

03063  
4  
24



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

U.A.C.P. Y P. DEL C.C.H.  
I.I.M.A.S.

SISTEMA PARA EL PROCESAMIENTO Y ANALISIS  
DE IMAGENES DEL APARATO FILTRADOR DE  
*Sardinops caeruleus*

(Segmentación basada en un conjunto incertidumbre.  
Etiquetado de componentes 8-conexo y Análisis.  
diseño e implementación orientada a eventos)

**T E S I S**

QUE PARA OBTENER EL GRADO DE:  
**MAESTRO EN CIENCIAS DE LA  
C O M P U T A C I O N  
P R E S E N T A :  
GUILLERMO MARTINEZ FLORES**

MEXICO, D. F.

MARZO DE 1998

2570098

**TESIS CON  
FALLA DE ORIGEN**



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Dedicatoria

A mis padres

**Ma. Del Carmen Flores Velarde y Guillermo Martínez Luna,**

a mis hermanas

**Carmen y Judith,** y a sus alegres retoños: **Alexandra y Víctor,**

a la memoria de mis abuelos

**Magdalena Velarde Velázquez y Nicolás Flores Navarro.**

## Agradecimientos

A mis padres, por brindarme siempre su incondicional apoyo en todos los aspectos de mi vida. A mis hermanas y mis queridos sobrinos Alexandra y Víctor, gracias por sus entusiastas bienvenidas y gratas vivencias compartidas. A mis abuelos Magdalena y Nicolás, por su gran amor, ejemplo y fe, que siempre mantendré presentes.

A mi tío José y Rocío, por compartir conmigo su hogar durante mi estancia en la ciudad de México, y junto con ellos, a Alma, Coty, Gina y mi abuela Consuelo, por su tiempo y apoyo. A mi abuelo Guillermo (QPD) por tan amena convivencia en mi infancia y gratos recuerdos...

Al Todopoderoso, por iluminar siempre mi camino y el de los míos...

¡Gracias!



A la Dra. María Garza por aceptar fungir como mi Directora de tesis. A los integrantes de la Comisión Revisora: Dra. Hanna Oktaba, MC. Guadalupe Ibarguengoitia, Dr. Bernardo Shirasago y MC. Rubén Rodríguez. A todos ellos muchas gracias por sus correcciones y opiniones.



Al Dr. Gustavo Nuñez por su gran apoyo, al permitirme el acceso y uso del Laboratorio de Agentes del Centro de Investigación en Computación (CIC-IPN). Al Dr. Juan Humberto Sossa, del mismo instituto, por permitirme participar como alumno especial en la asignatura "Reconocimiento de formas y visión por computadora".

A mis compañeros del CICIMAR, que siempre me ayudaron con sus opiniones y bibliografía: MC. Gustavo de la Cruz, ING. Alberto León y MC. Héctor Villalobos.

A mis compañeros de la Maestría: Margarita, Antonio, Manuel, Bernabé, Daniel, Hector, Alberto y Ricardo, por haber formado un excelente grupo de trabajo a base de una sólida amistad.

Al IIMAS-UNAM y a la UABCS, que a través del gran entusiasmo de la Dra. Hanna Oktaba, de la MC. Guadalupe Ibarguengoitia y del Mat. Bernabé Ortiz, hicieron posible la "*Maestría Voladora*". A los profesores que siempre creyeron en este proyecto, y que lo demostraron con profesionalismo durante sus estancias en La Paz. A la Sra. Lourdes González Lora, quien tanto nos ayudó en la realización de todos nuestros trámites.

Al CONACYT por el apoyo económico brindado durante los dos años de créditos.

Al MC. Rubén Rodríguez, director del proyecto (958132) "Ecología pesquera de *Sardinops caeruleus* (sardina monterrey) en el NW de México", de la División de Estudios de Posgrado e Investigación del IPN, por las facilidades otorgadas durante el desarrollo de este trabajo en el ámbito de tal proyecto.

A la Coordinación General de Año Sabático del Instituto Politécnico Nacional, que a través del Cap. Héctor García, hizo posible la concesión de un semestre sabático para la realización de este trabajo.

# Contenido

<b>1</b>	<b>Introducción</b>	<b>1</b>
	1.1 Generalidades	1
	1.2 El procesamiento digital de imágenes (PDI)	3
	1.2.1 Elementos básicos de un sistema de PDI	4
	1.2.2 Imagen digital e histograma de la imagen	6
	1.2.3 Mejora de la imagen	6
	1.2.4 Restauración de imágenes	7
	1.2.5 Segmentación de imágenes	8
	1.2.6 Representación y descripción	10
	1.3 Aparato filtrador de la sardina monterrey	11
<b>2</b>	<b>Antecedentes</b>	<b>13</b>
	2.1 La segmentación	13
	2.2 Preparando la escena	14
<b>3</b>	<b>Segmentación basada en umbral</b>	<b>16</b>
	3.1 Conjunto incertidumbre	17
	3.2 Refinamiento de la segmentación	20
<b>4</b>	<b>Imagen binaria</b>	<b>22</b>
	4.1 Etiquetado de componentes	22
	4.1.1 Actualización de la tabla de equivalencias	25
	4.2 Área y filtro de tamaño	26
	4.3 Perímetro	27

<b>5</b>	<b>Metodología de desarrollo</b>	<b>29</b>
	5.1 Análisis	29
	5.1.1 Análisis de requerimientos	30
	5.1.2 Enfoque orientado a eventos [objetos y mecanismos]	31
	5.2 Diseño	32
	5.2.1 Ambiente de desarrollo	32
	5.2.2 Archivo de recursos	33
	5.2.3 Diseño de la interfaz con el usuario	34
	5.3 Implementación	37
	5.3.1 Estructura de la aplicación	38
	5.3.2 Estructura de las clases	39
	5.3.3 Funcionalidad	41
<b>6</b>	<b>Resultados</b>	<b>46</b>
	6.1 Descripción	46
	6.1.1 Requerimientos	47
	6.1.2 Entrada al sistema	48
	6.2 Filtros de dominio espacial	48
	6.3 Modificación del histograma	54
	6.4 Segmentación	64
	6.5 Análisis	69
	6.6 El prototipo	71
<b>7</b>	<b>Discusión</b>	<b>75</b>
	7.1 Método de segmentación propuesto	75
	7.2 El prototipo	76
<b>8</b>	<b>Conclusiones</b>	<b>77</b>
	8.1 Tendencia al futuro	78
<b>9</b>	<b>Bibliografía</b>	<b>80</b>

# Introducción

Este trabajo se enfoca en la implementación de un prototipo para el procesamiento y análisis de imágenes digitales del aparato filtrador de *Sardinops caeruleus* (sardina monterrey). Con este fin se han desarrollado los siguientes aspectos:

- Propuesta de un algoritmo de segmentación basado en un conjunto de incertidumbre y máscara binaria.
- Propuesta de un algoritmo de etiquetado de componentes 8-conexos.
- Desarrollo de algoritmos para la mejora de imágenes a partir de transformaciones al histograma, así como de filtros de dominio espacial.
- Análisis, diseño e implementación de un prototipo de procesamiento y análisis, con enfoque orientado a eventos.

A continuación ubicaremos este trabajo dentro de un sistema conceptual global, describiendo también los aspectos generales de su entorno.

## 1.1 Generalidades

El uso de técnicas para el procesamiento de imágenes digitales está en continua evolución. Durante los últimos años ha crecido significativamente el interés por la morfología, redes neuronales, sistemas inteligentes con aplicaciones en diversos campos del conocimiento, como lo son la medicina, biología, astronomía, etc., involucrando áreas de estadística, inteligencia artificial (visión por computadora y reconocimiento automático de formas), bases de datos, entre otras.

Numerosos sistemas de propósito específico han sido diseñados y puestos en el mercado para facilitar la tarea del análisis de organismos microscópicos, logrando por lo general mejoras en comparación con la labor clásica del análisis morfométrico con el uso básico de microscopios, micrómetros u otros aparatos usados tradicionalmente en el tratamiento de imágenes.

La motivación de este trabajo surge de la necesidad detectada en el Proyecto de Investigación "Ecología pesquera de *Sardinops caeruleus* (sardina monterrey) en el NW de México" [Rodríguez-Sánchez, CICIMAR-IPN, 1996-1998], en el cual es necesaria la manipulación de gran cantidad de imágenes de arcos branquiales de sardina monterrey para su análisis morfométrico.

# FALTAN PAGINAS

De la: 2

A la: 3

- **Análisis.** Obtiene medidas cuantitativas de la imagen para producir una descripción de la misma.

Dado que el prototipo involucra aspectos de estas áreas, a continuación se dará una semblanza teórica de los elementos que dieron origen a cada uno de sus módulos.

### 1.2.1 Elementos básicos de un sistema de PDI

El tratamiento digital de imágenes comprende un amplio rango de hardware, software y recursos teóricos. En la siguiente figura observamos las etapas fundamentales que son precisas para realizar el procesamiento de una imagen [González-Rafael y Woods-Richard, 1996].

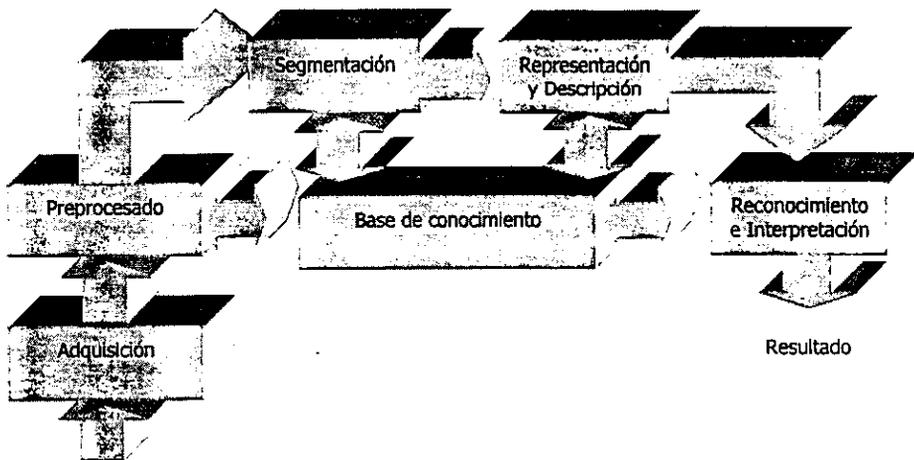


Figura 2. Elementos básicos de un sistema de PDI.

La primera etapa del proceso es la adquisición de la imagen digital. Para ello se requiere de un sensor y la posibilidad de digitalizar la señal producida por el sensor. En nuestras pruebas utilizamos un equipo de digitalización *Frame Grabber Computer Eyes/RT* de *Digital Vision, Inc.*

Una vez obtenida la imagen digital, la siguiente etapa trata del preprocesamiento de esa imagen. La función básica del preprocesamiento es la de mejorar la imagen (realce), de manera que se aumenten las posibilidades de éxito en los procesos posteriores. Aquí se implementaron técnicas para mejorar el contraste, eliminar el

ruido y aislar regiones cuya textura indica la probabilidad de definir subáreas de interés para el caso de estudio.

La siguiente etapa es la segmentación, que consiste en dividir una imagen de entrada en sus partes constituyentes u objetos. En general la segmentación automática es una de las labores más difíciles del PDI.

Como salida del proceso de segmentación habitualmente se tienen los datos del pixel bruto, que constituyen el contorno de una región, o bien todos los puntos de una región determinada. En cada caso es necesario convertir los datos a una forma adecuada para el procesamiento por computadora.

Aquí hay que determinar si los datos se van a representar como contorno o como una región completa. La representación de contorno es la adecuada cuando el interés radica en las características de forma exterior, como esquinas, inflexiones y perímetro. La representación regional es adecuada cuando el interés se centra en propiedades internas, como textura, estructuración y área. En el caso del prototipo, ambas representaciones coexisten, dadas las necesidades de cálculo.

También debe especificarse un método para describir los datos de forma que resalten los rasgos de interés. La descripción, también llamada selección de rasgos, consiste en extraer rasgos con alguna información cuantitativa de interés o que sean fundamentales para diferenciar una clase de objetos de otra.

La última etapa incluye el reconocimiento e interpretación. El reconocimiento es el proceso que asigna una etiqueta a un objeto, basándose en la información proporcionada por sus descriptores. La interpretación implica asignar significado a un conjunto de objetos reconocidos.

También existe la necesidad de contar con conocimiento previo acerca del dominio del problema, mediante una base de conocimiento que actúe como elemento de interacción con los módulos de procesamiento. Este conocimiento puede ser tan simple como detallar las regiones de una imagen donde se sabe que se ubica la información de interés, limitando así la búsqueda que ha de realizarse para hallar tal información. El conocimiento previo sobre nuestras imágenes en estudio, nos permitió hacer un refinamiento de la segmentación, tal como se detalla en el capítulo 3.

## 1.2.2 Imagen digital e histograma de la imagen

Una imagen es considerada como una función bidimensional de intensidad de luz  $f(x,y)$ , donde  $x$  e  $y$  representan coordenadas espaciales y el valor de  $f$  en un punto cualquiera  $(x,y)$  es proporcional al brillo de la imagen en ese punto.

Una imagen digital es una imagen  $f(x,y)$  que se ha discretizado tanto en las coordenadas espaciales como en el brillo. Una imagen digital puede considerarse como una matriz cuyos índices de la fila y la columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Los elementos de una distribución digital de este tipo se denominan elementos de la imagen o *pixels* (por la abreviatura inglesa de *picture elements*).

El histograma de una imagen digital con niveles de gris en el rango de  $[0, L-1]$ , donde  $L$  es el número total de niveles de gris, es una función discreta de la forma  $p(r_k) = n_k / n$ , donde  $r_k$  es el  $k$ -ésimo nivel de gris,  $n_k$  es el número de pixeles de la imagen con ese valor de gris,  $n$  es el número total de pixeles de la imagen y  $k = [0, 1, \dots, L-1]$ .

En forma general se dice que  $p(r_k)$  representa una estimación de probabilidad de la ocurrencia de un nivel de gris  $r_k$ . Una gráfica de esta función nos da una descripción global de la apariencia de la imagen.

## 1.2.3 Mejora de la imagen

El principal objetivo de las técnicas de mejora es procesar una imagen de forma que resulte más adecuada que la original para una aplicación específica. Así, por ejemplo, un método que sea muy útil para mejorar imágenes de rayos X puede no ser el ideal para mejorar imágenes de microscopio.

Los métodos implementados en la aplicación, para la mejora de imágenes son de los llamados de dominio espacial. El término dominio espacial se refiere al conjunto de pixeles que componen la imagen, y son procedimientos que operan directamente sobre los pixeles. Las funciones de procesamiento de la imagen en el dominio espacial pueden representarse como

$$g(x,y) = T[f(x,y)]$$

donde  $f(x,y)$  es la imagen de entrada,  $g(x,y)$  es la imagen procesada y  $T$  es un operador que actúa sobre  $f$ , definido en algún entorno de  $(x,y)$ .

La aproximación principal para definir un entorno alrededor de  $(x,y)$  es emplear un área de subimagen cuadrada o rectangular centrada en  $(x,y)$ , como se muestra en la siguiente figura.

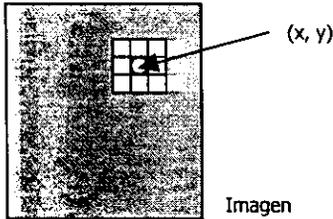


Figura 3. Entorno de 3x3 alrededor de un punto  $(x,y)$  de una imagen.

El centro de la subimagen se mueve pixel a pixel comenzando en la esquina superior izquierda y aplicando el operador en cada posición  $(x,y)$  para obtener  $g$ .

En el capítulo 6 se muestran los resultados de aplicar este tipo de filtros a una imagen de un corte transversal del cerebro.

## 1.2.4 Restauración de imágenes

Como en la mejora de imágenes, la finalidad de las técnicas de restauración es la de obtener una imagen útil para su posterior análisis. Como una diferenciación, consideramos la restauración como un proceso que trata de reconstruir o de recuperar una imagen degradada basándose en algún tipo de conocimiento a priori sobre el proceso de degradación.

Así, las técnicas de restauración se orientan hacia la introducción de modelos de degradación, que luego se aplican en sentido inverso para recuperar la imagen original. Por el contrario, las técnicas de mejora son básicamente procedimientos heurísticos diseñados para manipular la imagen aprovechando los aspectos psicofísicos del sistema visual humano.

## 1.2.5 Segmentación de imágenes

El primer paso del análisis de imágenes consiste generalmente en segmentar la imagen. La segmentación subdivide una imagen en sus partes constituyentes u objetos. El nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. Esto es, la segmentación deberá detenerse cuando los objetos de interés de una aplicación hayan sido aislados. Esta etapa del proceso determina el eventual éxito o fracaso del análisis.

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: la discontinuidad y la similitud. En la primera categoría, el método consiste en dividir una imagen basándose en los cambios bruscos de nivel de gris. Las principales áreas de interés de esta categoría son la detección de puntos aislados y la detección de líneas y bordes de una imagen. Los principales métodos de la segunda categoría están basados en la umbralización, crecimiento de región, división y fusión de regiones.

En el prototipo los algoritmos de segmentación se implementaron con base a la propiedad de similitud, usando el método de umbralizado [Otsu Nobuyuki, 1979] y el propuesto en este trabajo, que combina la definición de umbrales con el crecimiento de una región de incertidumbre basándose en una máscara binaria de convolución. Este tipo de segmentación fue elegido dadas las características de las imágenes digitalizadas, en las que se puede observar un histograma bimodal.

La umbralización es uno de los métodos más importantes de la segmentación de imágenes, cuyos fundamentos se basan en lo siguiente.

Suponiendo que el histograma que se muestra en la figura 4 corresponde a una imagen  $f(x,y)$ , compuesta de objetos luminosos sobre un fondo oscuro, de tal forma que los píxeles del objeto y del fondo tienen los niveles de gris agrupados en dos modas dominantes. Una forma de extraer los objetos del fondo es elegir un umbral  $T$  que los separe. Así, cualquier punto  $(x,y)$  para el que  $f(x,y) > T$  se denomina punto del objeto; en caso contrario, se denomina punto del fondo.

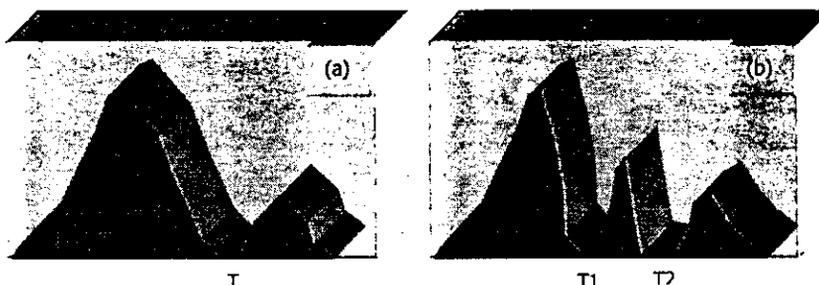


Figura 4. Histogramas de un umbral (a), y de umbrales múltiples (b).

El tipo de umbralización multinivel suele ser menos seguro que su contrapartida de nivel único, dada la dificultad de establecer umbrales múltiples que aislen efectivamente las regiones de interés, sobre todo cuando el número de modas de los histogramas correspondientes es grande.

Basándonos en lo anterior, la umbralización se puede contemplar como una operación que implica realizar comprobaciones frente a una función  $T$  de la forma:

$$T = T[x, y, p(x,y), f(x,y)]$$

Donde  $f(x,y)$  es el nivel de gris del punto  $(x,y)$  y  $p(x,y)$  representa alguna propiedad local de ese punto. Una imagen umbralizada  $g(x,y)$  se define como:

$$g(x,y) = \begin{cases} 1 & \text{si } f(x,y) > T \\ 0 & \text{si } f(x,y) \leq T \end{cases}$$

De este modo los pixeles marcados con 1 corresponden a objetos, mientras que los pixeles marcados con 0 corresponden al fondo. Cuando  $T$  depende solamente de  $f(x,y)$ , el umbral se denomina global; si  $T$  depende a la vez de  $f(x,y)$  y  $p(x,y)$ , el umbral se denomina local; si, además  $T$  depende de las coordenadas espaciales  $x$  e  $y$ , el umbral se denomina dinámico.

## 1.2.6 Representación y descripción

Después de segmentar una imagen en regiones, el conjunto resultante de píxeles segmentados se representa y describe normalmente en una forma adecuada para su posterior procesamiento por computadora.

Básicamente, el representar una región implica dos posibilidades: hacerlo en términos de sus características externas (contorno), o bien, en términos de sus características internas (píxeles que comprenden la región).

La elección de un esquema de representación es, sin embargo, solamente una parte de la tarea de hacer los datos útiles para ser procesados en una computadora. El siguiente paso consiste en describir la zona en la representación elegida. Así, una región se puede representar describiendo su contorno por medio de características tales como su longitud, la orientación de una línea recta que une los puntos extremos, y el número de conectividades.

Generalmente se elige una representación externa cuando el objetivo principal se centra en las características de forma y una representación interna cuando el principal interés se centra en las propiedades de reflectividad, tales como color y textura.

En la aplicación, se usan como descriptores de región, el perímetro (longitud del contorno) y área (definida como el número de píxeles contenidos dentro del contorno). Un empleo más frecuente de estos descriptores es en la medida de densidad de una región (o compacidad), definida como  $\text{perímetro}^2 / \text{área}$ . La compacidad es una cantidad sin dimensión (por lo tanto insensible a los cambios de escala) y es mínima para una región en forma de disco. También es insensible a la orientación.

Otras medidas utilizadas como descriptores de región incluyen la media y la mediana de los niveles de gris, los valores mínimo y máximo del nivel de gris, y el número de píxeles con valores por encima o por debajo de la media.

### 1.3 Aparato filtrador de la sardina monterrey

En este apartado revisaremos brevemente algunos aspectos generales de interés como referencia al entorno ecológico-pesquero al que se ha llevado nuestra aplicación de procesamiento de imágenes, desde la perspectiva de Rodríguez-Sánchez (1996) y Villalobos-Ortiz (1998), Maestros en Ciencias Pesqueras.

Las sardinas del género *Sardinops* han destacado entre los recursos pesqueros mayormente capturados en el ámbito mundial. Históricamente sus poblaciones han presentado fluctuaciones en su abundancia, mismas que han sido correlacionadas con la temperatura superficial del mar, entre muchos otros factores. La sardina monterrey (*Sardinops caeruleus*) habita en la corriente de California y en el noroeste de México, se ha mantenido como la especie más importante en la pesquería de pelágicos menores, en donde no ha sido ajena a variaciones en su tamaño poblacional.

Considerando los enfoques modernos para el estudio de ecosistemas marinos explotados, el determinar con precisión el tipo de alimento consumido por los elementos de una red trófica durante sus diferentes etapas de vida, reviste particular importancia. Por ello, se ha planteado la revisión de la morfología funcional del aparato filtrador de esta especie, utilizando la medición y análisis por computadora de los arcos branquiales, lo que permitirá una resolución superior a la del micrómetro ocular que tradicionalmente ha sido usado en este tipo de estudios.

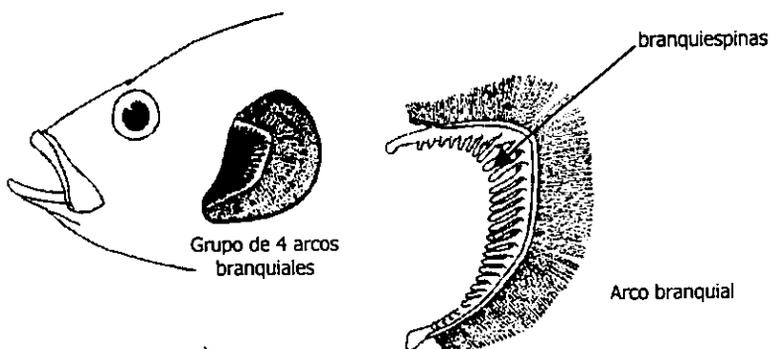


Figura 5. Aparato filtrador.

Bajo este enfoque, se pretende describir la capacidad física de las branquiespinas de *Sardinops caeruleus* para retener partículas durante su desarrollo ontogénico. Con este propósito, se realizarán mediciones en el primer arco branquial de ejemplares de sardinas colectados en el noroeste de México (Golfo de California y Bahía Vizcaíno), abarcando un intervalo de tallas de 11.6 a 29.5 mm de longitud patrón.

A partir de estas mediciones se pretende mostrar cambios en la disposición de las branquiespinas del arco durante el desarrollo ontogénico, así como variaciones en el espacio entre branquiespinas con relación a la longitud del pez. Con ello, para sardinas de diferentes longitudes, se determinará la probabilidad de retención de acuerdo al tamaño de la partícula, acoplando esta probabilidad de retención con diversos esquemas de clasificación del plancton, sugiriendo el tipo de presas más susceptibles de ser capturadas por la sardina.

Con este tipo de estudios también se puede conocer si durante el desarrollo ontogénico sucede un cambio en los hábitos alimentarios de la sardina y a partir de que tallas [Villalobos-Ortiz, 1998].

## Antecedentes

El análisis morfométrico de imágenes del aparato filtrador de peces requiere de la obtención de datos de contorno, áreas y distancias de numerosos especímenes que forman parte de una colección. El procedimiento seguido tradicionalmente consiste en tomar los especímenes de su colección, manipularlos y medirlos utilizando microscopio estereoscópico y micrómetro, para luego transcribir estas mediciones a una computadora. Esta secuencia suele ser llevada a cabo sobre un arco branquial en repetidas ocasiones por uno o más investigadores, según las medidas que deseen extraer para un estudio particular.

Esto resulta una labor intensa y rutinaria, pero que debe realizarse con mucha precisión para obtener buenos resultados. Hoy en día, los trabajos de investigación que utilizan estos métodos, exponen la dificultad enfrentada cuando es necesario obtener datos morfométricos de decenas o cientos de especímenes por área de estudio [Ponce-León, 1996 y Villalobos-Ortiz, 1998].

Muchas de estas funciones pueden ser ejecutadas mediante hardware y software a una velocidad mucho mayor, con mayor exactitud y un criterio común para procesar y presentar la información. Además, mediante un acervo de imágenes digitales clasificadas, las colecciones ictiológicas no serían manipuladas para efectuar las constantes mediciones, sino que se recurriría a la imagen almacenada. Este aspecto es importante en lo que se refiere a la conservación de colecciones, puesto que al manipular especímenes de manera frecuente, éstos se deterioran gradualmente, acortando el tiempo de vida útil de los elementos de la colección.

### 2.1 La segmentación

La segmentación de objetos o componentes de una escena constituye una de las tareas más difíciles del análisis de imágenes por computadora, razón por la cual muchas personas han abocado durante años sus estudios e investigaciones hacia mecanismos óptimos de segmentación.

El área de visión por computadora ha rendido buenos frutos en este campo en aplicaciones específicas, principalmente de robótica y automatización de procesos en la milicia y en la industria. La obtención de buenos resultados se debe en gran medida a la restricción el campo de acción del robot, así como los objetos con lo que interactúa. Así, se provee de información, por ejemplo, de los *patrones de objetos* que la máquina debe reconocer como piezas defectuosas para retirarlas de

un proceso de ensamble, o bien, de las características un vehículo en movimiento, al que debe alcanzar un misil, redefiniendo su trayectoria en tiempo real.

El establecer el entorno del problema es uno de los factores que contribuyen, en principio, a la definición de un método apropiado para una aplicación en particular. Definir un método de segmentación y análisis de imágenes de uso genérico, similar al sistema visual humano, parece estar aun muy lejos de ser factible, en virtud de la complejidad abstraída en forma tan excelente por la visión y comprensión humana. Así, mediante la delimitación del dominio del problema muchas de estas dificultades puedan ser minimizadas, opción a la que nos debemos apegar en tanto no desarrollemos sistemas de visión inteligente de uso genérico.

Gran parte del problema de segmentación proviene del principio mismo de la obtención de imágenes digitales: son representaciones en dos dimensiones de escenas en tercera dimensión. La influencia de sombras, deformaciones de rotación, traslación y escala son algunos de los factores que influyen en la obtención de nuestras representaciones digitales. Si en la etapa de adquisición no se tienen considerados estos factores, todos los procesos posteriores podrían conducir a resultados aberrantes en caso de no considerar la corrección de estas alteraciones (restauración de imágenes). De otra manera ¿Cómo podríamos asegurar que lo que hemos segmentado como objeto no es en realidad su sombra?.

## 2.2 Preparando la escena

Durante la fase de adquisición de imágenes para nuestro caso de estudio, se hicieron varias pruebas para lograr *relativamente* buenos resultados. Finalmente, como paso previo a la digitalización, con el fin de aumentar el contraste y facilitar las observaciones, los arcos branquiales fueron sometidos a un proceso de tinción, sumergiéndolos en una solución de hematoxilina concentrada durante un minuto, y enjuagando después el exceso con agua destilada. Posteriormente cada arco branquial se sujetó con alfileres entomológicos sobre una superficie que permitiera su contraste, negra en el caso de ejemplares grandes, con branquiespinas opacas, mientras que los organismos más pequeños, con branquiespinas transparentes, se usó una superficie blanca. La sujeción del arco permitió evitar en gran medida el solapamiento de las branquiespinas [Laboratorio de Pesquerías del CICIMAR-IPN].

Con el equipo de digitalización (*Frame Grabber Computer Eyes/RT de Digital Vision, Inc.*) instalado se procedió a captar las escenas, obteniendo imágenes en formato BMP.



Figura 6. Imágenes digitalizadas. Las dos primeras muestran problemas durante la adquisición. En la tercera se logra una mejora significativa con el método de tinción.

Con el método antes descrito se logró una mejora sustancial durante la etapa de adquisición, considerando el equipo con que se cuenta actualmente en el laboratorio. A partir de estas imágenes se aplicaron algoritmos para mejorarlas y segmentarlas.

Para el proceso de segmentación se implementó un algoritmo basado en el histograma y en una máscara binaria, puesto que las pruebas realizadas con el método de Otsu presentaban problemas para determinar el umbral (por el tipo de distribución de frecuencias). Así, el algoritmo propuesto toma algunos elementos de las definiciones formales de Otsu Nobuyuki (1979) para la segmentación basada en umbral, y de M. J. Duff (1986) para el uso de máscaras binarias (propuestas para el etiquetado de componentes).

## 3 Segmentación basada en umbral

Como ya se ha mencionado en el apartado de generalidades, la segmentación constituye el proceso por el cual los objetos que conforman una imagen son separados del fondo de la misma, para proceder posteriormente al etiquetado de componentes y al análisis de características.

La segmentación de una imagen digital puede ser llevada a cabo por diferentes métodos, de los cuales el de Otsu Nobuyuki es uno de los más referidos en la bibliografía, especialmente para imágenes con histograma bimodal.

Para la segmentación de imágenes del aparato filtrador de larvas de sardina se realizaron pruebas con este método, puesto que las características de la imagen (fondo claro y objeto oscuro) hacían suponer su buen desempeño.

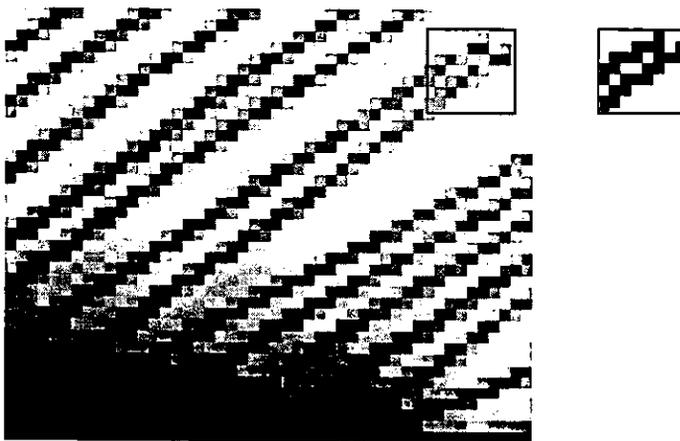


Figura 7. Porción de una imagen del arco branquial con problemas de segmentación.

El problema en esta etapa fue que muchos de los pixeles que deberían de separar dos fibras branquiales, se segmentaron como parte del objeto y no como parte del fondo de la imagen, en virtud de la similitud entre los pixeles borde y fondo, ocasionada por la cercanía entre las branquiespinas y la dificultad de mantenerlas suficientemente separadas durante la etapa de adquisición, lo que conduce a cierto nivel de incertidumbre para determinar el umbral. La figura anterior muestra una

porción de la imagen en 256 niveles de gris, en la cual se observa la situación descrita.

El método propuesto en este trabajo tiene el objetivo mejorar la respuesta ante este tipo de situaciones, con la finalidad de mantener separadas las branquiespinas durante la etapa de segmentación.

Otro problema presentado lo es la proliferación de objetos como fragmentos aislados, a causa de la localización de píxeles de alta intensidad (cercana al blanco) en algunas porciones de las branquiespinas. Para solventar esta situación es necesario reducir el número de branquiespinas aisladas durante la segmentación, integrándolas como parte del arco branquial, siempre y cuando exista una ruta de conexión como subconjunto del conjunto de incertidumbre.

En la siguiente sección se detalla el método propuesto para segmentar la imagen introduciendo este concepto. El conjunto incertidumbre contendrá los píxeles que en *primera instancia* no fueron clasificados como fondo ni como objeto.

La importancia de implantar este algoritmo radica en la necesidad de detectar y efectuar medidas que involucren el contorno del objeto (como el perímetro y compacidad), para lo cual una segmentación que permita la unión de fibras y la proliferación de objetos aislados, conduciría a mediciones erróneas.

Es claro también que el buen desempeño del algoritmo depende directamente de la imagen base, y aunque se han incluido en el sistema algunos algoritmos para mejorar la imagen, habrá casos en que aun con su uso óptimo no se lograrán mejoras sustanciales si las fallas durante la adquisición son grandes, problema que ha tratado de solventarse con el método descrito en la sección 2.2.

### 3.1 Conjunto incertidumbre

La segmentación de una imagen con base en su histograma bimodal por el método de Otsu, produce la clasificación de píxeles en dos conjuntos  $C_0$  y  $C_1$  con las siguientes distribuciones de probabilidad:

$$W_0 = \Pr(C_0) = \sum_{i=1}^k P_i = w(k)$$

$$W_1 = \Pr(C_1) = \sum_{i=k+1}^k P_i = 1 - w(k)$$

En nuestro caso de estudio, tomaremos en cuenta un tercer conjunto, que llamaremos conjunto incertidumbre. Este contendrá los píxeles cuya clasificación no ha sido determinada, de tal manera que no sabemos si corresponden al objeto o fondo. También redefinimos los dos conjuntos anteriores para ajustarlos a este planteamiento.

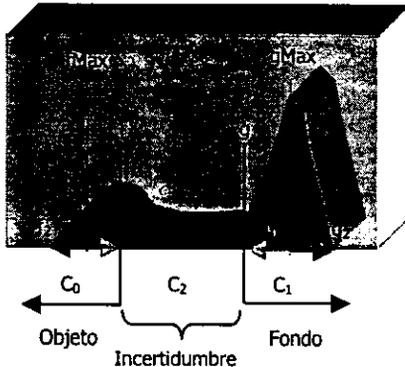


Figura 8. Clasificación de los píxeles en tres conjuntos: Objeto, Fondo e Incertidumbre.

Como se puede observar en la figura anterior, existen dos clases bien definidas  $C_0$  y  $C_1$ . Consideremos también:

- $fMax$  = Nivel de gris en la frecuencia máxima en la clase  $C_0$
- $f_2$  = Mínimo nivel de gris en la clase  $C_0$
- $gMax$  = Nivel de gris en la frecuencia máxima en la clase  $C_1$
- $g_2$  = Máximo nivel de gris en la clase  $C_1$

y calculamos:

$$f = fMax + (fMax - f_2)$$

$$g = gMax - (g_2 - fMax)$$

El intervalo  $f, g$  calculado contiene los píxeles de la clase  $C_2$ , que es nuestro conjunto incertidumbre. Puesto que la distribución de píxeles en este intervalo es muy homogénea, es difícil establecer la división de clases calculando un umbral.

A partir de la definición anterior obtenemos las siguientes distribuciones de probabilidad para las clases:

$$W_0 = \Pr(C_0) = \sum_{i=f_2}^f P_i = 1 - (w_1 + w_2)$$

$$W_1 = \Pr(C_1) = \sum_{i=g}^{g_2} P_i = 1 - (w_0 + w_2)$$

$$W_2 = \Pr(C_2) = \sum_{i=f+1}^{g-1} P_i = 1 - (w_0 + w_1)$$

Bajo esta consideración en un caso típico, el conjunto incertidumbre  $C_2$  será el de menor cardinalidad, pues aunque el rango  $f, g$  en el histograma puede ser relativamente amplio, la distribución de probabilidad en el mismo será mínima en comparación con  $W_0$  y  $W_1$ . Es por ello que el algoritmo propuesto revisa exhaustivamente los pixeles incertidumbre en una *segunda oportunidad* de clasificación.

Así, durante el proceso de binarización de la imagen se lleva a cabo el siguiente procedimiento, genéricamente descrito, para obtener el conjunto incertidumbre:

1. Cargar la imagen a memoria (256 niveles de gris)
2. Determinar los puntos  $f$  y  $g$  en el histograma de la imagen
3. Recorrer la imagen de arriba a abajo y de izquierda a derecha
4. Si el valor del pixel visitado es menor o igual a  $f$   
    Valor de pixel = Objeto  
    De lo contrario  
        Si el valor del pixel es mayor o igual a  $g$   
            Valor de pixel = Fondo  
        De lo contrario  
            ( $f < \text{Valor del pixel} < g$ )  
            Llenar arreglo Fuzzy (coordenadas del pixel incertidumbre)  
    Fin Si

Con lo anterior se establecen las regiones fondo y objeto con mayor certeza de una buena clasificación, mediante la ubicación de los puntos  $f$  y  $g$ . También se ha obtenido un arreglo que contiene los pixeles no clasificados, almacenados según su orden de aparición en el barrido de la imagen.

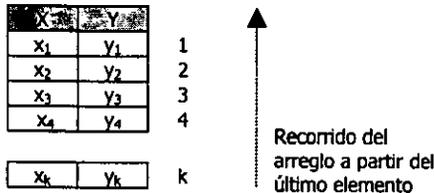
De esta manera el arreglo *Fuzzy* (Clase  $C_2$ ) contiene en sus primeros elementos los pixeles con posición hacia la esquina superior izquierda de la imagen. Conforme el arreglo es llenado, los pixeles contenidos en éste corresponderán a los de la esquina inferior derecha, en un recorrido de *filas primero*; aspecto importante en la segunda etapa de este desarrollo, misma que se detalla a continuación.

### 3.2 Refinamiento de la segmentación

Una vez obtenido el conjunto de pixeles incertidumbre es necesario definir una heurística que permita el refinamiento de la segmentación mediante la agregación de nuevos pixeles objeto o fondo.

Con este fin se ha propuesto la máscara que se muestra en la siguiente figura, construida con base en la morfología de las branquiespinas.

Conjunto Incertidumbre



↓

Se aplica la siguiente máscara para determinar si el punto corresponde a un objeto

0	0	*	*	*
0	*	**	*	0
0	*		*	0
*	1	1	*	0
*	*	*	0	0

Máscara binaria

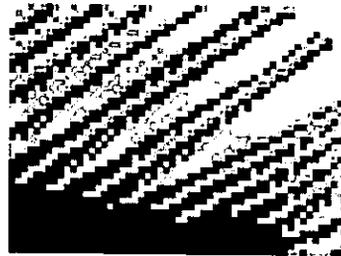
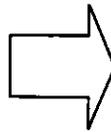


Imagen en proceso de refinamiento

Figura 9. Diagrama conceptual del refinamiento de la segmentación.

Los pixeles del conjunto  $C_2$  son extraídos del arreglo a partir del ultimo elemento, con lo cual nos aseguramos de hacer una revisión a partir de los pixeles objeto que se encuentran en la parte inferior de la imagen. A cada posición  $(x_i, y_i)$ ,  $i=1,2,\dots,k$ , se le aplica la máscara binaria para asegurar que la inserción del nuevo pixel objeto no ocasionaría la unión de fibras. Este crecimiento de región permite la eventual incorporación de pixeles aislados al cuerpo del objeto.

Este algoritmo permite el crecimiento de regiones basándonos en el conocimiento previo acerca del tipo de imágenes a segmentar, es por ello que lo consideramos propósito específico en la implementación del prototipo.

Como proceso posterior a la segmentación el tratamiento de la imagen binaria elimina los píxeles que fueron clasificados como objeto y que aun después del refinamiento permanecen como pequeñas áreas aisladas. El algoritmo no ha permitido su incorporación al objeto de interés porque de hacerlo se ocasionarían efectos como la unión de fibras branquiales, y en general, la proliferación de huecos.

# 4

## La imagen binaria

Como resultado del método de segmentación descrito en el apartado anterior, obtenemos una imagen binaria, la cual requiere de un procesamiento adicional con la finalidad de extraer algunas características de los objetos contenidos en la misma.

Los píxeles de una imagen pertenecientes a un objeto, deben ser etiquetados. El primer paso, es entonces, etiquetar los componentes y luego agrupar los píxeles etiquetados con el mismo valor para formar objetos. Aquí asumimos que estos píxeles están *especialmente cercanos*, para que el algoritmo los agrupe en un componente [Jain-Ramesh, 1995].

Además de aplicar este etiquetado, también es necesario calcular el área del o los objetos en la imagen, para hacer entonces un filtrado de tamaño sobre la imagen binaria, y por último el cálculo del perímetro mediante un recorrido 8-conexo sobre los píxeles borde de los componentes. La implementación de estos procesos se detalla a continuación.

### 4.1 Etiquetado de componentes

El algoritmo desarrollado para este fin se basa en el método propuesto por Jain-Ramesh (1995) para el etiquetado secuencial de componentes 4-conectados. Aquí consideramos un recorrido sobre 4 vecinos 8-conectados, que hace eficiente el manejo de la tabla de equivalencias al minimizar la generación de etiquetas nuevas y fomentar las equivalentes.

El proceso de etiquetado consiste en asignar un valor a cada grupo de *píxeles objeto conectados*, los cuales serán considerados como componentes de la imagen segmentada.

Idealmente, la segmentación en este caso de estudio, debería generar un solo objeto correspondiendo al arco branquial y branquiespinas en una *sola pieza*, sin embargo en la practica esto no sucede, principalmente por las siguientes razones:

- Ruido en la imagen generado durante la adquisición.
- El algoritmo de refinamiento no ha logrado integrar todos los objetos como parte del arco branquial.
- La imagen corresponde a una porción del arco branquial, lo cual ocasiona que las branquiespinas queden separadas del arco branquial (figura 7).

Un factor indispensable para determinar la agrupación de píxeles en un componente, lo es el tipo de vecindad de este píxel con respecto a los de su entorno. En una imagen digital representada sobre una retícula, un píxel tiene una frontera común con cuatro píxeles y comparten una esquina con cuatro píxeles adicionales. Decimos que dos píxeles son 4-vecinos si estos comparten una frontera común. De manera similar, dos píxeles son 8-vecinos si estos comparten al menos una esquina.

Así, el píxel en la posición  $[i, j]$  tiene 4-vecinos  $[i+1, j]$ ,  $[i-1, j]$ ,  $[i, j+1]$  y  $[i, j-1]$ . Los 8-vecinos de ese píxel incluye los 4-vecinos, más  $[i+1, j+1]$ ,  $[i+1, j-1]$ ,  $[i-1, j+1]$  y  $[i-1, j-1]$ . Se dice que un píxel es 4-conexo a sus 4-vecinos y 8-conexo a sus 8-vecinos.

En el prototipo desarrollado, el algoritmo propuesto encuentra componentes 8-conectados en la imagen binaria y asigna una etiqueta única a los puntos pertenecientes al mismo componente para posteriormente calcular características tales como tamaño, perímetro y compacidad de los mismos. El algoritmo es secuencial por naturaleza, pues la operación de encontrar componentes conectados es global. Bajo estas consideraciones implementamos la siguiente rutina una vez cargada la imagen binaria en memoria.

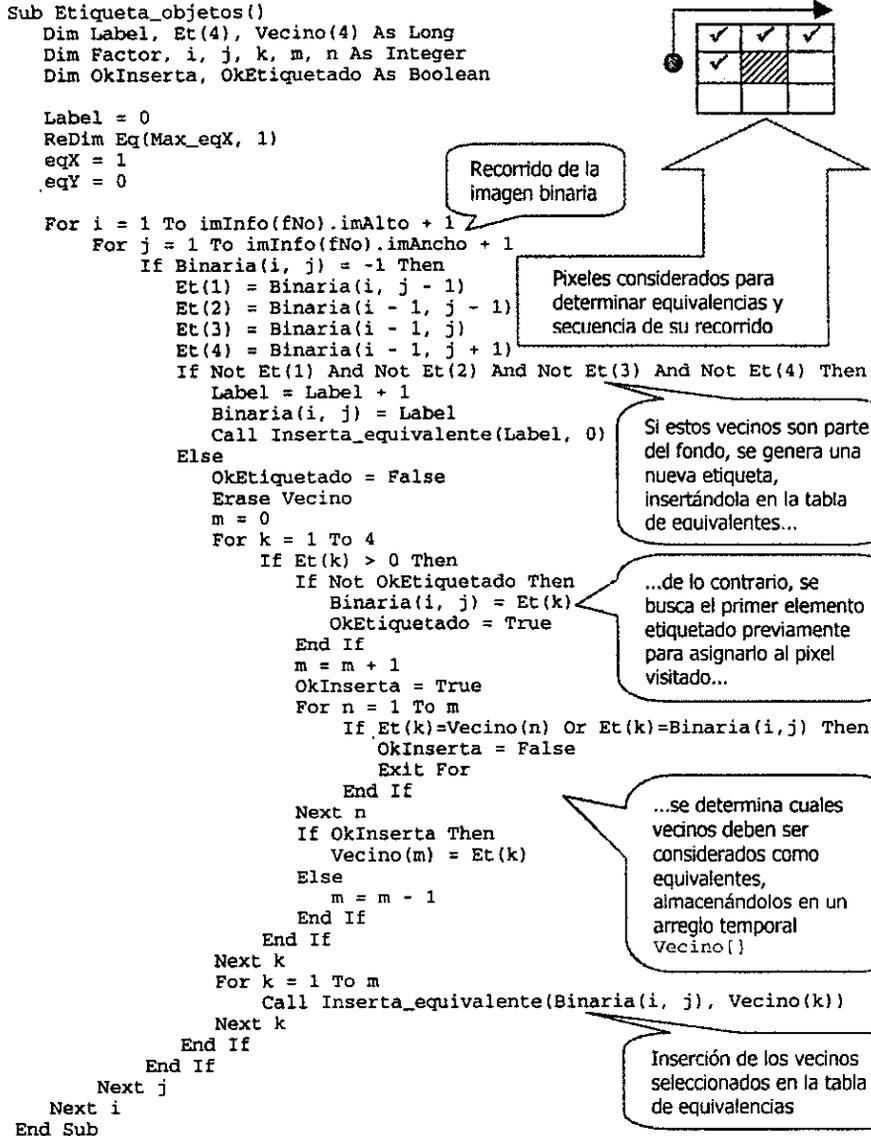


Figura 10. Algoritmo de etiquetado de componentes 8-conexo.

El algoritmo completo implica dos recorridos por la imagen, uno para etiquetar parcialmente los componentes (y llenar la una tabla de equivalencias  $E_q[x, y]$ ) y el segundo para remplazar las etiquetas por su *mínimo equivalente*, tomando como base la tabla  $E_q[x, y]$ . Como resultado de este primer recorrido se tendrá entonces, una imagen parcialmente etiquetada y la tabla de equivalencias actualizada.

### 4.1.1 Actualización de la tabla de equivalencias

El etiquetado de componentes 8-conexo se implementó utilizando una tabla de equivalencias bidimensional  $E_q[x, y]$ , cuyo número de filas y columnas es modificado dinámicamente dependiendo de la localización de nuevas etiquetas o de equivalencias (algoritmo de etiquetado listado en la figura anterior).

A continuación se muestra un ejemplo de la secuencia de expansión de la tabla ocasionado por la llamada a la función `Inserta_equivalente(Et1, Et2)`.

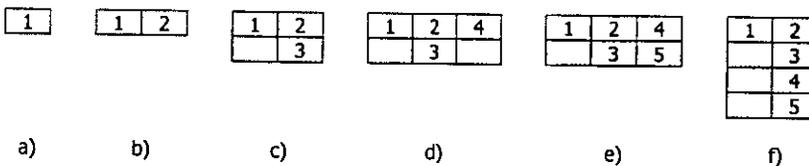


Figura 11. Expansión de la tabla de equivalencias.

Al encontrar la primera etiqueta el arreglo  $E_q[x, y]$  contiene únicamente una celda (figura 11.a). Conforme se encuentran etiquetas nuevas, el arreglo crece horizontalmente, agregando una columna por cada etiqueta nueva (figura 11.b).

Cuando una equivalencia es detectada, la etiqueta es ubicada en la columna donde se encuentra su equivalente. La figura 11.c, representa el caso de una equivalencia de 2 con 3.

Las siguientes representaciones de la figura 11 muestran, primero la inserción de 4 como una nueva etiqueta (figura 11.d), y posteriormente la inserción de 5 como equivalente 4 (figura 11.e).

Al recorrer la imagen se pueden detectar nuevas equivalencias en la tabla. Por ejemplo, teniendo la representación de la figura 11.e, pudiéramos encontrar que 5 es equivalente con 2, lo cual ocasiona la adición de las etiquetas de la tercera columna a las de la segunda, con la subsecuente eliminación de la columna trasladada (figura 11.f).

Cabe señalar que si existieran más columnas a la derecha de la trasladada, éstas tendrían que recorrerse una columna, eliminando la última.

Al final de este recorrido por la imagen, los *índices* del arreglo  $E_{q}[]$  en las columnas resultantes corresponden a las etiquetas que deben reemplazar sus equivalentes en el segundo recorrido, en donde además se calcula el área para cada objeto haciendo un conteo de los pixeles con etiquetas comunes.

## 4.2 Área y filtro de tamaño

La propiedad de área es utilizada como un parámetro que ayuda a determinar los *componentes parásitos* de la imagen. Así, todos los objetos con área menor a  $\beta$  serán eliminados de la escena. A esta operación se le llama *filtro de tamaño* [Jain- Ramesh, 1995].

Un filtro de tamaño debe considerar un valor de  $\beta$  tal que la probabilidad de afectar objetos de interés sea mínima. Puesto que las imágenes analizadas en este caso de estudio pueden pertenecer a porciones de branquiespinas, el tamaño de  $\beta$  puede variar con cada imagen, por lo tanto su valor es establecido después de calcular las áreas para todos los objetos. Así,  $\beta = \text{MAX}(\text{AreaObjeto}[])$ , donde  $\text{AreaObjeto}[]$  es un arreglo que contiene los valores de área para cada objeto etiquetado en la fase anterior.

Al Aplicar este filtro en la siguiente imagen binaria, los objetos sombreados serían convertidos en parte del fondo de la imagen.

	1	1																	
		1				1												1	
1					1					1								1	
			1	1			1	1											
		1	1			1	1			1	1								
	1	1			1	1				1	1								1
1	1			1	1			1	1									1	1
1	1	1	1	1	1			1	1									1	1
1	1	1	1	1	1			1	1									1	1
1	1	1	1	1	1	1	1			1	1							1	1
1	1	1	1	1	1	1	1	1	1	1								1	1
1	1	1	1	1	1	1	1	1	1	1	1							1	1
1	1	1	1	1	1	1	1	1	1	1	1	1						1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1					1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1				1	1

Figura 12. Componentes eliminados por el filtro de tamaño en una porción de la branquiespina.

### 4.3 Perímetro

Una vez que se han etiquetado los componentes, calculado áreas y eliminado los pixeles considerados como ruido, el siguiente paso es calcular el perímetro del objeto contenido en la imagen. Este proceso consiste en formar una *ruta* o *camino* en torno a cada objeto, contabilizando el número de pixeles que lo conforman.

La ruta del pixel en la posición  $[i_0, j_0]$  al pixel en  $[i_n, j_n]$  es la secuencia de pixeles  $[i_0, j_0], [i_1, j_1], [i_2, j_2], \dots, [i_n, j_n]$ , tal que el pixel en  $[i_k, j_k]$  es un vecino del pixel en  $[i_{k+1}, j_{k+1}]$  para toda  $0 \leq k \leq n - 1$ . Si la relación de vecindad usa 4-conexión, entonces la ruta es una 4-ruta, para 8-conexión será una 8-ruta.

Por lo anterior, nuevamente es importante determinar el tipo de conectividad a considerar antes de llevar a cabo el proceso. Por las características del objeto a medir (conexión diagonal de pixeles en branquiespinas), consideramos un recorrido 8-ruta, para lo cual utilizamos el siguiente arreglo  $S[]$  que almacena las coordenadas de los 8-vecinos de un pixel en la posición  $[i, j]$ .

```

Sub Calcula_perimetro(i, j, Idx)
    Dim Etiqueta, OrigenX, OrigenY As Integer
    Etiqueta = Binaria(i, j)
    OrigenX = i
    OrigenY = j

    Do While True
        f(fNo).PSet (j - 1, i - 1), QBColor(0)
        f(fNo).Refresh
        Do While Binaria(i + S(Idx, 1), j + S(Idx, 2)) <> Etiqueta
            Idx = Idx + 1
            If Idx = 9 Then Idx = 1

            Loop
            i = i + S(Idx, 1)
            j = j + S(Idx, 2)
            If i = OrigenX And j = OrigenY Then
                Exit Do
            End If
            Idx = Idx - 3
            If Idx <= 0 Then Idx = 8
            Perimetro(Etiqueta) = Perimetro(Etiqueta) + 1
        Loop
    End Sub
    
```

i	j
0	-1
-1	-1
-1	0
-1	1
0	1
1	1
1	0
1	-1

Figura 13. Algoritmo para recorrido de perímetro 8-conexo.

Los parámetros del procedimiento son las coordenadas *i*, *j* del pixel origen, e *Idx* el índice del arreglo *s* que es controlado dentro del ciclo más interno. *OrigenX* y *OrigenY*, almacenan el punto de inicio de la ruta. Cuando las coordenadas del pixel visitado corresponden con el punto de inicio el proceso termina, dejando en el arreglo *Perimetro[]* el número de pixeles contabilizados para el objeto con etiqueta determinada por la variable *Etiqueta*.

Hasta aquí se han implementado los algoritmos de etiquetado, filtro de tamaño, cálculo de área y perímetro de objetos con algunas variantes que favorecen el procesamiento del tipo de imágenes referidas en este trabajo, al que además se integran algunos algoritmos para el mejoramiento de tales imágenes.

Estos algoritmos de mejora no se describen detalladamente en este trabajo, sin embargo, en el capítulo 1 se encuentra una semblanza de estos métodos. Su explicación a detalle podrá encontrarse en [1] [2] [3] [4].

Tanto los algoritmos aquí descritos, así como de aquellos cuya finalidad es la mejora de la imagen, están considerados en el desarrollo del sistema prototipo de PDI, cuyo análisis, diseño e implementación orientada a eventos será tema del siguiente capítulo.



# Metodología de desarrollo

En esta etapa desarrollaremos el análisis y diseño orientado a eventos del sistema de PDI de propósito específico. En primera instancia, definiremos los requerimientos del sistema para determinar entonces el ambiente de desarrollo, diseño de las clases e interfaz con el usuario que permita una interacción sencilla con las imágenes a procesar y a analizar.

Así, la idea fundamental en esta fase, es la conceptualización de los elementos y mecanismos que intervendrán en la aplicación y de la interrelación entre ellos, hasta llegar a la implementación de los algoritmos necesarios para la extracción de características en imágenes de branquiespinas de larvas de peces, poniendo especial atención en los mecanismos provistos para la manipulación de objetos.

## 5.1 Análisis

Como primer paso en el desarrollo del sistema se hizo el análisis de requerimientos para el diseño mediante el examen de los procedimientos que se llevan a cabo manualmente para la obtención de datos morfométricos del aparato filtrador de *Sardinops caeruleus*, poniendo especial atención en los siguientes aspectos:

- Análisis de los especímenes de la colección (volumen y manejo de claves de identificación, tasa de crecimiento de la colección)
- Obtención de imágenes digitales de los especímenes
- Requerimientos para el tratamiento de las imágenes
- Necesidades de características a extraer
- Requerimientos de información para consulta y análisis
- Estudio de factibilidad de implementación
- Posibilidad de integrar nuevas funcionalidades al sistema

Con lo anterior se obtuvo una visión de los alcances del sistema, los procedimientos que se involucrarían (módulos genéricos del sistema), así como el tipo de representaciones de los datos gráficos, numéricos y textuales.

### 5.1.1 Análisis de requerimientos

Las imágenes insumo para el prototipo estarán contenidas en una colección inicial de 368 archivos *bitmap* en 256 tonos de gris de 300x300 pixeles aproximadamente, aunque se debe considerar tamaño variable, pues en ocasiones se trabajará con porciones de imágenes.

Las funciones básicas del prototipo serán por lo tanto cargar la imagen a memoria, desplegarla y construir su histograma, considerando que varias imágenes pueden ser cargadas a memoria durante la ejecución, por lo que se debe prever el manejo de múltiples imágenes e histogramas.

Una vez seleccionada la imagen de interés se estará en posibilidad de aplicar transformaciones con la finalidad de mejorarla. Estas alteraciones a la imagen deben producir imágenes nuevas (no modificar sobre la original), las cuales podrán ser comparadas con su antecesor para verificar los resultados. El llevar a cabo este preprocesamiento de mejora de la imagen incrementa las posibilidades de éxito en los procesos de segmentación y análisis posteriores.

Es posible definir transformaciones a partir del histograma generado para cada imagen cargada a memoria, aplicando funciones que modifiquen la distribución de frecuencia de los niveles de gris en la imagen. Los tipos de transformaciones al histograma pueden efectuarse basándose en funciones lineales, logarítmicas, senoidales, exponenciales o negativas.

Además de las transformaciones al histograma, también es posible la aplicación de filtros de dominio espacial, ya sea para atenuar las señales de alta frecuencia en la imagen (filtros pasa baja), o bien para resaltarlos (filtros pasa alta).

Una función de suma importancia en el sistema es la segmentación de la imagen, la cual produce una imagen binaria conteniendo las regiones que forman parte del o los objetos. Esta segmentación puede ser automática (Otsu o con región de incertidumbre) o asistida por el usuario (umbral definido por el usuario).

Una vez generadas las regiones objeto(s) y fondo, el siguiente paso consiste en etiquetar estos componentes. Durante este proceso es posible extraer algunos rasgos de los objetos que se presentan en la imagen, por lo cual en esta etapa se debe implementar un método de visualización y almacenado de estos rasgos. Puesto que cada transformación sobre una imagen produce una nueva, se deben de considerar métodos que permitan la administración de memoria disponible.

## 5.1.2 Enfoque orientado a eventos [objetos y mecanismos]

Muchos de los sistemas operativos recientes disponen de una Interfaz Gráfica del Usuario (GUI). Uno de los beneficios de la GUI es que provee un mecanismo consistente de interacción entre el usuario y las aplicaciones. De esta manera, una vez que el usuario se familiariza con el uso de objetos de interacción (botones, menús, cuadros de diálogo, etc.), la operación de diferentes aplicaciones es mucho más fácil. Otro beneficio, es que las GUIs responden a la interacción del usuario (eventos), en una secuencia definida por éste, en lugar de seguir una secuencia de operaciones definidas por el programa. Las aplicaciones desarrolladas bajo esta perspectiva son llamadas orientadas a eventos.

Las aplicaciones orientadas a eventos facilitan al usuario su interacción, pero en cierta forma, dificultan el diseño al programador, puesto que la funcionalidad del sistema no está determinada de manera intrínseca. Diferentes secciones de código pueden ser ejecutadas en diferente orden, dependiendo de los requerimientos del usuario. Aunque este tipo de aplicaciones pueden ser escritas en lenguajes de programación clásicos, serían difíciles de manejar mientras más grande sea el sistema. Diferentes eventos pueden cambiar partes del programa a diferentes tiempos, haciendo difícil el seguimiento de estos cambios (*estado de los objetos*).

Afortunadamente, la programación orientada a eventos se desarrolló en paralelo con las GUIs. En un sistema orientado a eventos, cada objeto tiene partes que representan su estado actual. Los eventos pueden ocasionar cambios en dicho estado. De esta manera, si nos aseguramos que los cambios en un objeto serán reflejados en la GUI, no tendremos que preocuparnos de modificar el sistema completo, sino sólo unos cuantos objetos. Si nos aseguramos que estos cambios pueden ser ocasionados sólo como resultado de métodos de éste tipo, podremos localizar el código que debemos considerar. Estableciendo de esta forma la propiedad de *encapsulación*.

Como plataforma de desarrollo para el prototipo se eligió Microsoft Windows 95™, básicamente por las siguientes razones: por ser un sistema operativo de interfaz gráfica, lo que en primera instancia favorece el tipo de aplicación; además, su capacidad de correr aplicaciones de 32 bits lo deja en una posición favorable para el tipo de operaciones de despliegue y procesamiento; finalmente, su disponibilidad en prácticamente en cualquier institución favorece la portabilidad de las aplicaciones.

En cuanto al lenguaje de programación a elegir, éste debería obtener el mayor provecho de la arquitectura del sistema operativo, una de las mejores opciones lo fue a nuestra consideración Visual Basic Pro 4.0 de Microsoft (a 32 bits), dada su orientación a eventos, la posibilidad de implementar *clases* y *herencia*, así como la

posibilidad de interactuar con las Bibliotecas de Enlace Dinámico (DLL) provistas por Windows 95™ y NT™.

## 5.2 Diseño

### 5.2.1 Ambiente de desarrollo

La siguiente figura muestra de manera conceptual el ambiente de desarrollo de la aplicación, sus componentes principales y la interacción entre ellos. En la parte superior se representan las Bibliotecas de Enlace Dinámico (DLL) que son provistas por el sistema operativo y por el lenguaje de programación, además de un tipo especial de DLLs definidas por el usuario, llamadas *Archivos de recursos*.

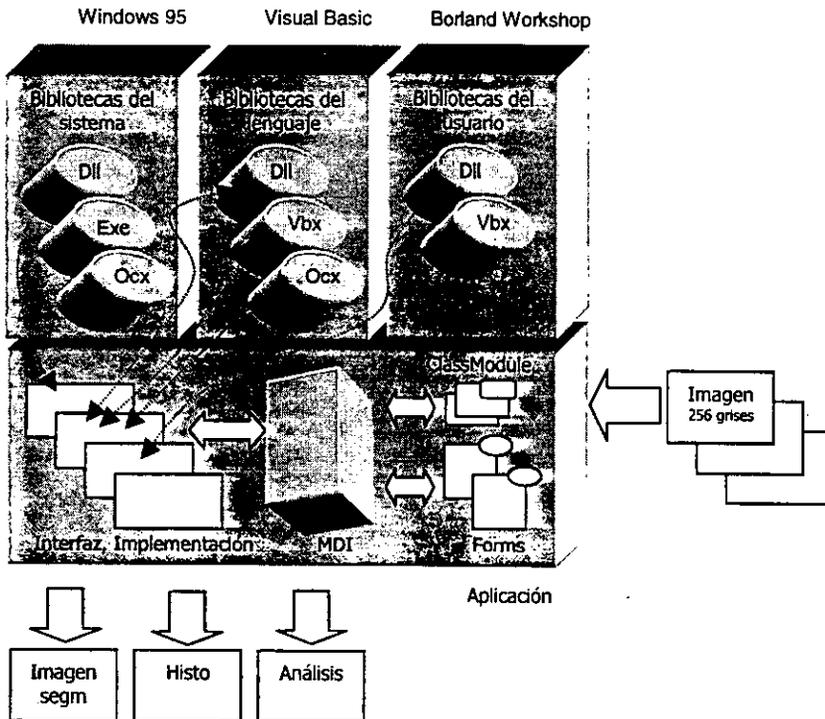


Figura 14. Ambiente de desarrollo.

El uso de estos *Archivos de recursos* proporciona gran flexibilidad a la aplicación para el manejo de algunos objetos de uso común en ambientes gráficos, lo que nos lleva a la creación de *bibliotecas de objetos reutilizables*. Esta flexibilidad radica en la posibilidad de generar, por ejemplo, bibliotecas de recursos para varios idiomas, consiguiendo con ello, cambiar el idioma de la aplicación sin la necesidad de recompilar el código.

El archivo de recursos del prototipo fue creado con el *Editor de Recursos Borland WorkShop Versión 4.5*, con el cual se ha generado una biblioteca de enlace dinámico.

## 5.2.2 Archivo de recursos

Algunos objetos usados en ambientes gráficos de desarrollo requieren gran cantidad de datos asociados a estos. Por ejemplo, los iconos de 16 colores de 32x32 pixeles requieren de 512 bytes para un *bitmap* independiente de contexto (DIB), 128 bytes adicionales para la máscara monocroma y un espacio adicional para la información de encabezado [Appleman-Daniel, 1996].

Otros objetos que pueden tomar grandes cantidades de información lo son los *bitmaps* (BMP), fuentes, *strings* (TXT) y *cursores* (CUR).

En muchos casos es deseable incluir uno o más de estos objetos en una biblioteca, lo cual resulta un problema interesante de resolver con programación estándar. En C o Pascal es posible codificarlos creando estructuras de datos estáticas que contengan la información binaria del objeto. Sin embargo, esto resulta altamente inconveniente por la dificultad de la codificación y modificación de los objetos contenidos en tales estructuras.

En el desarrollo del prototipo se solventa este problema mediante un mecanismo provisto por Windows llamado *Archivo de recursos*. Un recurso es un bloque de datos que representa objetos mediante una tabla que lista los recursos en el archivo. Estos recursos son cargados a una aplicación en particular mediante llamadas a funciones API (Interfaz para Programación de Aplicaciones) de Windows, cuyas declaraciones deben ser especificadas en alguno de los módulos (.BAS) del sistema (figura 14).

### 5.2.3 Diseño de la interfaz con el usuario

En esta etapa se definieron las opciones y se diseñaron los mecanismos con los que el usuario interactuaría con la aplicación, proveyendo los elementos que se mencionan a continuación.

#### Ventanas

El prototipo despliega imágenes, resultados de mediciones, histogramas, etc., en ventanas. Estas ventanas siguen las convenciones normales del ambiente Microsoft Windows™ en cuanto a formato y botones estándar. Las mediciones son desplegadas en la *Ventana de resultados*, así como algunos elementos de análisis, como el histograma, proporcionando información adicional sobre la imagen en ventanas separadas.

#### Barra de estado

La *Barra de estado* se localiza en la parte inferior de la ventana principal. En ésta son desplegados datos de interés para el usuario, como lo son las coordenadas de los píxeles por los que se desplaza el cursor sobre una imagen, nivel de gris, tamaño de la imagen, avisos al usuario, etc. Todo esto dependiendo del contexto activo.



Figura 15. Barra de estado.

#### Barra de herramientas

La *Barra de herramientas* contiene los botones que activan las opciones más comunes del sistema. Conforme el apuntador del mouse es desplazado sobre los botones de la *Barra de herramientas*, aparecerá una ayuda de contexto en la *Barra de estado* describiendo brevemente la función del botón.



Figura 16. Barra de herramientas.

Los botones de la *Barra de herramientas* se han clasificado en dos grupos. El primero contiene las opciones más comunes, al igual que la mayoría del software diseñado para Windows. El segundo grupo corresponde a operaciones específicas de procesamiento de imágenes, cuya funcionalidad se explica en el capítulo 6.



#### Abrir

Muestra un *cuadro de diálogo* en el que el usuario seleccionará unidad, directorio y nombre del archivo de imagen a abrir. El único formato soportado por el prototipo es BMP.

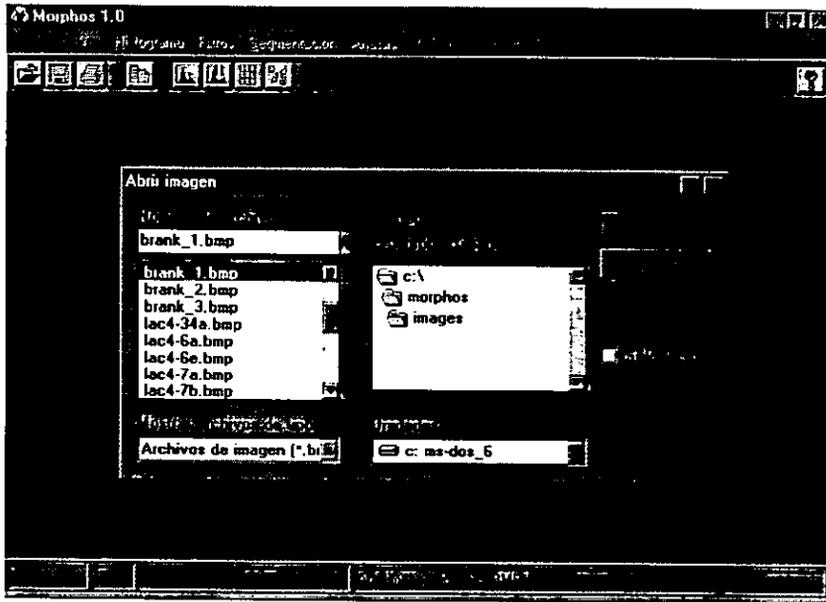


Figura 17. Diálogo Abrir imagen.



#### Guardar como

Muestra un *cuadro de diálogo* en el que el usuario especifica el nombre de la imagen a guardar con la ruta completa. El comando guardar permite almacenar resultados, histogramas e imágenes. Los histogramas e imágenes son almacenados en formato BMP, mientras que los resultados son almacenados en archivos ASCII (DAT).



#### Imprimir

Este comando permite imprimir imágenes, resultados e histogramas, dependiendo de la ventana activa, según la configuración establecida como predeterminada desde el sistema operativo.



### Copiar

Este botón le permite copiar el contenido de la ventana activa al portapapeles, y es usado para transferir imágenes, histogramas o resultados a cualquier aplicación que soporte el comando *Pegar*.

## Imágenes

El prototipo es diseñado como una aplicación con Interfaz de Múltiples Documentos (MDI), lo cual permite que múltiples imágenes sean desplegadas a la vez. La ventana activa tendrá la barra de título en color, todas las demás ventanas tendrán sus barras de títulos opacas. Todas las operaciones serán efectuadas en relación con la imagen activa.

## Tablas de color

Las imágenes de ocho bits son desplegadas usando una tabla de color (LUT por las siglas en inglés de *Look Up Table*), la cual determina el color a ser desplegado para cada uno de los 256 valores de pixel disponibles. El prototipo provee el soporte para modificar la tabla de color usada para una imagen mediante modificaciones al histograma. Usando las tablas de color apropiadamente podemos mejorar imágenes agregando tonos de gris, o ajustando el mapeo a la escala de grises para realzar ciertos rasgos.

## Ventana de resultados

La opción de análisis genera datos, los cuales son enviados a la *Ventana de resultados*. Por omisión la ventana es visible, aunque puede ocultarse ya sea cerrándola o bien usando la opción *Ventana de resultados* del menú *Ver*.

## Menú pop-down

Proporciona un método adicional para activar las opciones del sistema, algunas de las cuales pueden seleccionarse desde la *Barra de herramientas*.



Figura 18. Menú pop-down.

## 5.3 Implementación

Intrínsecamente los elementos que forman una aplicación Visual Basic son considerados como objetos formales con propiedades y métodos que pueden ser heredados a nuevas instancias.

En la programación orientada a eventos cada objeto dentro de una clase tiene cierto comportamiento determinado por acciones que se efectúan sobre éste (eventos). Así, es el usuario quien determina las acciones que cada objeto disponible en el sistema realizará, de acuerdo con los eventos que éste suministre (como arrastrar el cursor, hacer clic en determinadas áreas de una ventana, etc.).

Esta característica es sumamente útil para el tipo de aplicación, puesto que la conducta del sistema estará guiada por el usuario quien decidirá el tipo de transformación que se aplicará a la imagen en función de las mediciones que se requieran.

Los objetos ventana (*Forms*) encapsulan datos y métodos en declaraciones privadas, visibles solo en su entorno. También es posible hacer visibles estos elementos a toda la aplicación declarándolos públicamente.

Las declaraciones de la interfaz a los *recursos* son visibles globalmente en la aplicación, al igual que los tipos, variables y constantes definidas en los módulos .BAS.

La creación de clases dentro de un proyecto Visual Basic (VBP) es posible mediante la Opción *Class Module* del menú *Insert*. Una vez creados los *ClassModule* necesarios para una aplicación, se procede a implantar su interfaz y funcionalidad.

Dependiendo del tipo de implementación, las clases declaradas en la forma antes descrita, pueden ser instanciadas sólo por la aplicación en la cual fueron definidas, o bien, ser expuestas a su uso por otras aplicaciones (creación de servidores OLE [*Object Link and Embedding*]). En el prototipo se han declarado las clases sólo para su uso en esa aplicación.

### 5.3.1 Estructura de la aplicación

Desde el punto de vista del diseño, la aplicación misma de PDI puede ser considerada como un objeto sobre la cual pueden ocurrir diversos eventos en diferentes instancias, aplicando transformaciones y filtros a diversas imágenes durante el tiempo de vida de cada objeto.

Desde esta perspectiva consideramos apropiado el método de Grady Booch (1991) para representar tales objetos en la etapa de diseño, adaptándolo posteriormente a la implementación orientada a eventos. Así, el diagrama de un objeto PDI involucra los siguientes elementos.

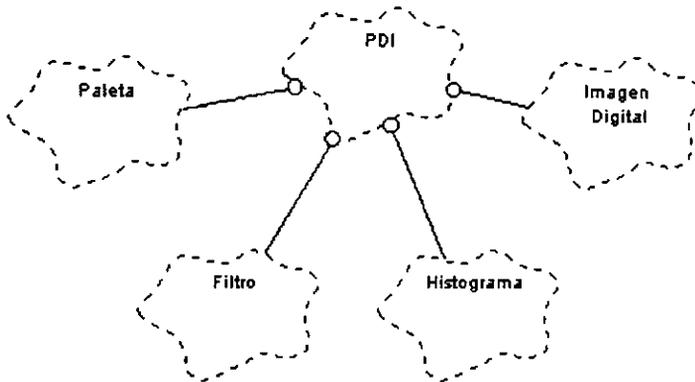


Figura 19. Sistema de PDI como objeto.

### 5.3.2 Estructura de las clases

Para llevar a cabo la abstracción de las clases que intervienen en el proceso de transformación de una imagen es necesario extraer sus interrelaciones y propiedades comunes. De esta manera podemos hacer las siguientes clasificaciones para el prototipo.

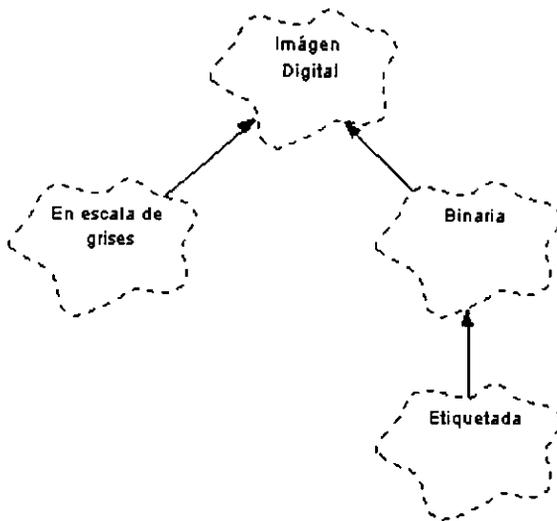


Figura 20. Clasificación del objeto imagen digital.

En esta abstracción pasamos por alto la existencia de imágenes a color, imágenes del mundo real, etc. Puesto que nuestro modelo no considera este tipo de escenas para la construcción del prototipo. Podría pensarse que se debe considerar la imagen del mundo real que da origen a nuestras imágenes digitales, sin embargo partiremos de la imagen digitalizada en 256 niveles de grises, ya que por el momento el proceso de adquisición está fuera de los límites del prototipo, y por lo tanto no se consideró para el diseño.

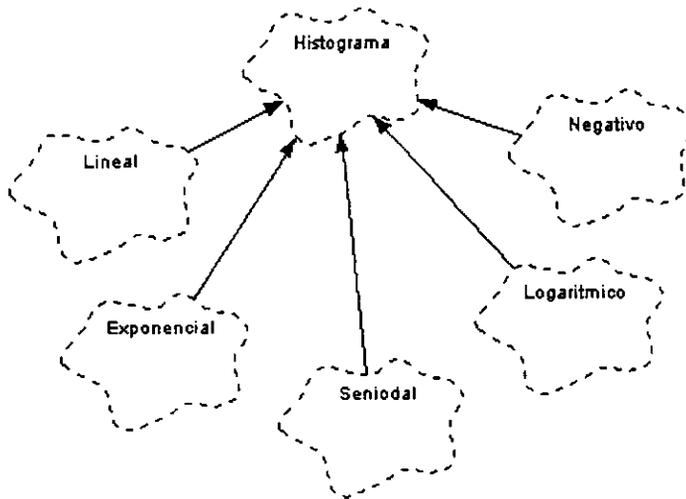


Figura 21. Clasificación del objeto histograma.

Es posible obtener instancias de la clase histograma, que especifiquen transformaciones a objetos imagen. Por ello consideramos necesaria la clasificación de esta, según el tipo de transformación que se requiera.

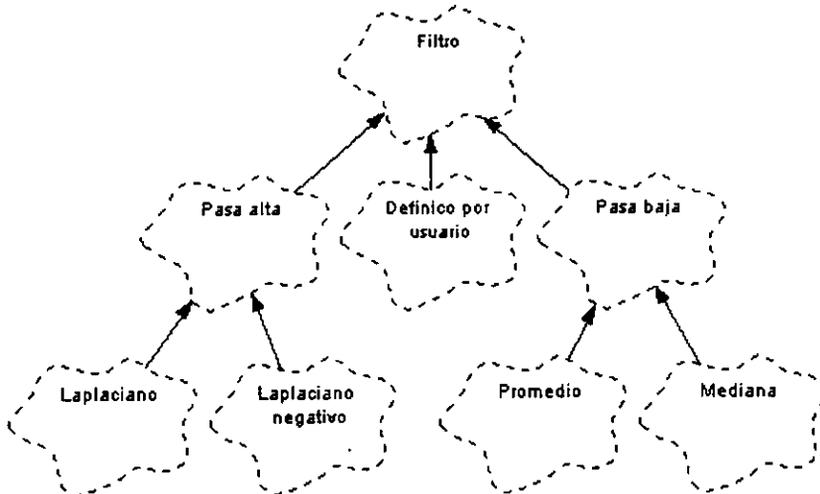


Figura 22. Clasificación del objeto filtro.

Un filtro digital puede ser clasificado de acuerdo con el tipo de señales que conservará en la imagen: Los que eliminan el ruido de alta frecuencia (pasa bajas) y los que resaltan los cambios de frecuencia en la señal (pasa altas). Un tipo adicional de filtro queda abierto para su especificación por el usuario. La figura anterior ilustra esta clasificación básica de los filtros. Evidentemente se podrían incluir en esta clasificación otro tipo de filtros pasa alta o pasa baja, sin embargo, hemos obviado esta situación para fines de delimitación de la clase.

### 5.3.3 Funcionalidad

El objetivo de este apartado no es plasmar la implementación de la funcionalidad completa del prototipo, sino mostrar lo que a nuestro juicio resultó lo más relevante de esta etapa, como la definición de las interfaces con las DLL, la codificación de métodos para la *clase imagen* y *genéricos*, así como poner a consideración el planteamiento de algunos aspectos que se tomaron en cuenta para hacer más eficiente el desempeño de la aplicación.

Durante la implementación se codificó la interfaz con las Bibliotecas de Enlace Dinámico y la funcionalidad de los objetos definidos. Las interfaces con las DLLs son implementadas como se muestra en el siguiente fragmento de programa. La primera parte de la declaración (*Function*) corresponde a la referencia que se hará dentro de Visual Basic, *Lib* especifica la biblioteca (en este caso del sistema, GDI) que contiene la función y *Alias* determina el nombre de la función dentro de la DLL.

```
Declare Function CreatePalette64% Lib "GDI" Alias "CreatePalette"  
    (lpLogPalette As LOGPALETTE64)  
  
Declare Function SelectPalette Lib "user" (ByVal hDC As Integer, ByVal  
    hPalette As Integer, ByVal bForceBackground As Integer) As Long  
  
Declare Function RealizePalette Lib "user" (ByVal hDC%) As Integer  
  
Declare Function UpdateColors% Lib "GDI" (ByVal hDC%)
```

Figura 24. Interfaz con las Bibliotecas de Enlace Dinámico (fragmento).

El siguiente fragmento de código establece algunas constantes y estructuras útiles para el manejo de imágenes y paleta de colores (LUT). `BITMAP`, `PALETTEENTRY` y `LOGPALETTE64`, son algunas de las estructuras simples utilizadas como parte de esta interfaz y por lo tanto no tienen funcionalidad propia.

```
Global Const VBM_GETPALETTE% = &H101C
Global Const PALENTRIES = 256
Global Const PC_RESERVED = &H1
Global Const CF_PALETTE = 9

Type BITMAP
    bmType As Integer
    bmWidth As Integer
    bmHeight As Integer
    bmWidthBytes As Integer
    bmPlanes As Byte
    bmBitsPixel As Byte
    bmBits As Long
End Type

Type PALETTEENTRY
    peRed As Byte
    peGreen As Byte
    peBlue As Byte
    peFlags As Byte
End Type

Type LOGPALETTE64
    palVersion As Integer
    palNumEntries As Integer
    palPalEntry(PALENTRIES) As PALETTEENTRY
End Type

Global UsePalettes%
Global logPalettes As LOGPALETTE64
```

Figura 23. Algunas constantes y estructuras para el manejo de imágenes y *Look-Up Table*.

A continuación se muestra la implantación de funcionalidad al objeto `frmImage`, que es la clase que encapsula el comportamiento de un *objeto imagen*. Como se puede observar, existen varios eventos predefinidos disponibles para especificar el código de respuesta a eventos (`MouseDown`, `MouseMove`, `MouseUp`, etc.).

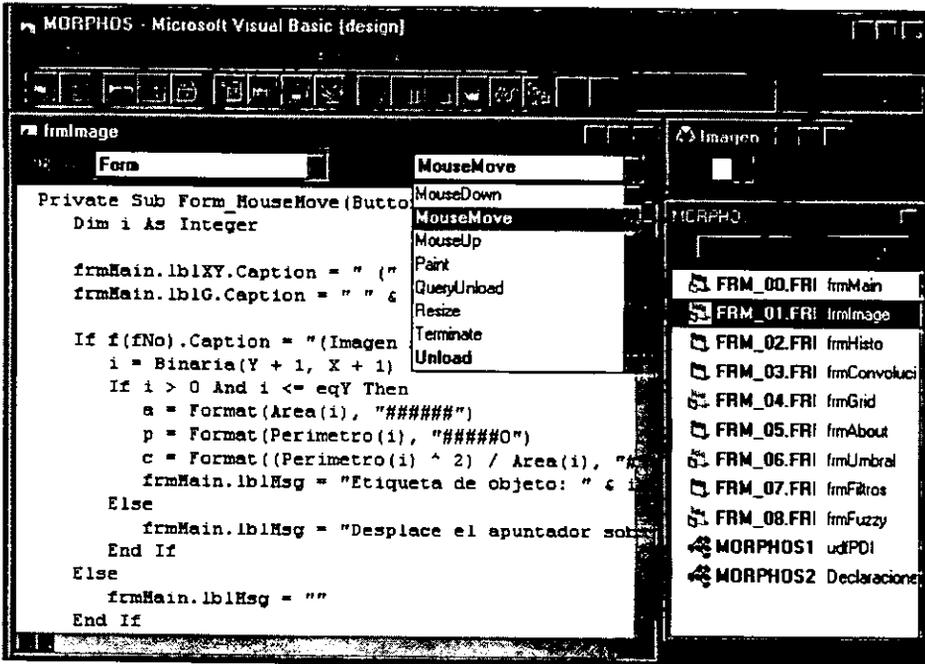


Figura 25. Dotando de funcionalidad a la clase `frmImage`.

La instanciación de un *objeto imagen* se logra declarando una variable `f(fNo)` del tipo `frmImage` (clase), con lo que el nuevo elemento `f(fNo)` hereda variables, funcionalidad y estado de la clase.

Redim Preserve `f(fNo)` as New `frmImage`

El control sobre las imágenes en memoria se obtiene mediante un arreglo del tipo `ImageInfo`, cuyo índice es actualizado con el evento `Activate` y `Unload` de `f(fNo)`, lo cual asegura la vigencia de estas propiedades cada vez que se selecciona una nueva imagen.

Estas propiedades pudieron haberse definido como parte de la clase `frmImage`, con el inconveniente de que al activarse el evento `Unload` de un objeto `f(fNo)`, estas se perderían. Esta separación objeto-propiedades se implementó así simplemente con la finalidad de llevar un historial sobre las imágenes procesadas. El control de su estado en el sistema queda determinado por el valor booleano de `imInfo(fNo).imActiva`.

```
Type ImageInfo
    imIndice As Integer
    imNombre As String
    imAncho As Integer
    imAlto As Integer
    imActiva As Boolean
End Type

Global imInfo() As ImageInfo
```

Figura 26. Registro de control sobre las imágenes.

De manera similar, los métodos de modificación del histograma, filtrado, segmentación, etiquetado de componentes y extracción de características se separaron del objeto, no así los métodos MoveMouse, Load, Unload, Activate, Actualiza\_histograma y Crea\_contexto. Hicimos esta separación tomando en cuenta las siguientes consideraciones:

- Solo formaran parte de la clase los métodos que tengan una fuerte cohesión con la misma, pues el sobrecargar un objeto de métodos complejos afecta el desempeño de la aplicación en situaciones de proliferación excesiva de objetos imagen.
- En determinado momento de la evolución de un objeto imagen algunas funcionalidades dejan de ser aplicables, consumiendo recursos innecesariamente. Por ejemplo, al obtener una imagen etiquetada o binaria, carece de sentido que estos objetos tuviesen implantado el método de segmentación.

Así, las funcionalidades que se pudieran desligar del objeto y que fuesen especialmente costosas en cuanto a recursos computacionales, se implementaron como *métodos globales*. Para afectar un *objeto imagen* particular, se tomaron como parámetros algunas propiedades del objeto  $f(fNo)$  activo en lugar de que el método estuviera implementado dentro de cada uno de los objetos  $f(fNo)$ .

```
Sub Aplica_mediana()  
  Dim k, Col, Columna, Lin, Linea As Integer  
  Dim Vecino(0 To 8) As Long  
  Dim eg As Integer  
  For Linea = 0 To imInfo(fNo).imAncho - 2  
    For Columna = 0 To imInfo(fNo).imAlto - 2  
      k = 0  
      For Lin = Linea To Linea + 2  
        For Col = Columna To Columna + 2  
          Vecino&(k) = GetPixel(f(DataSource).hDC, Lin, Col)  
          k = k + 1  
        Next Col  
      Next Lin  
      Call Ordena_arreglo(Vecino&(), mediana&)  
      eg = Int(mediana& / 65536)  
      f(fNo).PSet (Linea + 1, Columna + 1), RGB(eg, eg, eg)  
    Next Columna  
  Next Linea  
  f(fNo).picture = f(fNo).Image  
  Call Crea_histograma(aHisto_activo, Maximo_activo)  
  f(fNo).Actualiza_histograma  
End Sub
```

Figura 27. Procedimiento genérico para aplicar el filtro de mediana.

# 6 Resultados

En este capítulo presentamos como resultado el prototipo del sistema (al que hemos denominado *Morphos*) como una aplicación para el procesamiento y análisis de imágenes de propósito específico, desarrollado bajo Windows 95™.

En este prototipo se ha incorporado el método de segmentación propuesto, así como el método de etiquetado secuencial 8-conexo. De manera complementaria se han implementado algunos algoritmos para la mejora y extracción de características de imágenes así como otras opciones de segmentación, desarrollando estos conceptos en un ambiente de programación visual, con su correspondiente implementación orientada a eventos.

Al final de este capítulo se presenta un ejemplo de la funcionalidad del sistema, con la finalidad de cubrir una secuencia de procesos que van desde la selección de una imagen hasta su procesamiento y extracción de características, con lo cual se muestran los resultados de los métodos propuestos.

La serie de operaciones requeridas para mejorar la apariencia de una imagen podría ser tan variada como el usuario lo determine basándose en su experiencia. En esta sección se aplica únicamente la transformación lineal al histograma antes de proceder a la segmentación, sin embargo el usuario final podría efectuar diferentes transformaciones o filtrados para comparar resultados.

El adquirir experiencia en la interpretación de imágenes y sus histogramas es determinante para llegar a resultados óptimos en la segmentación, donde la calidad de la imagen adquirida juega un papel muy importante.

## 6.1 Descripción

*Morphos* recupera, despliega, procesa, analiza, imprime y guarda imágenes en formato BMP. Las funciones de análisis de imágenes incluyen medidas dimensionales así como de escala de grises (histograma con estadísticas). *Morphos* soporta las funciones estándar para el procesamiento de imágenes, tales como manipulación del contraste, detección de contornos, filtro de mediana, convoluciones espaciales con máscaras de convolución definidas por el usuario.

El programa es una aplicación con Interfaz de Múltiples Documentos (MDI), soportando cualquier número de ventanas (imágenes) simultáneamente, limitado solo por la memoria disponible.

A continuación se especifican los requerimientos para la instalación de *Morphos*, y posteriormente los resultados de las operaciones implantadas en el prototipo sobre algunas imágenes de prueba de diversa índole.

### 6.1.1 Requerimientos

Para llevar a cabo la instalación del prototipo, es necesario tener un equipo con las siguientes características:

- Procesador Intel 80486 (o superior)
- Memoria RAM mínima de 16MB, recomendada 32MB. Mayor RAM mejorará significativamente el desempeño de *Morphos*, incrementando además el número de imágenes que se pueden mantener abiertas a la vez.
- Monitor capaz de desplegar 256 colores (ó más) a una resolución de 640x480 pixeles.
- Sistema operativo Microsoft Windows 95™ ó Microsoft Windows NT™ versión 3.51 (ó más reciente).
- Unidad de 3½" de 1.44 M

## 6.1.2 Entrada al sistema

*Morphos* puede ser distribuido con dos discos de instalación. Al finalizar la instalación se genera un grupo de programa llamado *Morphos 1.0*, el cual podrá ser accedido desde el *Menú Inicio* de Microsoft Windows 95™ ó Microsoft Windows NT™.



Después de activar la entrada al sistema, una pantalla de presentación será desplegada, así como el *Menú del sistema*.

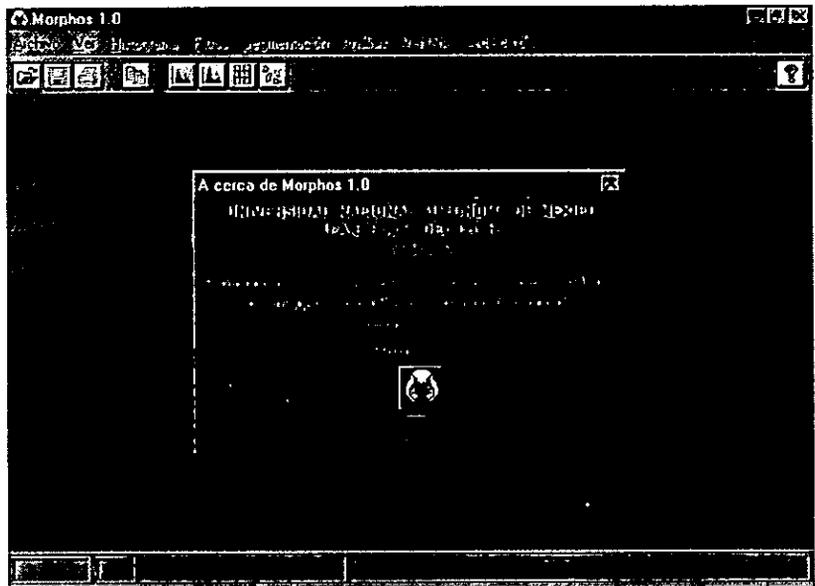


Figura 28. Entrada al sistema.

## 6.2 Filtros de dominio espacial

El *filtrado* de una imagen digital es una operación de dominio espacial que actúa sobre el valor de un pixel, tomando como parámetros los valores de sus pixeles vecinos. Esta operación, también llamada de convolución, es efectuada generalmente sobre la imagen completa.

1	2	3
4	5	6
7	8	9

Figura 29. Representación de los vecinos ( $N_8$ ) del pixel 5 en una máscara de 3x3.

*Morphos* utiliza matrices de 3x3 para aplicar filtros pasa baja y pasa alta, en virtud de que las máscaras de convolución más comunes (*Roberts, Sobel, Prewitt, Kirsch*, etc.) podrían ser aplicadas [Castleman-Kenneth, 1996] sin consumir excesivamente los recursos del sistema, como sucedería con filtros más grandes.



En el prototipo esta opción puede ser seleccionada con el comando *Filtros* del *Menú principal*, o bien con el icono dispuesto para este fin en la *Barra de herramientas*.

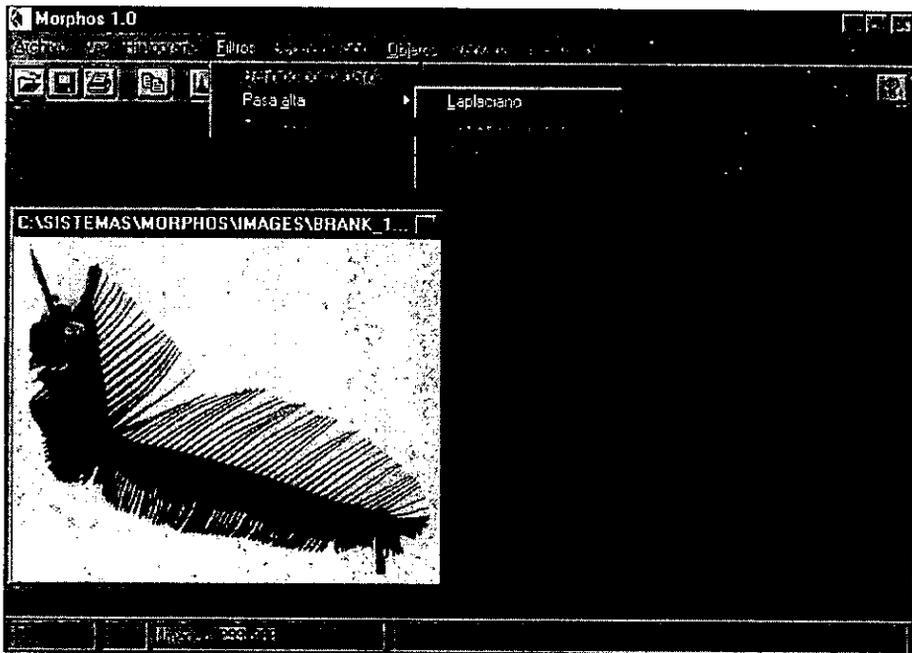


Figura 30. Imagen de branquiespina de sardina monterrey y opciones de filtrado.

A continuación se muestran los resultados de aplicar algunos filtros a imágenes de branquiespinas de sardina monterrey.

**Definido** Los filtros definidos por el usuario permiten una mayor flexibilidad en las operaciones de convolución, puesto que los valores de la máscara son asignados por el usuario de acuerdo a su criterio.

Para ello se dispone de la siguiente ventana, en la que estos valores pueden ser asignados, almacenados para su uso futuro, o bien recuperar algún filtro almacenado previamente.

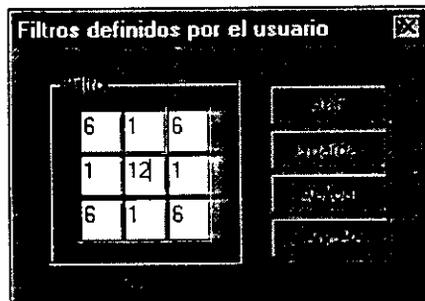


Figura 31. Filtros definidos por el usuario.

Una vez que se ha determinado el filtro a utilizar, bastará con activar el evento *click* sobre el botón *Aplicar* para llevar a cabo la operación de convolución definida.

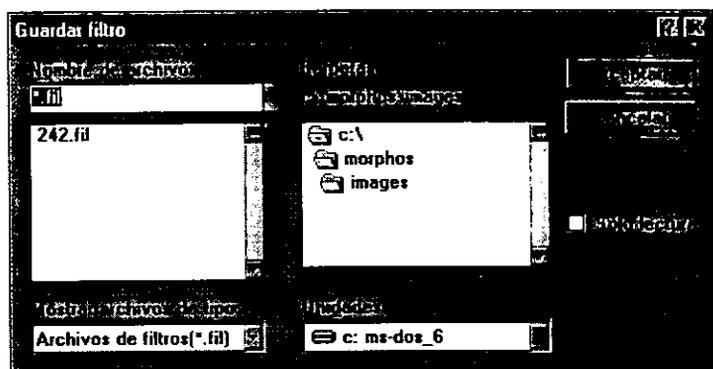


Figura 32. Cuadro de diálogo para guardar filtro.

Estos filtros son almacenados en archivos ASCII con la extensión .FIL.

### Pasa baja

Los filtros pasa baja son utilizados para suavizar la imagen, atenuando las altas frecuencias producidas por cambios bruscos en los valores de píxeles vecinos.

Esta característica es útil para reducir el ruido ocasionado durante la etapa de adquisición, sin embargo, dependiendo de las características de la imagen, esta convolución puede producir *imágenes borrosas*.

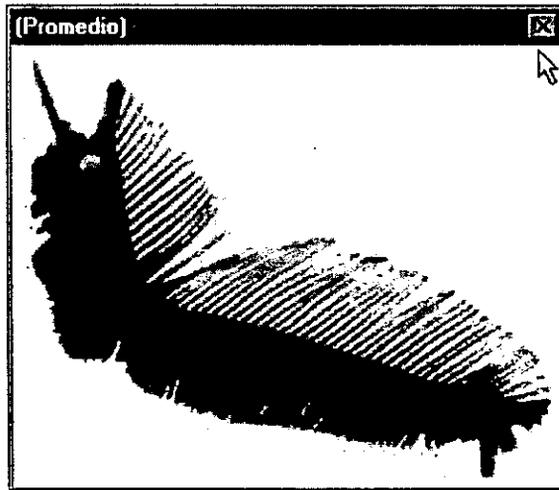


Figura 33. Efectos del filtro de promedio.

La figura anterior muestra el efecto de aplicar un filtro pasa baja (promedio) a la imagen de un arco branquial. Aquí, los valores de los nueve píxeles de la ventana de convolución son promediados, asignando el resultado al píxel central, con lo cual el ruido de alta frecuencia es atenuado, lo que afecta la definición de los bordes de las branquiespinas en contraste con el fondo.

Este proceso es repetido para cada píxel en la imagen. Es por ello que las operaciones de convolución son especialmente exigentes de recursos de cómputo, y por lo tanto costosas en

cuanto a tiempo de cálculo y despliegue.

### Pasa alta

Los filtros pasa alta enfatizan los cambios de intensidad de los píxeles, razón por la cual estos filtros son usados para detectar bordes de objetos.

Para evitar la detección de *bordes falsos*, ocasionados por ruido, sombras, o cualquier otra interferencia, debe considerarse un proceso de mejoramiento de la imagen antes de llevar a cabo la detección de bordes. Este mejoramiento puede realizarse utilizando las técnicas de modificación de histograma o los filtros pasa baja.

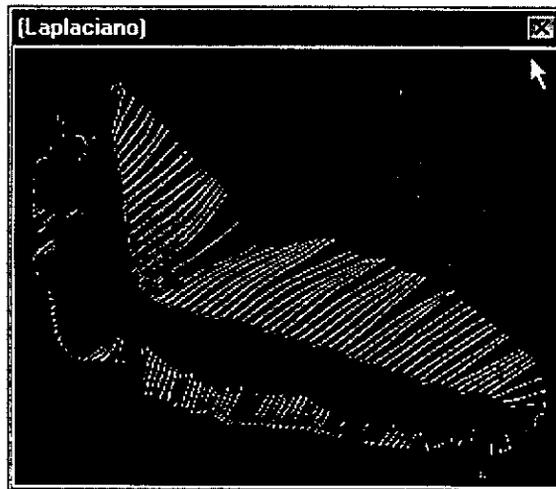


Figura 34. Efectos del filtro Laplaciano.

En la figura anterior se observa como la mayor parte de los bordes verdaderos han sido detectados con este filtro pasa alta.

Los siguientes son los filtros implementados en el sistema con sus respectivas mascarar. Algunos de los filtros incluyen además un ponderador que aparece a la derecha de las máscaras, haciendo la función de divisor. Cabe señalar que los valores que se producen al aplicar estos filtros son llevados al rango de números enteros de 0 a 255.

### Filtros pasa alta

1		1
	-4	
1		1

Laplaciano

-1		-1
	5	
-1		-1

Laplaciano negativo

1	-2	1
-2	5	-2
1	-2	1

Pasa alta 5 en centro

-1	-2	-1
-2	19	-2
-1	-2	-1

/ 7

Pasa alta 1/7

### Filtros pasa baja

1	1	1
1	1	1
1	1	1

/ 9

Promedio

1	1	1
1	2	1
1	1	1

/ 10

Pasa baja 1/10

1	2	1
2	4	2
1	2	1

/ 16

Pasa baja 1/16

1	1	1
1	1	1
1	1	1

Sort<sub>5</sub>

Mediana

El arreglo es ordenado y se selecciona el quinto elemento.

## 6.3 Modificación del histograma

En este apartado se muestran los resultados obtenidos de la implementación de algoritmos referentes a la modificación del histograma. En esta sección en particular, se usó una imagen diferente a la de nuestro caso de estudio, en virtud de que por las características de sus histogramas, no reflejaban claramente algunas transformaciones.

Por tal razón, en esta la fase, se experimentó con el procesamiento de la siguiente imagen con la finalidad de mostrar el efecto de estas transformaciones. Al final de este capítulo se describe un ejemplo en el que se aplican algunas de las operaciones aquí descritas, sobre imágenes de arcos branquiales.

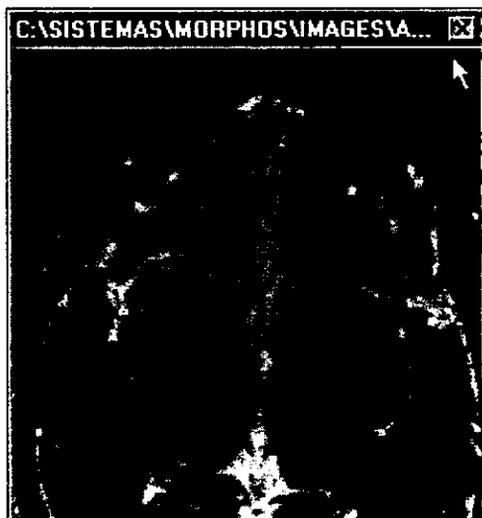


Figura 35. Tomografía de corte transversal del cerebro (256x256x256). Cortesía de la Dra. María Garza Vigil, IIMAS-UNAM).

La transformación del histograma tiene como objetivo la mejora de la imagen, considerando los métodos de procesamiento que se basan en la intensidad de píxeles individuales, los cuales son modificados mediante alguna función matemática que mapea los valores de la LUT a niveles de gris convenientes según el caso de estudio.



En el prototipo, estas transformaciones pueden ser seleccionadas con la opción *Histograma* del *Menú principal*, o bien con el icono correspondiente de la *Barra de herramientas*.

La selección de este comando implica principalmente dos funciones:

- Mostrar el histograma de la imagen activa, o bien,
- a partir de esta representación el usuario podrá determinar la transformación necesaria para modificar la apariencia de la imagen (realizando los objetos de interés).

El tipo de transformación a efectuar se especifica mediante el evento *Click* sobre un objeto *TabControl* que contiene las opciones de transformación disponibles, tal como se muestra en la figura 36.

Al finalizar el proceso de transformación se crea un nuevo *objeto imagen*, dejando el original sin modificaciones, lo cual permite efectuar otras transformaciones sobre la imagen original.

También es posible efectuar nuevas transformaciones sobre las imágenes generadas, lo cual permite encadenar diversas transformaciones a las imágenes resultantes.

A continuación se muestran los resultados de aplicar diferentes tipos de transformación al histograma sobre la imagen del corte transversal del cerebro.

### Lineal

La transformación lineal es utilizada para aumentar el contraste de una imagen a partir de la especificación de un rango en el histograma, delimitado por los puntos  $a$ ,  $b$ .

Para llevar a cabo esta transformación usamos la siguiente fórmula para calcular los valores de los píxeles en la nueva imagen.

$$Z' = \left[ \frac{Z_k - Z_1}{a - b} (Z - a) \right] + Z_1$$

Donde:

$Z'$  = Nuevo valor del píxel

$Z$  = Valor original del píxel

$Z_k$  = Máximo valor de gris (en LUT)

$Z_1$  = Mínimo valor de gris (en LUT)

a = Primer valor del intervalo a expandir  
b = Segundo valor del intervalo a expandir

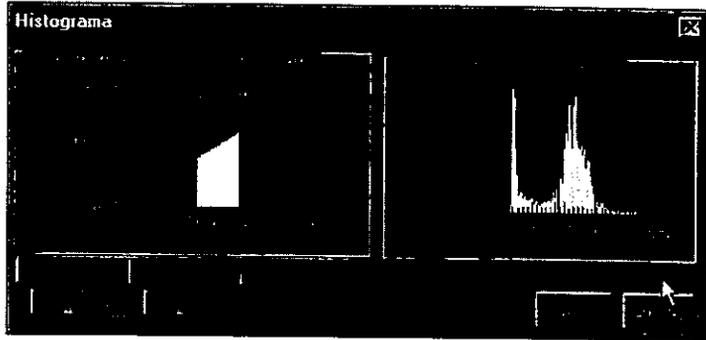


Figura 36. Selección de la región a contrastar.

Para llevar a cabo esta transformación es necesario especificar la porción del histograma que deseamos contrastar, mediante la selección de dos puntos *a* y *b* en el histograma de la imagen.

La selección de este segmento está determinada por el evento *Click* del *PictureBox Histograma de la imagen*. Una vez seleccionado, el sistema procede a modificar la imagen y en consecuencia su histograma, el cual se muestra a la derecha de la figura anterior con líneas oscuras, superpuestas al histograma original. El rango seleccionado se ha expandido uniformemente en la escala de grises.

La siguiente figura muestra el aumento en el contraste de la imagen como resultado de esta transformación.



Figura 37. Imagen resultante de la transformación lineal.

### Logarítmica

La transformación logarítmica modifica los píxeles de la imagen según la siguiente función, la cual produce la distribución de frecuencias que se muestra en la gráfica izquierda de la siguiente figura.

$$Z' = \text{Log}(Z + 1)\alpha$$

Donde:

$Z'$  = Nuevo valor del píxel

$Z$  = Valor original del píxel

$\alpha$  = Factor de ajuste

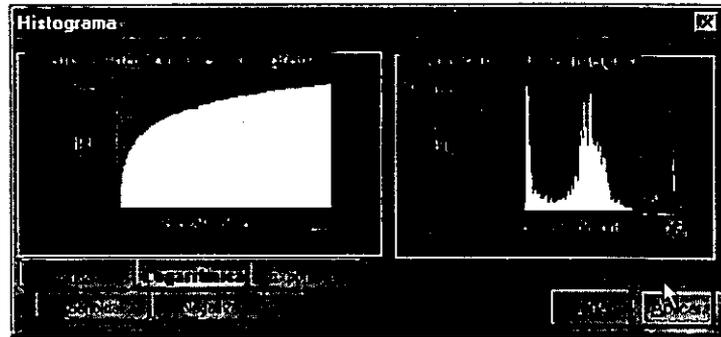


Figura 38. Función de transformación logarítmica.

En la figura anterior se puede observar el histograma resultante (tono más oscuro) que se encuentra sobrepuesto al original. Esta transformación tiende a estrechar el histograma manteniendo su forma básica, pero situándolo sobre los tonos más claros del nivel de gris, sin alterar las clases.



Figura 39. Imagen transformada logarítmicamente.

**Exponencial** Esta transformación utiliza la siguiente expresión para modificar los valores de píxeles en la imagen original.

$$Z' = \text{Exp}(Z\alpha)$$

Donde:

$Z'$  = Nuevo valor del píxel

$Z$  = Valor original del píxel

$\alpha$  = Factor de ajuste

La gráfica de esta función es mostrada en el *PictureBox Transformaciones al histograma* de la siguiente figura.

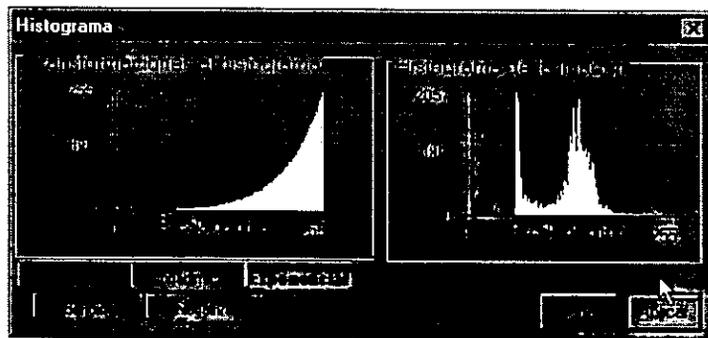


Figura 40. Función de transformación exponencial.

Por la distribución del histograma original, esta transformación, tiende a oscurecer la imagen, dado que hay un amplio rango de la función de transformación en el que los niveles de gris de la paleta original, mapean a valores bajos (oscuros) de la paleta resultante. Obteniendo píxeles claros, sólo para valores muy altos de tonos de gris.

Dado que para la imagen de prueba, los píxeles están situados en la parte media del histograma, la imagen resultante es por consiguiente muy oscura.

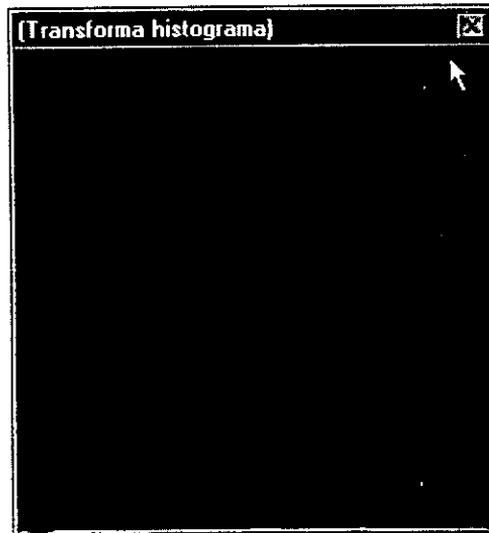


Figura 41. Imagen transformada exponencialmente.

## Senoidal

Esta transformación modifica los pixeles de la imagen según la siguiente función.

$$Z' = \text{Sin}[(Z\pi)/180]\alpha$$

Donde:

- $Z'$  = Nuevo valor del pixel
- $Z$  = Valor original del pixel
- $\pi$  = Constante PI
- $\alpha$  = Factor de ajuste

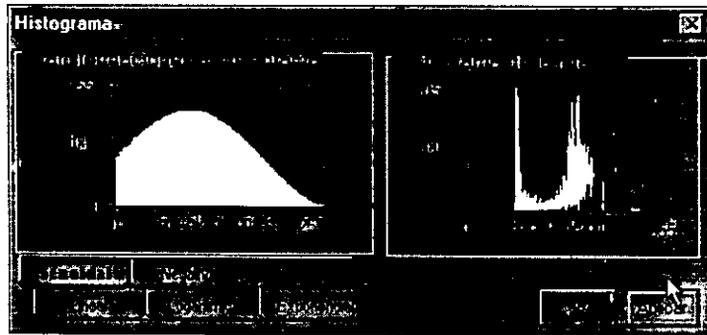


Figura 42. Función de transformación senoidal.

Por la forma del histograma original esta transformación actúa de manera similar a la transformación negativa, en virtud de que no existen valores bajos de gris (prácticamente todos están por encima del tono 64), la pendiente positiva de la función no afecta a la imagen.



Figura 43. Imagen transformada senoidalmente.

### Negativa

Esta transformación invierte el orden de la distribución de frecuencias en el histograma, de forma que la intensidad de la imagen de salida disminuya conforme la intensidad de la imagen de entrada aumenta. Esto es expresado por la siguiente función, cuya gráfica se muestra en el *PictureBox Transformaciones al histograma* de la siguiente figura.

$$Z' = Z_k - Z$$

Donde:

$Z'$  = Nuevo valor del pixel

$Z$  = Valor original del pixel

$Z_k$  = Máximo valor de gris (en LUT)

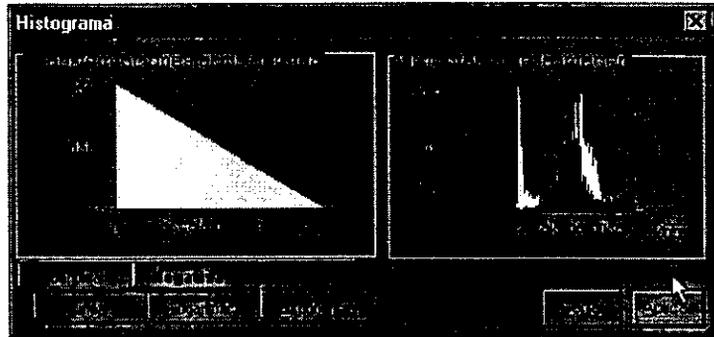


Figura 44. Función de transformación negativa.

Nótese como el histograma resultante (tono más oscuro) mostrado a la derecha de la figura anterior, está distribuido de manera opuesta al original, obteniendo de esta forma el negativo de la imagen.

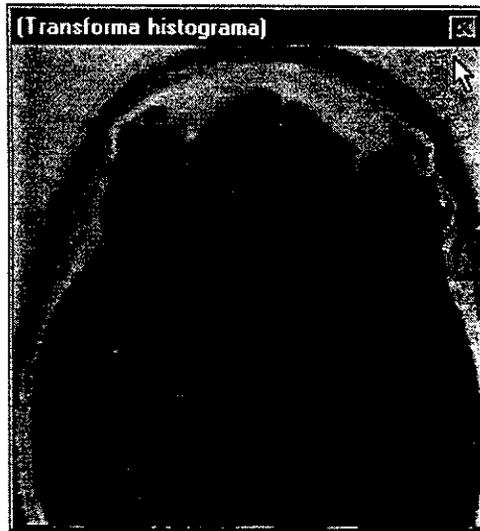


Figura 45. Transformación negativa de una imagen.

El uso de esta transformación podría interactuar con la aplicación de los métodos de umbralizado, ya que a partir del negativo de una imagen podemos invertir el valor de los píxeles que el sistema considera como *fondo* para ser considerados como *objeto* de la imagen.

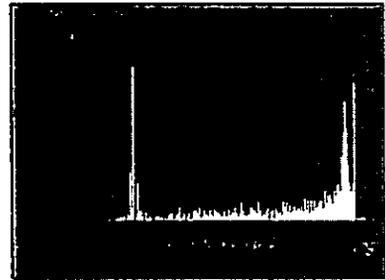
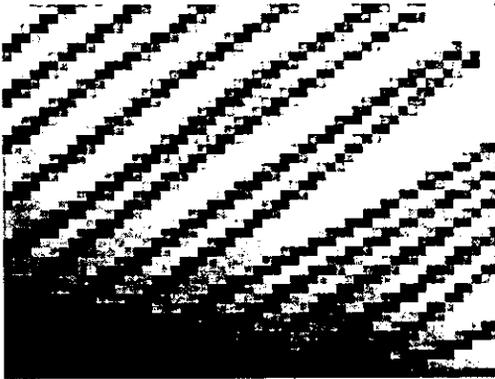
## 6.4 Segmentación

En esta sección se muestran los resultados de la implementación de los métodos de Otsu y de Incertidumbre discutidos en el capítulo 4. Adicionalmente, el prototipo dispone una opción de selección manual del umbral.



Para activar la opción de segmentación en el prototipo, seleccione el comando de *Segmentación* del *Menú principal*, o bien haga clic sobre el icono correspondiente en la *Barra de herramientas*.

La siguiente figura muestra una porción de arco branquial amplificada 5 veces con la finalidad de observar algunos detalles de la segmentación al aplicar las técnicas antes descritas.



Como se puede observar en una distribución de frecuencias de este tipo es difícil establecer un umbral óptimo.

Figura 46. Porción de arco branquial (54x40x256) y distribución de frecuencias de los niveles de gris.

### Otsu

El método de Otsu localiza el umbral en el nivel de gris 149, determinando con ello las clases *fondo* y *objeto* que se muestran en la siguiente imagen binaria.

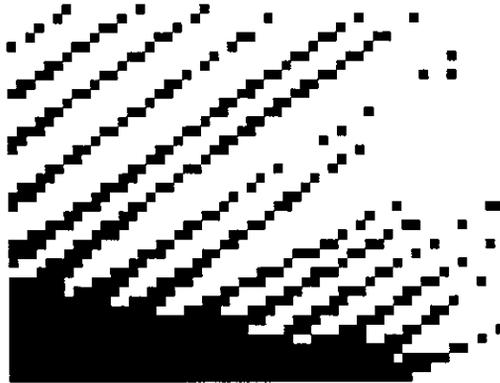


Figura 47. Segmentación por el método de Otsu.

Esta segmentación no ocasionó el problema de unión de branquiespinas en segmentos terminales, pero produjo el aislamiento pixeles y grupos de pixeles, algunos de los cuales debieron formar parte de las branquiespinas.

### Manual

Como una opción complementaria se implementó la selección manual del umbral. Mediante ésta es posible experimentar con variaciones en la especificación del umbral.

La especificación del umbral está implementada en el evento *click* del *PictureBox Histograma*, en el diálogo *Selección de umbral*. De manera similar, el evento *MouseMove* de *PictureBox*, activa el despliegue del nivel de gris en la *Barra de estado*.

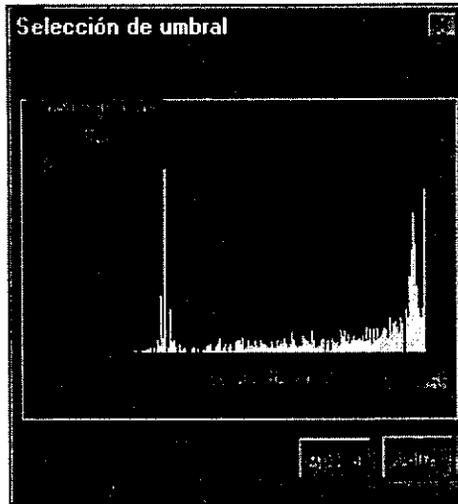


Figura 48. Selección manual de umbral.

Una vez seleccionado el umbral es posible hacer ajustes, haciendo clic nuevamente sobre el histograma.

La siguiente figura muestra la imagen binaria generada a partir de la segmentación con un umbral definido por el usuario en el nivel de gris 224.

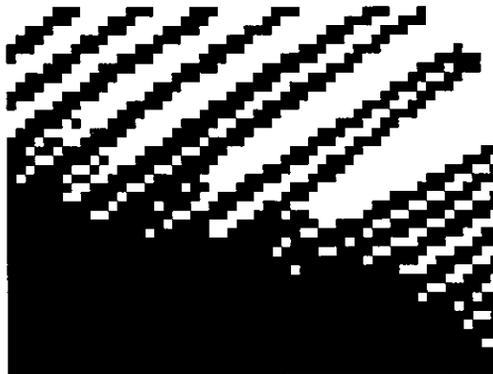


Figura 49. Selección manual del umbral en la imagen de branquiespina.

Como se puede observar, la especificación de un umbral muy alto ha producido que muchos de los pixeles correspondientes al fondo se consideren como parte del objeto.

**Incertidumbre:** Este es el resultado de la implementación del algoritmo propuesto en este trabajo, y que también se ha sido incluido en el prototipo.

Como se ha mencionado anteriormente, el algoritmo se ha desarrollado para un caso de estudio muy particular (imágenes que presenten objetos en forma de fibras).

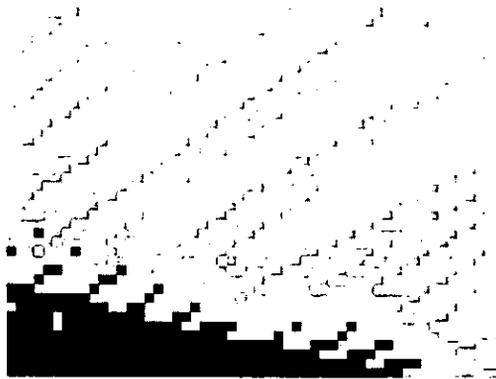


Figura 50. Objeto (negro), *Incertidumbre* (gris) y fondo (blanco).

En la figura anterior se muestran las clases que han sido *bien* definidas segun el algoritmo, que corresponden a los pixeles negros y blancos.

Tambien se puede observar el conjunto de pixeles que han sido considerados como de incertidumbre (pixeles en gris), a partir de éstos se utiliza una heurística adicional implementada con una mascara binaria para tratar de evitar la consideración de fragmentos que no estén conectados al cuerpo del arco branquial, como ya se explicó en el apartado *Un conjunto de incertidumbre*.

Después de aplicar las operaciones binarias entre la mascara definida y sus vecinos, se obtiene el siguiente refinamiento de la segmentación anterior.

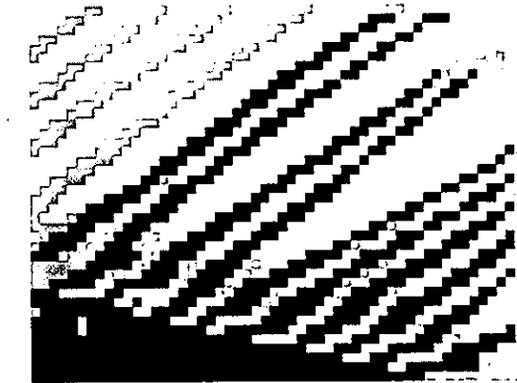


Figura 51. Refinamiento de la segmentación.

Los pixeles en gris oscuro han sido reclasificados como parte del objeto, y los pixeles gris claro como parte del fondo.

Nótese también que como consecuencia del uso de la mascara, la mayoría de las branquiespinas aisladas son eliminadas de la imagen, lo cual agiliza el desempeño del *filtro de tamaño* usado después del cálculo de área.

## 6.5 Análisis

Como ya se mencionó anteriormente, el análisis consiste en extraer características de los objetos una vez que la imagen ha sido segmentada. En aplicaciones de reconocimiento de formas, estas características son utilizadas como patrones para localizar objetos en una escena. En el caso de nuestra aplicación las utilizamos para encontrar rasgos de interés en el arco branquial y específicamente en las branquiespinas.

Este análisis nos conduce a la extracción de características básicas, por lo que será necesario desarrollar en un futuro, funcionalidades que permitan otros tipos de mediciones, como se señala en el capítulo 8.



Esta opción es activada con el comando *Análisis* del *Menú principal*, o con el icono destinado para este fin en la *Barra de herramientas*.

Para iniciar el proceso de análisis, es necesario que la imagen sobre la que se va a trabajar esté en formato binario, es decir, que se haya aplicado alguno de los métodos de segmentación descritos en el capítulo anterior.

La primera fase del análisis consiste en el *etiquetado de componentes*, cuyos principios y algoritmo propuesto se explicaron a detalle en la sección 4.1. Aquí se asigna un valor común a cada uno de los píxeles que conforman una región, considerando una región al conjunto de píxeles 8-conectados entre sí.

Una vez que se han etiquetado los objetos, el sistema procede a calcular las características de cada uno de ellos, mostrando la información resultante del análisis en la *Ventana de resultados*.

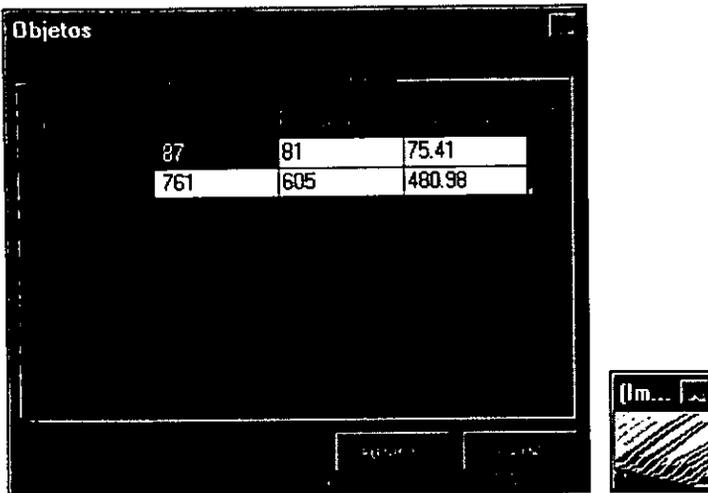


Figura 52. Resultados del análisis de la porción de arco branquial presentado en la sección 6.4 (segmentado con el método propuesto).

Las operaciones sobre los objetos que resultan de interés para el caso de estudio, consisten en determinar los siguientes atributos para cada objeto encontrado en la imagen. Aquí es necesario mencionar la necesidad de una opción de calibración para las imágenes, estableciendo la equivalencia de un pixel a medidas del mundo real. Para el caso de estas imágenes un pixel equivale a 0.033 mm.

**Area** El área de un objeto, está determinada por el número de pixeles que lo componen.

**Perímetro** El perímetro está determinado con el número de pixeles que forman el límite o borde del objeto.

**Compacidad** Calculada como:  $\text{Perímetro}^2 / \text{Area}$

El resultado determina de manera genérica la forma de los objetos. Si se trata de objetos alargados, estos tendrán una compacidad alta como resultado de su área pequeña y perímetro grande. En objetos circulares la compacidad disminuye.

Este parámetro podría ayudar a determinar cuáles de los objetos aislados en una imagen segmentada corresponden a segmentos de branquiespinas, tomando como referencia su forma alargada.

Una vez que esta información es mostrada por el prototipo (figura 52) es posible almacenarla en un archivo de texto para su uso posterior. De igual forma, la imagen segmentada puede ser guardada en formato BMP.

## 6.6 El prototipo

Para llevar a cabo la prueba del prototipo, partimos de imágenes digitales en formato BMP que han sido adquiridas con una tarjeta *Frame Grabber Computer Eyes/RT de Digital Vision, Inc.* La siguiente imagen corresponde a un arco branquial y su histograma.

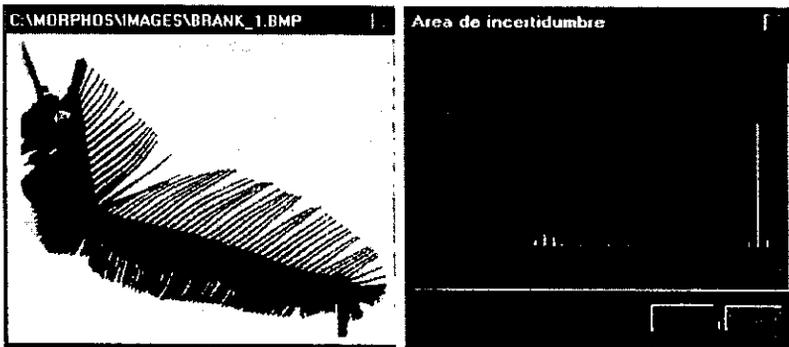


Figura 53. Imagen completa de arco branquial (300x239x256) y su histograma.

Como se puede observar en la figura anterior, el histograma muestra una amplia zona de incertidumbre entre las dos clases. Esta porción del histograma trató de reducirse, ampliando el contraste mediante una transformación lineal al histograma.

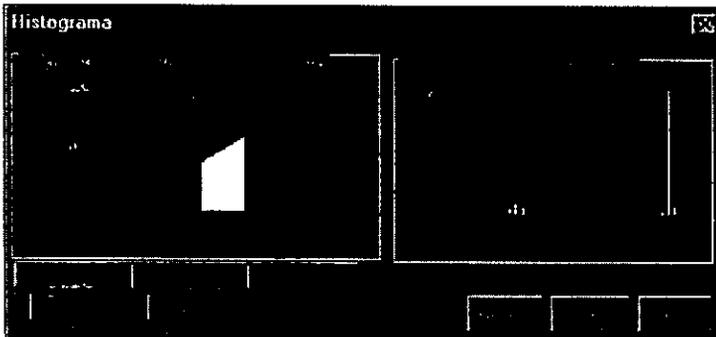


Figura 54. Transformación lineal al histograma.

Después de aplicar esta transformación, se procedió a segmentar la imagen por el método de Otsu, y posteriormente a la extracción de características, obteniendo el resultado que se muestra en la figura 55, donde podemos observar como se ha presentado uno de los problemas descritos en el capítulo 3: la proliferación de objetos aislados que debieron integrarse al arco branquial durante la segmentación.

A causa de lo anterior, se han generado 262 componentes, de los cuales la mayoría son porciones terminales de branquiespinas. En la *Ventana de resultados* se muestran las características de 9 de ellos.

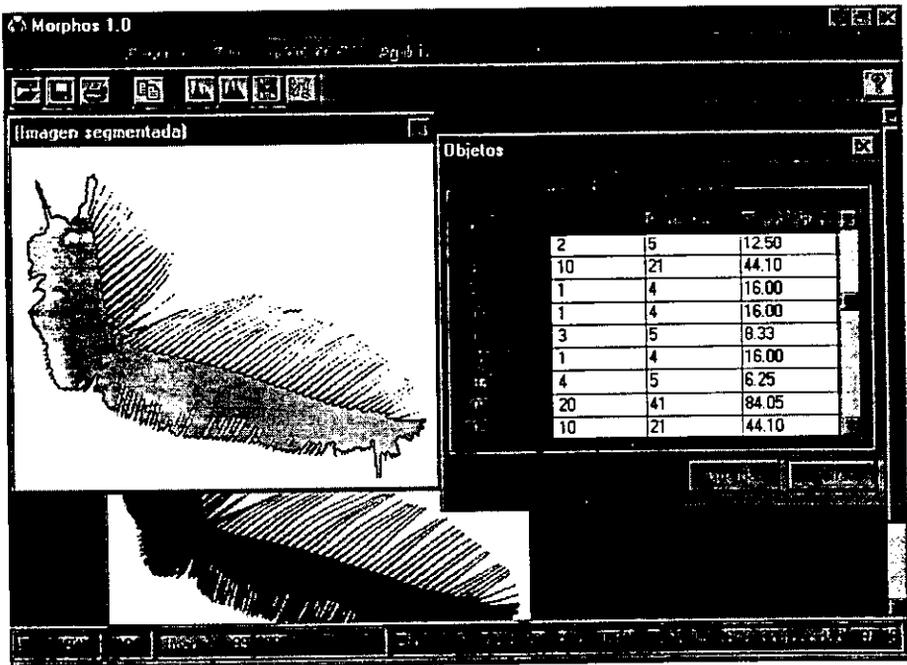


Figura 55. Segmentación por el método de Otsu y extracción de características.

En la siguiente figura mostramos los resultados de procesar la misma imagen, esta vez utilizando el método propuesto, después de la transformación al histograma mencionada anteriormente (figura 54).

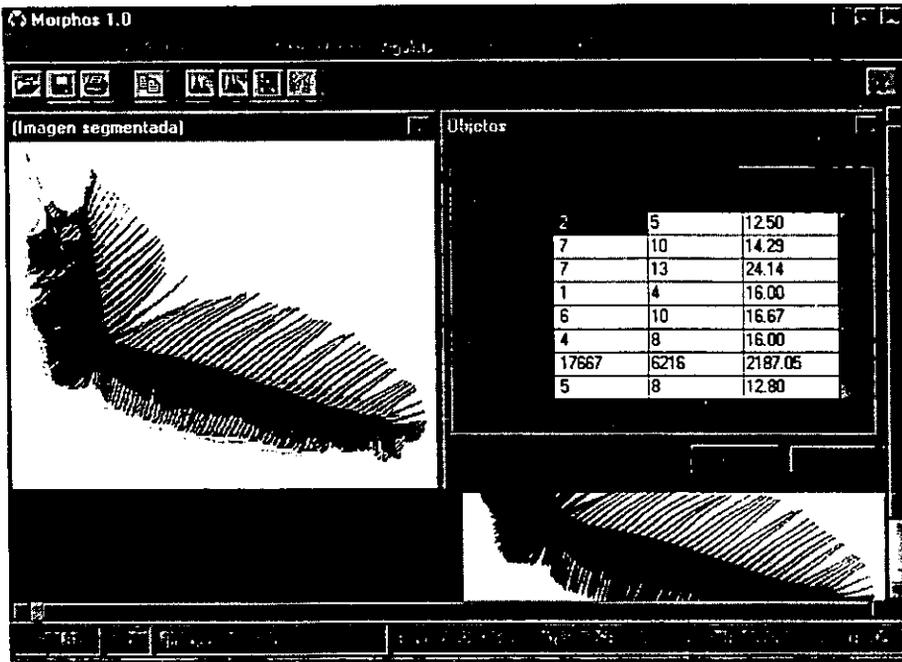


Figura 56. Segmentación por el método de incertidumbre y extracción de características.

En la imagen segmentada por el método propuesto, se puede observar la porción de incertidumbre que fué clasificada como fondo (tonalidad gris claro en torno a la imagen), así como la *Ventana de resultados* mostrando los 8 componentes localizados en la escena antes de aplicar el filtro de tamaño, después de lo cual la imagen sólo contendría el objeto con etiqueta 7 (por ser el de área mayor).

Además de la ventana de resultados, se ha implementado una opción de consulta de características sobre la imagen clasificada, en la que a partir de pocisionar el cursor sobre esta, la información acerca del objeto es mostrada en la *Barra de estado*, lo que facilita la obtención de información sobre objetos específicos. En las dos figuras anteriores la *Barra de estado* muestra las características de sus respectivos arcos branquiales.

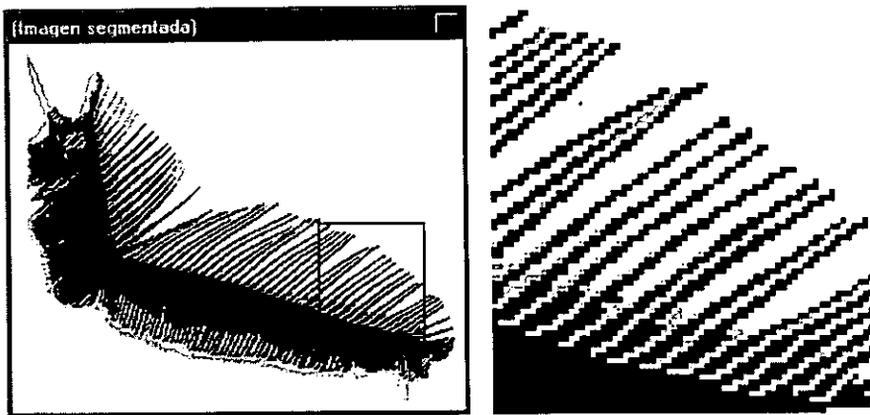


Figura 57. Acercamiento a la imagen segmentada.

En la figura 57 mostramos un acercamiento a una zona de posible conflicto en la imagen segmentada, en donde también se observa el recorrido efectuado por el algoritmo para calcular el perímetro, así como algunos píxeles del conjunto incertidumbre que fueron ignorados como parte del objeto.

Como parte final de este capítulo de resultados, solo restaría mencionar que durante estas etapas de procesamiento y análisis, el usuario tiene a su disposición la selección de imágenes que resulten de interés para guardarlas en formato BMP, o bien copiarlas al portapapeles para incrustarlas posteriormente en algún procesador de texto. Asimismo, los resultados del análisis pueden ser almacenados en formato ASCII, por si es necesario importarlos a otra aplicación.

En un futuro, estas salidas podrían estar asociadas a una base de datos (acopladas después de un análisis y diseño en ese sentido) para mantener actualizada la información y poder hacer uso del módulo de consulta bosquejado en la figura 1 del capítulo 1.



## Discusión

Como ya se ha planteado en los capítulos anteriores, este trabajo contempla primordialmente tres aspectos: Propuesta de un método de segmentación con refinamiento, modificación del algoritmo secuencial de etiquetado para 8-conectividad, análisis, diseño e implementación orientada a eventos de un prototipo de PDI al que se incorporan los algoritmos propuestos así como otros para la mejora de la imagen.

A continuación discutiremos algunos de los aspectos en torno a estos temas, con tendencia a encaminar el desarrollo a futuro que se plantea en el siguiente capítulo.

### 7.1 Método de segmentación propuesto

Como es de esperarse el desempeño de este algoritmo depende completamente de la distribución del histograma de frecuencias. A partir de éste podemos obtener parámetros que ayuden a determinar si la aplicación del algoritmo es factible.

Retomando la figura 8 donde se muestra el histograma modelo de la segmentación y las variables consideradas, haremos las siguientes observaciones:

- El método supone una distribución bimodal del histograma.
- Si  $g \leq f$  no se debería utilizar el método de incertidumbre, pues no se han determinado elementos en el conjunto  $C_2$ . Además de ocasionar un traslape de las clases  $C_0$  y  $C_1$ .
- Si el rango de la clase  $C_2$  es muy pequeño ( $f \approx g$ ) convendría utilizar el método de Otsu, pues segmentaría la imagen en menor tiempo con resultados muy similares.

El método de refinamiento de la segmentación se basa en el fomento del crecimiento de regiones a partir de los valores de una máscara binaria, lo que hace que este método sea susceptible a rotaciones drásticas. Por ello, sería muy conveniente generalizar este refinamiento generando máscaras invariantes a la rotación. Además, en virtud de que esta máscara se ha generado con el conocimiento a priori del tipo de fragmentos a procesar, esta tendría que ser modificada para su uso en otro tipo de imágenes, dependiendo de los objetos que se deseen segmentar.

Un efecto de lo descrito en el párrafo anterior lo observamos en la segmentación de la figura 56, en donde algunas porciones del arco branquial que no corresponden a branquiespinas han sido ignoradas como parte del objeto (parte superior e inferior del arco).

## 7.2 El prototipo

Sin duda las herramientas implementadas en el prototipo deben ser refinadas conforme se detecten necesidades, deficiencias o posibles mejoras. Por el momento es necesario integrar variadas opciones de medición, como áreas o segmentos definidos por el usuario, conteo automático de branquiespinas, calibración espacial, entre otras.

Definitivamente el propósito en este momento no es presentar la implementación completa, sino mostrar los aspectos que se consideraron para su desarrollo, como lo son el ambiente de programación, la interfaz con las bibliotecas del sistema, lenguaje y las definidas por el usuario, así como las consideraciones para el diseño del prototipo (análisis del sistema e interfaz con el usuario).

Reconsiderando la idea original planteada en la introducción acerca del sistema integral para la manipulación de colecciones de imágenes digitales (de órganos de peces, como en este caso), vemos que el prototipo aquí presentado constituye solo una pequeña parte del mismo, destinado a la extracción de características y medidas de estas imágenes. Sin embargo, si visualizamos el sistema completo en un ámbito genérico, podríamos extender este tipo de aplicaciones a otras áreas de investigación en donde se requiera del tratamiento digital de imágenes. Sobre este tema concluiremos en el siguiente capítulo.

# 8 Conclusiones

La segmentación como etapa final del preprocesamiento de imágenes, constituye un proceso crucial que en gran medida determina el éxito del análisis. Es por ello que muchos investigadores han puesto especial atención en la búsqueda de métodos que contribuyan al desarrollo de líneas de estudio en este sentido.

Desde esta perspectiva, una de las propuestas importantes del trabajo aquí presentado lo es el *algoritmo de segmentación*, diseñado para un tipo específico de imágenes. Esta propuesta involucra conceptos de segmentación *basada en umbral* y de *crecimiento de regiones*, adaptándolos según la morfología de los objetos a segmentar. El desarrollo de este algoritmo, así como de otros relativos al preprocesamiento de imágenes en un ambiente orientado a eventos, nos condujo a la implementación del prototipo, mismo que nos permitió comparar el desempeño de los algoritmos propuestos con los de Otsu Nobuyiky (1979, *segmentación basada en umbral*) y Jain Ramesh (1995, *etiquetado de componentes 4-conexas*). En este sentido los resultados obtenidos de este trabajo fueron satisfactorios al lograr mejores resultados en la segmentación y extracción de características básicas, con la restricción de que la *máscara binaria* propuesta para el *refinamiento de la segmentación*, fomenta el crecimiento de regiones en un sentido conocido a priori.

Para llevar a cabo el desarrollo de este trabajo se revisó el entorno del ambiente en el que residiría la aplicación prototipo, lo que nos permitió tener una perspectiva global del problema y no visualizarlo como un conjunto de mecanismos aislados. Partiendo de ese planteamiento originado de un *problema real*, no fue difícil encontrar la justificación de este desarrollo, aunque no estrictamente necesaria desde el punto de vista académico, si motivante al tratar de encontrar soluciones factibles a corto plazo.

Con este trabajo se ha integrado un prototipo que servirá de base para la implementación del sistema en el proyecto del CICIMAR-IPN "Ecología pesquera de *Sardinops caeruleus* (sardina monterrey) en el NW de México".

Aunque hoy en día numerosos sistemas de procesamiento de imágenes abundan en el mercado, debemos mantenernos a la vanguardia no solo en el uso, sino en el desarrollo de nuevas técnicas de propósito específico, obteniendo con ello el control completo sobre los procesos involucrados con áreas específicas, así como enormes posibilidades de crecimiento y adaptabilidad.

De ahí la importancia de los desarrollos propiciados a partir de proyectos que se originan en las instituciones cuya actividad principal se centra en la formación de recursos humanos orientados al desarrollo e implementación de sistemas de información, no solo como requisito académico sino como herramientas que puedan integrarse a la empresa, investigación o docencia, pues será mediante esta retroalimentación que podremos evaluar nuestro trabajo con la finalidad de mejorarlo, buscando aumentar el desarrollo de software de calidad en México, tarea que nosotros como comunidad informática, tenemos la obligación de fomentar, no solo mediante congresos, sino también a través de la distribución de software desarrollado institucionalmente.

## 8.1 Tendencia al futuro

El entorno en el que se desenvuelve el prototipo, como ya se ha mencionado, es el de colecciones de imágenes, especialmente de aquellas que requieren de una manipulación constante y en las que cada vez se realizan mediciones de diversa índole.

Si pensamos en *colecciones costosas* en las que especialistas espacialmente distantes pudieran aportar imágenes, así como opiniones o diagnósticos a petición de sus colegas, tendríamos *colecciones virtuales de escenas en un ambiente distribuido*, para lo cual habría que generar mecanismos comunes de procesamiento, análisis e intercambio de información que permitieran la interacción entre colegas situados en diversos *sitios*.

Como campo de desarrollo futuro en torno a estos mecanismos en nuestro caso de estudio, se proponen las siguientes ampliaciones o mejoras:

- Interacción con una base de datos formal, que permita mantener almacenada la información pictórica, numérica y textual, facilitando el acceso a la información.
- Creación de un módulo estadístico dentro del sistema de información, que permita el análisis de los datos calculados por *Morphos*, de una manera sistemática y con diversas opciones de selección (fechas de captura, áreas de estudio, etc.).
- Con lo anterior crear un acervo que se convierta en parte de una *colección ictiológica virtual*, cuyas imágenes puedan ser procesadas, analizadas e incluso agregadas por personas interesadas en el tema desde cualquier parte del mundo a través de W3.

- Generalización del método de refinamiento de segmentación, mediante la definición interactiva de los patrones de interés, o mediante la definición de otro tipo de heurísticas.

Estas propuestas de ampliación podrían considerarse como módulos complementarios que formen parte del sistema para manipular la *colección virtual de escenas en ambientes distribuidos*, tomando los elementos de análisis de imágenes contenidos en *Morphos* y agregando la funcionalidad que permita su manipulación y consulta en un ámbito global.

En el caso particular del proyecto en el que está inmerso este prototipo, se considera a mediano plazo la interacción con una base de datos, así como una calibración espacial que permita proveer medidas dimensionales del mundo real (milímetros o micras) en operaciones lineales o de área, así como mejoras a la interfaz con el usuario y la disposición de más opciones de medición y almacenamiento de resultados.



## Bibliografía

- [1]: Glasbey C. A. and Horgan C. W., "Image Analysis for the Biological Sciences". Statistics in practice, 1995.
- [2]: Castleman K. R., "Digital Image Processing". Prentice-Hall, 1996.
- [3]: Rosenfeld A. y A. Kak, "Digital Picture Processing", Volume 1 and 2, Academic Press, 1990.
- [4]: Rafael C. González y Richard E. Woods, "Tratamiento digital de imágenes", Addison-Wesley/Díaz de Santos, 1996.
- [5]: Russ J. C., "The image Processing Handbook", CRC Press, 1992.
- [6]: Bow S. T., "Patern Recognition: Aplication to Large Data-Set Problems", Marcel Dekker Inc., 1984.
- [7]: Jähne Bernd, "Digital Image Processing: Concepts, Algorithms and Scientific Applications", Springer-Verlag, 1991.
- [8]: Duff M. J. B., "Cellular Logic Image Processing", Academic Press, 1986.
- [9]: Davies E. R., "Machine Vision: Theory, Algorithms, Practicalities", Academic Press, 1990.
- [10]: Jain Ramesh, "Machine Vision", McGraw-Hill, Inc., 1995.
- [11]: Jain A. K., "Fundamentals of Digital Image Processing", Prentice Hall, 1989.
- [12]: Otsu Nobuyiki, "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man and Cybernetics, Vol SMC-9, No. 1, January 1979.
- [13]: Wright Minturn, "Computer Image Processing and Automatic Counting and Measuring of Fouling Organisms", Computer Biol Med, Vol 21, No. 3, 1991.
- [14]: Jerke Noel, "Visual Basic 4 API how-to", Waite Group Press, 1996.

- [15]: Appleman Daniel, "Visual Basic Programmer's Guide to the Win32 API", Ziff-Davis Press, 1996.
- [16]: Villalobos Hector, "Morfología funcional del aparato filtrador de *Sardinops caeruleus* del Noroeste de México", Tesis de maestría, CICIMAR-IPN, 1998, La Paz, B. C. Sur, México.
- [17]: Ponce de León Soledad, "Análisis comparativo del desarrollo de la morfología funcional de alimentación en la sardina monterrey (*sardinops caeruleus*) y la anchoveta norteña (*Engraulis mordax*)", Tesis de maestría CICESE, 1996, Ensenada, B. C., México.
- [18]: J. Guinea and F. Fernández, "Morphological and biometrical study of the gill rakers in four species of mullet", Departamento de Bioquímica y Fisiología, Universidad de Barcelona, España, 1992.
- [19]: Kevin D. Friedland, "Functional Morphology of the Branchial Basket Structures Associated with Feeding in the Atlantic Menhaden", Virginia Institute of Marine Science, 1985.
- [20]: Grady Booch, "Object Oriented Design With Applications", The Benjamin/Cummings Publishing Company, Inc., 1991.
- [21]: Budd Timothy, "An introduction to Object-Oriented Programming", Addison-Wesley Publishing Company, 1991.
- [22]: Cox Brad, "Object-Oriented Programming: An Evolution Approach", Addison-Wesley Publishing Company, 1991.