

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

**MÁSTER UNIVERSITARIO EN AUTOMÁTICA E
INFORMÁTICA INDUSTRIAL**



**UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA**

**DISEÑO, DESARROLLO Y EVALUACIÓN DE UN
SISTEMA DE VISIÓN PARA AUTOMATIZAR
EXPERIMENTOS HEALTHSPAN**

TRABAJO FIN DE MÁSTER

AUTOR:

Pablo Emmanuel Layana Castro

TUTOR:

Antonio José Sánchez Salmerón

Valencia 17 de septiembre de 2018

AGRADECIMIENTOS

A Dios por guiarme e iluminar en cada paso, por toda las oportunidades y bendiciones que pone en mi camino cada día.

¡Gracias!

A mi familia y seres queridos por su apoyo incondicional en cada paso de crecimiento personal y profesional, gracias por estar siempre presentes y haberme brindado todo su amor y comprensión en los momentos de inmensa alegría, así como también en aquellos momentos de preocupación y duras decisiones.

¡Los quiero mucho!

A mis amigos quienes se encuentran lejos, pero sus palabras, deseos y cariño tan cerca que siempre los llevo conmigo, y han sido un inmenso apoyo durante esta etapa de mi vida.

¡Gracias a todos!

A mis profesores, por los conocimientos y preparación que me brindaron durante toda mi formación del máster.

¡Gracias!

A mi tutor y mentor Antonio José Sánchez Salmerón, por la oportunidad de participar en este gran proyecto, el cual me ha permitido conocer y aprender más acerca de lo que es visión artificial, gracias por haber confiado en mí durante el desarrollo del proyecto, así como también por sus sabios consejos y conocimientos transmitidos durante este período.

¡Gracias!

RESUMEN

Cuando trabajamos con sistemas de control suponemos que estos son lineales, continuos, diferenciables, e invariantes en el tiempo, y que ante un estímulo de entrada responden de una manera determinada, caso contrario tratamos de que el sistema se comporte de acuerdo con estas suposiciones, esto es necesario para poder construir un modelo aproximado del sistema y poder desarrollar un controlador que cumpla con las especificaciones que requerimos.

Se entiende que ningún modelo es perfecto pero este debe ser funcional, debe ayudar a tener más conocimiento del control de proceso, mediante la facilidad el cálculo y entendimiento del mismo y las ecuaciones físicas que lo gobiernan. Cuando se construye un modelo no se debe tomar los resultados obtenidos como finales, se debe entender que un modelo puede evolucionar cuando haya datos adicionales que permitan conocer mejor el sistema, por consiguiente, no se debe descartar ningún dato por insignificante que parezca.

El control actual se relaciona directamente con diferentes partes, como la informática de la cual depende el uso de los diferentes softwares y hardwares para el análisis y procesamiento de la información; las matemáticas de la cual depende la respuesta del sistema, diferentes problemas físicos y complejos son resueltos haciendo uso del modelado de proceso, dichos modelos se pueden manejar mediante diferentes operaciones matemáticas. El control también depende de las comunicaciones las cuales existen entre el sistema de control, actuadores, sensores y el proceso. Por último, para realizar un control es necesario conocer el sistema, las ecuaciones físicas y químicas que gobiernan la funcionalidad del sistema, esto conlleva a contemplar el hardware y software para el diseño y desarrollo de la ingeniería necesaria realizar el control que se requiere para resolver un problema en específico.

El trabajo a realizar pretende desarrollar una aplicación automática de visión, la cual consiste en capturar imágenes en tiempo real de placas de Petri, realizar un pre-procesamiento de la misma usando un raspberry pi, dicha imagen se enviará por comunicación usando sockets de comunicación a un PC para su posterior procesamiento, localización y de detección de nematodos *C. elegans* con la finalidad de analizar su comportamiento ante diferentes grupos alimenticios.

Durante la fase de diseño de la aplicación se analizará y verificará del experimento con diferentes tipos de luz. La verificación se realizará con un operador de laboratorio, a fin de contrastar la información obtenida del proceso automático de visión y lo observado en microscopio.

Palabras claves: Visión artificial, sistema de visión, Java, Javafx, OpenCv, Raspberrypi, Raspicam, Artificial vision, Vision system.

ÍNDICE

RESUMEN	3
ÍNDICE DE CONTENIDO	5
ÍNDICE DE FIGURAS.....	5
ÍNDICE DE TABLAS.....	7
ÍNDICE DE ECUACIONES.....	7
1. INTRODUCCIÓN	8
1.1. DESCRIPCIÓN DEL PROBLEMA	8
1.2. SOLUCIÓN PROPUESTA	8
1.3. OBJETIVOS	9
1.3.1. Objetivos generales.....	9
1.3.2. Objetivos específicos	9
1.4. RESULTADOS ESPERADOS	9
2. ESTADO DEL ARTE	10
2.1. ANTECEDENTES.....	10
2.2. HEALTHSPAN.....	10
2.3. FILTRO DE PARTÍCULAS	11
2.4. C. ELEGANS.....	12
2.5. OPENCV.....	12
2.6. JAVA FX	13
2.7. MYSQL.....	13
3. DESCRIPCIÓN DEL PROCESO	15
3.1. DESCRIPCIÓN DEL PROCESO REALIZADO	15
3.2. ENCONTRAR ÁREA DE TRABAJO	15
3.3. ENCONTRAR TRAYECTORIAS SIN RUIDO.....	16
3.4. CONTAR Y GUARDAR TOTAL DE TRAYECTORIAS.....	21
3.5. ANALIZAR NÚMERO DE C. ELEGANS EN CADA TRAYECTORIA Y SU INICIO DE RECORRIDO.....	21
4. DESCRIPCIÓN DE LOS MODELOS.....	22
4.1. DETECCIÓN DE ELEMENTO EN EL SENSOR	22
4.2. MODELO 1: POSICIÓN VELOCIDAD.....	22
4.3. MODELO 2: POSICIÓN VELOCIDAD CABEZA – COLA	24
4.4. MODELO 3: POSICIÓN VELOCIDAD Y DESPLAZAMIENTO SENOIDAL.....	25
4.5. MODELO 4: POSICIÓN VELOCIDAD Y DESPLAZAMIENTO SENOIDAL RESTANDO TRAYECTORIA	29
5. DESCRIPCIÓN DEL PROCESAMIENTO	30
5.1. CREACIÓN DE PARTÍCULAS	30
5.2. EVALUACIÓN DE PARTÍCULAS Y PREDICCIÓN	31
5.3. REVISIÓN DE PREDICCIÓN	32
5.4. CÁLCULO DE MOVIMIENTO Y VALOR DE DISPERSIÓN	33

6. RESULTADOS	34
6.1. CÁLCULO DEL ERROR PROMEDIO.....	34
6.2. EVALUACIÓN Y RESULTADOS CON DIFERENTES VALORES DE SIGMA.....	34
6.3. EVALUACIÓN DE MODELOS Y COMPARACIÓN DE ERRORES EN TRAYECTORIAS INDIVIDUALES CON SIGMA IGUAL A 1	36
6.4. EVALUACIÓN DE MODELOS Y COMPARACIÓN DE ERRORES EN TRAYECTORIAS INTERSECANTES SEPARADAS.....	37
6.5. EVALUACIÓN DE MODELOS Y COMPARACIÓN DE ERRORES EN TRAYECTORIAS INTERSECANTES UNIDAS.....	38
7. DESARROLLO DE LA INTERFAZ GRÁFICA	40
7.1. DESCRIPCIÓN DE LA PANTALLA DE INICIO.....	41
7.2. DESCRIPCIÓN DE LA PANTALLA: NUEVA PRUEBA	41
7.2.1. Descripción de la pantalla: Seleccionar directorio de imágenes.	42
7.2.2. Descripción de la pantalla: Procesamiento de imágenes	44
7.3. DESCRIPCIÓN DE LA PANTALLA: PRUEBA HEALTHSPAN.....	44
7.4. DESCRIPCIÓN DE LA PANTALLA: ESTADÍSTICA DE PRUEBAS.....	45
8. MODELOS DE REPORTES GENERADOS	47
8.1. MODELO DE REPORTE PARA PRUEBAS INDIVIDUALES	47
8.2. MODELO DE REPORTE DE CONJUNTO DE PRUEBAS IGUALES	48
8.3. MODELO DE REPORTE ESTADÍSTICO PARA EVALUACIÓN DE DIFERENTES PRUEBAS.....	49
9. CONCLUSIONES Y TRABAJOS FUTUROS	50
9.1. CONCLUSIONES	50
9.2. TRABAJOS FUTUROS	50
10. BIBLIOGRAFÍA.....	51

Índice de contenido

Índice de figuras

Figura 1 Solución propuesta.....	8
Figura 2 Efecto de la medida médica y de salud pública antes y después del siglo XX	11
Figura 3 C. elegans anatomía y ciclo de vida	12
Figura 4 JavaFX es la navaja suiza de herramientas GUI de java.....	13
Figura 5 Modelo de desarrollo de base de datos	14
Figura 6 Diagrama del proceso.....	15
Figura 7 Localización del área de trabajo	16
Figura 8 Imagen en instante K.....	16
Figura 9 Imagen en instante K+1	17
Figura 10 Diferencia de la imagen actual y la imagen siguiente.....	17

Figura 11 Segmentación de áreas claras y oscuras	17
Figura 12 Sumatoria de áreas claras durante 60 imágenes en 2 placas diferentes	18
Figura 13 Sumatoria de áreas oscuras durante 60 imágenes en 2 placas diferentes	18
Figura 14 Sumatoria de segmentación cuerpo de C. elegán durante 60 imágenes en 2 placas diferentes.....	18
Figura 15 Operación AND sobre las 3 imágenes (I1 & I2 & I3)	19
Figura 16 Dilatación de 20 sobre resultado de operación AND	19
Figura 17 Operación OR sobre las 3 imágenes (I1 OR I2 OR I3).....	20
Figura 18 Operación AND sobre resultado de OR y dilatación	20
Figura 19 Operación Close sobre resultado de operación AND	20
Figura 20 Trayectorias separadas e identificadas	21
Figura 21 Trayectoria 5 diferencia de imágenes 25,26.....	21
Figura 22 Ejemplo de partes de la segmentación separadas	22
Figura 23 Estado inicial del modelo 1	23
Figura 24 Modelo 1 posición y velocidad	23
Figura 25 Evaluación de cada partícula para obtener matriz de estados del modelo1 ...	23
Figura 26 Predicción (estados) modelo 1	23
Figura 27 Matriz de estados (partículas evaluadas) del modelo 1.....	24
Figura 28 Estado inicial del modelo 2	24
Figura 29 Modelo 2 posición y velocidad cabeza - cola.....	24
Figura 30 Evaluación de cada partícula para obtener matriz de estados del modelo2 ...	25
Figura 31 Predicción (estados) modelo 2.....	25
Figura 32 Estado inicial modelo 3	26
Figura 33 Modelo 3 aproximación senoidal.....	26
Figura 34 Desplazamiento realizado por el C. elegán	26
Figura 35 Matriz de posición angular	27
Figura 36 Matriz de rotación alfa	27
Figura 37 Matriz de posición angular rotada.....	27
Figura 38 Matriz de partículas del modelo 3.....	27
Figura 39 Evaluación de cada partícula para obtener matriz de estados del modelo3 ...	28
Figura 40 Predicción (estados) modelo 3.....	28
Figura 41 Matriz de estados (partículas evaluadas) modelo 3	28
Figura 42 Diagrama del proceso mejorado	29
Figura 43 Construcción de círculo para crear partículas.....	30
Figura 44 Operación AND entre círculo obtenido y trayectoria actual	30
Figura 45 Partículas del filtro de partículas.....	31
Figura 46 Predicción en el instante k+1	32
Figura 47 Comparación de diferentes sigma en los modelos	35
Figura 48 Comparación de error entre modelos con sigma igual a 1	36
Figura 49 Comparación de errores en trayectorias intersecantes separadas	37
Figura 50 Comparación de errores en trayectorias intersecantes unidas	38
Figura 51 Comparación de recorridos del modelo 3.....	39
Figura 52 Diagrama de flujo de la interfaz gráfica de usuario	40
Figura 53 Pantalla de inicio	41
Figura 54 Pantalla nueva prueba.....	42
Figura 55 Directorio de imágenes.....	43
Figura 56 Pantalla seleccionar directorio	43

Figura 57 Pantalla procesamiento de imágenes.....	44
Figura 58 Pantalla prueba healthspan	45
Figura 59 Pantalla estadística de pruebas.....	46
Figura 60 Modelo de reporte para pruebas individuales	47
Figura 61 Modelo de reporte para pruebas iguales.....	48
Figura 62 Modelo de reporte para pruebas grupales.....	49

Índice de tablas

Tabla 1 Descripción de operaciones realizadas posterior al sensado	22
Tabla 2 Comparación de gráficas de diferentes valores de sigma.....	35
Tabla 3 Comparación de diferentes sigma en los modelos	35
Tabla 4 Comparación de errores en trayectorias individuales.....	36
Tabla 5 Comparación de errores en trayectorias intersecantes separadas.....	37
Tabla 6 Comparación de errores en trayectorias intersecantes unidas	38
Tabla 7 Comparación de recorridos del modelo 3	39

Índice de ecuaciones

Ecuación 1 Ángulo de trayectoria alfa.....	26
Ecuación 2 Amplitud generado por el cuerpo del C. elegans	27
Ecuación 3 Evaluación de estados para cada modelo	31
Ecuación 4 Función de gauss	31
Ecuación 5 Sumatoria de todos los pesos hallados.....	32
Ecuación 6 Suma pondera para hallar valor de predicción	32
Ecuación 7 Suma integrada del movimiento	33
Ecuación 8 Valor de dispersión del movimiento	33
Ecuación 9 Error medio en cada trayectoria	34
Ecuación 10 Error medio total.....	34

1. Introducción

1.1. Descripción del problema

Para poder analizar la conducta de los *C. elegans* ante diferentes tipos de alimentación se necesita conocer el total de movimiento realizado durante el recorrido de su trayectoria, así como su valor de dispersión.

La utilización del microscopio como instrumento de control para realizar inspección manual es un proceso que demanda mucho tiempo, si bien esto es una tarea más fácil de observación que usar el ojo humano, es una tarea imposible a la hora de puntualizar unidades de medidas de desplazamiento de objetos en movimiento, y si se quisiera estimar conductas o patrones hay que tener en cuenta que este método está sujeto a múltiples errores humanos, y de instrumentación sin mencionar la incertidumbres de los resultados.

1.2. Solución propuesta

Se propone realizar una interfaz gráfica de usuario (GUI), utilizando la plataforma de programación de Java en conjunto con las librerías de OpenCV para el procesamiento de imágenes, también contará con conexión a la base de datos MySQL y generación automática de tres tipos diferentes de reportes, el primero para cada prueba individual, el segundo para cada grupo de pruebas (pruebas del mismo grupo alimenticio) y un tercer reporte el cual conlleva gráficos y tablas estadísticas de lo procesado en todas las pruebas anexadas.

La interfaz tiene el objetivo de ayudar al usuario en el almacenamiento, control y presentación de la información procesada de forma automática y visual, a su vez gestionar la base de datos de forma dinámica desde la aplicación con el fin de corroborar los resultados obtenidos realizando las pruebas necesarias.



Figura 1 Solución propuesta

1.3. Objetivos

1.3.1. Objetivos generales

- Diseñar un sistema automático de visión para experimentos healthspan.
- Detección y seguimiento de nematodos *C. elegans* de forma individual durante toda su trayectoria.
- Análisis y comparación de distancia y valor de dispersión de *C. elegans* en diferentes tipos de pruebas.

1.3.2. Objetivos específicos

- Realizar una buena segmentación eliminando ruido y elementos no pertenecientes a los *C. elegans* durante su recorrido.
- Desarrollar un modelo de comportamiento de los nematodos *C. elegans*.
- Estimar localización de *C. elegans* usando filtro de partículas.
- Determinar y ajustar parámetros de control del procesamiento de imágenes.
- Diseñar diferentes interfaces gráficas para gestionar la información del procesamiento de imágenes realizado en cada prueba.
- Realizar la comunicación entre la plataforma de desarrollo de Java, la base de datos MySQL y la aplicación de generación de reportes JasperReport.
- Evaluación de resultados obtenidos.

1.4. Resultados esperados

- El algoritmo de seguimiento de los *C. elegans* podrá estimar la posición de cada nematodo y seguirlo durante todo su recorrido, esto incluirá también trayectorias cruzadas.
- La interfaz gráfica proporcionará una visualización de los resultados en tiempo real durante la ejecución de la aplicación.
- El usuario podrá crear o eliminar la información almacenada para tres tipos de reportes:
 - Reporte de pruebas individuales
 - Reporte de pruebas en conjunto
 - Reporte estadístico para evaluación de pruebas
- Los reportes podrán ser guardados en formato pdf.
- La aplicación automática de visión tendrá compatibilidad en los sistemas operativos de Windows y Linux.

2. Estado del arte

2.1. Antecedentes

El aumento de la competitividad es un requerimiento primordial en la situación económica actual. Un factor clave para mejorar la competitividad es aumentar la productividad incorporando sistemas de inspección que permitan la automatización de diferentes procesos. Los sistemas de inspección visual automática, basados en visión por computador, han demostrado ser una herramienta fundamental para mejorar los procesos. Estos sistemas de visión permiten la inspección continua, evitando fatigas y distracciones, y facilitando la cuantificación de las variables de calidad en prácticamente el 100% de la producción. Esto se traduce, no sólo en una mejora de la calidad final de los productos, sino también en un ahorro en términos económicos y medioambientales.

Durante las últimas décadas se han podido resolver multitud de aplicaciones mediante la implantación de sistemas de inspección 2D en la industria. El principal problema a resolver en estos sistemas ha sido la gran variabilidad de las imágenes que puede dificultar la segmentación de los objetos de interés. Para resolver este problema se han propuesto algunas técnicas robustas de segmentación que intentan absorber dicha variabilidad (Benlloch, 1995, Octubre); (Sánchez A. J., 2008). Además, también se han propuesto infinidad de soluciones de seguimiento de objetos, como por ejemplo (Sánchez A. &., 2000).

Por otro lado, los últimos avances producidos en la aplicación de técnicas de seguimiento de objetos articulados y deformables, (E.M. Berti, 2012a); (Berti, 2012b); (Martinez, 2016); (Martinez E. S., 2017a); (Martinez E. N., 2017b); (Martinez-Berti, 2017); están permitiendo automatizar algunas aplicaciones de monitorización complejas que hace unos años eran impensables.

2.2. HealthSpan

El healthspan (lapso de salud) que los organismos llevan durante su ciclo de vida es un punto importante para diversas investigaciones, el concepto en sí del “healthspan” es interpretado de diferentes maneras, de las cuales una de ellas se puede expresar como el lapso de vida de un organismo el cual está exento de enfermedades vinculadas con la edad o la falta de funciones propias. (A.I. Yashin, 2014)

Otro término que va a la par del healthspan es el lifespan (lapso de vida) ambos son estudiados utilizando diferentes tipos de modelos orgánicos, mamíferos, no mamíferos e inclusive en organismos de pequeña escala incluyendo en esta categoría a las bacterias. Dentro de los modelos no mamíferos se distinguen 3 organismos los cuales son usados para el aprendizaje del ciclo de vida y estos son: *S. cerevisiae*, *C. elegans*, *Melanogusler*. (Acton, 2012)

Algunos factores claves que diferencian algunos estudios relacionados con el lapso de salud en organismos son su alimentación, forma de vida, habitad, entre otros, siendo el más importante su régimen alimenticio.

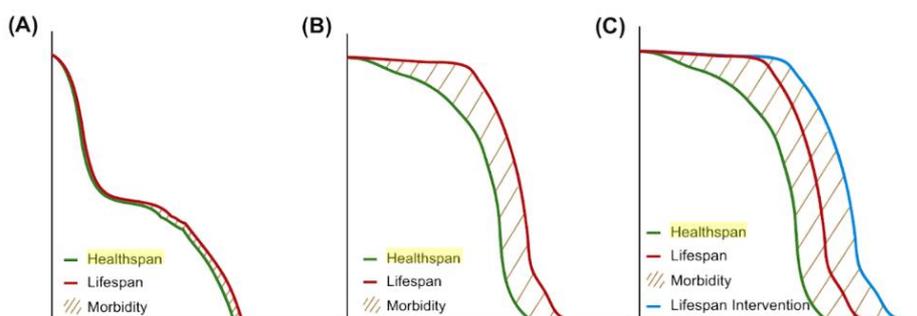


Figura 2 Efecto de la medida médica y de salud pública antes y después del siglo XX

(A) Antes del siglo XX, previo a la intervención de la medicina moderna cuando no se suministraba vacunas o antibióticos. (B) Entre el siglo XX e inicios del siglo XXI, con el uso de intervenciones médicas y medidas de salud pública. (C) Posterior al siglo XXI, luego de avances científicos en el campo de la medicina. (Theodore Friedmann, 2015)

2.3. Filtro de partículas

En el año de 1940 se originó el filtro de partículas con la investigación realizada por Nicholas Metropolis, y Norbert Wiener quien aportó también con algunas ideas, aunque similares a las de Metropolis, pero no fue hasta después de 40 años su implementación debido potencia computacional de la época. Metropolis en su investigación planteó la idea de analizar las características del grupo de partículas en vez de las características individuales de cada partícula, esta idea surgió luego de analizar el juego de cartas solitario, en el cual se realizaba algunas preguntas entre ellas ¿Cuál es la posibilidad de ganar?, la cual lo llevaba a un análisis numérico casi imposible de resolver, pero si el juego se realizaba muchas veces y en algunos casos se gana entonces la posibilidad de ganar se puede ajustar como el total de éxitos dividido para el número de intentos realizados, esto fue lo que propuso Metropolis luego de su análisis y fue lo que llevo al origen del filtro de partículas. (Simon, 2006)

En la actualidad existen diferentes tipos de algoritmos con funciones estadísticas para el cálculo, estimación y aproximación de parámetros, siendo uno de los más utilizados el filtro de partículas por su estimación en sistemas lineales y totalmente no lineales, a diferencia del filtro de kalman o filtro de kalman extendido (EKF, por su siglas en inglés), el filtro de partículas abarca un rango mayor de posibilidades en el momento de su implementación con el único inconveniente del coste computacional. (Candy, 2016)

2.4. C. elegans

Caenorhabditis elegans o su abreviatura *C. elegans* es una clase de nematodo de la familia Rhabditidae, no es un parásito, su promedio de vida es de 2-3 semanas en condiciones normales, su régimen alimenticio se basa en bacterias y hongos los cuales se desarrollan en frutas o vegetales en descomposición. Es utilizado desde el año de 1960 como modelo de desarrollo de gran importancia dentro del campo de la biología, para el estudio y tratamiento de diferentes tipos enfermedades, por su comportamiento durante su ciclo de vida, en especial en una edad adulta avanzada en la cual presentan pérdidas físicas, deterioro de tejidos y movilidad, entre otras características parecidas a los seres humanos en una edad avanzada, esto lo hace un modelo orgánico ideal para el análisis del healthspan. (Anders Olsen, 2016)

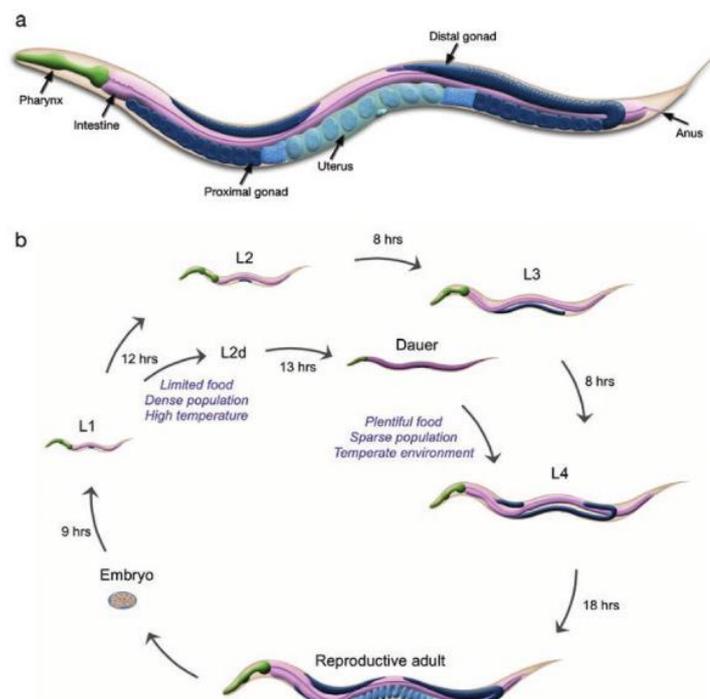


Figura 3 *C. elegans* anatomía y ciclo de vida

(Anders Olsen, 2016)

2.5. OpenCv

OpenCv o por sus siglas en inglés Open Source Computer Vision (visión por computadora de fuente abierta) se la puede definir como un conjunto de herramientas que utilizan algoritmos de procesamiento avanzados para el análisis de imágenes y video. (Modrzyk, 2018)

OpenCV fue presentado a inicios de 1999, como una herramienta de visión artificial sencilla de emplear, que ayudaría a los usuarios a desarrollar

aplicaciones de visión de imagen y video en tiempo real, en un principio estas herramientas fueron creadas solo para C++, pero no fue hasta después de 4 años en donde recién se pudo utilizar dentro de las librerías de Java. (Baggio, 2015)

La esencia o corazón de openCV dentro de la plataforma de Java es un objeto de n-dimensiones tipo matriz cuyo formato es Mat, en el cual se reconocen los diferentes tipos de archivos de imágenes y donde se pueden realizar todo tipo de operaciones. (Modrzyk, 2018)

2.6. JavaFx

JavaFX es un complemento de Java el cual abarca un grupo de herramientas y controles para crear una GUI (interfaz gráfica de usuario), aprovechando también las librerías y herramientas existentes java. La meta y finalidad de JavaFx es permitir que los usuarios puedan usar todas estas herramientas y controles para lograr desarrollar mejores aplicaciones de forma rápida y sencilla, dentro de un entorno más gráfico usando JavaFX Scene Builder o bien solo a través de líneas de comandos (java puro), dado que en la actualidad cada vez se están utilizando sistemas multiprocesador y sistemas multi-thread (multi-proceso), utilizando GPU (unidad de procesamiento gráfico) es ahí donde entra este complemento de Java que ayuda al usuario a desarrollar programas mucho más eficientes. (Gail Anderson, 2009)

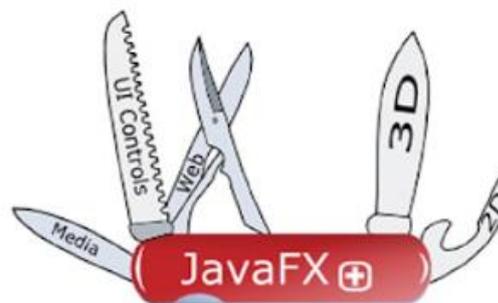


Figura 4 JavaFX es la navaja suiza de herramientas GUI de java

(Mark Heckler, 2014)

2.7. Mysql

La expresión base de datos usualmente se la relaciona con aquel sistema de almacenamiento capaz de realizar operaciones en lenguaje SQL, pero en el tiempo actual va más allá que almacenar datos, la terminología correcta sería RDBMS (Sistema Gestor de Bases de Datos Relacionales), la cual comprende múltiples herramientas de acceso, administración y gestión de la información almacenada, comunicaciones, seguridad, entre otras aplicaciones. (Mark Matthews, 2003).

Mysql al igual que otras bases de datos de código abierto se volvieron muy utilizadas luego del inmenso crecimiento que tuvo internet, estas a diferencia

de versiones comerciales se pueden ejecutar fácilmente en dispositivos de costo reducido y bajo recursos de hardware, pudiendo utilizar diferentes tipos de consultas, y herramientas del entorno de programación SQL.

Otra gran ventaja que poseen estas bases de datos de fuente abierta (open source) es la flexibilidad y facilidad que tiene para poder trabajar con múltiples plataformas de programación como lo son java, c++, net, Python, php, entre otras, todas estas utilizan diferentes tecnologías para poder conectarse a Mysql. Java utiliza la librería de JDBC para establecer la comunicación, una vez conectados puede realizar declaraciones o Statements para realizar las diferentes consultas (query) hacía la base de datos.

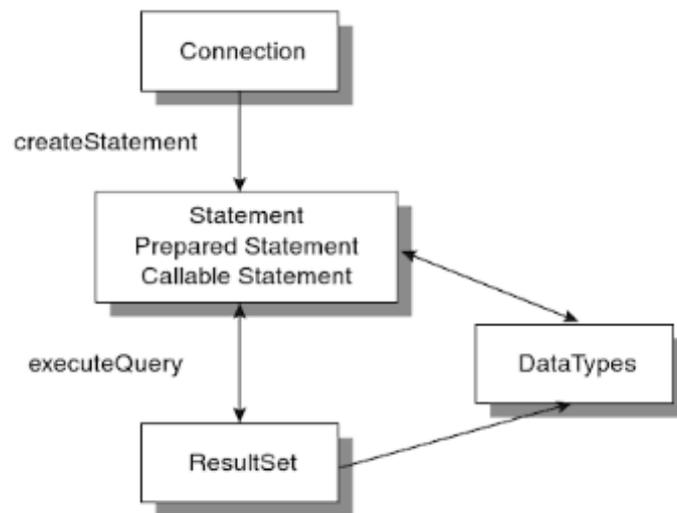


Figura 5 Modelo de desarrollo de base de datos

(Mark Matthews, 2003)

3. Descripción del proceso

3.1. Descripción del proceso realizado

El proceso para el seguimiento se ha dividido en dos etapas las cuales se las ha nombrado pre-procesamiento y procesamiento, estas se explican más detalladamente en los próximos capítulos.

El pre-procesamiento consiste en eliminar el ruido en la imagen y encontrar las trayectorias, el número total de c. elegans dentro de las mismas y el instante K más apropiado para comenzar el seguimiento del recorrido realizado.

El procesamiento utiliza toda la información del pre-procesamiento para crear partículas dentro de las trayectorias y evaluarlas utilizando diferentes tipos de parámetros y ajustes a fin de encontrar el recorrido de cada C. elegans dentro de las múltiples posibilidades de desplazamiento que pueden realizar.

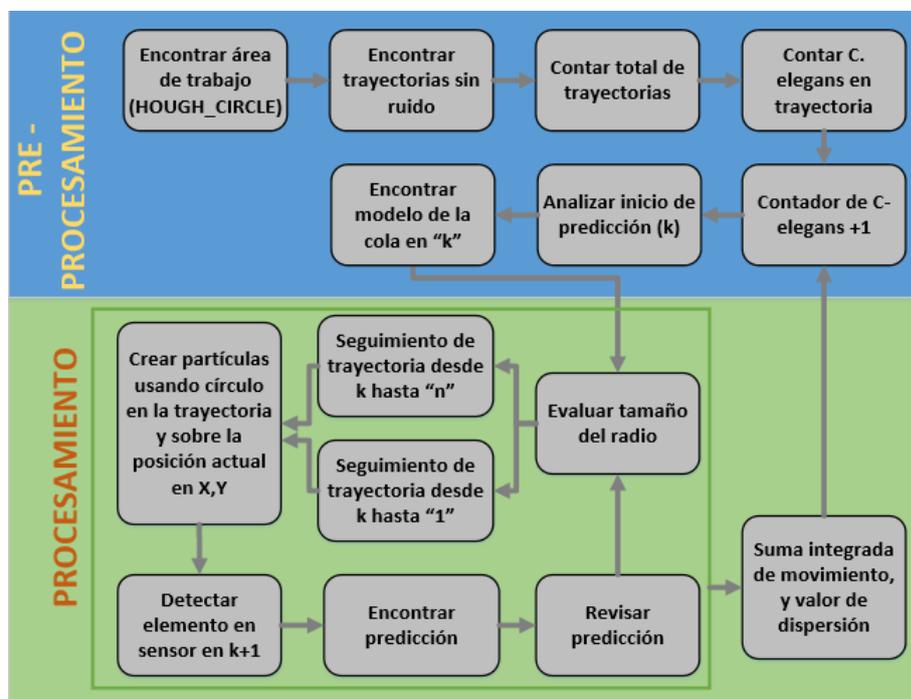


Figura 6 Diagrama del proceso

3.2. Encontrar área de trabajo

Para encontrar el área de trabajo (lugar de movimiento de los C. elegans), se utiliza la transformada de Hough de círculos para identificar de forma automática la placa de Petri. Esta herramienta nos proporciona la ubicación del centro y el radio donde se encuentra ubicada la paca de Petri dentro de la imagen, con lo cual se construye un círculo binario, es decir lo que está dentro de la placa de Petri tendrá valor de 1 y todo lo que está afuera valor de 0, esto se utiliza para eliminar el ruido fuera de la placa de Petri, así como también el borde de la misma la cual absorbe ruido por los cambios de iluminación.



Figura 7 Localización del área de trabajo

3.3. Encontrar trayectorias sin ruido

Esta parte consiste primero en identificar todas las trayectorias realizadas por los *C. elegans* durante las “ n ” imágenes, para ello es necesario una buena segmentación de la escena de trabajo, a fin de poder encontrar e identificar cada trayectoria realizada se realizan una serie de operaciones algébricas y lógicas (operadores puntuales) para eliminar el ruido en la imagen.

Se considera ruido a todo lo que no se parte del cuerpo de *C. elegans* durante toda la trayectoria, esto es el alimento que consumen, cambios de iluminación en la escena, restos de comida, entre otros.

Para no considerar el ruido durante la segmentación de la trayectoria se trabaja con la diferencia de la imagen actual y la imagen siguiente esto da como resultado algunas áreas claras y oscuras que provienen del desplazamiento de los *C. elegans* más un fondo de color diferente al desplazamiento.

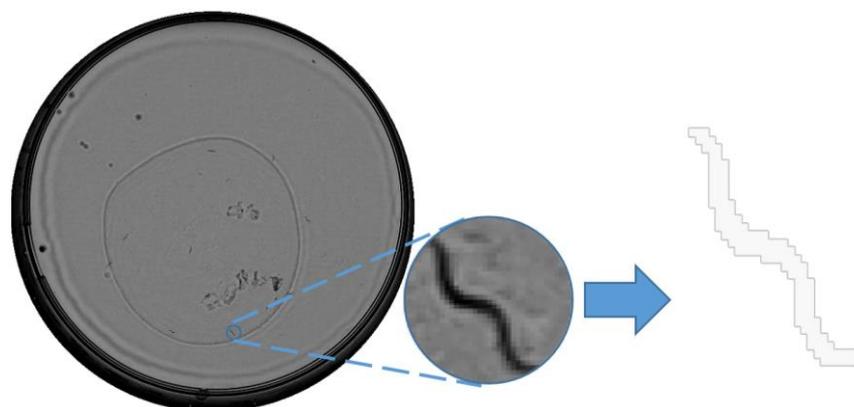


Figura 8 Imagen en instante K

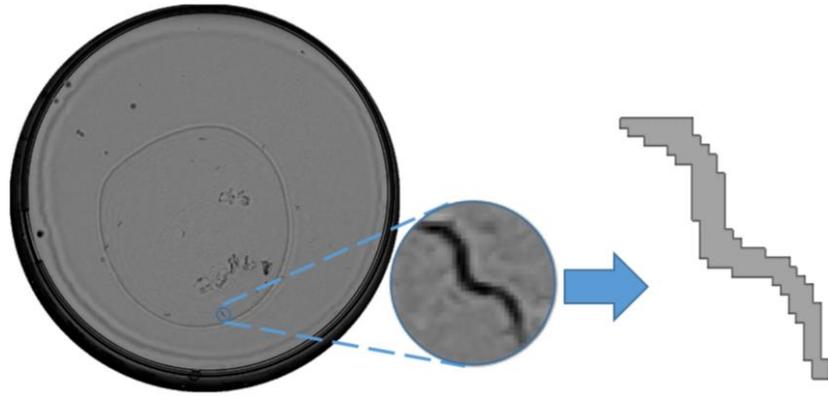


Figura 9 Imagen en instante K+1

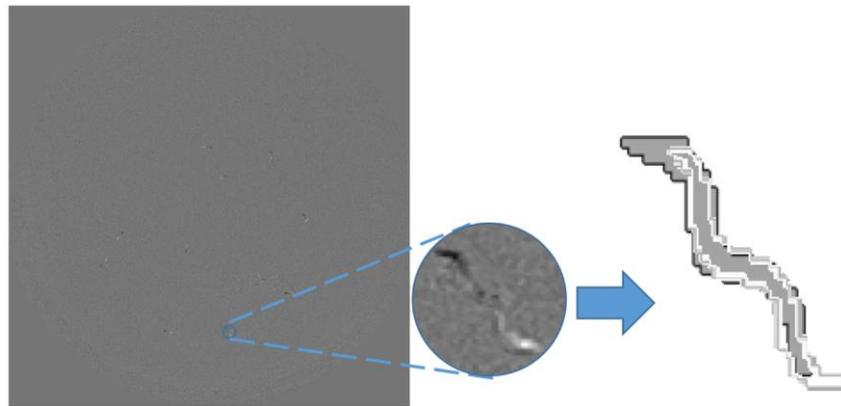


Figura 10 Diferencia de la imagen actual y la imagen siguiente

Como se puede apreciar en la **Figura 10**, la diferencia de la imagen en el instante k y la imagen en el instante $k+1$ produce áreas claras (cola) y oscuras (cabeza), estas áreas son segmentadas utilizando un valor de umbral y sumadas durante las n imágenes a procesar, de igual manera se realiza la segmentación del cuerpo del *C. elegans* utilizando la misma imagen de diferencias, pero con un valor diferente de umbral.

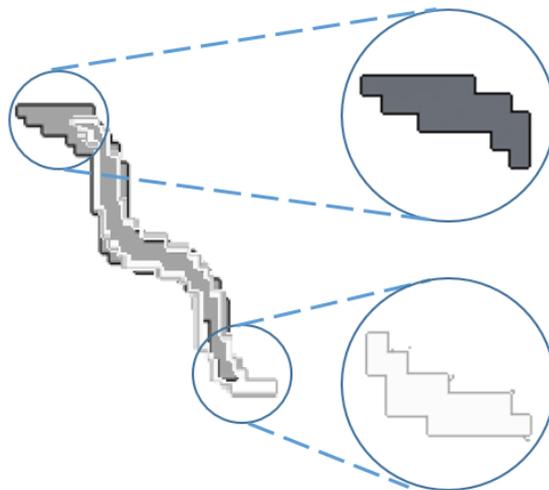


Figura 11 Segmentación de áreas claras y oscuras

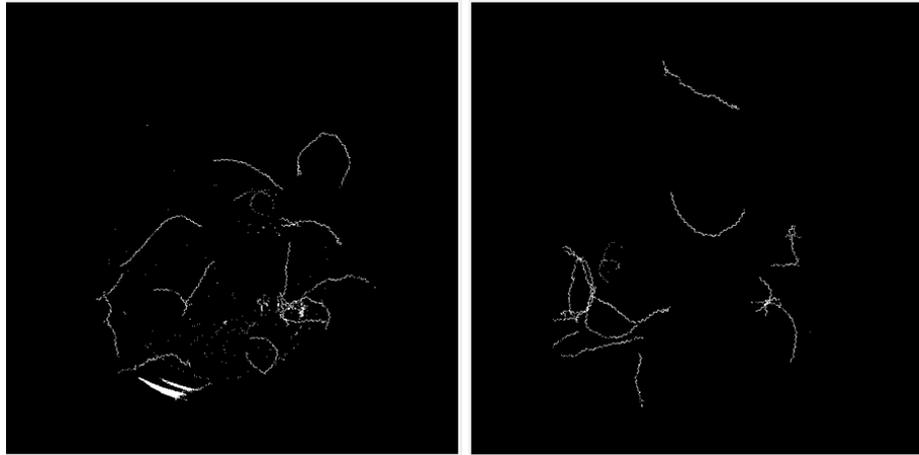


Figura 12 Sumatoria de áreas claras durante 60 imágenes en 2 placas diferentes



Figura 13 Sumatoria de áreas oscuras durante 60 imágenes en 2 placas diferentes

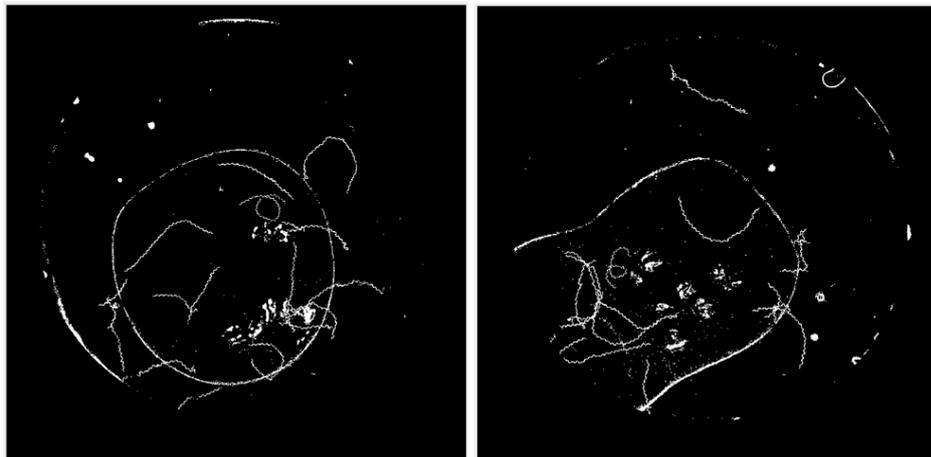


Figura 14 Sumatoria de segmentación cuerpo de C. elegans durante 60 imágenes en 2 placas diferentes

Como se puede apreciar en la figure 11 y 12, la segmentación de áreas claras y oscuras nos proporciona un recorrido del C. elegans en el cual podemos ver partes del recorrido separadas y otras unidas, adicional a esto no siempre se

elimina por completo el ruido en la escena, en ocasiones el ruido puede estar presente por diferentes factores, comúnmente se encuentra por los cambios de iluminación en el escenario durante breves momentos.

Por otro lado, la sumatoria de la segmentación del cuerpo del C. elegans nos da como resultado todo el desplazamiento realizado sin cortes en las trayectorias, pero con todo el ruido presente en la escena.

Para resolver este problema y poder obtener la segmentación de la trayectoria sin ruido, se realiza una serie de operaciones matemáticas usando estas 3 imágenes, la sumatoria de áreas claras, la sumatoria de áreas oscuras y la sumatoria de la segmentación del cuerpo de C. elegans, para la cual se las ha nombrado I1, I2 y I3 respectivamente.

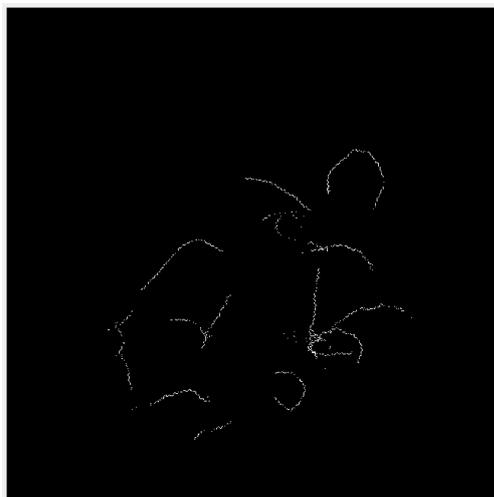


Figura 15 Operación AND sobre las 3 imágenes (I1 & I2 & I3)

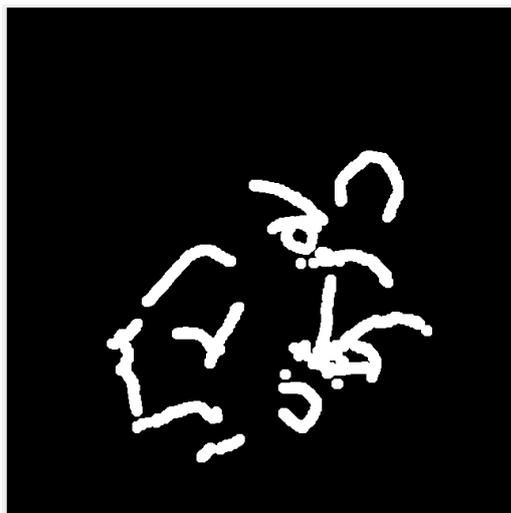


Figura 16 Dilatación de 20 sobre resultado de operación AND

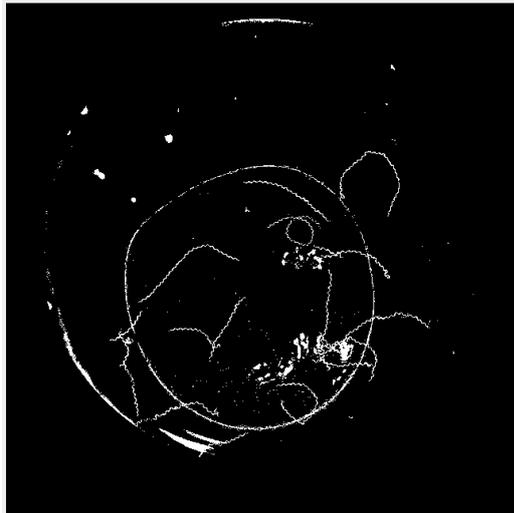


Figura 17 Operación OR sobre las 3 imágenes ($I1 \text{ OR } I2 \text{ OR } I3$)



Figura 18 Operación AND sobre resultado de OR y dilatación



Figura 19 Operación Close sobre resultado de operación AND

3.4. Contar y guardar total de trayectorias

Para esta operación se ha utilizado la función **findContours** la cual nos proporciona cada una de las trayectorias, estas son almacenadas en un arreglo a fin de tener las trayectorias separadas e identificadas individualmente.

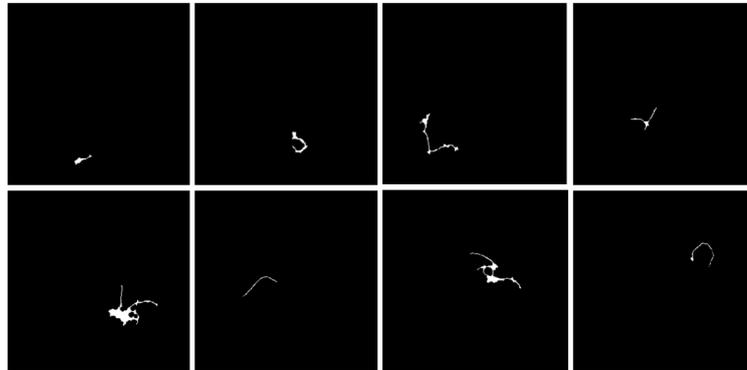


Figura 20 Trayectorias separadas e identificadas

3.5. Analizar número de C. elegans en cada trayectoria y su inicio de recorrido

Para esta parte del pre-procesamiento se selecciona cada una de las trayectorias y se realiza el procesamiento de las n imágenes nuevamente a fin de encontrar la imagen “k” donde hay el mismo número de desplazamientos positivos y negativos. Se realiza de esta manera dado que el comportamiento de los C. elegans es estar separados.

Para la segmentación de la cola del C. elegans, se realiza una pequeña dilatación con el fin de juntar áreas muy cercanas a la cola, dichas área en la mayoría de los casos pertenece a la misma cola del C. elegans, pero se ha dividido por contrastes de iluminación en la escena.



Figura 21 Trayectoria 5 diferencia de imágenes 25,26

Como se aprecia en la **figura 21**, en la trayectoria 5 en el instante K igual a 25 la cola de los cuatro C. elegans presentes en la trayectoria se encuentran separadas y el desplazamiento de los C. elegans en la diferencia de imágenes ha dado igual número de áreas claras y áreas oscuras.

4. Descripción de los modelos

4.1. Detección de elemento en el sensor

La detección de la parte inferior (cola) en el instante de tiempo k consiste primero en detectar el cuerpo del *C. elegans* en el tiempo " k " y en el instante " $k+1$ " y realizar una operación "OR" para obtener un área más grande y precisa que asegure el cuerpo del nematodo en 2 períodos de tiempo, el resultado de la operación OR se le aplica una operación AND con la diferencia de imágenes del período k y $k+1$ (cola del *C. elegans*). Realizar esta secuencia de operaciones garantiza la eliminación del ruido presente durante estos 2 períodos k , $k+1$. Posterior a estas operaciones lógicas se realiza operaciones de morfología de imágenes para unir partes de la cola separadas por la iluminación o desplazamientos atípicos del *C. elegans*. Se realiza las mismas operaciones para detectar la parte superior (cabeza) del *C. elegans*.



Figura 22 Ejemplo de partes de la segmentación separadas

BW11 en k	BW12 $k+1$	$BWW = BW11 \text{ OR } BW12$
BW2 cola en k	$BP = BWW \& BW2$	Open = 2
Dilatado = 20	Close = 20	Erosionado = 15

Tabla 1 Descripción de operaciones realizadas posterior al sensado

4.2. Modelo 1: Posición velocidad

El primer modelo utilizado es el más simple, el cual se trata de un modelo de posición y velocidad, donde para el estado inicial de posición se toman los valores X e Y del centroide de la cola del *C. elegans* en el instante " k " y para los valores de velocidad se toma cero como inicial.

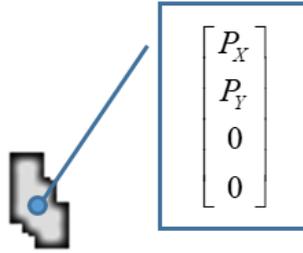


Figura 23 Estado inicial del modelo 1

La acción de control o modelo del sistema es una matriz 4x4 como se aprecia en la **figura 24**. Los estados de predicción se obtienen multiplicando cada uno de los vectores de partículas por la acción de control.

$$m = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 24 Modelo 1 posición y velocidad

La evaluación de cada partícula consiste multiplicar cada una de estas por el modelo. La creación de partículas se detalla en el siguiente capítulo.

$$\begin{bmatrix} X + Vx \\ Y + Vy \\ Vx \\ Vy \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Vx_1 \\ Vy_1 \end{bmatrix}$$

Figura 25 Evaluación de cada partícula para obtener matriz de estados del modelo 1

Luego de multiplicar cada partícula generada por la acción de control (modelo) se obtiene las posibles predicciones (estados) de la posición del C. elegans en k+1.

$$P_{k+1} = \begin{bmatrix} (X_1 + Vx_1)_{1,1} & (X_2 + Vx_2)_{1,2} & \cdots & \cdots & (X_{n-1} + Vx_{n-1})_{1,n-1} & (X_n + Vx_n)_{1,n} \\ (Y_1 + Vy_1)_{2,1} & (Y_2 + Vy_2)_{2,2} & \cdots & \cdots & (Y_{n-1} + Vy_{n-1})_{2,n-1} & (Y_n + Vy_n)_{2,n} \\ (Vx_1)_{3,1} & (Vx_2)_{3,2} & \cdots & \cdots & (Vx_{n-1})_{3,n-1} & (Vx_n)_{3,n} \\ (Vy_1)_{4,1} & (Vy_2)_{4,2} & \cdots & \cdots & (Vy_{n-1})_{4,n-1} & (Vy_n)_{4,n} \end{bmatrix}$$

Figura 26 Predicción (estados) modelo 1

$$\begin{bmatrix} X1_{1,1} & X1_{1,2} & \cdots & \cdots & X1_{1,n-1} & X1_{1,n} \\ Y1_{2,1} & Y1_{2,2} & \cdots & \cdots & Y1_{2,n-1} & Y1_{2,n} \\ Vx_{3,1} & Vx_{3,2} & \cdots & \cdots & Vx_{3,n-1} & Vx_{3,n} \\ Vy_{4,1} & Vy_{4,2} & \cdots & \cdots & Vy_{4,n-1} & Vy_{4,n} \end{bmatrix}$$

Figura 27 Matriz de estados (partículas evaluadas) del modelo 1

4.3. Modelo 2: Posición velocidad cabeza – cola

El segundo modelo es igual al modelo de posición velocidad descrito en el modelo anterior, pero este involucra la parte oscura dentro del modelo (cabeza del C. elegans). Para decidir cuál área de cola se relaciona con su respectiva de área de cabeza, se utiliza la menor distancia euclídea para comparar ambos grupos de áreas, esto es posible gracias al procedimiento de inicio de recorrido descrito en el capítulo anterior.

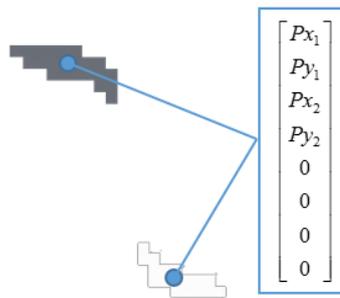


Figura 28 Estado inicial del modelo 2

La acción de control o modelo del sistema es una matriz 8x8 como se aprecia en la **figura 29**.

$$m = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 29 Modelo 2 posición y velocidad cabeza - cola

La creación de cada partícula (vector de posición y velocidad) se realiza de la misma forma que el modelo anterior.

$$\begin{bmatrix} X_1 + Vx_1 \\ Y_1 + Vy_1 \\ X_2 + Vx_2 \\ Y_2 + Vy_2 \\ Vx_1 \\ Vy_1 \\ Vx_2 \\ Vy_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ Vx_1 \\ Vy_1 \\ Vx_2 \\ Vy_2 \end{bmatrix}$$

Figura 30 Evaluación de cada partícula para obtener matriz de estados del modelo 2

De igual manera que el modelo anterior cada partícula es multiplicada por la acción de control obteniendo así las posibles predicciones de la posición en k+1.

$$P_{k+1} = \begin{bmatrix} (X_1 + Vx_1)_{1,1} & (X_1 + Vx_1)_{1,2} & & (X_1 + Vx_1)_{1,n} \\ (Y_1 + Vy_1)_{2,1} & (Y_1 + Vy_1)_{2,2} & & (Y_1 + Vy_1)_{2,n} \\ (X_2 + Vx_2)_{3,1} & (X_2 + Vx_2)_{3,2} & & (X_2 + Vx_2)_{3,n} \\ (Y_2 + Vy_2)_{4,1} & (Y_2 + Vy_2)_{4,2} & \dots & (Y_2 + Vy_2)_{4,n} \\ (Vx_1)_{5,1} & (Vx_1)_{5,2} & \dots & (Vx_1)_{5,n} \\ (Vy_1)_{6,1} & (Vy_1)_{6,2} & & (Vy_1)_{6,n} \\ (Vx_2)_{7,1} & (Vx_2)_{7,2} & & (Vx_2)_{7,n} \\ (Vy_2)_{8,1} & (Vy_2)_{8,2} & & (Vy_2)_{8,n} \end{bmatrix}$$

Figura 31 Predicción (estados) modelo 2

4.4. Modelo 3: Posición velocidad y desplazamiento senoidal

El tercer modelo utilizado es un modelo más complejo donde se utilizan más características para construir un modelo con más información del C. elegans, estas características se relacionan con la forma senoidal que realiza el C. elegans durante su movimiento, y estas son su posición y velocidad lineal, longitud del cuerpo, frecuencia de la forma senoidal realizada, el ángulo de trayectoria que sigue con respecto al eje X del plano de la imagen, estos parámetros son utilizados para calcular valores de posición con respecto a la forma senoidal descrita por el C. elegans en cualquier instante de tiempo.

Para el estado inicial se considera el valor de posición igual al valor del centroide en el instante "k", velocidad lineal y tangencial igual a cero, longitud igual a 10, frecuencia igual 0.6 y ángulo de dirección cero.

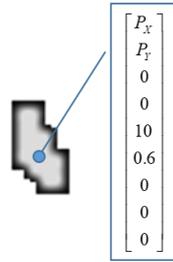


Figura 32 Estado inicial modelo 3

La acción de control o modelo del sistema es una matriz 9x9, donde los valores de posición dependen de los valores de velocidad lineal y senoidal como se aprecia en la **figura 33**.

$$m = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 33 Modelo 3 aproximación senoidal

La creación de partículas se realiza primero de misma forma que para el modelo 1 y 2, a excepción que para los valores de longitud frecuencia y ángulo, estos valores se generan de forma aleatoria dentro de un rango establecido, y estos son [6,12], [0.6,1.6], [0,360] respectivamente. Generado estos valores, estos se adicionan al valor actual y se proceden a evaluar para obtener los valores de posición angular.

El valor del ángulo alfa que el C. elegán sigue en su trayectoria se obtiene con la tangente inversa de la división de las velocidades lineales, mientras que el valor de la amplitud, depende de los valores de longitud y frecuencia como se describe en la **ecuación 2**.

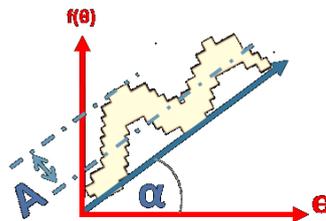


Figura 34 Desplazamiento realizado por el C. elegán

$$\alpha = \tan^{-1} \left(\frac{V_y}{V_x} \right)$$

Ecuación 1 Ángulo de trayectoria alfa

$$A = \frac{L \cdot F}{2\pi}$$

Ecuación 2 Amplitud generado por el cuerpo del C. elegán

Debido que el C. elegán describe un movimiento similar a una forma senoidal los valores de posición X e Y corresponde a los valores θ y $f(\theta) = A \sin(\theta)$ respectivamente.

$$V_{xy} = \begin{bmatrix} A \sin(\theta) \\ \theta \end{bmatrix}$$

Figura 35 Matriz de posición angular

Teniendo en cuenta que el C. elegán se desplaza sobre una trayectoria rotada se construye un matriz de rotación sobre el eje X, con el valor del ángulo alfa hallado.

$$Ma = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

Figura 36 Matriz de rotación alfa

La multiplicación de ambas matrices da como resultado la posición angular.

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} A \sin(\theta) \\ \theta \end{bmatrix}$$

Figura 37 Matriz de posición angular rotada

Se realiza de igual manera para generar el resto de partículas deseadas.

$$\begin{bmatrix} X_{1,1} & X_{1,2} & & X_{1,n-1} & X_{1,n} \\ Y_{2,1} & Y_{2,2} & & Y_{2,n-1} & Y_{2,n} \\ Vx_{3,1} & Vx_{3,2} & & Vx_{3,n-1} & Vx_{3,n} \\ Vy_{4,1} & Vy_{4,2} & & Vy_{4,n-1} & Vy_{4,n} \\ L_{5,1} & L_{5,2} & \dots & L_{5,n-1} & L_{5,n} \\ F_{6,1} & F_{6,2} & & F_{6,n-1} & F_{6,n} \\ \theta_{7,1} & \theta_{7,2} & & \theta_{7,n-1} & \theta_{7,n} \\ \Delta x_{8,1} & \Delta x_{8,2} & & \Delta x_{8,n-1} & \Delta x_{8,n} \\ \Delta y_{9,1} & \Delta y_{9,2} & & \Delta y_{9,n-1} & \Delta y_{9,n} \end{bmatrix}$$

Figura 38 Matriz de partículas del modelo 3

$$\begin{bmatrix} X + Vx + \Delta x \\ Y + Vy + \Delta y \\ Vx \\ Vy \\ L \\ F \\ \theta \\ \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Vx_1 \\ Vy_1 \\ L_1 \\ F_1 \\ \theta_1 \\ \Delta x_1 \\ \Delta y_1 \end{bmatrix}$$

Figura 39 Evaluación de cada partícula para obtener matriz de estados del modelo 3

Luego de multiplicar cada partícula generada por la acción de control (modelo) se obtiene las posibles predicciones (estados) de la posición del C. elean en k+1.

$$P_{k+1} = \begin{bmatrix} (X_1 + Vx_1 + \Delta x_1)_{1,1} & (X_2 + Vx_2 + \Delta x_2)_{1,2} & (X_n + Vx_n + \Delta x_n)_{1,n} \\ (Y_1 + Vy_1 + \Delta y_1)_{2,1} & (Y_2 + Vy_2 + \Delta y_2)_{2,2} & (Y_n + Vy_n + \Delta y_n)_{2,n} \\ (Vx_1)_{3,1} & (Vx_2)_{3,2} & (Vx_n)_{3,n} \\ (Vy_1)_{4,1} & (Vy_2)_{4,2} & (Vy_n)_{4,n} \\ (L_1)_{5,1} & (L_2)_{5,2} & \dots \dots (L_n)_{5,n} \\ (F_1)_{6,1} & (F_2)_{6,2} & (F_n)_{6,n} \\ (\theta_1)_{7,1} & (\theta_2)_{7,2} & (\theta_n)_{7,n} \\ (\Delta x_1)_{8,1} & (\Delta x_2)_{8,2} & (\Delta x_n)_{8,n} \\ (\Delta y_1)_{9,1} & (\Delta y_2)_{9,2} & (\Delta y_n)_{9,n} \end{bmatrix}$$

Figura 40 Predicción (estados) modelo 3

$$\begin{bmatrix} (X1)_{1,1} & (X2)_{1,2} & (Xn)_{1,n} \\ (Y1)_{2,1} & (Y2)_{2,2} & (Yn)_{2,n} \\ (Vx_1)_{3,1} & (Vx_2)_{3,2} & (Vx_n)_{3,n} \\ (Vy_1)_{4,1} & (Vy_2)_{4,2} & (Vy_n)_{4,n} \\ (L_1)_{5,1} & (L_2)_{5,2} & \dots \dots (L_n)_{5,n} \\ (F_1)_{6,1} & (F_2)_{6,2} & (F_n)_{6,n} \\ (\theta_1)_{7,1} & (\theta_2)_{7,2} & (\theta_n)_{7,n} \\ (\Delta x_1)_{8,1} & (\Delta x_2)_{8,2} & (\Delta x_n)_{8,n} \\ (\Delta y_1)_{9,1} & (\Delta y_2)_{9,2} & (\Delta y_n)_{9,n} \end{bmatrix}$$

Figura 41 Matriz de estados (partículas evaluadas) modelo 3

4.5. Modelo 4: Posición velocidad y desplazamiento senoidal restando trayectoria

El modelo 4 posee los mismos parámetros y ajustes hallados en el modelo 3, la diferencia está en los pasos que se agregan durante el pre-procesamiento y procesamiento, estos permiten ponderar de forma positiva y negativa en valor de uno el desplazamiento realizado por el C. elegans durante su trayectoria.

Para el proceso de sumar valores de desplazamiento se crea una matriz de unos del tamaño de las imágenes, posterior a esto se procesan las n imágenes nuevamente utilizando la función de detectar elemento en el sensor descrita al inicio del capítulo, dicha función es una copia de la función en la cual se cambia los parámetros de apertura, dilatación, cerrado y erosionado por los valores 1, 3, 3, 2 respectivamente. Obtenido el valor del sensor se procede a multiplicar elemento por elemento por la matriz de unos creada anteriormente y el resultado se va sumando en una nueva matriz.

El proceso de resta de valores de desplazamiento se realiza de la misma forma que el procedimiento descrito anteriormente con la diferencia que luego de obtener la predicción de la siguiente posición esta se compara con las áreas cercanas a la predicción en caso de que el elemento capturado por el sensor este fragmentado en varias partes, obtenida el área más cercana a la predicción se realiza una operación de multiplicación y resta con la matriz de unos y la matriz de sumatoria hallada, con el fin de ir quitando áreas en la trayectoria total, para que estas no influyan dentro del cálculo del filtro de partículas y de esta manera la predicción del C. elegans durante su recorrido sea única.

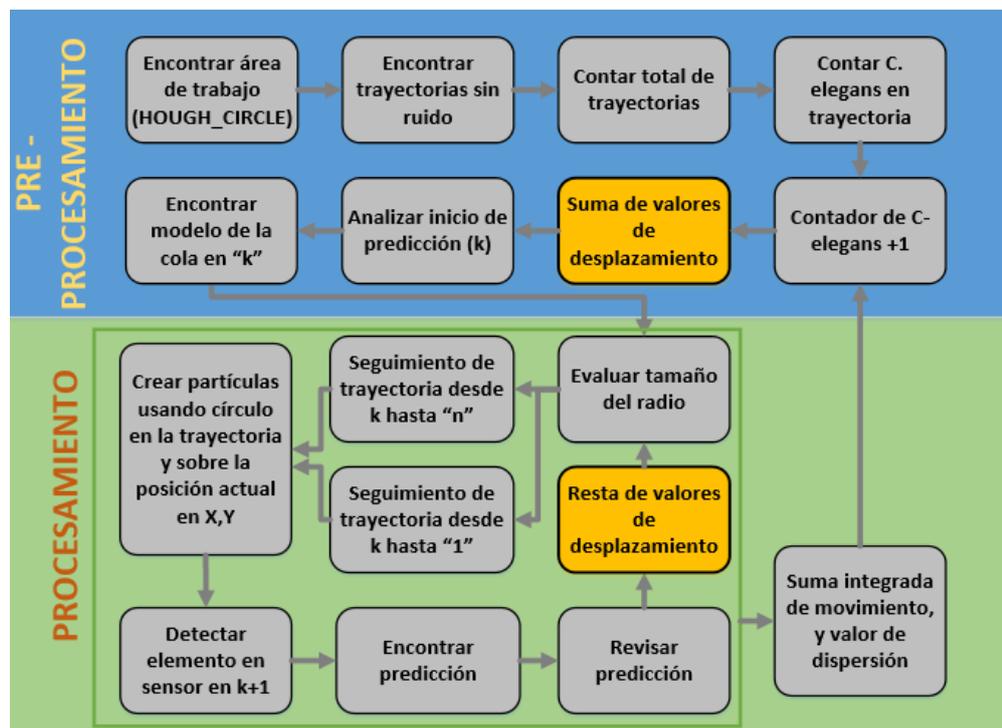


Figura 42 Diagrama del proceso mejorado

5. Descripción del procesamiento

5.1. Creación de partículas

Para la creación de partículas se construye un círculo binario (pixeles dentro del área del círculo igual a 1 y fuera de estos igual a 0), se utiliza como centro los valores de centroide obtenido de la predicción en el instante k , para el radio se utiliza dos consideraciones, cuando la trayectoria del *C. elegans* es individual, un solo nematodo por trayectoria, este se puede desplazar recorriendo grandes distancias como también lo puede hacer en pequeños desplazamientos entonces para esta primera consideración se toma como radio 20, el cual abarca gran cantidad de área. Para la segunda consideración se toma en cuenta el número de *C. elegans* en la trayectoria para construir un círculo de radio 10, el cual cubrirá los pequeños desplazamientos que realizan los nematodos cuando están cerca.

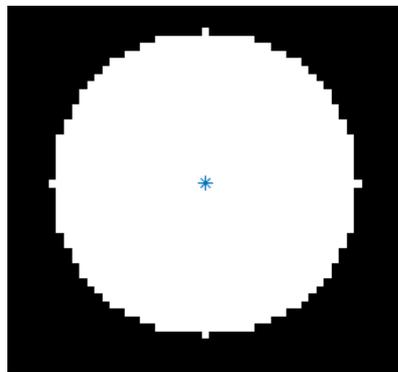


Figura 43 Construcción de círculo para crear partículas

Luego de obtener el círculo binario se realiza una operación matemática AND entre el círculo obtenido y la trayectoria en la que se está trabajando, esto dará como resultado las partículas del filtro de partículas que posteriormente se utiliza para predecir la siguiente posición del nematodo. El área de la trayectoria encontrada con la operación AND entre las dos imágenes mencionadas es el área donde se está desplazando el *C. elegans*.

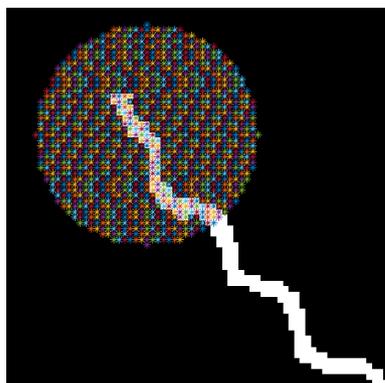


Figura 44 Operación AND entre círculo obtenido y trayectoria actual

Para el área encontrada se extrae sus valores de pixeles en X e Y, los cuales serán la parte de posición del filtro de partículas, para la parte de velocidad se realiza mediante la diferencia de la posición actual (predicción en el instante k) y el pixel obtenido en la parte de posición. Como se puede apreciar en la **figura 45**, está área nos proporciona menos partículas lo cual hace más fácil su evaluación.



Figura 45 Partículas del filtro de partículas

5.2. Evaluación de partículas y predicción

Para evaluar cada posible predicción (partícula) se haya primero un valor correspondiente a la distancia entre dicha predicción y el valor del sensor en el instante k+1, esto se obtiene utilizando la función “**distanceTransform**” para cada valor de X e Y de la posible predicción.

$$W_i = Bw_dist(X_i, Y_i)$$

Ecuación 3 Evaluación de estados para cada modelo

Obtenido el valor de distancia de la predicción con respecto a siguiente elemento del sensor, se procede a evaluar el resultado utilizando la función gaussiana o campana de Gauss para obtener el peso correspondiente a su distancia al centro del elemento. Entre más alejado esta la predicción con respecto al elemento su valor de peso será más pequeño, caso contrario si el valor de predicción está lo más cerca del elemento este tendrá un valor más alto. Los valores de sigma y mu son valores cercanos a la unidad, debido a que los valores de partículas generados no difieren en grandes cantidades.

$$WW(j) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(W_i - \mu)^2}{2\sigma^2}}$$

Ecuación 4 Función de gauss

A fin de obtener el valor de predicción en el instante $k+1$, se realiza una media ponderada con los “ m ” valores de peso encontrados con la función gaussiana multiplicados por los “ n ” valores de predicción hallados multiplicando la matriz modelo con cada partícula.

$$S_{ww} = \sum_m^{p=1} WW(p)$$

Ecuación 5 Sumatoria de todos los pesos hallados

$$P_f = \sum_m^{k=1} \left(\frac{WW(k)}{S_{ww}} \cdot P_{t+1}(K) \right)$$

Ecuación 6 Suma ponderada para hallar valor de predicción

Como se puede apreciar en la **figura 46** el punto en rojo indica el resultado del filtro de partículas luego de su evaluación de la media ponderada utilizando la función de gauss. Esto parecería innecesario para las trayectorias individuales pero como se muestra en la **figura 22** hay veces que estas trayectorias individuales cuentan con áreas de cola que se encuentran fragmentadas por diferentes factores.

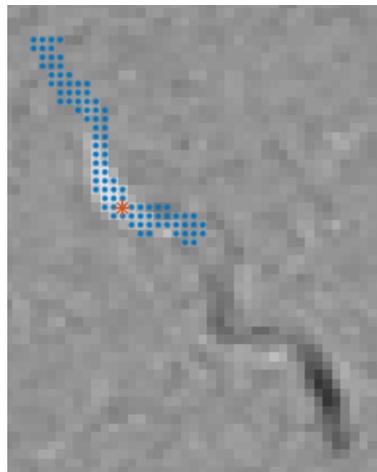


Figura 46 Predicción en el instante $k+1$

5.3. Revisión de predicción

Para los casos particulares donde el C. elegans no ha realizado ningún movimiento o el sensor no ha capturado movimiento, se tomará como valor de predicción en el instante $k+1$ el valor de predicción en el instante k , debido a que al no tener elemento en el sensor dará como resultado un valor de distancia elevado y por consiguiente un valor de peso igual a cero evaluado por la función gaussiana.

5.4. Cálculo de movimiento y valor de dispersión

Para cada procesamiento de trayectoria realizado se va almacenando el valor de predicción hallado con el fin de calcular el total de movimiento recorrido y el valor de dispersión realizado durante la trayectoria.

El total de movimiento recorrido es la suma integrada de la distancia euclídea de todos los puntos encontrados, es decir la norma de la diferencia de valores X e Y con el siguiente instante de tiempo de cada predicción, así como se describe en la **ecuación 7**.

$$SI = \sum_{i=1}^{n-1} \sqrt{(Px_i - Px_{i+1})^2 + (Py_i - Py_{i+1})^2}$$

Ecuación 7 Suma integrada del movimiento

El valor de dispersión se calcula mediante la distancia euclídea del punto inicial y final del recorrido realizado por el C. elegans, es decir la norma de la diferencia del primer valor y el último de la matriz de predicción, así como se describe en la **ecuación 8**.

$$D = \sqrt{(Px_1 - Px_n)^2 + (Py_1 - Py_n)^2}$$

Ecuación 8 Valor de dispersión del movimiento

6. Resultados

6.1. Cálculo del error promedio

Para hallar el error medio en cada trayectoria se ha utilizado la distancia euclídea entre los valores de predicción y los valores reales de posición, como se puede apreciar en la **ecuación 9**, donde n es el número total de imágenes procesadas. Los valores reales de posición se han obtenido utilizando la herramienta de “connectedComponents” de openCV en las respectivas imágenes de segmentación más la corroboración visual de las imágenes sin segmentación.

$$E_i = \frac{\sum_{i=1}^n \sqrt{(Px_medido_i - Px_estimado_i)^2 + (Py_medido_i - Py_estimado_i)^2}}{n}$$

Ecuación 9 Error medio en cada trayectoria

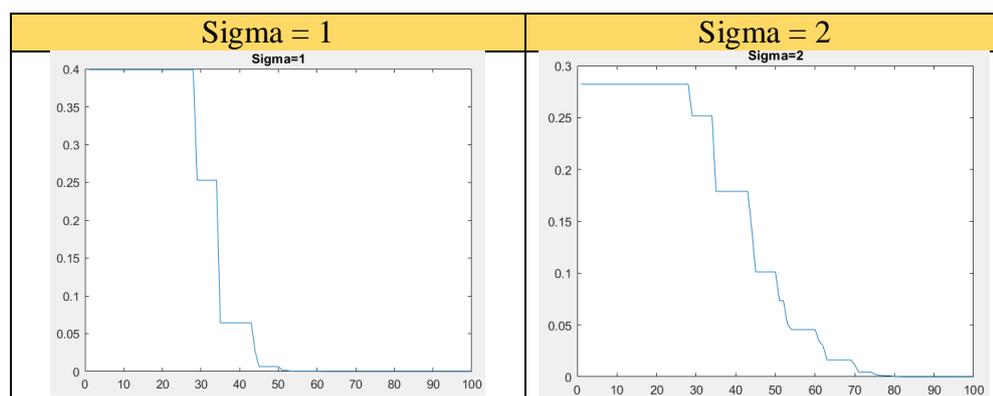
El cálculo del error total de todas las trayectorias se realiza sumando todos los errores medios de cada trayectoria y dividiéndolos para el total de trayectorias procesadas, así como se puede apreciar en la **ecuación 10**, donde m es el número total de trayectorias encontradas.

$$E_{total} = \frac{\sum_{j=1}^m E_j}{m}$$

Ecuación 10 Error medio total

6.2. Evaluación y resultados con diferentes valores de sigma

La siguiente tabla muestra la diferencia de valores de sigma, entre mayor sea su valor este ponderará con valores diferentes de cero a las partículas que se encuentren en un rango más grande, mientras que si su valor es pequeño la evaluación de las partículas y predicción va ser más ajustada y precisa.



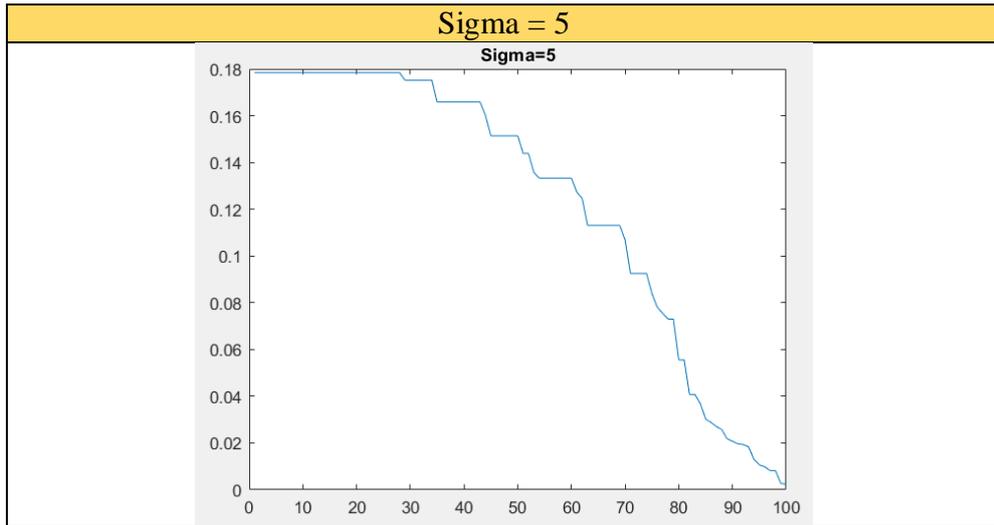


Tabla 2 Comparación de gráficas de diferentes valores de sigma

	Sigma =1	Sigma =2	Sigma =5
Modelo 1	2,2690	3,1381	2,7200
Modelo 2	3,8483	4,0308	4,6532
Modelo 3	1,6169	2,9508	3,4223
Modelo 4	2,6820	3,2660	3,6571

Tabla 3 Comparación de diferentes sigma en los modelos

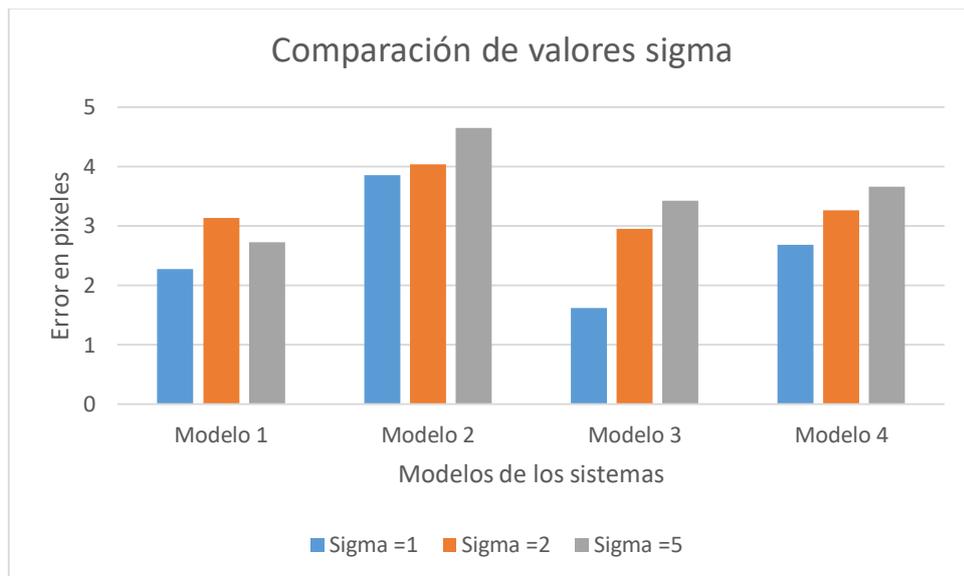


Figura 47 Comparación de diferentes sigma en los modelos

6.3. Evaluación de modelos y comparación de errores en trayectorias individuales con sigma igual a 1

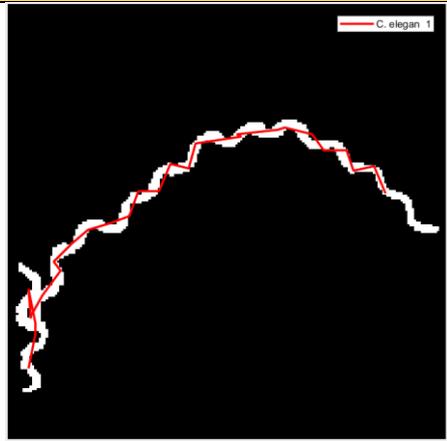
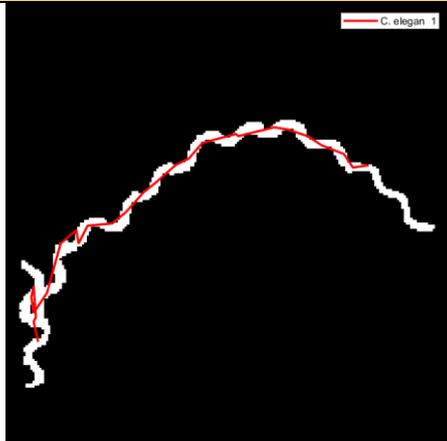
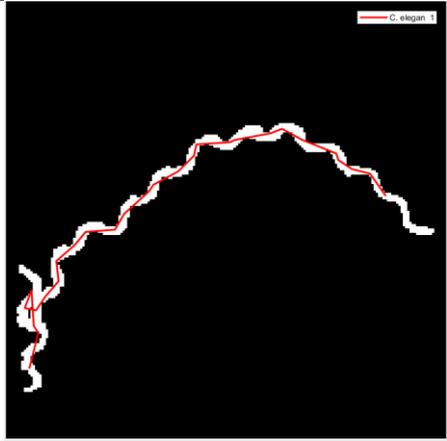
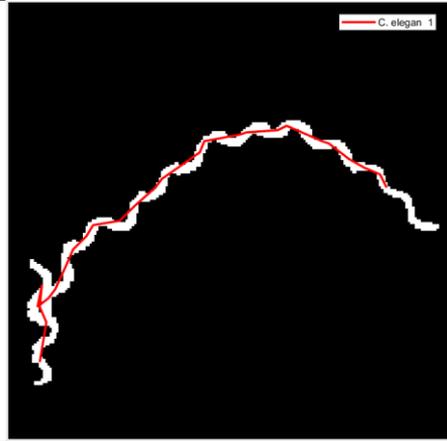
Modelo #1	Modelo #2
	
Error promedio = 2,2690	Error promedio = 3,8483
Modelo #3	Modelo #4
	
Error promedio = 1,6169	Error promedio = 2,6820

Tabla 4 Comparación de errores en trayectorias individuales

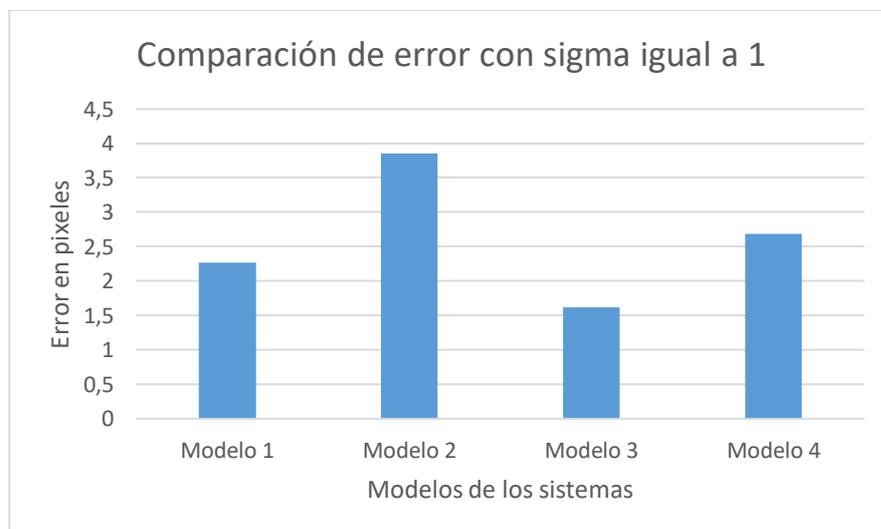


Figura 48 Comparación de error entre modelos con sigma igual a 1

6.4. Evaluación de modelos y comparación de errores en trayectorias intersecantes separadas

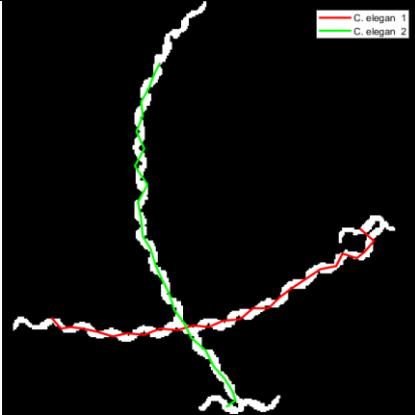
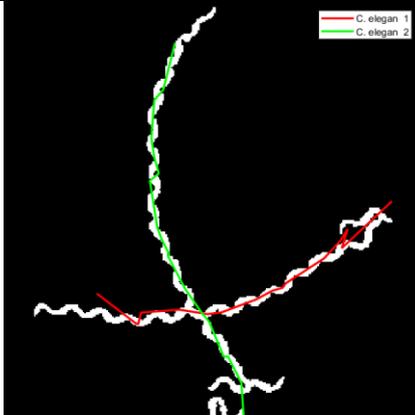
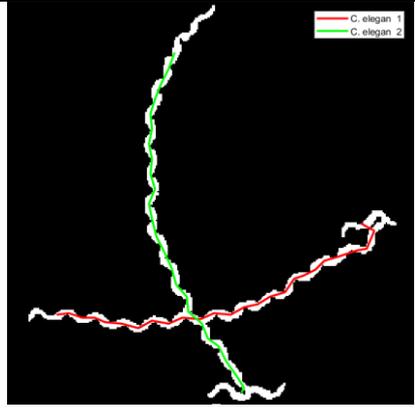
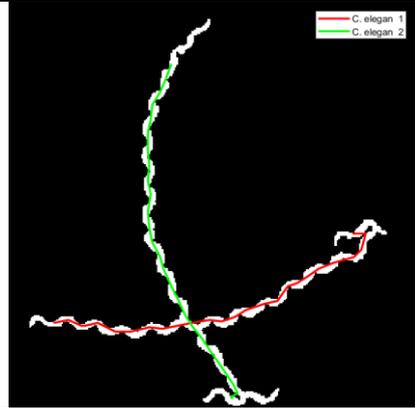
Modelo #1	Modelo #2
	
Error promedio = 3,6046	Error promedio = 8,1517
Modelo #3	Modelo #4
	
Error promedio = 1,4864	Error promedio = 2,6252

Tabla 5 Comparación de errores en trayectorias intersecantes separadas

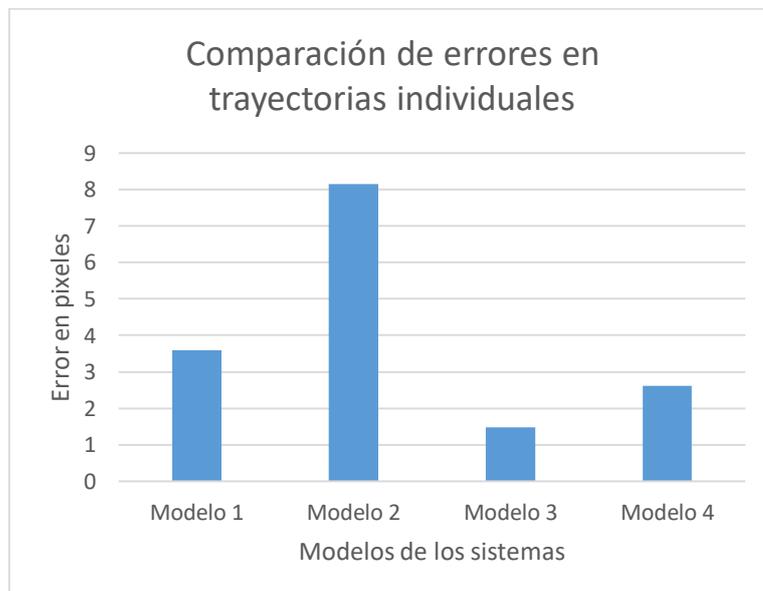


Figura 49 Comparación de errores en trayectorias intersecantes separadas

6.5. Evaluación de modelos y comparación de errores en trayectorias intersecantes unidas

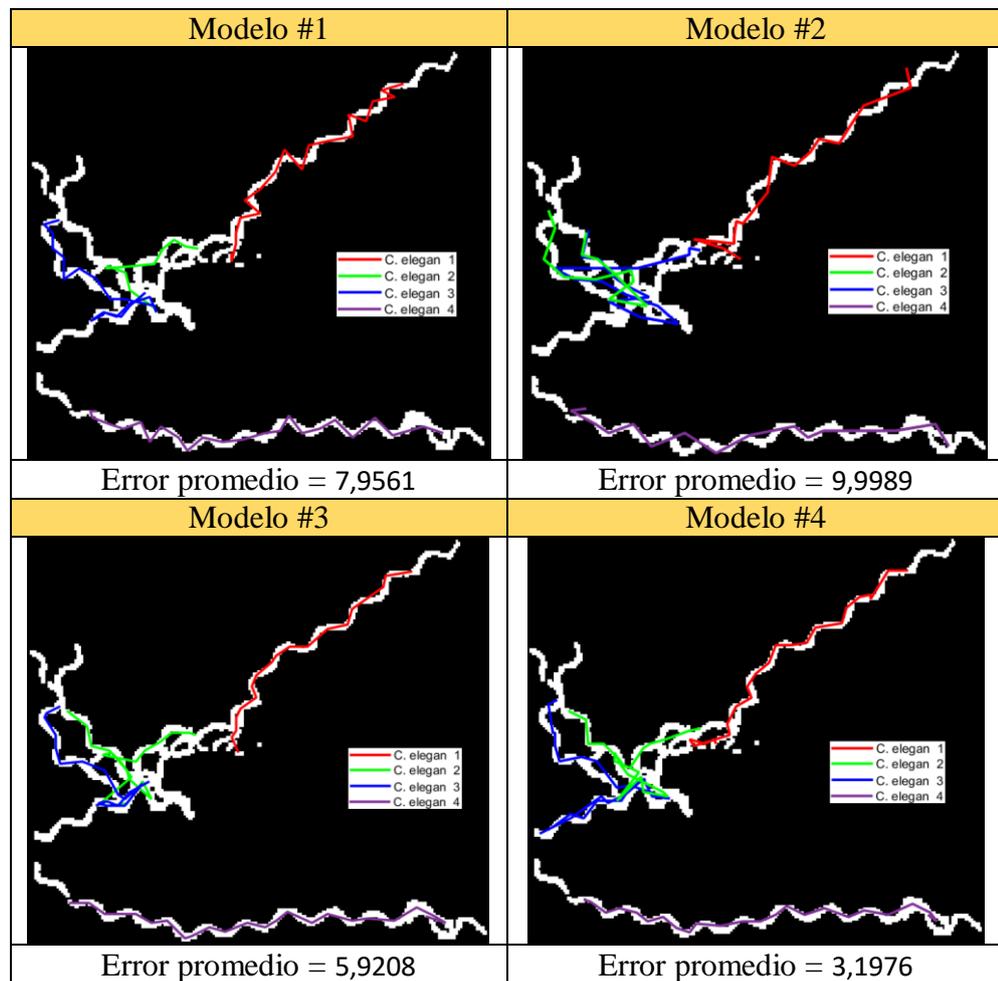


Tabla 6 Comparación de errores en trayectorias intersecantes unidas

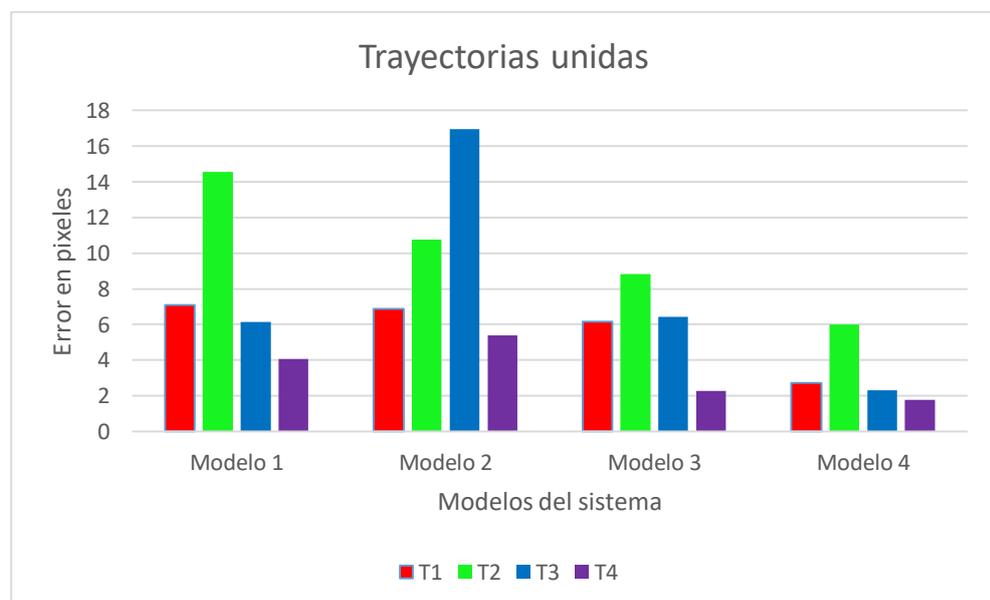


Figura 50 Comparación de errores en trayectorias intersecantes unidas

Si bien el modelo 3 obtiene buenos resultados, luego de algunas pruebas se obtiene 4 diferentes tipos de recorridos, siendo uno de estos las trayectorias correctas.

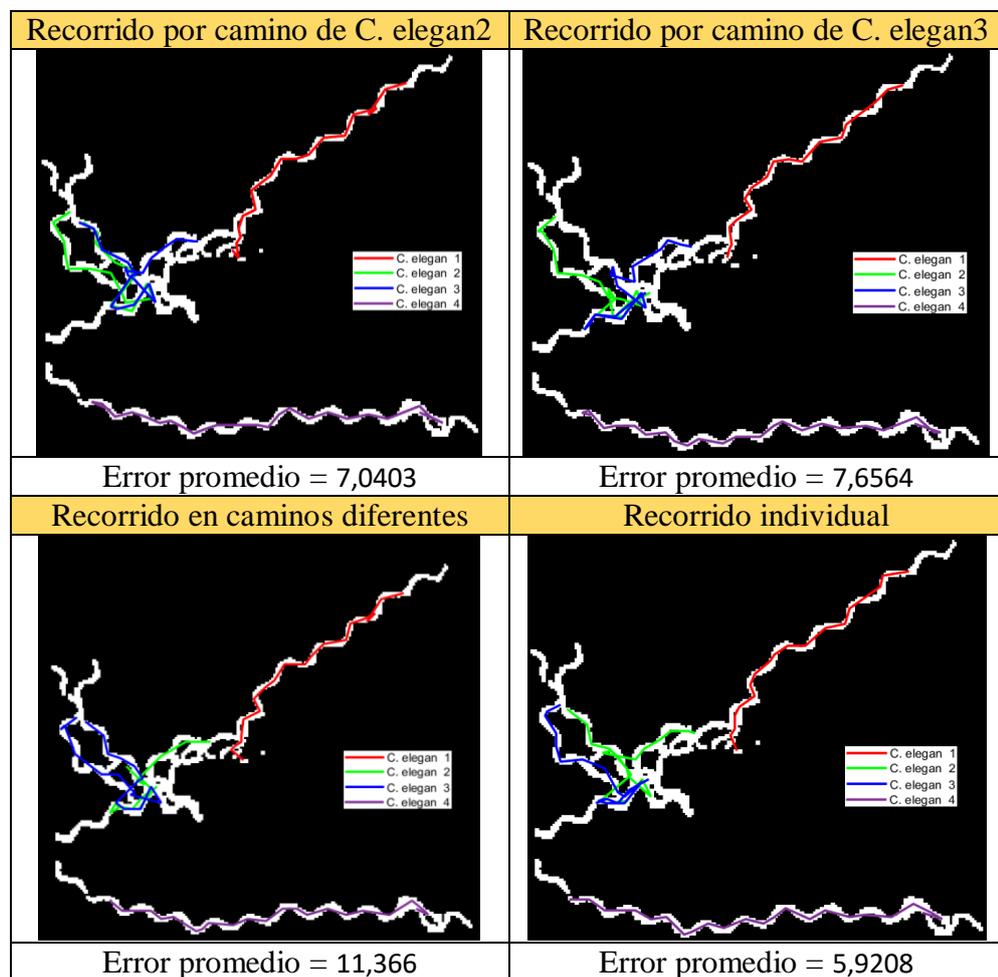


Tabla 7 Comparación de recorridos del modelo 3

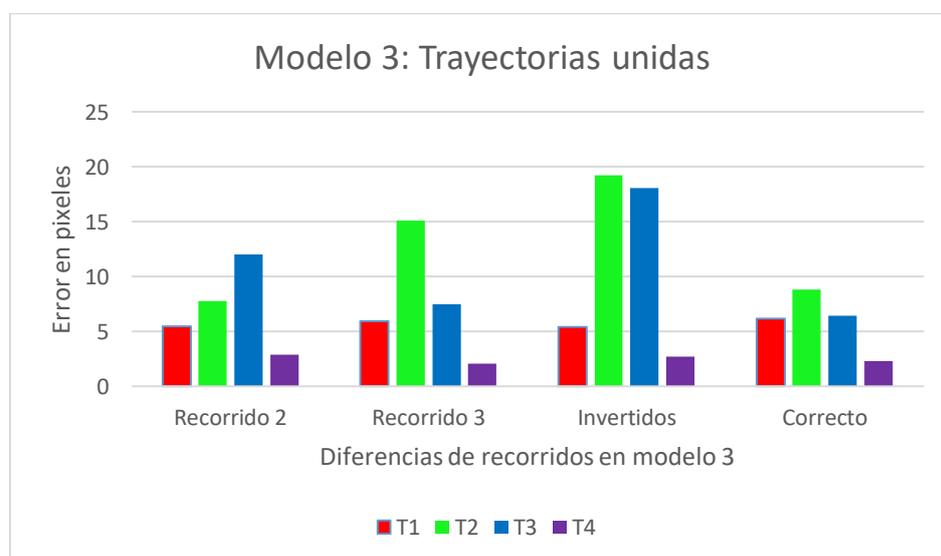


Figura 51 Comparación de recorridos del modelo 3

7. Desarrollo de la interfaz gráfica

La funcionalidad de la interfaz gráfica de usuario, es crear un entorno visual donde el usuario pueda trabajar con el programa de reconocimientos de trayectorias, visualizando su desarrollo durante todas las etapas mediante mensajes por pantallas e imágenes las cuales indican el estado en el que se encuentra el programa, a su vez el usuario puede trabajar también con la base de datos, accediendo a la información presente en diferentes tablas, también tendrá la posibilidad de visualizar la información mediante la generación de reporte automáticos, teniendo la posibilidad de crear o eliminar pruebas realizadas, así como también la posibilidad de poder crear 3 tipos de reportes con la información más relevante para el usuario.

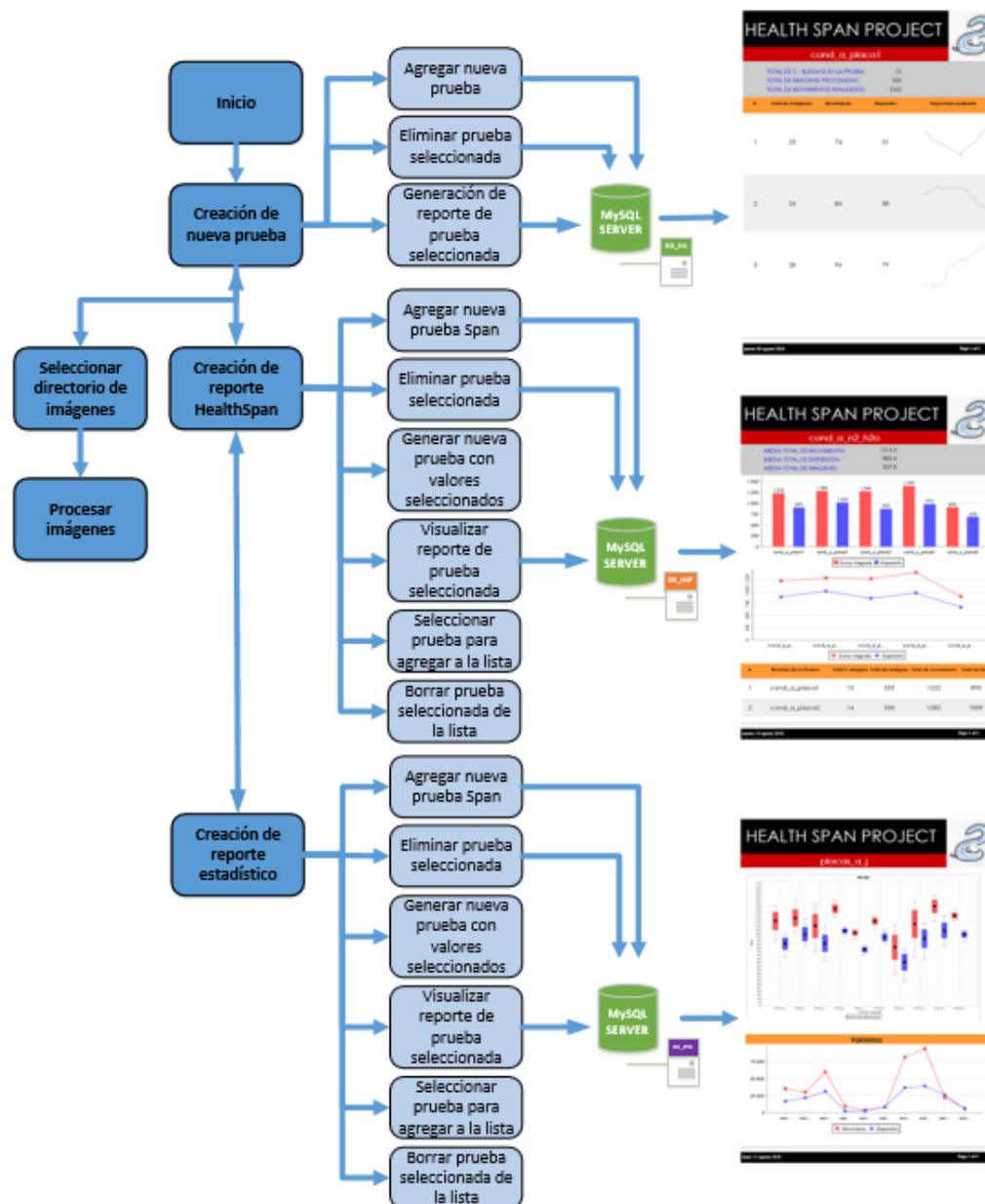


Figura 52 Diagrama de flujo de la interfaz gráfica de usuario

La interfaz gráfica cuenta con seis pantallas principales, las cuales permiten al usuario poder crear nuevas pruebas y acceder a toda la información guardada en la base de datos, así como generar los diferentes reportes de forma automática, estas pantallas se detallan a continuación.

7.1. Descripción de la pantalla de inicio

Botón salir. - Termina la ejecución de la aplicación y cierra la pantalla.

Botón Continuar. - Cierra la pantalla actual y abre la siguiente pantalla (Creación de nueva prueba).



Figura 53 Pantalla de inicio

7.2. Descripción de la pantalla: Nueva prueba

Botón salir. - Termina la ejecución de la aplicación y cierra la pantalla.

Botón agregar. – Captura el texto actual del campo de texto (TextField) para crear una nueva prueba, si el nombre de la nueva prueba está en blanco o si es igual a un nombre de prueba previamente creado o si el usuario ingresa algún símbolo, espacio entre palabras o el texto comienza con números aparecerá un mensaje de información indicando un error por haber cometido alguno de estos eventos.

Botón eliminar. – Permite eliminar una prueba creada de la lista de pruebas, en caso de no haber seleccionado una prueba de la lista aparecerá un mensaje de información indicando un error por no haber seleccionado ninguna prueba.

Botón siguiente. - Cierra la pantalla actual y abre la siguiente pantalla (Seleccionar directorio de imágenes), pasando como parámetro el nombre de la prueba creada, en caso de no haber agregado un nombre a la lista de pruebas aparecerá un mensaje de información indicando un error por no haber creado una prueba nueva.

Botón reporte. – Una vez seleccionado una prueba de la lista de pruebas se generará el reporte de la misma accediendo a la base de datos donde se encuentran los datos previamente almacenados luego de haber concluido la prueba, en caso de no haber seleccionado ninguna prueba y haber pulsado el botón aparecerá un mensaje de información indicando un error por no haber seleccionado ninguna prueba.

Botón H. SPAN. - Cierra la pantalla actual y abre la pantalla de creación de reporte HEALTHSPAN.

Botón R. STD. - Cierra la pantalla actual y abre la pantalla de creación de reporte estadístico.



Figura 54 Pantalla nueva prueba

7.2.1. Descripción de la pantalla: Seleccionar directorio de imágenes.

Botón salir. - Termina la ejecución de la aplicación y cierra la pantalla.

Botón atrás. – Cierra la pantalla actual y abre la pantalla NUEVA PRUEBA.

Botón seleccionar. – Abre una ventana del explorador de carpetas del sistema operativo, donde se podrá buscar el directorio donde se encuentran las imágenes para procesar, una vez seleccionado el directorio esta dirección se guardará en una variable tipo String.

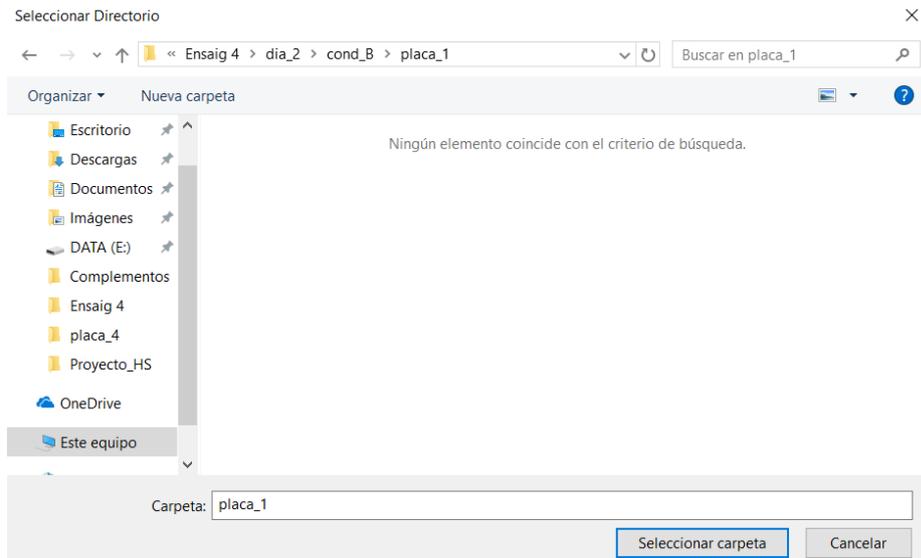


Figura 55 Directorio de imágenes

Botón siguiente. - Cierra la pantalla actual y abre la siguiente pantalla (PROCESAR IMÁGENES), pasando como parámetros el nombre de la prueba, la ruta de las imágenes, el tamaño de imágenes a procesar, y un arreglo tipo String con todos los nombres de las imágenes a procesar, en caso de no haber seleccionado la carpeta de imágenes aparecerá un mensaje de información indicando un error seleccionado una ruta.



Figura 56 Pantalla seleccionar directorio

7.2.2. Descripción de la pantalla: Procesamiento de imágenes

Botón salir. - Termina la ejecución de la aplicación y cierra la pantalla.

Botón inicio. – Cierra la pantalla actual y abre la pantalla NUEVA PRUEBA.

Esta pantalla cuenta con un área de texto el cual va indicando el estado actual del procesamiento de imágenes, y un visualizador de imágenes el cual proporciona de forma visual el recorrido realizado por el C. elegans luego de su procesamiento.



Figura 57 Pantalla procesamiento de imágenes

7.3. Descripción de la pantalla: Prueba HealthSpan

Botón atrás. – Cierra la pantalla actual y abre la pantalla NUEVA PRUEBA.

Botón agregar. – Captura el texto actual del campo de texto (TextField) para crear una nueva prueba HEALTHSPAN, si el nombre de la nueva prueba está en blanco o si es igual a un nombre de prueba previamente creado o si el usuario ingresa algún símbolo, espacio entre palabras o el texto comienza con números aparecerá un mensaje de información indicando un error por haber cometido alguno de estos eventos.

Botón eliminar. – Permite eliminar una prueba creada de la lista de pruebas HEALTHSPAN, en caso de no haber seleccionado una prueba de la lista aparecerá un mensaje de información indicando un error por no haber seleccionado ninguna prueba.

Botón generar. – Procesa la información de los reportes añadidos a la lista procesar y los ingresa en la nueva prueba HEALTHSPAN creada, en caso de

no haber creado la nueva prueba o no haber elementos en la lista para procesar aparecerá un mensaje de información indicando un error por no haber ingresado ningún elemento.

Botón visualizar. – Una vez seleccionado una prueba de la lista de pruebas BD SPAN se generará el reporte de la misma accediendo a la base de datos donde se encuentran los datos previamente almacenados luego de haber concluido la prueba, en caso de no haber seleccionado ninguna prueba y haber pulsado el botón aparecerá un mensaje de información indicando un error por no haber seleccionado ninguna prueba.

Botón seleccionar. – Selecciona una prueba de la lista de pruebas y la agrega a la lista para procesar, en caso de no haber seleccionado una prueba de la lista aparecerá un mensaje de información indicando un error por no haber seleccionado ninguna prueba.

Botón borrar. – Permite eliminar de la lista para procesar una prueba seleccionada, en caso de no haber seleccionado una prueba de la lista aparecerá un mensaje de información indicando un error por no haber seleccionado ninguna prueba.



Figura 58 Pantalla prueba healthspan

7.4. Descripción de la pantalla: Estadística de pruebas

Esta pantalla tiene los mismos botones y funciones que la pantalla anterior con la diferencia que usa la información generada en dicha pantalla para calcular la estadística de las pruebas agregadas a la lista para procesar.



Figura 59 Pantalla estadística de pruebas

8. Modelos de reportes generados

8.1. Modelo de reporte para pruebas individuales

Luego de haber concluido satisfactoriamente el procesamiento de imágenes, los datos de la prueba se almacenan en una base de datos con el nombre ingresado para la prueba en la pantalla NUEVA PRUEBA, estos datos son:

- Un gráfico de líneas donde se visualiza el recorrido realizado (2).
- El valor de dispersión el cual es la diferencia del punto inicial y el punto final del recorrido (3).
- Movimiento realizado durante la trayectoria, el cual es la suma integrada de la distancia euclídea de cada desplazamiento realizado (4).
- Total de imágenes donde el C. elegans realizó un movimiento (5).

El reporte cuenta también con un resumen del procesamiento realizado, el total de C. elegans encontrados, la sumatoria de todas las imágenes y el total de movimiento realizado (1).

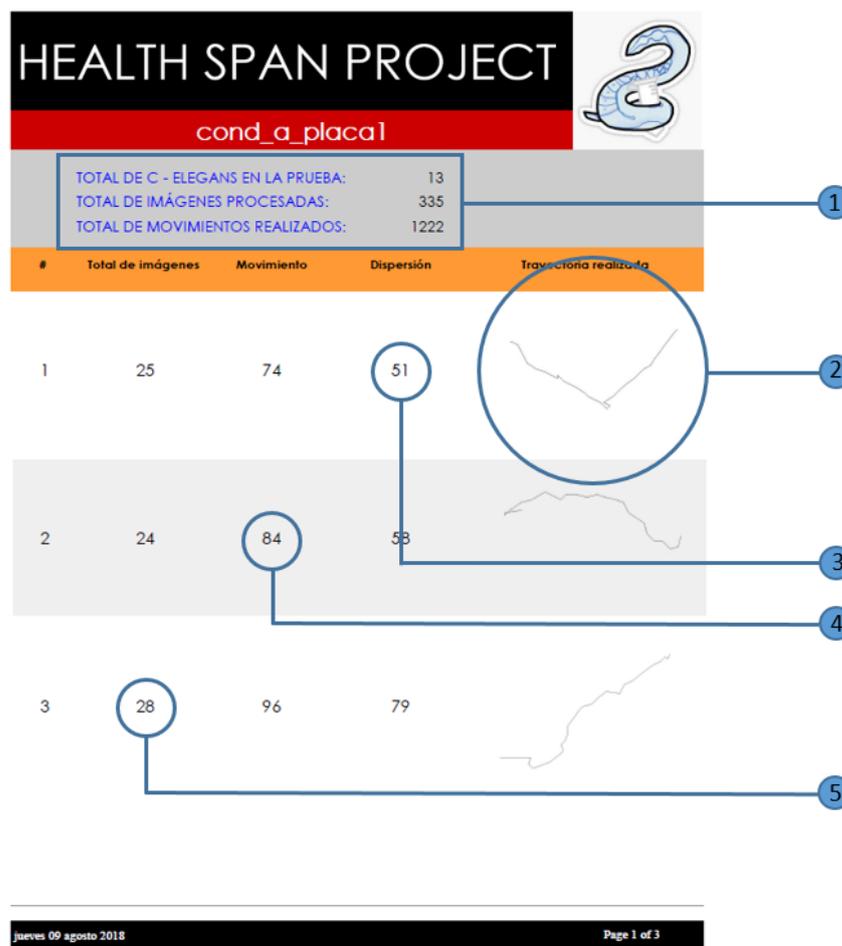


Figura 60 Modelo de reporte para pruebas individuales

8.2. Modelo de reporte de conjunto de pruebas iguales

El reporte a generar se realiza con la información de las pruebas agregadas a la lista para procesar en la pantalla Prueba HealthSpan del capítulo 7, los datos procesados se almacenan en una nueva tabla de la base de datos diferente de los datos de las pruebas, los datos almacenados son:

- El nombre de la prueba seleccionada, el cual corresponde al nombre del reporte creado luego del procesamiento de las imágenes.
- El total de imágenes, el cual es la sumatoria del total de imágenes procesadas para cada trayectoria realizada por el C. elegans
- El total de movimiento, el cual es la sumatoria del total de movimientos realizados de las trayectorias individuales realizadas por los C. elegans.

El reporte cuenta con toda la información utilizada de las pruebas agregadas (4), más gráficos de barras y de líneas (2), (3) los cuales muestra de forma visual los resultados obtenidos de cada prueba seleccionada procesada. Al igual que el reporte anterior cuenta con un resumen de los datos procesados (1).

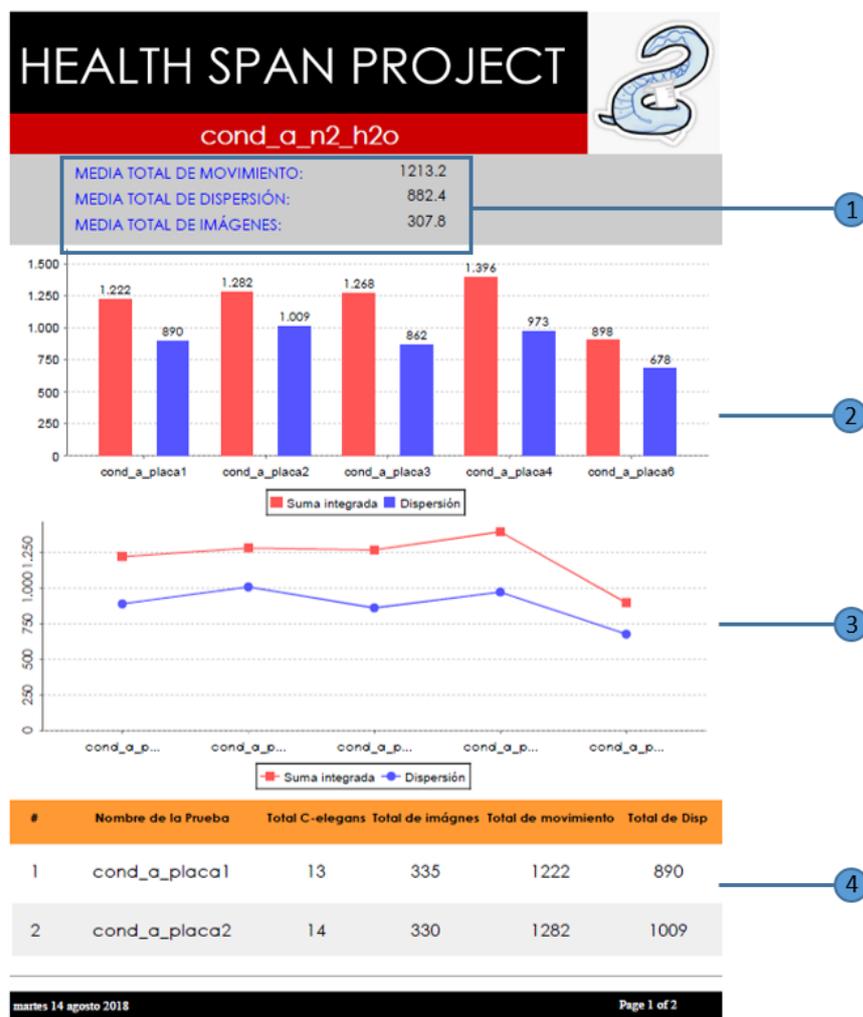


Figura 61 Modelo de reporte para pruebas iguales

8.3. Modelo de reporte estadístico para evaluación de diferentes pruebas

Este reporte utiliza la información obtenida del reporte anterior (conjunto de pruebas iguales), para generar un gráfico de cajas el cual muestra de forma gráfica la media, el máximo y mínimo de todos los conjuntos de pruebas, lo que permite analizar los datos de cada conjunto de prueba y compararlos con los demás, esto ayuda a ver qué tan separados están los datos de la prueba y de igual manera comparar sus distribuciones (1).

La varianza se puede visualizar en el reporte mediante una gráfica de líneas, esta gráfica permite observar que tan separados se encuentran los datos con respecto al promedio de cada conjunto de pruebas (2).

A continuación de las gráficas estadísticas se presenta en forma de tablas la información de cada conjunto de pruebas para el movimiento realizado (3) y para el valor de dispersión (4) respectivamente.

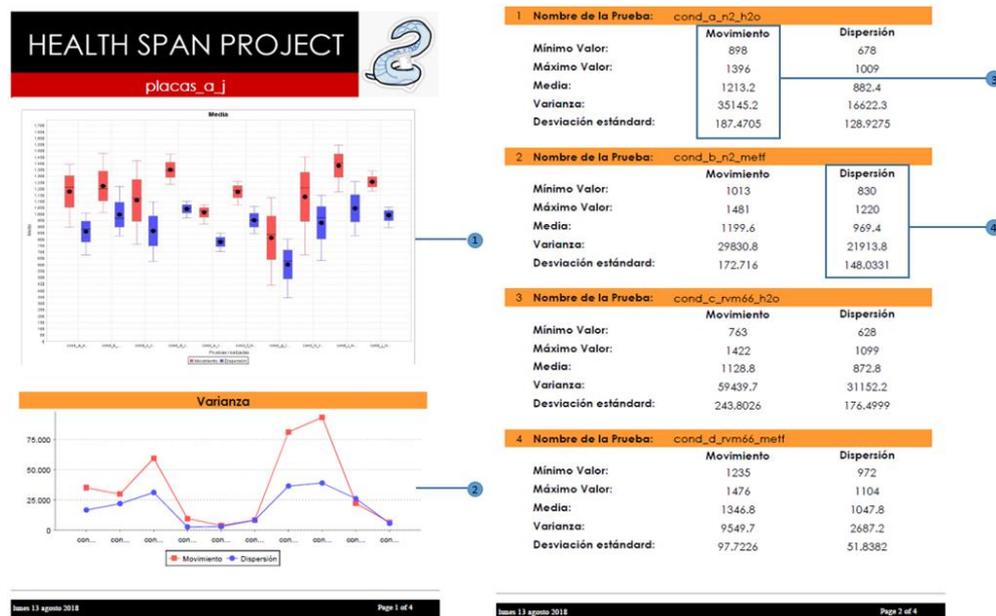


Figura 62 Modelo de reporte para pruebas grupales

9. Conclusiones y trabajos futuros

9.1. Conclusiones

El filtro de partículas es un buen predictor para encontrar la posición siguiente, pero este está sujeto a diferentes errores que hay que tomar en cuenta al momento de utilizarlo, tener un número elevado de partículas ayuda bastante en trayectorias individuales, pero en trayectorias donde hay colisiones provoca un seguimiento de trayectoria erróneo, y caso contrario tener pocas partículas ayuda un poco en trayectorias intersecantes pero en trayectorias separadas existe la posibilidad de no seguir el objetivo.

Luego de analizar algunos modelos que describen el movimiento realizado por los *C. elegans* se ha llegado a la conclusión que estos se desplazan de forma senoidal, pero el modelo puede evolucionar conforme más información sea obtenida, esto permitirá mejorar el modelo propuesto y disminuir los errores de modelado existentes.

Utilizar la diferencia de imágenes con un valor de umbral bajo, más la ayuda de operaciones matemáticas básicas resulta un buen método para encontrar la segmentación de las trayectorias durante las n trayectorias, eliminando todo el ruido presente.

La implementación de una interfaz gráfica proporciona una ayuda excepcional y una herramienta más amigable al usuario, permitiéndole a este controlar todo el sistema de visión, base de datos y generación de reportes con la utilización de algunas pantallas.

9.2. Trabajos futuros

Extender el proceso de seguimiento de trayectoria a un sistema multi-variable con un mayor número de individuos por muestra.

Revisar y procesar más imágenes para poder definir modelos más complejos para la predicción y estimación de la posición.

Encontrar mejoras en la predicción con la utilización de nuevas herramientas para el procesamiento de imágenes, herramientas que hagan más robusto y preciso al sistema de detección.

Incorporar la utilización de redes neuronales temporales para estimar comportamientos o eventos como paradas por colisión, alimentación o reproducción.

Utilizar la metodología estudiada para analizar comportamientos en otras especies animales o insectos y considerar la posibilidad de un enfoque orientado en el campo de la seguridad industrial o seguridad de personas en general.

10. Bibliografía

- A.I. Yashin, S. J. (2014). *Aging and Health - A Systems Biology Perspective*. New Orleans, La: Scientific Publishers.
- Acton, P. Q. (2012). *Issues in Life Sciences: Cellular Biology: 2011 Edition*. Atlanta, Georgia: ScholarlyEditions.
- Anders Olsen, M. S. (2016). *Ageing: Lessons from C. elegans*. New York, USA: Springer.
- Baggio, D. L. (2015). *OpenCV 3.0 Computer Vision with Java*. Birmingham, UK: Packt Publishing Ltd.
- Benloch, J. V. (1995, Octubre). Colour segmentation techniques for detecting weed patches in cereal crops. In Proc. of Fourth Workshop on Robotics in Agriculture and the Food-Industry.
- Berti, E. S. (2012b). Human-Robot Interaction and Tracking Using low cost 3D Vision Systems.
- Candy, J. V. (2016). *Bayesian Signal Processing: Classical, Modern, and Particle Filtering Methods*. New Jersey, USA: John Wiley & Sons.
- E.M. Berti, A. S. (2012a). Kalman filter for tracking robotic arms using low cost 3D vision systems. The Fifth International Conference on Advances in Computer-Human Interactions. Valencia.
- Gail Anderson, P. A. (2009). *Essential JavaFX*. Indiana, USA: Pearson Education.
- Grinev, S. (2018). *Mastering JavaFX 10: Build advanced and visually stunning Java applications*. Birmingham, UK: Packt Publishing Ltd.
- Heffelfinger, D. R. (2006). *JasperReports for Java Developers: Create, Design, Format, and Export*. Birmingham, UK: Packt Publishing Ltd.
- Mark Heckler, G. G. (2014). *JavaFX 8: Introduction by Example*. New York, USA: Apress.
- Mark Matthews, J. C. (2003). *MySQL and Java Developer's Guide*. Indiana, USA: John Wiley & Sons.
- Martinez, E. N. (2017b). Optimized 4D-DPM for Pose Estimation on RGBD Channels using polisphere models. In Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. Portugal.
- Martinez, E. S. (2016). Human Pose Estimation for RGBD Imagery with Multi-Channel Mixture of Parts and Kinematic Constraints.
- Martinez, E. S. (2017a). 4D-DPM model for pose estimation using Kalman filter constraints. Int. J. Adv. Robot. Syst.
- Martinez-Berti, E. S.-S.-V. (2017). *Dual quaternions as constraints in 4d-dpm models for pose estimation*. Switzerland.

- Modrzyk, N. (2018). *Java Image Processing Recipes: With OpenCV and JVM*. New York, USA: Apress.
- Pradeeka Seneviratne, J. S. (2017). *Raspberry Pi 3 Projects for Java Programmers*. Birmingham, UK: Packt Publishing Ltd.
- Ristic, B. (2012). *Particle Filters for Random Set Models*. New York, USA: Springer.
- Sánchez, A. &. (2000). *SEGUIMIENTO VISUAL DE OBJETOS UTILIZANDO TÉCNICAS DE PREDICCIÓN, XXI Jornadas de Automática*.
- Sánchez, A. J. (2008). Control of ham salting by using image segmentation. *Food Control*.
- Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. New Jersey, USA: John Wiley & Sons.
- Teodor Danciu, L. C. (2007). *The Definitive Guide to JasperReports*. New York, USA: Apress.
- Theodore Friedmann, J. C. (2015). *Advances in Genetics, Volume 90*. San Diego, USA: Academic Press.