su nombre de la ciudad del principado de Mónaco, donde anteriormente se conocía como la “ciudad del juego de azar”, debido a que este método hace uso extensivo de los números aleatorios para conseguir muestras representativas de una determinada distribución de probabilidad, si el número de muestras es suficiente, se puede obtener una estimación numérica de la distribución de probabilidad muy fiable [10].

Con lo anterior, se puede definir entonces el filtro de partículas, como la aplicación secuencial de Monte Carlo para la estimación de una densidad de probabilidad a *posteriori* mediante una dispersión de partículas en el espacio de estados. Las partículas, definidas como puntos en dicho espacio, representan estados posibles del sistema. Este filtro también es llamado de varias formas, tales como Método secuencial de Monte Carlo (*Sequential Monte-Carlo Method*)[28], Algoritmo de condensación (*Condensation Algorithm*) [29], Filtro *Bootstrap* [1], Algoritmo de Supervivencia del Más Apto, sin embargo, últimamente se le conoce sólo como Filtro de Partículas (*Particle Filter*) a todos ellos [10]. La idea central del filtro consiste en propagar las partículas (muestras de la función de densidad) sobre el espacio de estados mediante un modelo de movimiento y un modelo de observación, de esta manera, el peso combinado de las partículas de una región se aproxima a la integral de la función de densidad a *posteriori* en esa región. Dicho de otra forma, el filtro de partículas representa la densidad a *posteriori* mediante un conjunto discreto de N partículas (m_1, \dots, m_N) , y sus probabilidades asociadas (w_1, \dots, w_N) .

Desde un punto de vista algorítmico, en el filtro de partículas se pueden distinguir cuatro etapas, (ver Figura 2.1):

1. *Inicialización*: Como todo sistema, el filtro de partículas necesita de una etapa de inicialización, donde se distribuyen aleatoriamente las partículas por todo el espacio de posibles estados.
2. *Actualización*: Se toman medidas mediante el modelo de observación sobre el estado actual del sistema, las cuales más adelante se utilizaran para modificar la predicción del sistema.
3. *Estimación*: Mediante las medidas tomadas en la etapa de actualización, se estima el estado actual del sistema.
4. *Predicción*: Mediante un modelo de movimiento, se predice el estado del sistema para el siguiente instante de tiempo.

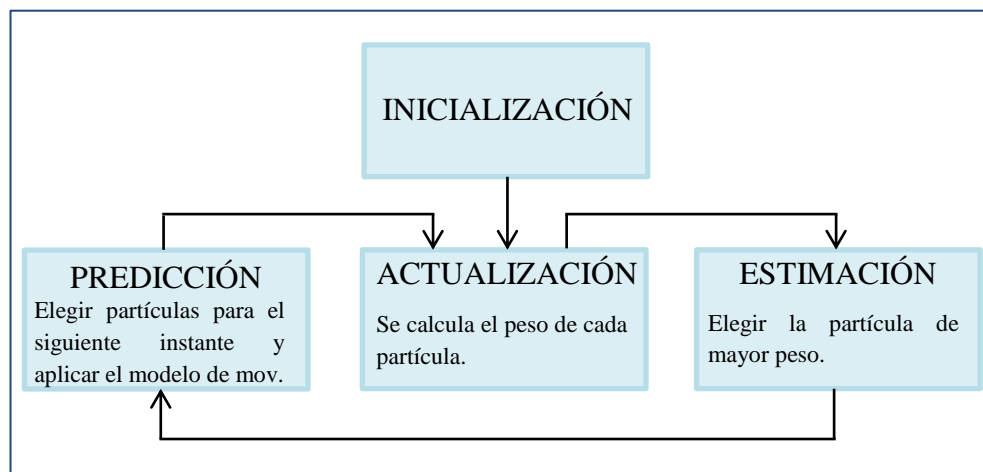


Figura 2.1 Etapas del Filtro de Partículas.

Como se puede ilustrar en la *Figura 2.2*, se comienza con una población de N partículas aleatoriamente distribuidas y ponderadas uniformemente $(m_k^i, 1/N)$, con $i = 1, 2, 3 \dots N$, que aproximan la densidad $p(x_k | z_{k-1})$. En este momento se reciben nuevas medidas de z_k y se calcula el peso para cada partícula que involucra a la función de densidad de probabilidad $p(z_k | x_k)$ que representa las observaciones sobre el sistema. El resultado es un conjunto de partículas con pesos asociados (m_k^i, w_k^i) , que constituyen una aproximación discreta de $p(x_k | z_k)$.

A continuación, se lleva a cabo el paso de estimación, en este paso, se seleccionan las partículas con mayor peso para obtener un conjunto de partículas $(m_k^i, 1/N)$ ponderadas uniformemente nuevamente. Por último, se realiza la predicción, cuyo objetivo es adaptar el conjunto de partículas al nuevo instante de tiempo, aproximando así la función de densidad de probabilidad $p(x_{k+1} | z_k)$.

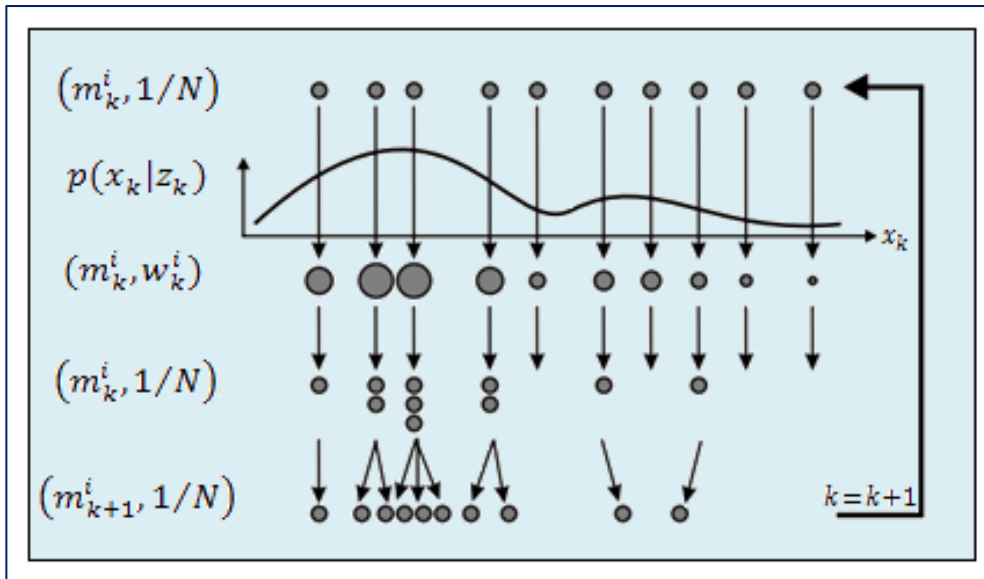


Figura 2.2 Representación gráfica de un filtro de partículas (adoptado de [6]).

2.2 Filtro de partículas monomodal.

Un objetivo principal de este trabajo radica en la estimación de la posición de la pupila de los ojos utilizando para ello únicamente información visual. El objetivo del filtro es utilizar como entrada las imágenes proporcionadas por la cámara IR-LED y obtener como salida los valores estimados de la posición de los objetivos en la escena.

Para cumplir con este objetivo, se considera en primer lugar el seguimiento de un único objeto en la escena, es decir, que cada una de las partículas empleadas se considera como una hipótesis de la posición de dicho objeto, la combinación adecuada de estas hipótesis, permite obtener una estimación de la distribución de probabilidad *a posteriori* $p(x_{k+1}|z_k)$. Para poder comprobar si las hipótesis resultan ser verdaderas o falsas (filtrado), se hace necesario obtener en primer lugar el modelo de la partícula, así como también el modelo de movimiento y el de observación, como se define a continuación.

2.2.1 Modelos involucrados.

Modelo de la Partícula: El modelo de la partícula, presenta el conjunto necesario de variables con las que se puede describir el estado del sistema. Para el seguimiento del objeto en la imagen de 2D, se ha tomado como modelo un conjunto de píxeles, llamado ventana, de tamaño 7x7, con el que se realizará el seguimiento del objetivo. Cada partícula estará

caracterizada por un estado y un peso, el estado de cada partícula viene definido por una coordenada y el peso es determinado por la similitud en componentes de color con el objetivo y por una distancia con respecto a la coordenada del centroide de la masa de las partículas.

Modelo de Movimiento: El modelo de movimiento empleado es el más genérico posible, esto, con el fin de adaptar fácilmente el modelo a cualquier potencial movimiento. Su filosofía se basa en suponer la continuidad del movimiento, es decir, las zonas más cercanas a la posición actual tienen más probabilidad que las zonas más lejanas. Para ello, las partículas que presentan un mayor peso, tienen más alta la probabilidad de ser elegidas que el resto, pero para que esta elección no se repita varias veces, el modelo aplica una difusión al estado de cada partícula, con la siguiente regla:

$$\begin{aligned}mX_{k+1} &= mX_k + \text{Aleatorio} + Vx_k \\mY_{k+1} &= mY_k + \text{Aleatorio} + Vy_k\end{aligned}\tag{2.2}$$

donde *Aleatorio* es un número generado aleatoriamente dentro de un rango definido alrededor de la partícula, mX_{k+1} representa la coordenada x de la partícula m en el instante $k+1$ y Vx_k representa la velocidad de movimiento (*pixel / frame*).

Modelo de Observación: El modelo de observación, es el encargado de filtrar la información de una imagen o una secuencia de imágenes para realizar su procesamiento, en otras palabras, es el encargado de cuantificar la relación entre el estado del sistema y las medidas realizadas sobre él. En el caso del seguimiento de la pupila, que es de lo que se ocupa este trabajo, se ha escogido un filtro de detección de color, debido a que la pupila resalta con un color llamativo del resto de los objetos de la escena, dadas las propiedades de reflexión de la luz infrarroja sobre la córnea. Cada partícula representa una posición en la imagen obtenida por la cámara, si la partícula pertenece al objeto, la zona que representa dicha partícula debe ser del mismo color que el objeto. Para el modelo de observación planteado, solo se necesita filtrar dicha zona de la imagen, de 7×7 píxeles como se mencionó anteriormente, y comprobar si esta hipótesis resulta verosímil o no. Para realizar esta comparación, se debe conocer de antemano las características de color del objetivo con el fin de determinar si existe algún grado de similitud con la partícula en observación, esta comparación se realiza calculando la distancia de Bhattacharyya [30], que da como resultado un valor normalizado que indica el grado similitud entre histogramas.

Con el fin de interpretar mejor los conceptos definidos anteriormente, en la sección siguiente se presenta el funcionamiento del filtro en sus diferentes etapas.

2.2.2 Funcionamiento del Filtro de partículas.

A continuación se presenta el funcionamiento del filtro detallando las siete etapas principales que lo conforman.

1. Generar datos

Para generar los datos de entrada que requiere el filtro de partículas para su buen funcionamiento, se debe indicar:

- El modelo del objetivo, es decir, el modelo con el cuál cada partícula será comparado, corresponde a los valores de color en escala de grises de una imagen de la pupila del ojo en las condiciones en que el filtro va a operar. La Imagen 2.1 muestra una sección de la pupila del ojo remarcada con un cuadrado rojo, de la cual se obtienen los valores del histograma de color y se convierten en las entradas al modelo de observación para determinar el grado de similitud de cada partícula.

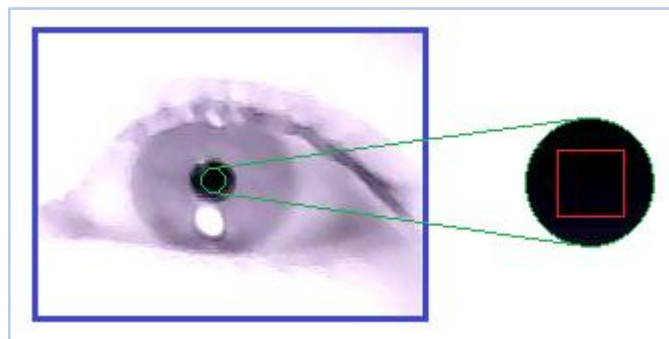


Imagen 2.1 Imagen del modelo de la pupila del ojo

- El número de partículas que se desean generar para crear las hipótesis del estado del sistema. Este elemento es decisivo a la hora de implementar el algoritmo, puesto que establece las posibilidades de capturar el objetivo. Un número bajo, no permite muestrear con suficiente precisión la función a seguir y un número muy alto, no asegura que se obtengan mejores resultados, además se incrementa el coste computacional.
- El tamaño de las partículas, es decir, el tamaño de la región que representa cada hipótesis y con la cual se realizan las comparaciones con el modelo de observación. La Imagen 2.2 muestra la representación de una partícula en el espacio de posibles soluciones, el tamaño en píxeles es de 7 x 7, siendo el píxel del centro de esta ventana el que determina la coordenada de su posición, es decir el estado, en la imagen este píxel está marcado con color rojo.

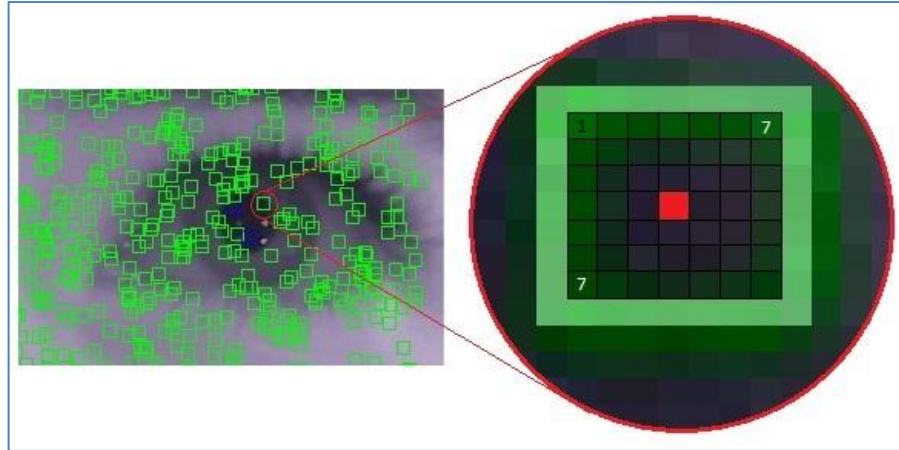


Imagen 2.2 Representación de una partícula.

2. Inicializar

Se generan $N = 1000$ partículas aleatorias por todo el espacio de posibles estados (recuadros verdes en la *Imagen 2.3*) y se inicializa la velocidad en cero $V_k^i = 0$, construyendo así la población inicial con pesos igualmente ponderados $w_k^i = N^{-1}$ y su respectivo estado m_k^i (coordenada), con lo que cada partícula es representada como (m_k^i, w_k^i, V_k^i) con k igual a 1, es decir, el primer instante de tiempo, e $i = 1, 2, 3 \dots 1000$.

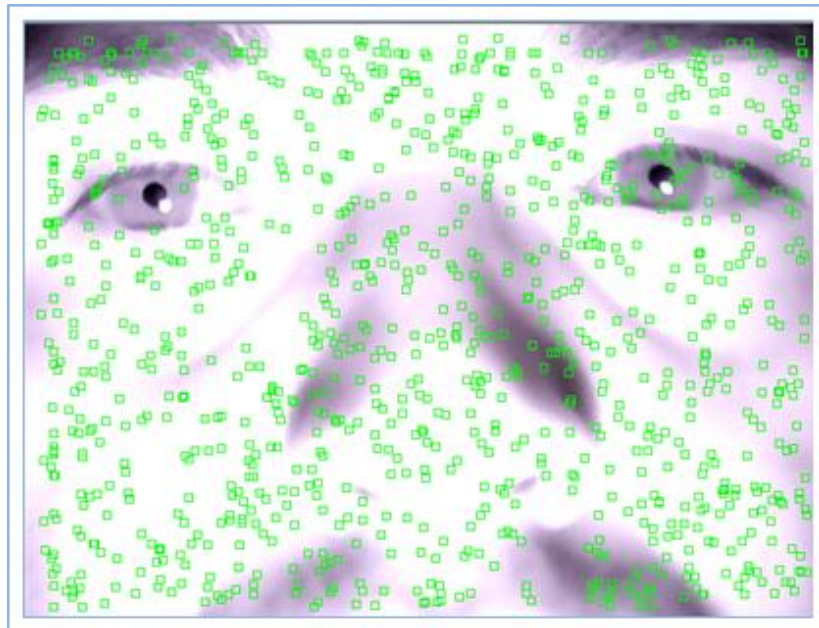


Imagen 2.3 Inicialización de las partículas.

Esta población inicial se utiliza para la búsqueda del objetivo durante las primeras iteraciones, el número de partículas creado es suficiente para cubrir de manera aleatoria todo

el espacio de estados, es decir, toda la imagen de 640 x 480 píxeles. Este número de partículas es excesivo al momento de realizar el seguimiento, debido a que el tamaño del objeto a seguir es de aproximadamente de 20 x 20 píxeles. Por lo tanto, una vez encontrado el objetivo se puede reducir el número de partículas para obtener un mejor rendimiento. En este caso en particular, se ha reducido de 1000 a 400 partículas una vez se detecta la pupila del ojo.

3. Calcular Pesos

Cada partícula tiene su peso correspondiente, dicho peso se calcula comprobando la similitud de histogramas entre la región que representa la partícula y el modelo del objetivo, mediante el cálculo de la distancia de Bhattacharyya. Para comprender mejor el cálculo de los pesos, se definen los dos conceptos claves de esta etapa, el histograma y la distancia de Bhattacharyya.

Histograma: El histograma es la representación cuantitativa de la intensidad lumínica de la imagen representada en 256 *bins* o tonos, en este caso en particular, en la escala de grises. Debido a que las imágenes adquiridas bajo luz infrarroja no muestran una gran variedad de colores, es más, se pueden distinguir prácticamente dos tonos, claros y oscuros, en otras palabras, los que dispersan la luz IR y los que no, por este motivo se ha escogido trabajar bajo la escala de grises para procesar las imágenes y las partículas, además que aliviana el coste computacional porque solo se procesa un canal de color, el gris.

Distancia de Bhattacharyya: La distancia de Bhattacharyya es una métrica que permitirá valorar de forma cuantitativa el parecido o verosimilitud entre dos histogramas [12]. Esta distancia puede tomar un valor entre cero y uno, donde cero indica una semejanza perfecta y uno todo lo contrario. Esto quiere decir que las partículas con valores cercanos a cero tienen una mayor probabilidad de encontrarse sobre el objetivo. Esta cuantificación de la similitud no es conveniente para el siguiente paso del filtro, debido a que según el peso de la partícula, será asignado el porcentaje de espacio que tenga en la ruleta de selección, es por esto que el valor arrojado por la distancia de Bhattacharyya se invierte, es decir, que un valor de cero indique ninguna semejanza y un valor de uno, semejanza perfecta. Para esto, simplemente al valor de la distancia se le resta la unidad, como lo indica la ecuación 2.3. Para obtener este valor, se ponen en comparación el histograma normalizado del modelo del sistema o del objetivo y el histograma normalizado de la región que representa cada partícula mediante la siguiente ecuación:

$$Dis_{Bhat} = 1 - \sum_{i=1}^{Nb} \sqrt{h_{mod}(i) * h_{par}(i)} \quad (2.3)$$

donde Nb es el número de *bins* con que se representan los histogramas, h_{mod} representa el histograma del modelo del sistema y h_{par} representa el histograma de la partícula, ambos histogramas deben tener la misma dimensión y rango, esto se asegura creándolos a partir de la imagen en grises y normalizando los histogramas.

Luego de calcular la distancia de Bhattacharyya para cada partícula, se obtiene el conjunto de hipótesis con probabilidades de ser el objetivo proporcional a su peso ($m_k^i, w_k^i, V_k^i = 0$), la velocidad de cada partícula permanece en 0, debido a que esta solo se afecta cuando se aplica el modelo de movimiento, es decir, cuando se realiza un desplazamiento por el espacio de estados.

4. Crear ruleta

El proceso de creación de la ruleta, comienza con normalizar el vector de partículas, esto se hace mediante la ecuación 2.4, donde cada peso es dividido por la suma acumulada de los mismos, dando lugar a un peso normalizado a uno, como se muestra en la ecuación 2.5. Posteriormente se ordena de mayor a menor las partículas dependiendo del peso de las mismas, esto se hace mediante el método de la burbuja, que realiza un ordenamiento comparando cada componente con el siguiente e intercambiando la posición si están en el orden equivocado, esto se hace recorriendo varias veces la lista, hasta que ya no sean necesarios más intercambios.

$$\bar{w}_k^i = \frac{w_k^i}{\sum_{i=1}^N w_k^i} \quad \text{con } i = 1, 2, 3, \dots, N \quad (2.4)$$

con lo que se obtiene que:

$$\sum_{i=1}^N \bar{w}_k^i = 1 \quad \text{con } i = 1, 2, 3, \dots, N. \quad (2.5)$$

De esta forma se puede asignar el porcentaje que cada partícula tendrá dentro de la ruleta, dando mayor probabilidad de ser elegidas aquellas partículas que tengan un mayor peso, consiguiendo el efecto de un gráfico circular asignando un espacio en ella proporcional al peso como lo ilustra la *figura 2.3*

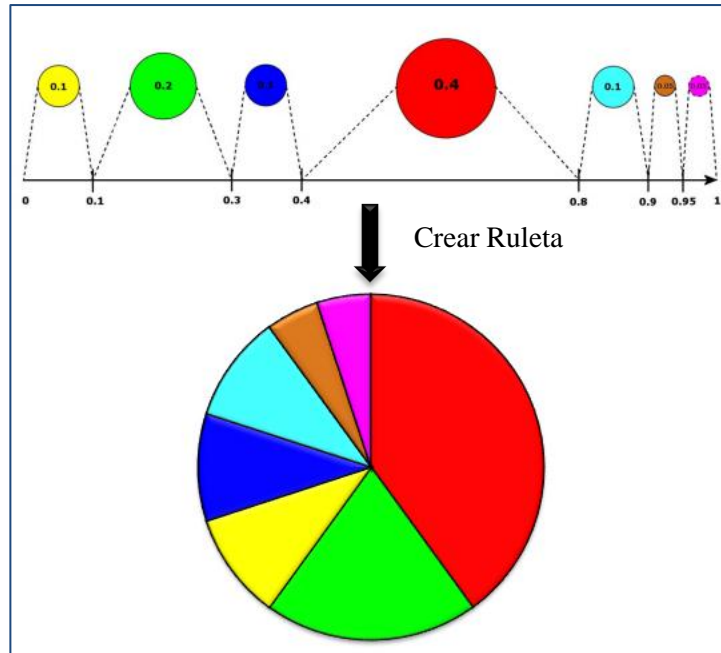


Figura 2.3 Ejemplo de creación de la ruleta.

Tal como se mencionó anteriormente, las partículas pueden reducirse en su número una vez se haya encontrado el objetivo, es por eso que no tiene sentido procesar las 1000 partículas si solo van a sobrevivir 400, este último número, es el número que procesa la ruleta para pasar las mejores partículas al siguiente instante.

5. Estimar

En esta etapa del filtro se estima el estado del sistema, esto se hace mediante una aproximación discreta de la función de densidad *a posteriori* dada por el filtro partículas con la siguiente ecuación:

$$p(x_k|z_k) \approx \sum_{i=1}^N w_k^i \delta(x_k - x_k^i) \quad (2.6)$$

donde w_k^i es el peso de la i -ésima partícula en el instante k y $\delta(x_k - x_k^i)$ es la representación discreta de cada partícula mediante la función delta de Dirac.

De esta manera se calcula la posición del filtro, utilizando los estados de aquellas partículas que superan un umbral determinado experimentalmente. Con estas partículas, se realiza un promedio de la posición y se determina la ubicación del objeto a seguir. En la imagen 2.4, se muestran las partículas de mayor peso coloreadas de rojo, con las cuales se realiza un promedio de posiciones y se establece la ubicación más probable del objeto en estudio.

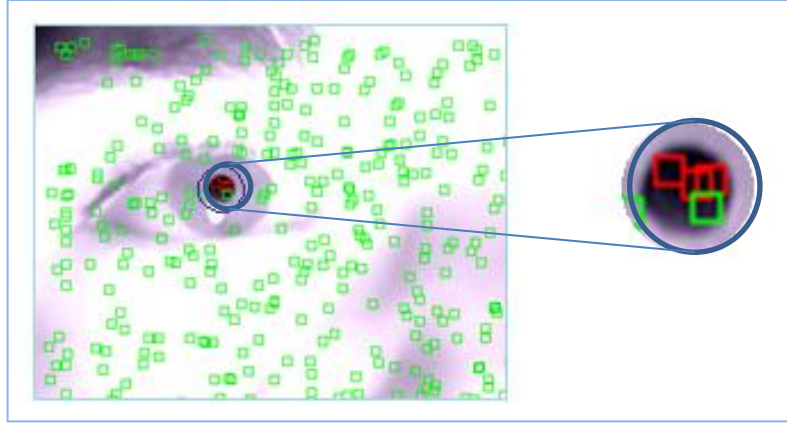


Imagen 2.4 Representación de las partículas de mayor peso.

6. Tirar ruleta

Para generar las nuevas partículas necesarias para el siguiente instante, se realiza un “sorteo” tirando la ruleta tantas veces como partículas se seleccionen, en este caso 400, teniendo así mayor probabilidad de que se elijan las partículas con mayor peso. Para ello, se escoge aleatoriamente un número entre 1 y 100 (0-100%), representando una posición de la ruleta, esta posición se compara con el vector que contiene los estados de cada partícula, obteniendo así un índice del estado que se elegirá para que permanezca en la siguiente iteración.

7. Difusión y Modelo de Movimiento

En esta última fase del filtro, se realiza una difusión de las partículas elegidas por sorteo aplicando el modelo de movimiento planteado en la ecuación 2.2, esta difusión se hace para que no coincidan estas partículas con las anteriores en el instante siguiente. Con esto se pretende predecir la posición del objetivo para la siguiente iteración aplicando un desplazamiento de las partículas en la dirección más probable. Este desplazamiento se genera con una velocidad que se calcula comparando la posición del objetivo en el instante anterior con el instante actual y determinando la distancia que se ha movido en pixeles (ecuación 2.7), teniendo en cuenta la continuidad del movimiento se predice que a esta misma distancia y dirección (ecuación 2.8) en el siguiente instante se posicionará el objetivo.

$$|\overrightarrow{x_{k-1} x_k}| = |\overrightarrow{x_k x'_{k+1}}| = d_k \quad (2.7)$$

$$\overrightarrow{\angle |x_{k-1} x_k|} = \overrightarrow{\angle |x_k x'_{k+1}|} = \theta_k \quad (2.8)$$

Con esta nueva posición x'_{k+1} predicha solamente con las posiciones de los instantes anteriores, es posible darle una idea de la velocidad y dirección con que se deben mover las partículas para el siguiente instante, restringiendo los valores de la variable *Aleatorio* de la ecuación 2.2. La velocidad con que se desplazan las partículas ha demostrado tener una constante, es decir que el recorrido en píxeles del objetivo, en este caso la pupila del ojo, de una iteración a la siguiente no presenta grandes cambios, por lo que se ha determinado establecer la velocidad constante y suprimir el cálculo de la misma reduciendo un poco el tiempo de ejecución. De esta manera se define $V_k^i = 2$ valor con el que se obtienen buenos resultados en el seguimiento.

Luego de la difusión de las partículas, es posible obtener una nueva población representada por $(m_{k+1}^i, w_{k+1}^i, V_{k+1}^i)$ con la que se estima la función de densidad *a posteriori* $p(x_{k+1}|z_k)$ y se predice el estado del sistema para la siguiente iteración.

La secuencia de etapas presentada anteriormente se puede resumir en un diagrama de bloques como sigue:

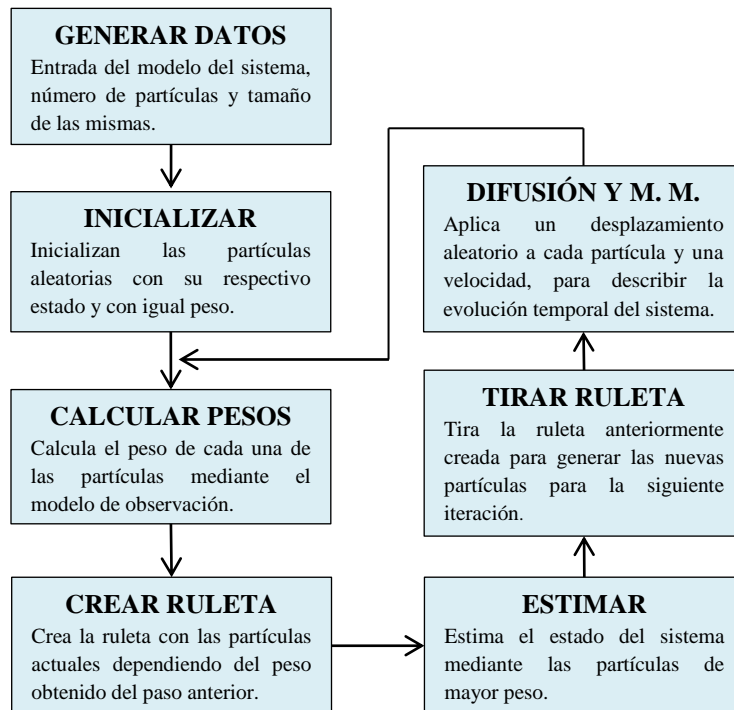


Figura 2.4 Diagrama de Bloques del Filtro de Partículas

Al aplicar todos los pasos del filtro de partículas monomodal, sobre una secuencia de imágenes del rostro del usuario obtenida con la cámara IR-LED y cargando como modelo

del objetivo una imagen con el color de la pupila, es posible obtener los resultados presentados en la Imagen 2.5.

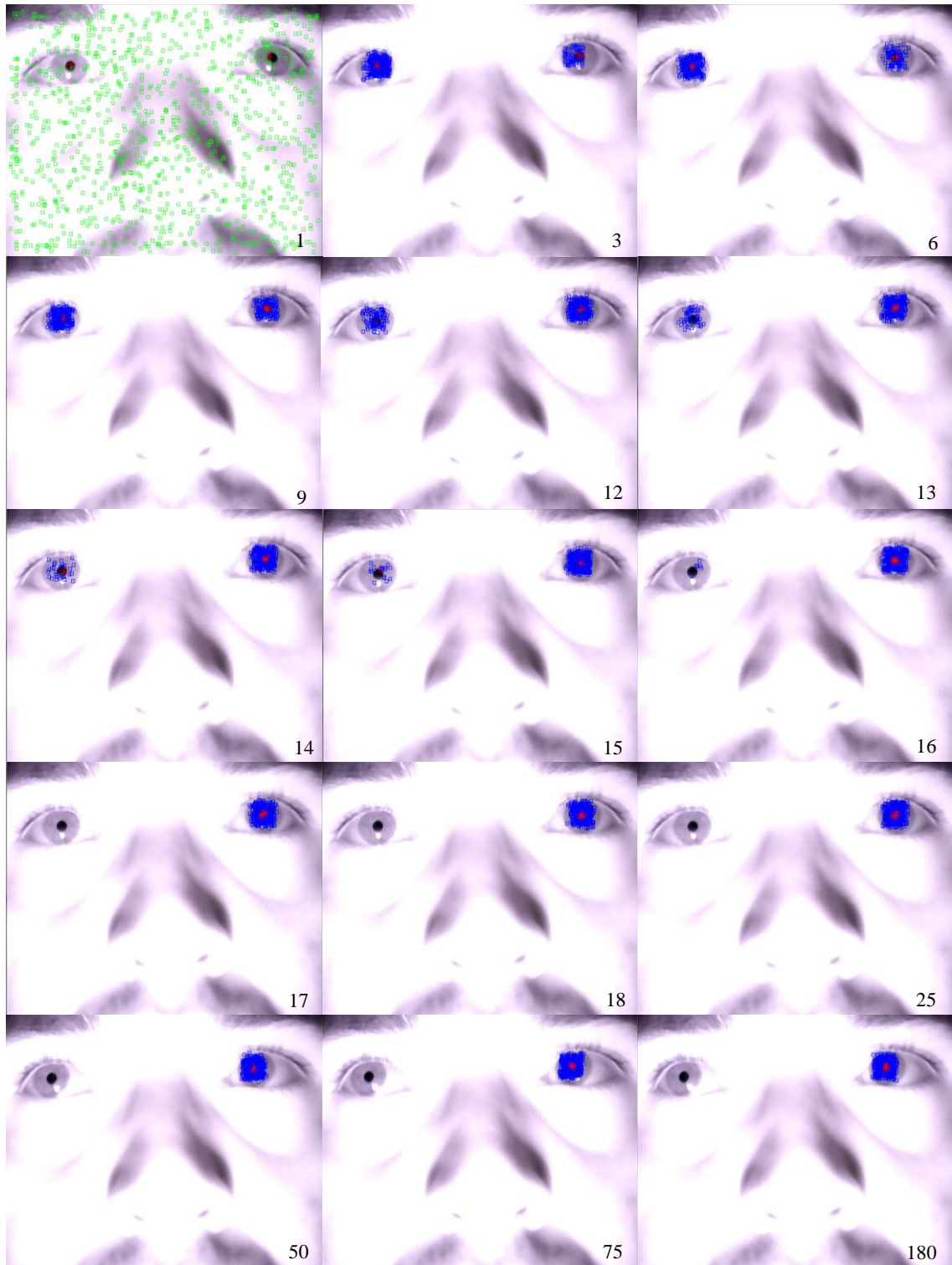


Imagen 2.5 Resultado del seguimiento monomodal.

La anterior ilustración muestra el funcionamiento del filtro de partículas monomodal para el seguimiento de la pupila del ojo utilizando imágenes capturadas bajo luz infrarroja, en la parte inferior derecha se ubica el número del fotograma al que corresponde la captura. Las partículas simbolizadas por un cuadrado rojo, representan las que mayor peso poseen, es decir, las que superan un umbral definido con anterioridad y aquellas partículas representadas por un cuadrado azul, indican partículas con buenos pesos obtenidas después de la difusión, pero que se utilizan para mantener una diversidad de muestras en el espacio de estados. En las primeras iteraciones del algoritmo, se pueden distinguir grupos de partículas que resultan ser igualmente válidas para el filtro, generando una discrepancia sobre qué posición es la correcta, dado que el modelo de observación no es capaz de separar las zonas que cumplen verazmente con el objetivo real, es por esto que el filtro puede mantener varias hipótesis de manera temporal mientras se encuentran indicios para elegir una solución entre todas las alternativas. Pero a la larga quedará únicamente una de las posiciones. Esto se debe a que el algoritmo coloca las partículas de manera ciega, es decir, el modelo de movimiento produce un paseo aleatorio por el espacio de estados sin tener en cuenta la información visual, por esto las partículas pueden tanto alejarse como acercarse a las posiciones correctas hasta que el modelo de observación se encarga de ajustar los pesos de las partículas para descartar aquellas mal posicionadas. Dentro de la secuencia también es posible observar claramente la fase de inicialización del filtro, que se ejecuta durante la primera iteración ubicando aleatoriamente partículas por todo el espacio de estados, sin embargo, cuando el objetivo es encontrado, las partículas se reducen para realizar el seguimiento, ya que no es necesario una cantidad elevada de las mismas para mantener muestreada las zonas de interés.

Dentro de los objetivos planteados para este proyecto, se estableció un seguimiento multimodal, es decir, tomar como objetivo a seguir la pupila de los dos ojos, por lo que se deben realizar modificaciones y mejoras al algoritmo presentado anteriormente, pero conservando la base del filtro.

2.3 Filtro de partículas multimodal.

Como se demostró anteriormente, no se puede asegurar la fiabilidad del filtro para el seguimiento multimodal, factor importante para el desarrollo de este trabajo, es por esto que se debe encontrar la manera de mantener en las zonas más probables un número de muestras que confirmen la hipótesis de que esas zonas pertenecen a los objetivos, la solución considerada consiste en integrarle un clasificador al filtro monomodal como se propone en [13], de esta forma se podrá mantener las zonas de interés en la función de probabilidad *posteriori* estimada. El filtro de partículas proporciona la detección y seguimiento de las zonas de interés, mientras que el algoritmo de clasificación previene el empobrecimiento de partículas en dichas zonas, de esta manera, es posible darle una sencilla solución al

problema de seguimiento multimodal. En la Imagen 2.6 se representa las zonas de interés en una imagen del rostro del usuario, mediante una función de densidad de probabilidad, donde en las pupilas de los ojos se ubican los máximos de la función, las cuales el filtro debe explorar, detectar y seguir.

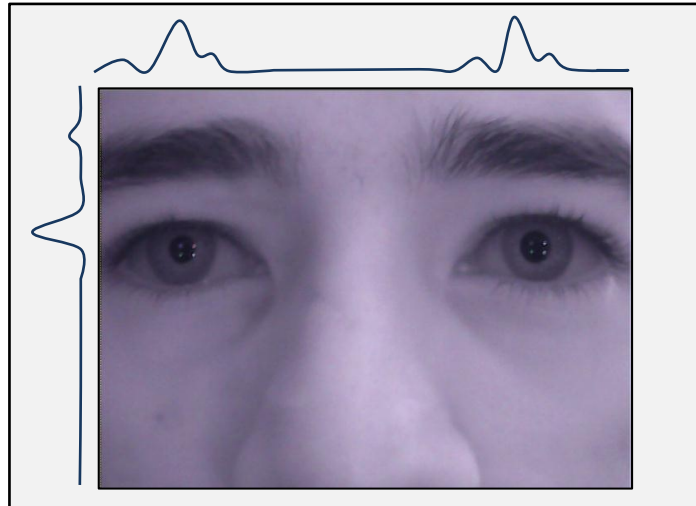


Imagen 2.6 Representación de las zonas de interés mediante una fdp.

En este trabajo se propone la utilización de un clasificador que cumpla con las características del seguimiento multimodal pero que además sea sencillo de implementar para reducir los costos computacionales. Dentro de las opciones encontradas de algoritmos de *Clustering*, se decidió la utilización del Algoritmo de las K-Medias básico [13], ya que aplica una lógica sencilla para clasificar o agrupar en un bajo tiempo de ejecución, además de eso, la única restricción que se presenta con respecto a su versión extendida, radica en el número de grupos a clasificar, para el K-Medias básico se debe determinar el número de grupos de antemano, mientras que para el K-Medias Extendido, él mismo resuelve el problema. El algoritmo está basado en la minimización de la distancia euclídea de cada patrón con respecto al centroide de agrupamiento, es decir, realiza el proceso de clasificación observando la distancia que existe entre cada partícula y cada uno de los centros estimados de las pupilas, por lo que solo se deben agrupar las partículas en dos, pupila derecha y pupila izquierda.

Es importante remarcar que en la mayoría de aplicaciones de seguimiento visual para posicionar la mirada en el espacio, como por ejemplo en [31] y [32], se utiliza el seguimiento de un solo ojo, o en su defecto, cuando se sigue los dos ojos, se utiliza algún dispositivo como gafas [16] o artefactos sobre la cabeza que permiten la captura de las imágenes, estos sistemas pueden presentar incomodidades al usuario por el permanente contacto con el sistema que soporta los sensores. A diferencia de los trabajos mencionados anteriormente, este propone el seguimiento de las pupilas, mediante el filtro de partículas y

el algoritmo de *clustering*, sin la utilización de artefactos o dispositivos que puedan incomodar al usuario, permitiendo que el sistema de captura esté ubicado frente al usuario y sobre la pantalla que despliega la interfaz.

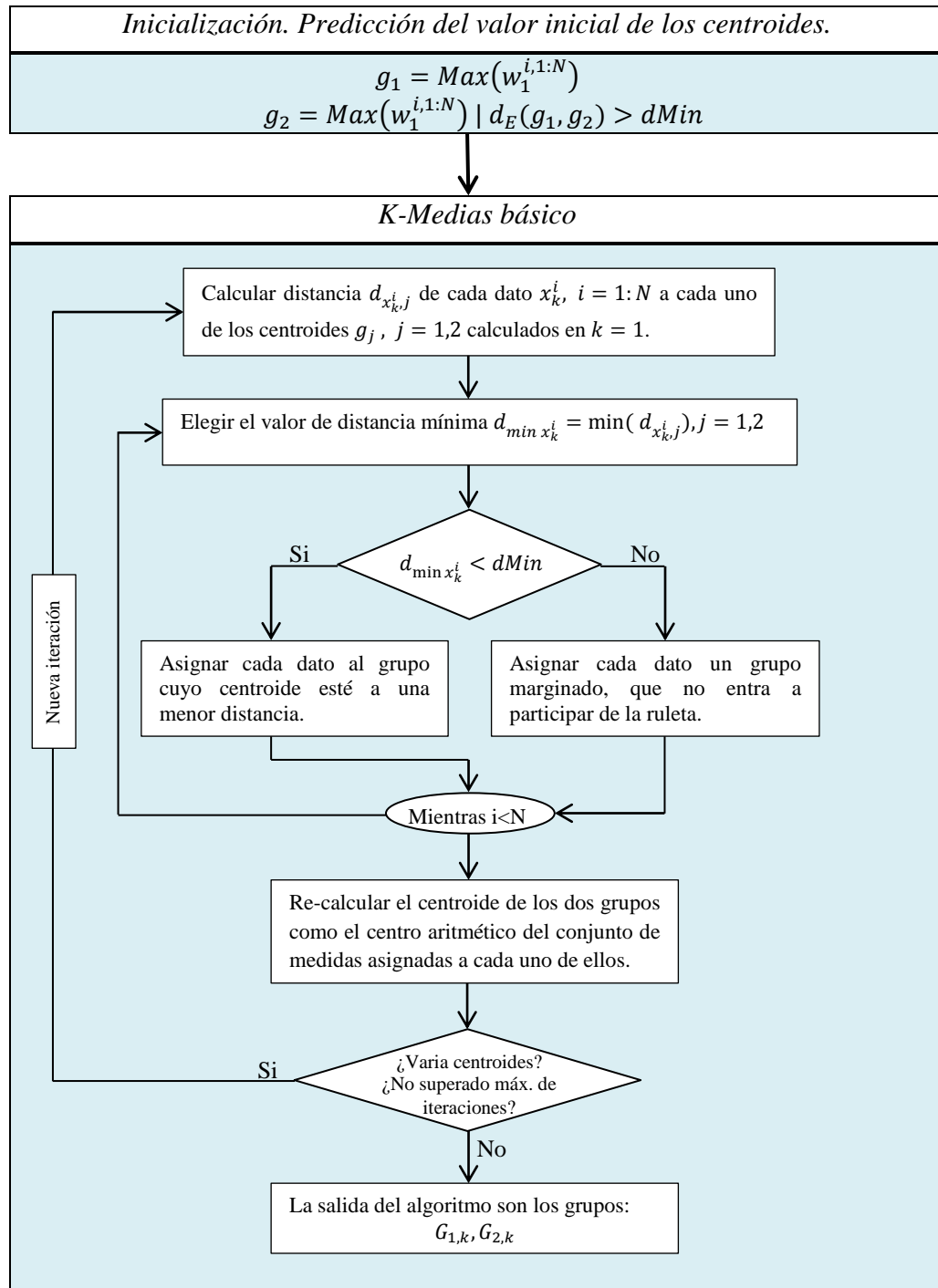
En las siguientes secciones se describe el funcionamiento del algoritmo K-Medias y la forma de implementarlo al filtro de partículas para añadir la propiedad multimodal, además se crea un proceso adicional de cálculo de los pesos, empleando las distancias que calcula K-Medias para cada partícula.

2.3.1 Algoritmo de las K-Medias [12].

Es tal vez el algoritmo de clasificación más conocido y basa su funcionamiento en las siguientes reglas para obtener k grupos:

1. Elegir k datos dentro del set de muestras para utilizarlos como valor inicial de los centroides de los grupos a crear. En este caso, se escogen las dos partículas con mayor peso separadas entre sí por una distancia mínima establecida, esto representa los dos posibles centroides de las pupilas.
2. Calcular la distancia entre cada dato y todos los centroides. Asignar el dato Y_i al grupo K_i situado a la menor distancia. Para el caso del filtro, se calcula la distancia en pixeles de cada partícula con respecto a las 2 partículas seleccionadas como centroides, y dependiendo de la distancia, se clasificarán en pupila derecha o pupila izquierda.
3. Una vez asignados todos los datos, se recalculan los centroides de cada grupo.
4. Se analiza si ha habido algún cambio en el valor de los centroides.
5. Si los ha habido, se repite el proceso desde el paso 2, hasta llegar a un número de iteraciones máximo, definido por el diseñador. En caso contrario, se eliminan los conjuntos sin miembros asociados, y el algoritmo concluye.

Hay que tener en cuenta que el K-Medias básico siempre debe llevar como parámetro de entrada el número de grupos a clasificar y que por lo tanto, para asegurar la convergencia de todos los datos, es necesario siempre en la primera iteración contar con los dos centroides que representan la posible ubicación de las pupilas, para ello se toma como valores iniciales, las partículas con mayor peso separadas a una distancia $dMin$ establecida de 300 pixeles, aprovechando el hecho que las pupilas nunca se traslapan. En la figura 2.5 se muestra esquemáticamente el diagrama funcional del algoritmo K-Medias básico.



De esta manera, al integrarle un algoritmo de clasificación al filtro de partículas, se puede asegurar el seguimiento de dos objetivos que presentan las mismas características para el modelo de observación como son las dos pupilas, pero que se pueden diferenciar por una distancia existente entre ellas.

2.3.2 Recalcular pesos.

Con el procedimiento del K-Medias ejecutado, es posible obtener la distancia de cada una de las partículas con respecto al centroide del grupo al que pertenece. Con el fin de mejorar la calidad de las partículas se recalcula el peso de cada una de ellas tomando como idea principal el concepto de cercanía con el objetivo, es decir, a parte de la similitud de color, la cercanía con el objeto a seguir tendrá una componente en el peso. Para determinar la componente de peso por distancia se hace uso de una función con forma gaussiana como lo ilustra la *figura 2.6*, debido a su propiedad isotrópica, que permite determinar un valor normalizado dependiendo de la distancia al centro de la campana, que representa el centroide de la nube de partículas, entre más cerca al centro se tendrá un valor aproximado a uno.

La función viene dada por la ecuación siguiente:

$$w_{d_k}^i = \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_k^i - \mu_k}{\sigma}\right)^2} \quad (2.9)$$

donde el valor de σ determina el ancho de la campana, x_k^i representa la posición de la partícula i en el instante de tiempo k y μ_k el centroide del grupo en el instante k , es decir que la diferencia entre x_k^i y μ_k resulta ser la distancia euclídea en píxeles entre la partícula en análisis y el centro de la campana. Es preciso mencionar que el ancho de la campana, es determinado por el tamaño de la pupila del ojo, por consiguiente, y sabiendo que el valor de la pupila está alrededor de 20 píxeles, σ toma un valor de 10 teniendo en cuenta que una de las propiedades de la distribución indica que dentro de -3σ y 3σ se encuentra casi la totalidad de la información necesaria para tomar valores significativos, dejando un margen de 10 píxeles para las partículas que se encargan de la diversidad de la información.

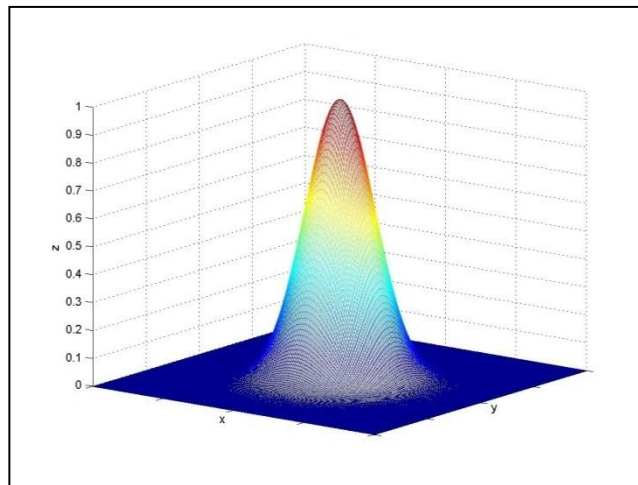


Figura 2.6 Campana de Gauss para el cálculo de peso.

Ahora es posible recalcular los pesos de las partículas teniendo los valores provenientes de la similitud entre histogramas de color y de la cercanía con el objetivo, para ello, se relacionan los dos valores normalizados con la multiplicación, con lo que se asegura que el peso sigue siendo un valor entre cero y uno.

$$w_{r_k}^i = w_k^i * w_{d_k}^i \quad (2.10)$$

A continuación se presenta el diagrama de bloques para el filtro multimodal.

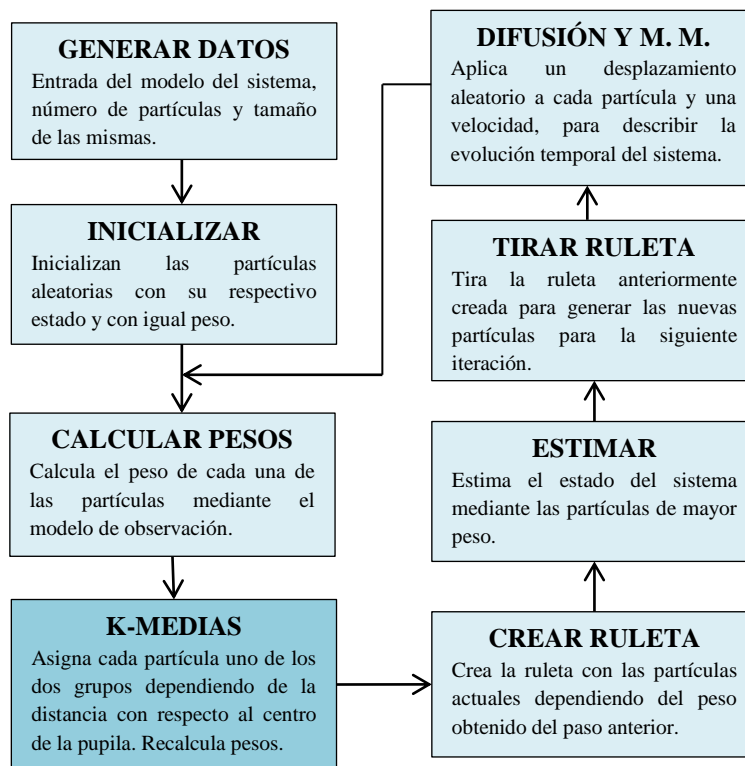


Figura 2.7 Diagrama de Bloques del Filtro de Partículas

Como se mencionó anteriormente, el algoritmo de clasificación cumple con el objetivo de mantener un número mínimo de muestras en las zonas de interés, gracias a eso, la fase de la ruleta en el filtro puede preservar para el instante siguiente las partículas mejor posicionadas en dichas zonas, pero además, la ruleta es creada para generar el mismo número de partículas en ambas zonas.

El filtro de partículas multimodal implementado, es capaz de mantener muestras en dos zonas con alta probabilidad de ser los objetivos, es decir, coloca partículas en los máximos de la función de densidad de probabilidad *a priori*. Al igual que en el filtro monomodal en la

Imagen 2.7, los recuadros verdes representan las partículas de búsqueda en el espacio de estados, luego de encontrar los posibles objetivos, las partículas representadas con un recuadro rojo son aquellas que tienen un peso elevado y son las más probables de ubicarse sobre las pupilas, por último los recuadros azules representan las partículas que se utilizan para mantener una diversidad de muestras alrededor de los máximos locales, en la parte inferior derecha se ubica el número del fotograma al que corresponde la captura. Dado que el modelo de observación solo basa su similitud en los componentes de color, las partículas mejor ubicadas, son las que corresponden en un alto grado de similitud en este aspecto, pero esto no asegura que estas partículas se encuentren sobre el objetivo real, ya que algún otro objeto puede tener los mismos componentes de color que el modelo, y el filtro puede converger a estas zonas, este es el gran inconveniente de la segmentación por color. En la secuencia de fotogramas de la Imagen 2.7 es posible observar algunas características del filtro, como por ejemplo la búsqueda ciega durante los fotogramas 405 y 408, que consiste en una búsqueda local con las partículas de seguimiento cuando los objetivos se han perdido, es decir que durante cinco iteraciones el algoritmo supone que las pupilas aparecerán de nuevo en la última posición conocida, de no ser así, se realiza una búsqueda por todo el espacio de estados nuevamente, como se aprecia en el fotograma 410. Esta pérdida de los objetivos da pie para hablar del inconveniente del filtro planteado anteriormente, una vez se realiza la búsqueda por toda la imagen, el filtro detecta zonas con buenas probabilidades de ser los objetivos convirtiendo a estas hipótesis en verdaderas, pero en realidad no son los objetivos, las cejas en este caso cumplen con el modelo de color y por lo tanto la ruleta se encarga de replicar partículas en estas zonas permitiendo que el filtro converja erróneamente, como se observa durante los fotogramas 412 hasta el 805, cuando se realiza una nueva búsqueda. Esto demuestra claramente que el funcionamiento del filtro se basa solo en las características de color exploradas por las partículas, sin embargo no es eficiente cuando algún objeto, como las cejas, presentan las mismas propiedades representadas en el histograma.

Para solucionar este problema, es necesario resaltar alguna otra característica distintiva de la pupila, una segmentación por color y forma aumenta las probabilidades de ubicar las partículas en las zonas reales del objetivo, incrementando la precisión en la estimación de la posición del filtro y asegurando un seguimiento a lo largo de las iteraciones. La segmentación por forma consiste en una búsqueda de un contorno específico en toda imagen o partes de la imagen, para descartar aquellas zonas que no cumplen con las características geométricas establecidas, en el capítulo 3 se profundiza sobre la técnica empleada para segmentar la pupila de los ojos.

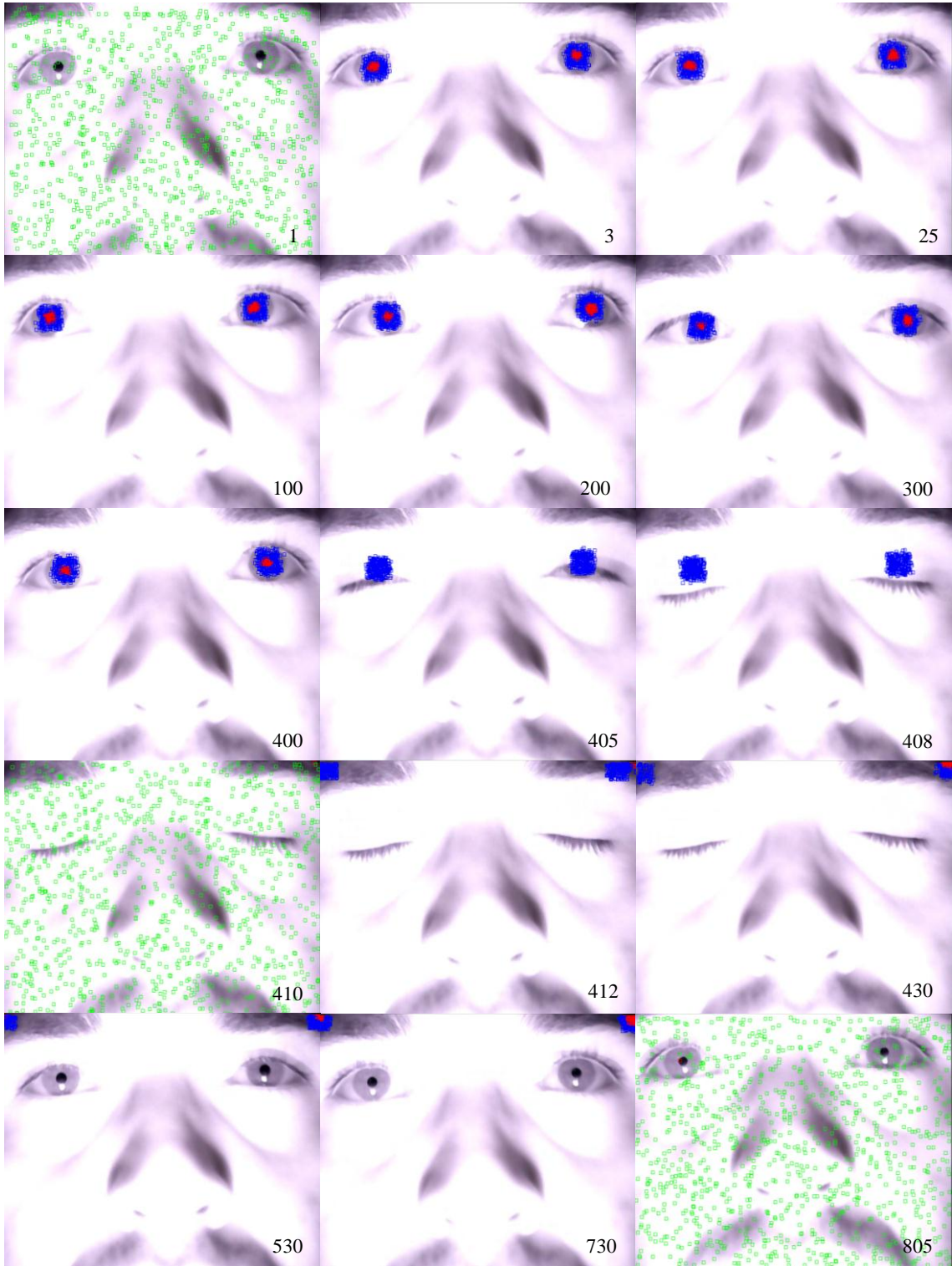


Imagen 2.7 Resultado del seguimiento multimodal.

Capítulo 3.

ALGORITMO INTELIGENTE

En este capítulo se describen los diferentes métodos propuestos aplicados al filtro de partículas multimodal, los cuales aumentan su precisión y respuesta al momento de estimar la posición del objetivo. Para ello, inicialmente se describe la segmentación por forma utilizada para mejorar la ubicación de las partículas, posteriormente se define el algoritmo inteligente y sus etapas, el cual es utilizado para realizar la hibridación con el filtro.

3.1 Segmentación por forma.

Una de las formas de reconocer objetos dentro de una imagen (o una región segmentada) es extraer características medibles de ellos en un sistema de visión artificial, como lo es la forma, el color o la textura, estas características permiten definir *vectores de características* que se usan para el reconocimiento de patrones.

Puede resultar difícil el reconocimiento de la forma de un objeto, ya que se necesita definir gráficamente la figura exterior o geometría del cuerpo, sin embargo, es posible hacerlo describiendo la forma 2D mediante algunas características y propiedades como son el área, el perímetro, la envolvente, distancias al centro de masa, entre otras.

En este caso, se requiere la descripción de una forma circular (pupila), para lo cual, es muy utilizada la transformada de Hough [33], que consiste básicamente en realizar un recorrido por todos los puntos de frontera o borde del objeto que se quiere caracterizar, planteando los infinitos círculos que pasan por ese punto con diferente radio y diferente centro, acumulando un voto en un espacio de parámetros de tres dimensiones (coordenadas x e y del centro y radio) para aquellas curvas que cumplan con la ecuación:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (3.1)$$

Para encontrar los bordes de la figura, es necesario aplicar un algoritmo que realice esta tarea, el más conocido de los filtros detectores de bordes es el filtro de Canny [34], que realiza una sencilla detección de contornos de los objetos de la imagen y de esta forma es posible aplicar la transformada de Hough para identificar las formas circulares. En la imagen 3.1 se observa el resultado del filtrado en una imagen que contiene la pupila.

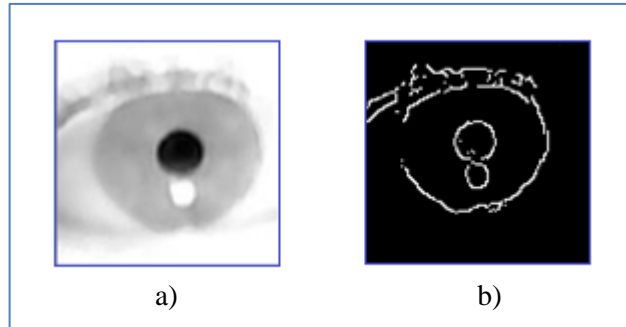


Imagen 3.1 Imágenes procesadas. a) Imagen Original en escala de gris, b) Imagen resultado de aplicar el filtro de Canny.

Con el filtro de Canny aplicado en la imagen, es posible ejecutar la transformada de Hough para encontrar las diferentes formas circulares, para ello se hace uso de la función presente en la librería OpenCV llamada *cvHoughCircles* [35], donde es posible establecer los parámetros para realizar la búsqueda, como por ejemplo, el número de círculos a encontrar, la distancia entre los círculos, el radio máximo y mínimo. En la Imagen 3.2 es posible observar la detección de la forma circular proporcionada por la transformada de Hough.

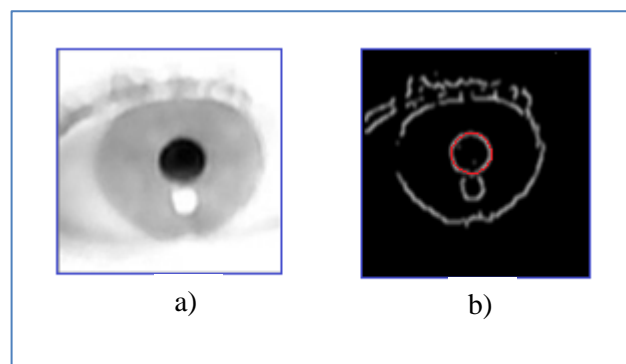


Imagen 3.2 Segmentación por forma. a) Imagen original, b) Imagen con forma circular (color rojo) encontrada con la Transformada de Hough.

Cómo se mencionó anteriormente, la búsqueda de formas circulares complementa la segmentación por color para mejorar la estimación del estado del sistema, con el objetivo de aumentar la eficiencia del algoritmo y a diferencia de lo que se realiza normalmente, se

propone no recorrer toda la imagen en búsqueda de la pupila, sino acudir a las mejores hipótesis halladas por el filtro para realizar una búsqueda en esas zonas. Esto delimita la búsqueda a dos secciones de 100 x 100 píxeles de la imagen, donde el filtro estima que se encuentran las pupilas, pero además reduce la posibilidad de encontrar formas circulares si existiesen en otras partes de la escena.

Una vez se determinada las zonas más probables de ser las pupilas, el algoritmo de optimización puede aumentar la calidad de las hipótesis, realizando una búsqueda exhaustiva y posicionando de mejor manera las partículas. En esta investigación se propone la utilización del algoritmo de búsqueda dispersa como se ha realizado en [7], [36], pero a diferencia de ellos y de la hibridación común, no se toma el conjunto de referencia del vector de partículas del filtro para inicializar el algoritmo, sino que se utilizan estas zonas segmentadas, en especial la zona delimitada por la forma circular, dentro la cual se generan nuevas partículas aleatorias con las que se crea el conjunto de referencia para iniciar la búsqueda dispersa. Esta forma de inicializar el algoritmo de optimización garantiza que las mejores soluciones se posicionen dentro de la pupila, por lo cual aumenta considerablemente la precisión en la estimación y además la supervivencia de estas hipótesis para el siguiente instante.

3.2 **Búsqueda Dispersa.**

Durante los últimos años, se ha desarrollado un interés particular por el uso de algoritmos inteligentes, en especial de metaheurísticas en la optimización combinatoria, para la solución de problemas dinámicos utilizando algoritmos híbridos, arrojando buenos resultados en este tipo de problemas [7].

Dentro del desarrollo de una optimización dinámica no solo se debe tener en cuenta la estrategia de optimización, sino también, una buena estrategia de adaptación, es decir, identificar la información útil y cómo se puede transferir a los siguientes instantes de tiempo con el fin de encontrar buenas soluciones en el menor tiempo posible. En este sentido, para el desarrollo del presente trabajo se intenta dar respuesta a este planteamiento mediante la hibridación entre el Filtro de Partículas Multimodal y el método de optimización de Búsqueda Dispersa, dados los requerimientos de coste computacional y velocidad de procesamiento.

La Búsqueda Dispersa es un procedimiento metaheurístico basado en formulaciones y estrategias introducidas durante la década de los setenta. Este método evolutivo ha sido aplicado en la resolución de un gran número de problemas de optimización y consiste en, dado un conjunto de soluciones, obtener otras nuevas mediante su combinación, de modo que mejore a las que la originaron. Para ello, dado un conjunto de soluciones diversas, se

escogerá un número, no muy elevado, de esas soluciones conformando el conjunto de referencia. Después de realizar una etapa de combinación sobre el conjunto de referencia, se obtendrán mejores soluciones que las iniciales. En este sentido, se dice que la Búsqueda Dispersa es un algoritmo evolutivo, ya que mantiene una población de soluciones a cada momento, y a diferencia de muchos otros algoritmos, este no está fundamentado en una aleatorización sobre un conjunto grande de soluciones, como por ejemplo un Algoritmo Genético, sino elecciones sistemáticas y estratégicas sobre un conjunto pequeño [7].

La importancia de la Búsqueda Dispersa radica en que con un número pequeño de soluciones se pueden obtener mejores soluciones que con cualquier otro algoritmo que utilice un conjunto mayor, lo que minimiza el coste computacional y, dependiendo del método de combinación, agiliza la búsqueda de soluciones de alta calidad.

Este algoritmo presenta las siguientes etapas definidas, y se puede representar mediante el diagrama de estados de la Figura 3.1:

1. Construir el RefSet: se elige un subconjunto de soluciones del conjunto original y se ordena de mejor a peor calidad, es decir de mayor a menor peso.
2. Se crea un conjunto RefSetN igual al conjunto RefSet.
3. Se aplica el método de combinación sobre el conjunto RefSetN.
4. Se realiza una mejora sobre las combinaciones del RefSetN.
5. Si los conjuntos fuesen distintos, se haría $\text{RefSet} = \text{RefSetN}$ y se volvería a aplicar el método, en caso de que no se puedan mejorar las soluciones, la Búsqueda Dispersa terminaría.

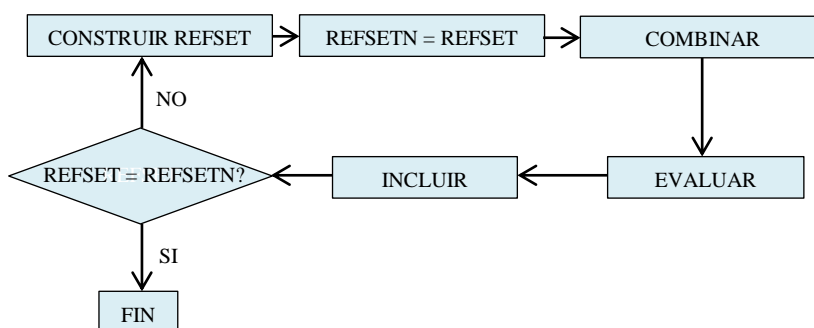


Figura 3.1 Diagrama de estados de la Búsqueda Dispersa

Una de las principales ventajas de utilizar este algoritmo para realizar la hibridación con el Filtro de Partículas, es la flexibilidad que presenta al permitir adaptarlo a los requerimientos de uso, es decir, la Búsqueda Dispersa brinda la oportunidad de escoger el método de combinación de soluciones y también el método de mejora de esas soluciones, haciendo

variable su complejidad dependiendo de la función que deba cumplir. Dicho esto, para el desarrollo de este trabajo se ha establecido utilizar un método de combinación sistemático por emparejamiento y para mejorar estas soluciones se decidió utilizar el algoritmo de búsqueda local, ambos procedimientos se detallan a continuación, donde se describe la funcionalidad que cumplen dentro del algoritmo inteligente.

3.2.1 Método de Combinación.

Una de las etapas más importantes de la Búsqueda Dispersa es la combinación, ya que con ella se obtienen las nuevas soluciones a evaluar, lo que quiere decir que dependiendo del procedimiento de combinación escogido, la calidad de las soluciones puede mejorar o empeorar dado que permite explorar regiones diferentes del espacio de búsqueda, y eso la convierte en una parte fundamental del algoritmo.

En este trabajo se utilizará la combinación lineal por emparejamiento, esto consiste en crear parejas o subconjuntos de soluciones para generar una nueva solución. Como se indicó anteriormente, el RefSet se ordena con soluciones de mayor a menor calidad, para este caso en particular se trata de 10 soluciones generadas aleatoriamente dentro de la pupila del ojo. Las combinaciones se realizan de tal forma que se mantenga el número inicial de la población, es decir que se efectúan 10 combinaciones, de tal forma que una solución de buena calidad se combine con una de baja calidad, como se muestra en la figura 3.2, donde S1 es la solución de más alta calidad y S10 la de más baja, se aprecia la combinación de la solución S1 de alta calidad con las soluciones S5 y S6 de más baja calidad, para generar dos nuevas soluciones que más adelante serán evaluadas.

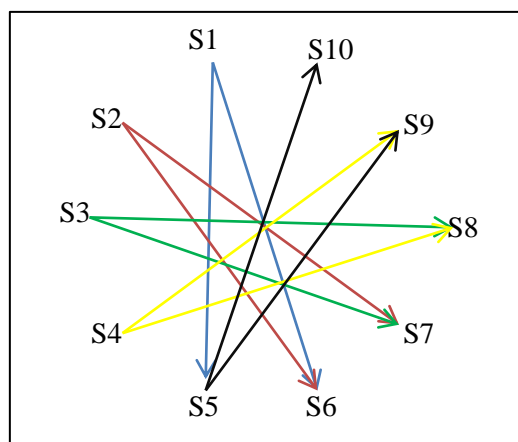


Figura 3.2 Diagrama de Combinación de soluciones.

Una vez determinado los emparejamientos de las soluciones, se procede a realizar la combinación lineal (en el espacio euclídeo) de cada una de ellas. De esta forma, existen

muchas variantes para obtener la nueva solución, la más usual de ellas consiste en encontrar el punto medio entre la recta que une las dos soluciones a combinar y generar la nueva solución en este punto, como lo muestra la figura 3.3. Donde, por ejemplo, la nueva solución S15 se genera en el punto medio de la distancia que une a S1 y S5 por una recta.

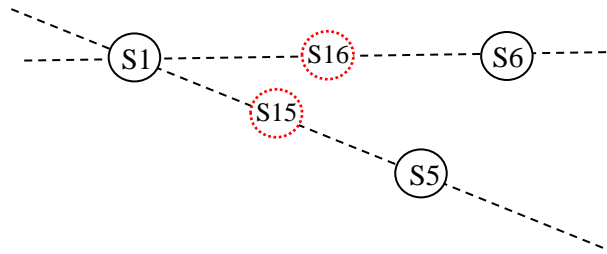


Figura 3.3 Combinación de dos soluciones.

De este modo se pueden generar las nuevas soluciones que conforman el RefSetN para mejorar el conjunto inicial, aplicado esto en las partículas generadas, el punto medio se halla promediando las coordenadas en X e Y de las dos soluciones que se combinan. El algoritmo también permite implementar una etapa de mejora de esas soluciones generadas, típicamente se utiliza la búsqueda local como etapa de mejora ya que es un procedimiento sencillo de realizar y se obtienen buenos resultados con su implementación, aunque se puede utilizar cualquier metaheurística trayectorial.

3.2.2 Método de mejora.

Al resultado de la combinación de las soluciones, se le aplicará un método de mejora, como se mencionó anteriormente, la Búsqueda Local ha dado buenos resultados para esta etapa, por lo que se implementará en este trabajo.

El fin de la Búsqueda Local consiste en mejorar la solución dada por el método de combinación a la solución real. Para ello se recorrerán los 8-vecinos de la solución, siempre y cuando se vaya mejorando la solución y se tomará, si existe, aquel vecino que mejore la solución inicial. En este caso, los 8-vecinos se toman alrededor de cada partícula, es decir las ventanas de 7 píxeles alrededor de la partícula serán evaluadas para definir si existe alguna de mejor calidad que la inicial, esto es, si la partícula presenta coordenadas (x, y) , los vecinos que recorre el algoritmo son aquellos que tengan las coordenadas $(x - 7, y)$, $(x, y - 7)$, $(x + 7, y)$, $(x, y + 7)$, $(x - 7, y + 7)$, $(x - 7, y - 7)$, $(x + 7, y - 7)$, $(x + 7, y + 7)$, en la figura 3.4 se puede observar los 8-vecinos de una partícula de 7 x 7 píxeles que tiene como estado la coordenada (x, y) .

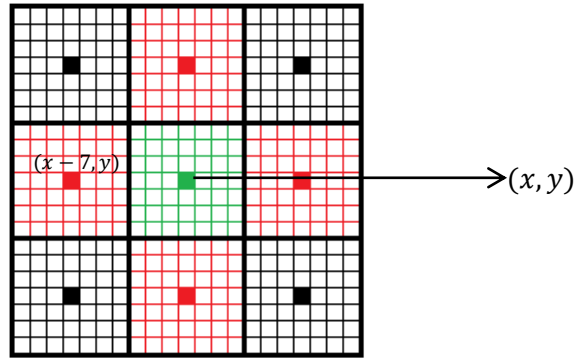


Figura 3.4 Representación de los 8-vecinos.

Al aplicar este método de mejora, se obtienen soluciones de mayor calidad en el espacio de búsqueda explorado por el algoritmo y brinda buenas soluciones a la etapa de creación de la ruleta que reemplaza las de más baja calidad, lo que permite aumentar la precisión en la estimación del estado y por lo tanto el posicionamiento de la mirada en la pantalla.

3.3 Hibridación del Filtro de Partículas con la Búsqueda Dispersa.

Para realizar la hibridación entre el Filtro y la Búsqueda Dispersa, se creará un conjunto de referencia RefSet con 10 soluciones generadas aleatoriamente dentro de la cada zona objetivo proporcionada por el Filtro de Partículas y la segmentación por forma, estas soluciones se representan en la imagen como una ventana de 7x7 píxeles semejante a la usada para representar las partículas. También se creará un conjunto RefSetN, igual a RefSet que se utilizará para comprobar si se han mejorado las soluciones iniciales, donde, mediante la distancia de Bhattacharyya, se obtendrán los pesos de esas soluciones con el fin de ordenarlas de mayor a menor calidad y así establecer las parejas con las que se ejecuta el método de combinación sistemática lineal. Esta combinación se realiza encontrando el punto o píxel medio de la distancia entre las dos soluciones que generarán la nueva partícula de mejor calidad, ubicada en dicho punto. Al final, el método dará como resultado 10 soluciones nuevas en cada zona objetivo almacenadas en el vector RefSetN que, al igual que las originales, se evalúan mediante la distancia de Bhattacharyya para comprobar si son mejores, esto se hace comparando el vector RefSetN con el RefSet. De ser así, se introducen estas nuevas soluciones al vector de partículas, garantizando la exploración de los máximos locales para las siguientes iteraciones y eliminando partículas de baja calidad. En la Figura 3.5 se muestra el diagrama de bloques del filtro de partículas multimodal híbrido con la segmentación por forma y búsqueda dispersa.

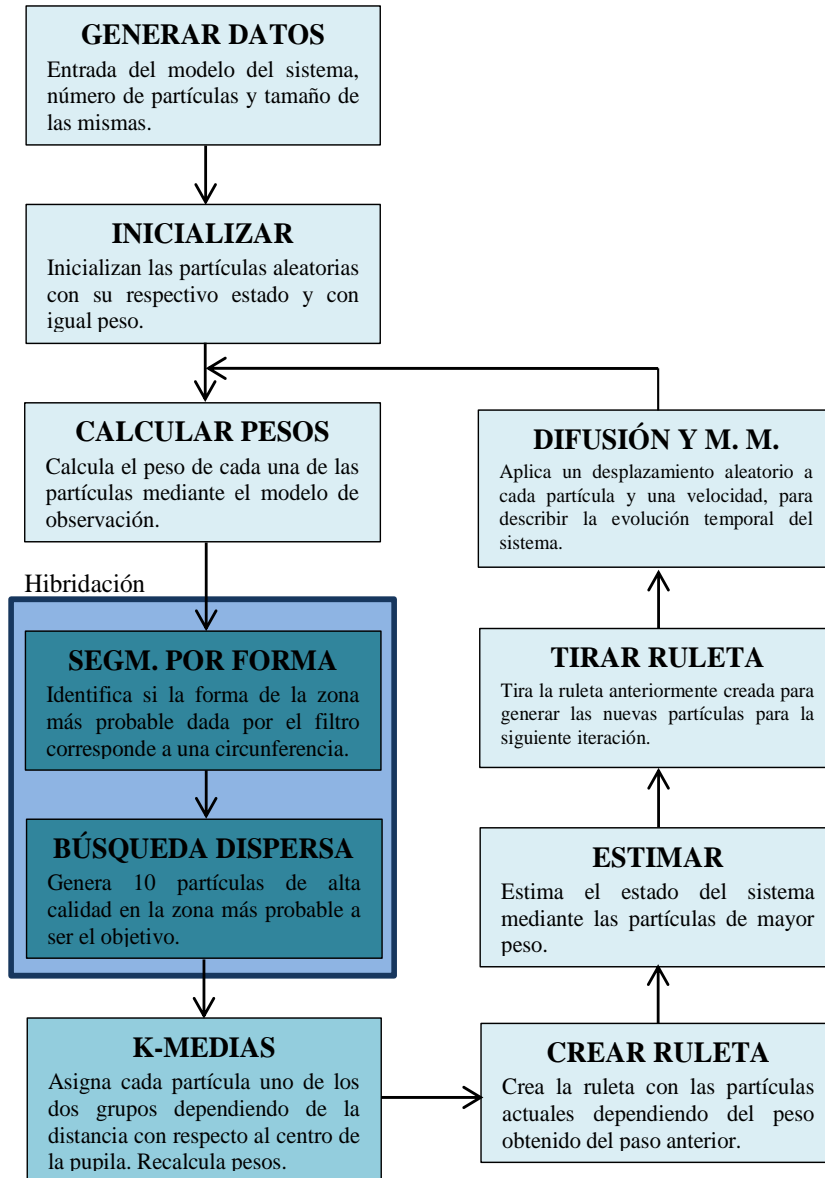


Figura 3.5 Diagrama de bloques del Filtro de Partículas Híbrido.

En la Imagen 3.3 se puede observar el funcionamiento del filtro de partículas híbrido sobre una secuencia de fotogramas, la ilustración permite ver que durante la etapa de inicialización se ubican partículas de alta calidad sobre las pupilas, y una vez encontradas realiza el seguimiento de las mismas hasta antes del fotograma 410, donde el usuario realiza un parpadeo involuntario y el filtro se reinicia, pero una vez que se abren los ojos nuevamente (fotograma 460), se detecta rápidamente los objetivos y se mantiene el seguimiento.

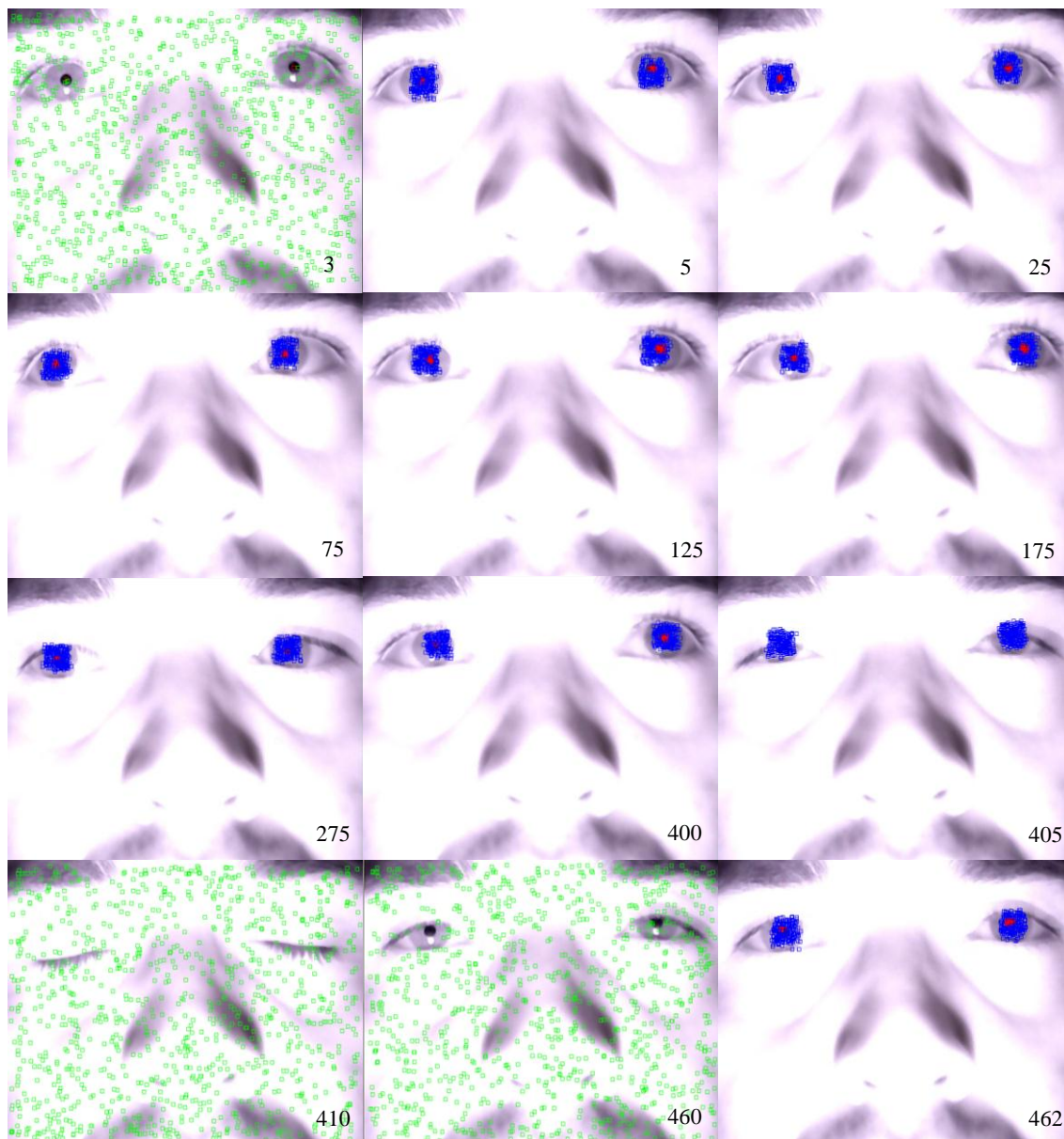


Imagen 3.3 Funcionamiento del filtro de partículas híbrido.

El algoritmo inteligente soluciona el problema presentado por el filtro multimodal expuesto en la sección 2.3, que delimitaba el modelo de observación simplemente a la caracterización por color mediante los histogramas. Con la ayuda de las etapas de hibridación en el filtro multimodal, se puede asegurar que el algoritmo realiza un seguimiento ocular, y que mantiene dicho seguimiento a lo largo de las iteraciones.

En el capítulo 5 se muestran los resultados del filtro de partículas híbrido integrando el posicionamiento en la pantalla, se analiza el efecto de la variación del número de partículas en el seguimiento, se describen los tiempos de ejecución y la precisión de la estimación del estado.

Capítulo 4.

SISTEMA DE COORDENADAS INCREMENTALES E INTERFAZ

En este capítulo se presenta el Sistema de Coordenadas Incrementales, el cual es utilizado para posicionar la mirada del usuario interpretada por el filtro de partículas, en la pantalla donde se despliega la interfaz. También se hace una descripción de la interfaz de usuario utilizada, con el fin de validar todo el sistema de seguimiento y posicionamiento de la mirada.

4.1 Sistema de Coordenadas Incrementales.

El sistema de Coordenadas Incrementales [16] difiere del sistema de coordenadas absoluto o del cartesiano, en la ubicación de la referencia en plano de coordenadas, es decir, mientras para el sistema de coordenadas cartesianas la referencia es siempre la misma (el origen) respecto a los demás puntos, para el sistema de coordenadas incrementales se utiliza la posición actual como referencia para el siguiente movimiento, es como si se trasladara el origen al último punto alcanzado, esto se puede apreciar en la figura 4.1.

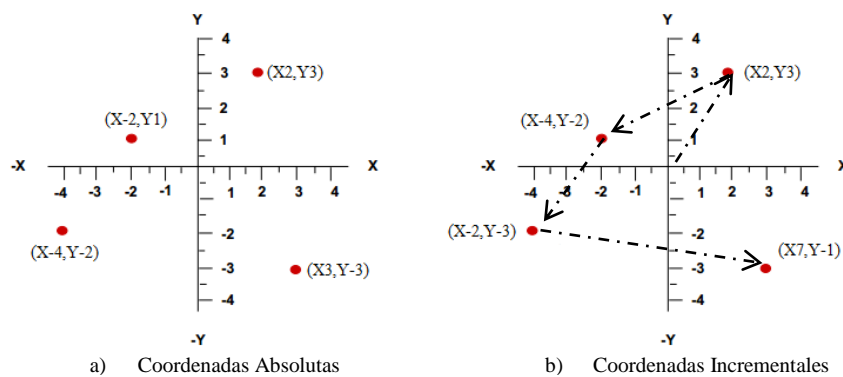


Figura 4.1 Sistema de coordenadas.

Como se mencionó anteriormente, este sistema de coordenadas se utiliza para posicionar la mirada del usuario en la pantalla donde se despliega la interfaz, controlando la posición y la velocidad del cursor durante la interacción. El usuario comienza fijando la mirada en el punto inicial O , que se traduce en la coordenada del ojo (Xe_o, Ye_o) en la imagen, también se establece la coordenada inicial de la posición del cursor en la pantalla (Xm_o, Ym_o) . Cuando se presenta un movimiento del ojo hacia otro punto de la pantalla O' , que en la imagen es (Xe_n, Ye_n) , el cursor deberá moverse hacia el punto (Xm_n, Ym_n) , de acuerdo a la ecuación 4.1, el siguiente punto es determinado a partir de la antigua coordenada del cursor.

$$(Xm_n, Ym_n) = V \times \vec{M} + (Xm_{n-1}, Ym_{n-1}), \quad (4.1)$$

donde \vec{M} es un vector que depende de las diferentes direcciones del movimiento, tal como lo muestra la Tabla 4.1, V es el parámetro de la velocidad que varía dependiendo de la distancia entre el punto anterior O y el punto actual O' , a mayor distancia, mayor velocidad (Ver figura 4.2), este parámetro está definido por:

$$\begin{aligned} |\overline{OO'}| = 0, & \quad V = 0, \\ 0 < |\overline{OO'}| < R, & \quad V = 1 \\ |\overline{OO'}| > R, & \quad V = 0.1 * |\overline{OO'}|, \end{aligned} \quad (4.2)$$

donde R es la referencia en píxeles que define la velocidad del cursor dependiendo de la distancia $|\overline{OO'}|$, en este caso $R = 10$, indicando que cuando la distancia entre el punto anterior y el punto actual es menor a 10, la velocidad de movimiento del cursor es igual a 1 pixel/fotograma y cuando la distancia entre el punto anterior y el punto actual es mayor que 10, la velocidad del movimiento es el 10% de esa distancia.

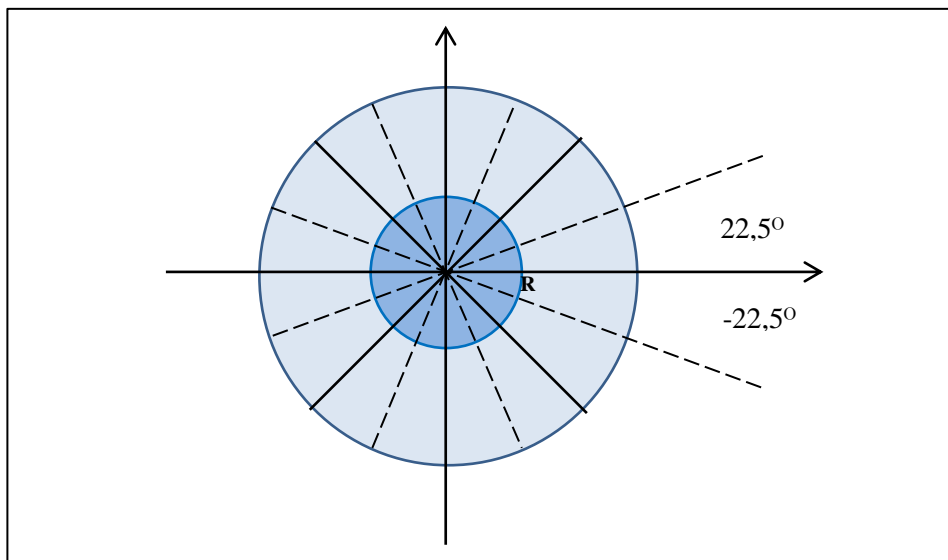


Figura 4.2 Diagrama del Sistema de Coordenadas Incrementales [Adoptado de 13]

Región	Ángulo	\vec{M}
1	$337.5^\circ < \angle \overrightarrow{OO'}$ < 22.5°	(1,0)
2	$22.5^\circ < \angle \overrightarrow{OO'}$ < 67.5°	(1,1)
3	$67.5^\circ < \angle \overrightarrow{OO'}$ < 112.5°	(0,1)
4	$112.5^\circ < \angle \overrightarrow{OO'}$ < 157.5°	(-1,1)
5	$157.5^\circ < \angle \overrightarrow{OO'}$ < 202.5°	(-1,0)
6	$202.5^\circ < \angle \overrightarrow{OO'}$ < 247.5°	(-1,-1)
7	$247.5^\circ < \angle \overrightarrow{OO'}$ < 292.5°	(0,-1)
8	$292.5^\circ < \angle \overrightarrow{OO'}$ < 337.5°	(1,-1)

Tabla 4.1 Tabla de ajuste del vector \vec{M} .

Para determinar las direcciones del movimiento, se encuentra el ángulo existente entre el punto anterior de la mirada O y el punto actual O' , es decir el ángulo entre el estado del sistema en el instante $k - 1$ y el estado del sistema en el instante k . Para ello se realiza una serie de consideraciones con el fin de establecer dos rectas y aplicar la geometría de ángulos cómo se muestra en la Figura 4.3, la primera recta L_1 se establece entre el punto O y el punto O' y la segunda L_2 se establece entre los puntos O' y P . La coordenada P se utiliza para crear la recta horizontal con la que se referencia el ángulo a encontrar, en este caso P se establece como el punto O' desplazado 10 píxeles en el eje x , con lo que siempre se obtendrá que $\angle O'P = 0^\circ$ y el ángulo a encontrar se define como $\theta = \angle \overrightarrow{OO'} = \angle \overrightarrow{OO'} \overrightarrow{OP} = \angle L_1 L_2$.

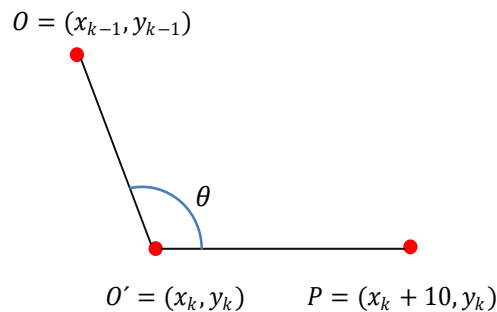


Figura 4.3 Ángulo entre dos rectas

Ahora es posible encontrar θ que se determina como el cociente entre el producto punto de los vectores directores de las rectas y la multiplicación de sus módulos como se muestra en la ecuación 4.3.

$$\theta = \cos^{-1} \frac{VD_{L1} \times VD_{L2}}{|VD_{L1}| |VD_{L2}|} \quad (4.3)$$

En primer lugar se definen los vectores directores de cada una de las rectas L_1 y L_2 de la siguiente manera:

$$VD_{L1} = (x_k - (x_k + 10), \quad y_k - y_k) \quad (4.4)$$

$$VD_{L2} = (x_{k-1} - x_k, \quad y_{k-1} - y_k) \quad (4.5)$$

Con ello, se puede concluir que el vector director de la recta L_1 no va a cambiar a lo largo de las iteraciones por lo que:

$$VD_{L1} = (-10, \quad 0) \quad (4.6)$$

El producto punto entre dos vectores VD_{L1} y VD_{L2} se define como:

$$VD_{L1} \times VD_{L2} = [(-10 * (x_{k-1} - x_k)) + (0 * (y_{k-1} - y_k))] \quad (4.7)$$

Que se puede expresar como:

$$VD_{L1} \times VD_{L2} = 10 * (x_k - x_{k-1}) \quad (4.8)$$

Finalmente el producto entre dos módulos se define de la siguiente manera:

$$|VD_{L1}| |VD_{L2}| = (-10^2 + 0^2) * (x_{k-1} - x_k)^2 + (y_{k-1} - y_k)^2 \quad (4.9)$$

Que se puede resumir en:

$$|VD_{L1}| |VD_{L2}| = 100 * (x_{k-1} - x_k)^2 + (y_{k-1} - y_k)^2 \quad (4.10)$$

Una vez encontrado el ángulo θ es posible determinar la dirección del movimiento del cursor ajustando el vector \vec{M} de la ecuación 4.1, esta dirección del cursor posee una velocidad V que depende de la distancia $|\overline{OO'}|$ estimada anteriormente. De esta manera es posible relacionar la posición en que el usuario fija la mirada, la cual es estimada por el filtro y la posición del cursor en la pantalla.

4.2 Interfaz.

Con el propósito de verificar el funcionamiento del sistema de coordenadas incrementales aplicado al posicionamiento de una coordenada en una pantalla, es necesario utilizar una interfaz en la que se pueda evaluar el algoritmo. Para el desarrollo de este trabajo, se ha

decido utilizar como interfaz de usuario el entorno gráfico de Windows 7, en el que se pueden realizar acciones como dar clic sobre algunos iconos, abrir o cerrar ventanas, entre otros, teniendo en cuenta que es uno de los sistemas operativos más utilizados en el mundo y por lo tanto es muy familiar para cualquier usuario.

Capítulo 5.

PLATAFORMA EXPERIMENTAL, PRUEBAS Y RESULTADOS

En este capítulo se describen los componentes hardware de la aplicación, es decir la cámara IR-LED y la fuente de luz infrarroja, además de ello se hace una descripción informática de la aplicación. En una segunda parte de este capítulo se muestran los resultados y el análisis obtenido con la aplicación en cuanto al seguimiento y el posicionamiento de la mirada.

5.1 Creación de la cámara IR-LED

Como se mencionó anteriormente, la técnica usada para medir los movimientos de la pupila del ojo es la detección por video basada en la pupila y la reflexión córnea, donde la secuencia de imágenes es adquirida con una cámara sensible a la luz infrarroja. Para conseguir este tipo de sensor es necesario hacer una serie de modificaciones a las cámaras digitales comunes para PC, ya que normalmente disponen de un sensor CMOS (*Complementary Metal Oxide Semiconductor*, por sus siglas en inglés) el cual es sensible tanto a la luz visible, hablando en términos del ojo humano, como también a la luz infrarroja, solo que cuentan con un filtro que no permite que la luz infrarroja sature los colores visibles y genere imágenes irreales. La modificación a la cámara consiste en retirar este filtro y colocarle un filtro denominado caliente, que permite solo el paso de luz infrarroja al sensor, bloqueando la visible.

Los filtros calientes comerciales suelen ser de un costo elevado, es por eso que para este propósito y manteniendo el bajo costo del proyecto, es posible reemplazar este vital componente con un trozo de negativo de fotografía velado, que cumple con la misma función y se obtienen muy buenos resultados.

La iluminación también es una parte fundamental del proceso de adquisición de las imágenes, en este caso, la fuente de luz infrarroja proviene de un anillo de 21 IR-LEDs concéntricos (*Imagen 5.1*) de tal forma que puedan iluminar todo el rostro del usuario. Estos LEDs emiten una luz con una longitud de onda de 950 nm, valor dentro del rango de longitudes de ondas no perjudiciales y no visibles para el ojo humano, por lo que no altera las variaciones de diámetro de la pupila. La disposición circuital de los LEDs se presenta en 7 grupos en paralelo, cada grupo cuenta con tres LEDs en serie limitados por una resistencia de 330 Ohmios.



Imagen 5.1 Fuente de luz infrarroja.

Esta fuente de luz infrarroja se alimenta con una fuente de alimentación comercial de 12 voltios corriente directa a 1 amperio, que a su vez se conecta a la red eléctrica de 110 voltios corriente alterna.

5.2 Disposición Hardware del sistema de seguimiento ocular.

El conjunto cámara, emisor de luz infrarroja y ordenador deben presentar una disposición adecuada para permitir una buena adquisición de las imágenes y por lo tanto un buen funcionamiento de la aplicación. El sistema fue diseñado para operar sobre un computador portátil debido al amplio uso de estos dispositivos tecnológicos, pero es fácilmente adaptable a computadores de escritorio.

En la Imagen 5.2 se presenta la ubicación de los componentes del sistema, el emisor de luz se ubica en la parte superior de la pantalla del ordenador y en línea sobre la cámara, de manera que los haces de luz se direccionen sobre el rostro del usuario. La cámara IR se posiciona en la parte central baja de la pantalla y enfoca su lente sobre la cara del usuario, de modo que solo se observe el área entre la frente y la nariz, sin capturar el fondo del lugar. El usuario se ubica frente al ordenador de tal forma que el rostro se centre con la pantalla y con la cámara a distancia aproximada de 60 cm, esta distancia puede variar un poco, siempre y cuando se cumpla con la condición de que el área de los ojos aparezca en la imagen.

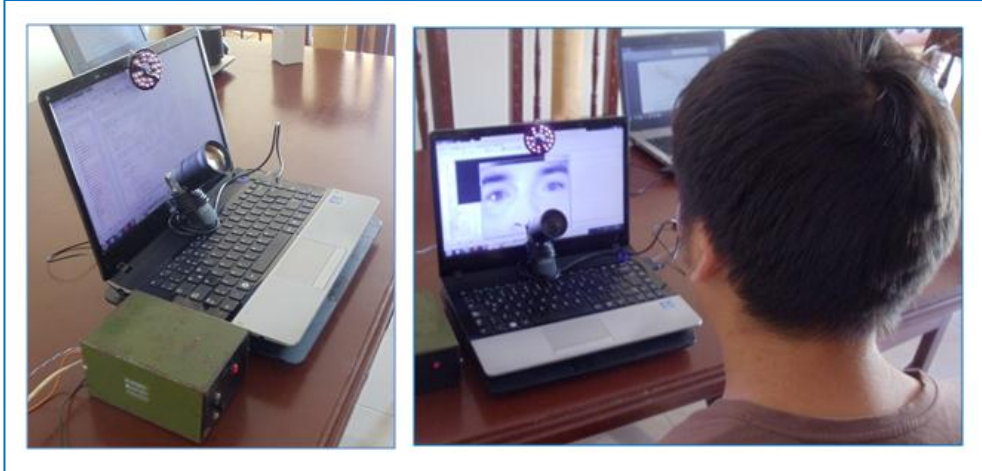


Imagen 5.2 Disposición hardware de la aplicación.

5.3 Descripción Informática.

Los proyectos informáticos deben cumplir una serie de condiciones, no es suficiente con que solo funcionen, además de eso, deben hacerlo en un tiempo razonable, ser adaptable a distintas situaciones, etc.

Dentro de los requisitos funcionales, se plantea que el Filtro de partículas debe ser capaz de seguir dos objetos al tiempo en una secuencia de imágenes, estos objetos deben presentar las mismas características de color ya que se utiliza un solo modelo de observación. También hay que tener en cuenta que dependiendo del número de partículas que se utilicen, el resultado del filtro puede cambiar, destacando que, cuantas más partículas se generen, mejora la precisión del seguimiento, pero la ejecución de la aplicación requerirá más tiempo y consumirá más recursos. Otro aspecto a tener en cuenta, radica en la capacidad del filtro para recuperarse en caso de perder el objeto durante determinado tiempo, esto se debe a que si la velocidad de movimiento del objeto es alta, el filtro no logra seguir con precisión el movimiento, debido a la limitación de velocidad de captura de la imagen y la capacidad de procesado de la aplicación.

Para los requisitos no funcionales, que describen como debe reaccionar la aplicación, es necesario capturar imágenes con una webcam para analizarlas, reconocer las pupilas y poder realizar el seguimiento de estas, eso sí, la cámara debe proporcionar las suficientes imágenes por segundo para poder seguir el objeto en un tiempo razonable, casi inmediato. Un punto importante dentro de estos requisitos, radica en la posición de la cámara, si esta varía durante la ejecución del filtro, es necesario detener la aplicación y volver a empezar de nuevo, ya que la calibración de la cámara es errónea y los resultados obtenidos no serán correctos.

Dentro de la parte de detección de las pupilas, es necesario realizar una comparación de histogramas, lo que conlleva a disponer de una imagen del objeto a seguir almacenada, esta imagen se convierte en la referencia para el modelo de observación y determinar la verosimilitud entre cada partícula y este modelo.

5.4 Requisitos de Desarrollo.

Para concluir exitosamente este proyecto, se ha seguido una metodología de desarrollo en espiral, debido a que permite una implementación incremental e iterativa y se pueden realizar tantas iteraciones como sea necesario para cumplir con los objetivos del proyecto. Este último se ha dividido en distintos ciclos:

- A. Análisis de la fase del proyecto a abordar.
- B. Estudio de la documentación relacionada.
- C. Discusión de las posibles alternativas para esa fase.
- D. Codificación.
- E. Pruebas y depuración.
- F. Pequeña documentación del trabajo realizado.

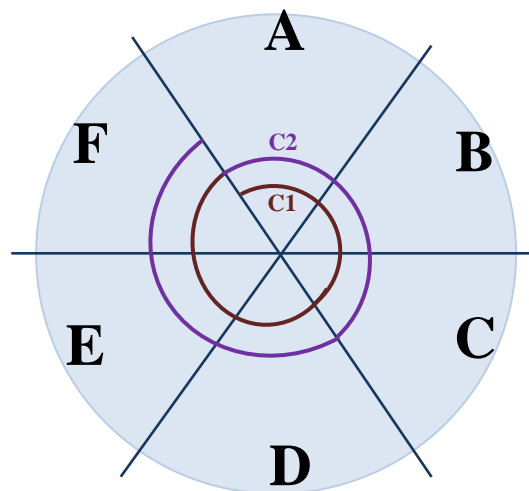


Figura 5.1 Modelo del ciclo de vida en espiral.

5.5 Implementación del Software.

Tomando en cuenta que la aplicación funcional requiere procesamientos en tiempos casi inmediatos para su ejecución, de tal manera que el cursor pueda seguir fielmente los movimientos del ojo del usuario y en los mejores tiempos de respuesta, resulta necesario utilizar un lenguaje de programación de bajo nivel. Dentro de una búsqueda para dar solución a las necesidades informática se encontró que la librería OpenCV (*Open Source*

Computer Vision) cuenta con más de 500 funciones diseñadas especialmente para el procesamiento de imágenes en tiempo real abarcando una gran área de la visión artificial. De esta manera se decide la utilización de la librería OpenCV para el desarrollo del software y con ella el lenguaje de programación C++.

Mediante la utilización del entorno de desarrollo Dev-C++, se diseñaron e implementaron las funciones y procedimientos de la cual se compone la aplicación creada, estas funciones hacen uso de la librería mencionada anteriormente y la descripción algorítmica de cada una de ellas se encuentra en la sección 2 de los anexos a este trabajo.

El resultado de implementar estas funciones y procedimientos es la aplicación Eye Tracking presentando una interfaz como la que muestra la Imagen 5.3, que permite al usuario calibrar de forma manual el foco de la cámara del sistema, la ubicación de la fuente de luz infrarroja y la posición del rostro para la ejecución.

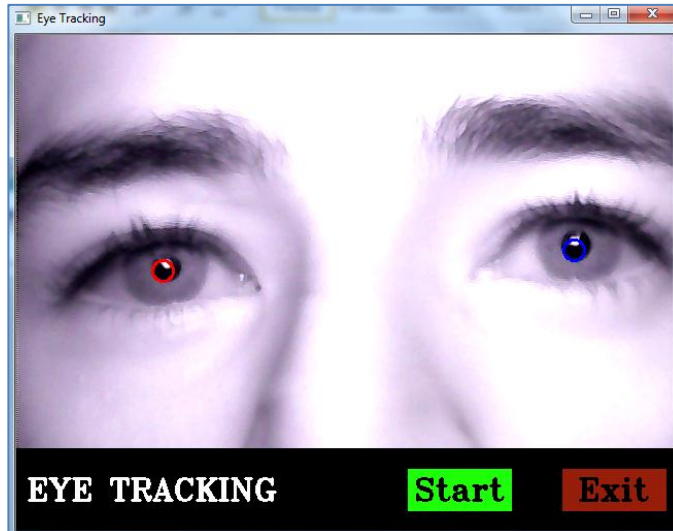


Imagen 5.3. Interfaz de la aplicación.

Una vez realizada la calibración, el usuario puede iniciar la manipulación del cursor oprimiendo el botón **Start**, en cuyo caso se cierra la ventana de la interfaz y la aplicación se apodera del puntero desplazándolo de acuerdo a los movimientos de las pupilas. Par salir del programa, es necesario que el usuario se ubique fuera del foco de la cámara para que aparezca de nuevo la ventana de la interfaz, dónde con el mouse del ordenador puede dar clic sobre el botón **Exit** cerrando la aplicación.

5.6 Pruebas y resultados.

En esta sección se describirán las pruebas realizadas para evaluar, analizar y comparar los algoritmos implementados, del Filtro de Partículas y el Filtro de Partículas Híbrido, de esta manera es posible obtener conclusiones y determinar algunas limitaciones de la aplicación.

Para el entorno de trabajo se ha utilizado un ordenador con procesador Intel Core I3 de 2.39GHz de velocidad y con memoria RAM de 4 Gb. El sistema operativo es Windows 7 Home Premium de 64 bits. La aplicación se ha desarrollado íntegramente en lenguaje C++ utilizando el IDE Dev-C++.

Para realizar pruebas sobre la aplicación se han empleado diversos vídeos que permiten destacar características de los algoritmos y el comportamiento que presentan en el seguimiento de los objetivos.

5.6.1 Pruebas de detección.

La detección de los objetivos es un factor importante dentro de la aplicación, para determinar el número adecuado de partículas en la etapa de búsqueda del filtro híbrido, se ha utilizado la secuencia de imágenes de la imagen 5.4, donde se calcula el número de iteraciones necesarias para encontrar las pupilas y el tiempo que le lleva al filtro realizar esto. La prueba se realiza con el filtro de partículas híbrido ya que esta etapa de búsqueda es igual para ambos filtros.

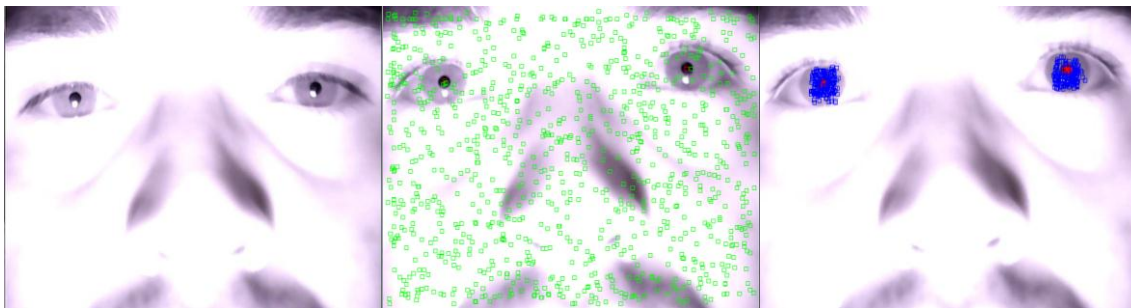


Imagen 5.4 Secuencia de imágenes para búsqueda.

En la tabla 5.1 se muestran los resultados obtenidos en la detección de las pupilas, para ello se varió el número de partículas de búsqueda.

Número de pruebas	500 Partículas		1000 Partículas		2000 Partículas	
	Tiempo (ms)	Iteraciones	Tiempo (ms)	Iteraciones	Tiempo (ms)	Iteraciones
1	312	5	109	1	358	1
2	156	2	156	1	468	1
3	156	2	265	2	920	2
4	499	7	171	1	358	1
5	156	2	218	1	421	1
6	218	4	296	1	421	1
7	514	8	187	1	436	1
8	327	4	390	4	530	1
9	156	2	390	5	592	1
10	390	6	358	3	374	1
Promedio	68,6666	4	127	2	443,4545	1

Tabla 5.1 Datos de detección de los objetivos.

Realizado este experimento, es posible determinar la cantidad adecuada de partículas para la etapa de búsqueda, según los resultados, para un número de 500 partículas se obtuvo un tiempo de detección de aproximadamente 69 ms, pero se obtuvo un promedio de 4 iteraciones para encontrar los objetivos. La ejecución con 1000 partículas arrojó un tiempo 127 ms con un promedio de 2 iteraciones para detectar los objetivos y finalmente para 2000 partículas se obtuvo un tiempo promedio de 443 ms con 1 iteración para detectar los objetivos. Los resultados muestran que a medida que aumenta el número de partículas, el número de iteraciones para encontrar los objetivos disminuye, pues el área que abarcan estas partículas es mayor, pero así mismo el tiempo de ejecución aumenta debido al procesamiento que debe hacer el algoritmo para cada partícula. Teniendo en cuenta el número de iteraciones y el tiempo que toma cada ejecución para realizar la búsqueda, se puede establecer un número de 1000 partículas como una buena opción para esta etapa, ya que asegura una rápida detección de los objetivos, 2 iteraciones, en un tiempo razonable.

5.6.2 Pruebas de seguimiento y precisión.

Se ha empleado una secuencia de imágenes del rostro del usuario para evaluar el comportamiento del filtro de partículas multimodal y el filtro de partículas híbrido en el seguimiento de las pupilas a lo largo de las iteraciones. Para comprobar el funcionamiento de estos algoritmo se ha medido la distancia existente entre las dos pupilas y se ha graficado este valor, teniendo en cuenta que esta distancia es de aproximadamente 430 píxeles y se mantiene constante durante los fotogramas, a continuación se presentan los resultados de la seguimiento en la Figura 5.2. Dada la naturaleza aleatoria del filtro de partículas todos los resultados mostrados en el estudio han sido promediados sobre 10 repeticiones el experimento en las mismas condiciones.

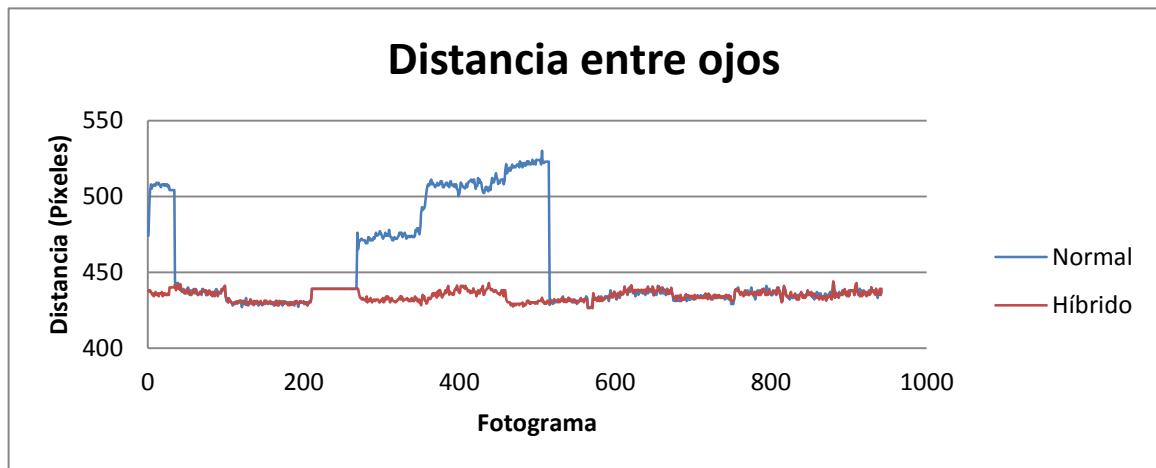


Figura 5.2 Distancia entre las pupilas.

En la gráfica es posible observar la fiabilidad del filtro de partículas híbrido (color rojo) en el seguimiento de los objetivos, durante todas las iteraciones la distancia entre las pupilas se mantiene alrededor de 430 píxeles, indicando que las zonas de alta probabilidad se mantienen exploradas durante la secuencia de imágenes. Caso contrario presenta el filtro de partículas multimodal (color azul), mostrando en algunas secciones del video una pérdida de los objetivos y exploración de zonas erráticas del rostro del usuario lo que aumenta la distancia en píxeles, el filtro converge en estas zonas y se mantiene por algunas iteraciones, hasta que el número partículas desciende a tal grado que se realiza una búsqueda por toda la imagen nuevamente. La ilustración permite concluir que con respecto al seguimiento de los objetivos, el filtro de partículas híbrido presenta una notable ventaja sobre el filtro de partículas multimodal, principalmente en la fiabilidad y confiabilidad del seguimiento, ya que siempre tiene exploradas las zonas reales de interés.

Una vez determinado la fiabilidad del seguimiento, ahora se compara la precisión de los algoritmos implementados, para ello se hace uso de una secuencia de imágenes elaborada, con el fin de conocer con exactitud las posiciones de los objetivo en cada fotograma. La secuencia de imágenes utilizada, presenta dos objetivos de iguales características cromáticas que se mueven simultáneamente a una velocidad constante, donde el punto central entre los dos traza un rectángulo en su recorrido que dura 160 fotogramas. La imagen 5.5 muestra algunos fotogramas de esta secuencia.

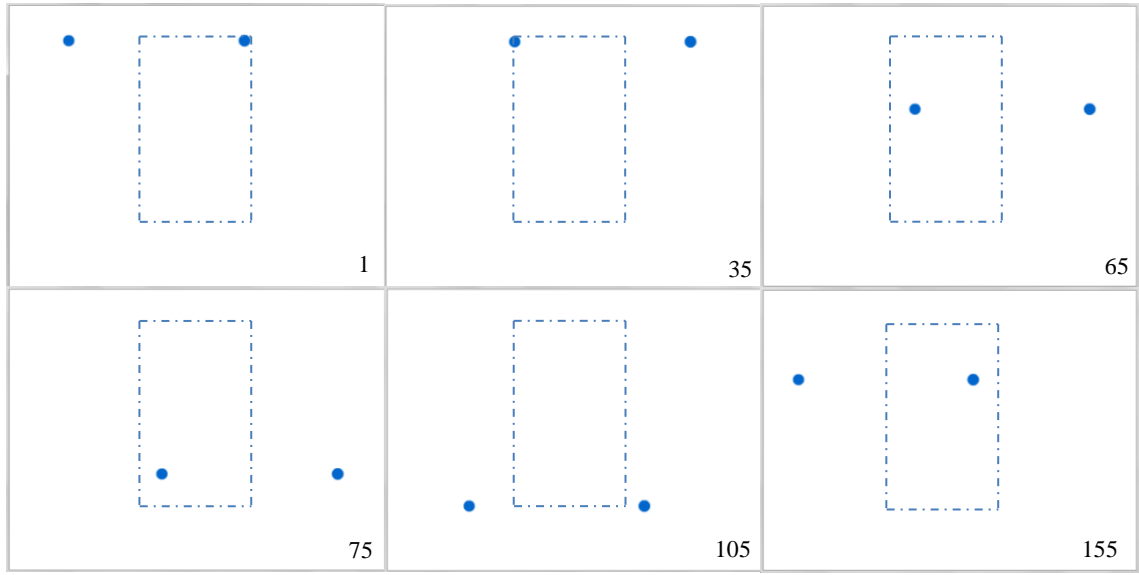


Imagen 5.5 Secuencia de vídeo para medir precisión.

Para medir la precisión de los algoritmos, cada uno de ellos se ejecuta sobre este video y se toman los valores que arroja el punto central entre los objetivos, estos valores se restan con los originales y se obtiene un valor de error con el que es posible determinar la precisión. La figura 5.3 muestra los resultados de la prueba, promediando 10 experimentos.

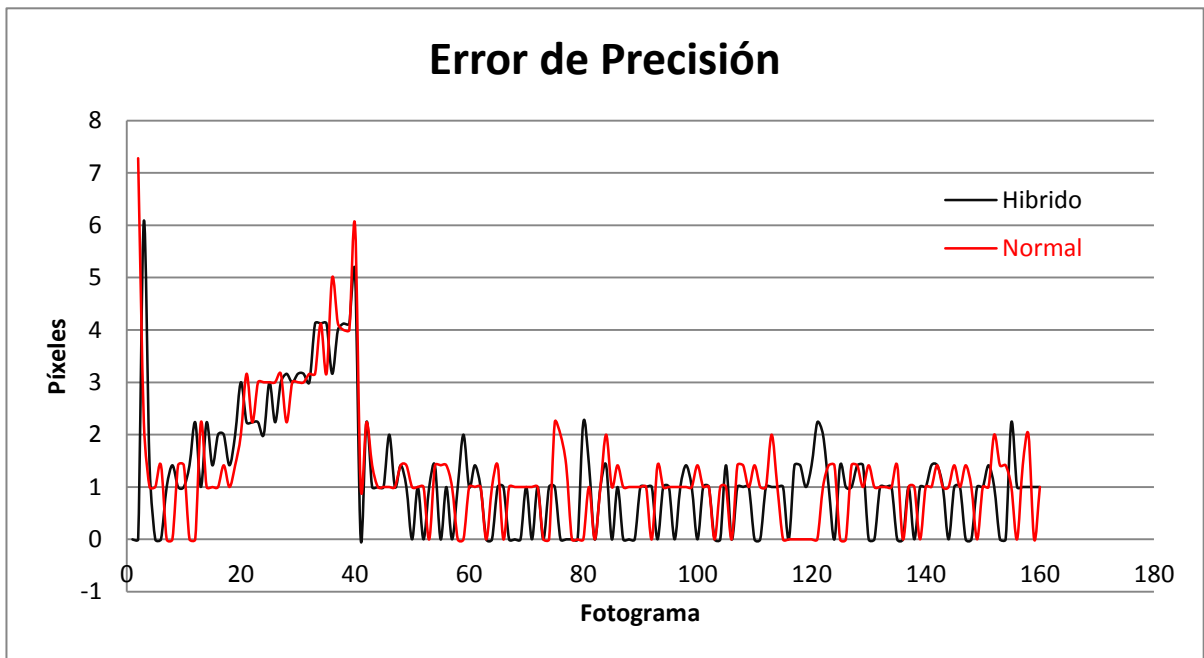


Figura 5.3 Resultados de precisión

La ilustración anterior permite ver la precisión de los algoritmos implementados durante el seguimiento, se distingue una cierta similitud entre ambos filtros, ya que durante las primeras iteraciones se realiza una búsqueda total por el espacio de estados (fotogramas 1 a 5), entre el fotograma 5 y 10 los filtros encuentran los objetivos y los siguen con un error de 1 píxel, pero durante los fotogramas 20 a 40, tiempo en el cual se realiza un cambio de dirección, los filtros pierden un poco de vista los objetivos llegando a un error de 5 píxeles en el caso del filtro híbrido y 6 píxeles el filtro multimodal. Luego de esto el los filtros determinan la velocidad adecuada del movimiento y siguen los objetivos con una buena precisión, oscilando el error entre 0 y 1 píxel durante 120 fotogramas, tiempo en el cual se realizan dos cambios de dirección.

En la siguiente prueba se utiliza la misma secuencia de imágenes que en el experimento anterior, pero se ha aumentado la velocidad en los primeros fotogramas (de 1 al 50), de 2 píxeles por cuadro a 4 píxeles por cuadro, con el fin de observar el comportamiento de los filtros frente a un cambio de velocidad, los resultados de esta prueba se ilustran en la figura 5.4.

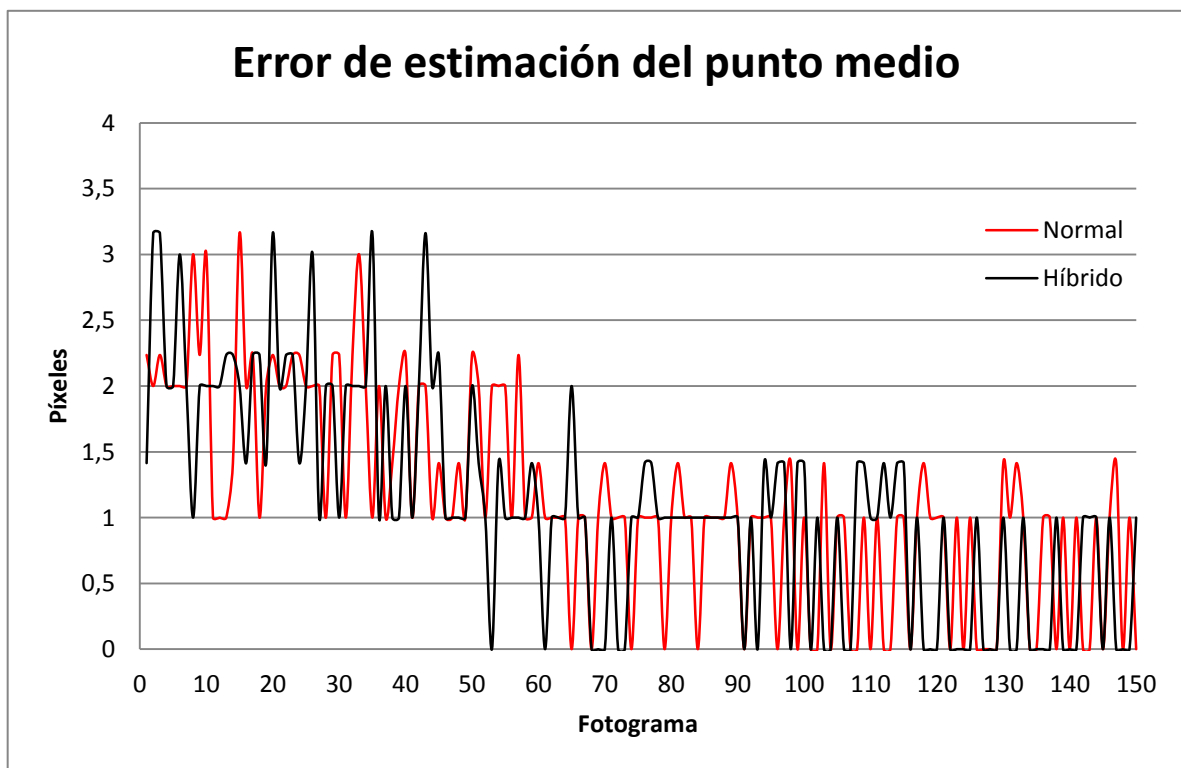


Figura 5.4 Resultados de precisión

De la gráfica anterior es posible observar una respuesta similar entre los filtros, esto se debe a que algorítmicamente el posicionamiento de las partículas es igual, durante las primeras

iteraciones, como se esperaba, el error de precisión oscila entre 1 y 3, debido a la velocidad con que se mueven los objetivos, pero para los siguientes fotogramas el error disminuye.

En el siguiente experimento se comparan el rendimiento los filtros frente a cambios de velocidades bruscos, se utiliza el mismo recorrido de los objetivos de la misma secuencia de imágenes que en el experimento anterior, pero en los cambios de dirección, también se cambia la velocidad, de los fotogramas de 1 al 100, los objetivos se mueven con una velocidad de 2 píxeles/fotograma, del 100 al 200 se mueven a 1 píxel /fotograma, del 200 a 240 a una velocidad de 4 píxeles/fotograma y finalmente del 220 al fotograma 420 a una velocidad de 2 píxeles/fotograma. Los resultados de este experimento se muestran en la figura 5.5.

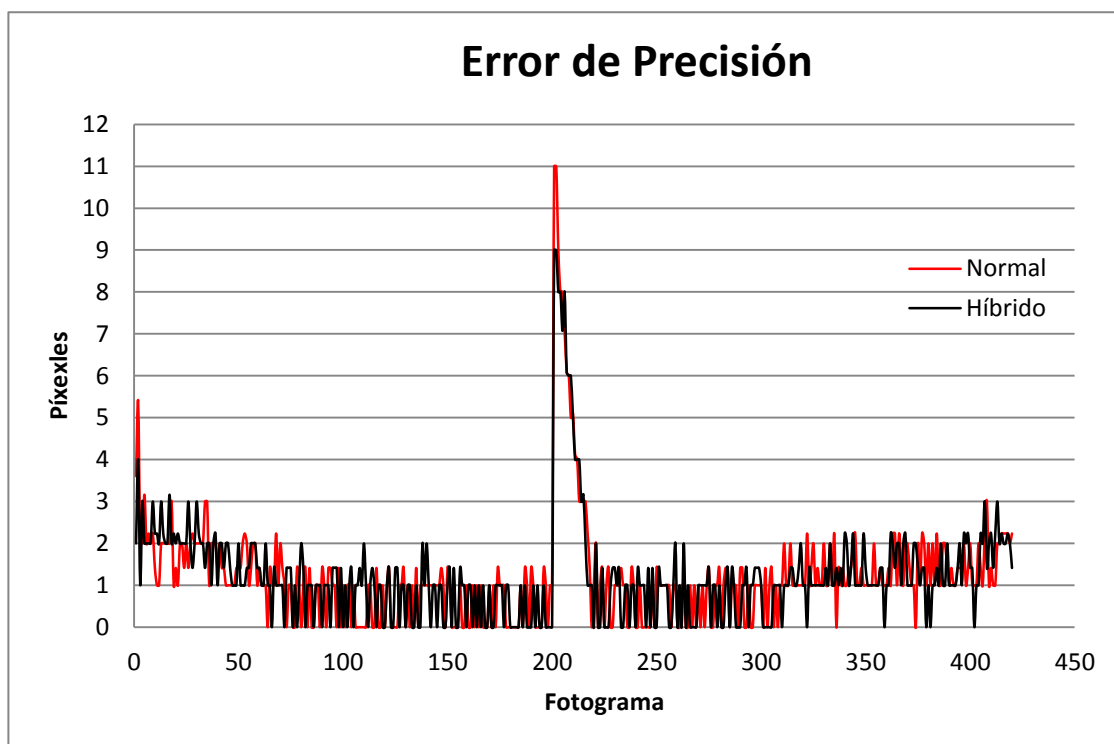


Figura 5.5 Resultados de Los filtros frente a cambios de velocidad.

En la ilustración anterior es posible observar el comportamiento de los filtros implementados frente a cambios de velocidad, ambos filtros presentan un comportamiento similar, y se puede observar que un punto crítico de la prueba se desarrolló entre los fotogramas 200 y 220 donde se aumenta la velocidad bruscamente y por pocos cuadros, en lapso de tiempo el filtro híbrido alcanzó un error máximo de 9 píxeles, mientras que el filtro normal llegó a 11. En los otros fotogramas el error osciló entre 0 y 2 píxeles, un error aceptable, teniendo en cuenta que el tamaño del objeto es 20 píxeles dentro de una imagen de 640 x 480 píxeles.

Un último experimento se realiza para determinar la velocidad límite en la que pueden operar los filtros, para ellos se utiliza una secuencia de imágenes en la que se varía la velocidad de los objetivos, a medida que transcurren los fotogramas. En la tabla 5.2, se presentan las velocidades y los rangos de fotogramas del movimiento de los objetos.

<i>Fotogramas</i>	<i>Velocidad (píxel/fotograma)</i>
<i>1 – 101</i>	<i>1</i>
<i>101 – 201</i>	<i>2</i>
<i>201 – 301</i>	<i>3</i>
<i>301 – 401</i>	<i>4</i>
<i>401 – 501</i>	<i>8</i>
<i>501 – 601</i>	<i>16</i>
<i>601 – 700</i>	<i>32</i>

Tabla 5.2 Rango de fotogramas y velocidades.

En la figura 5.6 es posible observar la respuesta de los filtros frente a cambios de velocidad, durante los fotogramas 1 a 601 donde se presenta una velocidad máxima de 16 píxeles por fotograma los filtros responden de buena forma, obteniendo errores de alrededor de 3 píxeles, pero durante las iteraciones 601 a 700, ninguno de los dos filtros es capaz de realizar un seguimiento preciso, generando errores de hasta 310 píxeles para el algoritmo normal y 240 para el híbrido dada la velocidad de los objetivos de 32 píxeles por fotograma.

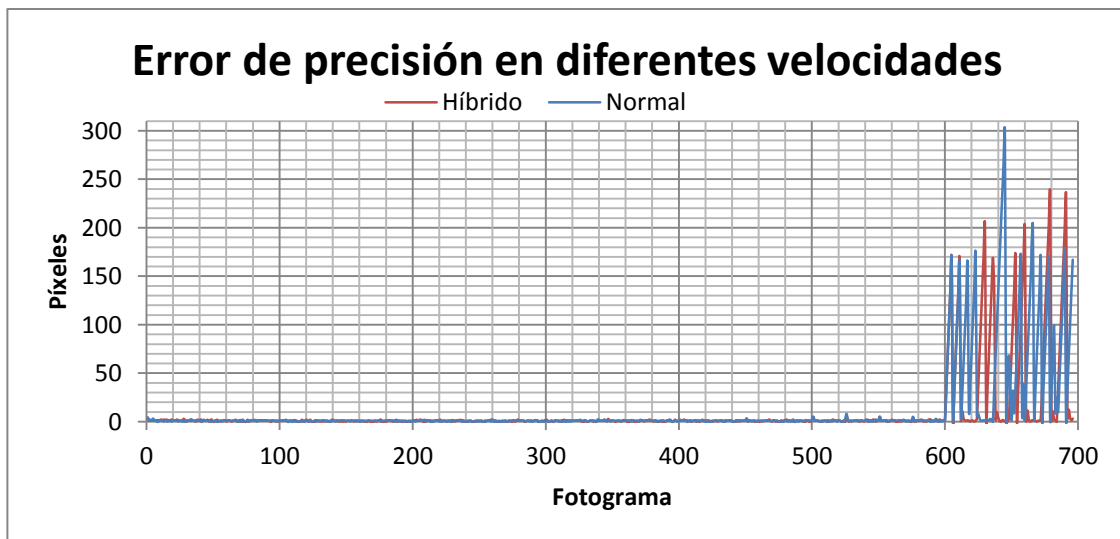


Figura 5.6 Respuesta de los filtros frente a cambios de velocidad.

El experimento anterior permite establecer un límite de velocidad para el seguimiento de los objetivos en la escena, este es de 16 píxeles por fotograma, velocidades superiores aumentan el error de precisión.

5.6.3 Pruebas de posicionamiento.

Aclarar que la estela del cursor es solo para la prueba

Dentro los objetivos establecidos para este proyecto, se estableció el posicionamiento de la mirada en una interfaz de usuario, como se mencionó en el Capítulo 4, la interfaz para realizar las pruebas de posicionamiento es el entorno gráfico de Windows 7.

En un primer experimento se ha establecido como objetivo mover el cursor a través de tres iconos presentes en el escritorio del ordenador, los iconos con un tamaño de 90 x 120 píxeles están ubicados dispersamente sobre la interfaz de un tamaño de 1366 x 768. Para la aplicación desarrollada, siempre el cursor en el instante inicial, es decir en la iteración 1, se ubica en el centro de la pantalla, en la coordenada (683,384). En la figura 5.7 es posible observar el recorrido que realizó el usuario para mover el cursor a través de pantalla, los puntos blancos recrean la velocidad con que se mueve el cursor, una mayor distancia entre estos puntos indica una mayor velocidad de movimiento. La aplicación inicia en el centro de la pantalla y de ahí espera los movimientos del ojo para trazar su trayectoria. Cabe aclarar que la estela de puntos blancos en la figura que representa la trayectoria generada por el cursor, solo se utiliza para realizar la prueba y analizar los resultados, mas no aparece en el funcionamiento de la aplicación. El experimento se realiza en condiciones normales de iluminación, con un solo bombillo ahorrador de 15 W en el sitio.

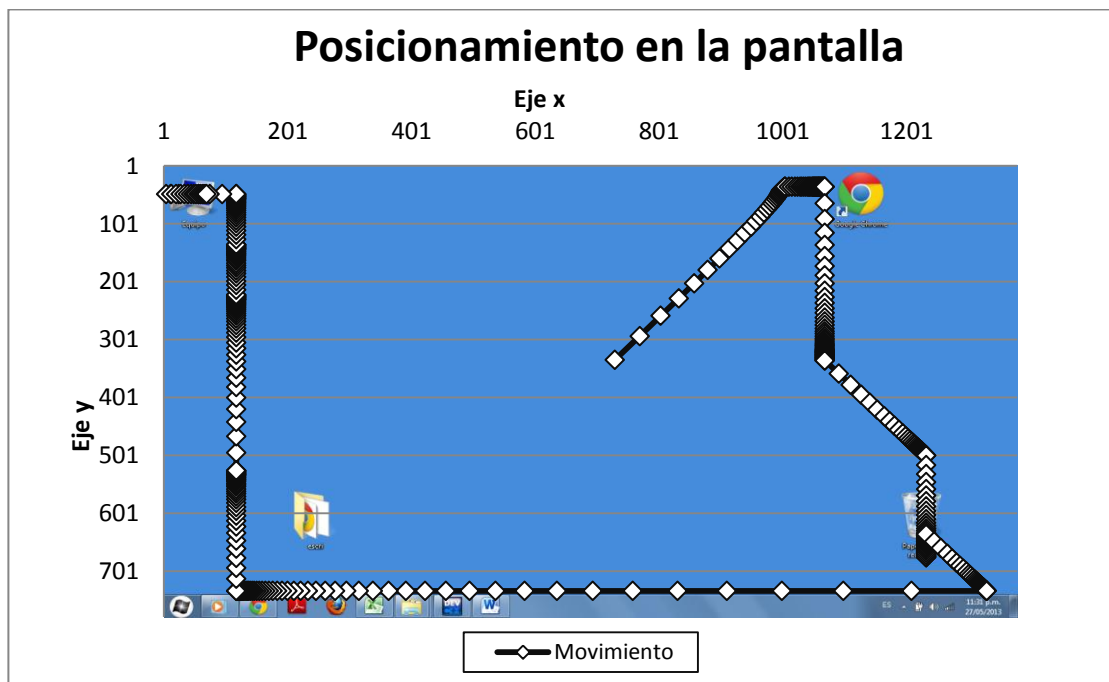


Figura 5.7 Posicionamiento del cursor en la pantalla.

En la ilustración anterior se puede observar el recorrido que el usuario realizó para ubicar el cursor sobre los iconos, los movimientos son en línea recta y la velocidad se determina de acuerdo a la distancia a recorrer, como se estableció en el sistema de coordenadas incrementales. El cursor se posicionó de manera adecuada en dos de los iconos mientras que un tercero no se alcanzó el objetivo.

El siguiente experimento realizado para determinar la eficiencia del sistema, se desarrolla en el mismo entorno gráfico, esta vez se la prueba consiste en posicionar el cursor en el *Menú de Inicio* y abrir la herramienta *Calculadora* que se encuentra en ese menú. El posicionamiento requiere un poco más de precisión que en el experimento anterior y se desarrolla bajo las mismas condiciones de iluminación. En la figura 5.8 es posible observar el resultado del movimiento del cursor a lo largo de la ejecución, inicialmente el cursor se ubica en el centro de la pantalla y posteriormente inicia su recorrido.

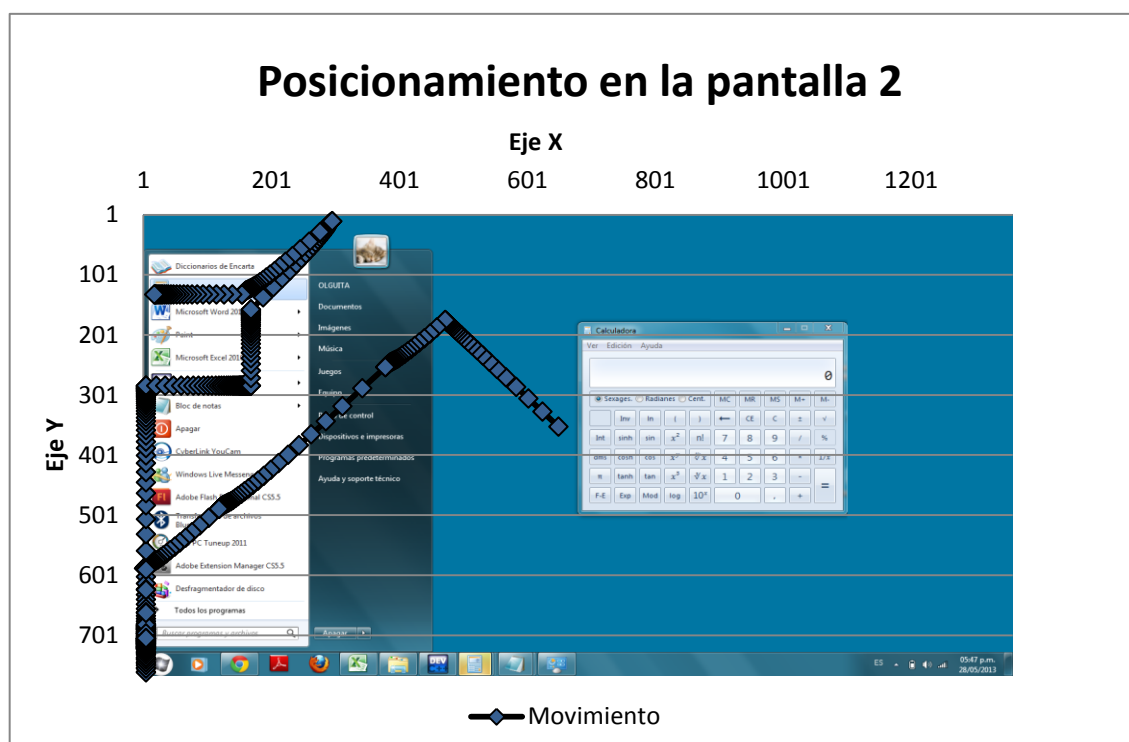


Figura 5.8 Posicionamiento del cursor en la pantalla, prueba 2.

La ilustración anterior deja en evidencia la dificultad con que se realiza el desplazamiento en la interfaz, luego de 620 iteraciones se logró posicionar el mouse dentro del objetivo para lograr abrir la herramienta de la calculadora. Los movimientos son algo torpes teniendo en cuenta que la mínima distancia que se recorre es de 40 píxeles, esto quiere decir que por un movimiento de 1 píxel en la imagen de la pupila, el cursor se mueve 40, dificultando así posicionar el mouse en la zona de interés.

Capítulo 6.

CONCLUSIONES Y TRABAJOS FUTUROS

Como se ha podido observar en el apartado anterior, el filtro de partículas híbrido implementado presentó muy buenos resultados en la detección y seguimiento dinámico de la pupila, sin embargo, el posicionamiento de la mirada en la pantalla presentó algunos retos que denotan una no muy buena eficiencia. En este capítulo se resumen los logros alcanzados durante el desarrollo de este trabajo y los resultados obtenidos, acompañados de alternativas y líneas de investigación que podrían mejorar dichos resultados.

El uso del sistema sensorial cámara IR y emisor IR, presenta grandes ventajas para el desarrollo de la aplicación, en primera medida, por su fácil consecución, bajo costo y fácil instalación, factores clave si se desea disponer el producto de esta investigación a cualquier persona o ampliar los estudios en la Universidad. Otra ventaja importante radica en el uso de la luz infrarroja para la adquisición de las imágenes, debido a que la fuente de luz IR ilumina completamente el rostro del usuario opacando otras fuentes de este tipo de luz como la artificial generada por los bombillos o la natural generada por el sol, permitiendo el buen funcionamiento de la aplicación frente a variaciones de luminosidad, incluso en la oscuridad total en un entorno cerrado.

La implementación del filtro de partículas como método para detectar y seguir la pupila de los ojos arrojó muy buenos resultados, esta técnica de filtrado bayesiano demostró su funcionalidad en aplicaciones de visión artificial como la desarrollada en esta investigación. A pesar de adquirir las imágenes y crear las partículas en escala de grises, espacio de color del cuál no es posible obtener mucha información comparada con otros, se reduce ampliamente el procesamiento de las mismas y el tiempo de ejecución, ya que solo es necesario crear un canal de color y por lo tanto un solo histograma, contrariando a muchas aplicaciones que utilizan los canales RGB o HSV para el procesamiento.

Uno de los principales problemas del filtro de partículas convencional radica en el hecho de no poder sostener a lo largo del tiempo más de una hipótesis en la estimación de la función de densidad, lo que limita el seguimiento fiable a solo un objetivo. En este trabajo se optó por implementar un algoritmo de agrupamiento, el K-Medias, que permite mantener más de una hipótesis durante la ejecución, brindando así la posibilidad de seguir las pupilas de los ojos, las pruebas realizadas sobre este algoritmo arrojaron buenos resultados a pesar de su sencillez. De cara a una mejora futura, se podría implementar otro algoritmo de *clustering* que brinde mejores prestaciones y no se limite solo a clasificar dos objetivos.

Al implementar la hibridación entre el filtro de partículas y el algoritmo inteligente, es posible evidenciar en los experimentos una notable mejora en la fiabilidad del seguimiento múltiple del filtro, ya que el éxito de la hibridación radica en el aumento de la calidad de las soluciones que permite la convergencia del filtro en las zonas de interés. Entre el filtro multimodal y el filtro híbrido no se presentaron grandes diferencias en cuanto al error de estimación del estado, básicamente porque ambos presentan las mismas características de comparación, difusión y movimiento de las partículas, sin embargo, la gran diferencia entre ellos se presenta en la baja fiabilidad de las partículas del filtro multimodal, estas partículas tienden a ubicarse sobre zonas erróneas que presentan componentes de color similar al modelo del objetivo, como por ejemplo las cejas o las pestañas. Una alternativa para mejorar la respuesta de filtro podría ser la implementación de algún otro algoritmo de optimización diferente a la búsqueda dispersa, se puede hacer uso de cualquier metaheurística poblacional o trayectorial, algoritmos meméticos o genéticos, existe una gran variedad de técnicas de mejoramiento que se pueden adaptar al filtro.

El posicionamiento de la mirada en la pantalla no presentó buenos resultados como se evidenció en las pruebas, esto se debe principalmente a dos aspectos, en primer lugar, el hecho de realizar el seguimiento de las pupilas en una sola imagen disminuye el tamaño de las mismas, en muchas aplicaciones de seguimiento visual, solo se toma un ojo para realizar el posicionamiento, este ojo en una imagen de 640 x 480 píxeles representa un 10% del total de píxeles, mientras que en esta investigación, los objetos a seguir suman un total de 0.2% de la imagen, esto limita los movimientos de las pupilas a pocos píxeles con los cuales se debe posicionar el mouse. El segundo aspecto por el que se ve reducida la eficiencia en el posicionamiento, tiene que ver con el ruido, como se evidenció en las pruebas de precisión, la estimación del estado oscila alrededor de 3 píxeles con respecto a la posición real y teniendo en cuenta que para recorrer toda la pantalla las pupilas se desplazan entre 7 y 15 píxeles, el ruido es un gran inconveniente. Sin embargo, realizando una calibración correcta, es posible desplazarse por la pantalla hacia puntos de interés y dar clic, tomando movimientos por encima del ruido y determinado la dirección de esos movimientos.

Dado que la aplicación generada en este trabajo se desarrolló e implementó bajo una metodología incremental, que la hace portable y escalable, y el filtro de partículas es una sección independiente del algoritmo inteligente, es posible realizar mejoras sobre la

aplicación fácilmente o puede ser tomada como base para el desarrollo de futuras aplicaciones.

Queda abierta la posibilidad de desarrollar trabajos futuros en este campo de investigación, cómo por ejemplo el uso de visión estéreo para un reconocimiento en 3D, añadir más gestos a la aplicación como lo hace un ratón convencional; dar clic derecho, clic sostenido, etc. El seguimiento ocular presenta también muchos campos de aplicación, se podría modificar esta implementación para manipular dispositivos como sillas, interruptores o una infinidad de artefactos.

El desarrollo de este proyecto fue un éxito, ya que se lograron cumplir con los objetivos planteados, seguir dinámicamente la pupila y posicionar el mouse en la pantalla con los movimientos del ojo, finalizando satisfactoriamente la elaboración de la aplicación.

BIBLIOGRAFÍA

- [1] N.J. Gordon, D.J. Salmond, A.F.M. Smith, “*Novel Approach to Nonlinear / Non-Gaussian Bayesian State Estimation*”, IEEE Proceedings-F, Vol. 140, No 2, Abril de 1993.
- [2] Y. Salih, A. S. Malik, “*Comparison of stochastic filtering methods for 3D tracking*”, Pattern Recognition 44 (2011) 2711–2737, Marzo 2011. [En línea]. Disponible en: Elsevier, <http://www.elsevier.com/locate/pr>. [Último acceso: Abril de 2013].
- [3] A. Herrera Rey, W. S. Herrera Rey, R. Herrera Rey, “*Target Tracking Using Particle Filters*”, Revista Universitaria en Telecomunicaciones, Informática y Control. vol. 1, no. 1, ISSN 2227-5894, Diciembre de 2011. [En línea]. Disponible en: <http://ciclope.unicauca.edu.co/rutic/index.php/rutic/article/view/13>. [Último acceso: Abril de 2013].
- [4] M. Bolic, “*Architectures for Efficient Implementation of Particle Filter*”, Stony Brook University, Agosto de 2004. [En línea]. Disponible en: http://www.site.uottawa.ca/~mbolic/Miodrag_Bolic_files/published/bolic_thesis.pdf. [Último acceso: Abril de 2013].
- [5] X. Zhang, W. Hu, S. Maybank, “*A Smarter Particle Filter*”, National Laboratory of Pattern Recognition, 2009. [En línea]. Disponible en: [nlpr.ia.ac.cn](http://www.nlpr.ia.ac.cn), <http://www.nlpr.ia.ac.cn/2009papers/gjhy/gh52.pdf>. [Último acceso: Marzo de 2013].
- [6] J. J. Pantrigo, A. Duarte, A. Sánchez, “*Hibridación entre Filtro de Partículas y Búsqueda Dispersa: aplicación al TSP Dinámico*”, Universidad Rey Juan Carlos, 2005. [En línea]. Disponible en: gavab.escet.urjc.es, http://gavab.escet.urjc.es/articulos/2005_25.pdf. [Último acceso: Abril de 2013].
- [7] J.J. Pantrigo, “*Resolución de Problemas de Optimización Dinámica mediante la Hibridación entre Filtros de Partículas y Metaheurísticas Poblacionales*”, Universidad Rey Juan Carlos, 2005. [En línea]. Disponible en: [escet.urjc.es](http://www.escet.urjc.es), <http://www.escet.urjc.es/~jjpantrigo/papers/mpfCMPI06.pdf>. [Último acceso: Marzo de 2013].
- [8] J. Hernández, “*Seguimiento Visual con Filtro de Partículas Híbridos con Algoritmos Meméticos*”, Universidad Rey Juan Carlos, 2010. [En línea]. Disponible en:

eciencia.urjc.es,<http://eciencia.urjc.es/bitstream/10115/4130/1/memoria%20Javier%20Hern%C3%A1ndez.pdf>. [Último acceso: Abril de 2013].

- [9] A. Rodríguez Moya, “*Estudio del Filtro de Partículas Aplicado al Seguimiento de Objetos en Secuencias de Imágenes*”, Universidad Carlos III de Madrid, Agosto de 2009. [En Línea]. Disponible en: e-archivo.uc3m.es, <http://e-archivo.uc3m.es/bitstream/10016/11173/1/PFC%20Alvaro%20Rodriguez.pdf>. [Último acceso: Febrero de 2013].
- [10] E. Burgueño, “*Aplicación del filtro de partículas al seguimiento de objetos en secuencias de imágenes*”, proyecto de fin de carrera, Universidad Rey Juan Carlos, 2003.
- [11] D. de Miguel Benito, “*Detección Automática del Color de la Piel en Imágenes Bidimensionales Basado en el Análisis de Regiones*”, Universidad Rey Juan Carlos, 2005. [En línea]. Disponible en etsii.urjc.es, <http://www.etsii.urjc.es/~jjpantrigo/PFCs/MemoriaPielFeb05.pdf>. [Último acceso: Febrero de 2013].
- [12] R. Llorca Queralt, “*Detección de Tumultos con Filtros de Partículas*”, Universidad Politécnica de Cataluña – Escuela Técnica Superior de Ingeniería de Telecomunicaciones de Barcelona, 2008. [En línea]. Disponible en <http://www.cristiancantan.org/data/pubs/theses/RamonLlorca.pdf>. [Último acceso: Diciembre de 2012].
- [13] A. Marcos Ramiro, “*Seguimiento de Múltiples Objetos en Espacios Inteligentes*”, Universidad de Alcalá, Julio de 2009.
- [14] P. Barrera, J.M. Cañas, C. Agüero, “*Seguimiento visual 3D de múltiples objetos combinando filtros de partículas*”, Universidad Rey Juan Carlos, Septiembre de 2008. [En línea]. Disponible en: [jopha.net](http://www.jopha.net), <http://www.jopha.net/waf/index.php/waf/waf08/paper/viewFile/22/6>. [Último acceso: Enero de 2013].
- [15] S. Kawato, N. Tetsutani, “*Detection and tracking of eyes for gaze-camera control*”, Image and Vision computing, no. 12, pp.1031–1038, Octubre de 2004. [En línea]. Disponible en: Elsevier, <http://www.sciencedirect.com>. [Último acceso: Diciembre de 2012].
- [16] C. S. Lin, C. W. Ho, C. N. Chan, C. R. Chau, Y. C. Wu, M. S. Yeh, “*An eye-tracking and head-control system using movement increment-coordinate method*”, Optics & Laser Technology no. 39 (2007), pp. 1218–1225, Septiembre de 2006. [En línea]. Disponible en: Elsevier, <http://www.sciencedirect.com>. [Último acceso: Noviembre de 2012].

- [17] Z. Zhua, Q. Ji, “*Robust real-time eye detection and tracking under variable lighting conditions and various face orientations*”, *Computer Vision and Image Understanding* 98 (2005) 124–154, Julio de 2004, [En línea]. Disponible en: Elsevier, <http://www.elsevier.com/locate/cviu>. [Último acceso: Noviembre de 2012].
- [18] A. De Santis, D. Iacoviello, “*Robust real time eye tracking for computer interface for disabled people*”, *Computer Methods and Programs in Biomedicine*, Universidad de Roma, Marzo de 2009. [En línea]. Disponible en: Elsevier, www.intl.elsevierhealth.com/journals/cmpb. [Último acceso: Febrero de 2013].
- [19] Y. Li, S. Wang, X. Ding, “*Eye/eyes tracking based on a unified deformable template and particle filtering*”, *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1377–1387, Agosto de 2010. [En línea]. Disponible en: Elsevier, <http://www.sciencedirect.com>. [Último acceso: Noviembre de 2012].
- [20] F. A. Justo, “*Creación de una Herramienta que Permita Mover el Cursor de una Computadora a Partir del Movimiento Ocular, Utilizando Técnicas de Visión Artificial*”, Universidad de los Andes – Mérida, Venezuela, Abril de 2009.
- [21] C. Puell Marín, “*Óptica Fisiológica: El sistema óptico del ojo y la visión binocular*”, Universidad Complutense de Madrid, ISBN Versión Digital: 1-4135-6363-5, 2006.
- [22] T. Winslow, “*Medical and Scientific Illustration*”, 2001. [En línea]. Disponible en: <http://www.teresewinslow.com/port.asp?keywords=no+match>.
- [23] E. Turégano Pedruelo, “*Eye Board: Un Periférico Alternativo Visual*”, 2006. [En línea]. Disponible en: <http://robolab.unex.es/index.php/downloads/Publications/Master-Thesis/2006.-EyeBoard-Un-Perif%C3%A9rico-Alternativo-Visual/>.
- [24] E. L. González, S. Luvoni, G. Irigoín, P. D. Agüero, J. C. Tulli, “*Electrooculografía, Nuevas Perspectivas Para Discapacitados Motrices*”, Universidad Nacional de Mar del Plata, 2010. [En línea]. Disponible en: <http://www.ufrgs.br/niece/eventos/CIIEE/2007/pdf/CE-%20130%20%20Argentina.pdf> [Último acceso: Marzo de 2013].
- [25] Qubit Systems, [En línea]. Disponible en: <http://qubitsystems.com/human/s225-eog-sensor/>. [Último acceso: Abril de 2013].
- [26] F. Meunire, “*On the Automatic Implementation of Eye Involuntary Reflexes Measurements Involved in the Detection of Human Liveness and Impaired Faculties*”. [En línea]. ISBN: 978-953-307-026-1, InTech, DOI: 10.5772/7054. Disponible en: <http://www.intechopen.com/books/image-processing/on-the-automatic-implementation-of-the-eye-involuntary-reflexes-measurements-involved-in-the-detecti>.

- [27] Z. Chen, “*Bayesian Filtering: From Kalman Filters To Particle Filters, and Beyond*”, Communications Research Laboratory, McMaster University, Ontario Canada, 2003. [En línea]. Disponible en http://www.cse.buffalo.edu/~peter/refs/DataAssimilation/GenRefsTutorials/Chen_2003.pdf.
- [28] J.S. Liu, R Chen, “*Sequential Monte Carlo Methods for Dynamics Systems*”, Journal of the American Statistical Association, Vol. 93, No. 433, pp. 1032 – 1044, Septiembre de 1998. [En línea]. Disponible en: http://www.people.fas.harvard.edu/~junliu/TechRept/98folder/liu&chen98_2.pdf. [Último acceso: Octubre de 2012].
- [29] M. Isard, A. Blake, “*Condensation- Conditional Density Propagation for Visual Tracking*”, Universidad de Oxford. International Journal of Computer Vision 29(1), 5-28, 1998. [En línea]. Disponible en: <https://www.cs.duke.edu/courses/cps296.1/spring05/handouts/Isard%20Blake%201998.pdf>. [Último acceso: Enero de 2013].
- [30] S. Dubuisson, “*The computation of the Bhattacharyya distance between histograms without histograms*”, Laboratorio de informática de Paris, Universidad Pierre et Marie Curie. Julio de 2010. [En línea]. Disponible en: <http://www-pequan.lip6.fr/~dubuisso/articles/IPTA10b.pdf>. [Último acceso: septiembre de 2012].
- [31] C. S. Lin, L. W. Lue, M. S. Yeh, T. S. Hwang, S. H. Lee, “*A new image processing method for evaluating the pupillary responses in a HMD-type eye-tracking device*”, Optics & Laser Technology, vol. 35, no. 7, pp. 505–515, Octubre de 2003. Disponible en: <http://www.sciencedirect.com>. [Último acceso: Noviembre de 2012].
- [32] C. S. Lin, K. C. Chang, Y. J. Jain, “*A new data processing and calibration method for an eye-tracking device pronunciation system*”, Optics & Laser Technology, vol. 34, no.5, pp. 405–413, Julio de 2002. Disponible en: Elsevier, <http://www.sciencedirect.com>. [Último acceso: Enero de 2013].
- [33] Hough, P. V. C. “*Method and means for recognizing complex patterns*”. U. S. Patent 3, 069 654, diciembre 18, 1962.
- [34] Canny, J., “*A Computational Approach to Edge Detection*”, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.
- [35] G. Bradski, A. Kaebler, “*Learning OpenCV, Computer Vision with the OpenCV Library*”, 2008. ISBN: 978-0-596-51613-0. [En línea]. Disponible en <http://www.cse.iitk.ac.in/users/vision/dipakmj/papers/OREilly%20Learning%20OpenCV.pdf>.

- [36] M. Ruiz Muñoz, “Filtro de Partículas con Búsqueda Dispersa para el Desarrollo de Interfaces de Usuario Avanzadas”, Universidad Rey Juan Carlos, Junio de 2005. [En línea]. Disponible en: http://www.etsii.urjc.es/~jjpantrigo/PFCs/PRPF_HCI.pdf. [Último acceso: Abril de 2013].