Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Fusión de Múltiples Imágenes en Microscopía

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Max Quirós López

Cartago, Junio 2011

INSTITUTO TECNOLOGICO DE COSTA RICA

ESCUELA DE INGENIERIA ELECTRONICA
PROYECTO DE GRADUACIÓN
TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

Ing. Eduardo Interiano
Profesor lector

Ing. Pablo Alvarado Moya
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 21 de Junio, 2011

Declaratoria de autenticidad

Declaro que el presente proyecto de graduación ha sido realizado enteramente por mi persona, introduciendo conocimientos propios y utilizando la literatura referente al tema.

En los casos en los que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 21/06/2011

Max Quirós López
Céd: 1-1153-0204

Resumen

El presente proyecto forma parte de una herramienta de visión por computadora que permite reducir el tiempo de trabajo y estudio de los nematodos. Estos organismos microscópicos viven en el suelo y en el agua, algunos de estos seres se nutren de plantas y sirven a su vez de alimento para otras especies. Los investigadores tratan de mejorar el control de las plagas de nematodos mediante el control biológico, sin embargo, para esto se necesita realizar el análisis de sus características y su comportamiento en las cosechas.

Para estos estudios, los nematólogos deben revisar muestras de nematodos haciendo uso del microscopio, sin embargo, esto les puede tomar desde 15 minutos hasta varias horas, pues no pueden observar la muestra completa, sino solamente sección a sección de la misma, lo cual ocasiona que exista un mayor riesgo de cometer errores y que se requieran más horas de trabajo.

En este proyecto se propone una solución haciendo uso de la teoría del procesamiento digital de imágenes y de las ventajas que brinda. Obteniendo imágenes digitales de cada segmento que se desea analizar de la muestra de nematodos, se implementa un algoritmo que permite crear una sola imagen con la escena completa.

Palabras claves: nematodos, procesamiento digital de imágenes, composición de imágenes.

Abstract

This project is part of a computer vision tool that allows improving the time required to study nematodes. These organisms are microscopic beings that live in the ground and water. Some of them feed on plants and are food for other species. Researchers seek to improve the control of nematode pests through biological control; however, this requires analysis of their characteristics and their behavior on crops.

Nematologists should review nematode samples using a microscope and it can takes time from fifteen minutes to a couple of hours, because they cannot analyze a complete sample scene, only section to section of it, which causes an increased risk of errors and required work hours.

This project presents a solution to the problem above using the theory of digital image processing and the benefits that it provides. Getting digital images of each segment of the sample of nematodes an algorithm can be implemented that allows the creation of a single image with the entire scene.

Keywords: nematodes, digital image processing, image composition, mosaicing.

Dedicatoria

A mi madre, Marielos López Ramírez, por su apoyo incondicional y la fuerza que me ha dado para poder culminar esta carrera.

A mi prometida, Halsyin León Solano, por apoyarme en todo momento y por sus buenas ideas.

A mi tío, Francisco López Ramírez, por la inspiración que me brindó desde que era niño para animarme a estudiar esta carrera.

Agradecimientos

Al Dr. Ing. Pablo Alvarado Moya por permitirme realizar este proyecto y por toda la ayuda brindada durante el proceso del mismo.

A mi jefe, Ricardo Delgado López-Calleja, por el apoyo y los permisos que me brindó en mi tiempo laboral para poder trabajar en este proyecto.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1: Introducción

1.1 Aspectos generales del proyecto

Los organismos en las imágenes que se utilizarán como muestras de estudio en el proyecto son los *nematodos* o *larvas del suelo*, que son gusanos cilíndricos no segmentados que viven en grandes cantidades en la tierra y en el agua [3]. Algunas de las especies existentes de estos microorganismos son parásitas de las plantas, mientras que otras sirven de alimento para otros seres vivos.

Todos los nematodos originan huevos y pasan por una serie de estados larvales antes de llegar a la forma adulta. Al alimentarse de las plantas, disminuyen el vigor de éstas y provocan lesiones, pudrición, deformación, agallas y nódulos en las raíces. Los cultivos infestados se ven disparejos, y sectores definidos de plantas afectados por el enanismo [3].

Como estos seres son organismos de tamaño del orden de micrómetros, los nematólogos hacen su estudio a través del microscopio y para ello deben tomar muestras y analizarlas con el fin de controlar las plagas que afectan los cultivos. Una sola muestra es capturada en distintas fotografías para obtener con más detalle las características de estudio de los nematodos. Además, deben tomarse varias imágenes consecutivas, ya que una sola no podría comprender el nivel de detalle que los nematólogos necesitan para su correcto estudio. Es por esto que las técnicas de fusión de imágenes son tan útiles en el estudio de los nematodos, pues permiten crear un mosaico final con el detalle deseado y así realizar el estudio correspondiente.

Se conoce como *fusión de imágenes* o *composición (mosaicing)* al campo de desarrollo dentro del procesamiento digital de imágenes en el cual un cuadro

es compuesto a partir de varias fotografías o dibujos que tienen una relación entre sí.

Para poder realizar una adecuada fusión de las imágenes, primero, se necesitan alinear las muestras visuales automáticamente. Esta parte del proceso recibe el nombre de *registro de imágenes* y se encarga de ubicar en un mismo plano o sistema de coordenadas los cuadros con el fin de que luego éstos puedan ser unidos. Una vez que han sido registrados, éstos pueden ser ensamblados a través de un algoritmo que permita diferenciar las zonas de traslape y compensar las diferencias que existan entre las uniones, de tal forma que, el diseño final no parezca un simple agrupamiento fotográfico, sino una imagen completa como si hubiera sido tomada en un solo instante[1].

El problema en lo que respecta a la fusión de imágenes radica en una combinación de tres inconvenientes:

- La corrección de deformaciones geométricas a partir de los datos de una imagen.
- La correcta unión de las imágenes.
- La eliminación de las uniones existentes entre cada imagen.

Los algoritmos de software típicamente usados para unir las muestras visuales conservan una secuencia de cuadros, y luego realizan una composición con todas las imágenes registradas en un mosaico final.

Para esto, se necesita tener un método que registre los pares de imágenes en relación con transformadas simples. Una vez hecho esto, aparecen nuevos problemas que dificultan la formación del ensamble final, pues pueden existir errores que intervengan entre la toma de un cuadro y el siguiente complicando el resultado de la fusión.

Ahora bien, los cuadros necesarios para terminar la composición del diseño deben tener una secuencia y un orden para que las zonas de traslape de una de las imágenes coincida con el fragmento siguiente y a su vez éste con el que le precede. Esto ocasiona algunos inconvenientes tales como, el ajuste del origen del sistema de coordenadas del mosaico final, pues dicho origen puede ir determinado por uno de los cuadros o por una mezcla de ambos, dependiendo del movimiento existente de uno hacia el otro y la zona de traslape.

Se han desarrollado varias técnicas para la composición de mosaicos de imágenes. Esta área de investigación será el eje central del presente proyecto, enfocando su aplicación a la fusión de múltiples imágenes en muestras de organismos microscópicos vivos.

1.2 Técnicas para el control de los nematodos

En la actualidad se utilizan los llamados controles químicos en las cosechas para eliminar la proliferación de los nematodos, sin embargo, estos productos pueden dañar los sembradíos y además causar problemas en la salud humana tanto en quienes lo aplican, como en aquellos que consumen dichos productos.

También son utilizadas las técnicas de control biológico, las cuales consisten en la utilización de otros microorganismos que puedan controlar a los nematodos de una manera natural. Para esto, los nematólogos deben hacer un análisis sobre los efectos que distintos agentes puedan tener al entrar en contacto con esta especie. Estos estudios se deben realizar con fotografías tomadas por los mismos microscopios. Estas imágenes deben ser tomadas con detalle para así ser analizadas con la mayor precisión posible. Sin embargo, una muestra de nematodos no puede ser capturada directamente con una sola toma, ya que el nivel de exactitud se pierde. Es por esto que se debe tomar una secuencia de

fotografías y luego utilizar técnicas de fusión para finalmente formar un mosaico general y así obtener los detalles deseados.

La síntesis del problema se da en cómo lograr fusionar imágenes de diferentes secciones de una muestra microscópica considerando que el contenido se encuentra en movimiento y puede haber variaciones de la iluminación entre las capturas de las secciones.

1.3 Composición de múltiples imágenes

Se propone crear una herramienta que sea capaz de formar un ensamble compuesto a partir de diferentes imágenes obtenidas de una cámara colocada sobre el microscopio.

El movimiento de la cámara ocasiona que la orientación y la zona de traslape varíe entre un cuadro y otro, por lo que se requiere de la alineación automática para la fusión resultante. Se pretenden usar algunos algoritmos de registro de imágenes ya probados que permiten alinear dos cuadros en una imagen final, sin embargo, estas alineaciones generadas deben ser reajustadas con respecto al origen del sistema de coordenadas en cada unión, pues dicho origen cambia con la inclusión de cada nueva fotografía. Es por esto que la creación de este algoritmo de reajuste forma parte en la solución dada al problema de la composición de las múltiples imágenes.

Otro aspecto a tomar en cuenta, es el tiempo que se requiere durante el proceso de la unión de las imágenes, pues una muestra de nematodos puede estar formada por más de cien cuadros, por lo que el análisis de cada uno de ellos y la mezcla final requieren una cantidad de tiempo considerable. Esto debido a que, conforme se va armando la imagen final la cantidad de análisis es mayor y crece de forma polinomial, pues por cada reagrupamiento de los cuadros hay una imagen resultante de mayor tamaño que debe ser analizada durante la

composición del cuadro siguiente. Para esto se pretende realizar un análisis del algoritmo de composición de manera que, se pueda encontrar una forma de reducir el tiempo de procesamiento de las imágenes y así volver más eficiente el trabajo de unirlas.

1.4 Objetivos y estructura del trabajo

La meta del proyecto ha sido fusionar imágenes de una muestra de nematodos vivos con una calidad tal que permita operaciones de identificación y conteo automatizados con la información de la imagen resultante. Esto se logra mediante el diseño e implementación de un algoritmo que realiza la composición de múltiples imágenes, creando un único cuadro resultante que mantiene las características de las fotografías iniciales.

La primera etapa consiste en la evaluación del método propuesto en [1] con el fin de ver qué tan robusto es cuando se tiene una cantidad mayor a dos cuadros una misma muestra de nematodos.

A partir de la evaluación se realizan las modificaciones necesarias con el fin de poder unir múltiples imágenes y presentar el mosaico final compuesto por todo el ensamble de las mismas.

En la última etapa se implementa el algoritmo que registre las imágenes alineadas en un solo marco y muestre el ensamble completo de todos los cuadros.

En la figura 1.1 se muestra un diagrama de bloques del proceso completo. Este proyecto implementa los algoritmos para las dos últimas etapas (ajuste de puntos característicos y registro de las imágenes), pues las primeras dos ya fueron realizadas e implementadas.

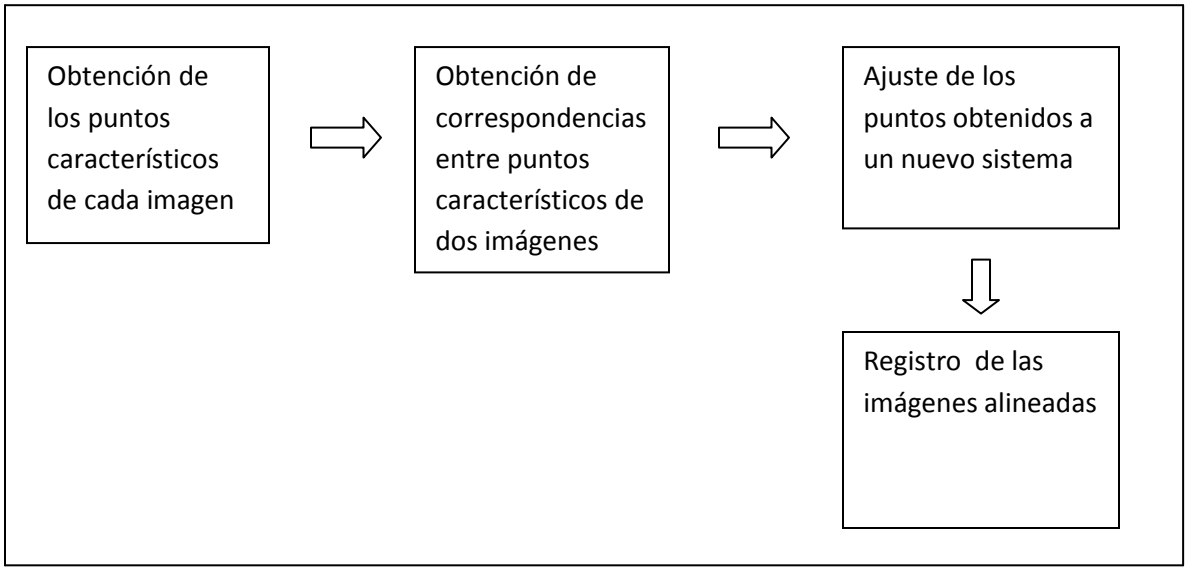

Figura 1.1 Diagrama de bloques general del proceso

El documento se estructura de la siguiente forma: en el capítulo dos se presentan los conceptos teóricos necesarios para la comprensión de la solución propuesta, incluyendo la referencia a algunos métodos existentes para realizar el registro de las imágenes. En el capítulo tres se presenta detalladamente la solución propuesta. En el capítulo cuatro se muestran los resultados obtenidos al implementar la solución. Finalmente, en el capítulo cinco se incluyen las conclusiones derivadas del análisis de la solución y las recomendaciones sugeridas para trabajos posteriores relacionados con este proyecto.

Capítulo 2: Marco Teórico

En este capítulo se presenta primero la teoría de la detección de los llamados puntos de interés y el método utilizado para resolver satisfactoriamente el problema planteado. Además se explica el algoritmo de alineación utilizado en el proyecto.

2.1 Algoritmo de detección de puntos de interés

Los puntos de interés son puntos determinados por características llamativas de una imagen. La vecindad de cada punto de interés puede describirse a través de un vector denominado *vector descriptor*, el cual contiene la información de la imagen en esa vecindad. Buscando similitudes entre los vectores descriptores se pueden asignar correspondencias en los puntos de interés de dos imágenes y luego con esos pares de correspondencias se puede estimar la transformación de similitud entre las dos imágenes. La etapa en donde se aplica la transformación de similitud se conoce como *registro de las imágenes*.

Aquellos puntos que vayan a ser localizados deben ser lo más estables posibles y permanecer invariantes ante cualquier cambio de orientación, rotación o escala de las imágenes para que puedan ser correctamente asociados en dos imágenes diferentes, aunque éstas hayan sido tomadas desde posiciones distintas [6].

Para este proyecto se utilizó una implementación del algoritmo detector Hessiano rápido de puntos de interés y para el cálculo de descriptores, el algoritmo SURF (Speeded Up Robust Features) [5]. El algoritmo SURF presenta un mejoramiento en la velocidad de cálculo en comparación con otros algoritmos usados, incluyendo su similar, el algoritmo SIFT (Scale Invariant Feature Transform) [6]. El SURF reduce la dimensión y la complejidad de los descriptores

obtenidos, pero lo hace de modo que éstos continúen siendo lo suficientemente característicos, significativos e igualmente repetitivos.

2.1.1 Detección de los puntos de interés

El detector hessiano rápido de puntos de interés está basado en una aproximación básica de la matriz Hessiana y utiliza ésta debido a su buen rendimiento en cuanto a velocidad de cálculo y precisión. Sin embargo, éste, al contrario del método empleado por otros detectores, utiliza el determinante de la matriz Hessiana para elegir la posición y la escala. Por lo tanto, dado un punto p = (x,y) de una imagen I, la matriz Hessiana H(p,σ) en el punto p a la escala σ se define como:

$$H(p,\sigma) = \begin{bmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{xy}(p,\sigma) & L_{yy}(p,\sigma) \end{bmatrix} \tag{2.1}$$

Donde $L_{xx}(p,\sigma)$ es la convolución de la derivada parcial de segundo orden de la Gaussiana $\frac{\partial^2}{\partial x^2} g(\sigma)$ de la imagen I en el punto p, lo mismo ocurre para $L_{xy}(p,\sigma)$ y $L_{yy}(p,\sigma)$.

Los espacios de escala son a menudo implementados como pirámides de imágenes, en las que éstas son suavizadas repetidamente con un filtro Gaussiano. A pesar de que éstos filtros son óptimos para el análisis de espacio escala, el detector hessiano utiliza una alternativa llamada *filtros de caja*. Éstos aproximan las derivadas parciales de segundo orden de las Gaussianas y pueden ser evaluados de manera muy rápida usando imágenes integrales, independientemente de su tamaño [6].

La escala más pequeña que se tiene corresponde a un filtro de caja de dimensión 9x9. Las aproximaciones de las derivadas parciales se denotan como D_{xx}, D_{xy} y D_{yy}. El determinante de la matriz Hessiana queda definido como:

$$\det\left(H_{aprox}\right) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (2.2)$$

Debido al uso de filtros de caja e imágenes integrales no es necesario aplicar iterativamente el mismo a la salida de la capa filtrada, sino que se pueden aplicar dichos filtros de cualquier tamaño a la misma velocidad directamente en la imagen original. De modo que el espacio escala es analizado al ir elevando el tamaño del filtro en vez de reducir el tamaño de la imagen.

Finalmente, para la localización de los puntos de interés en la imagen se procede a la eliminación de aquellos puntos que no sean máximos a una vecindad de 3x3x3. Así, el máximo determinante de la matriz Hessiana es interpolado en la escala y espacio de la imagen.

2.1.2 Asignación de la orientación

A cada punto de interés obtenido se le otorga una orientación específica para que el descriptor pueda ser invariable ante la rotación. Se calcula la llamada *Respuesta de Haar* en la dirección *x* e *y* (ver figura 2.1) en un área circular de vecindad de radio 6s alrededor del punto de interés, donde *s* es la escala del punto de interés detectado [2]. Para obtener la respuesta en la dirección *x* e *y* se necesitan únicamente 6 operaciones.

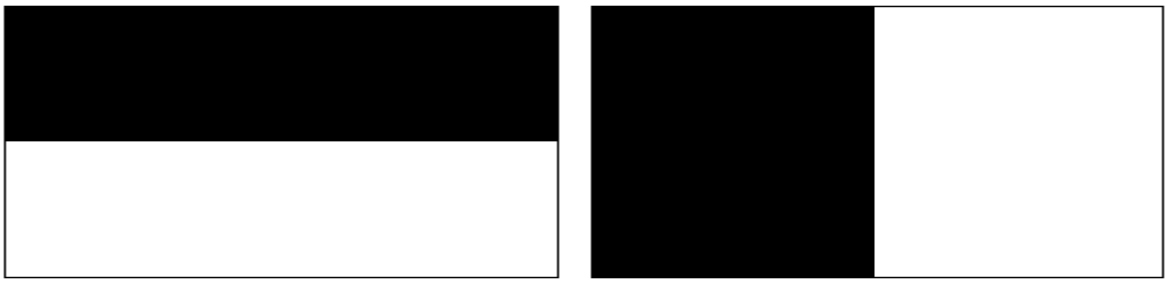

Figura 2.1 Filtros de Haar para el cálculo de las respuestas en la dirección *x* (izquierda) e *y* (derecha). El color negro identifica el valor de -1 y el color blanco del valor de +1

Una vez que las respuestas han sido calculadas, éstas son ponderadas con una Gaussiana $\sigma = 2.5s$ centrada en el punto de interés. Las respuestas se representan como vectores espaciales con la respuesta horizontal en el eje de las abscisas y la respuesta vertical en el eje de las ordenadas. Para la obtención de la orientación dominante todas las respuestas (horizontal y vertical) son sumadas en una ventana de orientación cubriendo un ángulo de $\pi/3$. Las sumas resultantes en cada ventana forman un nuevo vector y el vector de mayor longitud es el que le da la orientación al punto de interés.

2.1.3 Obtención de los descriptores SURF

Para la extracción del descriptor se construye una región cuadrada centrada alrededor del punto de interés (ver figura 2.2). Luego, ésta es reducida en regiones cuadradas más pequeñas de 4x4. Luego se calcula la respuesta de Haar en una muestra de puntos regularmente espaciados 5 x 5.

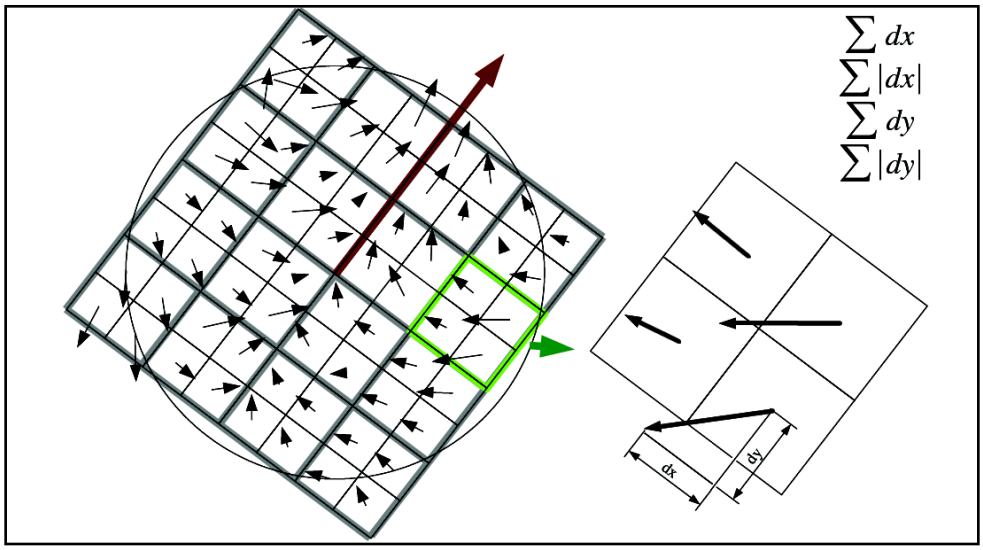

Figura 2.2 Región construida alrededor del punto de interés

La respuesta de Haar en la dirección horizontal es llamada d_x, mientras que en la dirección vertical es llamada d_y. Dichas respuestas se suman en cada

subregión conformando el primer conjunto de valores del vector de características. Ahora bien, con los valores absolutos de las respuestas de d_x y d_y se obtiene la información de la polaridad de los cambios de intensidad. De esta manera, cada subregión tiene como descriptor un vector *v* de dimensión 4 que describe su estructura de intensidad.

$$\mathrm{v} = \left(\sum \mathrm{d_x}, \sum \mathrm{d_y}, \left|\sum \mathrm{d_x}\right|, \left|\sum \mathrm{d_y}\right|\right) \tag{2.3}$$

2.1.4 Detección de puntos correspondientes

La última etapa consiste en detectar aquellos puntos de interés correspondientes hallados en dos imágenes consecutivas. A estos puntos se les llama *puntos de control*. Cada descriptor significativo en la imagen 1 es comparado con los descriptores de la imagen 2 y el emparejamiento es dado en caso de que la distancia relativa entre ambos sea menor a 0,7 veces la distancia respecto al segundo vecino más cercano [5].

2.2 Algoritmo de alineamiento de las imágenes

El algoritmo que utiliza los puntos de control es el conocido como RANSAC (Random Sample Consensus) [4]. Este algoritmo utiliza la lista de posibles correspondencias entre puntos de control. El criterio que se sigue para asociar las correspondencias es la distancia Euclídea, la cual debe ser menor que un umbral establecido th_0. Se establece como $m = (x_i, y_i)$el punto de control de la imagen 1 y $m' = (x_i', y_i')$el punto de control correspondiente de la imagen 2. Las correspondencias deben satisfacer la siguiente restricción geométrica:

$$A^2 + C^2 \approx B^2 + D^2 \tag{2.4}$$

Donde $A = (x_i^{'} - x_j^{'}),\quad B = (y_i^{'} - y_j^{'}),\quad C = (x_i - x_j),\ D = (y_i - y_j).$ En este caso se debe cumplir que

$$|(A^2 + C^2) - (B^2 + D^2)| < th_1 \tag{2.5}$$

Dondeth_1 es un umbral de error definido.Estos dos pares de correspondencias se utilizan para hallar los parámetros de alineación(t_x, t_y, θ) a partir de las siguientes ecuaciones:

$$t_x = x_i - x_i^{'} cos\theta - y_i^{'} sen\theta \tag{2.6}$$

$$t_y = y_i - y_i^{'} cos\theta - x_i^{'} sen\theta \tag{2.7}$$

$$\theta = arctan\left(\frac{BC - AD}{AC + BD}\right) \tag{2.8}$$

El algoritmo se encarga de buscar las posibles correspondencias que apoyan la transformación calculada en (2.5). Para cada punto del conjunto $m^{'}$ se le aplica dicha fómula y se busca la correspondencia en el conjunto *m*. Si esta cantidad de puntos consistentes es mayor o igual que cierto valor, se considera que el modelo es adecuado y finaliza el algoritmo. En caso contrario, se repite el proceso calculando de nuevo la transformación geométrica.

Para el presente proyecto, el algoritmo RANSAC permite obtener a partir de los puntos de control los valores de traslación, rotación y escala entre una imagen y otra. Esto alinea las fotografías en el mosaico final. Sin embargo, esos resultados deben ser evaluados y reajustados de ser necesario al sistema de coordenadas de la última fusión, ya que este movimiento de coordenadas varía con la unión de cada cuadro. El algoritmo de reajuste forma parte de la solución para concluir el alineamiento de las múltiples imágenes y se explica con detalle en el siguiente capítulo.

Capítulo 3: Alineación de Múltiples Imágenes

En este proyecto se extiende el método de fusión propuesto en [1] para que pueda tratar con más de dos imágenes provenientes de una muestra de nematodos y fusionarlas en una sola imagen compuesta. Una vez realizada la unión de todos los cuadros se procede a optimizar el procedimiento reutilizando el cálculo de datos para ahorrar tiempo en el procesamiento de la información.

Algunos de estos métodos se implementan como parte de la biblioteca de algoritmos y estructuras de datos utilizadas en el procesamiento digital de imágenes LTI-Lib-2 [7]. Todos los programas que contiene esta biblioteca son escritos en lenguaje C++, por lo que los algoritmos implementados son escritos en este mismo lenguaje.

En la estructura de clases de la LTI-Lib se implementó el código perteneciente a la solución del presente proyecto. Los algoritmos de detección de puntos, de descriptores SURF y de RANSAC ya existían en la biblioteca y fueron reutilizados. Como se muestra en la figura 1.1 las dos primeras etapas del diagrama general de bloques no son parte del desarrollo de la solución del proyecto. En la siguiente sección se explica el funcionamiento de los algoritmos de registro de imágenes como punto de partida para los algoritmos desarrollados en las etapas finales del proyecto. En la sección 3.2 se explican con detalle los algoritmos propios de la solución del problema planteado.

3.1 Obtención de los puntos de interés y descriptores

La biblioteca LTI-Lib contiene algoritmos para la detección de los puntos de interés y de los descriptores de una imagen. Estos métodos se encapsulan en clases.

3.1.1 Puntos de interés

La clase utilizada para la obtención de los puntos de interés recibe como argumento la imagen a analizar y su valor de retorno, consiste en una estructura de datos tipo lista con los puntos característicos ya calculados utilizando el algoritmo hessiano de detección rápida de puntos. Cada punto de interés contiene 4 valores flotantes: el valor de la posición en el eje x, el valor en la posición en el eje y, el ángulo de rotación en radianes y la escala. En la figura 3.1 se muestra los puntos característicos marcados en color anaranjado sobre una imagen de prueba.

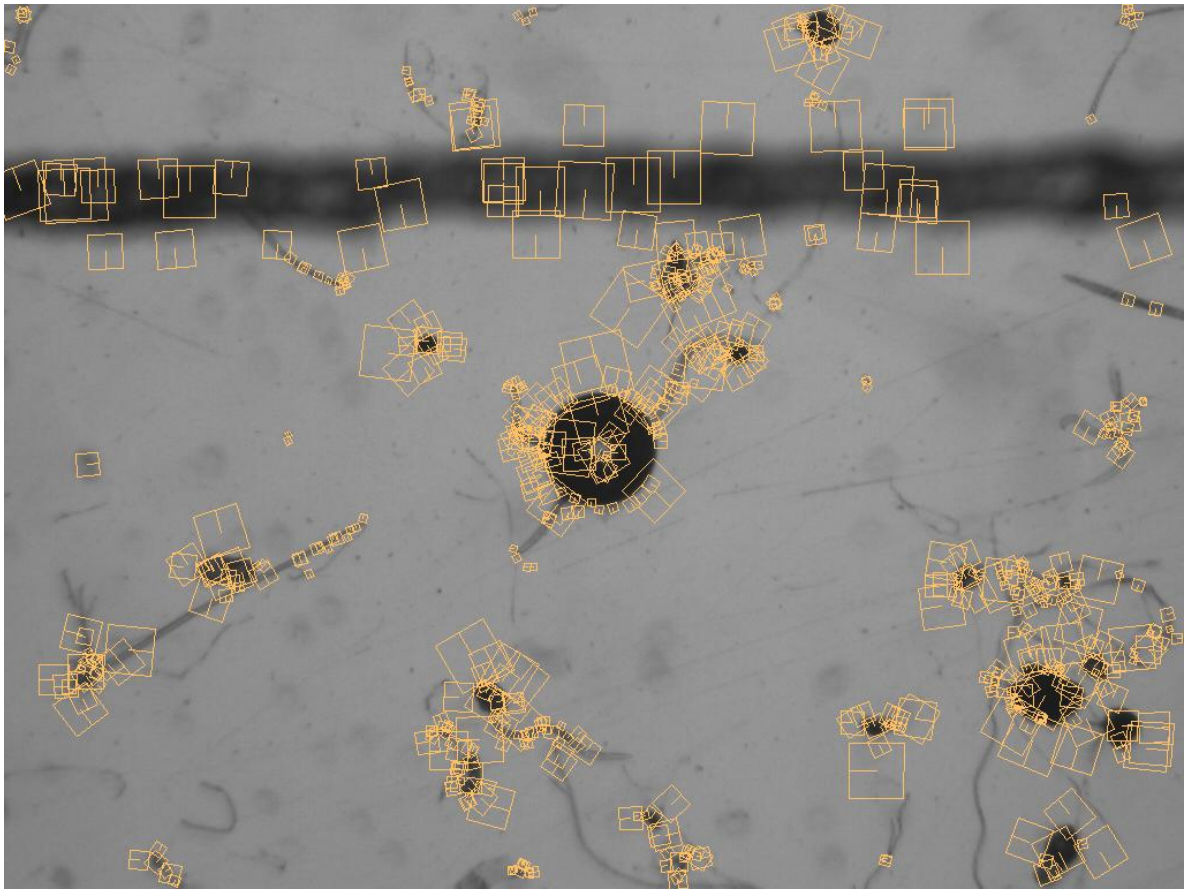

Figura 3.1 Puntos de interés en una imagen de prueba

En la implementación de la solución el algoritmo, antes mencionado, se utiliza para poder tener las listas con los puntos característicos de cada una de las

imágenes que se van a analizar. Estos puntos obtenidos se usan para obtener los descriptores y los puntos de control.

3.1.2 Descriptores

La obtención de los descriptores se implementa en la biblioteca LTI-Lib por medio del algoritmo SURF, la cual recibe como argumentos la imagen que se desea analizar y los puntos característicos obtenidos en la sección anterior. Retorna una estructura de datos de tipo vector con los descriptores de cada punto de interés calculado.

3.1.3 Puntos de control

Los puntos de control son aquellos puntos característicos de una imagen que tienen una correspondencia idéntica en otra imagen. Para obtenerlos, la biblioteca LTI-Lib utiliza la información de los descriptores asociados a los puntos de interés en dos imágenes e identifica aquellos que tienen correspondencias en ambas. Recibe como argumentos los dos cuadros a analizar, las dos listas con los puntos de interés y los dos vectores con los descriptores obtenidos. Las estructuras de datos que retorna son las listas de puntos de interés y sus respectivos descriptores de aquellos puntos que tengan reciprocidad entre sí en una imagen y en la otra. En la figura 3.2 se muestran los puntos de control localizados en dos imágenes de prueba.

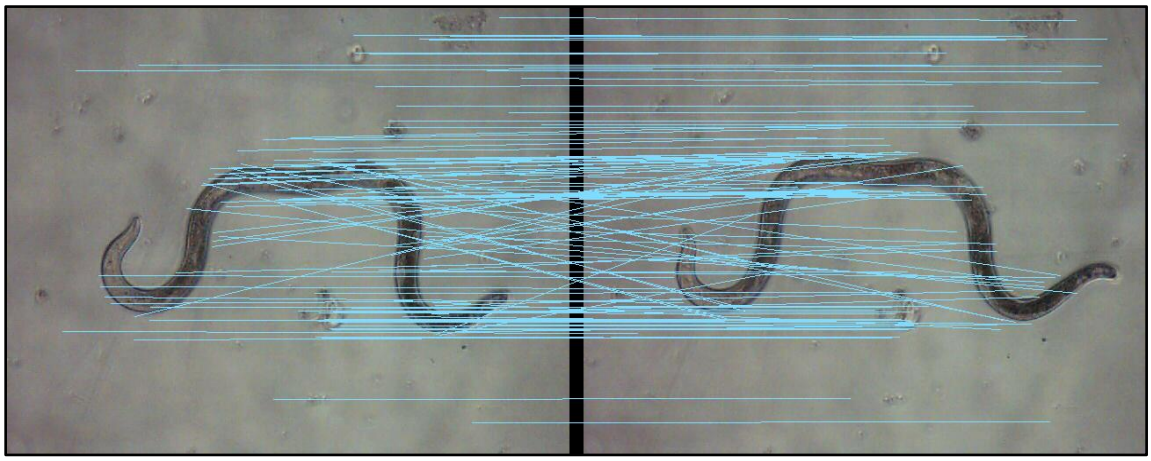

Figura 3.2 Puntos de control en dos imágenes de prueba

3.1.4 Alineación de las imágenes

La biblioteca LTI-Lib implementa el algoritmo conocido como RANSAC [4]. Este algoritmo es una estrategia de optimización para hacer estimaciones *simples* o *determinísticas* usando el mínimo número de correspondencias y luego repetir el proceso aleatoriamente para finalmente seleccionar el menor error.

Los puntos de control generados poseen los cuatro valores mencionados anteriormente (posición en el eje x, posición en el eje y, el ángulo de rotación en radianes y la escala), sin embargo, para el algoritmo RANSAC solamente se usan los datos correspondientes a las posiciones x e y, por lo que se deben extraer de los puntos de control para poder realizar el análisis de la manera correcta.

De esta forma, se recibe como argumentos dos estructuras tipo vector con estos valores de x e y, provenientes de cada uno de los puntos de control, y devuelve un vector con los puntos compatibles con la transformación estimada calculados por el RANSAC y además retorna la información acerca de la traslación en los ejes x e y, la rotación y la escala de las dos imágenes analizadas. Estos últimos datos determinan la ubicación de cada cuadro en el nuevo sistema de coordenadas de la imagen resultante.

3.2 Implementación de la unión de múltiples imágenes

Los algoritmos implementados para la unión de imágenes funcionan con solamente dos cuadros y se ha demostrado su eficiencia y robustez en los análisis experimentales en [1]. Este proyecto va enfocado a extender esos algoritmos para la unión de múltiples imágenes para generar una muestra completa de nematodos.

Básicamente, para obtener el resultado esperado, se deben contemplar dos aspectos: el ajuste a un nuevo sistema de coordenadas de los puntos característicos y por ende de las imágenes; y la reutilización de procesos para reducir el tiempo de análisis en la composición de los cuadros en el mosaico final.

3.2.1 Reajuste del marco resultante

Para cada ensamble de dos imágenes debe existir un marco que las contenga. En el caso de múltiples cuadros, este borde se modifica constantemente pues cada imagen es adherida a la imagen resultante y ésta a su vez va aumentando el tamaño de dicho borde. Estos cambios en las dimensiones del recuadro final deben ser considerados en el algoritmo de unión para evitar que una imagen se ubique fuera del marco resultante y también que cada cuadro sea ubicado en el lugar preciso según los cálculos realizados. El marco que contiene todos los cuadros se crea con base en un tamaño horizontal y vertical. Este tamaño es calculado utilizando los parámetros obtenidos en el algoritmo RANSAC y aplicando la transformación de similitud a las cuatro esquinas. Con esto se extiende el marco de manera tal que incluya las cuatro esquinas transformadas.

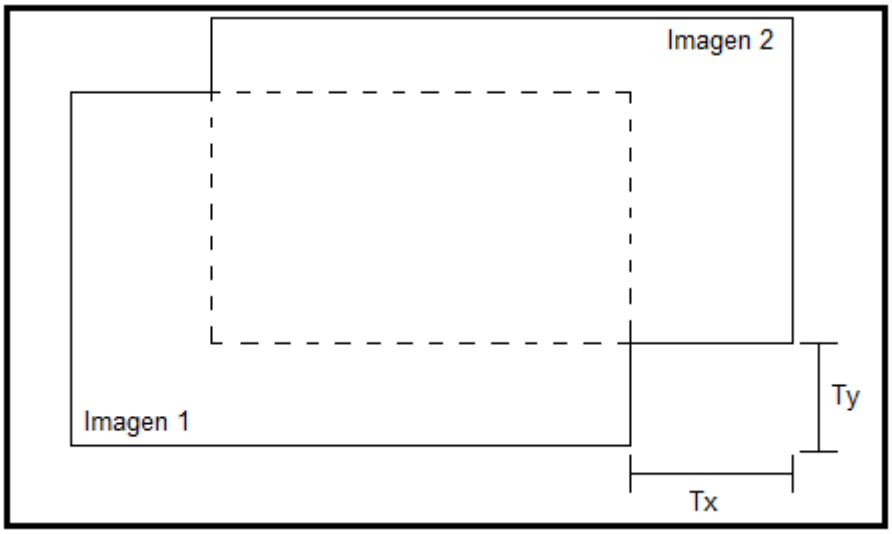

Figura 3.3 Traslación entre dos imágenes

En la figura 3.3 se muestran estos valores de traslación que son calculados por el algoritmo RANSAC. La traslación en el eje horizontal se define como T_x mientras que en el eje vertical se denomina T_y.

Estas traslaciones hacen que el marco final que contenga las imágenes crezca en proporción con dichos valores y se deben contemplar los casos en los que la imagen 1 se sobrepone más a la izquierda, más arriba, más abajo o más a la derecha que la imagen 2 y viceversa. De esta manera, los análisis con respecto a las traslaciones deben ser los siguientes:

1. **Cuando la traslación en x es negativa:** esto se debe a que la imagen 2 está más a la izquierda que la imagen 1, es decir, el movimiento de la cámara entre un cuadro y otro es hacia la izquierda. En este caso la imagen 2 se ubica en la posición 0 del eje x del marco resultante, mientras que la imagen 1 se ubica en la posición igual al valor absoluto de la traslación en x obtenida.

2. **Cuando la traslación en x es positiva:** esto se debe a que la imagen 1 se encuentra más a la izquierda que la imagen 2, es decir, el movimiento de la cámara entre un cuadro y otro es hacia la derecha. En este caso la imagen

1 se ubica en la posición 0 del eje x del marco resultante, mientras que la imagen 2 se ubica en la posición igual a la traslación en x obtenida.

3. **Cuando la traslación en x es cero:** esto se debe a que no existe traslación en el eje horizontal entre los cuadros. En este caso la imágenes 1 y 2 se ubican en la posición 0 del eje x del marco resultante.

4. **Cuando la traslación en y es negativa:** esto se debe a que la imagen 1 se encuentra más arriba que la imagen 2, es decir, el movimiento de la cámara entre un cuadro y otro es hacia abajo. En este caso la imagen 1 se ubica en la posición 0 del eje y del marco resultante, mientras que la imagen 2 se ubica en la posición igual al valor absoluto de la traslación en y obtenida.

5. **Cuando la traslación en y es positiva:** esto se debe a que la imagen 2 se encuentra más arriba que la imagen 1, es decir, el movimiento de la cámara entre un cuadro y otro es hacia arriba. En este caso la imagen 2 se ubica en la posición 0 del eje y del marco resultante, mientras que la imagen 1 se ubica en la posición igual al valor de la traslación en y obtenida.

6. **Cuando la traslación en y es cero:** esto se debe a que no existe traslación en el eje vertical entre los cuadros. En este caso las imágenes 1 y 2 se ubican en la posición 0 del eje y del marco resultante.

El algoritmo que reajusta el tamaño del borde resultante toma en cuenta estas consideraciones para crear el marco con las dimensiones apropiadas. Y para los casos en los que no hay ninguna rotación entre una imagen y la otra el tamaño del mosaico final siempre será igual al tamaño de una de las imágenes más la respectiva traslación existente. Excepto en los casos en que una imagen cae dentro de la otra y aunque puede haber traslación, ésta no supera el tamaño del cuadro que contiene al otro. En la figura 3.4 se muestra un ejemplo particular en la que la imagen 2 queda contenida verticalmente en la imagen 1, pero una parte de ella queda por fuera a nivel horizontal. En estos casos específicos el

algoritmo calcula la cantidad que debe aumentar en vertical y horizontal tomando en cuenta la diferencia existente en la traslación con respecto al tamaño de la imagen mayor. Estos casos se presentan con mayor frecuencia conforme se vayan uniendo apropiadamente todos los cuadros de la muestra completa.

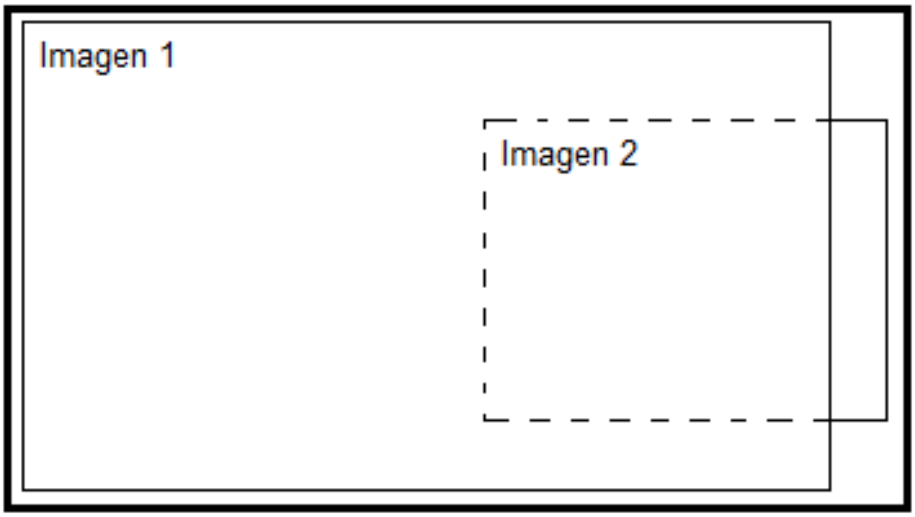

Figura 3.4 Ejemplo de una imagen contenida en otra

3.2.2 Reajuste de los puntos de interés

Debido a que el marco resultante cambia de tamaño constantemente se ocasiona que el tiempo de procesamiento crezca de manera polinomial, por lo que a mayor cantidad de cuadros a analizar, mayor es el tiempo invertido en el proceso. Por lo tanto, el algoritmo de unión de múltiples imágenes debe reducir el tiempo de procesamiento reutilizando datos, ya que, muchos de los cálculos obtenidos se repiten al fusionar una nueva imagen a la estructura previa.

Cada vez que ocurre este ensamble se deben determinar los puntos de interés y los descriptores. Una vez que se realizó la unión, esta imagen resultante toma el lugar de la primera imagen para realizar la siguiente adición. Esto ocasiona que nuevamente se deban computar los puntos característicos y los descriptores tanto del primer cuadro como del segundo, sin embargo, los puntos y

los descriptores encontrados en el primer cuadro son iguales a los calculados en la primera unión.

Debido a esto se pueden almacenar los puntos de interés y los descriptores procedentes de ambas imágenes para que éstos sirvan en el análisis de la unión siguiente. Y, de esta manera, los puntos y descriptores solamente se calculan en el primer ensamble y de ahí en adelante la estimación se realiza únicamente para el segundo cuadro que es la nueva figura que se integra al mosaico resultante. Con esto se obtiene una reducción del tiempo de procesamiento en la unión de todas las fotografías que se tienen que procesar.

Sin embargo, almacenar los puntos de interés trae consigo una nueva consideración que debe ser tomada en cuenta. Al modificarse el tamaño el marco final con cada nueva unión, los puntos de interés se ven afectados por un cambio de posición, rotación y escala para reajustarse al sistema de coordenadas del recuadro obtenido. Es por esta razón que los puntos característicos deben pasar por un proceso de transformación en sus valores originales antes de ser almacenados para continuar el proceso de fusión.

Los valores de traslación en cada punto deben ser modificados con respecto a las traslaciones y a la rotación, cuya obtención se explicó en el apartado anterior. Esto produce un nuevo punto de interés acomodado al nuevo sistema de coordenadas del marco final.

En la figura 3.5 se muestra un diagrama de bloques general del algoritmo de fusión de múltiples imágenes que contempla todos los procesos explicados anteriormente.

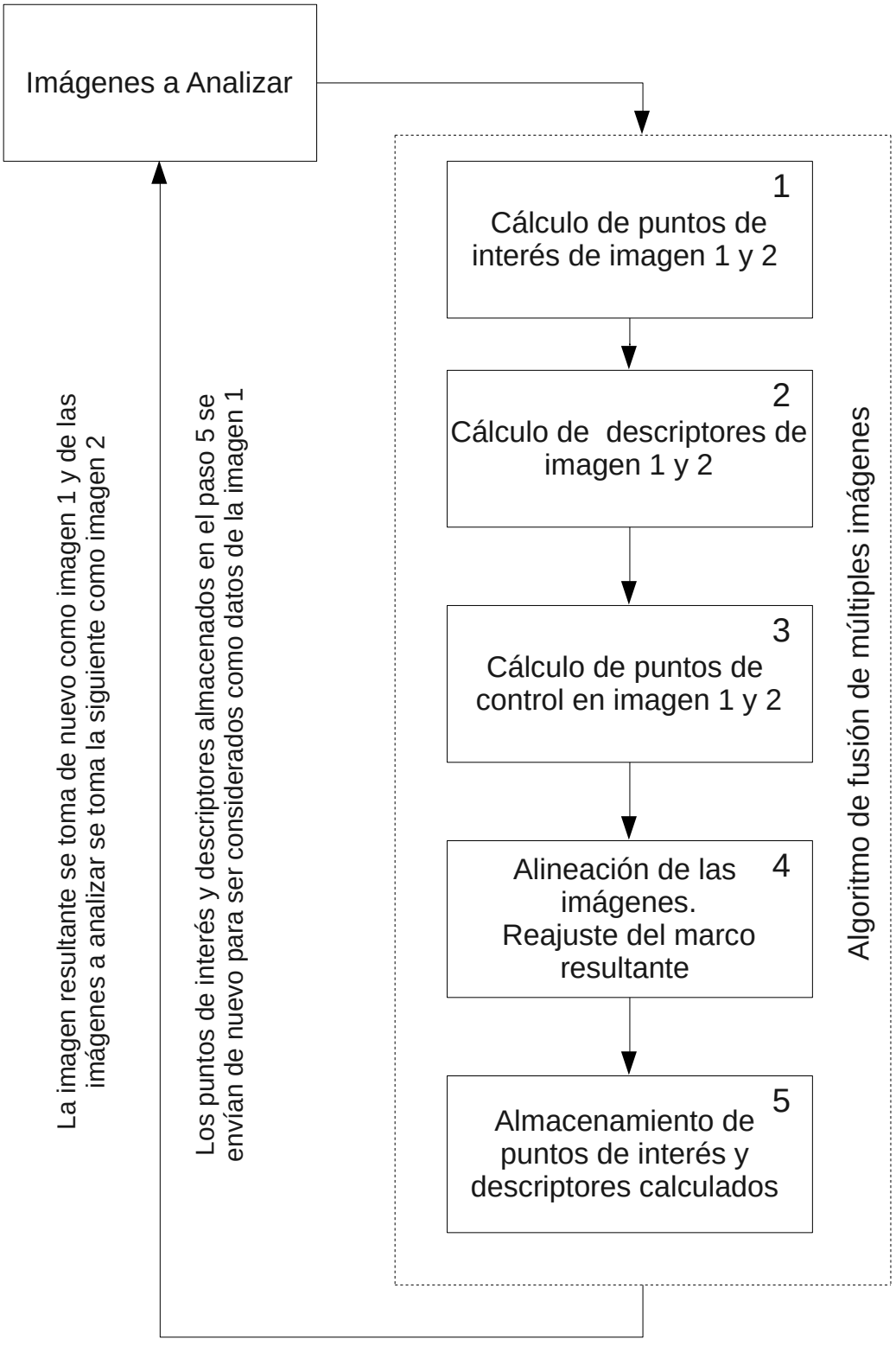

Figura 3.5 Diagrama general del algoritmo de fusión de múltiples imágenes

Capítulo 4: Análisis de Resultados

Una vez descrita e implementada la solución, se presentan los resultados obtenidos y un análisis de éstos tanto para el algoritmo de ajuste del marco final como para el algoritmo de reajuste de puntos de interés. En el caso del marco final, se presentan ejemplos con imágenes tomadas de muestras de nematodos ilustrando la variación teórica y la resultante del recuadro que contiene la unión. Para el caso del reajuste de puntos de interés, se muestran resultados de las fusiones de múltiples imágenes de muestras completas y la comparación de los tiempos de duración, tanto cuando se implementa el algoritmo de reajuste como cuando no se implementa.

4.1 Ajuste del marco resultante

Se desarrolla un programa de partición de imágenes que toma una imagen y la parte en una matriz de cuadros de un tamaño deseado. Además, permite generar cuadros con un traslape determinado en el eje x e y. Con este programa se puede conocer la ubicación de la esquina superior izquierda de cada cuadro dentro del marco resultante final. Estos valores de posición son tomados como los valores teóricos de ubicación que cada cuadro debe tener cuando se forme la imagen completa.

Luego se define un algoritmo que toma lo datos resultantes generados por el método RANSAC y que, con base en ellos, recalcula el tamaño del marco que contiene las imágenes fusionadas con respecto a los valores de traslación de los cuadros. Además, calcula la posición de la esquina superior izquierda de cada imagen que está siendo añadida y estos valores de posición son tomados como los valores experimentales de ubicación que son contrastados con los datos teóricos generados por el programa de partición de imágenes.

Para el análisis se toma una imagen de entrada de tamaño 1024x768 pixeles. Esta imagen es partida en varios cuadros con traslape en el eje x, luego en el eje y, y también en ambos. Los cuadros generados de la imagen de entrada son del tamaño de 256x256 o de 512x512 pixeles según los ejes en los que haya traslape. Para cada partición se generan 3 tipos de traslape: al 90%, al 80% y al 70% aproximadamente. Debido a que los cuadros generados de la imagen de entrada no exceden los 512 pixeles en sus dimensiones, no se establecen traslapes menores al 70% pues experimentalmente el algoritmo de detección de puntos no determina la suficiente cantidad de puntos correspondientes como para que se puedan unir los cuadros de manera correcta.

Una vez generada la matriz de cuadros y conociendo las posiciones de cada cuadro dentro de la imagen resultante se procede a ejecutar el algoritmo de unión para que vuelva a armar la imagen completa con cada uno de los cuadros generados. De esta forma se evalúan las posiciones calculadas de cada imagen dentro del marco resultante, así como el error de posicionamiento, el error generado conforme avanza el algoritmo, y el error promedio y desviación estándar de los errores.

4.1.1 Traslape en el eje x

En este primer caso se tiene una imagen de entrada mostrada en la figura 4.1. Usando el programa de partición, se divide la imagen en cuadros de tamaño 256x768 pixeles y con solamente traslape en el x.

En la tabla 4.1 se muestran los datos y errores de posicionamiento para cada cuadro generado con un traslape de 230 pixeles en x que es aproximadamente un 90% de traslape. Con este valor de traslape se generan 31 cuadros en total. En la figura 4.2 se muestra una gráfica con la tendencia del error de posicionamiento contra la iteración del algoritmo y en la figura 4.3 se muestra la imagen resultante luego de unir todos los cuadros.

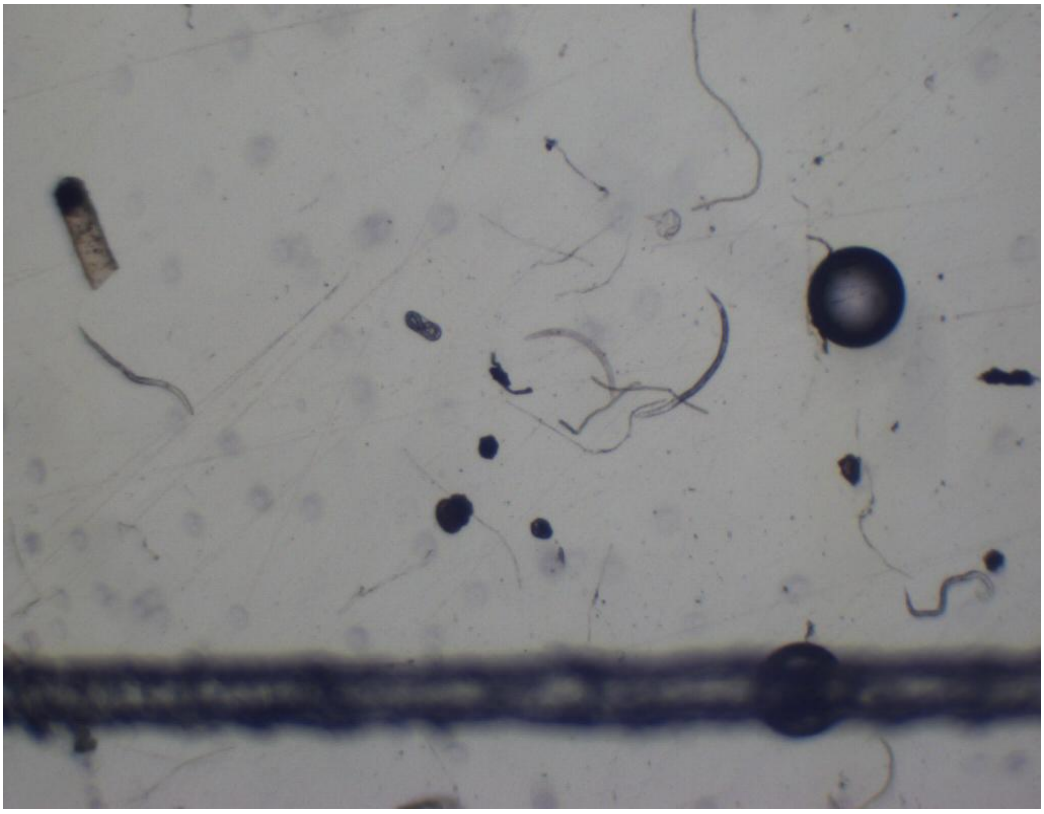

Figura 4.1 Imagen de entrada para el análisis con traslape en el eje x

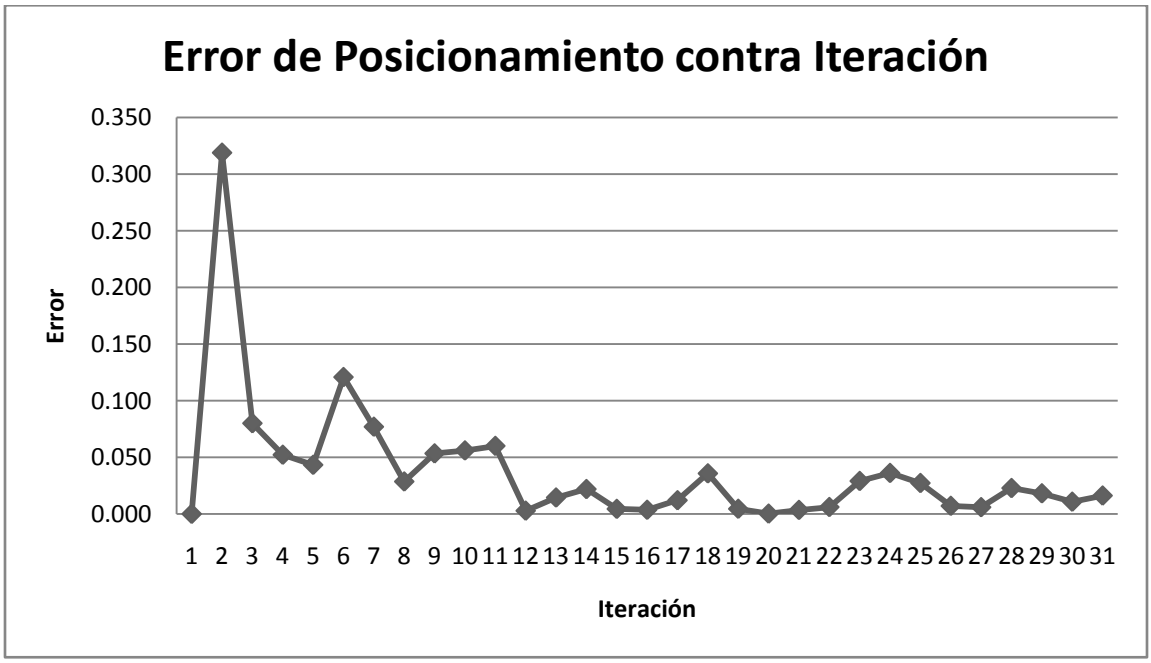

Figura 4.2 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.1

Tabla 4.1 Datos de posicionamiento de los cuadros generados de tamaño 256x768 a partir de la figura 4.1 con un traslape del 90%

No. Iteración	Posicionamiento Teórico Eje x	Posicionamiento Experimental Eje x	Error (%)	Error Promedio (%)	Desviación Estándar del Error
1	0	0	0.000	0.038	0.0593
2	26	25.9171	0.319		
3	52	52.0416	0.080		
4	78	77.9593	0.052		
5	104	103.955	0.043		
6	130	129.843	0.121		
7	156	155.88	0.077		
8	182	182.052	0.029		
9	208	208.111	0.053		
10	234	234.131	0.056		
11	260	260.156	0.060		
12	286	285.992	0.003		
13	312	312.045	0.014		
14	338	337.926	0.022		
15	364	364.016	0.004		
16	390	389.986	0.004		
17	416	415.95	0.012		
18	442	442.158	0.036		
19	468	467.979	0.004		
20	494	493.999	0.000		
21	520	520.018	0.003		
22	546	545.967	0.006		
23	572	572.166	0.029		
24	598	598.216	0.036		
25	624	624.17	0.027		
26	650	650.046	0.007		
27	676	675.96	0.006		
28	702	702.16	0.023		
29	728	728.132	0.018		
30	754	754.081	0.011		
31	780	779.874	0.016		

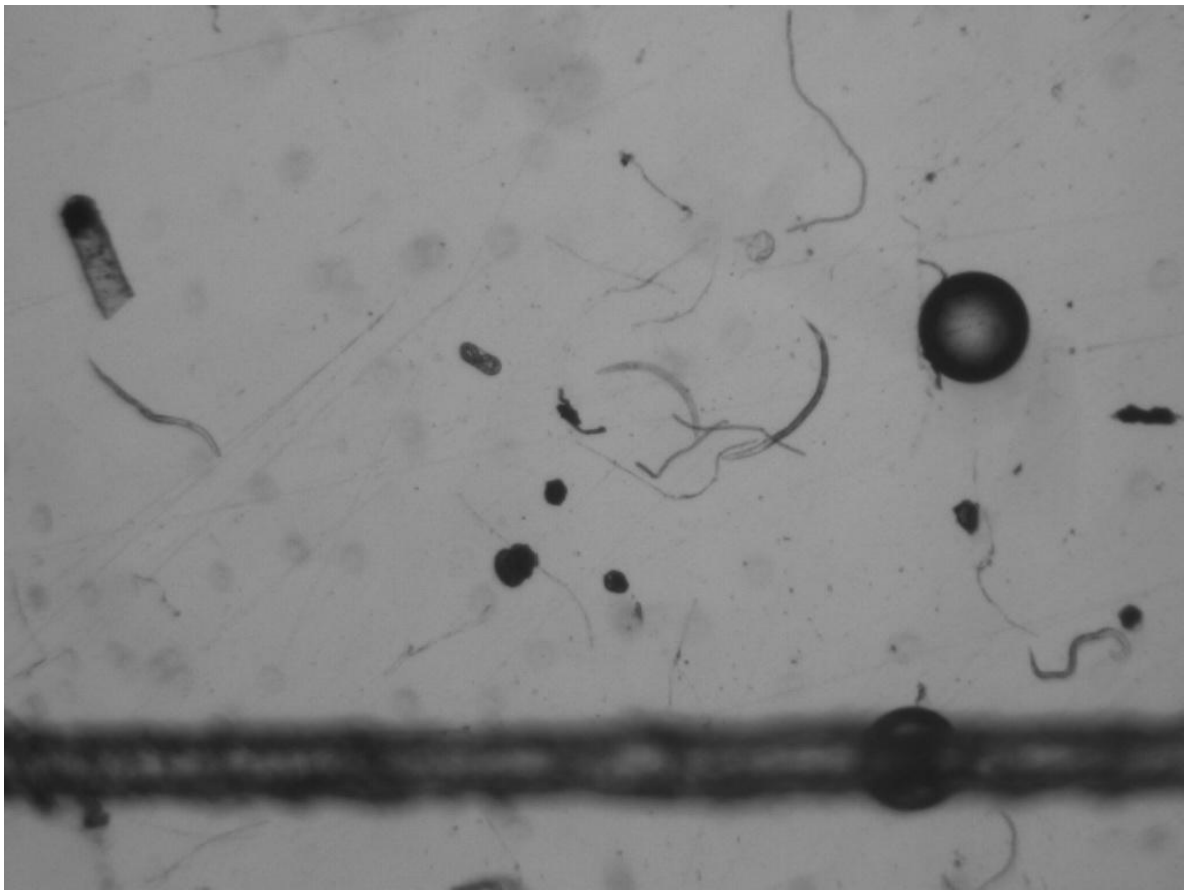

Figura 4.3 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.1 con un traslape en el eje x de 90%

Como se aprecia en la figura 4.2 el algoritmo de unión de múltiples imágenes presenta una tendencia de disminución del error en el posicionamiento de los cuadros conforme avanza en el número de iteraciones, manteniéndose cercano al valor del error promedio. En la tabla 4.1 se puede apreciar que los valores de posición de la esquina superior izquierda de cada cuadro son muy cercanos a los valores teóricos lo cual produce un error promedio de apenas 0.038% con una desviación estándar de 0.0593. Y como se muestra en la figura 4.3 la imagen resultante de la unión es idéntica a la imagen entrada, esto debido al traslape implementado y al poco error generado.

En el siguiente caso se muestra el resultado de aplicar un 80% de traslape en el eje x en los cuadros generados a partir de la figura 4.1. El porcentaje de

traslape representa aproximadamente 207 pixeles con cuadros de tamaño 256x768 pixeles, generando un total de 17 cuadros. En la tabla 4.2 se muestran los datos de posicionamiento. En la figura 4.4 la gráfica de error de posicionamiento contra iteración y en la figura 4.5 la imagen resultante después aplicar el algoritmo de unión.

Tabla 4.2 Datos de posicionamiento de los cuadros generados de tamaño 256x768 a partir de la figura 4.1 con un traslape del 80%

No. Iteración	Posicionamiento Teórico Eje x	Posicionamiento Experimental Eje x	Error (%)	Error Promedio (%)	Desviación Estándar del Error
1	0	0	0.000	0.066	0.0401
2	49	48.9763	0.048		
3	98	97.8732	0.129		
4	147	146.903	0.066		
5	196	195.804	0.100		
6	245	245.027	0.011		
7	294	293.911	0.030		
8	343	342.83	0.050		
9	392	391.806	0.049		
10	441	440.506	0.112		
11	490	489.502	0.102		
12	539	538.424	0.107		
13	588	588.166	0.028		
14	637	637.102	0.016		
15	686	686.647	0.094		
16	735	735.635	0.086		
17	784	784.753	0.096		

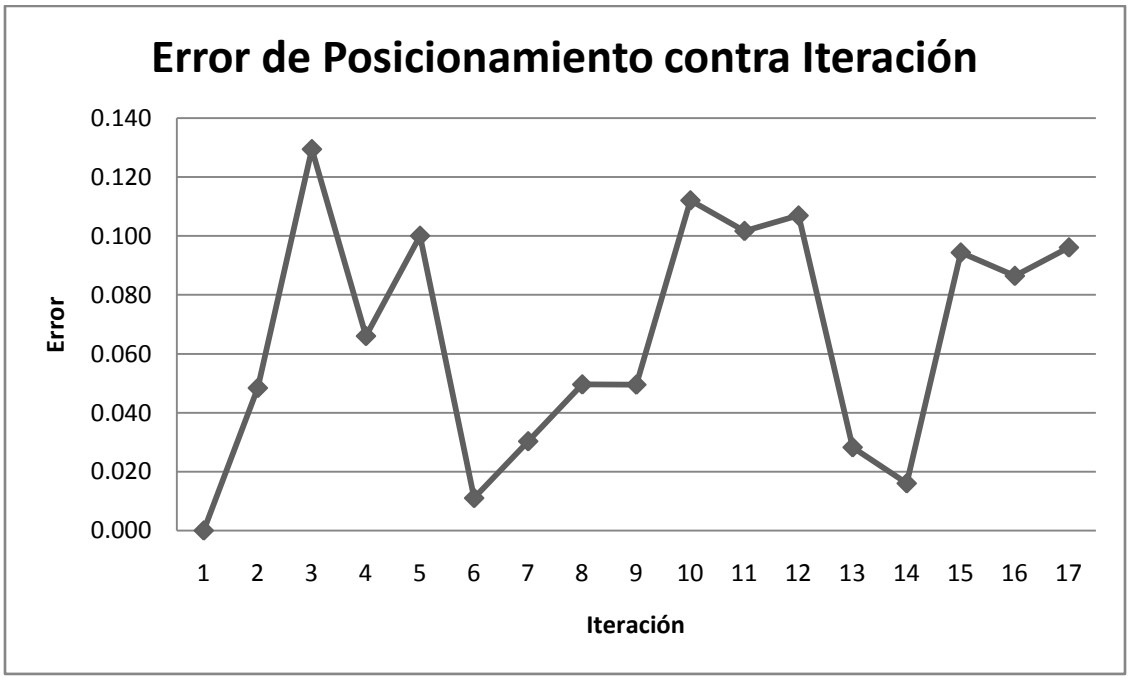

Figura 4.4 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.2

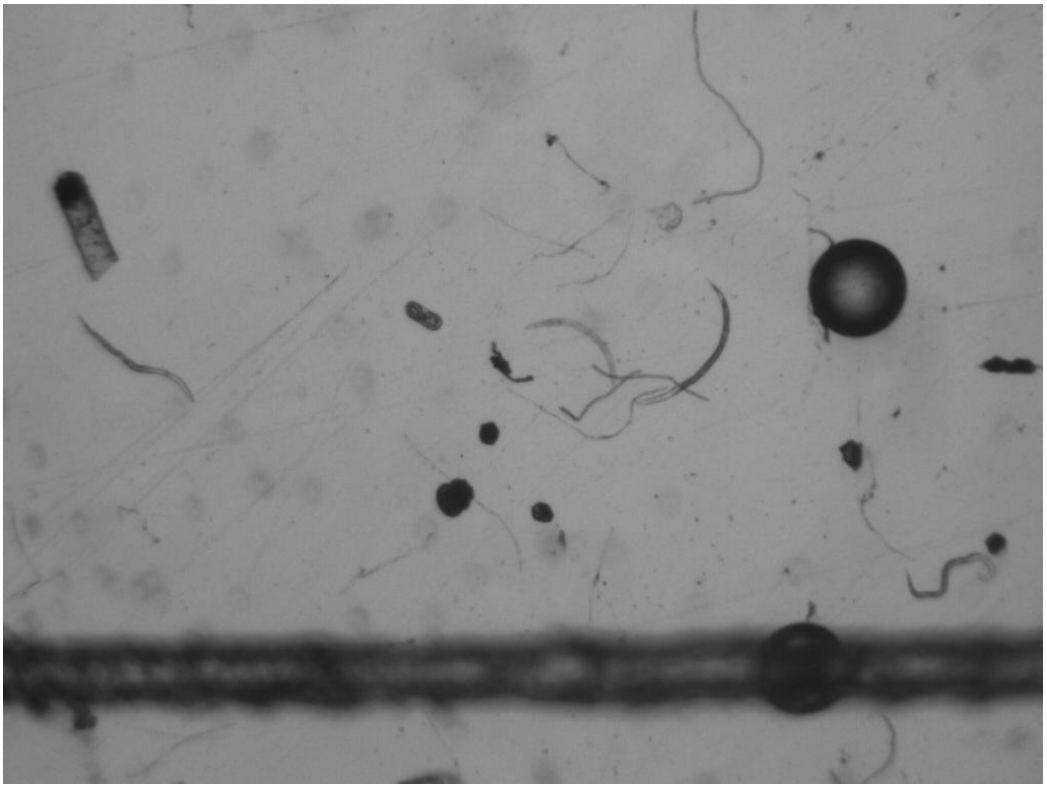

Figura 4.5 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.1 con un traslape en el eje x de 80%

En el caso en que se aplica un 80% de traslape en el eje x se observa en la tabla 4.2 que los valores de posicionamiento calculados son bastante parecidos a los teóricos, sin embargo, no se aprecia una disminución en el error conforme avanzan las iteraciones del algoritmo (ver figura 4.4), esto debido a que la cantidad de cuadros generados es menor en comparación a cuando se aplica un traslape del 90%, ocasionando que los errores de posicionamiento se mantengan distribuidos durante todas las iteraciones alrededor del error promedio. Aún así, el algoritmo de unión fusiona los cuadros de forma idéntica a la imagen de entrada.

En el siguiente caso se muestra el resultado de aplicar un 70% de traslape en el eje x en los cuadros generados a partir de la figura 4.1. El porcentaje de traslape representa aproximadamente 180 pixeles con cuadros de tamaño 256x768 pixeles, generando un total de 12 cuadros. En la tabla 4.3 se muestran los datos de posicionamiento. En la figura 4.6 la gráfica de error de posicionamiento contra iteración y en la figura 4.7 la imagen resultante después aplicar el algoritmo de unión.

Tabla 4.3 Datos de posicionamiento de los cuadros generados de tamaño 256x768 a partir de la figura 4.1 con un traslape del 70%

No. Iteración	Posicionamiento Teórico Eje x	Posicionamiento Experimental Eje x	Error (%)	Error Promedio (%)	Desviación Estándar del Error
1	0	0.000	0.000	0.024	0.0179
2	76	75.991	0.012		
3	152	151.950	0.033		
4	228	228.007	0.003		
5	304	304.167	0.055		
6	380	380.131	0.034		
7	456	456.176	0.039		
8	532	532.151	0.028		
9	608	608.065	0.011		
10	684	684.212	0.031		
11	760	760.303	0.040		
12	836	836.013	0.002		

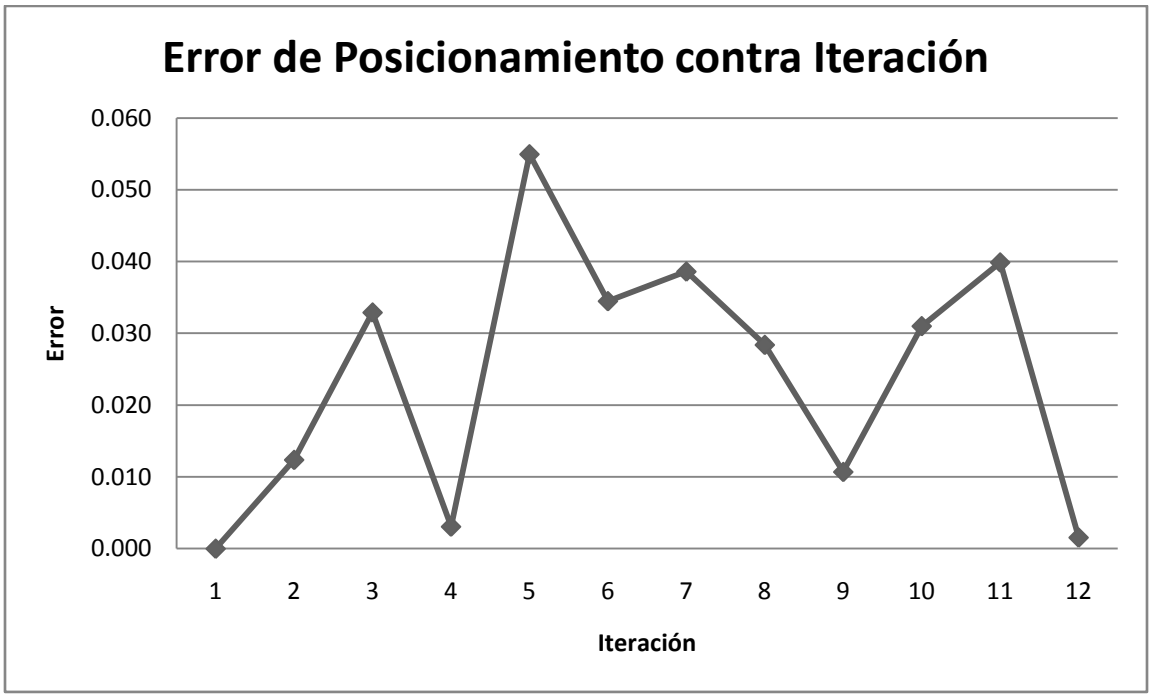

Figura 4.6 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.3

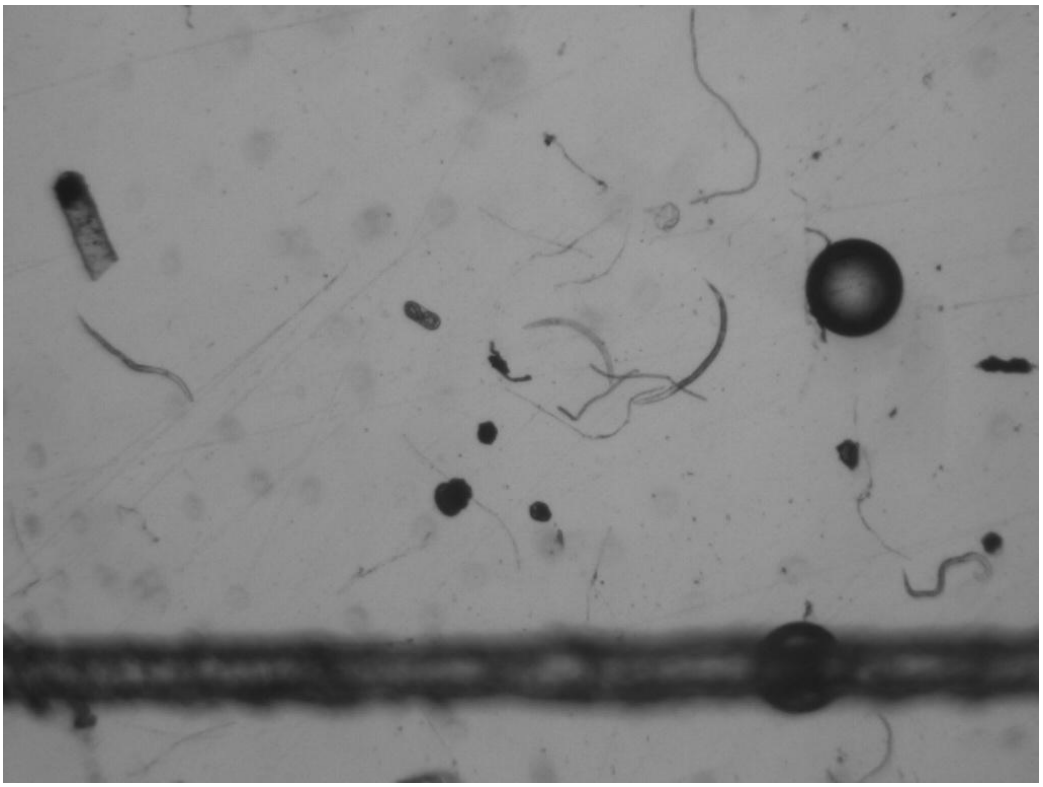

Figura 4.7 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.1 con un traslape en el eje x de 70%

Con un 70% de traslape en el eje x la tendencia del error contra la iteración se mantiene similar a cuando se aplica un 80% de traslape, esto es, los errores se distribuyen durante todas las iteraciones alrededor del error promedio. La imagen resultante sigue manteniéndose idéntica a la imagen de entrada.

En general, cuando se aplica solamente el traslape en el eje x los valores de posicionamiento experimentales en cada uno de los cuadros son cercanos a los valores teóricos, sin embargo, se necesita de un traslape de más del 90% para que el error disminuya conforme aumenten las iteraciones, de lo contrario los errores de posicionamiento se mantienen alrededor del error promedio durante todas las iteraciones.

4.1.2 Traslape en el eje y

En la figura 4.8 se muestra la imagen de entrada utilizada para generar cuadros con traslape únicamente en el eje y. Estos cuadros tienen un tamaño de 1024x256 pixeles.

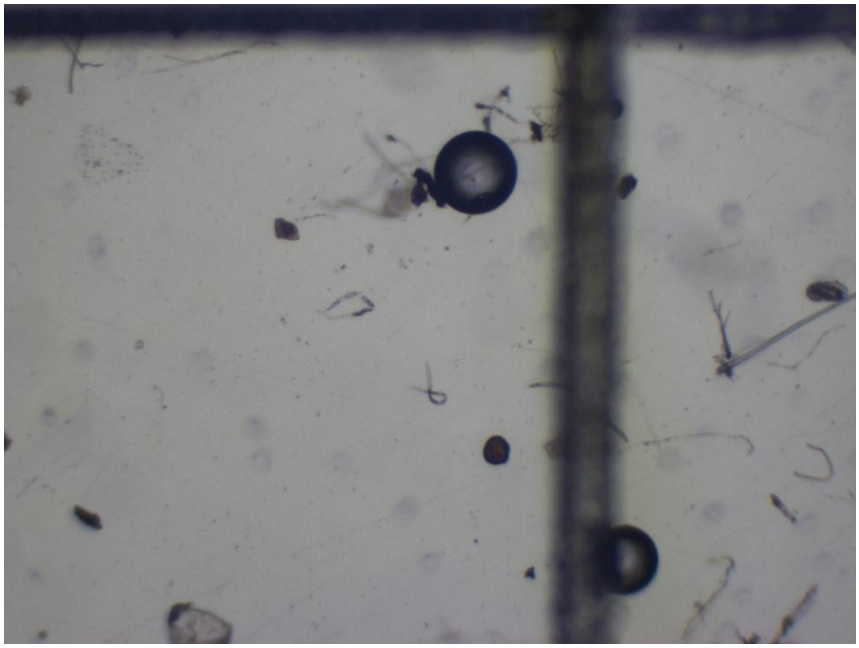

Figura 4.8 Imagen de entrada para el análisis con traslape en el eje y

En la tabla 4.4 se muestran los datos y errores de posicionamiento para cada cuadro generado con un traslape de 230 pixeles en y que es aproximadamente un 90% de traslape. Con este valor de traslape se generan 21 cuadros en total. En la figura 4.9 se muestra una gráfica con la tendencia del error de posicionamiento contra la iteración del algoritmo y en la figura 4.10 se muestra la imagen resultante final resultante luego de unir todos los cuadros.

Tabla 4.4 Datos de posicionamiento de los cuadros generados de tamaño 1024x256 a partir de la figura 4.8 con un traslape del 90%

No. Iteración	Posicionamiento Teórico Eje y	Posicionamiento Experimental Eje y	Error (%)	Error Promedio (%)	Desviación Estándar del Error
1	0	0	0.000	0.219	0.168
2	26	25.8146	0.713		
3	52	51.6964	0.584		
4	78	77.6588	0.437		
5	104	103.725	0.264		
6	130	129.667	0.256		
7	156	155.662	0.217		
8	182	181.525	0.261		
9	208	207.58	0.202		
10	234	233.578	0.180		
11	260	259.619	0.147		
12	286	285.612	0.136		
13	312	311.603	0.127		
14	338	337.522	0.141		
15	364	363.499	0.138		
16	390	389.512	0.125		
17	416	415.524	0.114		
18	442	441.679	0.073		
19	468	467.382	0.132		
20	494	493.241	0.154		
21	520	519.005	0.191		

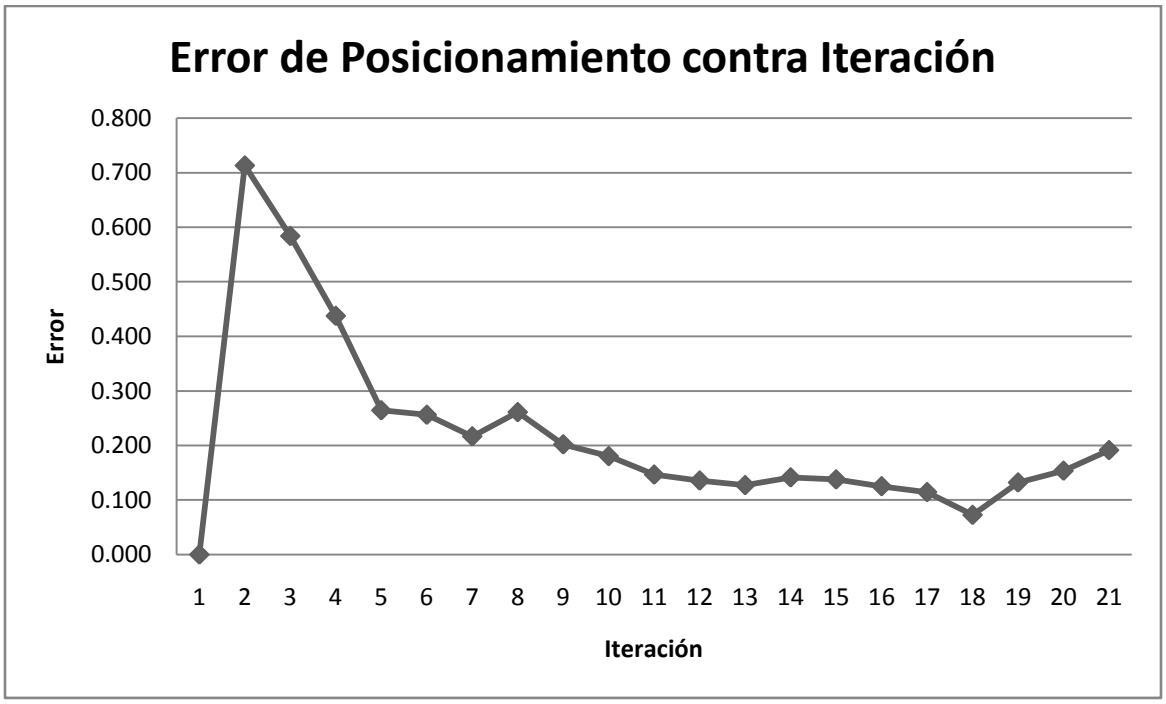

Figura 4.9 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.4

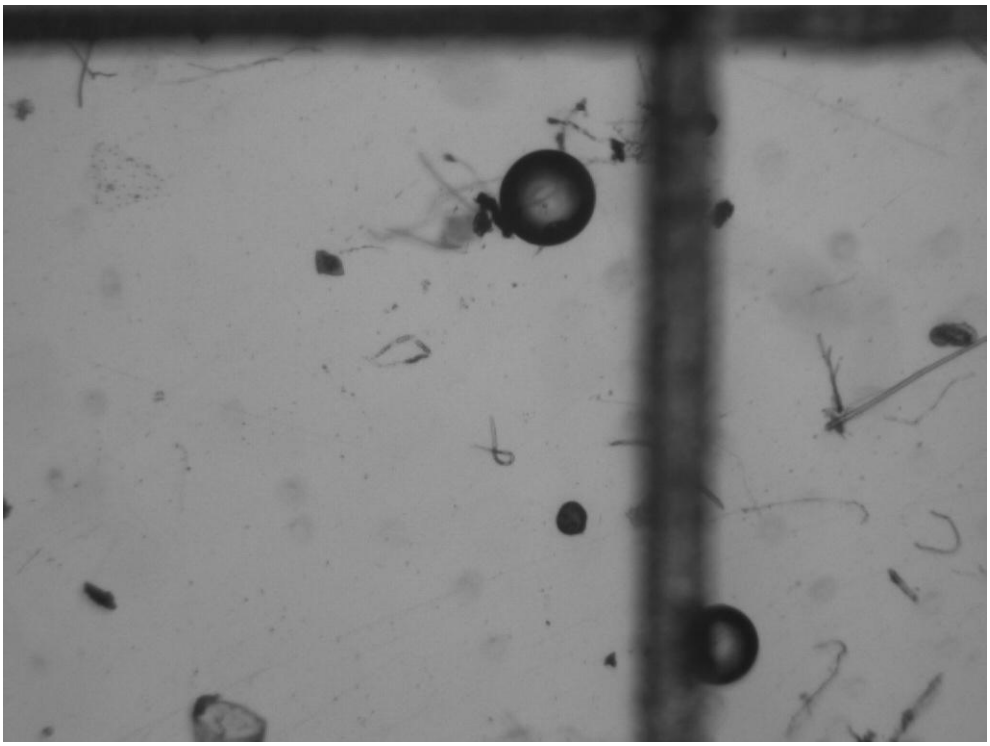

Figura 4.10 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.8 con un traslape en el eje y de 90%

En la tabla 4.4 se observa que los valores de posicionamiento de los cuadros se mantienen cercanos a los datos teóricos. El error tiene una tendencia similar a cuando se aplica traslape de 90% en el eje x, de manera que se incrementa con las primeras iteraciones, pero luego presenta una disminución considerable conforme avanza el algoritmo hasta mantenerse cerca del error promedio. La unión de la imagen resultante se mantiene idéntica a la imagen original demostrando la eficiencia del algoritmo cuando hay traslape en el eje y.

En el siguiente caso se muestra el resultado de aplicar un 80% de traslape en el eje y en los cuadros generados a partir de la figura 4.8. El porcentaje de traslape representa aproximadamente 207 pixeles con cuadros de tamaño 1024x256 pixeles, generando un total de 12 cuadros. En la tabla 4.5 se muestran los datos de posicionamiento. En la figura 4.11 la gráfica de error de posicionamiento contra iteración y en la figura 4.12 la imagen resultante después aplicar el algoritmo de unión.

Tabla 4.5 Datos de posicionamiento de los cuadros generados de tamaño 1024x256 a partir de la figura 4.8 con un traslape del 80%

No. Iteración	Posicionamiento Teórico Eje y	Posicionamiento Experimental Eje y	Error (%)	Error Promedio (%)	Desviación Estándar del Error
1	0	0.000	0.000	0.077	0.109
2	49	48.811	0.386		
3	98	98.074	0.075		
4	147	146.988	0.008		
5	196	195.734	0.136		
6	245	244.860	0.057		
7	294	294.041	0.014		
8	343	342.998	0.001		
9	392	392.019	0.005		
10	441	440.683	0.072		
11	490	489.825	0.036		
12	539	538.247	0.140		

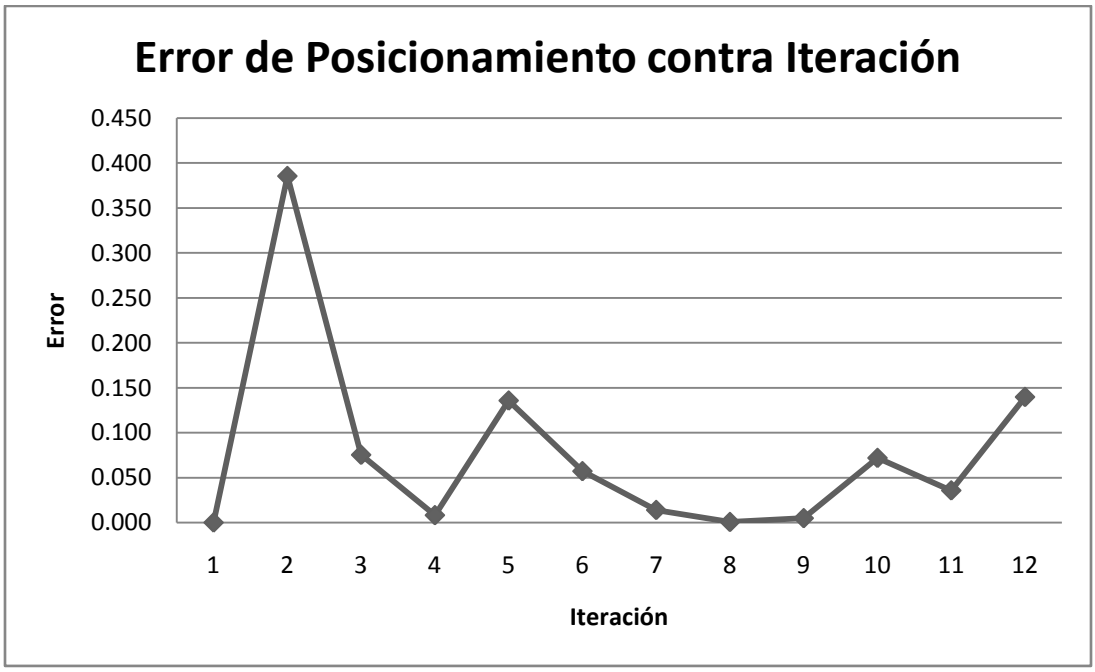

Figura 4.11 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.5

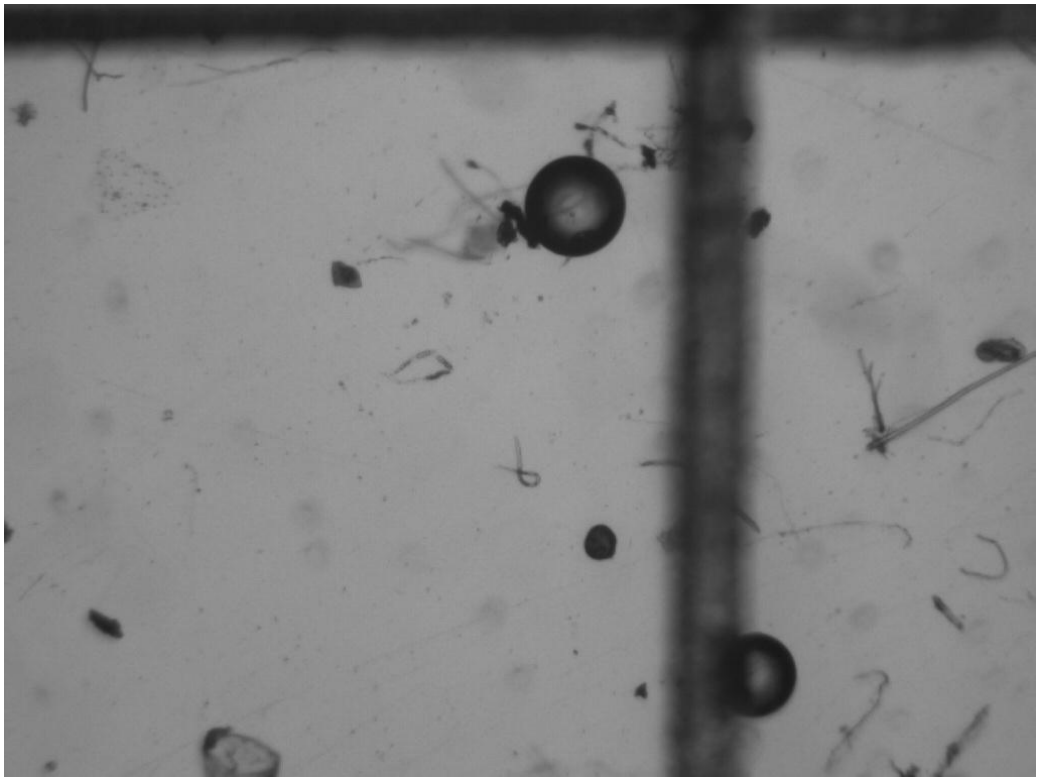

Figura 4.12 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.8 con un traslape en el eje y de 80%

De nuevo, los valores de posición teórico y experimentales son bastante parecidos con el traslape de 80% en el eje y (ver tabla 4.5). Sin embargo, la tendencia del error contra la iteración se mantiene similar a cuando se aplica un traslape del 90%. Situación que no se daba cuando se aplica un traslape sólo en el eje x. La imagen resultante se mantiene idéntica a la imagen de entrada cuando el traslape es de un 80%.

En el siguiente caso se muestra el resultado de aplicar un 70% de traslape en el eje y, en los cuadros generados a partir de la figura 4.8. El porcentaje de traslape representa aproximadamente 180 pixeles con cuadros de tamaño 1024x256 pixeles, generando un total de 8 cuadros. En la tabla 4.6 se muestran los datos de posicionamiento. En la figura 4.13 la gráfica de error de posicionamiento contra iteración y en la figura 4.14 la imagen resultante después aplicar el algoritmo de unión.

Tabla 4.6 Datos de posicionamiento de los cuadros generados de tamaño 1024x256 a partir de la figura 4.8 con un traslape del 70%

No. Iteración	Posicionamiento Teórico Eje y	Posicionamiento Experimental Eje y	Error (%)	Error Promedio (%)	Desviación Estándar del Error
1	0	0.000	0.000	0.048	0.042
2	76	76.104	0.137		
3	152	152.107	0.070		
4	228	228.074	0.032		
5	304	304.082	0.027		
6	380	380.132	0.035		
7	456	456.112	0.025		
8	532	532.308	0.058		

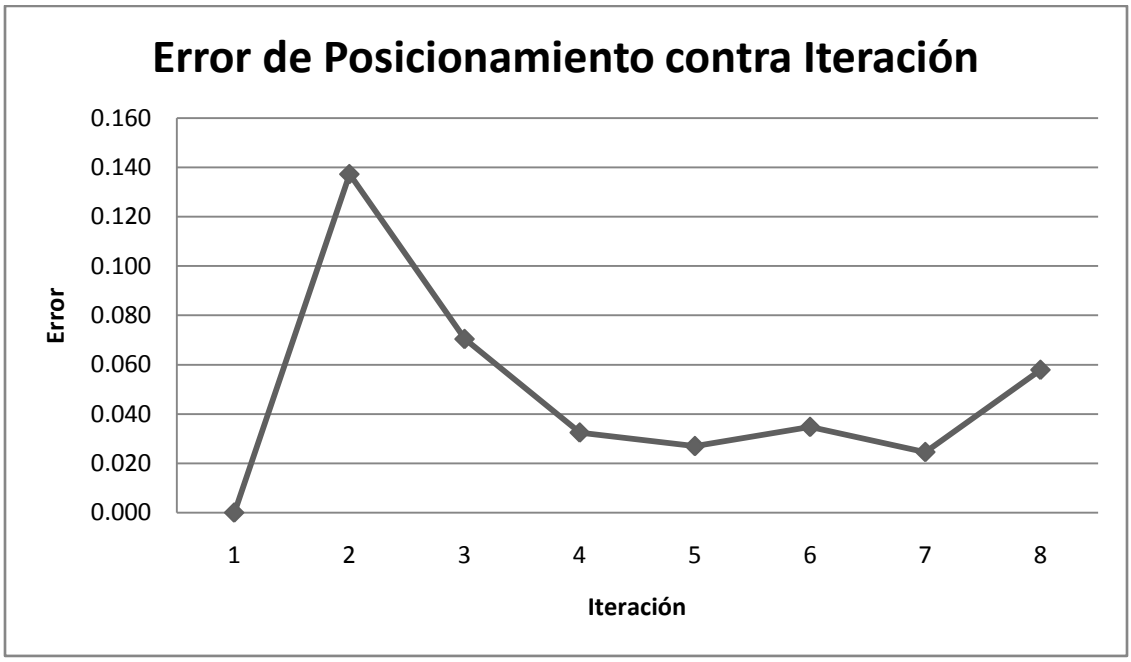

Figura 4.13 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.6

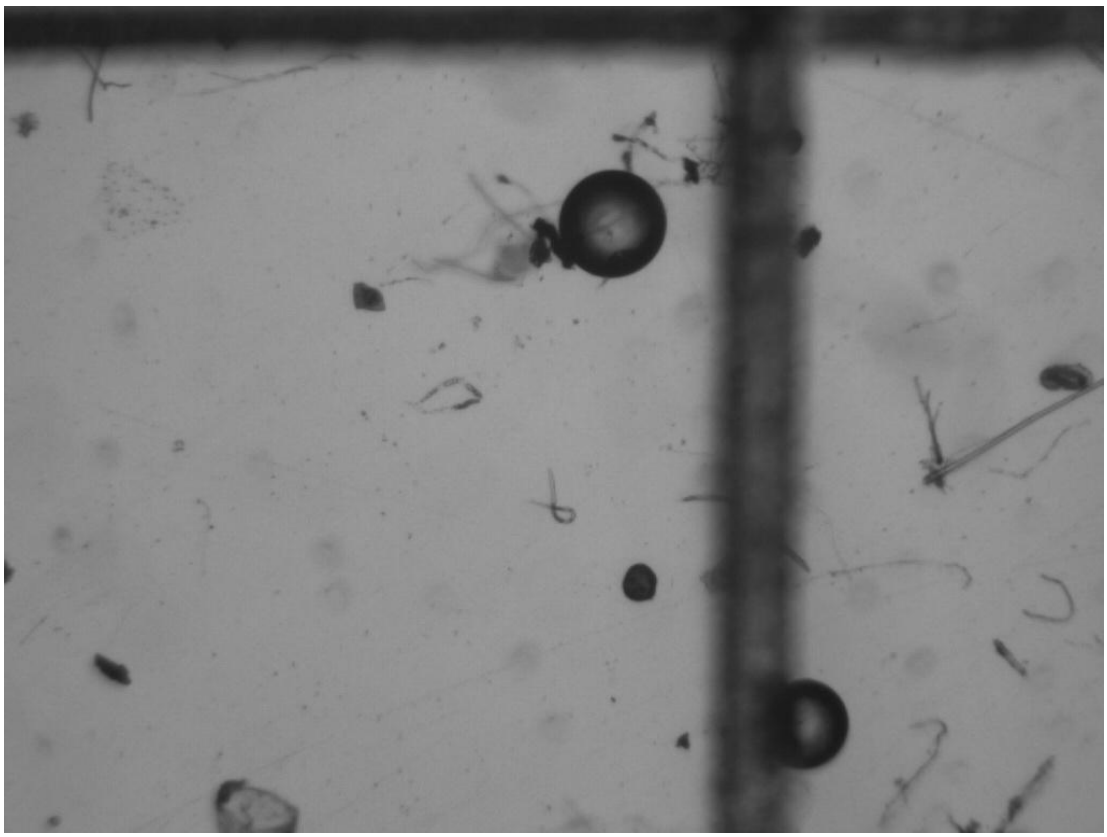

Figura 4.14 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.8 con un traslape en el eje y de 70%

Nuevamente con las reducción de traslape en el eje y no hay mucha diferencia en la tendencia del error con respecto a cada iteración del algoritmo en comparación a cuando se aplican porcentajes de traslape mayores. La imagen resultante se mantiene idéntica a la imagen de entrada.

En general, el comportamiento del algoritmo cuando se presenta traslape solamente en el eje y no es muy variante conforme se varíen los porcentajes de traslape, caso contrario a cuando hay traslape en el eje x. Esto se debe a que cada uno de los cuadros cuando hay solamente traslape en el eje y es de mayor área (1024x256 pixeles) que cuando se aplica sólo traslape en el eje x (256x768). Debido a esto, el algoritmo de detección de puntos puede localizar mayor cantidad de correspondencias cuando hay traslape sólo en el eje y, en comparación a cuando hay traslape sólo en el eje x, debido a que el primero presenta mayor área que el segundo.

4.1.3 Traslape en ambos ejes

A continuación se presentan los casos en los que hay traslapes en ambos ejes y se analizan los errores en los posicionamientos experimentales de los cuadros con respecto a los valores teóricos generados por el programa de partición de imágenes. Se presentan tres casos donde los traslapes entre los cuadros varían en 90%, 80% y 70%. Los tamaños de cada cuadro se establecieron en 512x512 pixeles, éste, es el tamaño menor que permite al algoritmo de unión de múltiples imágenes poder fusionar los cuadros de manera correcta en el marco final. Dimensiones menores a 512 pixeles causan que el algoritmo no encuentre las suficientes correspondencias entre los cuadros ocasionando que no se puedan alinear correctamente. La figura de entrada para el análisis de traslape en ambos ejes se muestra en la figura 4.15.

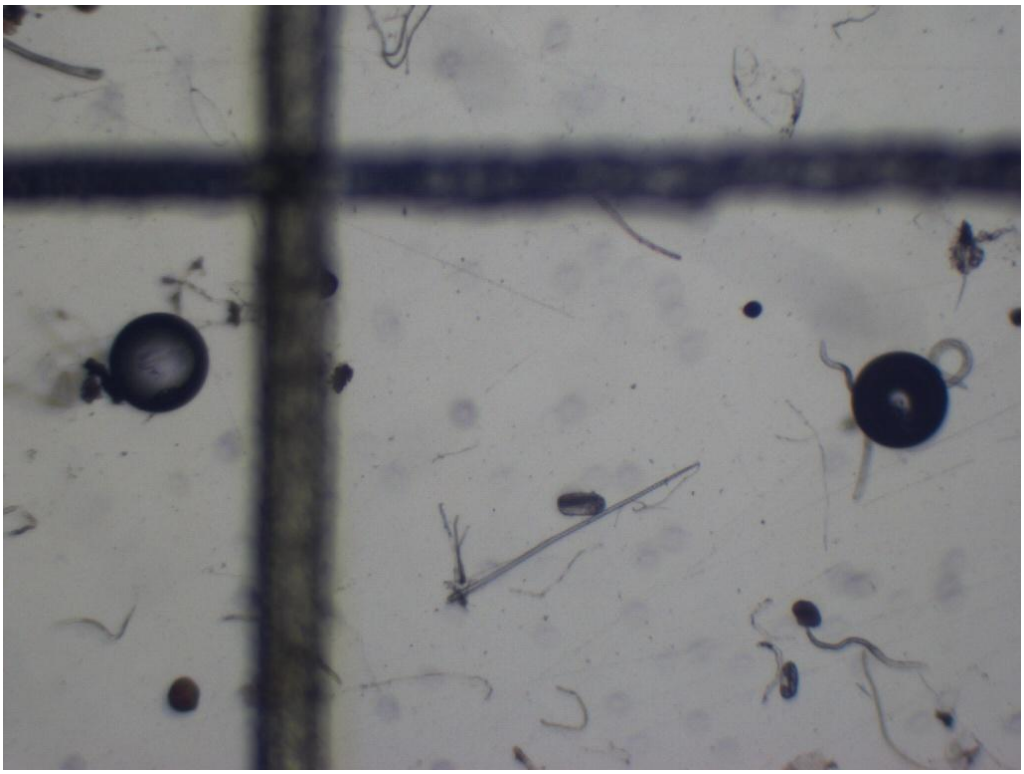

Figura 4.15 Imagen de entrada para el análisis con traslape en los ejes x e y

En el siguiente caso se muestra el resultado de aplicar un 90% de traslape en los ejes x e y, en los cuadros generados a partir de la figura 4.15. El porcentaje de traslape representa aproximadamente 470 pixeles con cuadros de tamaño 512x512 pixeles, generando un total de 112 cuadros. En la tabla 4.7 se muestran los datos de error promedio y desviación estándar calculado para cada eje. En la figura 4.16 la gráfica de error de posicionamiento contra iteración y en la figura 4.17 la imagen resultante después aplicar el algoritmo de unión.

Tabla 4.7 Datos de error de posicionamiento de los cuadros generados de tamaño 512x512 a partir de la figura 4.15 con un traslape del 90%

No. Iteraciones	Error Promedio (%)		Desviación Estándar	
	Eje X	Eje Y	Eje X	Eje Y
112	0.133	0.216	0.1177	0.234

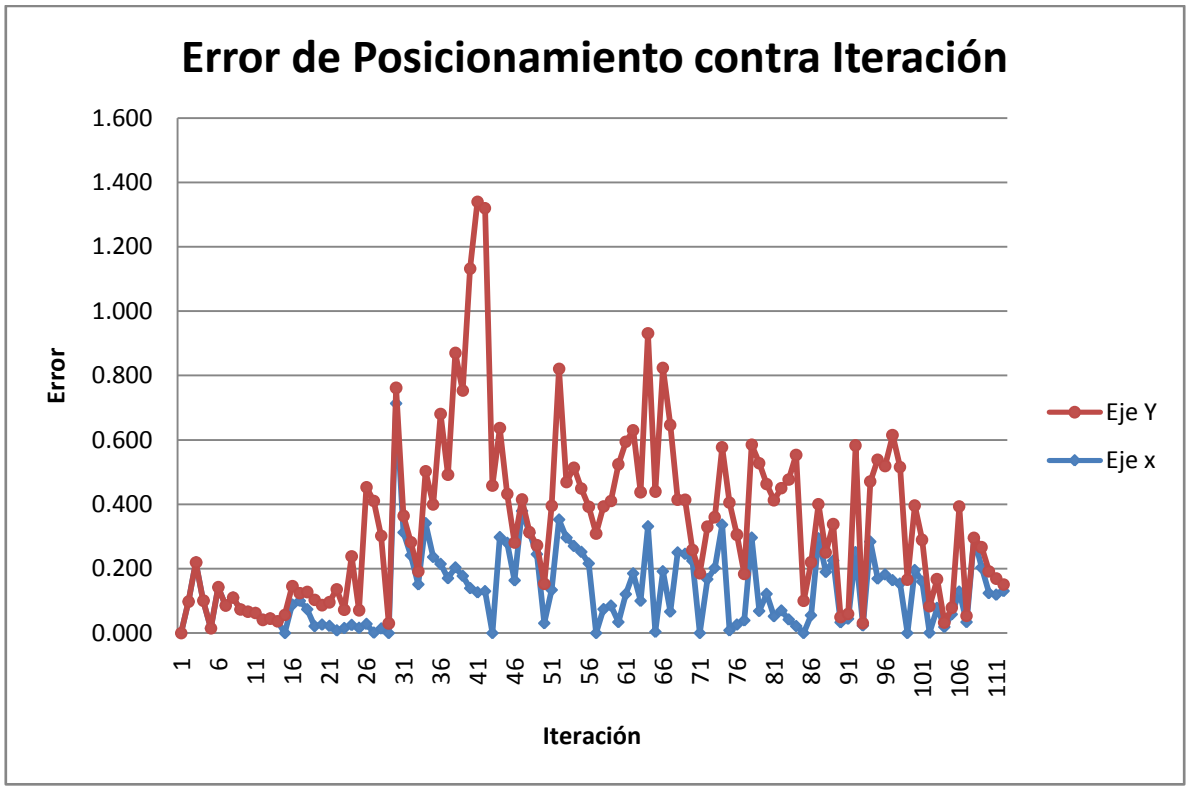

Figura 4.16 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.7

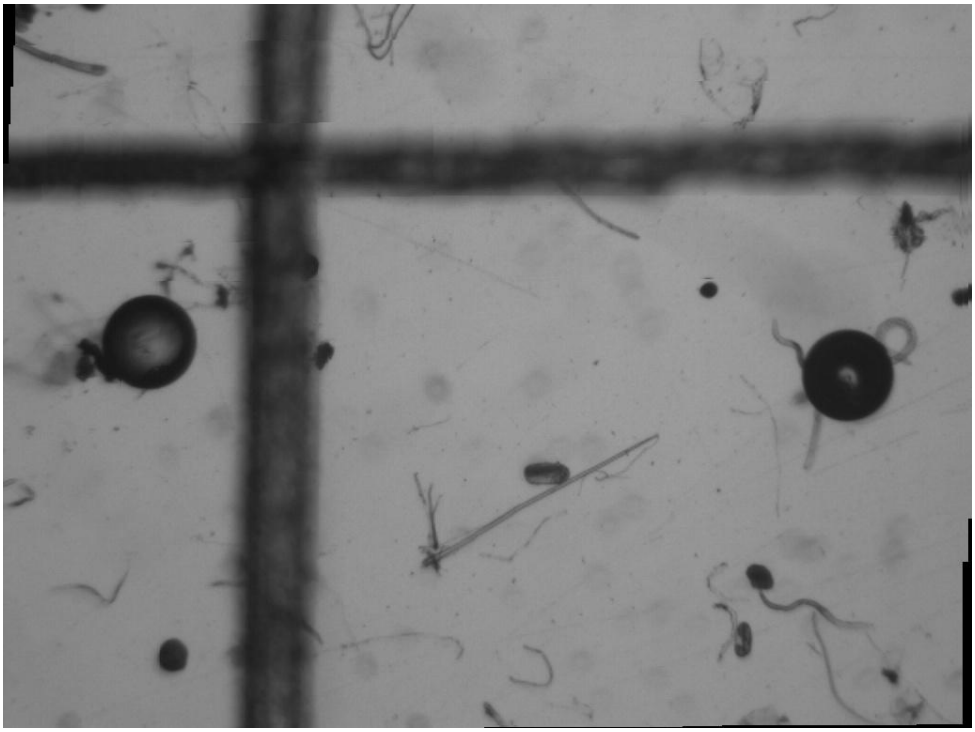

Figura 4.17 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.15 con un traslape en los ejes x e y de 90%

Como se aprecia en la figura 4.16, la tendencia del error de posicionamiento conforme avanza el algoritmo, cuando se tiene traslape de 90% en ambos ejes, tiende a mantenerse en un rango de valores alrededor del error promedio. No hay tendencia a disminuir como cuando se aplican traslapes en sólo uno de los ejes. Se puede ver que en los primeros cuadros que se unen el error es mucho menor que cuando la imagen resultante es mayor. La imagen resultante (figura 4.17) presenta ciertos desfases en algunas uniones cuando hay traslapes en ambos ejes, sin embargo, aún mantiene una integridad muy similar a la imagen de entrada.

En el siguiente caso se presenta el resultado de aplicar un 80% de traslape en los ejes x e y, en los cuadros generados a partir de la figura 4.15. El porcentaje de traslape representa aproximadamente 430 pixeles con cuadros de tamaño 512x512 pixeles, generando un total de 40 cuadros. En la tabla 4.8 se muestran los datos de error promedio y desviación estándar calculado para cada eje. En la figura 4.18 la gráfica de error de posicionamiento contra iteración y en la figura 4.19 la imagen resultante después aplicar el algoritmo de unión.

Tabla 4.8 Datos de error de posicionamiento de los cuadros generados de tamaño 512x512 a partir de la figura 4.15 con un traslape del 80%

No. Iteraciones	**Error Promedio (%)**		**Desviación Estándar**	
	Eje X	**Eje Y**	**Eje X**	**Eje Y**
40	0.205	0.456	0.1971	0.370

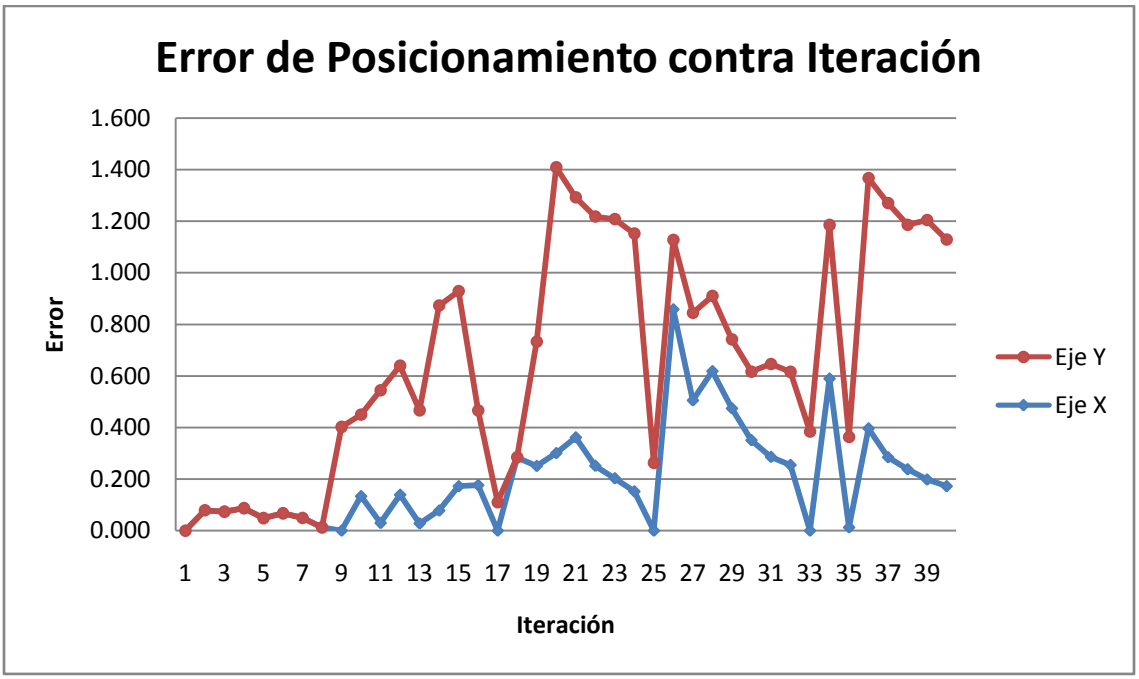

Figura 4.18 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.8

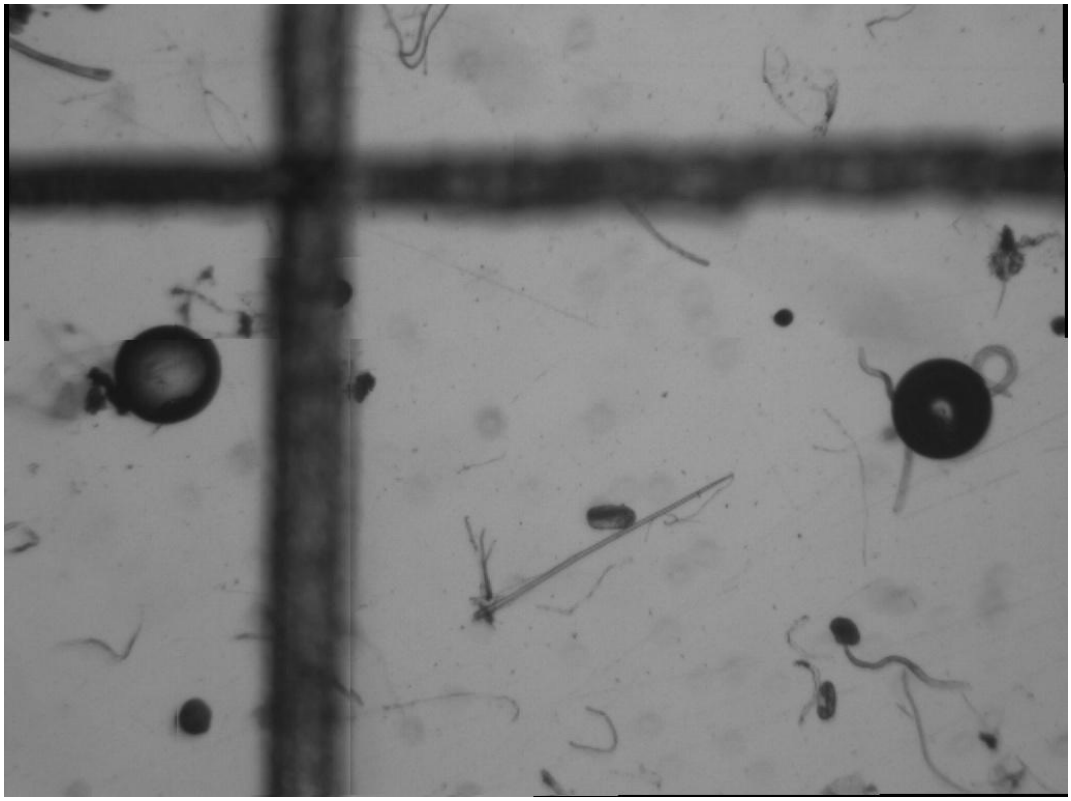

Figura 4.19 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.15 con un traslape en los ejes x e y de 80%

Con un traslape de 80% en ambos ejes, la tendencia del error conforme avanza el algoritmo se incrementa con cada iteración, especialmente en el eje y (ver figura 4.18). Esto ocasiona que se vean algunas uniones entre los cuadros en la imagen resultante, tal y como se aprecia en la figura 4.19.

En el siguiente caso presenta el resultado de aplicar un 70% de traslape en los ejes x e y, en los cuadros generados a partir de la figura 4.15. El porcentaje de traslape representa aproximadamente 360 pixeles con cuadros de tamaño 512x512 pixeles, generando un total de 15 cuadros. En la tabla 4.9 se muestran los datos de error promedio y desviación estándar calculado para cada eje. En la figura 4.20 la gráfica de error de posicionamiento contra iteración y en la figura 4.21 la imagen resultante después aplicar el algoritmo de unión.

Tabla 4.9 Datos de error de posicionamiento de los cuadros generados de tamaño 512x512 a partir de la figura 4.15 con un traslape del 70%

No. Iteraciones	Error Promedio (%)		Desviación Estándar	
	Eje X	Eje Y	Eje X	Eje Y
15	0.083	0.094	0.1260	0.137

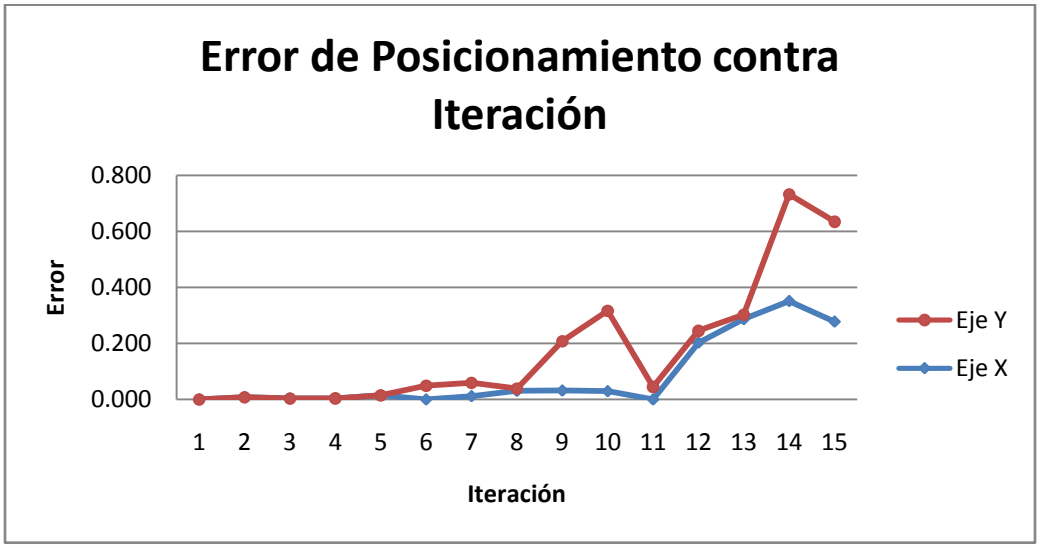

Figura 4.20 Gráfica de error de posicionamiento contra iteración obtenida de la tabla 4.9

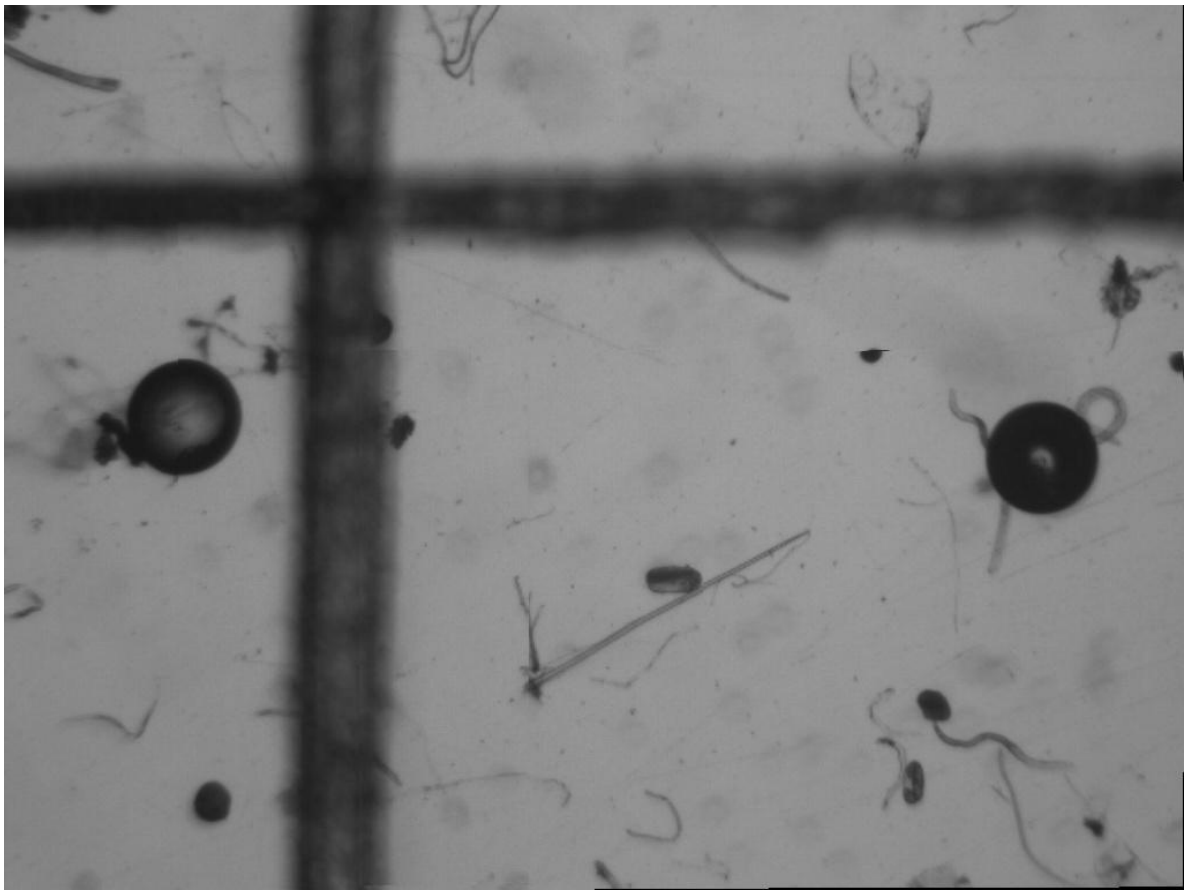

Figura 4.21 Imagen resultante de la unión de los cuadros generados a partir de la figura 4.15 con un traslape en los ejes x e y de 70%

Para el caso del traslape del 70% en ambos ejes la tendencia del error conforme avanza el algoritmo se mantiene similar a cuando se aplican traslapes mayores, es decir, se mantiene bajo en las primeras iteraciones y luego comienza a incrementarse, sin embargo, como la cantidad de cuadros para este caso es menor en comparación con los casos anteriores, ocasiona que error promedio disminuya pues la cantidad de iteraciones son menores. De ahí que la imagen resultante mostrada en la figura 4.21 presenta una mayor similitud a la imagen de entrada en comparación a cuando se aplican traslapes mayores al 70%.

En general, cuando se aplican traslapes en ambos ejes, las primeras iteraciones producidas por el algoritmo de unión presentan errores menores en comparación a las iteraciones finales. Esto ocasiona que con una cantidad mayor de iteraciones el error promedio aumente y se produzca la visibilidad de las

divisiones existentes entre un cuadro y otro. Esto se puede reducir aumentando el traslape o el tamaño de los cuadros que conforman la imagen final.

4.2 Reutilización de los puntos de interés

Como se explicó en el marco teórico, el algoritmo de fusión de múltiples imágenes toma en cuenta la reutilización de datos calculados con el fin de reducir el tiempo de procesamiento a la hora de unir todos los cuadros necesarios. A continuación se presentan tres resultados hechos con muestras reales de nematodos. Se utilizó el programa de partición de imágenes para generar cuadros de 512x512 pixeles con traslape en ambos ejes a partir de una imagen de entrada mostrada en la figura 4.22.

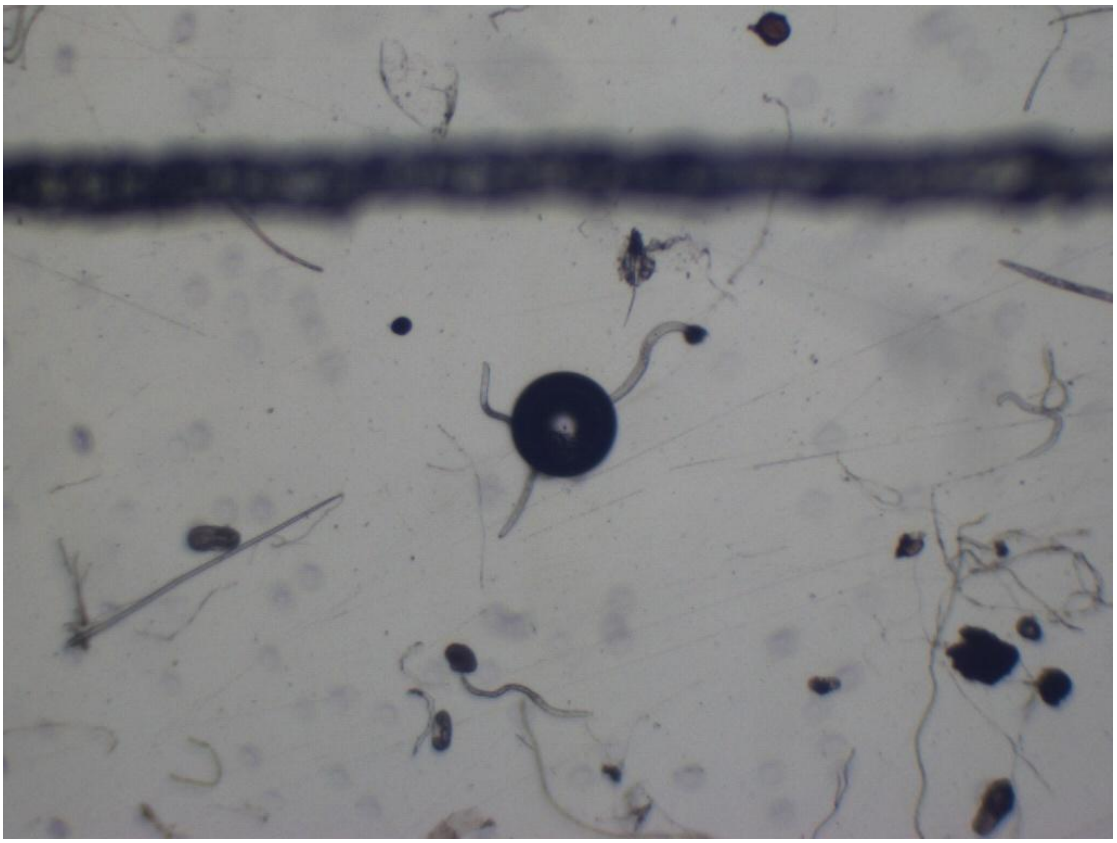

Figura 4.22 Imagen de entrada para el análisis de reducción de tiempo

En el primer caso presenta el resultado de aplicar un 90% de traslape en los ejes x e y, en los cuadros generados a partir de la figura 4.22. El porcentaje de traslape representa aproximadamente 470 pixeles con cuadros de tamaño 512x512 pixeles, generando un total de 112 cuadros. En la tabla 4.10 se muestran los datos de tiempo generado cuando se aplica el algoritmo de reutilización de datos y cuando no se aplica, así como el porcentaje de reducción de tiempo. En la figura 4.23 se muestra la imagen resultante sin aplicar el algoritmo de reutilización y en la figura 4.24 la imagen resultante cuando se aplica dicho algoritmo.

Tabla 4.10 Datos de tiempo al unir los cuadros generados de tamaño 512x512 a partir de la figura 4.22 con un traslape del 90%

No. Iteraciones	Tiempo sin la reutilización de datos (s)	Tiempo con la reutilización de datos (s)	Porcentaje de reducción (%)
112	21.2	15.8	25.5

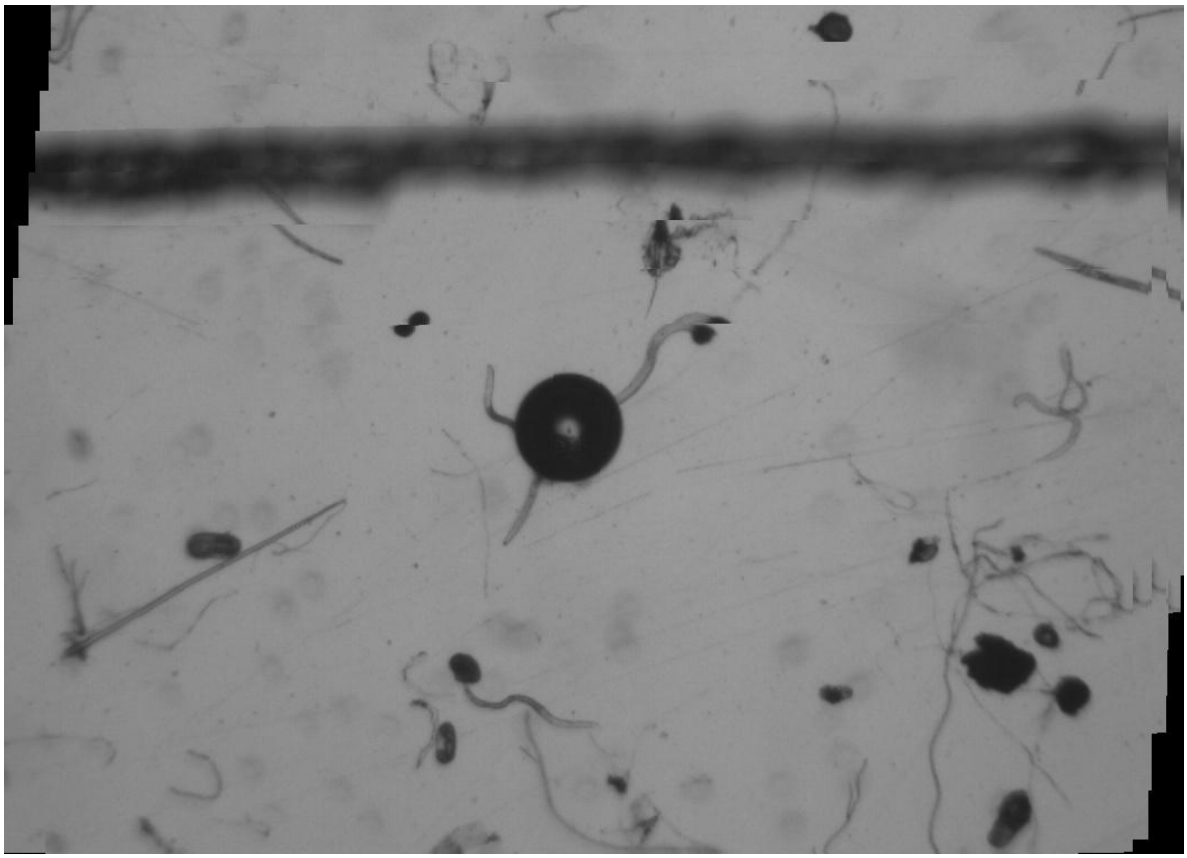

Figura 4.23 Unión resultante de los cuadros generados a partir de la figura 4.22 con un traslape de 90% en ambos ejes sin reutilización de datos

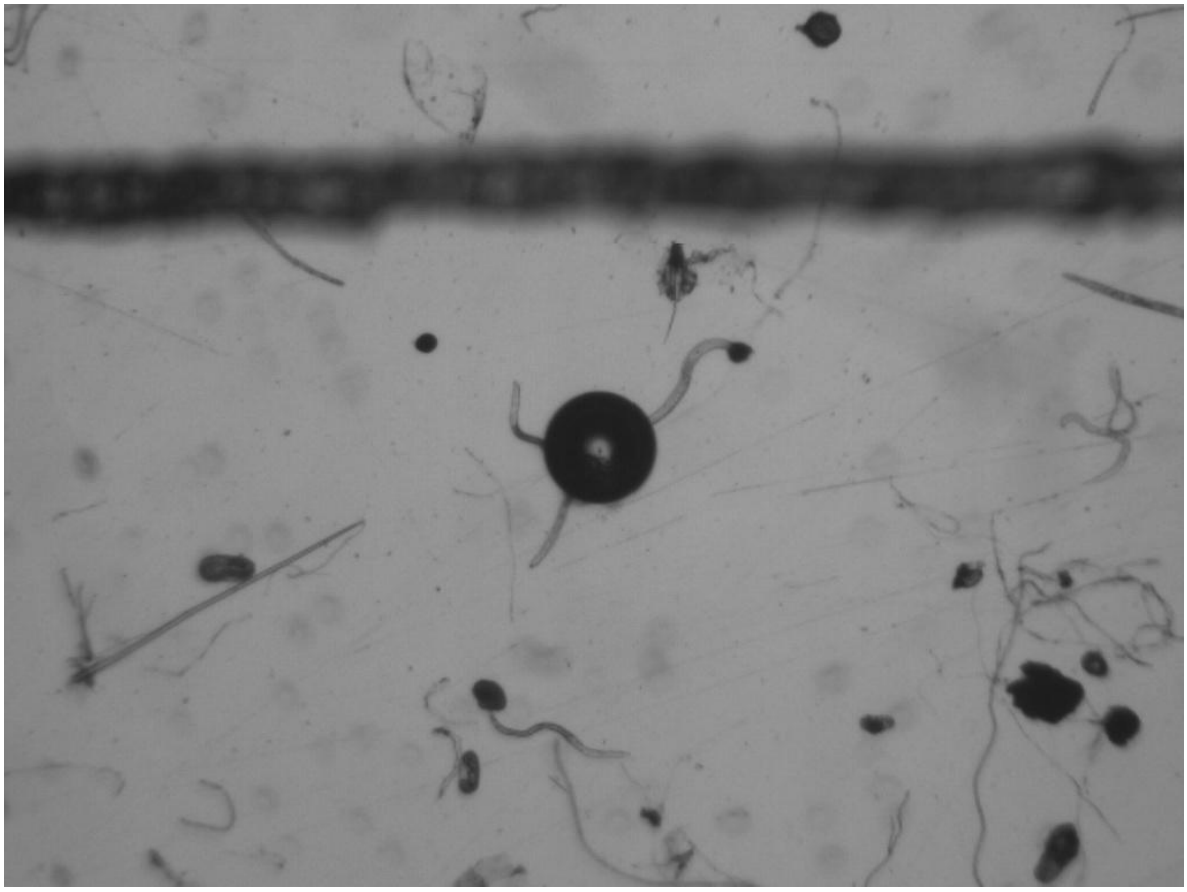

Figura 4.24 Unión resultante de los cuadros generados a partir de la figura 4.22 con un traslape de 90% en ambos ejes con reutilización de datos

Como se aprecia en la tabla 4.10 al implementar la reutilización de datos en el algoritmo de unión de múltiples imágenes se presenta una reducción del tiempo, que para el caso en que hay traslape de 90% en ambos ejes de 25.5%. Además, otra ventaja de reutilizar los datos es que la calidad en la unión mejora considerablemente, tal y como se muestra en las figuras 4.23 y 4.24.

En el siguiente caso se presenta el resultado de aplicar un 80% de traslape en los ejes x e y, en los cuadros generados a partir de la figura 4.22. El porcentaje de traslape representa aproximadamente 430 pixeles con cuadros de tamaño 512x512 pixeles, generando un total de 40 cuadros. En la tabla 4.11 se muestran los datos de tiempo generado cuando se aplica el algoritmo de reutilización de

datos y cuando no se aplica, así como el porcentaje de reducción de tiempo. En la figura 4.25 se muestra la imagen resultante sin aplicar el algoritmo de reutilización y en la figura 4.26 la imagen resultante cuando se aplica dicho algoritmo.

Tabla 4.11 Datos de tiempo al unir los cuadros generados de tamaño 512x512 a partir de la figura 4.22 con un traslape del 80%

No. Iteraciones	Tiempo sin la reutilización de datos (s)	Tiempo con la reutilización de datos (s)	Porcentaje de reducción (%)
40	7.5	4.5	40

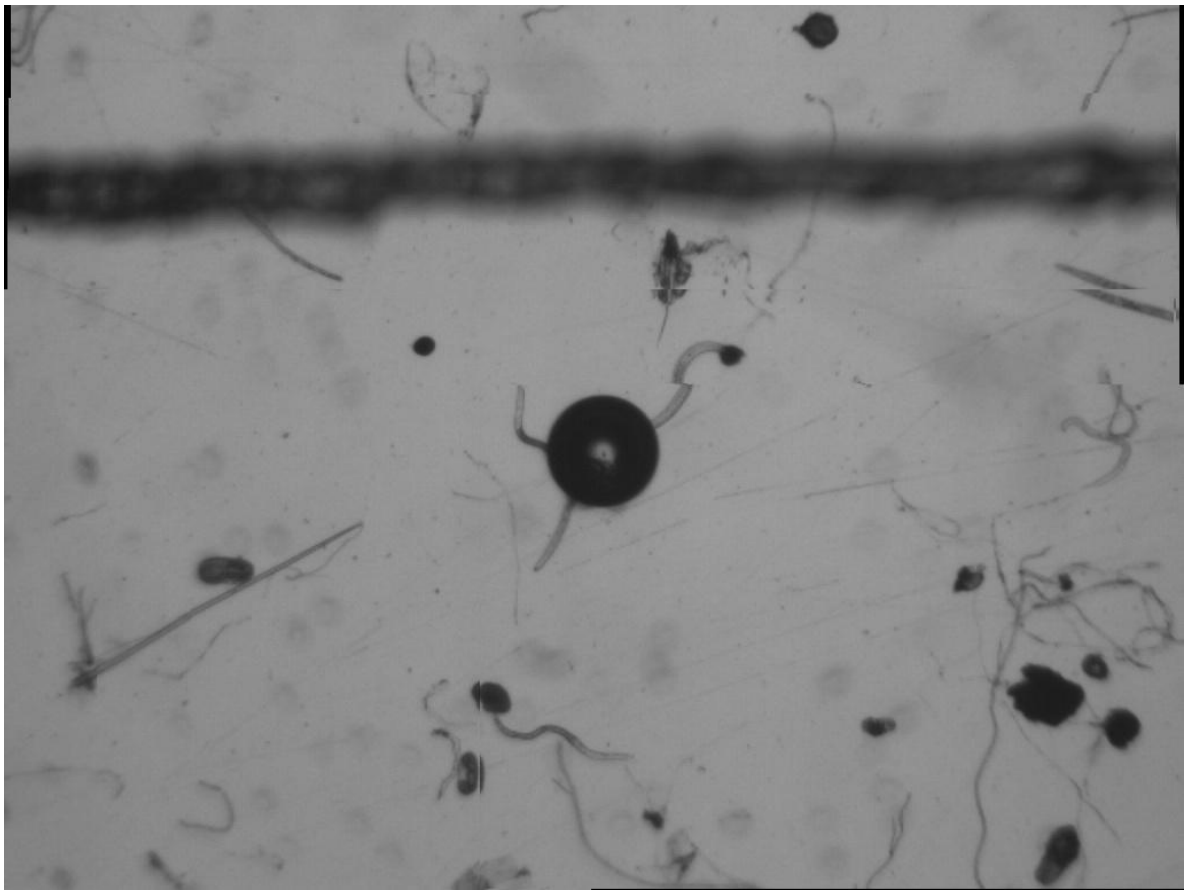

Figura 4.25 Unión resultante de los cuadros generados a partir de la figura 4.22 con un traslape de 80% en ambos ejes sin reutilización de datos

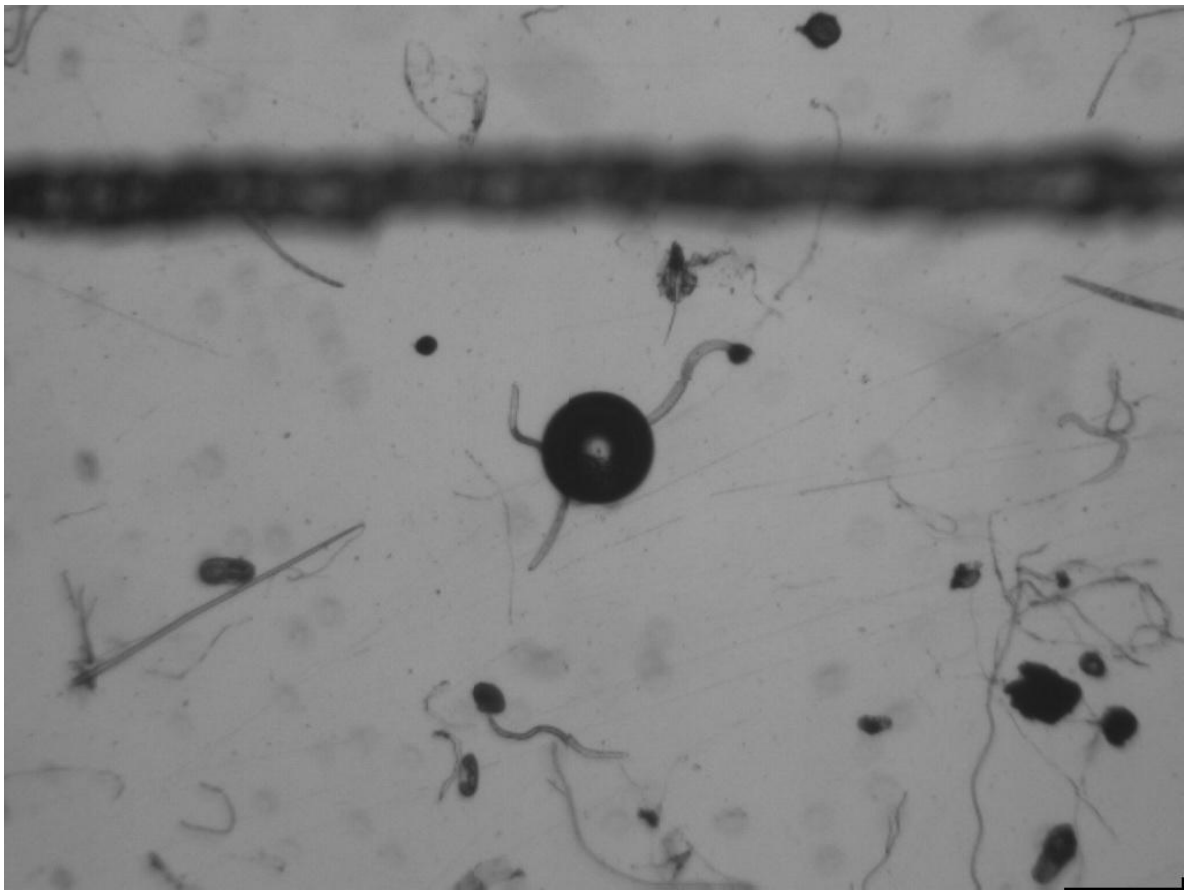

Figura 4.26 Unión resultante de los cuadros generados a partir de la figura 4.22 con un traslape de 80% en ambos ejes con reutilización de datos

Para el caso del traslape de un 80% el porcentaje de reducción de tiempo es mayor en comparación con el traslape del 90%. La imagen resultante cuando no se aplica el algoritmo de reutilización de datos presenta mayor similitud a la imagen de entrada, pero siempre denota algunos desfases entre las uniones. Sin embargo, esto se corrige al implementar el algoritmo de reutilización tal y como se ve en la figura 4.26.

Por último se presenta el caso en donde se aplica un 70% de traslape en los ejes x e y, en los cuadros generados a partir de la figura 4.22. El porcentaje de traslape representa aproximadamente 360 pixeles con cuadros de tamaño 512x512 pixeles, generando un total de 15 cuadros. En la tabla 4.12 se muestran los datos de tiempo generado cuando se aplica el algoritmo de reutilización de

datos y cuando no se aplica, así como el porcentaje de reducción de tiempo. En la figura 4.27 se muestra la imagen resultante sin aplicar el algoritmo de reutilización y en la figura 4.28 la imagen resultante cuando se aplica dicho algoritmo.

Tabla 4.12 Datos de tiempo al unir los cuadros generados de tamaño 512x512 a partir de la figura 4.22 con un traslape del 70%

No. Iteraciones	Tiempo sin la reutilización de datos (s)	Tiempo con la reutilización de datos (s)	Porcentaje de reducción (%)
15	2.5	1.3	48

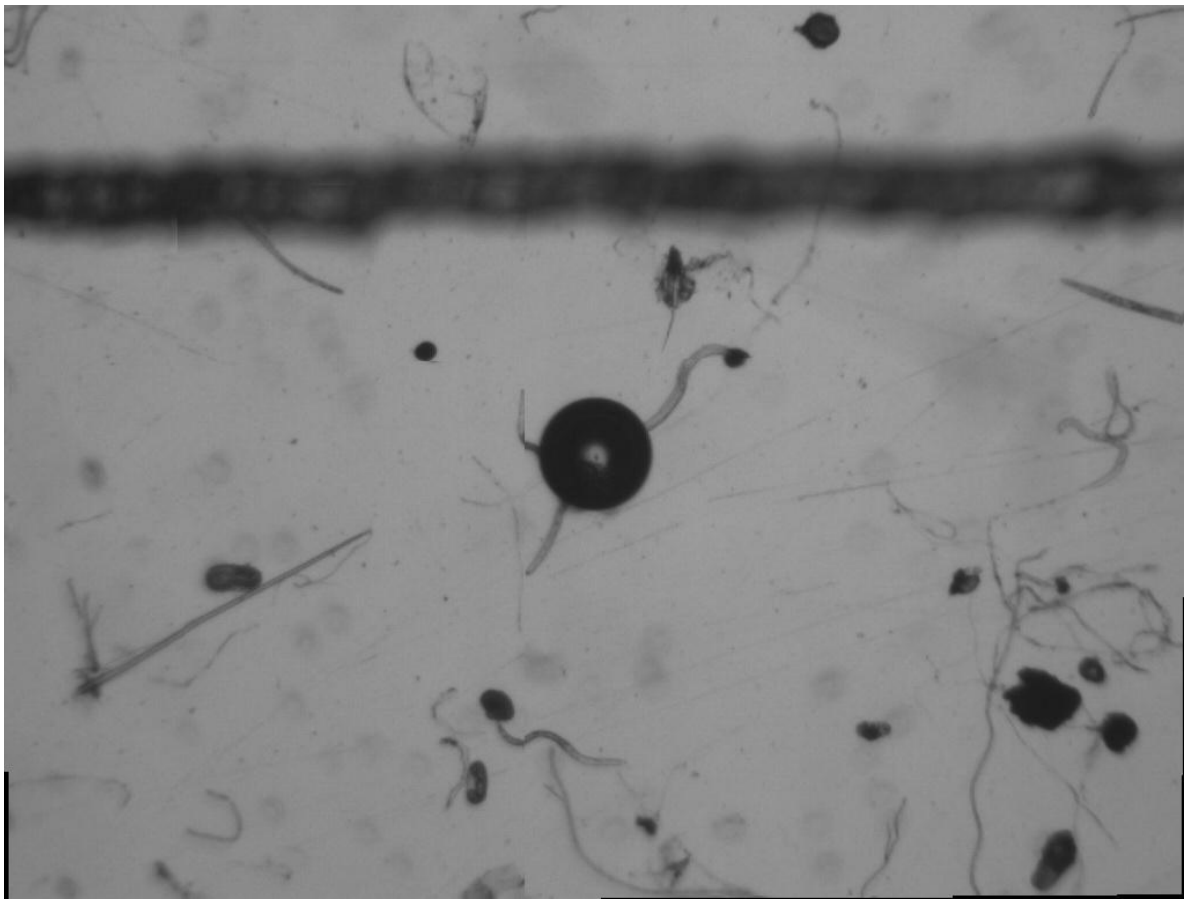

Figura 4.27 Unión resultante de los cuadros generados a partir de la figura 4.22 con un traslape de 70% en ambos ejes sin reutilización de datos

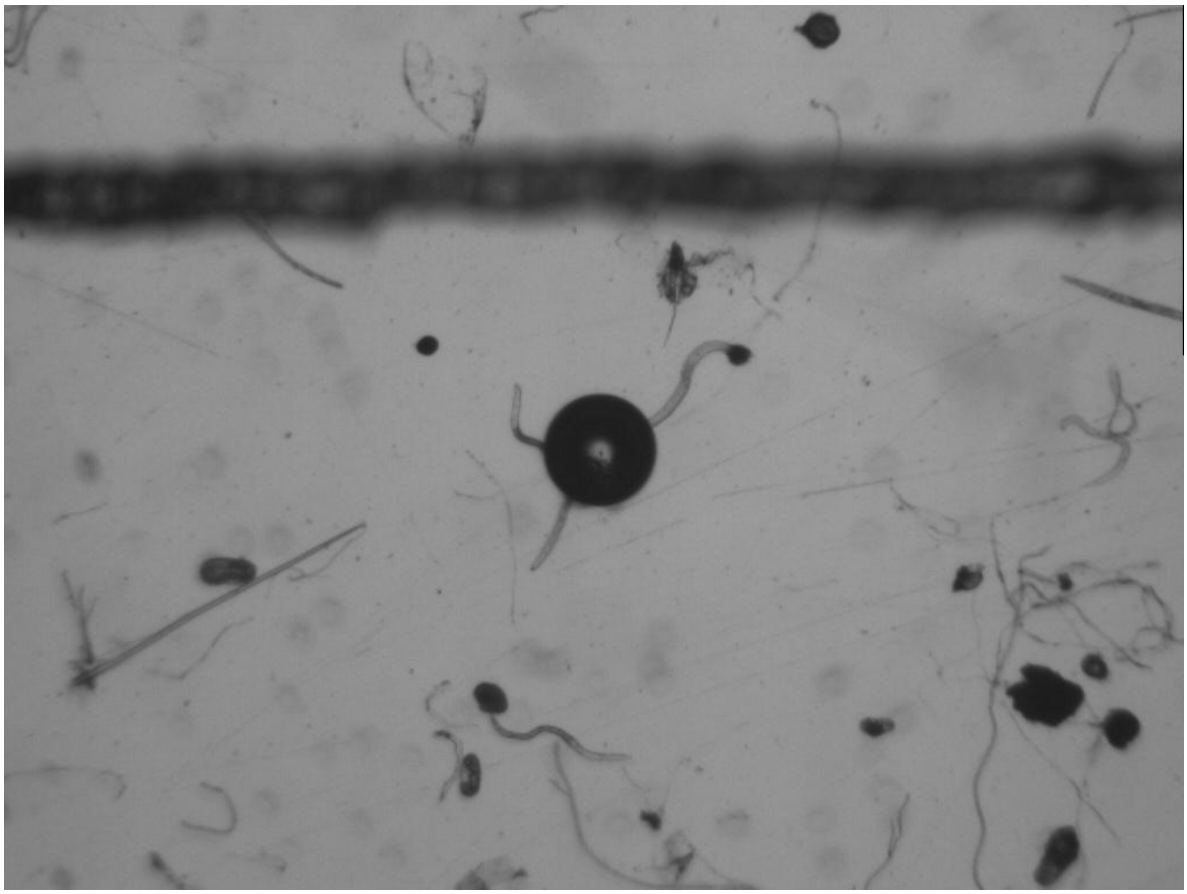

Figura 4.28 Unión resultante de los cuadros generados a partir de la figura 4.22 con un traslape de 70% en ambos ejes con reutilización de datos

Con un 70% de traslape en ambos ejes se generan menos cuadros que unir y ello ocasiona que el algoritmo de unión de imágenes calcule menos datos si se aplica la reutilización, ocasionando que el porcentaje de reducción de tiempo aumente en comparación con los traslapes mayores presentados anteriormente. En la figura 4.27 se muestra que existen desfases en las secciones inferiores de la imagen resultante cuando no se aplica el algoritmo de reducción, sin embargo, al aplicar el algoritmo de reutilización estos desfases desaparecen dejando la imagen muy similar a la imagen de entrada, tal y como se muestra en la figura 4.28.

4.3 Unión de múltiples imágenes en casos reales

A continuación se presentan 3 ejemplos de unión de múltiples imágenes utilizando una cantidad mayor de cuadros de tomas reales de nematodos. Cada cuadro tiene un tamaño de 1024x768 pixeles. Se analizan las diferencias de unir todos los cuadros sin implementar el algoritmo de reutilización de datos e implementándolo, para así determinar las diferencias en tiempo y calidad al utilizar una muestra de mayor tamaño que las mostradas en las secciones anteriores de este capítulo.

En la figura 4.29 se observa el resultado de unir 50 cuadros de una muestra de nematodos sin la reutilización de datos y en la figura 4.30 se expone la unión de la misma muestra pero implementando el algoritmo de reutilización.

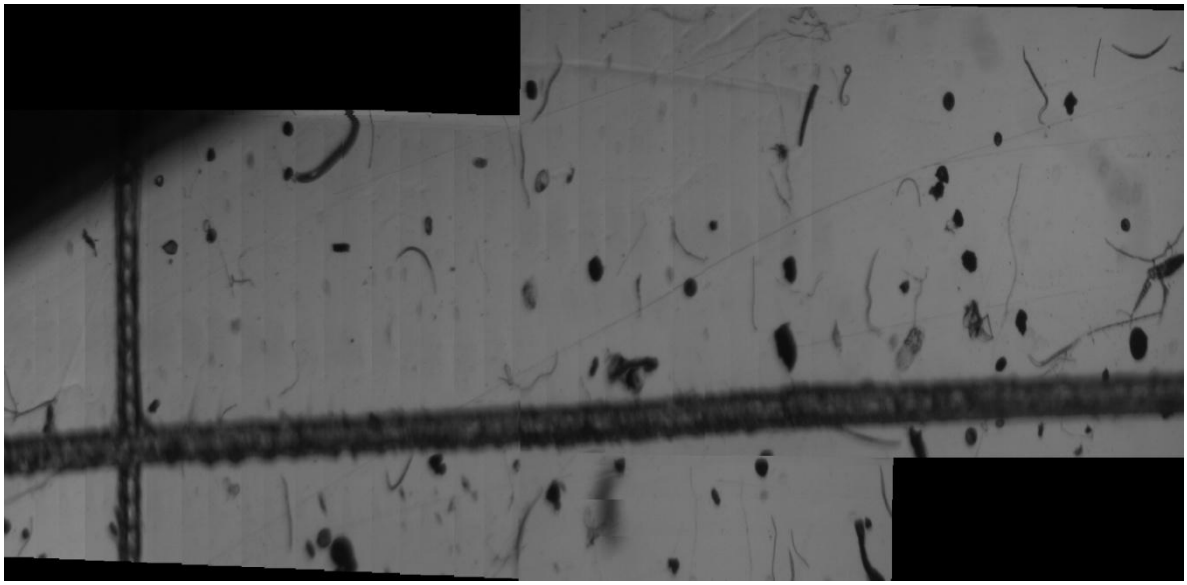

Figura 4.29 Unión resultante de 50 cuadros de una muestra de nematodos sin la reutilización de datos

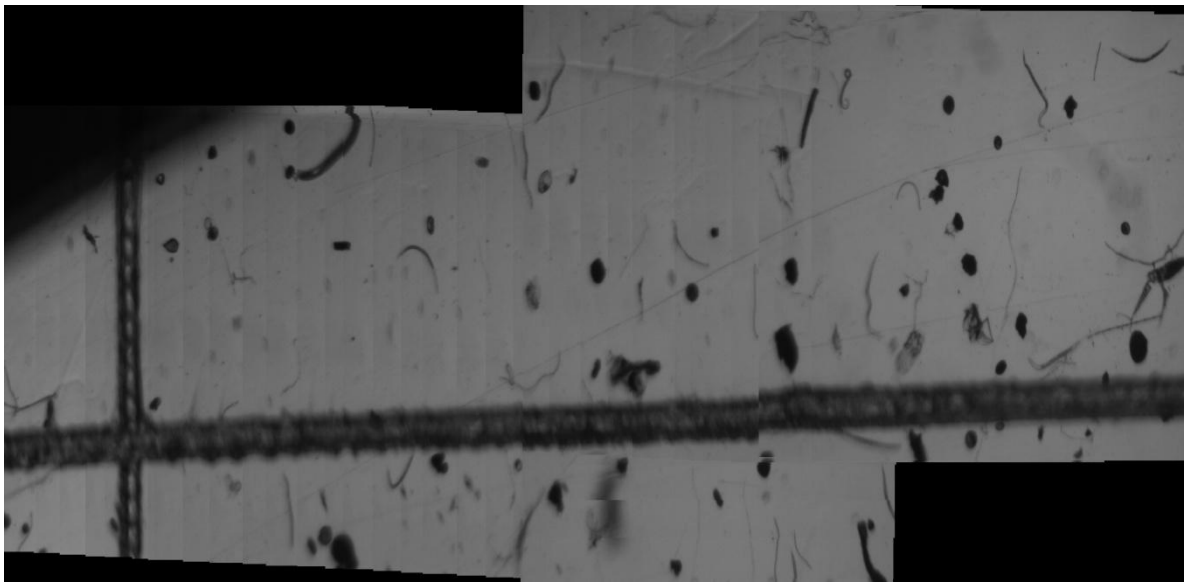

Figura 4.30 Unión resultante de 50 cuadros de una muestra de nematodos con la reutilización de datos

En la tabla 4.13 se muestran los resultados del tiempo de procesamiento obtenidos de las figuras 4.29 y 4.30.

Tabla 4.13 Resultados del tiempo de procesamiento de las uniones mostradas en las figuras 4.29 y 4.30

No. Iteraciones	Tiempo sin la reutilización de datos	Tiempo con la reutilización de datos	Porcentaje de reducción (%)
50	27.29 s	23.46 s	14

Como se aprecia en la tabla 4.13 existe una reducción en tiempo de un 14% en la unión de los 50 cuadros. Sin embargo, se puede visualizar una pérdida de calidad en algunas uniones cuando se implementa el algoritmo de reutilización de datos. Las dimensiones de la figura 4.29 son de 3245x991 pixeles, mientas que las dimensiones de la figura 4.30 son de 3244x993 pixeles. Existe una diferencia en estas dimensiones lo cual puede significar que haya diferencias en los valores de posicionamiento entre los cuadros cuando se aplica el algoritmo de reutilización y cuando no se aplica. Con esto puede verse afectada la calidad de las uniones entre los cuadros.

A continuación se presentan dos casos más de unión de múltiples imágenes. El primer caso es con una muestra compuesta por 60 cuadros y el segundo caso con una muestra compuesta por 155 imágenes.

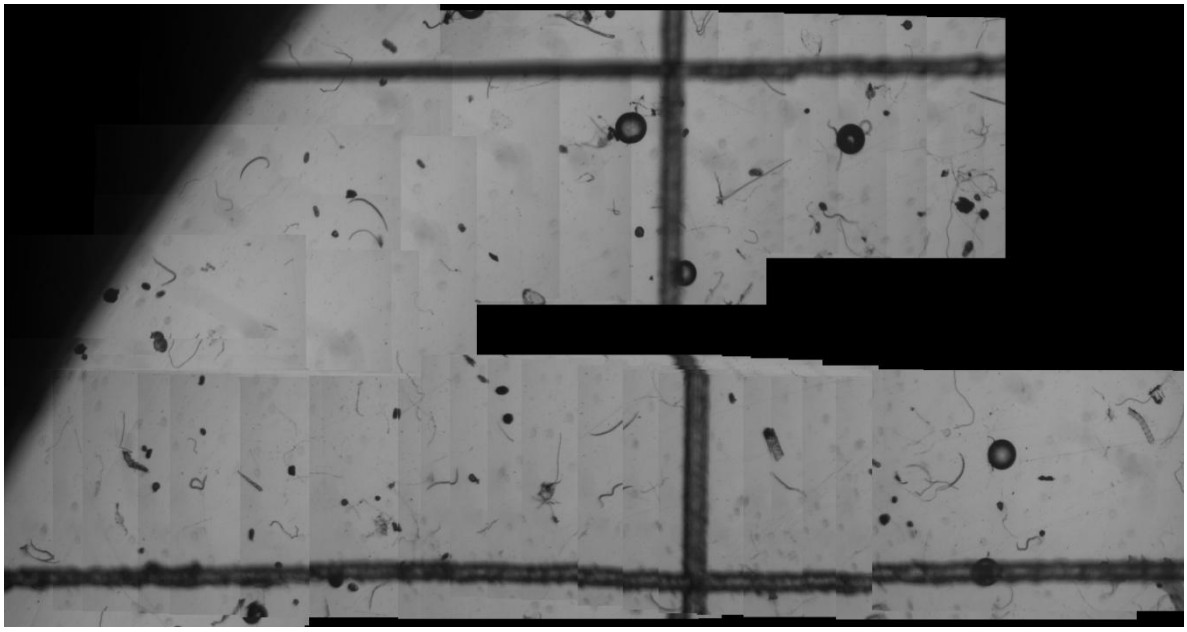

Figura 4.31 Unión resultante de 60 cuadros de una muestra de nematodos sin la reutilización de datos

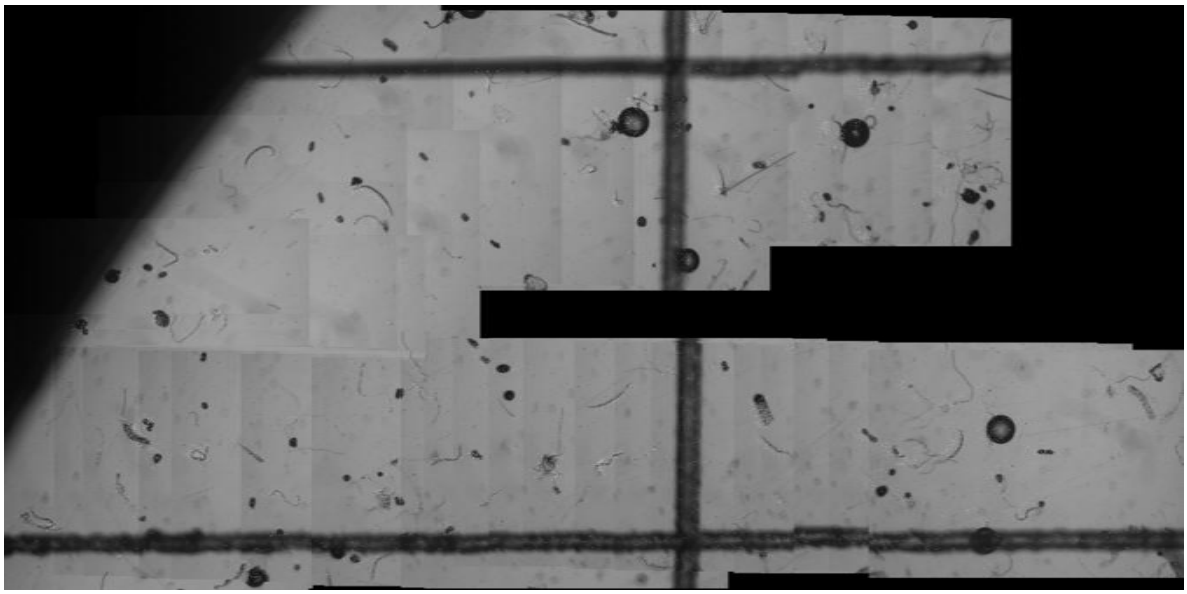

Figura 4.32 Unión resultante de 60 cuadros de una muestra de nematodos con la reutilización de datos

En la tabla 4.14 se muestran los resultados del tiempo de procesamiento obtenidos de las figuras 4.31 y 4.32. El porcentaje de reducción en el tiempo es de un 10%.

Tabla 4.14 Resultados del tiempo de procesamiento de las uniones mostradas en las figuras 4.31 y 4.32

No. Iteraciones	Tiempo sin la reutilización de datos	Tiempo con la reutilización de datos	Porcentaje de reducción (%)
60	53.13 s	47.79 s	10

Figura 4.33 Unión resultante de 155 cuadros de una muestra de nematodos sin la reutilización de datos

Figura 4.34 Unión resultante de 155 cuadros de una muestra de nematodos con la reutilización de datos

En la tabla 4.15 se muestran los resultados del tiempo de procesamiento obtenidos de las figuras 4.33 y 4.34. El porcentaje de reducción en el tiempo es de un 15%.

Tabla 4.15 Resultados del tiempo de procesamiento de las uniones mostradas en las figuras 4.33 y 4.34

No. Iteraciones	Tiempo sin la reutilización de datos	Tiempo con la reutilización de datos	Porcentaje de reducción (%)
155	2 min y 39 s	2 min y 15 s	15

Como se puede observar en los resultados planteados anteriormente, éstos son los esperados en lo que respecta al tiempo cuando se aplica la reutilización de datos, pues se nota una reducción de al menos un 10%, sin embargo, el mejoramiento en la calidad no es tan significativo como cuando se hicieron las pruebas y análisis en los cuadros generados por el programa de partición (sección 4.2). Dos de los factores que influyen en estos resultados son los valores de traslape en ambos ejes, así como los tamaños de los cuadros, pues con dimensiones mayores en las imágenes se debe realizar mayor cantidad de análisis y por ende puede existir una mayor probabilidad de error de posicionamiento.

También existen cambios en las dimensiones de las imágenes resultantes cuando se aplica el algoritmo de reutilización y cuando no se aplica, pues la figura 4.31 mide 4018x1991 pixeles, mientras que la figura 4.32 tiene un tamaño de 3977x1971 pixeles. Asimismo, las dimensiones de la figura 4.33 son de 6331x1228 pixeles y las de la figura 4.34 son de 6447x1229. Esto denota diferencias en las posiciones de cada cuadro lo cual afecta la calidad de las uniones que forman la imagen final. Aún así, la integridad de la imagen resultante permite visualizar una muestra más completa de nematodos con o sin la aplicación del algoritmo de reutilización de datos.

Capítulo 5: Conclusiones y Recomendaciones

5.1 Conclusiones

Para el correcto funcionamiento del algoritmo de unión de múltiples imágenes es necesario reajustar el marco que contiene la unión con base en la información brindada por el algoritmo RANSAC y el reajuste de los puntos característicos al nuevo sistema de coordenadas del marco final.

Las imágenes que se desean unir para formar el cuadro resultante deben tener dimensiones y traslapes tales que la cantidad de correspondencias calculadas permita el correcto registro entre un cuadro y otro.

Cuando se implementan traslapes en ambos ejes los errores de posicionamiento en los cuadros que conforman la imagen resultante son menores en las primeras iteraciones y comienzan a incrementarse conformen se adicionan nuevas imágenes al marco resultante. Sin embargo, este error puede controlarse aumentando la dimensión de los cuadros a unir y/o el traslape entre ellos.

El algoritmo de unión de múltiples imágenes implementa un ciclo en el cual se analizan los cuadros que se desean unir. En cada ensamble se calculan los diversos parámetros necesarios para la adición. Debido a esto, el incremento en los análisis y cálculos de estos datos se dan de manera polinomial, ya que, cada vez que se adhiere una imagen al marco resultante los algoritmos deben procesar aún más información que en el ensamble anterior. Esto ocasiona un gasto de tiempo considerable durante el procesamiento.

Se hizo un análisis estudiando las propiedades de los puntos característicos y de los descriptores. De este estudio se llegó a la conclusión que en cada empalme de dos imágenes los puntos y descriptores generados pueden ser almacenados en una estructura de datos para poder usarse en el siguiente

ensamble y así, evitar el cálculo repetitivo de los mismos y obligar al algoritmo a calcular solamente aquellos valores que pertenecen a la imagen nueva que se está adhiriendo.

Se comprobó que al reutilizar los valores de los puntos característicos y los descriptores en cada ensamble de dos imágenes se produce una reducción del tiempo de procesamiento en la fusión de múltiples imágenes y un mejoramiento en la calidad de las uniones imagen resultante.

5.2 Recomendaciones

Una de las recomendaciones es mejorar el desempeño del algoritmo que reutiliza los puntos de interés y descriptores en cada unión de dos imágenes con el fin de ahorrar el tiempo de procesamiento. Esto debido a que cada punto característico debe ser reajustado en cuanto a sus valores de traslación, rotación y escala para que pueda ser válido en el siguiente ensamble. Al hacer los cálculos para el reajuste se debe recorrer toda la estructura que contienen los puntos y asignar los nuevos valores. Esto ocasiona una pérdida de tiempo que va creciendo polinomialmente conforme se vayan uniendo más cuadros. El mejoramiento de esta parte del algoritmo influye positivamente en una reducción del tiempo de duración del procesamiento aún más considerable que la que se obtuvo experimentalmente.

Bibliografía

[1] Alfaro, Fabián. “Composición de Múltiples Imágenes de Microscopía”. Tesis, Instituto Tecnológico de Costa Rica. Junio, 2008.

[2] Boullosa, Oscar. “Estudio comparativo de descriptores visuales para la detección de escenas cuasi-duplicadas”. Tesis de Licenciatura, Universidad Autónoma de Madrid. 2011.

[3] Cimmyt.org [en línea]: Nematodos. <http://www.cimmyt.org/spanish/docs/field_guides/enfplagastrigo/Nematodos.pdf> [Consulta: 26 Mayo 2010]

[4] Fischlet, M. y Bolles, R. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. Communications of the ACM. v.24 (6): 381-395, jun., 1981.

[5] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” in Proceedings of the ninth European Conference on Computer Vision, May 2006.

[6] Peraza, Juan Manuel. “Estimación de la distancia recorrida por un robot móvil mediante la utilización de descriptores SURF”. Tesis de Licenciatura, Universidad Carlos III de Madrid. 2009.

[7] Sourceforge.net. *C++ Computer Vision & Robotics Library* [en línea]. <http://sourceforge.net/projects/cvrlib/> [Consulta: 27 may. 2011].