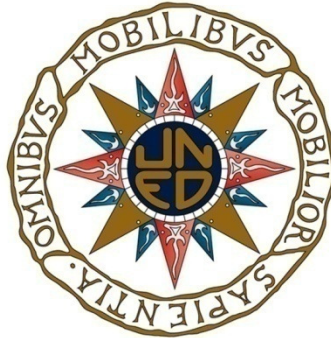


UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de Ingeniería Informática

Departamento de Inteligencia Artificial



Localización y Segmentación de la Cabeza del Nervio Óptico en Imágenes de Fondo de Ojo mediante Representación Piramidal y Algoritmos Genéticos

TRABAJO FINAL DEL MÁSTER OFICIAL EN INTELIGENCIA ARTIFICIAL:

FUNDAMENTOS, MÉTODOS Y APLICACIONES

José María Molina Casado

Ingeniero en Informática por la Universidad de Granada

Director

Enrique J. Carmona Suárez

Profesor titular del Departamento de Inteligencia Artificial

de la Universidad Nacional de Educación a Distancia

Resumen

La localización y segmentación de la cabeza del nervio óptico (CNO), también llamada disco óptico o papila, es de crítica importancia en el análisis de imágenes de retina, ya que apoya y facilita el diagnóstico de enfermedades oculares. Este Trabajo de Fin de Máster se centra en la creación de dos métodos automáticos para la segmentación de la cabeza del nervio óptico en imágenes fotográficas de retina. Ambos métodos utilizan una fase previa que consiste en la localización de la CNO. Una vez localizada, se extrae una subventana, centrada aproximadamente en la papila, con el objetivo de obtener un tamaño de imagen de entrada más pequeño y reducir así el coste computacional del proceso. Básicamente, esta fase previa transforma la subventana de entrada en una representación multinivel mediante la aplicación de una Pirámide Gaussiana. Se facilita así la búsqueda, en niveles altos de la pirámide, de píxeles que tienen una alta probabilidad de pertenecer a la CNO, dada la propiedad que poseen los píxeles centrales de la CNO de ser píxeles muy brillantes. Tras la fase previa, cada método aplica dos fases, una fase de obtención de puntos de interés (PI's) y otra fase de segmentación del contorno de la CNO mediante la utilización de Algoritmos Genéticos (AGs). Los PI's son puntos de la imagen que tienen una alta probabilidad de pertenecer al contorno de la CNO.

La estrategia seguida para implementar cada una de las dos fases mencionadas anteriormente varía en ambos métodos. Concretamente, la operativa del primer método comienza fusionando los tres canales RGB de la subventana de imagen en una nueva imagen, de un solo canal, mediante la ponderación fija y predeterminada de cada uno de los canales RGB. Seguidamente, la nueva imagen obtenida se transforma en una representación multinivel mediante una pirámide Laplaciana. Esta nueva representación facilita la búsqueda de puntos de alta frecuencia o, lo que es lo mismo, la búsqueda de PI's en cada uno de los niveles de la pirámide. A continuación, partiendo de que la CNO tiene forma ovalada, la búsqueda de su contorno en la imagen se aproxima por la búsqueda de aquella elipse que contiene, en una corona circunscrita a su perímetro, el mayor número de PI's. Esta búsqueda se realiza a partir de la combinación de varios algoritmos genéticos. Así, el primer AG (AG-1) realiza la búsqueda de la elipse papilar utilizando el conjunto de PI's obtenidos en el nivel-N. Seguidamente, un segundo AG (AG-2) realiza la búsqueda de una nueva elipse papilar, utilizando ahora el conjunto de PI's obtenidos en el nivel N-1 y, como población inicial, la población final de contornos elípticos obtenidos por AG-1. Esto significa que el resultado de la búsqueda

realizada en el nivel N se utiliza como punto de partida para comenzar la búsqueda en el nivel de pirámide N-1. Este proceso se repite tantas veces como sea necesario hasta llegar al nivel 0. De esta forma, se va realizando una búsqueda progresiva de la elipse papilar, desde el nivel de pirámide N hasta la imagen original.

Para evitar el esquema fijo de ponderación de los 3 canales RGB, utilizado en la primera etapa del método descrito anteriormente, en el segundo método presentado en este trabajo, la fusión de los tres canales se realiza mediante lo que se denomina fusión por Wavelets. La nueva imagen así fusionada, compuesta por un único canal, es procesada mediante diferentes filtros de imagen de detección de bordes (Canny y Sobel), con el objeto de obtener un conjunto de PI's. Finalmente, se utiliza un AG para la búsqueda de la mejor elipse papilar, siguiendo el mismo objetivo que en el método anterior. Sin embargo, obsérvese que aquí, a diferencia del método anterior, sólo se utiliza un único AG y la búsqueda se realiza sobre los PI's obtenidos a partir de la imagen fusionada, es decir, sobre el nivel 0.

Ambos métodos han sido probados en una base de datos de imágenes de fondo de ojo (DRIONS-DB)¹, descrita en [1]. Por otra parte, los resultados son comparados con los obtenidos en [1] y [2]. Los resultados obtenidos con el método de Pirámide Laplaciana son mejores que [2] y ligeramente inferiores a [1]. Sin embargo, comparado con [1], el proceso es simplificado notablemente, reduciendo así el coste computacional. Finalmente, los resultados obtenidos con el método de fusión por wavelets mejoran tanto los presentados en [1] como en [2].

¹ <http://www.ia.uned.es/personal/ejcarmona/DRIONS-DB.html>

Agradecimientos

En primer lugar deseo expresar mi más sincero agradecimiento a Enrique J. Carmona por su paciencia y por la confianza depositada en mí.

A mi gente de Cazorla, por tanto cariño y apoyo demostrado durante la duración del Máster. Y por último, y no menos importante, dar gracias por todo a mi familia.

Nomenclatura

- ✓ CNO : Cabeza del Nervio Óptico
- ✓ PI : Punto de Interés
- ✓ AG : Algoritmo Genético
- ✓ RPB : Región de Pixeles más Brillantes
- ✓ MPL : Método de la Pirámide Laplaciana
- ✓ MFWF : Método de Fusión por Wavelet y Filtrado
- ✓ TWD : Transformada Wavelet Discreta
- ✓ TWC : Transformada Wavelet Continua
- ✓ TDF : Transformada Discreta de Fourier
- ✓ SR : Segmento Radial

Índice de Contenidos

Resumen.....	2
Agradecimientos	4
Nomenclatura.....	5
Índice de Contenidos.....	6
Índice de Figuras y Tablas.....	8
Capítulo 1. Introducción.....	10
Capítulo 2 : Estado del Arte.....	13
Capítulo 3. Metodología.....	15
3.1 Fase previa: Localización de la cabeza del nervio óptico mediante una pirámide Gaussiana	15
3.2 Método basado en Pirámide Laplaciana (MPL)	21
3.2.1 Obtención de los Puntos de Interés	23
3.2.2 Búsqueda del contorno papilar basado en un AG de dos fases.....	25
3.2.2.1 Caracterización del AG en dos fases	25
3.3 Método basado en fusión por wavelet y filtrado (MFWF).....	28
3.3.1 Obtención de los Puntos de Interés	29
3.3.2 Búsqueda del contorno papilar basado en un AG de una fase	33
Capítulo 4 . Casos de estudio	34
4.1 Método basado en Pirámide Laplaciana (MPL)	35
4.1.1 Resultados experimentales	35
4.1.2 Evaluación y comparación de resultados	35
4.2 Método basado en fusión por Wavelet y filtrado (MFWF).....	38
4.2.1 Resultados experimentales	38
4.2.2 Evaluación y comparación de resultados	38
Capítulo 5. Conclusiones y trabajo futuro.....	41
Apéndice A. Manual de Usuario de la aplicación.....	42
Menú - Fichero	43
Menú - BBDD Imágenes	44
Menú – Puntos de Interés.....	45
Menú - Detección de Papila	48
Menú - Segmentación de Papila	48
Menú - Ayuda	50

Apéndice B. Código	51
Fichero PRINCIPAL.M para el método MPL.....	51
Fichero DETECPI23.M para el método MPL	52
Fichero PAPILALAPLACE.M para ambos métodos.....	59
Fichero PRINCIPAL.M para el método MFWF	61
Fichero DETECPI23.M para el método MFWF.....	62
Bibliografía	68

Índice de Figuras y Tablas

Figura 1: Esquema de la retina y la localización del contorno del CNO	10
Figura 2 : Pirámide de Imagen. El tamaño de cada nivel es $2^j \times 2^j$, donde $0 \leq j \leq J$, siendo J el tamaño de nivel base	16
Figura 3 : Ejemplo de pirámide Gaussiana para una imagen de fondo de ojo, mostrando los Niveles 0,1 y 2	17
Figura 4 : Diagrama de bloques del proceso de localización de papila y extracción de subventana	19
Figura 5 : Un ejemplo que muestra las diferentes fases de la tarea de localización de CNO y extracción de subventana. (a) imagen original en el Nivel-0; (b-d) canal rojo de la pirámide Gaussiana, mostrando los niveles 1,2 y 3, respectivamente; (e) region de los píxeles más brillantes (RPB) en el nivel 3, representado por los píxeles negros centrados. (f-h) pirámide Gaussiana inversa (conteniendo el RPB): niveles 2, 1 y 0 respectivamente; (i) subventana final centrada en el centroide del RPB en el Nivel-0.....	20
Figura 6 : Ejemplo de pirámide Laplaciana para una imagen de fondo de ojo, mostrando los Niveles 0,1 y 2	22
Figura 7 - Diagrama de Bloques correspondiente al método de la pirámide Laplaciana	23
Figura 8 : Puntos obtenidos en el Nivel-2 (a) y en el Nivel-1 (b) de la pirámide Laplaciana. La línea continua representa el contorno papilar solución y las líneas discontinuas representa la corona elíptica asociada.....	24
Figura 9 : Diagrama de Bloques correspondiente al método basado en fusión por wavelet y filtrado	29
Figura 10 - Proceso de Fusión de los canales RGB mediante TWD. Inicialmente se fusionan los canales Verde y Azul, el resultado es fusionado con el canal Rojo.....	32
Figura 11 - Obtención de los PI's tras aplicar los filtros de Canny y Sobel sobre la imagen fusionada.....	33
Figura 12 : Comparativa de la discrepancia acumulada para los métodos de Carmona, Lowell y MPL.....	36
Figura 13 : Elipses Genéticas obtenidas con el método MPL (línea continua) comparadas con el Gold Standard (línea punteada). También se muestra el valor de discrepancia obtenido.....	37
Figura 14 : Comparativa de la discrepancia acumulada para los métodos de Carmona, Lowell y MFWF	39
Figura 15 : Elipses Genéticas obtenidas con el método MFWF (línea continua) comparadas con el Gold Standard (línea punteada). También se muestra el valor de discrepancia obtenido.....	40
Figura 16 : Pantalla principal de la interfaz gráfica	42
Figura 17 : Opciones del Menú Fichero.....	43
Figura 18 : Imagen cargada en el tapiz central	43
Figura 19 : Zoom sobre la imagen cargada sobre el tapiz central.....	44
Figura 20 : Panel de control de la opción BBDD Imágenes	45
Figura 21 : Menú Puntos de Interés	45
Figura 22: Obtención de PIs por el método MPL	46
Figura 23 : PI's obtenidos en los Niveles 0, 1 y 2 de la Pirámide Laplaciana para una imagen de fondo de ojo determinada.	47

Figura 24 : PI's obtenidos con el método MFWF	47
Figura 25 : Localización del Contorno Papilar	48
Figura 26 : Resultado de la segmentación con el MPL.....	49
Figura 27 : Resultado de la segmentación con MFWF	50

Tabla 1 : Dominio de definición (expresado en píxeles) para cada uno de los genes (variables) de un cromosoma (elipse genética)	26
--	----

Tabla 2 : Resumen de la configuración de los dos AG.....	27
--	----

Capítulo 1. Introducción

El nervio óptico se encuentra situado en la parte posterior del ojo y es el encargado de transmitir la información visual desde la retina hasta el cerebro. La localización y segmentación de la cabeza del nervio óptico (CNO) es de crítica importancia en el análisis de imágenes de retina ya que apoya y facilita el diagnóstico de enfermedades oculares. El CNO, también llamado disco óptico o papila, tiene forma ovalada y está localizada en el área donde convergen las fibras nerviosas para formar el comienzo del nervio óptico, abandonando el globo ocular por su parte posterior. En el centro del CNO hay una zona sin fibras nerviosas denominada excavación y, alrededor de ésta, existe otra zona, algo más elevada que el resto de la retina, denominada anillo neuro-retiniano. Este anillo tiene, en condiciones normales, una coloración anaranjado o rosáceo y su perímetro externo delimita finalmente el contorno papilar. En la Figura 1 se muestra un esquema de la retina y la localización del CNO.

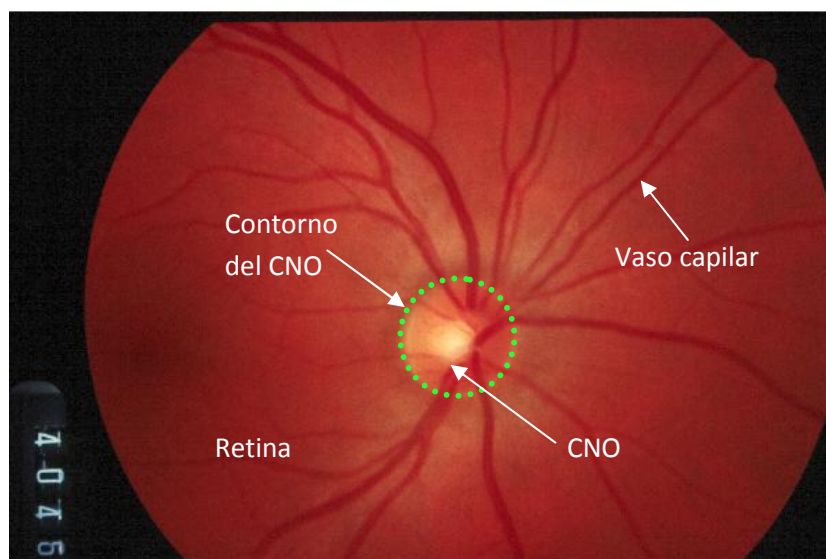


Figura 1: Esquema de la retina y la localización del contorno del CNO

En este trabajo, se han implementado dos métodos para localizar y segmentar el CNO. Ambos métodos utilizan una fase previa que consiste en la localización de la CNO. Una vez localizada, se genera una representación multinivel mediante una pirámide Gaussiana a fin de extraer una subventana, la cual se encuentra centrada aproximadamente en la papila. El objetivo de este paso es obtener un tamaño de imagen de entrada más pequeño y reducir así el coste computacional del proceso. Se facilita así la búsqueda, en niveles altos de la pirámide, de píxeles que tienen una alta probabilidad de pertenecer a la CNO, dada la propiedad que

poseen los píxeles centrales de la CNO de ser píxeles muy brillantes. Tras la fase previa, cada método aplica dos fases, una fase de obtención de puntos de interés (PI's) y otra fase de segmentación del contorno de la CNO mediante la utilización de Algoritmos Genéticos (AGs). Los PI's son puntos de la imagen que tienen una alta probabilidad de pertenecer al contorno de la CNO.

El primer método comienza fusionando los tres canales RGB de la subventana de imagen en una nueva imagen, de un solo canal, mediante la ponderación fija y predeterminada de cada uno de los canales RGB. Posteriormente, la nueva imagen obtenida se transforma en una representación multinivel mediante una pirámide Laplaciana, a fin de facilitar la búsqueda de PI's en cada uno de los niveles de la pirámide. A continuación, partiendo de que la CNO tiene forma ovalada, la búsqueda de su contorno en la imagen se aproxima por la búsqueda de aquella elipse que contiene, en una corona circunscrita a su perímetro, el mayor número de PI's. Esta búsqueda se realiza a partir de un Algoritmo Genético (el primer AG ,AG-1, realiza la búsqueda de la elipse papilar con los PI's del nivel-N y el segundo AG, AG-2, utiliza los PI's obtenidos en el nivel N-1, utilizando como población inicial la población final de contornos elípticos obtenidos por AG-1). El resultado de la búsqueda realizada en el nivel N se utiliza como punto de partida para comenzar la búsqueda en el nivel de pirámide N-1. Este proceso se repite tantas veces como sea necesario hasta llegar al nivel 0. De esta forma, se va realizando una búsqueda progresiva de la elipse papilar desde el nivel de pirámide N hasta la imagen original.

En el segundo método presentado en este trabajo, la fusión de los tres canales se realiza mediante lo que se denomina fusión por Wavelets (para evitar el esquema fijo de ponderación de los 3 canales RGB). La nueva imagen fusionada es procesada mediante diferentes filtros de imagen de detección de bordes (Canny y Sobel), con el objeto de obtener un conjunto de PI's. Finalmente, se utiliza un AG para la búsqueda de la mejor elipse papilar, siguiendo el mismo objetivo que en el método anterior. Es importante reseñar que en este método sólo se utiliza un único AG y la búsqueda se realiza sobre los PI's obtenidos a partir de la imagen fusionada, es decir, sobre el nivel 0.

Los dos métodos aquí presentados son una variación del método descrito en [1]. La principal diferencia con éste radica en la forma de obtener el conjunto de PI's. Por un lado, en [1], los puntos se obtenían utilizando un método basado en conocimiento del dominio y, por otro lado, en este trabajo, dichos puntos se obtienen sin hacer uso de dicho conocimiento. Así,

en nuestro primer método, los PI's se obtienen mediante una pirámide Laplaciana y, en el segundo método, mediante la aplicación de filtros de imagen para la obtención de puntos de borde.

Esta memoria está organizada de la siguiente forma: en el Capítulo 2 se comenta el estado del arte relacionado con el problema aquí abordado. En el Capítulo 3 se describe detalladamente la metodología utilizada en los dos métodos propuestos. En el Capítulo 4, los resultados obtenidos son evaluados y comparados con otros métodos. En el Capítulo 5 se presentan las conclusiones y las propuestas para futuros trabajos. Por último, en el Capítulo 6, se incluye un apéndice, el manual de usuario y el código de la aplicación.

Capítulo 2 : Estado del Arte

Los sistemas de apoyo para el diagnóstico de enfermedades oculares basadas en la información de imágenes de fondo de ojo, implican la localización semiautomática y automática del contorno papilar. En concreto, los primeros métodos semiautomáticos se basaban en propiedades geométricas de los píxeles de la imagen y sus valores de intensidad. A modo de ejemplo, en [3], Cox y Wood sugieren que los límites de la CNO cambian su silueta conforme la enfermedad del glaucoma va progresando. El método utiliza realce, umbralización y filtros de imagen. Para la segmentación, los expertos marcan cuatro puntos con una alta probabilidad de pertenecer al contorno papilar, luego el algoritmo traza una elipse conteniendo esos cuatro puntos en una fase de aproximación inicial, para finalmente refinar el contorno papilar utilizando los niveles de gris cercanos a esta elipse.

Otras aproximaciones basadas en contornos activos (*snakes*) han demostrado ser útiles para este tipo de problema. Sin embargo, una propiedad característica de este tipo de técnicas es la alta dependencia de una etapa preliminar de inicialización del contorno, a partir de la cual se obtiene la solución final mediante refinamiento. En algunos estudios, esta etapa de inicialización ha sido hecha de forma manual [4], y en otros casos de forma automática [5] y [6]. Así, en [4], el éxito del algoritmo es muy dependiente del preprocesamiento de la imagen para mejorar el contraste entre la retina y la CNO. Dicho preprocesamiento se lleva a cabo mediante ecualización de histograma y un detector de bordes basado en una representación piramidal. En [2], utilizando conocimiento del dominio, la localización del disco óptico es alcanzada utilizando *template matching* (método de filtrado utilizado para detectar características particulares en una imagen) y la segmentación se realiza mediante contornos activos (también conocidos como *snakes*). El método selecciona de forma automática la localización del centro de la CNO para luego realizar la segmentación del contorno. En [5], se utiliza un método basado en dos etapas, en una primera etapa se procesa la imagen utilizando morfología matemática del nivel de gris a fin de eliminar las regiones de los vasos sanguíneos, reemplazándolos por píxeles representativos del fondo del disco óptico. Luego un *snake* es manualmente situado alrededor del disco óptico para evolucionar en su límite. Este *snake* es una versión mejorada, un contorno activo con fuerza externa de flujo de vector de gradiente (GVF). En [6] se utiliza un simple *template matching* para estimar la posición del centro del disco, la cual permita la inicialización del *snake* de forma automática. En primer lugar, se preprocesa la imagen dilatándola a fin de eliminar los vasos sanguíneos para después

restaurarla mediante una erosión. En el segundo paso, se aplica un *snake-GVF* para realizar la segmentación. Este método utiliza un radio el cual es calculado con un algoritmo basado en mínimos cuadrados.

Los Algoritmos Genéticos también han sido utilizados de forma eficiente como un método de búsqueda para identificar el contorno papilar en [1] y [7]. Así, en [1], se localiza y segmenta la CNO mediante un AG. En este trabajo, el perímetro papilar se modela mediante una elipse de puntos que tienen la propiedad de poseer un alto contraste de intensidad respecto a sus vecinos y que se denominan puntos hipótesis (PH's). En este contexto, el AG realiza la búsqueda en la imagen de aquella elipse que contenga el mayor número de PH's en su perímetro, proporcionando evidencia experimental de que esta estrategia ofrece soluciones muy cercanas al contorno papilar buscado. El conjunto de PH's se obtiene mediante un método basado en conocimiento del dominio. En [7], se propone un método de localización mediante Redes Activas Topológicas. Se trata de un modelo deformable que integra características de técnicas basadas en segmentación de regiones y segmentación de bordes, siendo capaz de ajustar los bordes de los objetos y modelar su topología interior. En este trabajo las redes activas incorporan nuevos términos de energía para la localización del CNO y su optimización, realizando su optimización mediante un GA.

Otras recientes aproximaciones a la solución del problema han utilizado diferentes técnicas, como por ejemplo, morfología matemática [8], filtros de imagen [9] o la transformada de Hough [10]. En [8], se utiliza la morfología matemática (teoría basada en conceptos de geometría, álgebra, topología y teoría de conjuntos) a fin de examinar las estructuras geométricas de las imágenes de retina por superposición de pequeños patrones (elementos estructurantes). Esta técnica permite procesar las imágenes de fondo de ojo con el objetivo de segmentar la CNO. En [9], se tiene en cuenta que la CNO está cerca del punto focal de convergencia de los vasos sanguíneos. En primer lugar se detectan los vasos sanguíneos mediante filtros de Gabor, a continuación se detecta la orientación de los vasos sanguíneos para detectar los puntos de convergencia. En [10], se describe un método para localizar la CNO mediante el filtro de Sobel y la utilización de la transformada de Hough para detectar círculos.

Capítulo 3. Metodología

El objetivo de este trabajo es localizar y segmentar el CNO en imágenes fotográficas de fondo de ojo y para ello se han propuesto dos métodos. En primer lugar, con el fin de localizar la CNO, se realiza una fase previa común para ambos métodos. Además, como resultado final de esta fase, se extrae una subventana con el fin de reducir el coste computacional de las etapas de procesamiento posteriores. La subventana extraída se encuentra centrada aproximadamente en el centro del área papilar, de esta forma, la CNO es localizada y el proceso siguiente de segmentación se centra en el área de interés de la imagen, descartando el resto. Posteriormente, cada uno de los dos métodos aquí propuesto opera de distinta forma manera, aunque el objetivo de ambos es común: segmentar la papila. A continuación se detalla la fase común de localización y extracción de subventana, y el resto de fases de cada uno de los métodos propuestos.

3.1 Fase previa: Localización de la cabeza del nervio óptico mediante una pirámide Gaussiana

Es posible obtener características diferentes de los objetos de una imagen a diferentes escalas, utilizando filtros que suavizan gradualmente las características de alta frecuencia de dichos objetos en función de la escala utilizada. Una herramienta poderosa para manipular información de multiescalado son las llamadas pirámides de imágenes. Se desarrollaron originalmente para aplicaciones de visión por computadora y métodos de compresión. Una pirámide de imágenes es una colección de imágenes con resolución decreciente dispuestas en forma de pirámide. En la Figura 2 podemos ver una representación de la estructura piramidal.

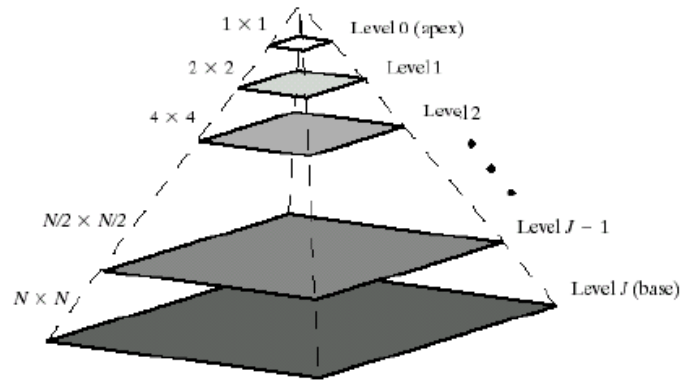


Figura 2 : Pirámide de Imagen. El tamaño de cada nivel es $2^j \times 2^j$, donde $0 \leq j \leq J$, siendo J el tamaño de nivel base

La imagen de la base de la pirámide contiene la más alta resolución, mientras que la situada en la punta de la pirámide contiene la más baja resolución. Por tanto, conforme nos movemos hacia arriba de la pirámide, tanto el tamaño como la resolución de las imágenes disminuyen.

La pirámide Gaussiana fue presentada por primera vez por Burt&Adelson [11] en el contexto de compresión de imágenes. En una pirámide de tipo Gaussiana cada nivel de la pirámide es suavizado por un kernel Gaussiano simétrico y submuestreado para obtener la siguiente capa. La siguiente formula representa los cálculos para producir la pirámide Gaussiana (Notacion: S^\downarrow submuestra la imagen I):

$$P_{Gaussiana}(I)_{j-1} = S^\downarrow (G_\sigma \otimes P_{Gaussiana}(I)_j)$$

donde, $P_{Gaussiana}(I)_j = I$; es la original

En la Figura 3 podemos ver un ejemplo de representación piramidal Gaussiana para una imagen de fondo de ojo.



Figura 3 : Ejemplo de pirámide Gaussiana para una imagen de fondo de ojo, mostrando los Niveles 0,1 y 2

Algunas de las aplicaciones de este tipo de representación son:

- Búsqueda de escala: Muchos objetos pueden ser representados con pequeños patrones. Por ejemplo las caras tiene patrones muy bien definidos: dos ojos hundidos en fondos oscuros, debajo de dos rallas oscuras (cejas), separados de una luz especular (nariz) y sobre una barra oscura (boca). En una pirámide podemos buscar caras grandes medianas y pequeñas a lo largo de sus diferentes capas.
- Búsqueda espacial: En visión estereoscópica, o análisis de movimiento, se buscan pares de puntos que casen en una escala gruesa y luego se va buscando en escalas más finas, con más detalle pero en zonas de búsqueda significativamente menores.
- Seguimiento de características (tracking): la mayoría de las características encontradas en escalas gruesas están asociadas con cambios bruscos de contraste. Típicamente, encontrar objetos en escalas gruesas subestima tanto el tamaño como la localización de los objetos. Por ejemplo un error de un sólo pixel a escala gruesa representa un error de múltiples pixeles en escala fina. En escalas finas existen muchos eventos asociados con cambios pequeños de contrastes bajos. Una estrategia para mejorar la caracterización de objetos en escalas finas es hacer el seguimiento de la característica de escalas finas a gruesas y quedarse con aquellas

que tiene sus respectivos padres a lo largo de la pirámide (por ejemplo, diferencia entre ruido y estructuras de ciertas texturas).

Centrándonos nuevamente en nuestro método y, concretamente, en lo que hemos denominado *fase previa*, se aplica una pirámide Gaussiana en N niveles de la imagen original (600x400 píxeles). La idea es suavizar progresivamente los niveles de intensidad de la imagen con el objetivo de eliminar las altas frecuencias. De hecho, podemos ver la pirámide Gaussiana como un conjunto de copias de la imagen original, cada una de ellas con una resolución que es la mitad que la copia del nivel anterior y filtradas mediante un filtro paso bajo. En cada nivel de la pirámide, la intensidad de los píxeles pertenecientes a la retina, los vasos capilares y la papila son suavizados en diferentes intervalos de intensidad. El proceso de suavizado es más pronunciado conforme aumenta el nivel de la pirámide. Aquí utilizamos la característica por la cual la mayoría de los píxeles de la papila tienen un nivel de intensidad más alto que los píxeles de la retina y el de los vasos capilares. Se comprueba experimentalmente que, para un valor de N adecuado, en el Nivel-N, la región de los píxeles más brillantes corresponderá a píxeles pertenecientes a la papila.

Los pasos empleados en localizar la papila se podrían resumir de la siguiente manera:

- (1) Se calcula la pirámide Gaussiana de la imagen de fondo de ojo (Nivel 0 de pirámide).
- (2) Se localiza la región de los píxeles más brillantes (RBP) en el Nivel-N de la pirámide.
- (3) Se calcula la pirámide Gaussiana inversa del Nivel-N, expandiéndolo hasta el Nivel-0 y considerando en dicha expansión la RPB obtenida en el paso anterior.
- (4) Se calcula el centroide de la RBP en el nivel 0 y este punto permite determinar dónde está localizado aproximadamente el centro de la papila.
- (5) Por último, se extrae de la imagen original una subventana tomando como referencia el centroide calculado anteriormente. El resultado es una subventana de tamaño 261x261 centrada aproximadamente en la zona papilar con el objeto de obtener una nueva imagen con un número de píxeles menor y reducir así el coste computacional de las siguientes fases.

En la Figura 4 podemos ver un diagrama de bloques del proceso de extracción.

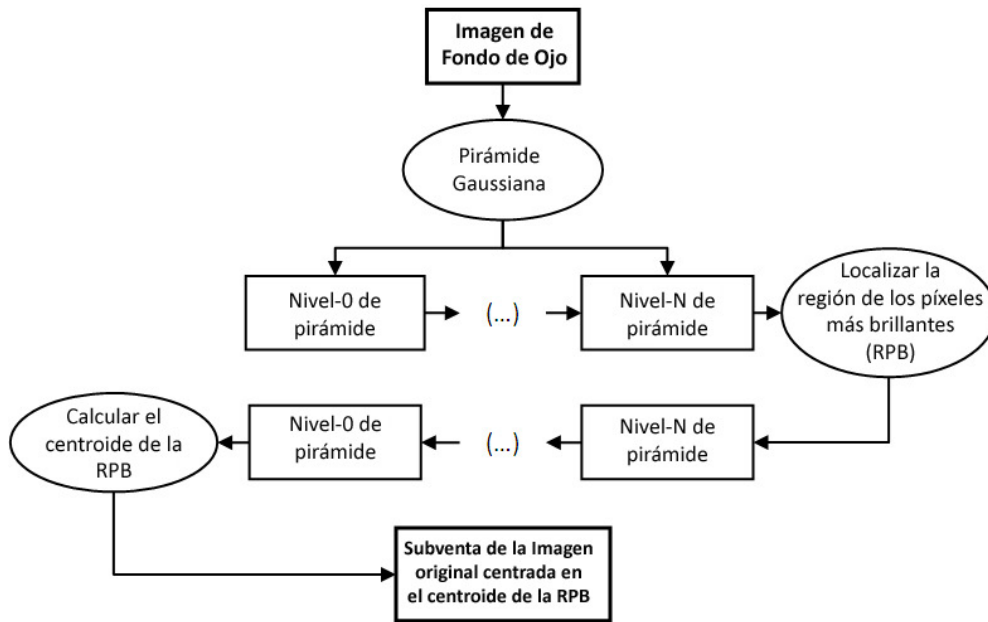


Figura 4 : Diagrama de bloques del proceso de localización de papila y extracción de subventana

Una idea similar es usada en [12], donde la pirámide es creada utilizando una Transformación Discreta Wavelet (TWD). De la misma forma, en [13], se aplica un umbral a cada uno de los tres canales RGB con el objetivo de obtener una primera aproximación al área papilar.

De forma experimental, este método para la localización de la región papilar se ha probado con la base de imágenes DRIONS [14], obteniendo un 100% de efectividad. En dicho experimento sólo fue necesario utilizar el canal rojo de la imagen y una pirámide Gaussiana de 3 niveles.

Finalmente, la Figura 5 muestra un ejemplo de las diferentes fases utilizadas para la localización de CNO y extracción de subventana sobre una imagen de fondo de ojo.

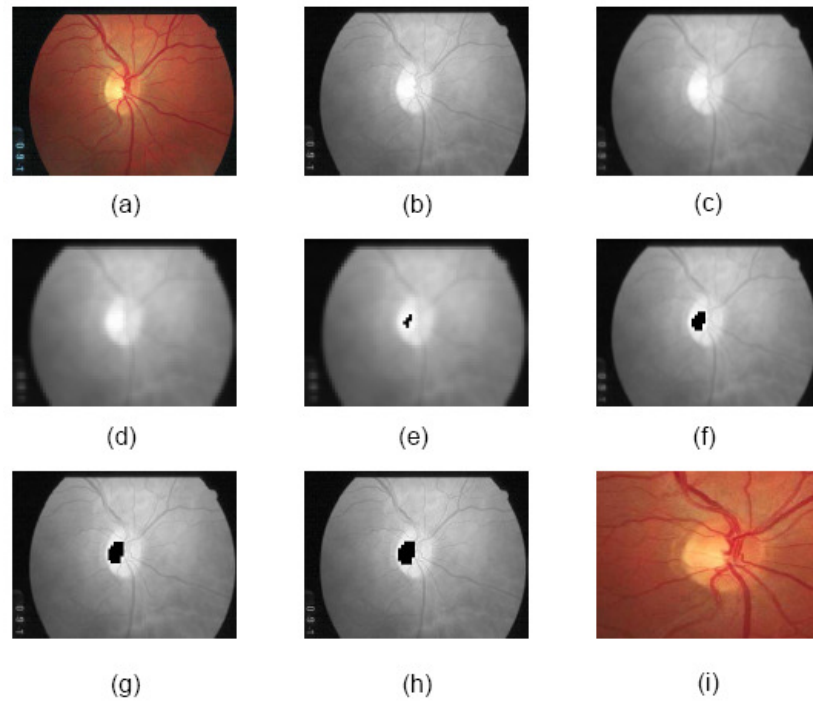


Figura 5 : Un ejemplo que muestra las diferentes fases de la tarea de localización de CNO y extracción de subventana. (a) imagen original en el Nivel-0; (b-d) canal rojo de la pirámide Gaussiana, mostrando los niveles 1,2 y 3, respectivamente; (e) región de los píxeles más brillantes (RPB) en el nivel 3, representado por los píxeles negros centrados. (f-h) pirámide Gaussiana inversa (conteniendo el RPB): niveles 2, 1 y 0 respectivamente; (i) subventana final centrada en el centroide del RPB en el Nivel-0.

3.2 Método basado en Pirámide Laplaciana (MPL)

La pirámide Laplaciana fue presentada por primera vez por Burt&Adelson [11] en el contexto de compresión de imágenes. Sin embargo, esta tecnología ha sido probada de forma efectiva en otras tareas de análisis de imágenes: generación y reconstrucción, transmisión progresiva, o en la detección y realce de rasgos. De la misma forma que podemos ver la pirámide Gaussiana como un conjunto de copias de la imagen original a diferentes escalas y filtradas con un filtro paso bajo, también podemos ver la pirámide Laplaciana como un conjunto de copias de la imagen original a diferentes escalas y filtradas con un filtro pasa banda. Es decir, al aplicar esta pirámide, las altas y bajas frecuencias son eliminadas, por tanto, la idea es conservar los rasgos de la imagen en los distintos niveles de la pirámide. La escala del operador Laplaciano dobla su valor de nivel a nivel, mientras que el punto de frecuencia central del filtro pasa banda se reduce en un octavo. Así, la pirámide Laplaciana permite una representación de la imagen original en una jerarquía de imágenes de forma que cada nivel corresponde a diferentes bandas de frecuencias.

Por tanto, las pirámides Laplacianas hacen uso del hecho de que la capa de nivel N de la pirámide Gaussiana predice la apariencia de la capa de nivel N-1. Si utilizamos un operador de sobremuestreo que pueda producir una versión de capa de nivel N del mismo tamaño que la siguiente capa de nivel N-1, entonces sólo necesitamos almacenar la diferencia entre estas dos predicciones y la siguiente de nivel N-1. Es decir, es una secuencia de imágenes de error, diferencias de dos capas de la pirámide Gaussiana. Cada una de las capas de nivel N-1 de la pirámide Laplaciana es la diferencia entre una capa de pirámide Gaussiana y una predicción obtenida sobremuestreandola siguiente capa Gaussiana de la pirámide. La siguiente formula representa los cálculos para producir la pirámide Laplaciana (Notacion: S^{\uparrow} sobremuestrea la imagen I):

$$P_{Laplaciana}(I)_k = P_{Gaussiana}(I)_k - S^{\uparrow}(P_{Gaussiana}(I)_{k+1})$$

En la Figura 6 podemos ver un ejemplo de representación piramidal Laplaciana para una imagen de fondo de ojo.

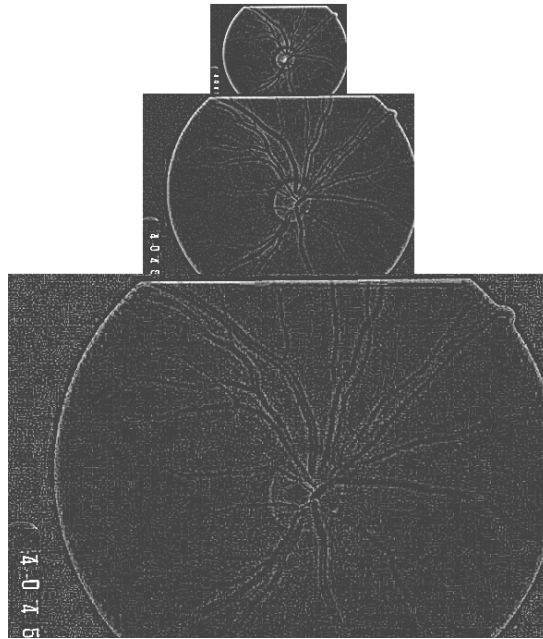


Figura 6 : Ejemplo de pirámide Laplaciana para una imagen de fondo de ojo, mostrando los Niveles 0,1 y 2

Algunas de las aplicaciones de este tipo de representación son:

- Aplicación de compresión de imágenes: La naturaleza de este tipo de pirámides permite reducir el número de bits necesarios para codificar la imagen con ligeras pérdidas de calidad.
- Detección de bordes: este tipo de pirámides se utilizan para mejorar la nitidez de la imagen por su capacidad de captar altas discontinuidades como resultado de aplicar la segunda derivada.

Nuestro primer método de segmentación de la CNO, al que hemos denominado método basado en Pirámide Laplaciana (MPL), se basa en la pirámide Laplaciana para la obtención de los PI's. Dicha pirámide es aplicada a la subimagen resultante de la fase previa con el objetivo de obtener una representación multiescalar. En cada nivel de la pirámide, se obtiene un conjunto de PI's, correspondiente a puntos de media-alta frecuencia, entre los cuales se encuentran los puntos correspondientes al borde del contorno papilar además de otros puntos que serán fuente de ruido. En nuestros experimentos fue suficiente con llegar hasta el nivel de pirámide $N=2$. Posteriormente se utiliza un Algoritmo Genético en dos fases: AG-2 y AG-1. El objetivo es obtener una elipse que contenga el máximo número de PI's en su perímetro. La justificación de esta estrategia se justificará en la siguiente subsección. Con los PI's calculados en el Nivel-2 se aplica el AG-2 y con las mejores elipses solución obtenidas en

esta fase se inicializa parte de la población del AG-1. La otra parte de la población inicial de AG-1 se hace de forma aleatoria. Finalmente, el AG-1 se aplica al conjunto de PI's obtenidos en el Nivel-1, escogiendo la mejor elipse como solución de nuestro problema. En la Figura 7, podemos ver un diagrama de bloques resumen del método propuesto y en los apartados siguientes se detallan todas los pasos utilizados por este método.

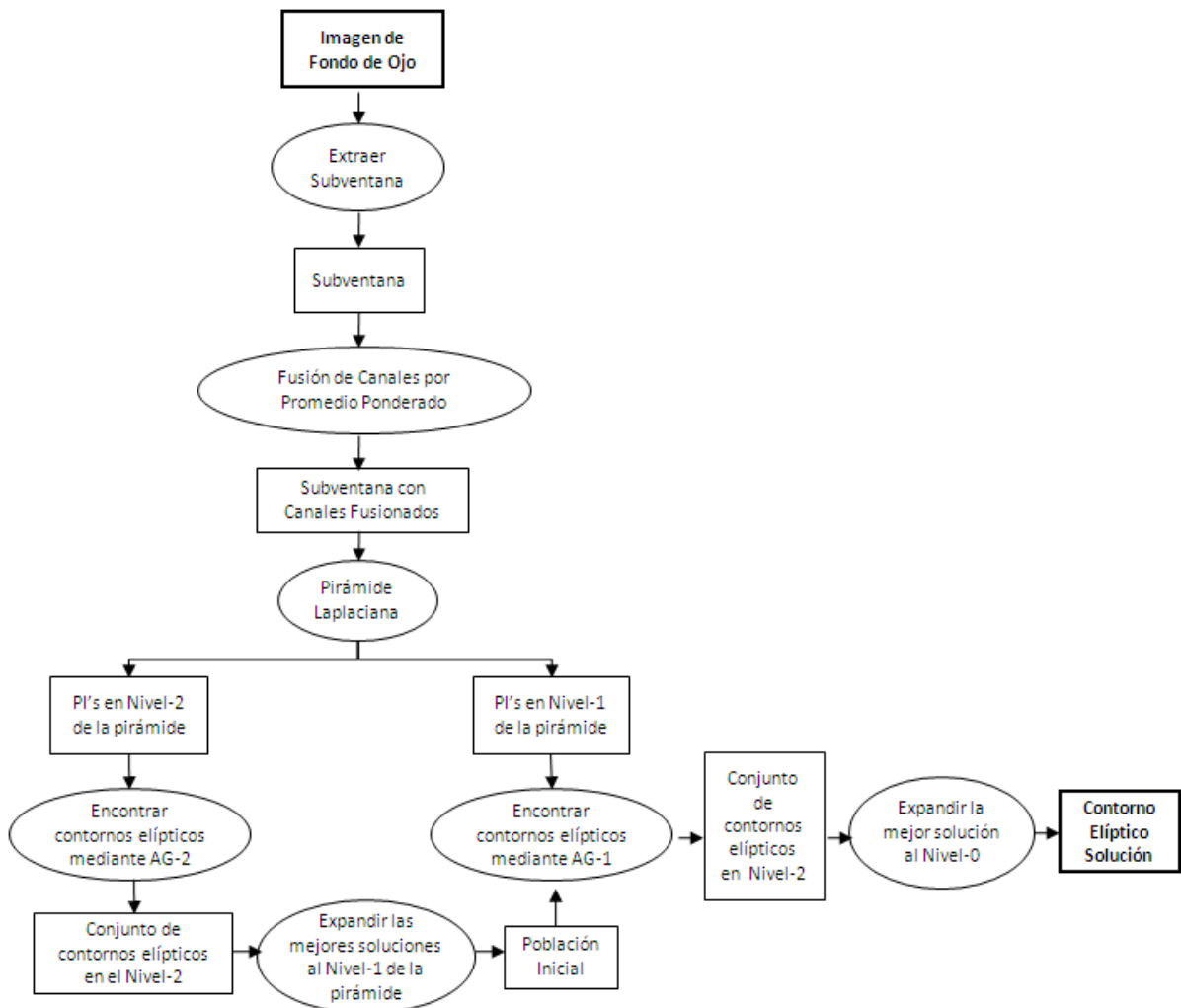


Figura 7 - Diagrama de Bloques correspondiente al método de la pirámide Laplaciana

3.2.1 Obtención de los Puntos de Interés

La característica de un PI está basada principalmente en la propiedad por la cual un punto de la imagen correspondiente al contorno papilar, presenta un cambio en su nivel de

intensidad en relación a sus píxeles vecinos. No obstante, esta propiedad representa una condición necesaria pero no suficiente, ya que puede haber otros cambios de intensidad debido a la existencia, por ejemplo, de otro tipo de estructuras anatómicas oculares (vasos capilares) o a la existencia de ruido en la imagen (la retina no tiene un color totalmente uniforme). Por otro lado, dado que cada imagen está compuesta por 3 canales, se plantea el problema de cómo manejar la información procedente de las tres fuentes. En este caso, se propone realizar una fusión ponderada y fija de los tres canales RGB (80% canal Rojo, 10% canal Verde y 10% canal Azul) para obtener una imagen de un solo canal. Luego, como consecuencia de aplicar el operador Laplaciano, se obtienen los puntos de interés en el Nivel-2 y Nivel-1 de la pirámide, el Nivel-0 no se utiliza porque la cantidad de Pí's ruidosos que aparecen en este nivel no facilita la convergencia de la solución del AG.

El proceso concreto de obtención del conjunto de Pí's es el siguiente: a partir de subventana extraída de la imagen original, se genera la pirámide Laplaciana obteniendo 3 niveles de imágenes con sus correspondientes conjuntos de puntos (niveles 0, 1 y 2). A continuación, se realiza la binarización de la imagen Laplaciana obtenida en los niveles 1 y 2 con el objeto de quedarnos con los puntos de más intensidad. Concretamente, en el caso de la imagen del nivel-2 se escogen los 200 puntos más brillantes. Se actúa de igual forma en el Nivel-1 de la pirámide Laplaciana, obteniendo un segundo conjunto de puntos Pí's en este nivel. En la Figura 8, se pueden observar ejemplos de Pí's obtenidos en los dos niveles.

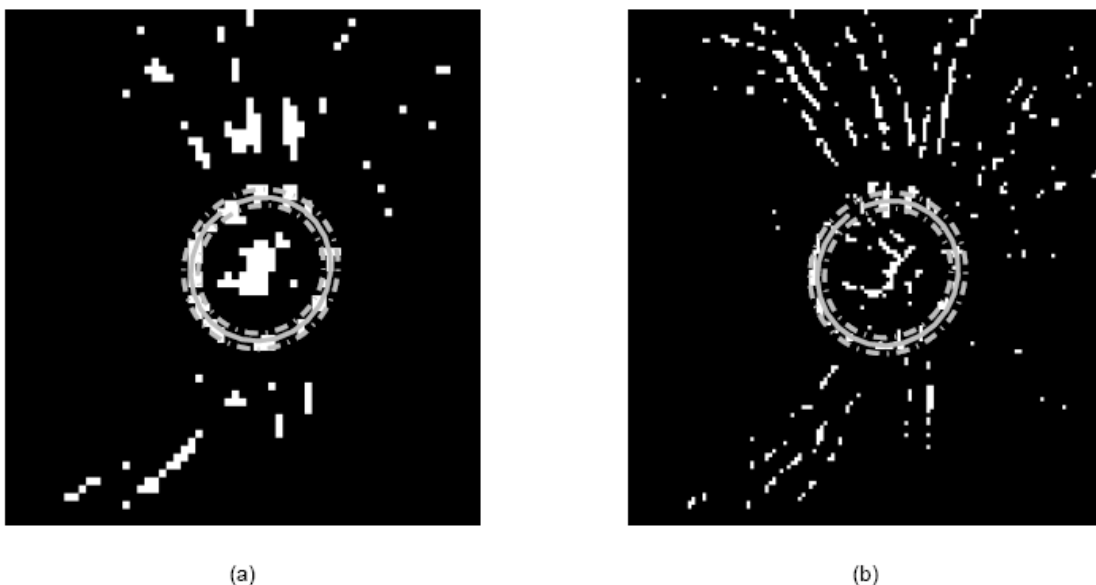


Figura 8 : Puntos obtenidos en el Nivel-2 (a) y en el Nivel-1 (b) de la pirámide Laplaciana. La línea continua representa el contorno papilar solución y las líneas discontinuas representa la corona elíptica asociada.

3.2.2 Búsqueda del contorno papilar basado en un AG de dos fases

Los Algoritmos Genéticos son uno de los paradigmas más frecuentemente utilizados en computación evolutiva y deben su inspiración al proceso biológico de la evolución: selección natural y supervivencia de los individuos mejor adaptados [15]. De esta forma, dado un problema de un dominio específico, estos algoritmos codifican potenciales soluciones utilizando una estructura de datos que simula un cromosoma (individuo) donde los genes son los parámetros del problema propuesto. La aproximación a la solución de cada cromosoma es calculado a partir de una función de evaluación o función fitness. Finalmente, tras aplicar operadores de selección y de variación (cruce y mutación), la población de soluciones va evolucionando hacia una mejor solución en cada generación hasta que, finalmente, la solución es alcanzada o el proceso de evolución finaliza por alcanzar un determinado criterio de parada. En este último caso, se utilizaría la mejor solución obtenida en la última generación.

Teniendo en cuenta que el contorno papilar humano tiene forma ovalada, este trabajo propone la búsqueda de la solución aproximando el contorno mediante una elipse, como se ha hecho en otros trabajos [1], [2] y [16]. A partir de este momento se asume que: (1) la forma geométrica de la papila puede ser aproximada mediante una elipse no deformable, y (2) a mayor número de PI's en la corona elíptica, mayor probabilidad de que dicha elipse asociada se acerque al contorno real de la papila. Los resultados obtenidos en [1] y en el presente trabajo apoyan la validez de estas dos hipótesis.

En MPL se utiliza un Algoritmo Genético en dos fases: AG-2 y AG-1 (ver Figura 7). El denominado AG-2 se ejecuta utilizando los PI's obtenidos en el Nivel-2 de la pirámide para encontrar la elipse con el mayor número de PI's en su corona elíptica asociada. El conjunto de contornos papilares obtenidos será una primera aproximación a nuestro problema. Ahora, expandimos los mejores contornos del Nivel-2 al Nivel-1, utilizando dichos contornos como parte de la población inicial para el AG-1. Concretamente, se utiliza el 50% de las mejores soluciones obtenidas en el paso anterior y el resto de individuos necesarios para completar la población son creados de forma aleatoria. Luego, se ejecuta el AG-1 utilizando los PI's obtenidos en el Nivel-1 de la pirámide Laplaciana y con el mismo propósito que se hizo al aplicar AG-2. Finalmente, la mejor elipse papilar de la población final obtenida mediante AG-1 es expandida desde el Nivel-1 al Nivel-0 (subventana original). Esta elipse será el contorno papilar solución proporcionado por este método.

3.2.2.1 Caracterización del AG de dos fases

El espacio fenotípico solución del problema original está formado por el espacio de coronas elípticas resultantes de extender una anchura ϵ el perímetro de las infinitas elipses que pueden ser trazadas en la imagen. Para codificar este tipo de soluciones, el espacio fenotípico es transformado en el espacio genotípico mediante vectores reales de 5 variables $[x, y, a, b, \omega]$. Siendo, (x,y) el centro de la elipse, (a,b) las magnitudes de su semieje mayor y menor, respectivamente, y ω el ángulo que el eje mayor forma respecto al eje-x.

La función fitness es definida de forma simple: cuenta el número de PIs contenidos en una corona concéntrica a la potencial elipse solución. En la Figura 8 podemos ver elipses solución en varios niveles y sus coronas elípticas asociadas.

Teniendo en cuenta que las dimensiones de la subventana de entrada son finitas y conocidas, para facilitar la búsqueda de la solución, se acota el tamaño del espacio de búsqueda de cada una de las variables que intervienen en la solución. La Tabla 1 resume el dominio de definición de cada gen utilizado en las ejecuciones de AG-2 y AG-1. Estos valores han sido obtenidos de [1], pero han sido adaptados proporcionalmente a las resoluciones empleadas en el Nivel-2 y Nivel-1 de la pirámide Laplaciana.

Tabla 1 : Dominio de definición (expresado en píxeles) para cada uno de los genes (variables) de un cromosoma (elipse genética)

	AG-2	AG-1
Tamaño de Imagen	65x65	129x129
$[x_{\min}, x_{\max}]$	[15,50]	[30,100]
$[y_{\min}, y_{\max}]$	[15,50]	[30,100]
$[a_{\min}, a_{\max}]$	[8,15]	[16,30]
$[b_{\min}, b_{\max}]$	[8,15]	[16,30]
$[\omega_{\min}, \omega_{\max}]$	[15°, 175°]	[15°, 175°]

Finalmente, la Tabla 2 resume los valores de los parámetros de configuración empleados para AG-2 y AG-1:

Tabla 2 : Resumen de la configuración de los dos AG

Parámetros de AG	AG-2	AG-1
Inicialización	Aleatoria	Aleatoria y las mejores soluciones de AG-2
Representación	Vector real (dimensión: 1 X 5)	
Cruce	Cruce en un punto	
Mutación	No uniforme con distribución Gaussiana	
Selección de Padres	Uniforme estocástico	
Selección de supervivientes	Generacional sin elitismo	Generacional con elitismo

3.3 Método basado en fusión por wavelet y filtrado (MFWF)

En el método anterior (MPL), y con la intención de reducir el número de canales de imagen, se realizaba una fusión ponderada de los tres canales RGB para obtener una imagen con un único canal. Normalmente, el canal Rojo ha demostrado ser el más útil ya que es el que proporciona mayor información en la búsqueda de los PI's. No obstante, en determinadas imágenes, los otros dos canales podrían aportar más información que la que aporta el canal rojo pero que, sin embargo, no podría utilizarse si adoptamos un esquema fijo de promediado de canales ponderado y sesgado hacia el rojo. Con el objeto de solucionar este inconveniente, es decir, de hacer la fusión de los 3 canales independiente del valor de los coeficientes de ponderación, se aplica una fusión de los canales RGB basada en la técnica de fusión de imágenes por Wavelets [17].

Ahora, con el objetivo de encontrar el conjunto de PI's, se aplican por separado dos filtros de imagen al canal resultado de la fusión, los cuales son aplicados de forma independiente sobre la imagen obteniendo dos conjuntos de PI's. Tras escoger los PI's comunes de los dos conjuntos, se aplica un AG de una sola fase, con el mismo objetivo que perseguían tanto AG-1 como AG-2 en el método anterior, es decir, el de encontrar la elipse que presente un mayor número de PI's en una corona de su perímetro. En la Figura 9, podemos ver un diagrama de bloques resumen de este segundo método propuesto.

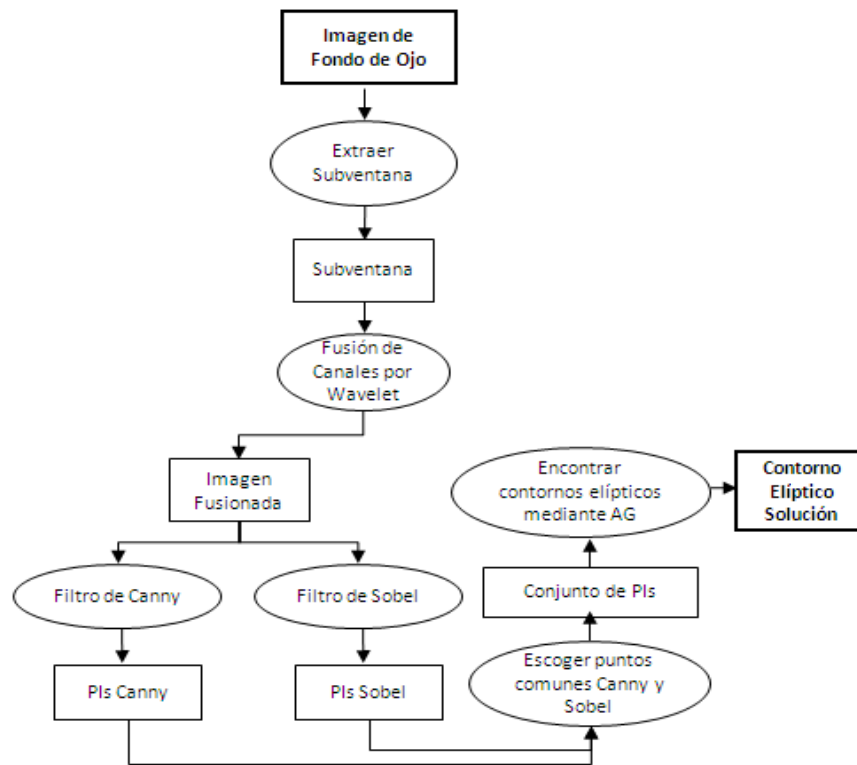


Figura 9 : Diagrama de Bloques correspondiente al método basado en fusión por wavelet y filtrado

3.3.1 Obtención de los Puntos de Interés

El aplicar transformaciones matemáticas adecuadas a una señal permite obtener de ella más información que aquella que se puede extraer de la señal pura. Entre un gran número de transformaciones existentes, sin lugar a dudas, la más conocida es la Transformada de Fourier. Esta transformada permite descomponer una señal en sus componentes sinusoidales de diferentes frecuencias, es decir, permite transformar una señal desde la base de tiempo a la base de la frecuencia. Existe un tipo especial de transformada de Fourier, las cuales están siendo cada vez más empleadas en un amplio campo de especialidades, a menudo sustituyendo a la transformada de Fourier original, es la denominada transformada Wavelet.

La transformada Wavelet, originalmente desarrollada en los años 80, es una herramienta de análisis de señales que proporciona una descomposición multiresolución de una imagen. Entre las aplicaciones de esta técnica podemos encontrar el tratamiento de imágenes, la reducción del ruido, estudio de fractales, la compresión o la detección de determinados objetos en ciertos tipos de imágenes.

En concreto, la transformada Wavelet consiste en comparar la señal con ciertas funciones Wavelet, las cuales se obtienen a partir de la Wavelet madre. La comparación permite obtener unos coeficientes que son susceptibles de interpretación y posterior manipulación. Una forma sencilla de comprender el funcionamiento de la transformada Wavelet es pensar que la señal en base de tiempo es pasada por varios filtros paso-altas y paso-bajas, los cuales permiten separar las porciones de la señal de alta frecuencia de aquellas de baja frecuencia. Como resultado se obtiene una descomposición de la señal original en versiones trasladadas (en tiempo) y escaladas de la wavelet original (wavelet madre). Existe una importante cantidad de familias de funciones Wavelets madre que han probado ser especialmente útiles, entre ellas destacan la Haar, Biortogonal, Coiflets, Morlet y Meyer, entre otras.

Las transformadas Wavelet puede ser discreta (TWD) o continua (TWC). La TWD se utiliza comúnmente en ingeniería y ciencias para la codificación de señales, mientras la TWC se utiliza en investigación científica para el análisis de señales. En este trabajo, nos centraremos en el primer tipo.

Lo que se pretende aquí es utilizar la TWD para fusionar los tres canales RGB [17]. La idea es mejorar el contraste de la imagen, aprovechando la información relevante de cada canal. La TWD aplicada a imágenes proporciona una matriz de coeficientes, conocidos como coeficientes Wavelet. Si a una imagen le aplicamos la TWD obtenemos los *coeficientes de aproximación* (obtenidos de la salida del filtro paso-bajas) y los *coeficientes de detalle* (obtenidos de la salida del filtro paso-altas). Los *coeficientes de aproximación* contiene la mayor parte de la energía de la imagen, es decir, la información más importante, mientras que los *coeficientes de detalles* tienen valores próximos a cero. Los valores más altos en los coeficientes Wavelet en las bandas altas corresponden a las características de la imagen, como pudiera ser los bordes de la imagen. En este trabajo se utiliza esta característica para realzar la imagen por medio de una fusión de canales. En la Figura 10 podemos ver un esquema de la fusión mediante Wavelets.

En cada fusión solamente se puede mezclar dos señales, por tanto, para mezclar los 3 canales de imagen, deberemos realizar la fusión en dos pasos: por un lado mezclar los canales verde y azul, para posteriormente mezclar este resultado parcial con el canal rojo. El proceso detallado de mezcla es el siguiente:

1. Se transforma la imagen correspondiente al canal verde en una representación Wavelet, según la señal Wavelet madre y configuración escogida.
2. Se transforma la imagen correspondiente al canal azul en una representación Wavelet, según la señal Wavelet madre y configuración escogida.
3. Se mezclan las dos imágenes transformadas anteriores, escogiendo los valores máximos de los *coeficientes de aproximación* y *coeficientes de detalle*. El objetivo es preservar las características más destacadas en las altas frecuencias (*detalles*) y las bajas frecuencias (*aproximaciones*).
4. Se calcula la transformada Wavelet inversa de la imagen resultante del paso 3. Esto produce como resultado la imagen fusionada de los canales verde y azul.
5. Se transforma la imagen fusionada de los canales verde-azul en una representación Wavelet según la señal Wavelet madre y configuración escogida.
6. Se transforma la imagen correspondiente al canal rojo en una representación Wavelet según la señal Wavelet madre y configuración escogida.
7. Se mezclan las dos imágenes transformadas, escogiendo los valores máximos de los *coeficientes de aproximación* y *coeficientes de detalle*.
8. Se calcula la transformada Wavelet inversa de la imagen resultante del paso 7. Se obtiene, finalmente, la imagen fusionada de los canales verde, rojo y azul.

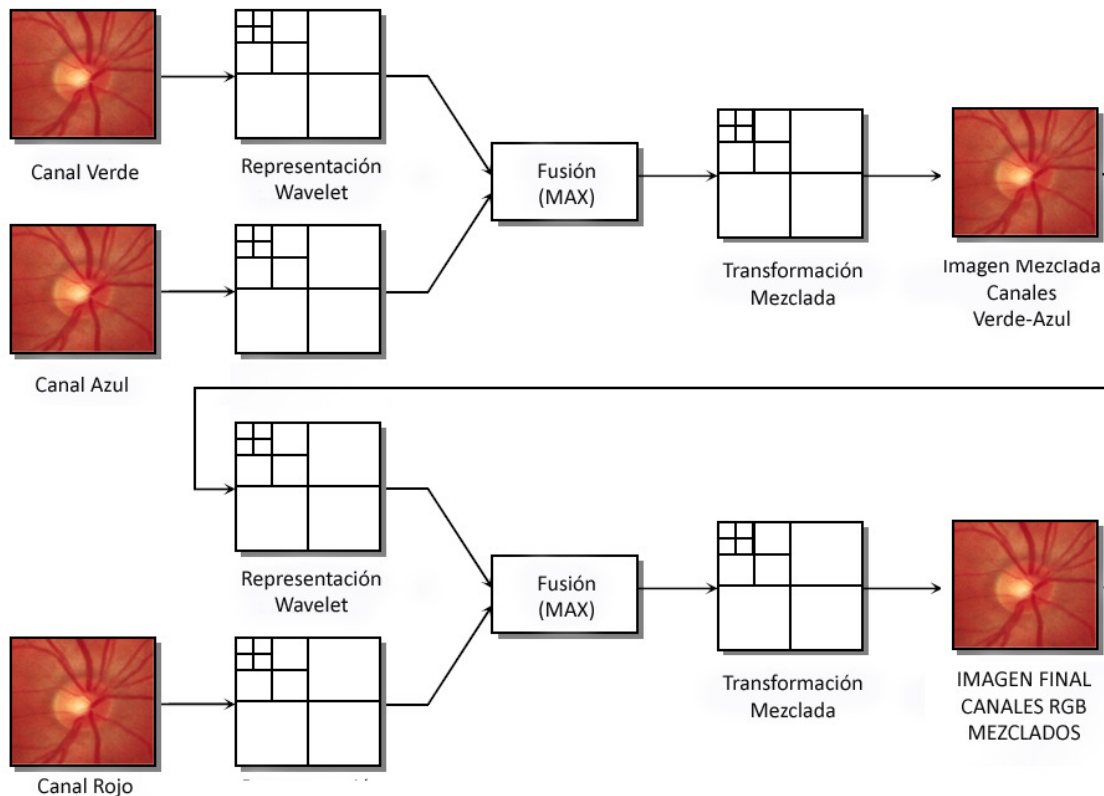


Figura 10 - Proceso de Fusión de los canales RGB mediante TWD. Inicialmente se fusionan los canales Verde y Azul, el resultado es fusionado con el canal Rojo

Tras fusionar los tres canales, se procede a aplicar los filtros de imagen de Canny y Sobel, de forma independiente, sobre el canal fusionado y obtener así dos conjuntos de PI's. Concretamente, el filtro de Canny es un algoritmo de detección de contornos que, al ser aplicado a la imagen, produce un máximo local donde existe un contorno, eliminando el efecto ruido. Por otro lado, el filtro de Sobel es un operador diferencial discreto que calcula una aproximación al gradiente de la función de intensidad de una imagen. Para cada punto de la imagen a procesar, el resultado del operador Sobel es tanto el vector gradiente correspondiente como la norma de éste vector.

El siguiente paso consiste en seleccionar, como conjunto final de PI's, los puntos comunes de ambos conjuntos. En la Figura 11 podemos ver un ejemplo del resultado de aplicar ambos filtros a la subventana de la imagen fusionada, los PI's obtenidos en cada caso y el conjunto final de PI's comunes a ambos filtros.

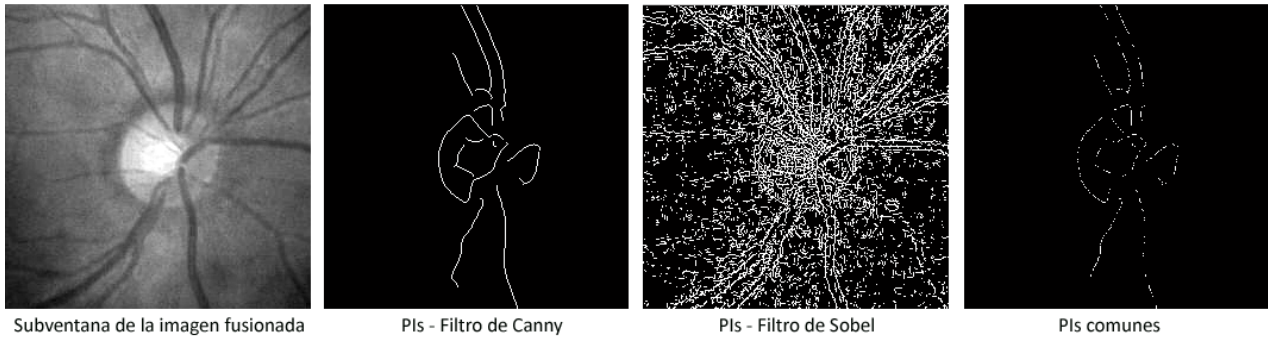


Figura 11 - Obtención de los PI's tras aplicar los filtros de Canny y Sobel sobre la imagen fusionada

Por último, es importante señalar que se trabaja directamente con el Nivel-0, es decir, a diferencia del método MPL, en este método trabajamos siempre en el nivel 0.

3.3.2 Búsqueda del contorno papilar basado en un AG de una fase

El AG utilizado en este método utiliza la misma estrategia que los dos AGs utilizados en el método basado en pirámide Laplaciana. Es decir, a partir del conjunto final de PI's obtenidos en el paso anterior, el AG busca aquella elipse que contenga en una corona de su perímetro el máximo número de PI's. La elipse que maximice este criterio corresponderá al contorno papilar solución. Puesto que ahora el conjunto de PI's se obtuvieron sobre el tamaño de subventana original, la búsqueda de las elipses papilares también se hace sobre el nivel 0. Es decir, a diferencia de lo que ocurría en el MPL, no es necesario expandir la elipse solución para obtener el contorno papilar con la resolución adecuada. La otra diferencia con MPL radica en que ahora la ejecución del AG se hace en una única fase, y no en dos como allí se hacía. La configuración de parámetros del AG aquí utilizado es la misma que la que se utilizó para AG-2 del método anterior (ver Tabla 2).

Capítulo 4 . Casos de estudio

Para medir la efectividad de los dos métodos propuestos, se utiliza la base de datos DRIONS [14]. Con el objetivo de hacer que los resultados de la evaluación sean reproducibles cuantitativamente, se mide la discrepancia media δ (concepto definido en [1] y [2]) entre los puntos del contorno obtenidos en la segmentación y la media de dos contornos trazados por expertos (Gold Standard).

De forma detallada, el cálculo de la discrepancia es el siguiente:

1. Partiendo del centro de la elipse solución obtenida por uno de los métodos, se traza un haz de N segmentos equiespaciados, o segmentos radiales (SR). En concreto trazamos 36 SR's (N=36).
2. Se calcula la longitud del i-ésimo radio desde el centro de la elipse hasta el punto de corte de la elipse solución sobre cada uno de los SR. Esta medida se denota como m_i^j (longitud del i-ésimo radio correspondiente a la imagen j).
3. Se calcula μ_i^j y σ_i^j , siendo la media y desviación típica, respectivamente, de las longitudes de los radios del punto i-ésimo, medidas desde el centro de la elipse al punto de corte de los contornos trazados por los expertos y cada SR (pertenecientes a la imagen j).
4. Se tiene en cuenta un valor $\varepsilon = 0.5$, para evitar la división por cero cuando la medición de los expertos es exactamente igual para un punto determinado.

La siguiente ecuación detalla el cálculo de la discrepancia δ para la imagen j:

$$\delta_j = \frac{1}{N} \cdot \frac{\sum_{i=1}^N |m_i^j - \mu_i^j|}{\sigma_i^j + \varepsilon}$$

Finalmente, para visualizar los resultados de discrepancia obtenidos en toda la base de imágenes de forma más compacta, se traza la ojiva de discrepancia en una gráfica, donde el eje-y representa el porcentaje de imágenes por debajo de cierta discrepancia y el eje-x los diferentes valores de discrepancia. A continuación, se muestran los casos de estudio para cada uno de los dos métodos propuestos.

4.1 Método basado en Pirámide Laplaciana (MPL)

4.1.1 Resultados experimentales

Diferentes experimentos fueron hechos a fin de elegir los mejores valores relativos a los parámetros intrínsecos de nuestro AG de dos fases: tamaño de la población (P), número de generaciones (G), tamaño de la población elite (E), probabilidad de cruce (C), y el operador de corte *shrink* (S). Este último parámetro controla el ratio por el cual la magnitud de mutación media decrece y es relativo al operador de mutación no uniforme con distribución Gaussiana aplicado con una probabilidad de uno por gen.

A la luz de los resultados obtenidos en los experimentos, se escogió la configuración final de los parámetros: $P_2=200$, $G_2=20$, $C_2=0.2$, $S_2=0.8$, $E_2=0$ (Para AG-2), y $P_1=100$, $G_1=50$, $C_1=0.2$, $S_1=0.8$, $E_1=1$ (para AG-1). La población inicial del AG-1 fue creada utilizando el 50% de las mejores soluciones obtenidas con AG-2. El resto de la población inicial de AG-1 fue obtenida aleatoriamente. A fin de alcanzar un resultado competitivo, buscamos un compromiso entre calidad de resultados y coste computacional.

En cuanto a la configuración para la obtención de los PIs se escogieron los 200 puntos más brillantes en los Niveles 1 y 2 de la Pirámide Laplaciana. El ancho de corona escogido fue de (-0.5, -3) píxeles.

4.1.2 Evaluación y comparación de resultados

En la bibliografía consultada relacionada con la segmentación de CNO partiendo de imágenes fondo de ojo en color y descartando las evaluaciones hechas de forma subjetiva y no cuantitativa, los métodos de [1] y [2] han aportado evidencia de ser competitivos. Por tanto, se usarán estas aproximaciones como referencia. La Figura 12 muestra las ojivas de discrepancia del resultado de aplicar los tres métodos a la base de datos DRIONS [14]. Específicamente, las ojivas de los métodos de [1] y [2] se han obtenido directamente de [1]. Por otra parte, debido a la naturaleza estocástica inherente de un AG, la curva de discrepancia obtenida por el método MPL corresponde al resultado de realizar la media de cinco ejecuciones del AG.

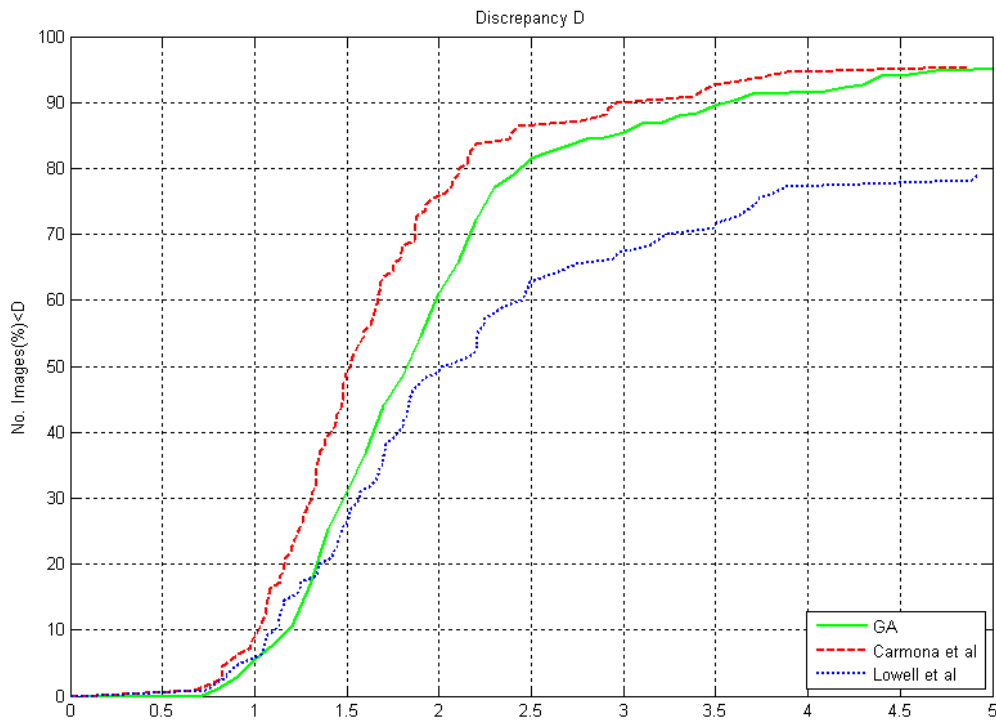


Figura 12 : Comparativa de la discrepancia acumulada para los métodos de Carmona, Lowell y MPL

Una comparación directa de la curva de discrepancia obtenida por el método de descrito en [2] y MPL revela dos áreas muy diferentes con un comportamiento opuesto, cuyo límite está marcado por el valor de discrepancia $\delta=1.4$. Por debajo de este valor, los resultados del método de [2] son ligeramente mejores que los obtenidos por MPL, y por encima de este valor ocurre precisamente lo contrario. La explicación para los resultados obtenidos en el primer subintervalo de discrepancia (0, 1.4), puede deberse a que la fase de deformación local utilizada por el método descrito en [2] hace posible una ligera deformación de la elipse, obteniendo una mejor aproximación a los trazados reales de los expertos. Sin embargo, el cambio de tendencia en el segundo subintervalo (1.4 , 5) revela la robustez del método MPL. De hecho, mientras que en [2] solo obtiene el 80% de imágenes por debajo de discrepancia 5, el método MPL obtiene un 95%.

La comparación de la curva de discrepancia obtenida por el método de [1] revela que el método MPL tiene un comportamiento ligeramente peor en todo el rango de discrepancias. La explicación de este comportamiento podría ser debido al hecho de que el contorno genético del método MPL obtenido en el Nivel-1 de la Pirámide Laplaciana es

expandido al nivel de la imagen original (Nivel-0). Este hecho podría explicar también el porqué la diferencia entre las curvas es más pronunciada para valores bajos de discrepancia y menos pronunciada para valores altos. De hecho, para un valor de discrepancia igual a 5, ambos métodos igualan sus resultados. Por otra parte, el coste computacional de nuestro método es menor: 9.000 evaluaciones de la función fitness contra 20.000 evaluaciones utilizadas por el método usado en [1]. Finalmente la Figura 13 muestra diferentes ejemplos de contornos papilares obtenidos con este método, los cuales son comparados con el contorno papilar promedio trazada por los dos expertos (Gold Standard).

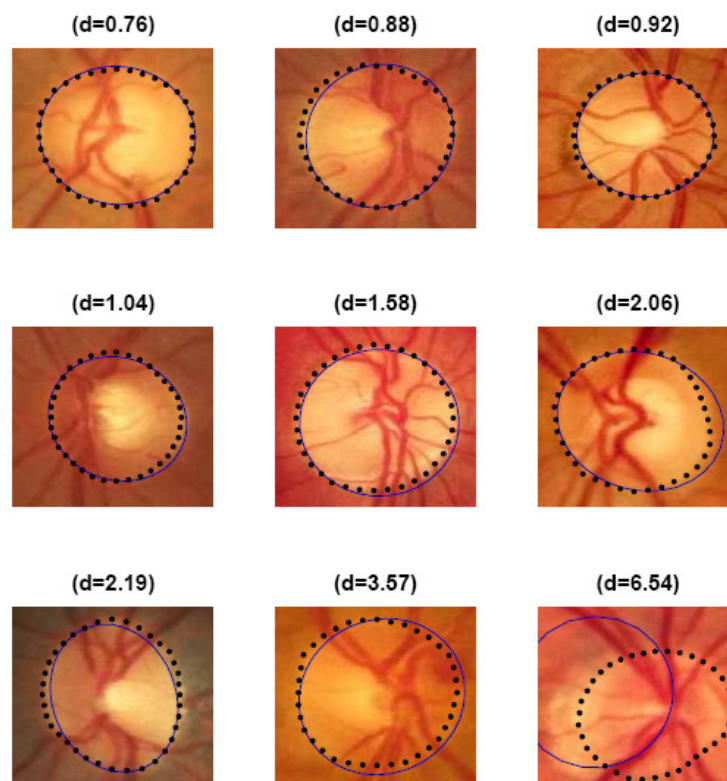


Figura 13 : Elipses Genéticas obtenidas con el método MPL (línea continua) comparadas con el Gold Standard (línea punteada). También se muestra el valor de discrepancia obtenido

4.2 Método basado en fusión por Wavelet y filtrado (MFWF)

4.2.1 Resultados experimentales

Al igual que para el método MPL, diferentes experimentos fueron hechos a fin de elegir los mejores valores relativos a los parámetros intrínsecos de nuestro AG de dos fases. A la luz de los resultados obtenidos en los experimentos, se escogió la configuración final de los parámetros: $P=400$, $G=50$, $C=0.2$, $S=0.8$, $E=0$. A fin de realizar una comparativa real con los métodos de Carmona y Lowell, el número de evaluaciones de la función fitness se ha ajustado a 20.000.

Tras los experimentos realizados, la configuración escogida para el método de fusión por Wavelet fue: Familia de Wavelet: Haar, número de escalas=3. La configuración escogida para el filtro de Canny fue de: umbral=0.4 y $\sigma=3$. La configuración escogida para el filtro de Sobel fue de 6 tanto para el umbral vertical como para el horizontal. Por último, la configuración escogida para el ancho de corona fue de (3,-3), un ancho mayor que para el método MPL puesto que ahora trabajamos directamente en el nivel de la imagen original (Nivel-0).

4.2.2 Evaluación y comparación de resultados

Al igual que en el método MPL, se usarán los métodos de [1] y [2] como referencia. La Figura 14 muestra las curvas de discrepancia de la aplicación de los tres métodos a la base de datos DRIONS [14]. Como se ha comentado anteriormente, las curvas de los métodos de [1] y [2] se han obtenido directamente de [1]. Nuevamente, la curva de discrepancia obtenida por el método MFWF, también corresponde al resultado de realizar la media de cinco ejecuciones del AG.

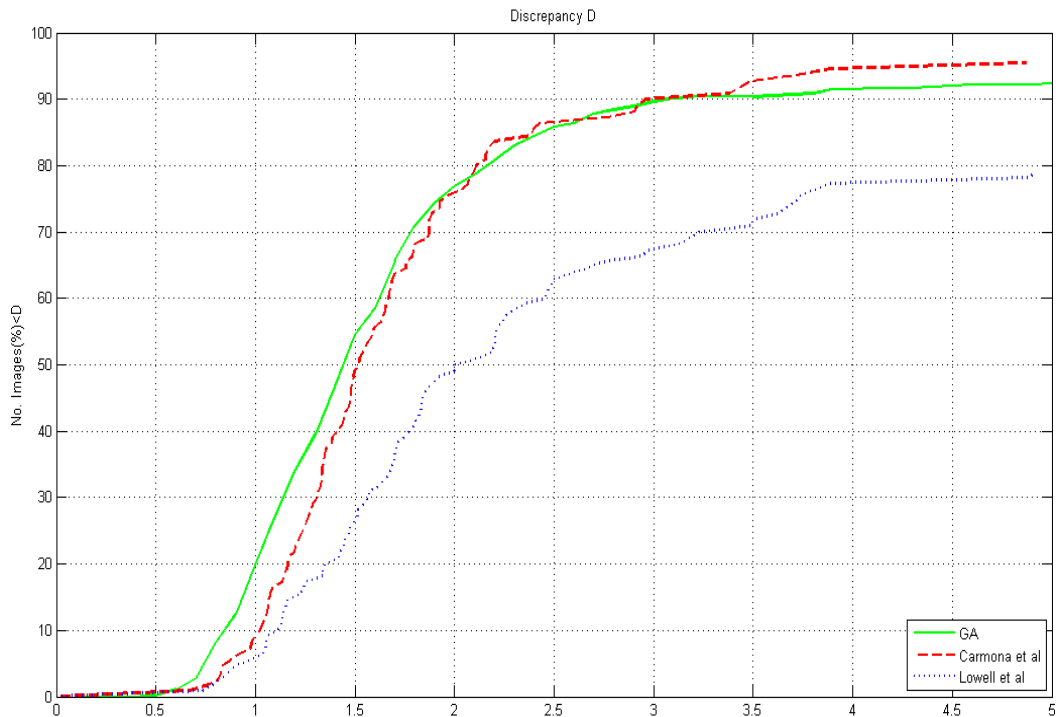


Figura 14 : Comparativa de la discrepancia acumulada para los métodos de Carmona, Lowell y MFWF

Una comparación directa de la curva de discrepancia obtenida por el método MFWF y la de [2] revela que el método propuesto sobrepasa con creces los resultados obtenidos por este último.

La comparación de la curva de discrepancia obtenida por [1] revela que el método MFWF tiene un mejor comportamiento en el rango de discrepancias menores, para valores altos de discrepancia los resultados se igualan.

Por tanto, el método MFWF es más exacto en la segmentación que el método de [1] para casi el 80% de las imágenes estudiadas. Esta exactitud es más notable para valores bajos de discrepancia: en [1] se obtiene un 7% de imágenes con discrepancia menor que 1, en el método MFWF se obtiene un 20% de imágenes con discrepancia menor que 1. Esta diferencia de porcentajes se va acortando ligeramente conforme va subiendo el valor de discrepancia.

La explicación de este mejor comportamiento podría ser debido al hecho de que, con en este método, trabajamos directamente en el Nivel-0 y no es necesario realizar la

expansión extra de la elipse genética que implica el método MPL. Para valores altos de discrepancia las curvas prácticamente se igualan, este hecho podría explicarse por la dificultad de segmentación para casi un 20% de las imágenes en estudio. Esta dificultad podría ser debida al ruido provocado por el color no uniforme de la retina o por la existencia masiva de vasos sanguíneos. Por otra parte, el coste computacional de nuestro método es igual al método de Carmona: 20.000 evaluaciones de la función fitness. Finalmente la Figura 15 muestra diferentes ejemplos de contornos papilares obtenidos con este método, los cuales son comparados con el Gold Standard.

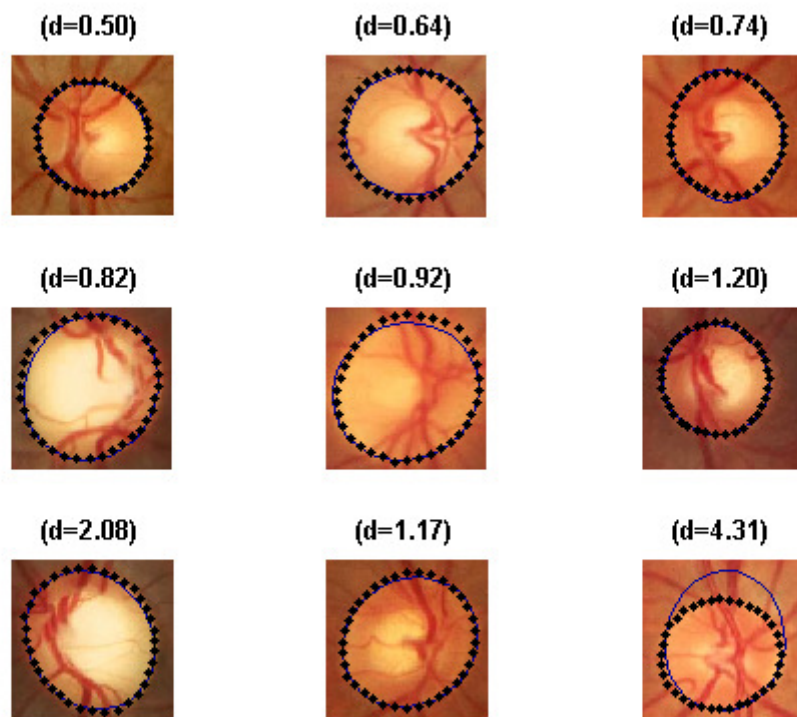


Figura 15 : Elipses Genéticas obtenidas con el método MFWF (línea continua) comparadas con el Gold Standard (línea punteada). También se muestra el valor de discrepancia obtenido

Capítulo 5. Conclusiones y trabajo futuro

Este Trabajo de Fin de Máster se ha centrado en la creación de dos métodos automáticos para la segmentación de la cabeza del nervio óptico en imágenes fotográficas de retina, el método MPL y el método MFWF. Ambos métodos utilizan una fase previa que consiste en la localización de la CNO.

Ambos métodos han sido probados con una base de datos de imágenes de fondo de ojo (DRIONS-DB)², descrita en [1]. Por otra parte, los resultados son comparados con los obtenidos en [1] y [2] . Los resultados obtenidos con el método de Pirámide Laplaciana son mejores que [2] y ligeramente inferiores a [1]. Sin embargo, comparado con [1], el proceso es simplificado notablemente reduciendo así el coste computacional. Finalmente, los resultados obtenidos con el método de Wavelets mejoran tanto los presentados en [1] como los obtenidos en [2].

Una desventaja del método MPL es que no podemos trabajar en el Nivel-0 de la Pirámide Laplaciana porque el conjunto de PI's obtenidos en este nivel contiene una gran cantidad de ruido. Por tanto, tenemos que expandir directamente la elipse solución obtenida desde el Nivel-1 hasta el Nivel-0, pero esta operación parece producir una degradación en la precisión de la solución. Esta desventaja es resuelta en el método MFWF ya que trabajamos directamente en el Nivel-0 de imagen. Los resultados corroboran esta mejora, obteniendo el doble de porcentaje de imágenes para discrepancias bajas en comparación a los resultados presentados en [1].

Como trabajo futuro, se propone seguir investigando métodos de realce de imagen y/o eliminación de ruido a fin de mejorar la exactitud de la segmentación. Por otra parte, se propone estudiar nuevos métodos de fusión de canales con el objetivo de aprovechar la información contenida en cada uno de los tres canales de color. Los resultados obtenidos con el método MFWF invitan a realizar aproximaciones híbridas o mejoras sobre este método a fin de seguir mejorando los resultados de discrepancia.

² <http://www.ia.uned.es/personal/ejcarmona/DRIONS-DB.html>

Apéndice A. Manual de Usuario de la aplicación

La interfaz gráfica de este Trabajo Fin de Máster está creada en Matlab. Para ejecutarla, habrá que ejecutar el script *tfm.m* localizado en el directorio del Trabajo Fin de Máster. La Figura 16 muestra la pantalla principal, mostrando la barra de menú, el tapiz central, los botones de zoom y la zona de paneles de opciones (donde se controlan las opciones de los algoritmos).



Figura 16 : Pantalla principal de la interfaz gráfica

La interfaz permite realizar las siguientes tareas:

- Cargar una imagen de fondo de ojo en el tapiz central, con la posibilidad de mostrar los contornos de los expertos y/o el contorno Gold Standard.
- Hacer zoom in/zoom out para cualquier imagen cargada en el tapiz central.
- Guardar la imagen mostrada actualmente en el tapiz central
- Cargar una base de datos de imágenes completa, con la posibilidad de mostrar los contornos de los expertos y/o el contorno Gold Standard. Por defecto se carga la base de datos DRIONS pero cabe la posibilidad de cargar otra base de datos.

- Obtener Pls de la imagen cargada en el tapiz central, tanto con el MPL como con el MFWF
- Localizar la papila de la imagen mostrada actualmente en el tapiz central.
- Segmentar la papila de una imagen de fondo de ojo, tanto con el MPL como con el MFWF.

Menú - Fichero

El menú Fichero permite cargar la imagen de fondo de ojo en el tapiz central, con la posibilidad de mostrar los contornos de los expertos y/o el contorno Gold Standard (Figura 17). Para ello pinchamos en Fichero y luego en la opción deseada. Aparecerá como directorio inicial la base de datos DRIONS, para seleccionar una imagen hacemos doble click sobre ella y la imagen aparecerá sobre el tapiz central (Figura 18). Por otra parte, para salir de la aplicación tendremos que pulsar en la opción Salir de este menú.



Figura 17 : Opciones del Menú Fichero

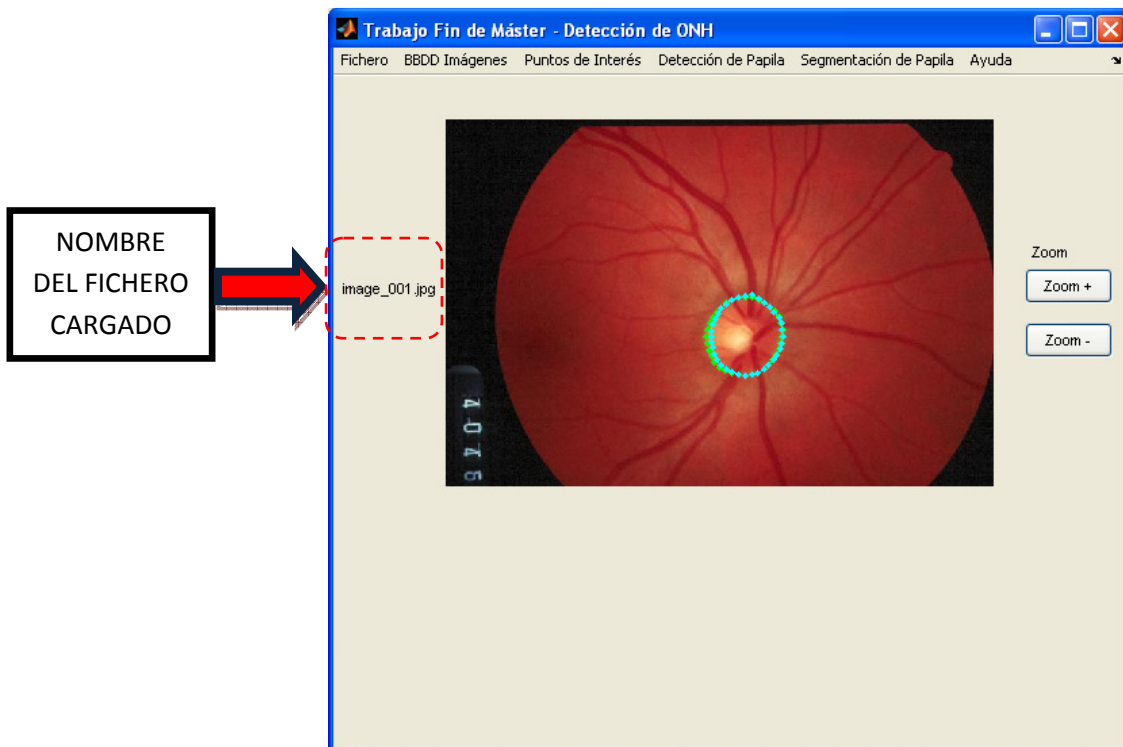


Figura 18 : Imagen cargada en el tapiz central

En cualquier momento podemos hacer zoom in/zoom out sobre la imagen cargada sobre el tapiz central (Figura 19). También podremos mover la imagen con total libertad para centrar alguna región de interés, para ello tendremos que situar el curso encima del tapiz central donde aparecerá la herramienta *mano*. Dejando pulsado el botón izquierdo del ratón podremos mover la imagen.

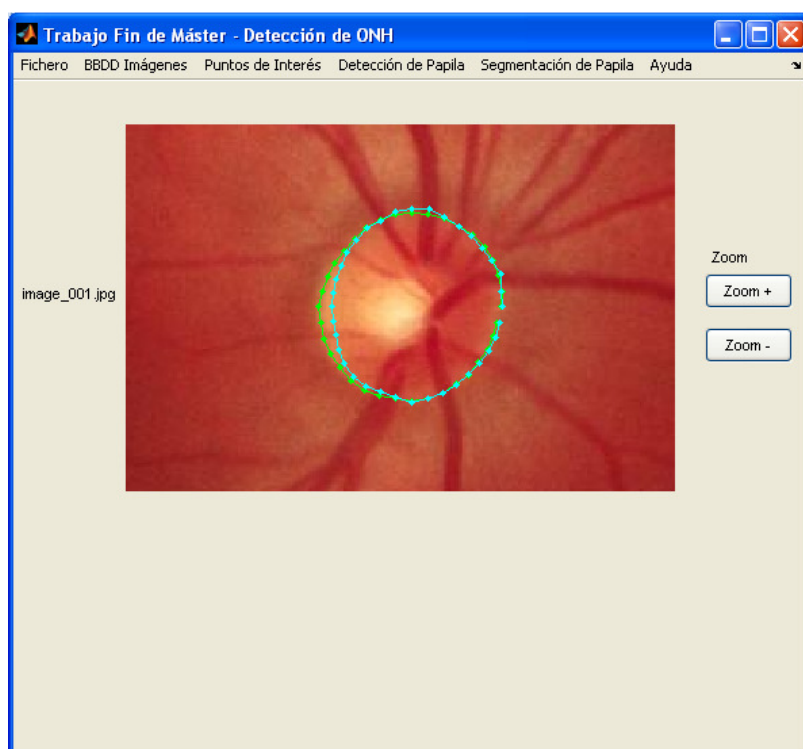


Figura 19 : Zoom sobre la imagen cargada sobre el tapiz central

En cualquier momento, desde cualquier opción de la aplicación podremos guardar la imagen cargada en el papel tapiz, para ello pulsaremos sobre la opción Guardar del menú Fichero, escogiendo un nombre para la imagen a guardar.

Menú - BBDD Imágenes

Esta opción nos permite cargar una la base de datos de imágenes de fondo de ojo para ser examinada sin contornos, con los contornos de los expertos y/o con el contorno promedio Gold Standard. La base de datos que se carga por defecto es la DRIONS [14], pero el programa permite cargar otra base de datos cualquiera. La Figura 20 muestra el panel de control de esta opción. La barra de desplazamiento permite movernos por el conjunto de imágenes cargado. En cualquier momento podremos activar/desactivar los contornos

pinchando en los checkbox del panel. Por otra parte, un pequeño panel nos muestra el número de imagen cargado actualmente y el número total de imágenes dentro de la carpeta seleccionada. Si deseamos cargar una nueva base de datos debemos pinchar en el botón *Examinar*, nos aparecerá el árbol de directorios y podremos escoger una carpeta con imágenes, una vez escogida deberemos pinchar en el botón *Cargar Imágenes*.

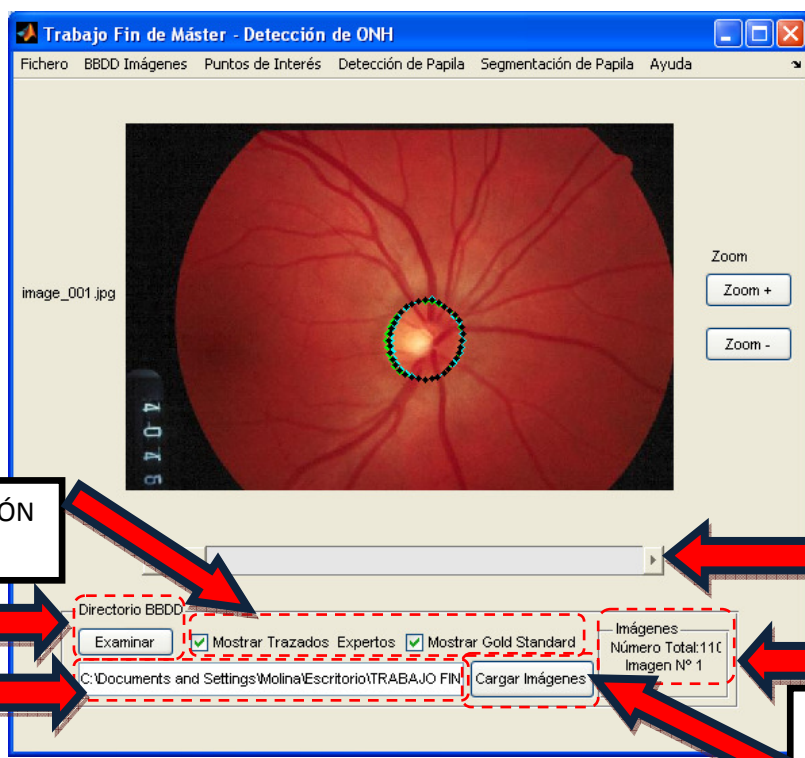


Figura 20 : Panel de control de la opción BBDD Imágenes

Menú - Puntos de Interés

Esta opción permite obtener PI's de la imagen cargada en el tapiz central, tanto con el MPL como con el MFWF. Al pinchar sobre esta opción podremos escoger el método a aplicar Figura 21.

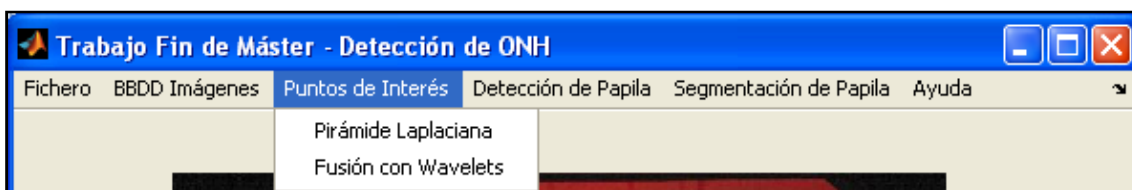


Figura 21 : Menú Puntos de Interés

Al seleccionar uno u otro método se activa o desactiva el botón de Cálculo de puntos. Podremos cambiar de método seleccionando los radiobotones de la parte superior del panel de control o escogiendo el método en el menú superior. Para el método MPL se podrán escoger los niveles de pirámide a calcular y el máximo número de puntos a obtener en cada uno de los niveles. Tras seleccionar la configuración deseada pulsaremos en el botón *Calcular PIs* y los puntos se mostrarán en el tapiz central, pudiendo pasar de un nivel a otro con la barra de desplazamiento . La Figura 23 muestra los puntos obtenidos en los niveles 0, 1 y 2 de la Pirámide Laplaciana.

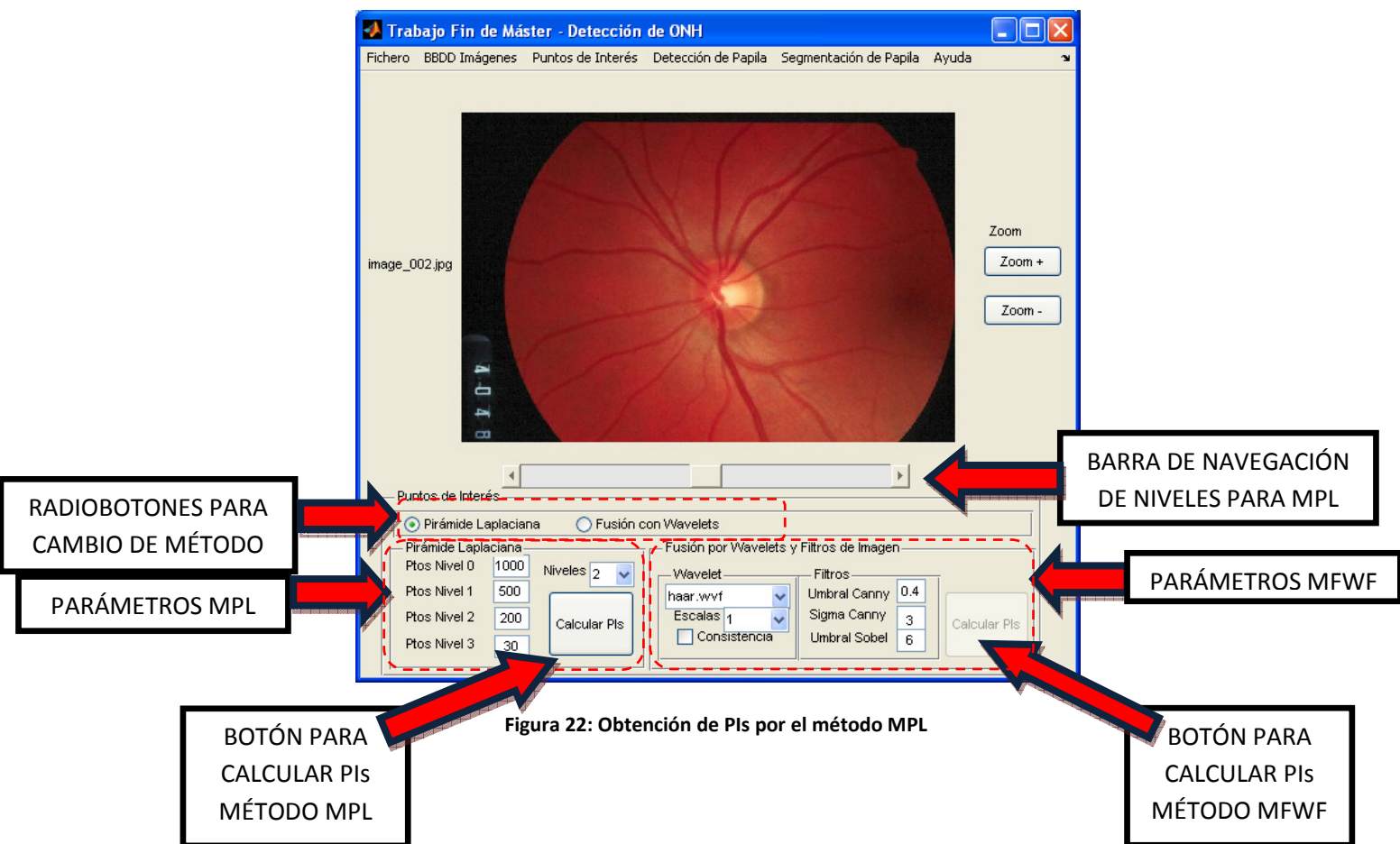


Figura 22: Obtención de PIs por el método MPL

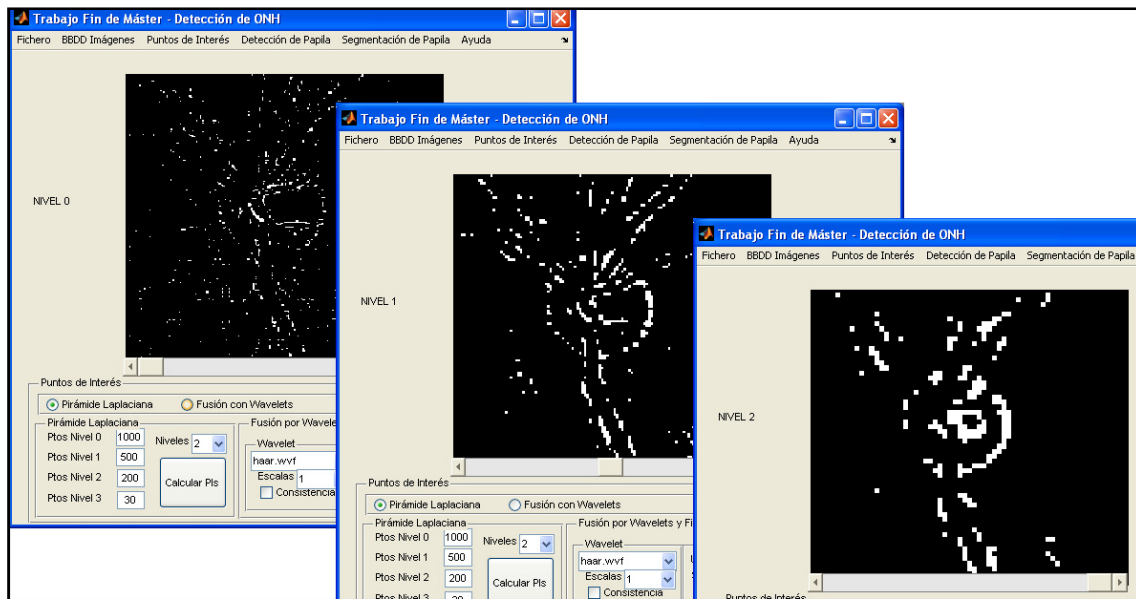


Figura 23 : PI's obtenidos en los Niveles 0, 1 y 2 de la Pirámide Laplaciana para una imagen de fondo de ojo determinada.

Para el método MFWF se podrán escoger los parámetros de la fusión Wavelet (fichero Wavelet, número de escalas y consistencia) y los parámetros para los filtros de imagen (umbral para Canny, factor Sigma para Canny y umbral para Sobel). Tras seleccionar la configuración deseada pulsaremos en el botón Calcular PI's y los puntos se mostrarán en el tapiz central. La Figura 24 muestra los puntos obtenidos en los niveles.

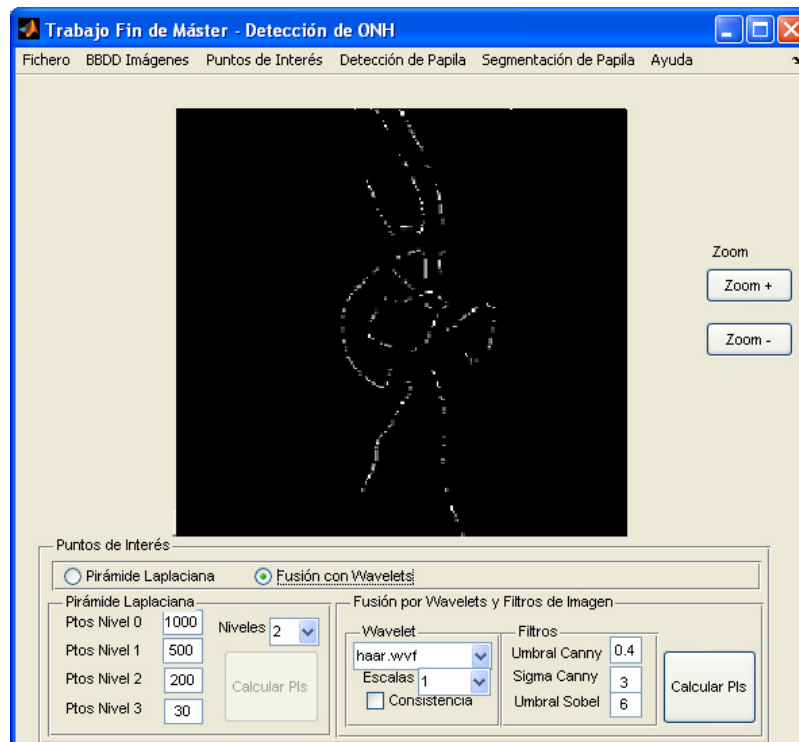


Figura 24 : PI's obtenidos con el método MFWF

Menú - Detección de Papila

Esta opción permite localizar la papila de la imagen mostrada actualmente en el tapiz central. Como podemos ver en la Figura 25, la papila es localizada rodeando la zona detectada con un círculo rojo.

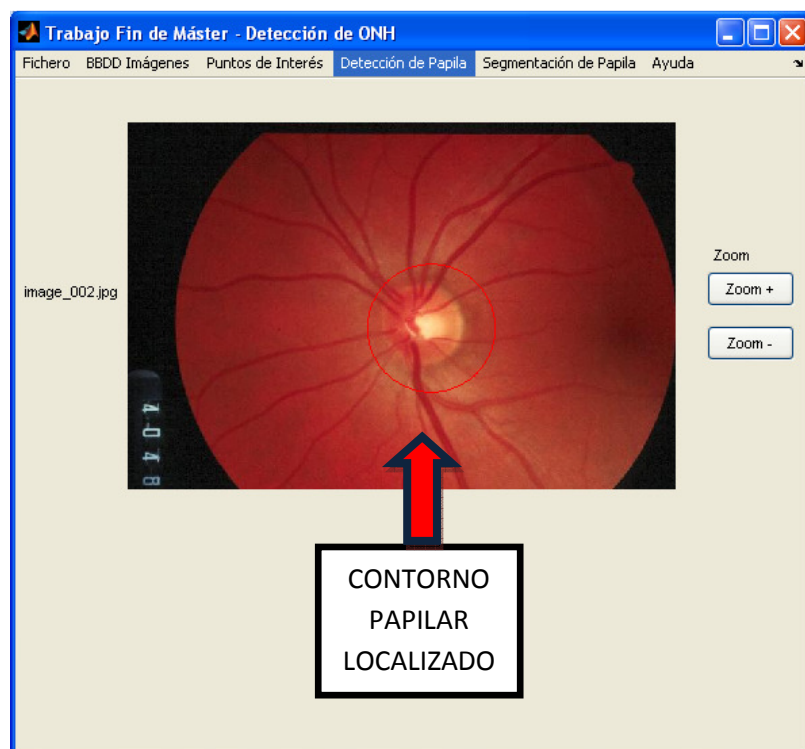


Figura 25 : Localización del Contorno Papilar

Menú - Segmentación de Papila

Este menú controla la segmentación la papila de una imagen de fondo de ojo, tanto con el MPL como con el MFWF. Inicialmente, al escoger uno de los dos métodos, el programa nos obliga a seleccionar una imagen de la base de datos. Una vez escogida la imagen deberemos elegir la configuración para cada uno de los métodos. En el caso de MPL: parámetros comunes de los AGs (probabilidad de cruce , probabilidad de mutación y tamaño de corona elíptica), los parámetros para el AG-1 (número de generaciones y población) y los parámetros para el AG-2 (número de generaciones y población). Finalmente, para segmentar la papila pulsaremos el botón *Segmentar Papila*. Inicialmente el proceso comenzará localizando la papila y extrayendo la subventana, posteriormente se podrá ir viendo el proceso

de segmentación conforme van evolucionando los AGs. Finalmente aparecerá una ventana emergente informando del resultado de la segmentación , adicionalmente aparecerá el tiempo y discrepancia en un subpanel de la parte inferior de la pantalla (como podemos ver en la Figura 26)

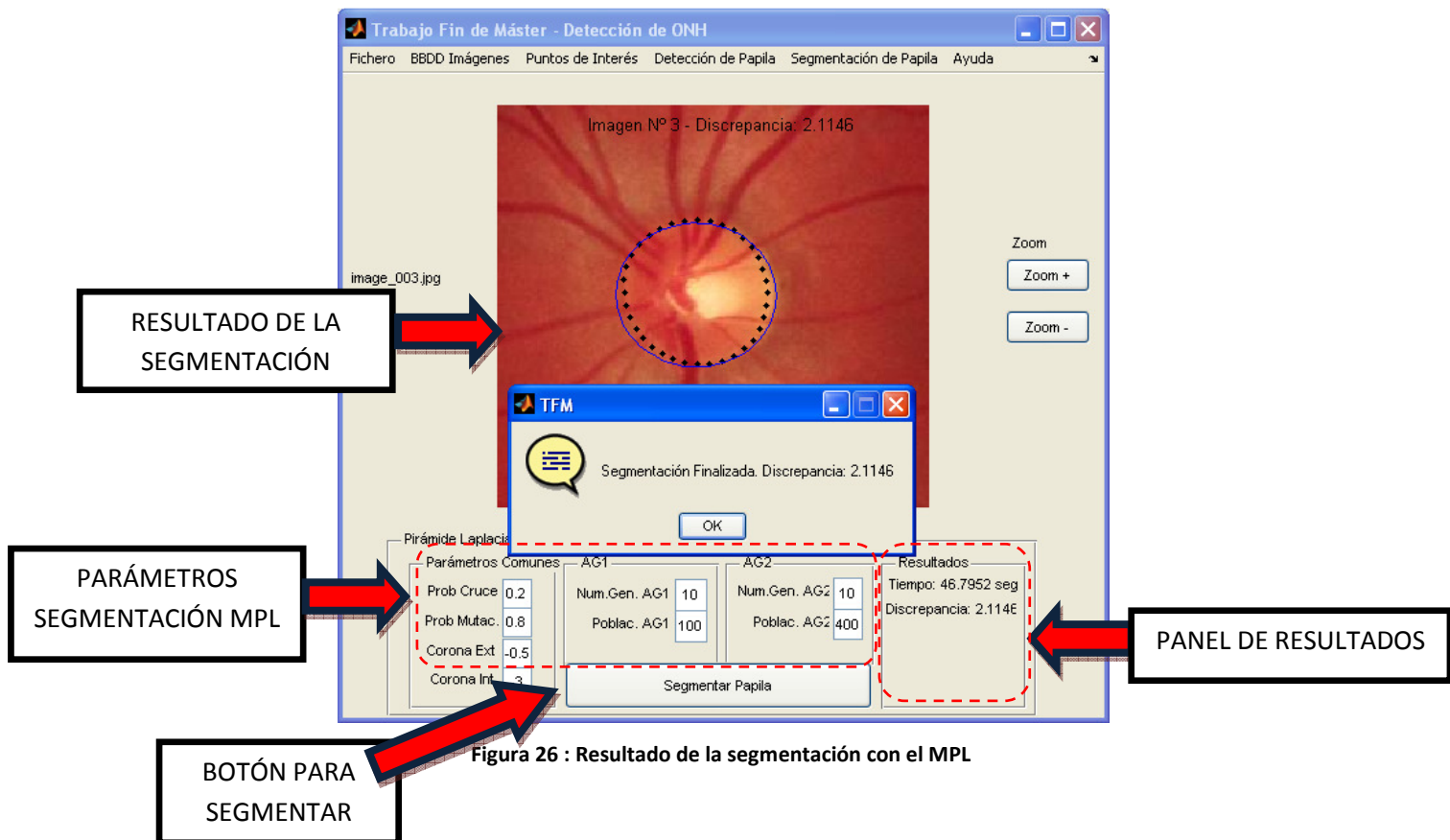


Figura 26 : Resultado de la segmentación con el MPL

En el caso de MFWF los parámetros a escoger son los propios del AG: número de generaciones y población, probabilidad de cruce , probabilidad de mutación y tamaño de corona elíptica ,número de generaciones y población. Finalmente, para segmentar la papila actuaremos de igual forma que con el método MPL, es decir, pulsando el botón *Segmentar Papila*. Inicialmente el proceso comenzará localizando la papila y extrayendo la subventana, posteriormente se podrá ir viendo el proceso de segmentación conforme van evolucionando los AGs. Finalmente aparecerá una ventana emergente informando del resultado de la segmentación , adicionalmente aparecerá el tiempo y discrepancia en un subpanel de la parte inferior de la pantalla (como podemos ver en la Figura 27)

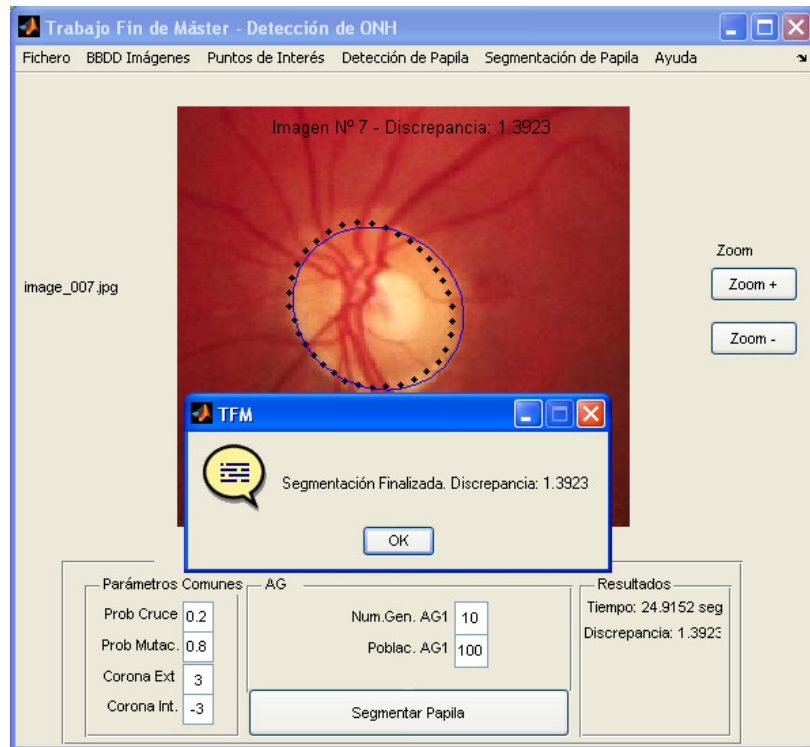


Figura 27 : Resultado de la segmentación con MFWF

Menú - Ayuda

Este menú muestra un panel con información del autor y el director del TFM

Apéndice B. Código

Para la aplicación de ambos métodos se utilizan varios ficheros principales y otros ficheros que realizan funciones auxiliares. Se utilizan tres ficheros principales por cada método:

- Fichero *principal.m* : carga la configuración del método y llama a *detecpi23.m*
- Fichero *detecpi23.m*: es el encargado de calcular los PIs y de ejecutar el AG. Hay dos versiones, una para el MPL y otra para el MFWF
- Fichero *papilalaplace.m*: es el encargado de contar los PIs dentro de la corona, es decir, es la función fitness. Ambos métodos utilizan la misma versión.

Fichero PRINCIPAL.M para el método MPL

FICHERO: PRINCIPAL.M para método MPL

```

%*****
% Trabajo Fin De Master - Detección de ONH
% UNED Curso 2010/2011
% José María Molina Casado
%*****
%
%                               FUNCIÓN principal
%
%
%Función que genera los resultados de un conjunto de experimentos resultado
%de evaluar la función detecpiXX para distintos valores de parámetros
%que configuran la parte de AG's de dicho programa. Dichos parámetros se
%pasan como variables globales.
%
%ENTRADAS
%
%EJEMPLO
%
%   principal
%   detecpiXX(NumPapila,NumPapila,pausa,grabar,verbose);
%   pausa=1 Activa, otro número Desactiva
%*****
function principal

clear all
close all

global NUM_ITER
global CTE_GENERACIONES
global CTE_POBLACION
global CTE_NUM_ELITE
global CTE_PROB_CROSS
global CTE_PROB_MUTA
global CTE_LIM_GEN_SIN_CAMBIO

global umbral_canny
global sigma_canny
global umbral_sobel_v
global umbral_sobel_h
global CoronaCteInt
global CoronaCteExt
global elitismo
global G2
global P2

for i=1:5 %5 pruebas

```

```

nombre=[num2str(i),'.txt']
param=lee_fich_config(num2str(nombre));
fid=fopen(['tiempos_',nombre, '.txt'],'w');
tic
for NUM_ITER=1:1
    param=str2double(param(1:7))
    CTE_GENERACIONES=param(1)
    CTE_POBLACION=param(2)
    CTE_NUM_ELITE=0;
    CTE_PROB_CROSS=0.2;
    CTE_PROB_MUTA=0.8;
    CTE_LIM_GEN_SIN_CAMBIO=100;
    elitismo=param(3);%para AG2
    CoronaCteExt=param(4)
    CoronaCteInt=param(5)
    G2=param(6)
    P2=param(7)
    detecpi23(1,110,0,0,0,num2str(i));%paso el número de iteración para guardar
resultados
end
nombrefich=[nombre, '.jpg'];
EvaluaPapilaLaplace(nombrefich)
close all
fprintf(fid,'%d\n',toc);
end

fclose(fid);
end

```

Fichero DETECPI23.M para el método MPL

FICHERO: DETECPI23.M para método MPL

```

%*****
% Trabajo Fin De Master - Detección de ONH
% UNED Curso 2010/2011
% José María Molina Casado
%*****
%
%                FUNCIÓN DE GENERACIÓN DE IPS Y MÉTODO MPL
% PASOS PREVIOS
%   -GENERACIÓN DE CONSTANTES Y VARIABLES
%   -INICIALIZACIÓN
% PASO 0 - LECTURA DE IMAGEN Y GENERACIÓN DE VARIABLES GENERALES
% PASO 1 - BUCLE DE LECTURA - IMÁGENES Y ANOTACIONES DE EXPERTOS
% PASO 2 - OBTENCIÓN DE ELIPSES APROXIMADAS A ANOTACIONES DE EXPERTOS
% PASO 3 - OBTENCIÓN DE SUBVENTANA - TAMAÑO 289X289
% PASO 4 - NORMALIZACIÓN DE LOS CANALES a [0,255]
% PASO 5 - OBTENCIÓN DE PUNTOS DE INTERÉS.
%           PASO 5.1 - PIRÁMIDE LAPLACIANA
%           PASO 5.2 - OBTENCIÓN DE LOS PI
% PASO 6 - ALGORITMO GENÉTICO 1
% PASO 7 - DECODIFICACIÓN DEL CROMOSOMA
%           PASO 8.1 - EXTRACCIÓN DE LOS X MEJORES INDIVIDUOS. TRASLADO AL SIGUIENTE
NIVEL
%           PASO 8.2 - EXPANSIÓN DE PUNTOS Y SELECCIÓN DE MEJORES
%           INDIVIDUOS
%           PASO 8.3 - ALGORITMO GENÉTICO 2
%           PASO 8.4 - DECODIFICACIÓN DEL CROMOSOMA
% PASO 9 - PLOTEADO DE RESULTADOS

function detecpi23(NumPapilaIni,NumPapilaFin,pausa,grabar,verbose,numiter)

%% CONSTANTES Y VARIABLES

global verbose % Información
global pausa;

%Variables Elipses
global x0 % origen del inicio de la subventana
global y0 % origen del inicio de la subventana

%Variables globales para realizar los distintos experimentos estadísticos

```

```

global CTE_GENERACIONES
global CTE_POBLACION
global CTE_NUM_ELITE
global CTE_PROB_CROSS
global CTE_PROB_MUTA
global CTE_LIM_GEN_SIN_CAMBIO
global NUM_ITER
global PoblacionMejoresIndividuos
PoblacionMejoresIndividuos=[];
crear_figXX=0;

global elitismo
global G2
global P2

%Variables globales que se generan aquí
global MatrizPtosX %Almacena la coordenada X de todos los ptos hipótesis
global MatrizPtosY %Almacena la coordenada Y de todos los ptos hipótesis

%Segunda Corona
global CoronaCteExt
global CoronaCteInt

%% INICIALIZACIÓN
mkdir('resultados');

%ALGORITMO GENÉTICO
fitnessFunction = @PapilaLaplace;%Fitness function

nvars=5;%Number of Variables
POBLACION=CTE_POBLACION;%100*nvars;%Tamaño de la población
NUM_ELITE=CTE_NUM_ELITE;%4;%Número de individuos élite
PROB_CROSS=CTE_PROB_CROSS;%0.8;%Probabilidad de crossover
PROB_MUTA=CTE_PROB_MUTA;%0.8;%Probabilidad de mutación
GENERACIONES=CTE_GENERACIONES;%Número de generaciones
LIM_GEN_SIN_CAMBIO=CTE_LIM_GEN_SIN_CAMBIO;%30;%Núm máx de generaciones que finalizan la
ejecución
CTE_NO_CONVERG=1e3;% Para descartar soluciones no deseables
%Dominio de definición de cada una de las variables
global LI_EjeMay
global LS_EjeMay
LI_EjeMay=8; LS_EjeMay=15;
global LI_EjeMen
global LS_EjeMen
LI_EjeMen=8; LS_EjeMen=15;
global LI_Desf
global LS_Desf
LI_Desf=deg2rad(15);%15 grados
LS_Desf=deg2rad(175);%175 grados
global LI_CentroX
global LS_CentroX
LI_CentroX=15;%Límite para que la elipse no excede la subventana de 65pix
LS_CentroX=50;%Límite para que la elipse no excede la subventana de 65pix
global LI_CentroY
global LS_CentroY
LI_CentroY=15;%Límite para que la elipse no excede la subventana de 65pix
LS_CentroY=50;%Límite para que la elipse no excede la subventana de 65pix

%GRABAR TODAS LAS ELIPSES IMPLICADAS EN LA SOLUCIÓN DE UNA ONH
grabar_elipses=1;
RutaGrabarElipses='C:\elipses.mat';

%DEFINE MANEJADORES DE FIGURAS
id_fig_Elipse=figure(1); %Plot para Imagen Expertos
set(id_fig_Elipse,'WindowStyle','docked');
id_fig_PYE=figure(2); %Plot Puntos y Elipses
set(id_fig_PYE,'WindowStyle','docked');
id_fig_Nivel1=figure(3); %Plot para Imagen en Nivel 1
set(id_fig_Nivel1,'WindowStyle','docked');

%Medidas de tiempo
Tiempo_total=0;
ALMACENAR_TIEMPOS_MEDIOS=1;
global Tiempos_medios_MIN

```

```

%% PASO 0 - LECTURA DE IMAGEN Y GENERACIÓN DE VARIABLES GENERALES
subx=160;%Inicio subventana x
suby=80;%Inicio subventana y
% Las imagenes deben haber sido convertidas a ojos izquierdos mediante
Reordena_Ojos_Derechos.m"
path_ExpertContornos='C:\Archivos de programa\MATLAB\R2007b\work\DRIONS-
DB\experts_annotacion\';
path_Imagenes='C:\Archivos de programa\MATLAB\R2007b\work\DRIONS-DB\imagenes\';

%% PASO 1 - BUCLE DE LECTURA - IMÁGENES Y ANOTACIONES DE EXPERTOS
for im=NumPapilaIni:NumPapilaFin
    if im<10
        path_Expert1Contorno=[path_ExpertContornos,'anotExpert1_00',num2str(im),'.txt'];
        path_Expert2Contorno=[path_ExpertContornos,'anotExpert2_00',num2str(im),'.txt'];
        %path_Imagen=[path_Imagenes,'image_00',num2str(im),'.jpg'];
    else if im<100

path_Expert1Contorno=[path_ExpertContornos,'anotExpert1_0',num2str(im),'.txt'];

path_Expert2Contorno=[path_ExpertContornos,'anotExpert2_0',num2str(im),'.txt'];
        %path_Imagen=[path_Imagenes,'image_0',num2str(im),'.jpg'];
    else

path_Expert1Contorno=[path_ExpertContornos,'anotExpert1_',num2str(im),'.txt'];

path_Expert2Contorno=[path_ExpertContornos,'anotExpert2_',num2str(im),'.txt'];
        %path_Imagen=[path_Imagenes,'image_',num2str(im),'.jpg'];
    end
end

%% PASO 2 - OBTENCIÓN DE ELIPSES APROXIMADAS A ANOTACIONES DE EXPERTOS
Exp1Contorno=load(path_Expert1Contorno);
Exp2Contorno=load(path_Expert2Contorno);
%Se ajusta una elipse a los dos contornos de los expertos para ver solución
%óptima
[a,b,x0i,y0i,phi]=ellipse_fit([Exp1Contorno(:,1) ; Exp2Contorno(:,1)], [Exp1Contorno(:,2)
; Exp2Contorno(:,2)])
%phi en radianes, inclinación con respecto a eje X

%% PASO 3 - OBTENCIÓN DE SUBVENTANA - TAMAÑO 256X256 (Subventana en nivel 2
%% a 65x65
[I, subx, suby]=ExtraeSubventana2(im);% Con este script se extrae la subventana
automáticamente

figure(1);
imshow(I);

hold on;
%Dibuja elipses expertos. Resto de forma invertida porque la imagen tiene
%origen inferior izquierda, y al dibujar puntos tiene origen superior
%izquierda
plot(Exp1Contorno(:,1)-suby,Exp1Contorno(:,2)-subx,'g.-');%VERDE
plot(Exp2Contorno(:,1)-suby,Exp2Contorno(:,2)-subx,'c.-');%CYAN
%Dibuja elipse aproximada
DibujaElipse(x0i-suby,y0i-subx,a,b,rad2deg(phi),1);%Trabaja en Grados
I=im2double(I);
if pausa
    pause;
end
hold off;

%% PASO 4 - NORMALIZACIÓN DE LOS CANALES a [0,255]
for c=1:3 % 3 Niveles de Color RGB
I(:, :, c)=(I(:, :, c)-min(min(I(:, :, c))))*(255/(max(max(I(:, :, c)))-(min(min(I(:, :, c))))));
end

%% PASO 5 - OBTENCIÓN DE PUNTOS DE INTERÉS
%% PASO 5.1 - PIRÁMIDE LAPLACIANA
%% REDUCCIONES Y EXPANSIONES
R0=impyramid(I,'reduce');%Reducción de la imagen original Ic (Nivel 0)
E0=impyramid(R0,'expand');%Expansión hacia el Nivel 0
R1=impyramid(R0,'reduce');%Reducción del Nivel 1
E1=impyramid(R1,'expand');%Expansión hacia el Nivel 1
R2=impyramid(R1,'reduce');%Reducción del Nivel 2
E2=impyramid(R2,'expand');%Expansión hacia el Nivel 2

%% CREACIÓN DE PIRÁMIDE LAPLACIANA y LAPLACIANAS NORMALIZADAS

```

```

L0=I-E0;%Laplaciana de Nivel 0
L1=R0-E1;%Laplaciana de Nivel 1
L2=R1-E2;%Laplaciana de Nivel 2
%Normalizo la Laplaciana tomando solo los valores positivos del Nivel 2
L2_dif(:, :, 1)=L2(:, :, 1)*(255/(max(max(L2(:, :, 1)))));
L2_dif(:, :, 2)=L2(:, :, 2)*(255/(max(max(L2(:, :, 2)))));
L2_dif(:, :, 3)=L2(:, :, 3)*(255/(max(max(L2(:, :, 3)))));
if verbose
    hold off;
    imshow(L2, []);xlabel('Laplaciana Nivel 2');
    if pausa
        pause;
    end
end;
%% PASO 5.2 - OBTENCIÓN DE LOS PI

%Se Ajusta el umbral automáticamente para obtener 200 puntos en L2
L2_dif_RGB=0.8*L2_dif(:, :, 1)+0.1*L2_dif(:, :, 2)+0.1*L2_dif(:, :, 3);
umbral_L2=255;
L2_dif_max_bw=(L2_dif_RGB>umbral_L2);
while sum(sum(L2_dif_max_bw))<200% Se queda con 200 Puntos
    umbral_L2=umbral_L2-1;
    L2_dif_max_bw=(L2_dif_RGB>umbral_L2);
end
figure(2);
imshow(L2_dif_max_bw);
%pause;

% Se guardan los puntos Hipótesis en dos vectores X, Y
% Se almacenan al revés porque usa OSI (origen superior izquierdo)
[MatrizPtosY,MatrizPtosX]=find(L2_dif_max_bw>0);

%% PASO 6 - ALGORITMO GENÉTICO

%CONFIGURACIÓN DE LAS OPCIONES DEL AG
%Start with default options
options = gaoptimset;

%%MODIFICA ALGUNOS PARÁMETROS POR DEFECTO:

%DOMINIO DE DEFINICIÓN DE LAS VARIABLES
options = gaoptimset(options, 'PopInitRange', ...
    [LI_CentroX,LI_CentroY,LI_EjeMay,LI_EjeMen,LI_Desf;...
    LS_CentroX,LS_CentroY,LS_EjeMay,LS_EjeMen,LS_Desf]);
options = gaoptimset(options, 'PopulationType' , 'doubleVector');

%TAMAÑO DE LA POBLACIÓN Y NÚMERO DE GENERACIONES
options = gaoptimset(options, 'PopulationSize' ,POBLACION);
options = gaoptimset(options, 'Generations' ,GENERACIONES);

%OPERADORES DE SELECCIÓN
%options = gaoptimset(options, 'FitnessScalingFcn' ,@fitscalingprop); %Valor por
defecto @fitscalingrank
options = gaoptimset(options, 'FitnessScalingFcn' , @fitscalingrank);
options = gaoptimset(options, 'SelectionFcn' , @selectionstochunif); %Valor por
defecto @selectionstochunif
options = gaoptimset(options, 'EliteCount' , NUM_ELITE);%Valor por defecto 2

%OPERADOR DE CRUCE
%options = gaoptimset(options, 'CrossoverFcn',@crossoverscattered); %Valor por
defecto @crossoverscattered
options = gaoptimset(options, 'CrossoverFcn' ,@crossoversinglepoint);
options = gaoptimset(options, 'CrossoverFcn' ,@crossovertwopoint);
options = gaoptimset(options, 'CrossoverFraction' ,PROB_CROSS);%Valor por defecto 0.8

%OPERADOR DE MUTACIÓN
%options = gaoptimset(options, 'MutationFcn',@mutationgaussian);%Valor por defecto
@mutationgaussian
options = gaoptimset(options, 'MutationFcn' , { @mutationuniform
PROB_MUTA});%(1/nvars) );%Valor por defecto 0.01
options = gaoptimset(options, 'MutationFcn' , { @mutationgaussian 0.1 PROB_MUTA });

%CONDICIONES DE PARADA
options = gaoptimset(options, 'StallGenLimit' , LIM_GEN_SIN_CAMBIO);
options = gaoptimset(options, 'StallTimeLimit' , Inf);
options = gaoptimset(options, 'FitnessLimit' , 1);

```

```

%SELECCIÓN DE FUNCIÓN DE SALIDA
options = gaoptimset(options,'PlotFcns' ,{ @gaplotbestindiv });

%OPCIONES DE SALIDA
if verbose
    options = gaoptimset(options,'PlotFcns' ,{ @gaplotbestf });
    %options = gaoptimset(options,'Display' ,'iter');
end
%%Run GA
[CromosomaTop,FVAL,REASON,OUTPUT,FINAL_POPULATION,SCORES] =
ga(fitnessFunction,nvars,options);

% if FVAL~=CTE_NO_CONVERG
%if verbose
    disp('FVAL')
    disp(FVAL)
    disp('REASON')
    disp(REASON)
    disp('OUTPUT')
    disp(OUTPUT)
    disp('FINAL_POPULATION')
    disp(FINAL_POPULATION)
    disp('SCORES')
    disp(SCORES)
    disp('CromosomaTop')
    disp(CromosomaTop)
%end

%% PASO 7 - DECODIFICACIÓN DEL CROMOSOMA

%DECODIFICA EL CROMOSOMA
CentroX_Gen=CromosomaTop(1);
CentroY_Gen=CromosomaTop(2);
EjeMay_Gen=CromosomaTop(3);
EjeMen_Gen=CromosomaTop(4);
desf_Gen=CromosomaTop(5);

%GUARDA LOS RESULTADOS EN LA ESTRUCTURA GENERAL DE RESULTADOS
%Multiplico por 2 para llevar al Subnivel 1 de la Pirámide Laplaciana
Resultados.cGenx{im}=CentroX_Gen*2;
Resultados.cGeny{im}=CentroY_Gen*2;
Resultados.EjeMay_GenAll{im}=EjeMay_Gen*2;
Resultados.EjeMen_GenAll{im}=EjeMen_Gen*2;
Resultados.desf_GenAll{im}=rad2deg(desf_Gen);%Paso a grados porque luego van a ser
utilizados desde el cálculo de la discrepancia

figure(2);
DibujaElipse(CentroX_Gen,CentroY_Gen,EjeMay_Gen,EjeMen_Gen,rad2deg(desf_Gen),1);%Trabaja
en Grados
%pause;

%% PASO 8.1 - EXTRACCIÓN DE LOS X MEJORES INDIVIDUOS. TRASLADO AL SIGUIENTE NIVEL
SCORESmodif=SCORES;
switch (P2)
    case 50, Mejores=25;
    case 100, Mejores=50;
    case 200, Mejores=100;
    case 400, Mejores=200;
    case 1000,Mejores=200;
end
for k=1:Mejores
    [val indice]=min(SCORESmodif);%Busco el mejor individuo
    NUEVAPOB(k,:)=FINAL_POPULATION(indice,:);%Llevo la solución al siguiente subnivel
multiplicando por 2
    SCORESmodif(indice)=1e5;%anulo el mejor para que no lo tenga en cuenta
end
NUEVAPOB=NUEVAPOB*2;%Llevo la solución al siguiente subnivel multiplicando por 2
NUEVAPOB(:,5)=NUEVAPOB(:,5)/2;% Deshago el cambio para el ángulo
%% PASO 8.2 - PUNTOS EN SUBNIVEL 1
%Normalizo la Laplaciana tomando solo los valores positivos del Nivel 2
L1_dif(:, :, 1)=L1(:, :, 1) * (255 / (max(max(L1(:, :, 1)))));
L1_dif(:, :, 2)=L1(:, :, 2) * (255 / (max(max(L1(:, :, 2)))));
L1_dif(:, :, 3)=L1(:, :, 3) * (255 / (max(max(L1(:, :, 3)))));

hold off;
figure(2);

```



```

imshow(L2, []);xlabel('Laplaciana Nivel 1');
%pause;

%Se Ajusta el umbral automáticamente para obtener 300 puntos en L1
L1_dif_RGB=0.8*L1_dif(:, :, 1)+0.1*L1_dif(:, :, 2)+0.1*L1_dif(:, :, 3);
umbral_L1=255;
L1_dif_max_bw=(L1_dif_RGB>umbral_L1);
while sum(sum(L1_dif_max_bw))<200% Se queda con 200 Puntos
    umbral_L1=umbral_L1-1;
    L1_dif_max_bw=(L1_dif_RGB>umbral_L1);
end
imshow(L1_dif_max_bw);

% Se guardan los puntos Hipótesis en dos vectores X, Y
% Se almacenan al revés porque usa OSI (origen superior izquierdo)
[MatrizPtosY,MatrizPtosX]=find(L1_dif_max_bw>0);

%% PASO 8.3 - SEGUNDA LLAMADA AL GENÉTICO
options = gaoptimset(options, 'PopulationSize' ,P2);
P=creapoblacion(5,options);% Llamo a la función para crear la nueva población aleatoria
size(P)
P=P*2;%Llevo la solución al siguiente subnivel multiplicando por 2
P(:,5)=P(:,5)/2;% Deshago el cambio para el ángulo
NP=[P(1:(P2-Mejores),:); NUEVAPOB];% Creo la nueva población, 50 individuos aleatorios y
los 50 mejores del AG anterior
%pause;
options = gaoptimset(options, 'Generations' ,G2);
if elitismo~=0
    options = gaoptimset(options, 'EliteCount' ,1);
    %disp 'Elitismo'
    %pause;
end

%Modifico los límites porque trabajamos al doble de resolución
LI_EjeMay=LI_EjeMay*2;
LS_EjeMay=LS_EjeMay*2;
LI_EjeMen=LI_EjeMen*2;
LS_EjeMen=LS_EjeMen*2;
LI_CentroX=LI_CentroX*2;
LS_CentroX=LS_CentroX*2;
LI_CentroY=LI_CentroY*2;
LS_CentroY=LS_CentroY*2;

options = gaoptimset(options, 'PopInitRange', ...
    [LI_CentroX,LI_CentroY,LI_EjeMay,LI_EjeMen,LI_Desf;...
    LS_CentroX,LS_CentroY,LS_EjeMay,LS_EjeMen,LS_Desf]);
options = gaoptimset(options, 'InitialPopulation', NP);
[CromosomaTop,FVAL,REASON,OUTPUT,FINAL_POPULATION,SCORES] =
ga(fitnessFunction,nvars,options);

    disp('FVAL')
    disp(FVAL)
    disp('REASON')
    disp(REASON)
    disp('OUTPUT')
    disp(OUTPUT)
    disp('FINAL_POPULATION')
    disp(FINAL_POPULATION)
    disp('SCORES')
    disp(SCORES)
    disp('CromosomaTop')
    disp(CromosomaTop)

%% PASO 8.4 - DECODIFICACIÓN DEL CROMOSOMA

%DECODIFICA EL CROMOSOMA
CentroX_Gen=CromosomaTop(1);
CentroY_Gen=CromosomaTop(2);
EjeMay_Gen=CromosomaTop(3);
EjeMen_Gen=CromosomaTop(4);
desf_Gen=CromosomaTop(5);

%GUARDA LOS RESULTADOS EN LA ESTRUCTURA GENERAL DE RESULTADOS

```

```

%Multiplico por 2 para llevar al Subnivel 0 de la Pirámide Laplaciana
Resultados.cGenx{im}=CentroX_Gen*2;
Resultados.cGeny{im}=CentroY_Gen*2;
Resultados.EjeMay_GenAll{im}=EjeMay_Gen*2;
Resultados.EjeMen_GenAll{im}=EjeMen_Gen*2;
Resultados.desf_GenAll{im}=rad2deg(desf_Gen);%Paso a grados porque luego van a ser
utilizados desde el cálculo de la discrepancia

%% PASO 9 - PLOTEADO DE RESULTADOS
%CONSTRUYE ELIPSE GENÉTICA
figure(2);
hold on;

DibujaElipse(CentroX_Gen,CentroY_Gen,EjeMay_Gen,EjeMen_Gen,rad2deg(desf_Gen),1);%Trabaja
en Grados
saveas(figure(2),['c:\elipsegen_',num2str(im),'.jpg']);
hold on;
%MEJOR CROMOSOMA
disp('Mejor Cromosoma');
disp(CromosomaTop)
if pausa pause;
end
%CREACIÓN DE LA ELIPSE GENÉTICA EN NIVELES DE GRIS
figure(3);
IG2=zeros(65,65);
IG2=im2bw(IG2); % Para que la elipse se expanda en forma 'binaria'
imshow(IG2);

ptos=DibujaElipse(CentroX_Gen,CentroY_Gen,EjeMay_Gen,EjeMen_Gen,rad2deg(desf_Gen),1);
%dibujo en Grados
vec(:,1)=ptos(:,1);%get(h,'xdata');
vec(:,2)=ptos(:,2);%get(h,'ydata');
for i=1:size(vec,1)
    IG2(round(vec(i,2)),round(vec(i,1)))=255;
    %IG2(round(vecy),round(vecx))=255;
end

IG2_=funcion3(IG2); % Llamada a la función 3. Expande 1 nivel
%ELIPSE GENÉTICA EN NIVEL 1
imshow(IG2_);
hold on;
plot(Exp1Contorno(:,1)-suby,Exp1Contorno(:,2)-subx,'g.-');%VERDE
plot(Exp2Contorno(:,1)-suby,Exp2Contorno(:,2)-subx,'c.-');%CYAN
DibujaElipse(x0i-suby,y0i-subx,a,b,rad2deg(phi),1); %AZUL. Trabajo en Radianes pero
luego dibujo en Grados

hold off;
if pausa
    pause;
end

%Modifico los límites, deshago el cambio para que el primer genético
%funcione correctamente
LI_EjeMay=LI_EjeMay/2;
LS_EjeMay=LS_EjeMay/2;
LI_EjeMen=LI_EjeMen/2;
LS_EjeMen=LS_EjeMen/2;
LI_CentroX=LI_CentroX/2;
LS_CentroX=LS_CentroX/2;
LI_CentroY=LI_CentroY/2;
LS_CentroY=LS_CentroY/2;

end % Fin de Bucle de imágenes

save 'Resultados.mat' Resultados

end

%En esta versión se va a trabajar directamente con el perímetro de la
%elipse teniendo en cuenta un offset 0

%Crea una nueva población aleatoria
function Population = creapoblacion(GenomeLength,options)

```

```

tmp_aux = gaoptimget (options, 'PopInitRange');
tampobla = gaoptimget (options, 'PopulationSize');
for i=1:tampobla
    Population(i,:)=(tmp_aux(2,:)-tmp_aux(1,:))*rand)+tmp_aux(1,:);
end
end

```

Fichero PAPILALAPLACE.M para ambos métodos

FICHERO: PAPILALAPLACE.M para ambos métodos

```

%*****
%*****
%          FUNCIÓN PapilaLaplace : Cuenta el número de puntos dentro de la corona
elíptica
%
%RESTRICCIÓN
%
%ENTRADAS
%  Cromosoma  ->  Vector real de 7 variables que codifica los parámetros que definen
una elipse:
%
%              (1) Coord X del centro
%              (2) Coord Y del centro
%              (3) Semieje mayor.
%              (4) Semieje menor.
%              (5) Desfase (ángulo del eje mayor con el eje X).
%SALIDAS
%  out        ->  Valor de la función a minimizar
%
%*****
%*****f
function out=PapilaLaplace(Cromosoma)

%DEFINICIÓN DE VARIABLES GLOBALES
%Variables globales que se generan en 'GenPtosHipo9'
global verbose;
global pausa;

global VPC % Vector de Puntos pertenecientes a la Corona

global MatrizPtosX %Almacena la coordenada X de todos los ptos hipótesis
global MatrizPtosY %Almacena la coordenada Y de todos los ptos hipótesis

global LI_CentroX
global LS_CentroX

global LI_CentroY
global LS_CentroY

global LI_EjeMay
global LS_EjeMay

global LI_EjeMen
global LS_EjeMen

global LI_Desf
global LS_Desf

global CTE_NO_CONVERG %Determina si hubo (1) o no (0) una solución que, al menos,
satisface las restricciones

global CoronaCteExt
global CoronaCteInt

alfa=50; %pendiente de las dos rectas (+-alfa) que delimitan las zonas temp-sup-
nasal-inf de la elipse

%Decodifica el cromosoma y construye la señal de distancia de la elipse

```

```

CentroX_Gen=Cromosoma(1);
CentroY_Gen=Cromosoma(2);
EjeMay=Cromosoma(3);
EjeMen=Cromosoma(4);
desf=Cromosoma(5);

%Restricciones de limitación de centro y ejes
if (EjeMay<EjeMen | (EjeMay/EjeMen)>1.4 |...
    CentroX_Gen<LI_CentroX | CentroX_Gen>LS_CentroX |...
    CentroY_Gen<LI_CentroY | CentroY_Gen>LS_CentroY |...
    EjeMay<LI_EjeMay | EjeMay>LS_EjeMay |...
    EjeMen<LI_EjeMen | EjeMen>LS_EjeMen )
    %out=CTE_NO_CONVERG; % Si se violan las restricciones la solución es inválida
    out=100000; % Si se violan las restricciones la solución es inválida
    %CentroX_Gen;

    %disp 'no valida';pause;

else
%Se dibujan las elipses
a2=EjeMay+CoronaCteExt;
b2=EjeMen+CoronaCteExt;
a=EjeMay+CoronaCteInt;
b=EjeMen+CoronaCteInt;

if pausa
    ellipse(a,b,desf, CentroX_Gen, CentroY_Gen, ['y']);
    ellipse(a2,b2,desf, CentroX_Gen, CentroY_Gen, ['y']);
    pause;
end
c=sqrt(a*a-b*b);e=c/a;
c2=sqrt(a2*a2-b2*b2);e2=c2/a2;
% pause;
hold on;

%COMPROBACIÓN: PERTENENCIA DE UN PUNTO A LA CORONA

PI_in=0;%Inicializa contador puntos en corona
VPC=VPC*0;%Inicializo el vector de puntos dentro de corona

for j=1:size(MatrizPtosX)%Número de puntos
    x=MatrizPtosX(j)-CentroX_Gen;
    y=MatrizPtosY(j)-CentroY_Gen;
    xt=x*cos(desf)+y*sin(desf);%Rotamos componente X
    yt=y*cos(desf)-x*sin(desf);%Rotamos componente Y
    t1=a*(sqrt((1-e*e)/(1-(e*e)*cos(atan((yt/xt))))));% Primera región de
aceptación
    reg_acep=sqrt((xt*xt)+(yt*yt));
aceptación
    t2=a2*(sqrt(((1-e2*e2)/(1-(e2*e2)*cos(atan(yt/xt))))));% Segunda región de
    if ((t1<=reg_acep)&&(reg_acep<=t2)==1)
        if (verbose==1)
            plot(x+CentroX_Gen,y+CentroY_Gen,'gs','LineWidth',1); %Punto
pertenece a la corona
        end
        PI_in=PI_in+1;
        VPC(PI_in,1)=x+CentroX_Gen; %Guarda componente X en Vector Puntos
Corona
        VPC(PI_in,2)=y+CentroY_Gen; %Guarda componente Y en Vector Puntos
Corona
        %pause;
    else
        if (verbose==1)
            plot(x+CentroX_Gen,y+CentroY_Gen,'rs','LineWidth',1); %Punto no
pertenece a la corona
        end
        %pause;
    end
end

out=1000/PI_in; % Invertimos el número ya que la función fitness está minimizando
end

```

Fichero PRINCIPAL.M para el método MFWF

FICHERO: PRINCIPAL.M para el método MFWF

```

%*****
%*****
% Trabajo Fin De Master - Detección de ONH
% UNED Curso 2010/2011
% José María Molina Casado
%*****
%*****
%                               FUNCIÓN principal
%
%
%Función que genera los resultados de un conjunto de experimentos resultado
%de evaluar la función detecpiXX para distintos valores de parámetros
%que configuran la parte de AG's de dicho programa. Dichos parámetros se
%pasan como variables globales.
%
%
%ENTRADAS
%
%EJEMPLO
%
%   principal
%   detecpiXX(NumPapila, NumPapila, pausa, grabar, verbose);
%   pausa=1 Activa, otro número Desactiva
%*****
%*****
function principal

clear all
close all

global NUM_ITER
global CTE_GENERACIONES
global CTE_POBLACION
global CTE_NUM_ELITE
global CTE_PROB_CROSS
global CTE_PROB_MUTA
global CTE_LIM_GEN_SIN_CAMBIO

global umbral_canny
global sigma_canny
global umbral_sobel_v
global umbral_sobel_h
global CteInt
global CteExt
global elitismo
global wavfile
global escalas

for i=1:5 % 5 ejecuciones
tic
for NUM_ITER=1:1
    CTE_POBLACION=400;
    CTE_NUM_ELITE=0;
    CTE_PROB_CROSS=0.2;
    CTE_PROB_MUTA=0.8;
    CTE_LIM_GEN_SIN_CAMBIO=20;
    nombre=[num2str(i), '.txt'];
    param=lee_fich_config(num2str(nombre));
    wavfile=cell2mat(param(9));
    param=str2double(param(1:10))
    umbral_canny=param(1)
    sigma_canny=param(2)
    umbral_sobel_v=param(3)
    umbral_sobel_h=param(4)
    CteExt=param(5)
    CteInt=param(6)
    elitismo=param(7)
    CTE_GENERACIONES=param(8);
    escalas=param(10);

```

```

    detecpi23(1,110,0,0,0);
end
tiempos(i)=toc
nombrefich=[nombre, '.jpg'];
EvaluaPapilaLaplace(nombrefich)
close all
end
save(['tiempo', nombre], 'tiempos');

end

```

Fichero DETECPI23.M para el método MFWF

FICHERO: DETECPI23.M para el método MFWF

```

%*****
%*****
% Trabajo Fin De Master - Detección de ONH
% UNED Curso 2010/2011
% José María Molina Casado
%*****
%*****
%
%                               FUNCIÓN DE GENERACIÓN DE IPS Y MÉTODO MFWF
% PASOS PREVIOS
%   -GENERACIÓN DE CONSTANTES Y VARIABLES
%   -INICIALIZACIÓN
% PASO 0 - LECTURA DE IMAGEN Y GENERACIÓN DE VARIABLES GENERALES
% PASO 1 - BUCLE DE LECTURA - IMÁGENES Y ANOTACIONES DE EXPERTOS
% PASO 2 - OBTENCIÓN DE ELIPSES APROXIMADAS A ANOTACIONES DE EXPERTOS
% PASO 3 - OBTENCIÓN DE SUBVENTANA - TAMAÑO 289X289
% PASO 4 - NORMALIZACIÓN DE LOS CANALES a [0,255]
% PASO 5 - OBTENCIÓN DE PUNTOS DE INTERÉS. BUCLE PARA CADA CANAL, UMBRAL Y MODO DE
COLOR
%
%           PASO 5.1 - FUSIÓN DE CANALES
%           PASO 5.2 - OBTENCIÓN DE LOS PI POR FILTROS DE CANNY Y SOBEL
% PASO 6 - ALGORITMO GENÉTICO
% PASO 7 - DECODIFICACIÓN DEL CROMOSOMA
% PASO 8 - PLOTEADO DE RESULTADOS

function detecpi23(NumPapilaIni,NumPapilaFin,pausa,grabar,verbose)

%% CONSTANTES Y VARIABLES

global verbose % Información
global pausa;

%Variables Elipses
global x0 % origen del inicio de la subventana
global y0 % origen del inicio de la subventana

%Variables globales para realizar los distintos experimentos estadísticos
global CTE_GENERACIONES
global CTE_POBLACION
global CTE_NUM_ELITE
global CTE_PROB_CROSS
global CTE_PROB_MUTA
global CTE_LIM_GEN_SIN_CAMBIO
global NUM_ITER
global PoblacionMejoresIndividuos
PoblacionMejoresIndividuos=[];
crear_figXX=0;

%Variables globales que se generan aquí
global MatrizPtosX %Almacena la coordenada X de todos los pts hipótesis
global MatrizPtosY %Almacena la coordenada Y de todos los pts hipótesis

%Segunda Corona
global CoronaCteExt
global CoronaCteInt

%Parámetros que lee desde el fichero de configuración

```

```

global umbral_canny
global sigma_canny
global umbral_sobel_v
global umbral_sobel_h
global CteInt
global CteExt
global elitismo
global escalas
global wavfile

%% INICIALIZACIÓN
mkdir('resultados');

CoronaCteExt=-0.5;
CoronaCteInt=-1.5;

%ALGORITMO GENÉTICO
fitnessFunction = @PapilaLaplace;%Fitness function

nvars=5;%Number of Variables
POBLACION=CTE_POBLACION;%100*nvars;%Tamaño de la población
NUM_ELITE=CTE_NUM_ELITE;%4;%Número de individuos elite
PROB_CROSS=CTE_PROB_CROSS;%0.8;%Probabilidad de crossover
PROB_MUTA=CTE_PROB_MUTA;%0.8;%Probabilidad de mutación
GENERACIONES=CTE_GENERACIONES;%Número de generaciones
LIM_GEN_SIN_CAMBIO=CTE_LIM_GEN_SIN_CAMBIO;%30;%Núm máx de generaciones que finalizan la
ejecución
CTE_NO_CONVERG=1e3;% Para descartar soluciones no deseables
%Dominio de definición de cada una de las variables
global LI_EjeMay
global LS_EjeMay
LI_EjeMay=8;          LS_EjeMay=15;
global LI_EjeMen
global LS_EjeMen
LI_EjeMen=8;          LS_EjeMen=15;
global LI_Desf
global LS_Desf
LI_Desf=deg2rad(15);%15 grados
LS_Desf=deg2rad(175);%175 grados
global LI_CentroX
global LS_CentroX
LI_CentroX=15;%Límite para que la elipse no excede la subventana de 65pix
LS_CentroX=50;%Límite para que la elipse no excede la subventana de 65pix
global LI_CentroY
global LS_CentroY
LI_CentroY=15;%Límite para que la elipse no excede la subventana de 65pix
LS_CentroY=50;%Límite para que la elipse no excede la subventana de 65pix

%GRABAR TODAS LAS ELIPSES IMPLICADAS EN LA SOLUCIÓN DE UNA ONH
grabar_elipses=1;
RutaGrabarElipses='C:\elipses.mat';

%DEFINE MANEJADORES DE FIGURAS
    id_fig_Elipse=figure(1); %Plot para Imagen Expertos
    set(id_fig_Elipse,'WindowStyle','docked');
    id_fig_PYE=figure(2); %Plot Puntos y Elipses
    set(id_fig_PYE,'WindowStyle','docked');
    id_fig_Nivel1=figure(3); %Plot para Imagen en Nivel 1
    set(id_fig_Nivel1,'WindowStyle','docked');

%Medidas de tiempo
Tiempo_total=0;
ALMACENAR_TIEMPOS_MEDIOS=1;
global Tiempos_medios_MIN

%% PASO 0 - LECTURA DE IMAGEN Y GENERACIÓN DE VARIABLES GENERALES
subx=160;%Inicio subventana x
suby=80;%Inicio subventana y
% Las imagenes deben haber sido convertidas a ojos izquierdos mediante
"Reordena_Ojos_Derechos.m"
path_ExpertContornos='C:\Archivos de programa\MATLAB\R2007b\work\DRIONS-
DB\experts_anoatation\';
path_Imagenes='C:\Archivos de programa\MATLAB\R2007b\work\DRIONS-DB\images\';

```

```

%N=3; % 3 Niveles de Pirámide borrar

%% PASO 1 - BUCLE DE LECTURA - IMÁGENES Y ANOTACIONES DE EXPERTOS
%for im=[2,3,4,8,9,26,35,47,56,57,79,88,92,98]
for im=NumPapilaIni:NumPapilaFin
    if im<10
        path_Expert1Contorno=[path_ExpertContornos,'anotExpert1_00',num2str(im),'.txt'];
        path_Expert2Contorno=[path_ExpertContornos,'anotExpert2_00',num2str(im),'.txt'];
        %path_Imagen=[path_Imagenes,'image_00',num2str(im),'.jpg'];
    else if im<100
        path_Expert1Contorno=[path_ExpertContornos,'anotExpert1_0',num2str(im),'.txt'];
        path_Expert2Contorno=[path_ExpertContornos,'anotExpert2_0',num2str(im),'.txt'];
        %path_Imagen=[path_Imagenes,'image_0',num2str(im),'.jpg'];
    else
        path_Expert1Contorno=[path_ExpertContornos,'anotExpert1_',num2str(im),'.txt'];
        path_Expert2Contorno=[path_ExpertContornos,'anotExpert2_',num2str(im),'.txt'];
        %path_Imagen=[path_Imagenes,'image_',num2str(im),'.jpg'];
    end
end

%% PASO 2 - OBTENCIÓN DE ELIPSES APROXIMADAS A ANOTACIONES DE EXPERTOS
Exp1Contorno=load(path_Expert1Contorno);
Exp2Contorno=load(path_Expert2Contorno);

%% PASO 3 - OBTENCIÓN DE SUBVENTANA - TAMAÑO 256X256 (Subventana en nivel 2
%% a 65x65
[I, subx, suby]=ExtraeSubventana2(im);% Con este script se extrae la subventana
automáticamente

figure(1);
imshow(I);

hold on;
%Dibuja elipses expertos. Resto de forma invertida porque la imagen tiene
%origen inferior izquierda, y al dibujar puntos tiene origen superior
%izquierda
plot(Exp1Contorno(:,1)-suby,Exp1Contorno(:,2)-subx,'g.-');%VERDE
plot(Exp2Contorno(:,1)-suby,Exp2Contorno(:,2)-subx,'c.-');%CYAN
%Dibuja elipse aproximada
%DibujaElipse(x0i-suby,y0i-subx,a,b,rad2deg(phi),1);%Trabaja en Grados
I=im2double(I);
if pausa
    pause;
end
hold off;

%% PASO 4 - NORMALIZACIÓN DE LOS CANALES a [0,255]
for c=1:3 % 3 Niveles de Color RGB
I(:,:,c)=(I(:,:,c)-min(min(I(:,:,c))))*(255/(max(max(I(:,:,c)))-(min(min(I(:,:,c))))));
%imshow([I(:,:,i)],[]);pause;
end

%% PASO 5 - CONFIGURACIÓN ALGORITMO GENÉTICO

%CONFIGURACIÓN DE LAS OPCIONES DEL AG
%Start with default options
options = gaoptimset;

%%MODIFICA ALGUNOS PARÁMETROS POR DEFECTO:

%DOMINIO DE DEFINICIÓN DE LAS VARIABLES
options = gaoptimset(options,'PopInitRange',...
    [LI_CentroX,LI_CentroY,LI_EjeMay,LI_EjeMen,LI_Desf;...
    LS_CentroX,LS_CentroY,LS_EjeMay,LS_EjeMen,LS_Desf]);
options = gaoptimset(options,'PopulationType' , 'doubleVector');

%TAMAÑO DE LA POBLACIÓN Y NÚMERO DE GENERACIONES
options = gaoptimset(options,'PopulationSize' ,POBLACION);
options = gaoptimset(options,'Generations' ,GENERACIONES);

%OPERADORES DE SELECCIÓN

```



```

    %options = gaoptimset(options,'FitnessScalingFcn',@fitscalingprop); %Valor por
defecto @fitscalingrank
    options = gaoptimset(options,'FitnessScalingFcn', @fitscalingrank);
    options = gaoptimset(options,'SelectionFcn', @selectionstochunif); %Valor por
defecto @selectionstochunif
    options = gaoptimset(options,'EliteCount' ,NUM_ELITE);%Valor por defecto 2

    %OPERADOR DE CRUCE
    %options = gaoptimset(options, 'CrossoverFcn',@crossoverscattered); %Valor por
defecto @crossoverscattered
    options = gaoptimset(options,'CrossoverFcn' ,@crossoversinglepoint);
    %options = gaoptimset(options,'CrossoverFcn' ,@crossovertwopoint);
    options = gaoptimset(options,'CrossoverFraction' ,PROB_CROSS);%Valor por defecto 0.8

    %OPERADOR DE MUTACIÓN
    %options = gaoptimset(options, 'MutationFcn',@mutationgaussian);%Valor por defecto
@mutationgaussian
    %options = gaoptimset(options,'MutationFcn' ,{ @mutationuniform
PROB_MUTA});%(1/nvars) );%Valor por defecto 0.01
    options = gaoptimset(options,'MutationFcn' ,{ @mutationgaussian 0.1 PROB_MUTA });

    %CONDICIONES DE PARADA
    options = gaoptimset(options,'StallGenLimit' ,LIM_GEN_SIN_CAMBIO);
    options = gaoptimset(options,'StallTimeLimit' ,Inf);
    %options = gaoptimset(options,'FitnessLimit' ,1);

    %SELECCIÓN DE FUNCIÓN DE SALIDA
    options = gaoptimset(options,'PlotFcns' ,{ @gaplotbestindiv });

    %OPCIONES DE SALIDA
    if verbose
        options = gaoptimset(options,'PlotFcns' ,{ @gaplotbestf });
        %options = gaoptimset(options,'Display' ,'iter');
    end

%% PASO 6 - FUSIÓN Y FILTROS CANNY Y SOBEL

rojo=I(:,:,1);
verde=I(:,:,2);
azul=I(:,:,3);

consis=0;
reflejo=0;

wave_va=wavelet(verde,azul,escalas,consis,wavfile,reflejo);
wave=wavelet(rojo,wave_va,escalas,consis,wavfile,reflejo);
final=wave;
BW_sobel_v=edge(final,'sobel',6,'vertical');
BW_sobel_h=edge(final,'sobel',6,'horizontal');
BW_canny=edge(final,'canny',0.4,3);
BW_sobel=or(BW_sobel_v,BW_sobel_h);
mezcla_fusion=and(BW_sobel,BW_canny);

[MatrizPtosY,MatrizPtosX]=find(mezcla_fusion>0);
%% PASO 7 - LLAMADA AL ALGORITMO GENÉTICO
if elitismo~=0
    options = gaoptimset(options,'EliteCount' ,1);
    disp 'Elitismo'
    pause;
end

%options = gaoptimset(options,'Generations' ,50);

%Modifico los límites porque trabajamos al cuádruple de resolución (nivel
%0)
LI_EjeMay=LI_EjeMay*4;
LS_EjeMay=LS_EjeMay*4;
LI_EjeMen=LI_EjeMen*4;
LS_EjeMen=LS_EjeMen*4;
LI_CentroX=LI_CentroX*4;
LS_CentroX=LS_CentroX*4;
LI_CentroY=LI_CentroY*4;
LS_CentroY=LS_CentroY*4;

CoronaCteExt=CteExt;

```

```

CoronaCteInt=CteInt;

options = gaoptimset(options,'PopInitRange',...
    [LI_CentroX,LI_CentroY,LI_EjeMay,LI_EjeMen,LI_Desf;...
    LS_CentroX,LS_CentroY,LS_EjeMay,LS_EjeMen,LS_Desf]);
[CromosomaTop,FVAL,REASON,OUTPUT,FINAL_POPULATION,SCORES] =
ga(fitnessFunction,nvars,options);
% if FVAL~CTE_NO_CONVERG
%if verbose
disp('FVAL')
disp(FVAL)
disp('REASON')
disp(REASON)
disp('OUTPUT')
disp(OUTPUT)
disp('FINAL_POPULATION')
disp(FINAL_POPULATION)
disp('SCORES')
disp(SCORES)
disp('CromosomaTop')
disp(CromosomaTop)
%end

%% PASO 8 - DECODIFICACIÓN DEL CROMOSOMA

%DECODIFICA EL CROMOSOMA
CentroX_Gen=CromosomaTop(1);
CentroY_Gen=CromosomaTop(2);
EjeMay_Gen=CromosomaTop(3);
EjeMen_Gen=CromosomaTop(4);
desf_Gen=CromosomaTop(5);

%GUARDA LOS RESULTADOS EN LA ESTRUCTURA GENERAL DE RESULTADOS
%Multiplico por 2 para llevar al Subnivel 0 de la Pirámide Laplaciana
Resultados.cGenx{im}=CentroX_Gen;
Resultados.cGeny{im}=CentroY_Gen;
Resultados.EjeMay_GenAll{im}=EjeMay_Gen;
Resultados.EjeMen_GenAll{im}=EjeMen_Gen;
Resultados.desf_GenAll{im}=rad2deg(desf_Gen);%Paso a grados porque luego van a ser
utilizados desde el cálculo de la discrepancia

%% PASO 9 - PLOTEADO DE RESULTADOS

%CONSTRUYE ELIPSE GENÉTICA
figure(1);
hold on;

DibujaElipse(CentroX_Gen,CentroY_Gen,EjeMay_Gen,EjeMen_Gen,rad2deg(desf_Gen),1);%Trabaja
en Grados
hold on;
disp('Mejor Cromosoma');
disp(CromosomaTop)
if pausa pause;
end
%CREACIÓN DE LA ELIPSE GENÉTICA EN NIVELES DE GRIS
figure(3);
IG2=zeros(256,256);
imshow(IG2);

ptos=DibujaElipse(CentroX_Gen,CentroY_Gen,EjeMay_Gen,EjeMen_Gen,rad2deg(desf_Gen),1);
%dibujo en Grados
vec(:,1)=ptos(:,1);%get(h,'xdata');
vec(:,2)=ptos(:,2);%get(h,'ydata');
for i=1:size(vec,1)
    IG2(round(vec(i,2)),round(vec(i,1)))=255;
    %IG2(round(vecy),round(vecx))=255;
end

imshow(IG2);
hold on;
plot(Exp1Contorno(:,1)-suby,Exp1Contorno(:,2)-subx,'g.-');%VERDE
plot(Exp2Contorno(:,1)-suby,Exp2Contorno(:,2)-subx,'c.-');%CYAN

hold off;
if pausa
pause;

```

```
end

%Modifico los límites, deshago el cambio para que el primer genético
%funcione correctamente
LI_EjeMay=LI_EjeMay/4;
LS_EjeMay=LS_EjeMay/4;
LI_EjeMen=LI_EjeMen/4;
LS_EjeMen=LS_EjeMen/4;
LI_CentroX=LI_CentroX/4;
LS_CentroX=LS_CentroX/4;
LI_CentroY=LI_CentroY/4;
LS_CentroY=LS_CentroY/4;

end % Fin de Bucle de imágenes

save 'Resultados.mat' Resultados

end

% En esta versión se va a trabajar directamente con el perímetro de la
% elipse teniendo en cuenta un offset 0
%
% Crea una nueva población aleatoria
function Population = creapoblacion(GenomeLength,options)

    tmp_aux = gaoptimget (options, 'PopInitRange');
    tampobla = gaoptimget (options, 'PopulationSize');
    for i=1:tampobla
        Population(i,:)=((tmp_aux(2,:)-tmp_aux(1,:))*rand)+tmp_aux(1,:);
    end
end
```

Bibliografía

- [1] E.J. Carmona, M. Rincón, J. García-Feijoo and J.M. Martínez de la Casa,. «Identification of the Optic Nerve Head with Genetic Algorithms.» *Artificial Intelligence in Medicine* Vol. 43 (3), pp 243-259 (2008).
- [2] J. Lowell, A. Hunter, D. Steel, A. Basu, R. Ryder, E. Fletcher and L. Kennedy,. «Optic nerve head segmentation.» (IEEE Transaction on Medical Imaging) Vol. 23(2), pp. 256-64 (2004).
- [3] M.J. Wood, and I.C. Cox. «Computer-assisted optic nerve head assessment.» *Ophthalmic and Physiological Optic* Vol. 11 (1) , pp. 27-35. (1991).
- [4] D. Morris and C. Donnison. «Identifying the neuroretinal rim boundary using dynamic contours.» *Image and Vision Computing* Vol.17 (3-4), pp. 169-174. (1999).
- [5] F. Mendels, C. Heneghan and J. Thiran,. «Identification of the optic disk boundary in retinal images using active contours.» *In Proceedings of the Irish Machine Vision and Image Processing Conference (IMVIP '99)*. 1999.
- [6] A. Osareh, M. Mirmehdi, B. Thomas and R. Markham,. «Colour morphology and snakes for optic disc localisation.» *In Proceedings of the 6th Medical Image Understanding and Analysis Conference*. 2002.
- [7] J. Novo, M. González and J. Santos,. «Localisation of the optic disc by means of GA-optimised Topological Active Nets.» *Image and Vision Computing* Vol. 27 (10),pp. 1572-1584 (2009).
- [8] D. Welfer, J. Scharcanski, C.M. Kitamura, M.M. Dal Pizzol, L.W. Ludwig and D. Ruschel,. «Segmentation of the optic disk in color eye fundus images using an adaptive morphological approach.» *Computers in Biology and Medicine* Vol. 40, pp. 124-137 (2010).
- [9] R. Rangayyan, X. Zhu, F. Ayres and A. Ells,. «Detection of the Optic Nerve Head in Fundus Images of the Retina with Gabor Filters and Phase Portrait Analysis.» *Journal of Digital Imaging* Vol. 23(4), pp. 438-453 (2010).
- [10] X. Zhu, R. Rangayyan and A. Ells,. «Detection of the Optic Nerve Head in Fundus Images of the Retina Using the Hough Transform for Circles.» *Journal of Digital Imaging* Vol. 23(3), pp. 332-341 (2010).
- [11] E. Burt and P. Adelson. «The Laplacian Pyramid as a Compact Image Code.» (IEEE Transactions on Communications) Vol. 31(4), pp. 532-540 (1983).

- [12] M. Lalonde, M. Beaulieu and L. Gagnon,. «Fast and Robust Optic Disc Detection Using Pyramidal Decomposition and Hausdor-Based Template Matching.» (IEEE Transactions on Medical Imaging) Vol. 20 (11), pp. 1193-1200 (2001).
- [13] M. Bachiller, M. Rincón , J. Mira and J. García-Feijoo,. *An automatic system for the location of the optic nerve head from 2D images*. LNCS Volume 2085, 2001.
- [14] *DRIONS-DB: Digital Retinal Images for Optic Nerve Segmentation Database*, accessed on August 21. 2009. <http://www.ia.uned.es/personal/ejcarmona/DRIONS-DB.html> .
- [15] J. Holland. *Adaption in Natural and Artificial Systems*. 1992: Cambridge: MIT Press.
- [16] M. Shields. *Color atlas of glaucoma*. Baltimore: Williams and Wilkins, 1998.
- [17] E. Fernández. *Image Fusion*. University of Bath, 2002.