

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



**Ubicación de un nematodo en imágenes digitales utilizando modelos activos
de forma**

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

Juan Esteban Marín Hernández

Cartago, 9 de noviembre de 2009

INSTITUTO TECNOLÓGICO DE COSTA RICA

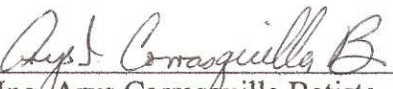
ESCUELA DE INGENIERÍA ELECTRÓNICA

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.

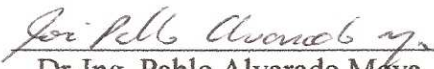
Miembros del Tribunal


Ing. Arys Carrasquilla Batista

Profesora lectora


Ing. Eduardo Interiano Salguero

Profesor lector


Dr-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, 9 de noviembre del 2009

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en 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, 9 de noviembre de 2009



Juan Esteban Marín Hernández
Céd: 2-0618-0415

Resumen

Los nemátodos son organismos microscópicos, algunos de ellos parásitos de cultivos como el plátano, las raíces y los tubérculos. El análisis de una imagen de muestra por parte de un especialista en nematología puede tardar hasta 3 horas, lo cual impide utilizar su tiempo en otras actividades. Este proyecto es parte del desarrollo de una herramienta computacional que permita disminuir la cantidad de tiempo invertido en el estudio de imágenes de nemátodos.

La ubicación de los nematodos en una imagen digital de microscopía es necesaria para continuar con el desarrollo de la herramienta computacional. La solución propuesta en este proyecto para ubicar un nemátodo consiste en construir un modelo matemático de forma, a partir de una colección representativa de imágenes, la cual es segmentada manualmente. El algoritmo que realiza la búsqueda utiliza traslaciones, rotaciones, escalado y al mismo tiempo realiza un cambio de parámetros del modelo de forma hasta encontrar una aproximación de la ubicación del nemátodo. Para evaluar el rendimiento del algoritmo se utilizan *Frentes de Pareto*, que permiten tratar con problemas multiobjetivo, en donde se trabaja con dos o más funciones de aptitud que pueden estar en conflicto, y compiten entre sí.

⁰ **Palabras Claves:** nemátodos, Active Shape Model, frente de pareto, modelo matemático de forma.

Abstract

Nematodes are microscopic organisms, and some of them parasites of crops such as bananas, roots and tubers. The analysis of an image sample by a nematology specialist may take up to three hours, which certainly prevents to use his time in other activities. This project is part of the development of a computational tool which aims to reduce the amount of time spent in the study of nematodes' images.

The location of nematodes in a digital microscopy image is required to continue with the tool's development. The proposed solution in this project to locate a nematode, is to construct a mathematical shape model from a collection of representative images, which are manually segmented.

The algorithm searches using translations, rotation and scaling, and at the same time adjusting the parameters of the shape model to find an approximation of the nematode's location.

To evaluate the performance of the algorithm the Pareto Fronts are used to deal with multiobjective problems, and work with two or more fitness functions that may conflict, and compete among themselves.

⁰ **Keywords:** *nematodes, Active Shape Model, Pareto Front, mathematical shape model.*

A mis queridos padres ...

Agradecimiento

A mi novia Mariseth Granados, que ha sido una excelente compañera de carrera y que me ayudado mucho durante la misma.

Al Dr.Pablo Alvarado Moya por darme la oportunidad de realizar este proyecto, además por todo el apoyo y la valiosa ayuda brindada durante la realización del mismo.

Índice general

Índice de figuras	iii
Índice de tablas	v
Lista de símbolos y abreviaciones	vii
1 Introducción	1
1.1 Entorno del Proyecto	1
1.2 Objetivos y Estructura de Trabajo	3
2 Marco Teórico	5
2.1 Nematodos	5
2.2 Trabajos anteriores realizados para la ubicación de nematodos	7
2.3 Análisis de Componentes Principales	9
2.4 La distancia de Mahalanobis	11
2.5 Modelos Activos de Forma	12
2.5.1 Entrenamiento del Modelo de Forma	12
2.5.2 Algoritmo de búsqueda	16
2.6 Evaluación multiobjetivo	19
3 Algoritmos Implementados	21
3.1 Algoritmos propuestos para agregar en el entrenamiento del ASM	21
3.1.1 Segmentación de la colección de imágenes	21
3.1.2 Extracción de la información del nivel de gris	22
3.2 Algoritmo de búsqueda propuesto	25
3.2.1 Cálculo de los movimientos sugeridos por medio de perfil de grises	25
4 Análisis de Resultados	29
4.1 Entrenamiento del modelo matemático	29
4.2 Evaluación de los algoritmos de búsqueda	33

4.2.1	Resultados con algoritmo de búsqueda por gradiente (ABG)	34
4.2.2	Resultados de algoritmo de búsqueda con perfil de gris (ABPG) . .	37
4.2.3	Comparación entre ABG y ABPG	40
5	Conclusiones y recomendaciones	43
5.1	Conclusiones	43
5.2	Recomendaciones	44
	Bibliografía	45
A	Cálculo para alinear un par de figuras	47
B	Reducción del cálculo de la distancia de Mahalanobis	49

Índice de figuras

1.1	Estructura del sistema automatizado de procesamiento y análisis de imágenes.	2
1.2	Diagrama de bloques del proceso de entrenamiento	3
1.3	Diagrama de bloques del proceso de búsqueda	3
2.1	Ejemplo de nematodos en imágenes de microscopía	6
2.2	Resultado final del ASM	7
2.3	Resultado de segmentación final	8
2.4	Ejemplo de aplicación de PCA en dos dimensiones.	9
2.5	Conjunto de imágenes para el modelo de forma de una mano.	12
2.6	Contorno que presenta una mano, con los posibles puntos marcados.	13
2.7	Algoritmo utilizado para alinear las figuras.	15
2.8	Ilustración del proceso de búsqueda por medio del gradiente	17
2.9	Soluciones al problema multiobjetivo.	20
3.1	Ejemplo de la segmentación manual realizada	22
3.2	Algoritmo propuesto para obtener las coordenadas de los n puntos	23
3.3	Ilustración del proceso de entrenamiento del nivel de gris	24
3.4	Algoritmo de búsqueda propuesto	26
4.1	Conjunto de imágenes para el entrenamiento del modelo matemático	30
4.2	Resultado de la alineación de la colección de imágenes.	31
4.3	Efecto obtenido al variar los primeros cuatro parámetros del modelo	32
4.4	Imagen de referencia para el frente de Pareto.	33
4.5	Frente de Pareto del algoritmo de búsqueda por gradiente	35
4.6	Resultado obtenido con ABG	36
4.7	Frente de Pareto del algoritmo de búsqueda con perfil de gris	38
4.8	Resultado obtenido con ABPG	39
4.9	Comparación de ABG y ABPG	41
4.10	Comparación de ABG y ABPG con las tres funciones de aptitud	41

Índice de tablas

4.1	Porcentaje de varianza para seis parámetros del modelo	31
4.2	Valores de los parámetros utilizados en el frente de Pareto	34
4.3	Valores óptimos de los parámetros para ABG	35
4.4	Valores óptimos de los parámetros para ABPG	38

Lista de símbolos y abreviaciones

Notación general

dX	Vector $dX = (dX_0, dY_0, \dots, dX_{n-1}, dY_{n-1})^T$
x_i	Vector $x_i = [x_{i0}, y_{i0}, x_{i1}, y_{i1}, \dots, x_{ik}, y_{ik}, \dots, x_{in-1}, y_{in-1}]^T$
X_c	Vector $X_c = (X_c, Y_c, \dots, X_c, Y_c)^T$
\overline{D}	Promedio de la distancia punto a punto entre dos vectores x_1 y x_2 : $\overline{D} = \frac{1}{n} \sum_{k=1}^n \sqrt{(x_{1k} - x_{2k})^2 + (y_{1k} - y_{2k})^2}$

Abreviaciones

ABG	Algoritmo de búsqueda por gradiente
ABPG	Algoritmo de búsqueda por perfil de gris
ASM	<i>Active Shape Model</i> .
PCA	Análisis de componentes principales (<i>Principal Component Analysis</i>)
PDM	Modelo de distribución de puntos (<i>Point Distribution Model</i>)
PESA	<i>Pareto Envelope-based Selection Algorithm</i>
RAM	Memoria de acceso aleatorio (<i>Random Access Memory</i>)

Capítulo 1

Introducción

1.1 Entorno del Proyecto

Los nematodos fitoparásitos constituyen un factor amenazante para los cultivos de raíces y tubérculos en América Latina y el Caribe. A estas plagas se les atribuye la disminución en la capacidad de función de las plantas, a las vez que permiten el ingreso para nuevos organismos patógenos [7].

Las raíces, tubérculos y el plátano representan el segundo lugar de consumo respecto a otros productos alimenticios y un alto porcentaje de la población mundial se alimenta regularmente con ellos. Por ejemplo, el plátano es uno de los cultivos más fuertes de América Latina y el Caribe, el cual se dedica al consumo local y es la fuente de empleo de pequeños y medianos productores [7].

Para el control de nematodos, se emplea una variedad de nematicidas que impactan de manera negativa sobre organismos benéficos del suelo, sobre el agua subterráneas, y además sobre los seres humanos que son los consumidores finales.

Debido a lo fundamental de los cultivos para América Latina y el Caribe, es necesaria la búsqueda de soluciones alternativas para el control de nematodos, que conlleve a una preservación del medio ambiente, y asegure la buena salud de los consumidores, así como de sus productores.

El proyecto de investigación: *“Prospección, caracterización y evaluación de las relaciones de organismos benéficos para el control de nematodos patógenos en condiciones del trópico”*, que se desarrolla en el Instituto Tecnológico de Costa Rica, busca una alternativa basada en el control biológico. Dentro del proyecto se ha propuesto la elaboración de

una herramienta computacional que permita la valoración de imágenes de microscopía.

El análisis de una imagen de muestra capturada con el microscopio puede tardar de 15 minutos hasta varias horas, dependiendo de su complejidad. Debido al elevado costo de un investigador especialista en nematología, es importante la utilización de su tiempo en otras actividades más complejas. Además, en el caso de Costa Rica al igual que en otros países, la cantidad de profesionales especializados en nematología es reducida, por lo tanto surge la necesidad de una herramienta que facilite la optimización del recurso humano.

Sin embargo, en el análisis de imágenes por sistemas computacionales, específicamente en la identificación de nematodos no existe suficiente investigación, debido a la dificultad que implica: existe variabilidad en la forma de los objetos y carencia de pistas robustas que permitan su reconocimiento, por ejemplo el color. Esto provoca que la mayoría de algoritmos existentes no sean efectivos. Las imágenes de microscopía tienen características de poco contraste lo que dificulta el reconocimiento.

La ubicación de los nematodos en una imagen digital de microscopía es necesaria para continuar con el desarrollo de la herramienta computacional que permitirá a los investigadores la valoración de imágenes en un tiempo menor.

Esta herramienta computacional está conformada por varios módulos comúnmente utilizados en sistemas de visión por computadora. El diagrama general de estos sistemas, se muestra en la figura 1.1.

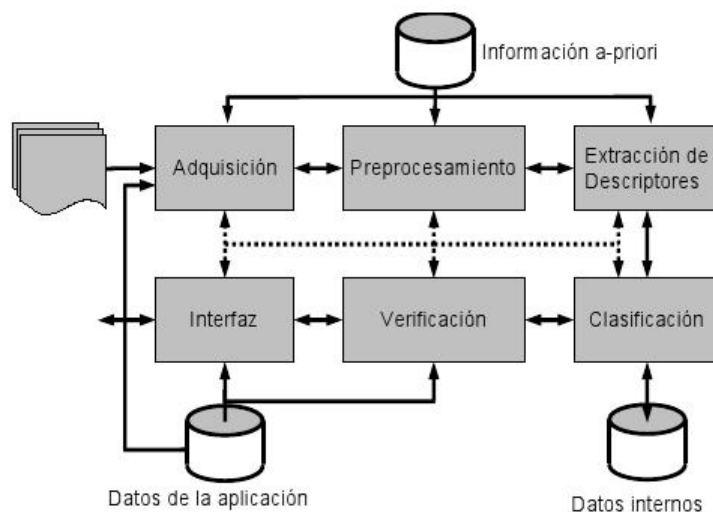


Figura 1.1: Estructura del sistema automatizado de procesamiento y análisis de imágenes. Tomado de [1]

Este proyecto implementa una estrategia de segmentación en la etapa de preprocesamiento.

1.2 Objetivos y Estructura de Trabajo

El objetivo principal de este trabajo es proponer un algoritmo que permita la ubicación de un nematodo en una imagen digital por medio de un computador. La primera etapa consiste en construir una colección de imágenes de referencia. Esta colección cuenta con figuras representativas en las cuales se pueden encontrar los nematodos. En la segunda etapa las imágenes de referencia son segmentadas manualmente, dibujando un contorno sobre el borde del nematodo. Con esta segmentación es posible extraer las coordenadas de todos los puntos correspondientes al contorno, y con esto se puede encontrar un modelo matemático de forma. El proceso se muestra en el diagrama de bloques de la figura 1.2 .

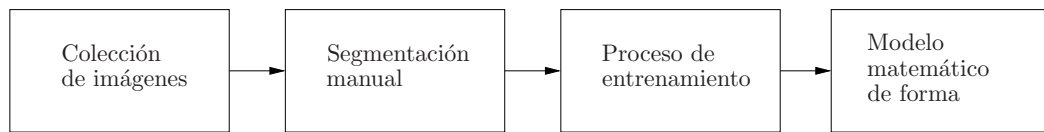


Figura 1.2: Diagrama de bloques del proceso de entrenamiento

En la tercera y última etapa se implementa un algoritmo de búsqueda que posicione y parametrize el modelo de forma para hacerlo “encajar” con un nematodo en una imagen. En la figura 1.3 se muestra el diagrama de bloques.

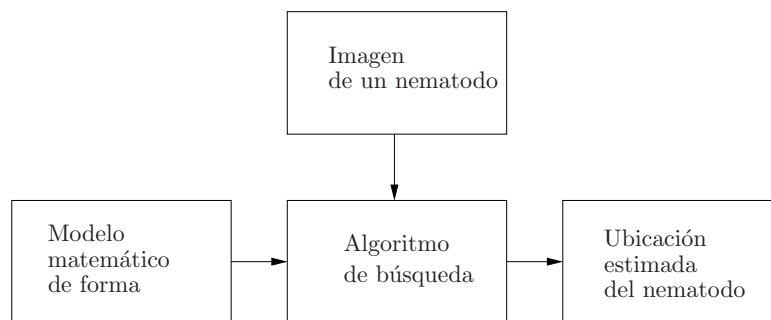


Figura 1.3: Diagrama de bloques del proceso de búsqueda

Este documento se estructura de la siguiente forma: En el capítulo 2 se presentan los conceptos necesarios para comprender la solución propuesta, además se hace referencias a

trabajos realizados con ASM. En el capítulo 3 se explica en detalle la solución implementada. En el capítulo 4 se muestra y analiza los resultados obtenidos para los algoritmos usados. Por último, en el capítulo 5 se incluyen las conclusiones obtenidas del análisis de resultados, y las recomendaciones para trabajos posteriores.

Capítulo 2

Marco Teórico

En este capítulo se presentan los conceptos que dan soporte a la solución implementada. Primero se muestra una breve reseña del concepto de nematodos, luego se incluyen referencias a trabajos anteriores realizados en algoritmos para ubicación de nematodos, se explica las herramientas matemáticas: análisis de componentes principales y la distancia de Mahalanobis. Por último se detallan los conceptos de los modelos activos de forma y el método de evaluación multiobjetivo de algoritmos llamado frente de Pareto.

2.1 Nematodos

Los nematodos son los animales multicelulares más numerosos el planeta: una muestra de tierra contiene miles [16]. Son parásitos de plantas, insectos y animales, aunque existen especies que se alimentan de bacterias, hongos u otros nematodos. Estructuralmente los nematodos son organismos sencillos: cuentan con sistema digestivo, nervioso, excretorio y reproductivo; sin embargo, no cuentan con sistema circulatorio o respiratorio. Su tamaño puede variar desde los 0.3 mm hasta 8 metros. Existen cerca de 20,000 especies clasificadas en el phylum Nemata [16], la mayoría son, biológicamente hablando, poco entendidas.

La variabilidad existente en la apariencia de los nematodos en imágenes de microscopía obliga a desarrollar algoritmos que puedan lidiar con variaciones de tamaño, con la forma debido a sus contorsiones, con la translucidez de sus cuerpos y según la edad de los nematodos. En la figura 2.1 se muestran imágenes de microscopía de nematodos con diferentes formas y tamaños.

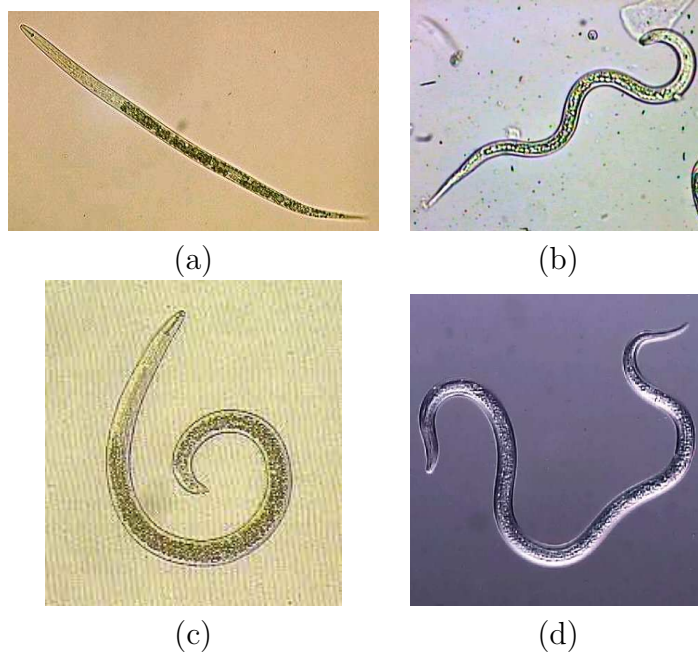


Figura 2.1: Ejemplo de nematodos en imágenes de microscopía. Figuras (a) y (b) tomado de [15], (c) tomado de [11], (d) tomado de [5].

2.2 Trabajos anteriores realizados para la ubicación de nematodos

En [2] los autores presentan una solución al problema de ubicar nematodos en imágenes digitales, basado en ASM presentado por Cootes et al. [3], sin embargo, agregan un nuevo paso en el entrenamiento, que consiste en capturar la información del nivel de gris para cada punto del modelo a través de todo el conjunto de imágenes. En el algoritmo de búsqueda se utiliza esta información para decidir los próximos movimientos del modelo. En la figura 2.2 se muestra el resultado obtenido.

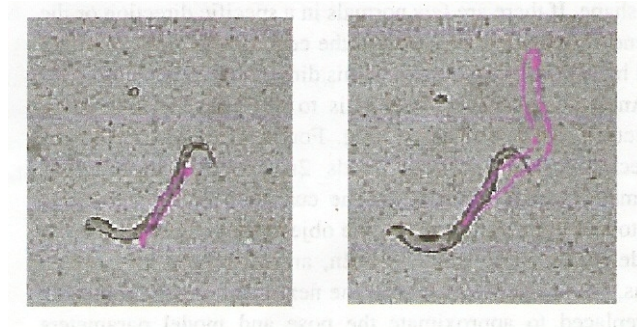


Figura 2.2: Resultado final del ASM. Tomado de [2]

Los autores atribuyen el error en la búsqueda, a una inadecuada inicialización y además a la naturaleza no lineal del movimiento del gusano, que causa que el ASM falle al intentar encontrar los bordes. En éste trabajo se extiende éste método para incorporar mayor cantidad de información en la descripción del borde (ver Capítulo 3 para más detalles).

Por otro lado, en [8] el autor propone una segmentación de dos niveles utilizando redes neuronales. En el primer nivel de segmentación las regiones encontradas no corresponden necesariamente a un objeto específico. En el segundo nivel de segmentación, se utiliza información de alto nivel para reducir el número de regiones. Se busca separar la textura de los nematodos de la textura que podría ser fondo, u otro objeto [8]. En la figura 2.3 se muestra el resultado obtenido después de realizar las dos segmentaciones.

En los resultados obtenidos se muestra que existen regiones dentro del nematodo que son clasificados como fondo, por otro lado existen objetos que son reconocidos como nematodos, cuando en realidad no lo son. Esto se debe a que la textura de los nematodos varía de un individuo a otro. La solución propuesta en este trabajo utiliza información basada en la forma de los nemátodos y no en su textura, sin embargo, el método propuesto en [8] puede ser utilizado como una segmentación previa para la solución implementada en este

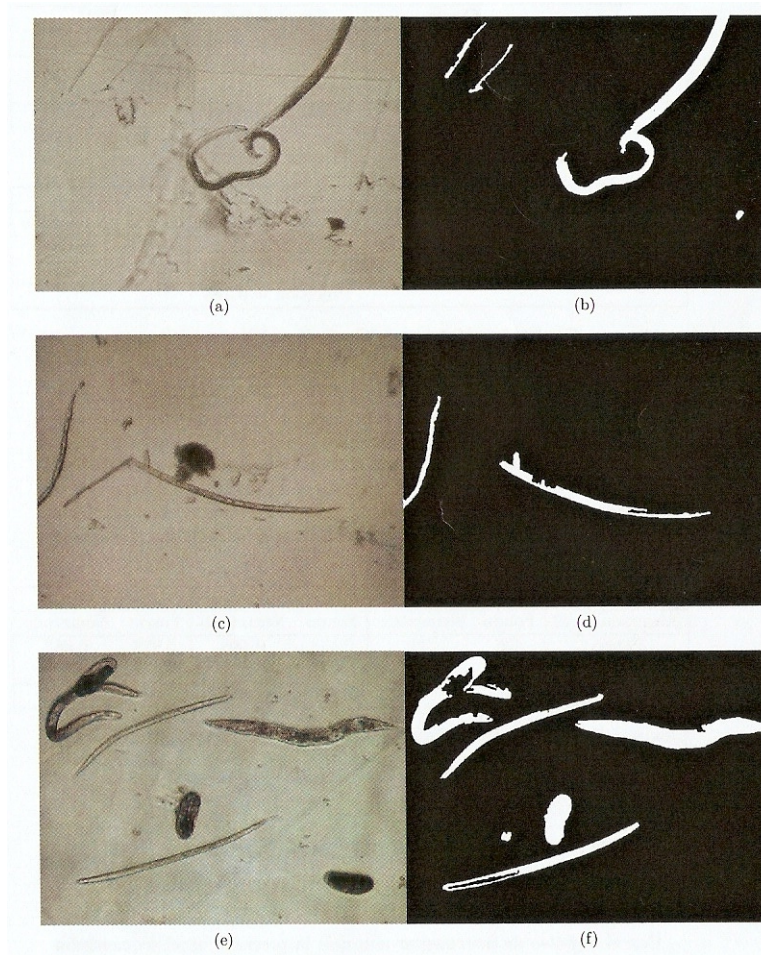


Figura 2.3: Resultado de segmentación final. Tomado de [8]

proyecto.

2.3 Análisis de Componentes Principales

El análisis de componentes principales (PCA, por sus siglas en inglés) es un método para extraer la información relevante de un conjunto de datos ruidoso con media cero. El objetivo principal es calcular una nueva base para re-expresar el conjunto de datos, filtrar el ruido, y revelar las nuevas dimensiones que representan la mayoría de los datos. En la figura 2.4 se muestra geoméricamente un ejemplo en dos dimensiones.

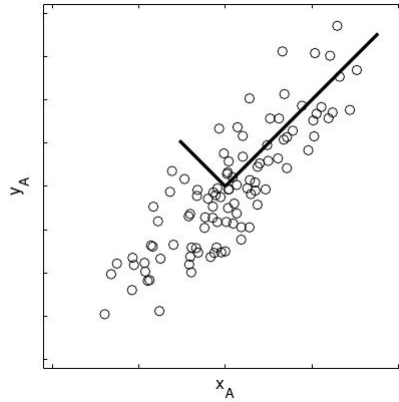


Figura 2.4: Ejemplo de aplicación de PCA en dos dimensiones. Tomado de [12]

PCA re-expresa los datos como una combinación lineal de vectores en una base. Si el conjunto de datos ruidoso con media cero es \mathbf{X} , una matriz de dimensiones $m \times n$, existe otra matriz \mathbf{Y} de $m \times n$ relacionada por una transformación lineal \mathbf{P} , en este caso \mathbf{Y} es la nueva representación de \mathbf{X} .

$$\mathbf{P}\mathbf{X} = \mathbf{Y} \quad (2.1)$$

La ecuación (2.1) representa un cambio de base y se puede interpretar de varias maneras

- \mathbf{P} es una matriz que transforma \mathbf{X} en \mathbf{Y}
- Geométricamente, los nuevos vectores de la base \mathbf{P} es una rotación y escalado que igualmente transforma \mathbf{X} en \mathbf{Y}

- \mathbf{Y} representa la proyección de \mathbf{X} en la nueva base \mathbf{P}

Ahora el problema se reduce a encontrar el cambio de base adecuado. Consiste en buscar la rotación de los vectores en la base con respecto a los “ejes” originales que maximice la relación señal a ruido de los datos, y en disminuir el factor de redundancia de los datos eliminando las dimensiones que no aportan nueva información. Si se asume que cada fila de \mathbf{X} corresponde a todas las mediciones de una variable en particular, por ejemplo distancia, humedad, entre otros, y que cada columna de \mathbf{X} corresponde a una muestra de mediciones, entonces la matriz de covarianza se puede definir como [12]

$$\mathbf{C}_\mathbf{X} \equiv \frac{1}{n-1} \mathbf{X} \mathbf{X}^T \quad (2.2)$$

Si se asume que la distribución de probabilidad de cada fila de \mathbf{X} es normalmente distribuida, la matriz de covarianza refleja el ruido y el factor de redundancia de los datos. Valores pequeños en los términos diagonales corresponden a ruido alto, y valores grandes en los términos no diagonales corresponden a alta redundancia. Por lo tanto, la matriz buscada de covarianza $\mathbf{C}_\mathbf{Y}$ debe ser una matriz diagonal.

La solución se puede encontrar utilizando álgebra lineal, basado en la descomposición de vectores propios. Primero se debe escribir $\mathbf{C}_\mathbf{Y}$ en términos de \mathbf{P}

$$\mathbf{C}_\mathbf{Y} \equiv \frac{1}{n-1} \mathbf{Y} \mathbf{Y}^T \quad (2.3a)$$

$$\equiv \frac{1}{n-1} (\mathbf{P} \mathbf{X}) (\mathbf{X}^T \mathbf{P}^T) \quad (2.3b)$$

$$\mathbf{C}_\mathbf{Y} \equiv \frac{1}{n-1} \mathbf{P} \mathbf{X} \mathbf{X}^T \mathbf{P}^T \quad (2.3c)$$

Para la matriz simétrica $\mathbf{X} \mathbf{X}^T$ se sabe que

$$\mathbf{X} \mathbf{X}^T = \mathbf{E} \mathbf{D} \mathbf{E}^T \quad (2.4)$$

En donde \mathbf{D} es una matriz diagonal conformada por los valores propios de $\mathbf{X} \mathbf{X}^T$ y \mathbf{E} es una matriz de vectores propios de $\mathbf{X} \mathbf{X}^T$, organizados en columnas. Ahora se asume que \mathbf{P} es una matriz ortogonal donde cada fila es un vector propio de $\mathbf{X} \mathbf{X}^T$, de esta manera $\mathbf{P} \equiv \mathbf{E}^T$ y sustituyendo en (2.4) se puede continuar con (2.3c).

$$\mathbf{C}_\mathbf{Y} \equiv \frac{1}{n-1} \mathbf{P} (\mathbf{P}^T \mathbf{D} \mathbf{P}) \mathbf{P}^T \quad (2.5a)$$

$$\mathbf{C}_Y \equiv \frac{1}{n-1} (\mathbf{P}\mathbf{P}^T) \mathbf{D} (\mathbf{P}\mathbf{P}^T) \quad (2.5b)$$

$$\mathbf{C}_Y \equiv \frac{1}{n-1} (\mathbf{P}\mathbf{P}^{-1}) \mathbf{D} (\mathbf{P}\mathbf{P}^{-1}) \quad (2.5c)$$

$$\mathbf{C}_Y \equiv \frac{1}{n-1} \mathbf{D} \quad (2.5d)$$

Los componentes principales de \mathbf{X} son los vectores propios de $\mathbf{X}\mathbf{X}^T$, o las filas de \mathbf{P} . Por lo tanto para realizar el análisis de componentes principales de un conjunto de datos con media cero \mathbf{X} se debe calcular los vectores propios de $\mathbf{X}\mathbf{X}^T$.

2.4 La distancia de Mahalanobis

La distancia de Mahalanobis es una herramienta de la estadística cuyo propósito consiste en determinar la similitud existente entre dos variables aleatorias multidimensionales, $\underline{\mathbf{m}}_0$ y $\underline{\mathbf{m}}_1$. Este concepto se representa mediante la siguiente expresión, según [4]

$$\Delta = \sqrt{(\underline{\mathbf{m}}_1 - \underline{\mathbf{m}}_0)^T \Sigma^{-1} (\underline{\mathbf{m}}_1 - \underline{\mathbf{m}}_0)} \quad (2.6)$$

Esta expresión contempla la correlación entre dichas variables, y para poder ser aplicada se requiere que ambas variables aleatorias mantenga una misma distribución de probabilidad.

Adicionalmente, es necesario tomar en consideración las tres propiedades fundamentales que deben ser satisfechas para garantizar que la expresión corresponda a una distancia, y las cuales se mencionan a continuación:

- **Semipositividad:** La distancia siempre es positiva cuando se trata de dos puntos con distintas coordenadas, o cero si poseen las mismas, es decir: $d(a, b) \geq 0 \forall a, b \in X$ y además $d(a, b) = 0$ si y solo si $a = b$
- **Simetría:** La distancia entre los puntos a y b , es la misma que entre b y a : $d(a, b) = d(b, a) \forall a, b \in X$
- **Desigualdad triangular:** $d(a, b) \leq d(a, c) + d(c, b) \forall a, b, c \in X$

Al utilizar la distancia de Mahalanobis se pretende igualar la importancia de las variables en cuestión, ya que considera la estadística de los datos, ponderando su varianza, de manera que las variables que posean una menor varianza sean las más representativas.

2.5 Modelos Activos de Forma

La implementación de los modelos activos de forma se divide en dos etapas: entrenamiento del modelo de forma, que sólo se realiza una vez, y el algoritmo de búsqueda, que utiliza el modelo previamente entrenado. Las dos etapas se explican a continuación.

2.5.1 Entrenamiento del Modelo de Forma

Colección de Imágenes

Para construir un modelo que sea lo suficientemente flexible para cubrir las más típicas variaciones de un objeto, se debe emplear una colección de N imágenes suficientemente grande [9]. En la figura 2.5 se muestra un ejemplo de una colección de imágenes para el modelo de una mano.

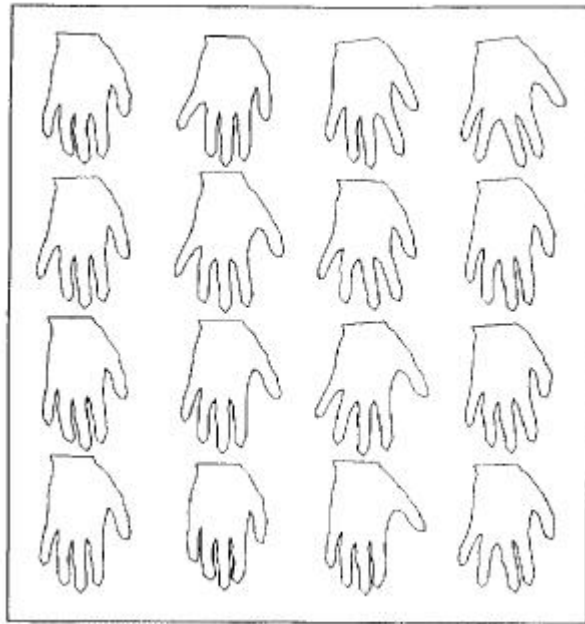


Figura 2.5: Conjunto de imágenes para el modelo de forma de una mano. Tomado de [3]

Para cada una de las imágenes se colocan manualmente n puntos distribuidos a lo largo del borde, la cantidad debe ser la necesaria para representar adecuadamente la forma y estar ubicados en lugares representativos del objeto. En la figura 2.6 se muestra un ejemplo.

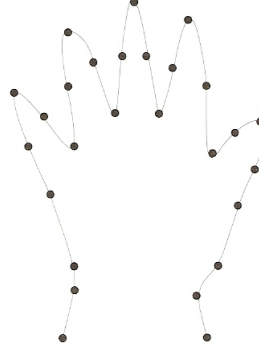


Figura 2.6: Contorno que presenta una mano, con los posibles puntos marcados. Tomado de [13]

De esta manera se puede obtener un vector $\underline{\mathbf{x}}$ para cada imagen que contiene las coordenadas en el plano xy para los n puntos.

Alineamiento de las figuras

El método de modelado se basa en la información estadística aportada por los puntos en la colección de imágenes. Sin embargo, para poder comparar puntos de diferentes figuras, deben estar alineados con respecto a los mismos ejes. Esto se logra modificando su escala, rotación y traslación de manera que correspondan lo más cercano posible.

Para alinear un par de figuras $\underline{\mathbf{x}}_i$ y $\underline{\mathbf{x}}_j$, se debe minimizar la distancia entre los puntos equivalentes de cada una. Esto se logra con una rotación θ_j , un factor de escalado s_j , y una traslación (t_{xj}, t_{yj}) que mapea $\underline{\mathbf{x}}_j$ en $\mathbf{M}(s_j, \theta_j) [\underline{\mathbf{x}}_j] + t_j$ tal que minimiza (2.7). [3].

$$E_j = \left(\underline{\mathbf{x}}_i - \mathbf{M}(s_j, \theta_j) [\underline{\mathbf{x}}_j] - t_j \right)^T \mathbf{W} \left(\underline{\mathbf{x}}_i - \mathbf{M}(s_j, \theta_j) [\underline{\mathbf{x}}_j] - t_j \right) \quad (2.7)$$

en donde

$$\mathbf{M}(s, \theta) = \begin{pmatrix} (s \cos \theta) x_{jk} - (s \sin \theta) y_{jk} \\ (s \sin \theta) x_{jk} + (s \cos \theta) y_{jk} \end{pmatrix} \quad (2.8)$$

$$\mathbf{t}_j = (t_{xj}, t_{yj}, \dots, t_{xj}, t_{yj})^T \quad (2.9)$$

\mathbf{W} es una matriz diagonal de pesos para cada punto. Los pesos se escogen de manera que

den más importancia a aquellos puntos que tienden a permanecer más estables a través de todas las figuras de la colección. Si R_M es la distancia entre los puntos k y l en una figura, y V_{RM} es la varianza de esta distancia en todas las N figuras, el valor del peso w_k para el punto k es

$$w_k = \left(\sum_{l=0}^{n-1} V_{RM} \right)^{-1} \quad (2.10)$$

La ecuación para encontrar s_j , θ_j y \mathbf{t}_j se encuentra en el Apéndice A.

En la figura 2.7 se muestra el algoritmo que se utiliza para alinear las N figuras. En este caso, la etapa de normalización se refiere a escalar, rotar y trasladar la figura promedio hasta alinearla con la primera figura en la colección, o se puede utilizar una postura por defecto. La normalización de la figura promedio asegura que el algoritmo converga.

Obtención del Modelo

Cada figura en la colección de entrenamiento, ahora alineada, puede ser considerada como un punto en un espacio de $2n$ dimensiones, y el conjunto de todas las N figuras, forma una nube de N puntos en este espacio. Si se asume que todos los N puntos están dentro de una región \mathbb{R} , en donde cada uno de los puntos dentro de este dominio representa una figura que es similar a las utilizadas en la colección de entrenamiento, entonces al moverse a través de la región \mathbb{R} se pueden generar nuevas figuras en una manera sistemática.

Los ejes principales de la región \mathbb{R} de $2n$ dimensiones pueden ser calculados aplicando PCA, sin embargo, el conjunto no tiene media cero, para ello se aplica (2.11) a cada figura.

$$d\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}} \quad (2.11)$$

en donde

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i \quad (2.12)$$

La mayoría de la variación en el modelo puede obtenerse por medio de un menor número de t modos. Esto significa que la región \mathbb{R} de $2n$ dimensiones puede ser aproximada por una región de t dimensiones. Un método para encontrar t es tomar el menor número de

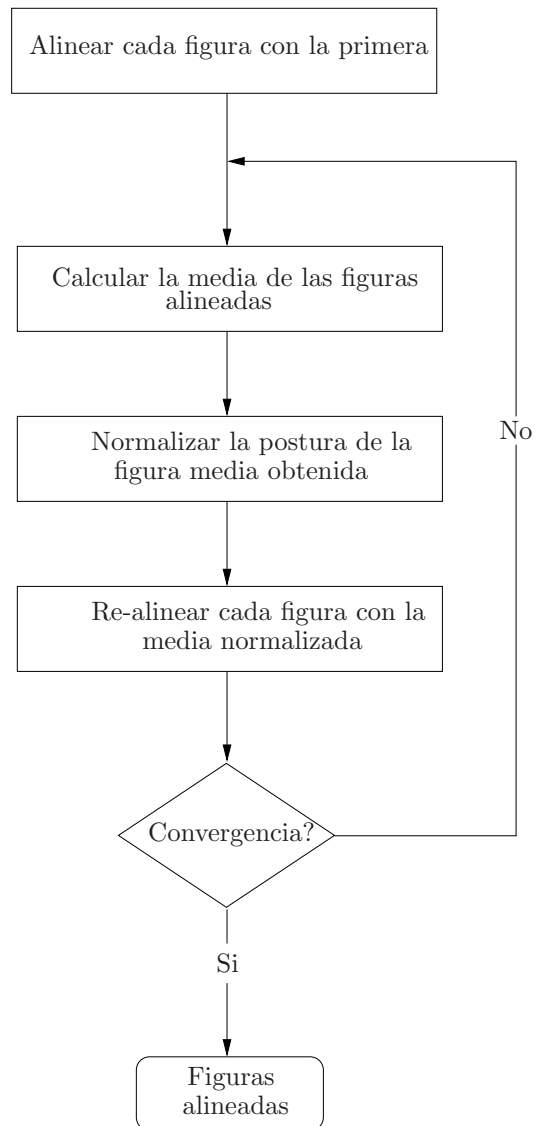


Figura 2.7: Algoritmo utilizado para alinear las figuras. Tomado de [9]

modos, tal que la suma de sus varianzas o valores propios, sea una porción dada de λ_T , la varianza total, donde

$$\lambda_T = \sum_{k=1}^{2n} \lambda_k \quad (2.13)$$

Para reconstruir las figuras de la colección a partir de los primeros t modos se utiliza (2.14).

$$\underline{\mathbf{x}} = \underline{\bar{\mathbf{x}}} + \mathbf{P}\underline{\mathbf{b}} \quad (2.14)$$

en donde \mathbf{P} es la matriz con los primeros t vectores propios y $\underline{\mathbf{b}}$ es un vector de pesos. La ecuación (2.14) permite generar nuevas figuras, variando los parámetros b_k dentro de límites adecuados, para que las nuevas figuras sean parecidas a la colección de entrenamiento. Los límites se encuentran típicamente en el orden de tres veces la desviación estándar de la media [3].

$$-3\sqrt{\lambda_k} \leq b_k \leq 3\sqrt{\lambda_k} \quad (2.15)$$

2.5.2 Algoritmo de búsqueda

El algoritmo de búsqueda tiene como entradas la imagen del nematodo a buscar, y el modelo matemático entrenado. Su salida es un polígono dibujado sobre la imagen de entrada, formado por todos los puntos del modelo en su ubicación final. Una instancia del modelo esta dada por

$$\underline{\mathbf{X}} = \mathbf{M}(s, \theta) [\underline{\mathbf{x}}] + \underline{\mathbf{X}}_{\mathbf{c}} \quad (2.16)$$

La posición $\underline{\mathbf{X}}_{\mathbf{c}}$, el factor de escala s y la orientación θ inicial del modelo. No es necesario que el valor inicial de $\underline{\mathbf{X}}$ este cerca de la solución final.

Cálculo de los movimientos sugeridos para cada punto

El vector gradiente apunta en la dirección en que la tasa de cambio es más grande, y su magnitud es el valor de la tasa de cambio en esa dirección [6]. La función magnitud del gradiente es utilizada para la detección de bordes. Si se aplica ésta función a una imagen se obtendrá otra con las mismas dimensiones, conocida como imagen gradiente.

Para calcular el desplazamiento del punto i del modelo, se recorre una línea de largo L normal al vector que va desde el punto $(i - 1)$ hasta el punto $(i + 1)$ (figura 2.8), y se elige el pixel que presente la mayor magnitud del gradiente. En el caso de la figura de ejemplo, el borde, representado con una línea roja, está conformado por los puntos que tienen el valor mayor del gradiente. Si esta magnitud es menor a Th_g el desplazamiento para el punto i es cero.

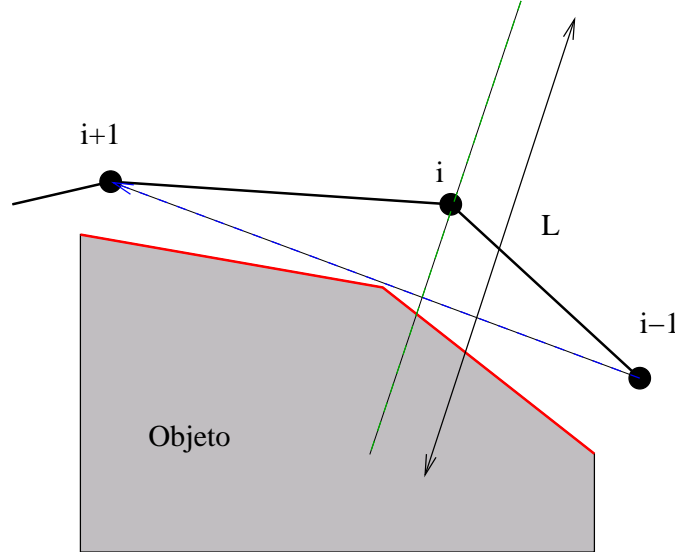


Figura 2.8: Ilustración del proceso de búsqueda por medio del gradiente

Cálculo de cambios en la postura y en los parámetros del modelo

Ahora el objetivo es ajustar la postura y los parámetros del modelo para mover los puntos de su localización actual, $\underline{\mathbf{X}}$, hasta la nueva posición encontrada $(\underline{\mathbf{X}} + d\underline{\mathbf{X}})$ mientras se sigan satisfaciendo las limitaciones del modelo. El modelo estimado actual está centrado en (X_c, Y_c) con una orientación θ y escala s , primero se deben actualizar estos parámetros, esto se logra encontrando la traslación (dX_c, dY_c) , rotación $d\theta$ y el factor de escala $(1 + ds)$ que realiza el mapeo de $\underline{\mathbf{X}}$ en $(\underline{\mathbf{X}} + d\underline{\mathbf{X}})$, utilizando el método dado en el Apéndice A.

Después de ajustar la postura, se procede a realizar cambios en el modelo de forma. Es necesario calcular los ajustes, $d\underline{\mathbf{x}}$, en el modelo, requeridos para que los puntos de $\underline{\mathbf{X}}$ se muevan en la dirección $d\underline{\mathbf{X}}$ cuando se combinan con el efecto de la nueva escala, rotación y traslación [3].

La posición inicial de los puntos está dada por (2.16). Se desea encontrar $d\underline{\mathbf{x}}$ tal que:

$$\mathbf{M}(s(1+ds), (\theta+d\theta)) [\underline{\mathbf{x}} + d\underline{\mathbf{x}}] + (\underline{\mathbf{X}}_{\mathbf{c}} + d\underline{\mathbf{X}}_{\mathbf{c}}) = (\underline{\mathbf{X}} + d\underline{\mathbf{X}}) \quad (2.17)$$

Reacomodando términos

$$\mathbf{M}(s(1+ds), (\theta+d\theta)) [\underline{\mathbf{x}} + d\underline{\mathbf{x}}] = (\mathbf{M}(s, \theta) [\underline{\mathbf{x}}] + \underline{\mathbf{X}}_{\mathbf{c}} + d\underline{\mathbf{X}} - (\underline{\mathbf{X}}_{\mathbf{c}} + d\underline{\mathbf{X}}_{\mathbf{c}})) \quad (2.18)$$

y ya que

$$\mathbf{M}^{-1}(s, \theta) [\] = \mathbf{M}(s^{-1}, -\theta) [\] \quad (2.19)$$

se obtiene

$$d\underline{\mathbf{x}} = \mathbf{M}((s(1+ds))^{-1}, -(\theta+d\theta)) [y] - \underline{\mathbf{x}} \quad (2.20)$$

en donde

$$y = \mathbf{M}(s, \theta) [\underline{\mathbf{x}}] + d\underline{\mathbf{X}} - d\underline{\mathbf{X}}_{\mathbf{c}} \quad (2.21)$$

Con (2.20) se pueden encontrar los movimientos necesarios para los puntos del modelo $\underline{\mathbf{x}}$, sin embargo, estos movimientos no son en general consistentes con los límites del modelo. Para esto se utiliza (2.14). Se necesita encontrar $d\underline{\mathbf{b}}$ tal que

$$\underline{\mathbf{x}} + d\underline{\mathbf{x}} \approx \underline{\bar{\mathbf{x}}} + \mathbf{P}(\underline{\mathbf{b}} + d\underline{\mathbf{b}}) \quad (2.22)$$

Ya que sólo se están utilizando t modos de variación, y el vector $d\underline{\mathbf{x}}$ puede mover sus puntos en $2n$ grados de libertad, sólo se obtendrá una aproximación de la deformación requerida. Despejando la ecuación para $d\underline{\mathbf{b}}$ se obtiene

$$d\underline{\mathbf{b}} = \underline{\mathbf{P}}^T d\underline{\mathbf{x}} \quad (2.23)$$

Actualización de los parámetros del modelo y de la postura

En la última etapa del algoritmo de búsqueda se actualizan los parámetros de la siguiente manera

$$X_c + dX_c \rightarrow X_c \quad (2.24)$$

$$Y_c + dY_c \rightarrow Y_c \quad (2.25)$$

$$\theta + d\theta \rightarrow \theta \quad (2.26)$$

$$s(1 + ds) \rightarrow s \quad (2.27)$$

$$\underline{\mathbf{b}} + d\underline{\mathbf{b}} \rightarrow \underline{\mathbf{b}} \quad (2.28)$$

Para asegurar que el modelo sólo se deforma en figuras consistentes con la colección de entrenamiento se deben aplicar los límites a cada uno de los valores b_k dados por (2.15). Una vez que los parámetros han sido actualizados, y se han aplicado los límites, la nueva posición de $\underline{\mathbf{X}}$ puede ser calculada, y de nuevo se pueden calcular movimientos sugeridos para cada punto. El proceso se repite una r cantidad de veces o hasta que no hayan cambios significativos.

2.6 Evaluación multiobjetivo

Los problemas multiobjetivo trabajan con dos o más funciones de aptitud, las cuales posiblemente pueden estar en conflicto entre sí. Ante esta situación, los algoritmos genéticos surgen como una opción para solucionar este tipo de problemas, optimizando simultáneamente las funciones involucradas.

Si suponemos a manera de ilustración, dos funciones f_1 y f_2 como medidas de aptitud para un algoritmo parametrizable, resulta particularmente útil obtener un gráfico de f_1 contra f_2 (ver figura 2.9) en donde cada punto representa el rendimiento del algoritmo con un valor específico de sus parámetros.

En el caso de la figura 2.9 se puede observar que el punto q_1 domina sobre todos los puntos que se encuentran en la región gris, las líneas punteadas indican las regiones de dominancia para los demás puntos, y la línea sólida indica el frente de Pareto para los cuatro puntos en cuestión [1]. El frente de Pareto es el conjunto de puntos que representa las parametrizaciones que producen el mayor rendimiento del algoritmo. En una optimización multi-objetivo no existe concretamente una única solución óptima con respecto al resto de objetivos sino que cuenta con múltiples soluciones, donde cada una puede ser igualmente satisfactoria dependiendo de la aplicación.

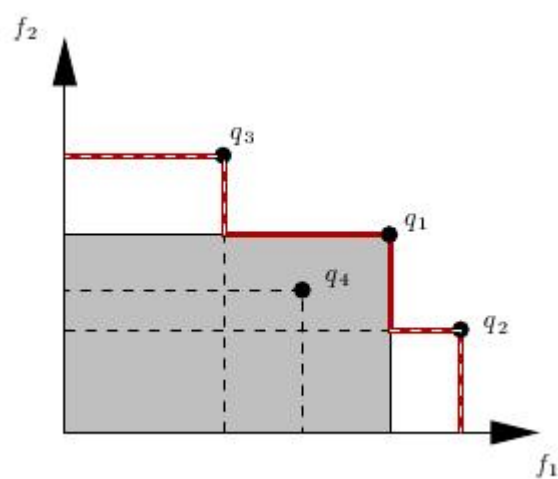


Figura 2.9: Soluciones al problema multiobjetivo. Tomado de [1]

Capítulo 3

Algoritmos Implementados

En este capítulo se presentan los algoritmos propuestos para utilizarse con el algoritmo clásico del ASM. En el caso de la segmentación manual de imágenes, se desarrolló un algoritmo para extraer los puntos representativos del contorno del nematodo y para la detección de bordes en las imágenes, se utiliza una modificación del método usado en [2] de perfil de grises.

Todos los algoritmos se desarrollaron en C++, como parte de la biblioteca de algoritmos y estructuras de datos usados en procesamiento de imágenes y visión por computadora LTI-Lib2 [14].

3.1 Algoritmos propuestos para agregar en el entrenamiento del ASM

3.1.1 Segmentación de la colección de imágenes

Primero se disminuye en un 10% el nivel de brillo de la imagen, y se realiza una segmentación manual, dibujando un contorno de color blanco sobre todo el borde del nemátodo, luego se indican los dos extremos del animal con un pixel de color rojo en cada uno (ver figura 3.1), se realiza la disminución del brillo para asegurar que ningún pixel sea de color totalmente blanco y pueda ser erróneamente reconocido como borde. Por último cada imagen es procesada por un algoritmo que asigna los n puntos distribuidos equitativamente en todo el contorno (ver figura 3.2). Este algoritmo utiliza los puntos rojos marcados previamente como referencia, luego realiza un conteo de todos los pixels que forman parte

del borde(j), calcula la distancia entre cada punto como $\frac{n}{j}$, los asigna en todo el borde del nematodo, almacena las coordenadas xy de cada punto en un vector, y el proceso se repite para todas las imágenes de la colección.



Figura 3.1: Ejemplo de la segmentación manual realizada

3.1.2 Extracción de la información del nivel de gris

En esta sección se explica cómo encontrar la información del nivel de gris para cada punto de las figuras de entrenamiento. El método descrito en [2] para almacenar la información del nivel de gris recorre una línea de largo w perpendicular al vector que va desde el punto $(i - 1)$ hasta el punto $(i + 1)$, y almacena la información en un vector. Sin embargo, el método utilizado en [2] disminuye su efectividad en imágenes que contengan nematodos y otros objetos, ya que el algoritmo de búsqueda tiende a trasladar el modelo de forma hacia objetos que no son nematodos.

La solución propuesta en este trabajo toma en cuenta la información de nivel de gris de los pixels cercanos al punto i , de esta manera almacena la apariencia del borde del nematodo alrededor de cada punto del modelo de forma. Si se almacena la información de nivel de gris de cada pixel en un rectángulo de dimensiones $m \times w$ centrado en el punto i , entonces la matriz de covarianza obtenida para el i -ésimo punto posee dimensiones $mw \times mw$, que a su vez incrementa la cantidad de memoria requerida para el programa y el tiempo necesario para el cálculo de las operaciones. Para minimizar el tiempo de ejecución y la memoria utilizada por el programa, la solución propuesta en este proyecto consiste en utilizar tres líneas de largo w paralelas centradas en el punto i y perpendiculares al vector que va desde el punto $(i - 1)$ hasta el punto $(i + 1)$ y con distancia de d pixels entre cada

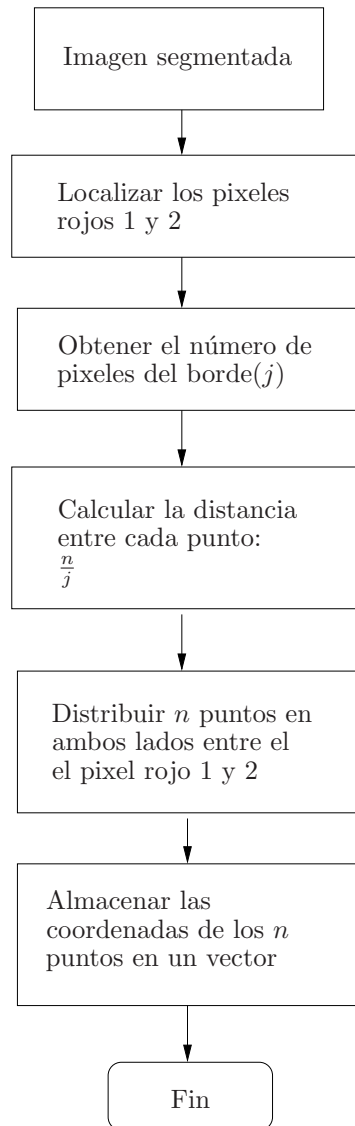


Figura 3.2: Algoritmo propuesto para obtener las coordenadas de los n puntos

una (figura 3.3).

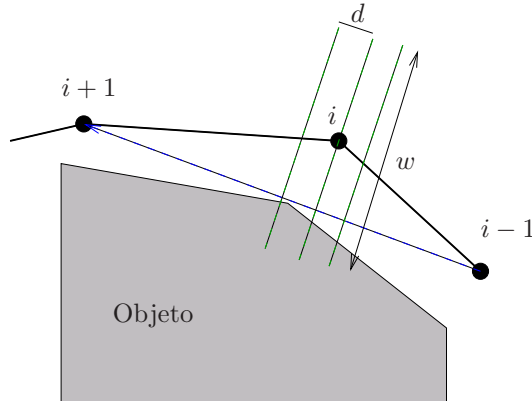


Figura 3.3: Ilustración del proceso de entrenamiento del nivel de gris

El algoritmo para el punto i es

1. Para la imagen j en la colección.
 - 1.1. Se deben recorrer las tres líneas y almacenar el perfil de grises de todos los pixels que atraviesan en un único vector:

$$\underline{\mathbf{G}}_{\mathbf{j}\mathbf{i}} = [g_{ji,0}, g_{ji,1}, \dots, g_{ji,3w-1}]$$

- 1.2. Calcular la derivada del perfil de gris:

$$d\mathbf{G}_{\mathbf{j}\mathbf{i}} = [g_{ji,1} - g_{ji,0}, g_{ji,2} - g_{ji,1}, \dots, g_{ji,3w-1} - g_{ji,3w-2}]$$

- ### 1.3. Normalizar la derivada:

$$\underline{Y_{\mathbf{j}i}} = \frac{dG_{ji}}{\sum_{k=0}^{3w-2} |d\mathbf{G}_{\mathbf{j}i,k}|}$$

2. Calcular la media de la derivada normalizada para todas las imágenes del conjunto de entrenamiento:

$$\overline{\mathbf{Y}_i} = \frac{1}{N} \sum_{j=1}^N d\mathbf{G}_{ji}$$

3. Calcular la matriz de covarianza de la derivada normalizada:

$$\mathbf{C}_i = \frac{1}{N} \sum_{j=1}^N \left(\underline{\mathbf{Y}}_{\mathbf{j}i} - \overline{\mathbf{Y}}_i \right) \left(\underline{\mathbf{Y}}_{\mathbf{j}i} - \overline{\mathbf{Y}}_i \right)^T$$

La matriz \mathbf{C}_i describe la apariencia del borde del nematodo exactamente en el punto i del modelo de forma.

3.2 Algoritmo de búsqueda propuesto

La posición del nemátodo es totalmente desconocida, y el ASM necesita una inicialización aproximada a la solución final. En el presente proyecto se implementó el algoritmo de búsqueda que se muestra en la figura 3.4

Se realizan m iteraciones del algoritmo ASM, en cada iteración, las posiciones X_c , Y_c , el factor de escala s y la orientación θ son seleccionados de manera aleatoria, y el valor total de la suma de las distancias de Mahalanobis de todos los puntos a los cuales el algoritmo ASM converge es almacenado. Cuando se han alcanzado las m iteraciones, se escoge el vector $\underline{\mathbf{X}}$ que presente la menor distancia de Mahalanobis como el resultado de la búsqueda.

3.2.1 Cálculo de los movimientos sugeridos por medio de perfil de grises

Este algoritmo se propone como reemplazo del método clásico para la búsqueda de bordes usado en ASM, que se basa en la magnitud del gradiente. Partiendo de una posición inicial dada para los puntos del modelo, se realiza una búsqueda para encontrar un conjunto de ajustes que mueva cada punto a una mejor posición, conocido como el vector $d\underline{\mathbf{X}}$. En este caso ASM utiliza el perfil de grises y la matriz de covarianza de la derivada normalizada promedio, generada en la etapa de entrenamiento para encontrar el vector $d\underline{\mathbf{X}}$.

Para el punto i en $\underline{\mathbf{x}}_j$, se recorren tres líneas de largo L ($L \gg w$) perpendiculares al vector que va desde el punto $(i - 1)$ hasta el punto $(i + 1)$, con distancia de d pixels entre cada una y se realiza el siguiente algoritmo para encontrar la distancia de Mahalanobis [2]:

1. Se deben recorrer las tres líneas y almacenar el perfil de grises de todos los pixels que atraviesan en un único vector:

$$\underline{\mathbf{S}}_i = [g_{i,0}, g_{i,1}, \dots, g_{i,3L-1}]$$

2. Calcular la derivada del perfil:

$$d\underline{\mathbf{S}}_i = [g_{i,1} - g_{i,0}, g_{i,2} - g_{i,1}, \dots, g_{i,3L-2} - g_{i,3L-1}]$$

3. Normalizar la derivada:

$$\underline{\mathbf{Y}}_{\mathbf{S}i} = \frac{dS_i}{\sum_{k=0}^{3L-2} |d\underline{\mathbf{S}}_{i,k}|}$$

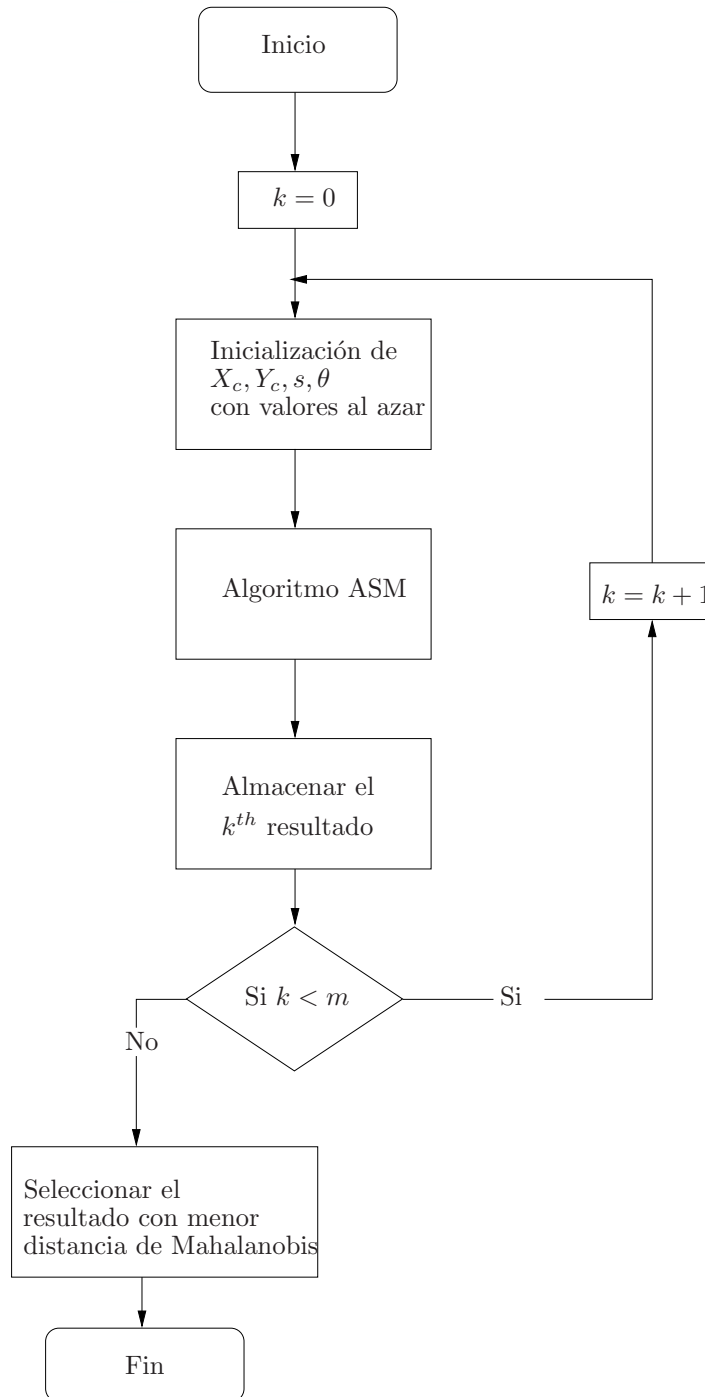


Figura 3.4: Algoritmo de búsqueda propuesto

4. $\underline{\mathbf{h}}(d)$ es un vector de tamaño $3w$ sub-interno de $\underline{\mathbf{Y}}_{\mathbf{Si}}$ y está centrado en el pixel d de $\underline{\mathbf{Y}}_{\mathbf{Si}}$
5. Encontrar d tal que la distancia de Mahalanobis sea mínima:

$$f(d) = \sqrt{(\underline{\mathbf{h}}(d) - \overline{\mathbf{Y}}_{\mathbf{i}})^T \mathbf{C}_{\mathbf{Yi}}^{-1} (\underline{\mathbf{h}}(d) - \overline{\mathbf{Y}}_{\mathbf{i}})}$$

6. Calcular $d\mathbf{X}_{\mathbf{i}}$ utilizando las coordenadas del pixel d y del punto i .

El cálculo de la distancia de Mahalanobis se ejecuta L veces para los n puntos, lo cual incrementa en gran parte el tiempo de ejecución del programa. Para minimizar ese tiempo la matriz de covarianza se invierte antes de almacenarla en el proceso de entrenamiento. Sin embargo, el cálculo aún se puede optimizar, reduciendo el número de operaciones realizadas ya que la matriz de covarianza invertida continúa siendo simétrica, el desarrollo completo se muestra en el Apéndice B.

Capítulo 4

Análisis de Resultados

En este capítulo se presentan los resultados obtenidos a partir del entrenamiento del modelo de forma y de la aplicación del algoritmo de búsqueda. En el caso del algoritmo de búsqueda, se realizó una comparación entre los dos métodos descritos en el capítulo 3 para la detección de bordes y elección de los movimientos sugeridos para cada punto. Los dos algoritmos se evaluaron mediante el frente de Pareto. Para la evaluación se empleó una computadora con un procesador Intel Core 2 Duo 6400 de 2.13 GHz y con 3 GB de RAM.

4.1 Entrenamiento del modelo matemático

El primer paso para realizar el entrenamiento del modelo, es recolectar un conjunto de imágenes y realizar una segmentación manual. En este proyecto se utilizó el conjunto de cinco imágenes segmentadas manualmente que se muestra en la figura 4.1. Cada figura está conformada por cincuenta puntos, dos en los extremos y el resto distribuidos a distancias equivalentes en los costados.

Los cinco nematodos tienen forma, tamaño y orientación diferentes, por eso se deben alinear antes de poder aplicar el PCA. En la figura 4.2 se muestra los puntos correspondientes a las cinco figuras alineadas, y además la figura media.

Como se aprecia en la figura, la distribución de los puntos cerca del origen es similar para las cinco figuras, sin embargo en los extremos de la figura media, los puntos se encuentran distanciados. Esto indica que los parámetros más significativos del modelo cambiarán la forma del nematodo mayormente en los extremos que en el medio.

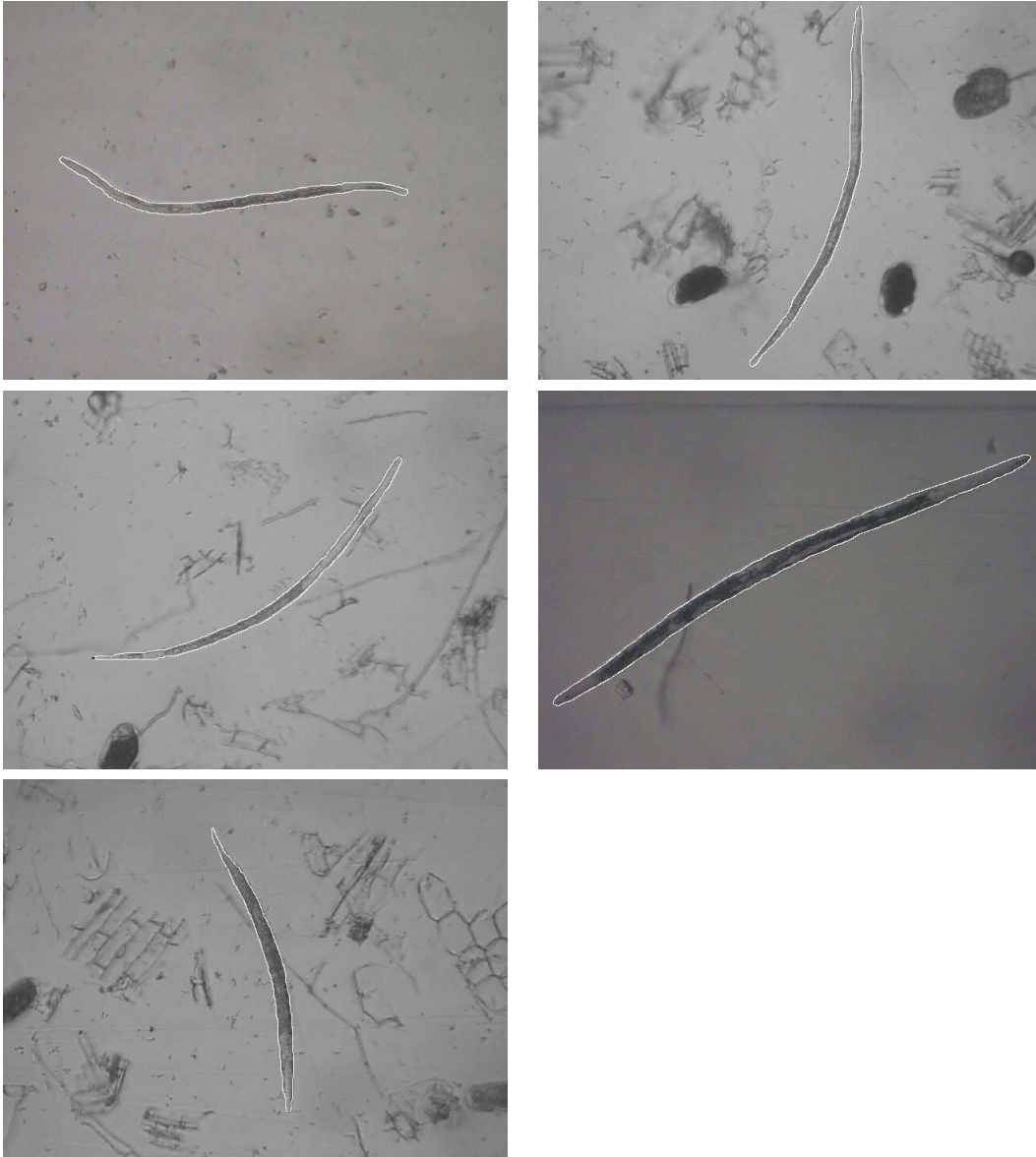


Figura 4.1: Conjunto de imágenes para el entrenamiento del modelo matemático

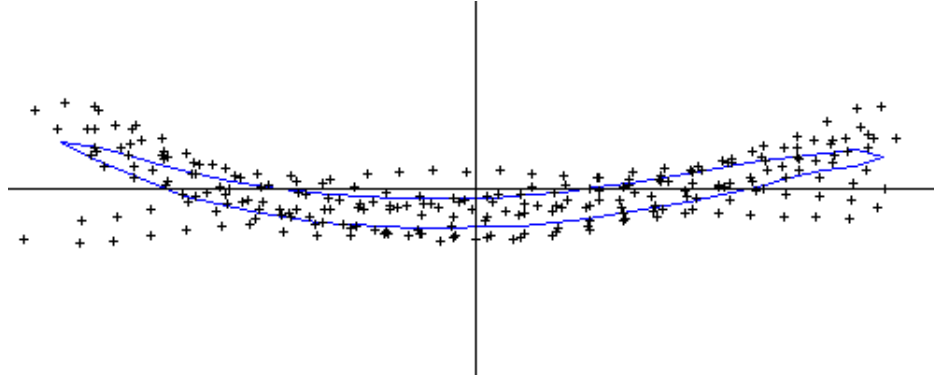


Figura 4.2: Resultado de la alineación de la colección de imágenes.

El siguiente paso para obtener el modelo es aplicar PCA a las figuras alineadas del conjunto de entrenamiento. En la tabla 4.1 se muestra el porcentaje de varianza que corresponde a los primeros seis parámetros del modelo.

Tabla 4.1: Porcentaje de varianza para seis parámetros del modelo

Valor propio	$\frac{\lambda_i}{\lambda_T} \times 100\%$
λ_1	75,8239%
λ_2	14,6297%
λ_3	6,46252%
λ_4	3,08383%
λ_5	0,00051%
λ_6	0,00013%

Con los cuatro primeros parámetros, tal y como puede observarse en la tabla 4.1, es posible representar el 99,99% de las formas permitidas para el modelo. Es decir, la región R , que en este caso es de cien dimensiones (ver sección 2.5.1), puede ser aproximada en un 99,99% por un subespacio de cuatro dimensiones. Por lo tanto, se trabajará solo con los primeros cuatro parámetros en el algoritmo de búsqueda.

Para visualizar el efecto de los primeros cuatro parámetros, se generaron figuras en donde se varía sólo un parámetro a la vez, desde $-2\sqrt{\lambda_k}$ hasta $2\sqrt{\lambda_k}$. El resultado se muestra en la figura 4.3.

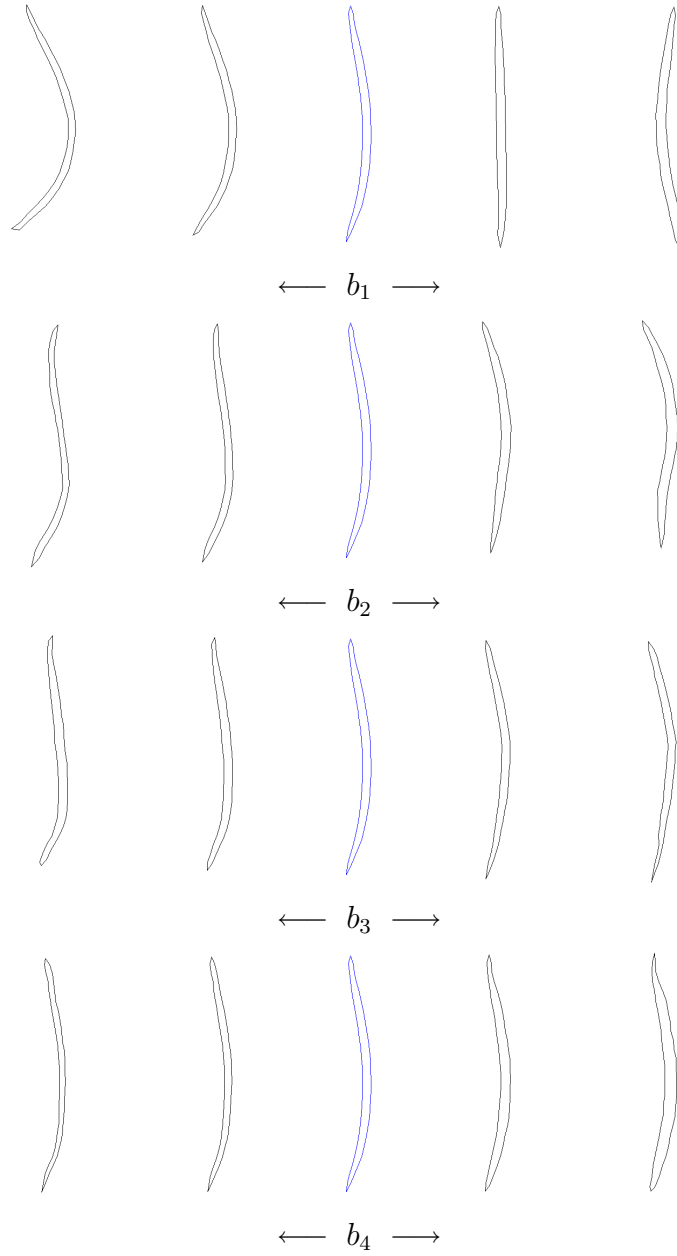


Figura 4.3: Efecto obtenido al variar los primeros cuatro parámetros del modelo. La figura promedio se muestra en color azul

4.2 Evaluación de los algoritmos de búsqueda

Para la evaluación de los algoritmos de búsqueda se utiliza un frente de Pareto. Éste necesita de una referencia o conjunto de referencias para poder evaluar los algoritmos. En este caso se utilizó la imagen que se muestra en la figura 4.4, la misma fue segmentada manualmente para extraer las coordenadas de los puntos en el contorno, y así obtener un vector de referencia.



Figura 4.4: Imagen de referencia para el frente de Pareto.

En el frente de Pareto desarrollado en este proyecto, se utilizaron las siguientes funciones de aptitud:

- Precisión del algoritmo: Se calcula como el inverso del promedio de la función de distancia punto a punto \overline{D} , entre la figura de referencia y el resultado final del algoritmo de búsqueda. Esta función mide que tan bien el algoritmo llega al resultado correcto.

- Rendimiento: Esta función de aptitud se calcula como el inverso del tiempo en segundos necesitado para completar el algoritmo de búsqueda sobre una imagen, que se puede interpretar como el número de imágenes procesadas por segundo.
- Factor de memoria disponible: Se calcula como el inverso de la memoria RAM en bytes necesaria para ejecutar todo el programa. En la actualidad, con la gran capacidad de las computadoras, no constituye un factor determinante para escoger entre dos parametrizaciones; sin embargo, se incluyó para decidir los casos en que las otras funciones se encuentren con magnitudes similares.

Para encontrar las parametrizaciones de los algoritmos se utiliza el algoritmo PESA [1], implementado en una clase de la biblioteca LTI-Lib2 [14]. PESA es un algoritmo genético que toma la representación binaria de los parámetros y busca un frente de Pareto no dominado mediante la aplicación de mutación y cruzamiento entre los individuos, tratando de encontrar los que sean aptos [8]. Los valores de los parámetros utilizados en las dos evaluaciones, se muestran en la tabla 4.2.

Tabla 4.2: Valores de los parámetros utilizados en las evaluaciones del frente de Pareto

Parámetro	Valor
Probabilidad de cruce	70%
Probabilidad de mutación	3,03%
Población externa	1000
Poblacion interna	10
Iteraciones	10000

4.2.1 Resultados con algoritmo de búsqueda por gradiente (ABG)

El algoritmo de búsqueda por gradiente presentado por Cootes et al. [3] utiliza cuatro parámetros: el largo L , que corresponde a la longitud de la línea normal recorrida para cada punto (figura 2.8); El número r de iteraciones del algoritmo de búsqueda; el nivel umbral de la magnitud del gradiente Th_g y la cantidad de iteraciones de posición y orientación aleatorias m .

En la figura 4.5 se aprecia que el algoritmo puede llegar tan rápido para procesar 26 imágenes por segundo, sin embargo, con un aumento considerable en \bar{D} hasta 200 pixeles. En el otro extremo del frente de pareto, la parametrización adecuada puede lograr 8 imágenes por segundo y un valor de 22 pixeles para \bar{D} . El frente de Pareto obtenido

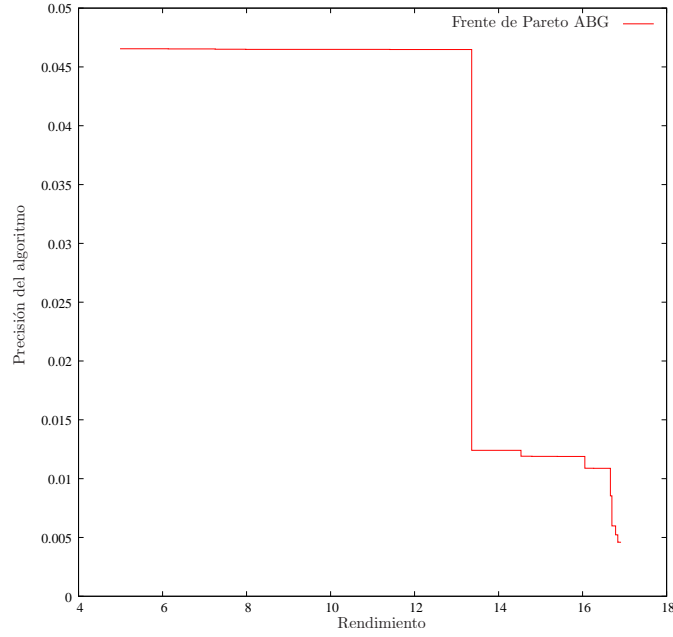


Figura 4.5: Frente de Pareto del algoritmo de búsqueda por gradiente

muestra que existe un cambio rápido en la precisión obtenida para el algoritmo, esto se debe principalmente al parámetro Th_g , ya que se utiliza para decidir, por medio de una comparación, si el valor máximo obtenido del gradiente corresponde a un borde y por lo tanto, existe un valor crítico que produce que el resultado final pierda precisión de manera abrupta.

En la tabla 4.3 se muestran los valores de los parámetros para obtener un rendimiento de 8 imágenes por segundo, y un valor de 22 píxeles para \bar{D} . Los resultados obtenidos para el algoritmo de búsqueda para tres imágenes utilizando los parámetros de la tabla 4.3 se muestra en figura 4.6, el contorno rojo denota la ubicación estimada del nematodo.

Tabla 4.3: Valores óptimos de los parámetros para ABG

Parámetro	Valor
L	5
r	5
Th_g	9
m	172

En la figura 4.6b y 4.6d se observa que el algoritmo de búsqueda logra encontrar la posición

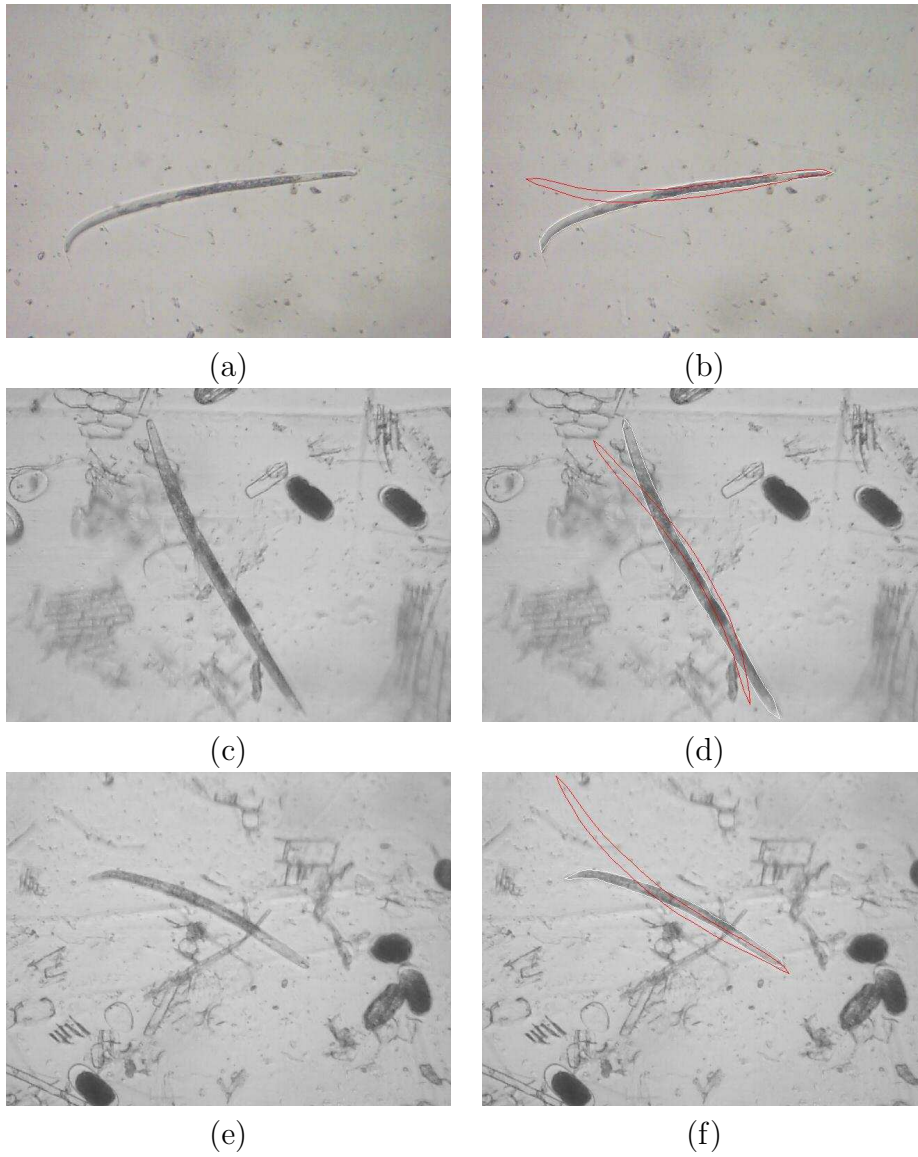


Figura 4.6: Resultado obtenido con ABG. El contorno rojo denota la aproximación final.

y orientación adecuados del nematodo, sin embargo, falla al intentar “encajar” los extremos del modelo de forma, esto sugiere que el modelo de forma utilizado puede no ser el adecuado para este tipo de figuras. Por otro lado en la figura 4.6f se muestra que el resultado no logra adecuar la escala del modelo, lo que demuestra que al igual que la posición y rotación, la escala debe ser inicializada con un valor próximo a la solución final.

4.2.2 Resultados de algoritmo de búsqueda con perfil de gris (ABPG)

En el caso del algoritmo propuesto en este trabajo para la búsqueda por medio del perfil de gris se utilizan cinco parámetros: el largo de la ventana w de medición; el largo L de recorrido, que corresponde a la longitud de las tres líneas recorridas para cada punto; la distancia entre cada una de las tres líneas recorridas d ; el número de iteraciones del algoritmo de búsqueda r y el número m de iteraciones aleatorias de posición y orientación. Para cada nuevo valor de w se debe extraer de nuevo la información del nivel de gris de las imágenes (ver sección 3.1.2), e invertir las matrices de covarianza, sin embargo el tiempo necesario para realizar esas operaciones no está contemplado en la función de aptitud de rendimiento, ya que esos cálculos sólo se realizan una vez durante el entrenamiento. La matriz de covarianza para cada uno de los n puntos es de dimensiones $3w \times 3w$, por lo tanto la función de aptitud de factor de memoria depende directamente del valor de w . La matriz de covarianza se utiliza en el cálculo de la distancia de Mahalanobis, éste cálculo se realiza Ln veces en cada iteración del algoritmo de búsqueda, la matriz de covarianza invertida continúa siendo simétrica, lo que permite reducir la cantidad de operaciones y por ende el tiempo de ejecución. Por ejemplo para una parametrización de $L = 10$, $w = 3$, $d = 2$, $r = 15$, $m = 2000$ el tiempo de ejecución se reduce desde 230 segundos hasta 32 segundos. La reducción de éste cálculo se muestra en el Apéndice B.

En la figura 4.7 se muestra el frente de Pareto obtenido para el algoritmo propuesto en este proyecto. El algoritmo puede alcanzar un rendimiento de hasta 18 imágenes por segundo, sin embargo el valor de \overline{D} incrementa hasta 150 píxeles, en el otro extremo, el algoritmo puede reducir \overline{D} hasta 8 píxeles, pero con un rendimiento de dos imágenes por minuto. El resultado en la figura 4.7 demuestra el compromiso que existe entre las funciones de aptitud de precisión y rendimiento del algoritmo, al incrementar la precisión deseada para el algoritmo de búsqueda se disminuye el rendimiento y si se quiere optimizar el tiempo de ejecución, se pierde precisión en el resultado.

En la tabla 4.4 se enlista el valor de los parámetros que produce el máximo valor en la precisión del algoritmo de búsqueda, dejando de lado la pérdida de rendimiento. En la figura 4.8 se muestran los resultados de aplicar el ABPG.

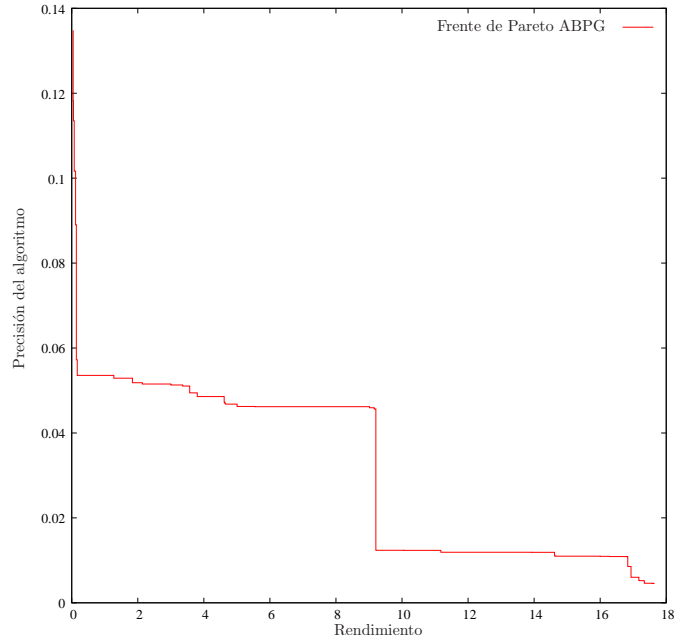


Figura 4.7: Frente de Pareto del algoritmo de búsqueda con perfil de gris

Tabla 4.4: Valores óptimos de los parámetros para ABPG

Parámetro	Valor
L	15
w	3
r	21
d	10
m	688

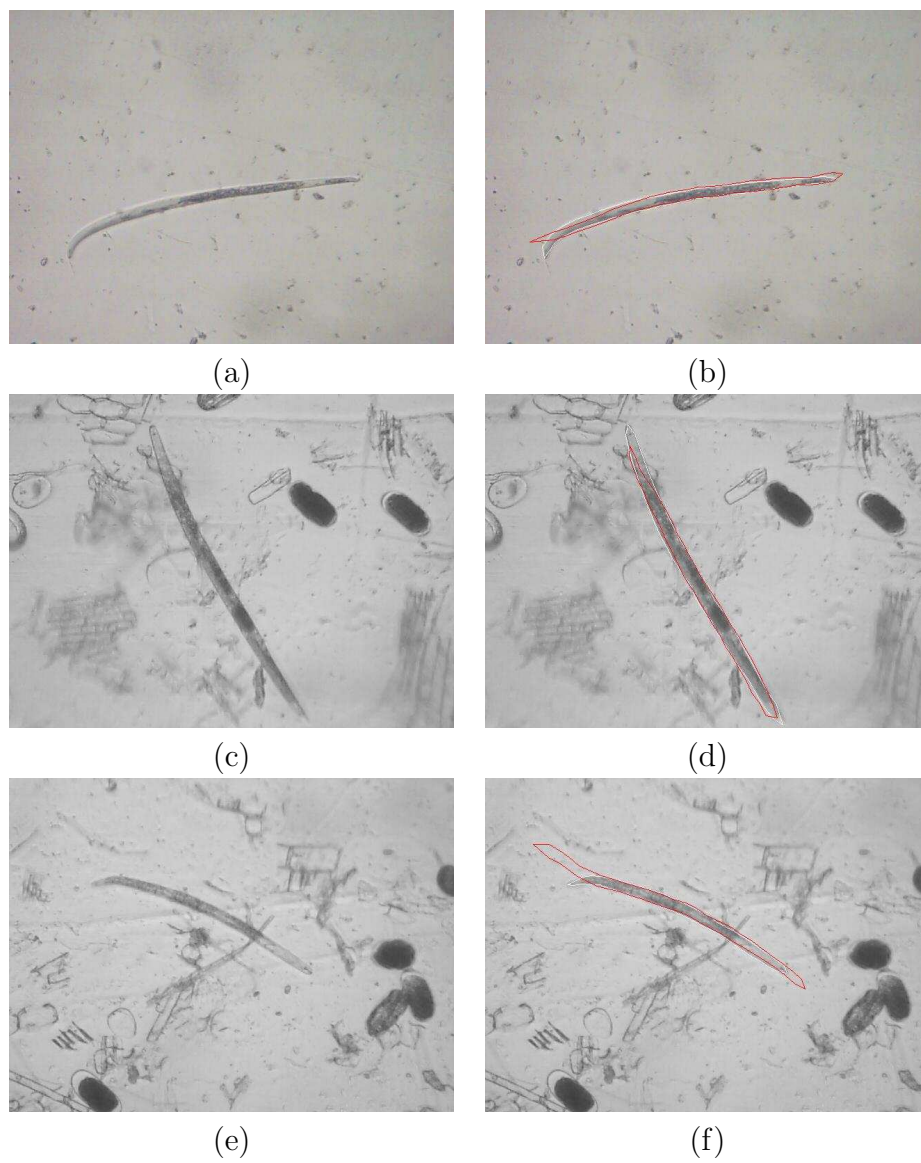


Figura 4.8: Resultado obtenido con ABPG. El contorno rojo denota la aproximación final.

En la figura 4.8b y 4.8d el algoritmo de búsqueda se desvía en los extremos del nematodo, en el caso de la figura 4.8f, la igual que en ABG, el algoritmo no logra ajustar la escala del modelo. El problema se debe a que, como se aprecia en la figura, los puntos del modelo correspondientes a los bordes laterales del nematodo se encuentran en la posición correcta, lo cual no permite que el modelo disminuya su escala, en este caso, se necesita que el modelo sólo disminuya su longitud, y no su ancho, sin embargo, esta operación no se puede llevar a cabo con el modelo de forma usado.

4.2.3 Comparación entre ABG y ABPG

En la figura 4.9 se muestra la comparación entre ABG y ABPG. El algoritmo ABPG obtiene un buen desempeño en la precisión del resultado final, lo mismo no ocurre en el rendimiento, que decae a menos de una imagen por segundo, sin embargo el objetivo principal del presente trabajo es determinar la ubicación del nematodo, y desde ese punto de vista, el algoritmo ABPG es el que presenta la mejor solución.

En la figura 4.10 se observa el resultado del frente de Pareto con las tres funciones de aptitud de ABG y ABPG. El algoritmo ABPG utiliza n matrices de covarianza de dimensiones $3w \times 3w$, esto implica que la cantidad de bytes de memoria RAM utilizados por el programa depende directamente del parametro w , por eso el frente de Pareto de ABPG en la figura 4.10 presenta diferentes niveles bien definidos. En el caso del algoritmo ABG la cantidad de memoria RAM ocupada por el programa no presenta una variabilidad significativa, ya que tres de los parámetros del algoritmo corresponden a números enteros y el cuarto al tamaño del vector utilizado para la búsqueda.

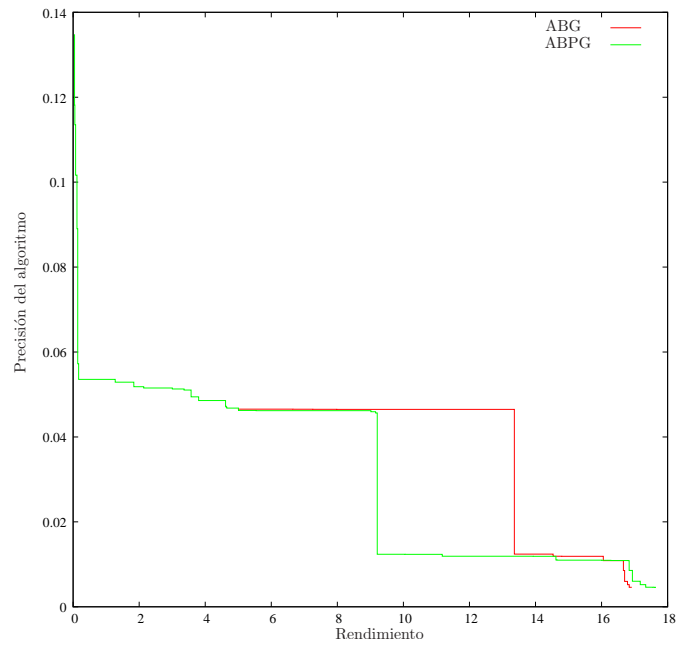


Figura 4.9: Comparación de ABG y ABPG

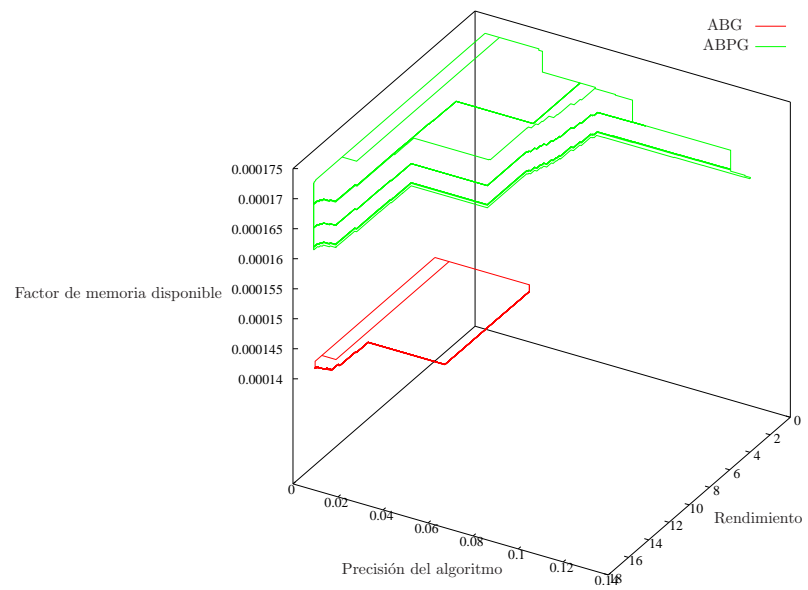


Figura 4.10: Comparación de ABG y ABPG con las tres funciones de aptitud

Capítulo 5

Conclusiones y recomendaciones

En este capítulo se presentan las principales conclusiones obtenidas como producto de la solución del problema planteado en este trabajo, además se incluye recomendaciones para la ejecución de trabajos posteriores relacionados con este proyecto.

5.1 Conclusiones

Se construyó una colección de cinco imágenes representativas para el entrenamiento, y se demostró que el modelo de forma sólo necesita de cuatro parámetros para cubrir el 99.99% de la variabilidad encontrada en el conjunto de las formas de entrenamiento.

El resultado final de la aproximación del algoritmo de búsqueda depende de los valores iniciales del algoritmo, los mejores resultados se obtienen si la orientación, la escala, y la posición inicial del modelo de forma se encuentran cercanos a los valores finales.

Para inicializar el algoritmo de búsqueda, se utilizó un algoritmo que asigna valores aleatorios a la posición y orientación del modelo de forma. Este proceso se repite para cada búsqueda realizada en la imagen.

Se comprobó que el modelo activo de forma encontrado es apto para describir los nematodos utilizados en las imágenes de referencia, ya que al evaluar el algoritmo con el frente de Pareto, se encontró que existen parámetros que minimizan la media del error por pixel entre cada punto del modelo de forma y la segmentación manual de referencia.

Al comparar el algoritmo de búsqueda de bordes por gradiente usado en [3] y el algoritmo de búsqueda de bordes por perfil de gris propuesto en este trabajo se logró determinar que

el segundo presenta un mejor desempeño para ubicar un nematodo.

El modelo de forma utilizado no puede representar todas las variaciones de forma en los cuales se encuentran los nematodos, razón que explica el desajuste que se obtiene con el algoritmo de búsqueda de bordes por perfil de gris propuesto en este trabajo.

El método de búsqueda utilizado se basa en la detección del borde por medio del recorrido de una línea perpendicular al contorno del modelo. Debido a este motivo, la solución propuesta no abarca las formas en las cuales el nematodo se encuentra en “espiral”, en forma de “s”, o un ejemplar mixto de las dos anteriores, ya que el algoritmo encuentra más de un posible borde para cada punto.

Los resultados obtenidos con el algoritmo de búsqueda de bordes por perfil de gris, propuesto en este trabajo, demuestran que es posible utilizando modelos activos de forma segmentar la silueta de un nematodo en una imagen digital (ver figura 4.8).

5.2 Recomendaciones

- a. En el caso de búsquedas realizadas en nematodos con forma “espiral” o en “s”, el algoritmo de búsqueda propuesto en este proyecto no puede aproximar la forma final del nemátodo. Se recomienda implementar el kernelPCA en el entrenamiento y en el algoritmo de búsqueda.
- b. Para mejorar el desempeño se recomienda incluir como etapa previa el trabajo realizado en [8] de segmentación basada en las características de textura, que filtre el fondo y los objetos que definitivamente no son nematodos, luego este resultado se puede utilizar como inicialización para el algoritmo propuesto en este proyecto.

Bibliografía

- [1] Pablo Alvarado. *Segmentation of color images for interactive 3D object retrieval*. Tesis doctoral. RWTH-Aachen, 2004.
- [2] N. Babaii, Rizvandi, A. Pizurica, and W. Phillips. Deformable shape description using active shape model (asm). *Department of Telecommunications and Information Processing(TELIN), Ghent University*, 2007.
- [3] T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, Enero 1995.
- [4] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Stochastic Modelling and Applied Probability. Springer-Verlag New York, Inc., 1996.
- [5] El Hogar Natural. Plagas [online, visitado el 10 de noviembre de 2009]. URL <http://www.elhogarnatural.com/Plagas.htm>.
- [6] Rafael Gonzales and Richard Woods. *Digital Image Processing*. Pearson Prentice Hall, Estados Unidos, 3ra edici008.
- [7] Tomás Guzmán. *Prospección, caracterización y evaluación de las relaciones de organismos benéficos para el control de nematodos patógenos en condiciones del trópico*. Proyecto de investigación. Instituto Tecnológico de Costa Rica, 2005.
- [8] Olman Eduardo Gómez Solórzano. Segmentación de nematodos en imágenes digitales usando redes neuronales. Master's thesis, Departamento de Computación, Instituto Tecnológico de Costa Rica, Cartago, Costa Rica, Marzo 2009.
- [9] Ghassan Hamarneh, Rafeef Abu-Gharbieh, and Tomas Gustavsson. Active shape models - part i: Modeling shape and gray level variations. *Department of Signals and Systems, Imaging and Image Analysis Group, Chalmers University of Technology, Göteborg,Sweden.*, 1998.

- [10] R. Haupt and S. Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.
- [11] INFOJARDIN. Nematodos [online, visitado el 10 de noviembre de 2009]. URL http://articulos.infojardin.com/PLAGAS_Y_ENF/PLAGAS/Nematodos.htm.
- [12] Jonathon Shiens. A tutorial on principal component analysis. Diciembre 2005, disponible en <http://www.sn1.salk.edu/shlens/pub/notes/pca.pdf>.
- [13] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson, Estados Unidos, 3ra edici008.
- [14] Sourceforge.net. LTI computer vision library [online, visitado el 10 de noviembre de 2009]. URL <http://ltilib.sourceforge.net>.
- [15] Universitat Politècnica de Catalunya. Epidemiología y manejo de nematodos fitoparásitos [online, visitado el 10 de noviembre de 2009]. URL <https://deab.upc.edu/investigacion/grupos-de-investigacion/pocio/1>.
- [16] University of Nebraska Lincoln Nematology. What are nematodes? [online, visitado el 10 de noviembre de 2009]. URL <http://nematode.unl.edu/wormgen.htm>.

Apéndice A

Cálculo para alinear un par de figuras

El cálculo para encontrar s_j, θ_j y t_j necesarios para alinear dos figuras similares x_1 y x_2 se muestra a continuación. Tomado de [3]

$$E = (x_1 - M(s, \theta) [x_2] - t)^T W (x_1 - M(s, \theta) [x_2] - t) \quad (\text{A.1})$$

en donde

$$M(s, \theta) = \begin{pmatrix} (s \cos \theta) x_{jk} - (s \sin \theta) y_{jk} \\ (s \sin \theta) x_{jk} - (s \cos \theta) y_{jk} \end{pmatrix} \quad (\text{A.2})$$

$$t = (t_x, t_y, \dots, t_x, t_y)^T \quad (\text{A.3})$$

W es una matriz diagonal de pesos para cada punto. Ahora, si se escribe:

$$a_x = s \cos \theta \quad a_y = s \sin \theta \quad (\text{A.4})$$

Se puede obtener por medio de una aproximación con mínimos cuadrados (diferenciación con respecto a las variables a_x, a_y, t_x, t_y) cuatro ecuaciones lineales,

$$\begin{pmatrix} X_2 & -Y_2 & W & 0 \\ Y_2 & X_2 & 0 & W \\ Z & 0 & X_2 & Y_2 \\ 0 & Z & -Y_2 & X_2 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} X_1 \\ Y_1 \\ C_1 \\ C_2 \end{pmatrix} \quad (\text{A.5})$$

En donde

$$X_i = \sum_{k=0}^{n-1} w_k x_{ik} \quad Y_i = \sum_{k=0}^{n-1} w_k y_{ik} \quad (\text{A.6})$$

$$Z = \sum_{k=0}^{n-1} w_k (x_{2k}^2 + y_{2k}^2) \quad W = \sum_{k=0}^{n-1} w_k \quad (\text{A.7})$$

$$C_1 = \sum_{k=0}^{n-1} w_k (x_{1k}x_{2k} - y_{1k}y_{2k}) \quad (\text{A.8})$$

$$C_2 = \sum_{k=0}^{n-1} w_k (y_{1k}x_{2k} - x_{1k}y_{2k}) \quad (\text{A.9})$$

Finalmente esto puede ser resuelto utilizando métodos estandar del álgebra lineal.

Apéndice B

Reducción del cálculo de la distancia de Mahalanobis

La distancia de Mahalanobis utilizada está dada por

$$f(d) = \sqrt{(\underline{\mathbf{h}}(d) - \underline{\mathbf{Y}}_{\mathbf{i}})^T \mathbf{C}_{\mathbf{Y}_{\mathbf{i}}}^{-1} (\underline{\mathbf{h}}(d) - \underline{\mathbf{Y}}_{\mathbf{i}})} \quad (\text{B.1})$$

Si se define

$$\underline{\mathbf{V}} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_m \end{bmatrix} \quad \underline{\mathbf{V}} = (\underline{\mathbf{h}}(d) - \underline{\mathbf{Y}}_{\mathbf{i}})$$

y la matriz de covarianza invertida es

$$\mathbf{C}_{\mathbf{Yi}}^{-1} = \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & \dots & c_{1,m} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & \dots & c_{2,m} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & \dots & c_{3,m} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & \dots & c_{4,m} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & c_{m,3} & c_{m,4} & \dots & c_{m,m} \end{pmatrix}$$

entonces

$$\underline{\mathbf{V}}^T \mathbf{C}_{\mathbf{Yi}}^{-1} \underline{\mathbf{V}} = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & \dots & v_m \end{bmatrix} \begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} & \dots & c_{1,m} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} & \dots & c_{2,m} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} & \dots & c_{3,m} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} & \dots & c_{4,m} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{m,1} & c_{m,2} & c_{m,3} & c_{m,4} & \dots & c_{m,m} \end{pmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_m \end{bmatrix}$$

El resultado de la multiplicación de filas por columnas es

$$\begin{aligned} \underline{\mathbf{V}}^T \mathbf{C}_{\mathbf{j}}^{-1} \underline{\mathbf{V}} = & v_1 (v_1 c_{1,1} + v_2 c_{2,1} + v_3 c_{3,1} + v_4 c_{4,1} + \dots + v_m c_{m,1}) + \\ & v_2 (v_1 c_{1,2} + v_2 c_{2,2} + v_3 c_{3,2} + v_4 c_{4,2} + \dots + v_m c_{m,2}) + \\ & v_3 (v_1 c_{1,3} + v_2 c_{2,3} + v_3 c_{3,3} + v_4 c_{4,3} + \dots + v_m c_{m,3}) + \\ & v_4 (v_1 c_{1,4} + v_2 c_{2,4} + v_3 c_{3,4} + v_4 c_{4,4} + \dots + v_m c_{m,4}) + \dots + \\ & v_m (v_1 c_{1,m} + v_2 c_{2,m} + v_3 c_{3,m} + v_4 c_{4,m} + \dots + v_m c_{m,m}) \end{aligned}$$

La matriz de covarianza invertida es simétrica, por lo tanto, $c_{i,j} = c_{j,i}$, lo que permite reducir las operaciones a

$$\begin{aligned}\underline{\mathbf{V}}^{\mathbf{T}}\mathbf{C}_{\mathbf{Yi}}^{-1}\underline{\mathbf{V}} &= v_1^2 c_{1,1} + 2v_1 \sum_{k=2}^m v_k c_{k,1} + \\ &v_2^2 c_{2,2} + 2v_2 \sum_{k=3}^m v_k c_{k,2} + \\ &v_3^2 c_{3,3} + 2v_3 \sum_{k=4}^m v_k c_{k,3} + \\ &v_4^2 c_{4,4} + 2v_4 \sum_{k=5}^m v_k c_{k,4} + \dots + \\ &v_m^2 c_{m,m}\end{aligned}$$

El resultado puede implementarse fácilmente con un par de bucles.