



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

“CONTORNOS ACTIVOS EN TRES DIMENSIONES PARA LA SEGMENTACIÓN Y PARAMETRIZACIÓN DE ESTRUCTURAS BIOLÓGICAS”

Tesis para optar al Título de:
Ingeniero Civil en Informática

Profesor Patrocinante:
Sr. Steffen Härtel Gründler
Instituto de Ciencias Biomédicas
Universidad de Chile

Profesor Co-Patrocinante:
Sr. Erick Araya Araya
Instituto de Informática
Universidad Austral de Chile

Profesor Informante:
Sra. Gladys Mansilla Gómez
Instituto de Informática
Universidad Austral de Chile

JORGE FRANCISCO JARA WILDE
Valdivia - Chile
2006

Este Proyecto de Título fue desarrollado y patrocinado por los siguientes laboratorios y proyectos de investigación:

1. Laboratorio de Biofísica y Fisiología Molecular del Centro de Estudios Científicos, Valdivia, Chile, a cargo del Dr. Francisco Sepúlveda Valenzuela, y patrocinado por los proyectos
 - a. FONDECYT 3030065 "Mechanisms of Apoptotic and Necrotic Cell Death: Time resolved Multi-Parameter Analysis of Membrane Related Processes". Investigador responsable: Dr. Steffen Härtel Gründler. Patrocinante: Dr. Felipe Barros Olmedo.
 - b. FONDECYT 1020648 "Patogenia de la muerte necrótica en células hepáticas: disección de mecanismos moleculares efectos". Investigador responsable: Dr. Felipe Barros Olmedo.
 - c. FONDECYT 1030627 "Estudio celular y molecular de la vía basolateral de salida de cloruro en el epitelio intestinal". Investigador responsable: Dr. Francisco Sepúlveda Valenzuela.
2. Programa de Anatomía y Biología del Desarrollo del Instituto de Ciencias Biomédicas, Facultad de Medicina de la Universidad de Chile, Santiago, Chile, a cargo del Dr. Miguel Concha, proyecto FONDECYT 1060890 "Development and application of scaling index methods for the automated quantification of colocalization in fluorescence microscopy" a cargo del Dr. Steffen Härtel Gründler.

Índice

Resumen	1
Summary	2
1 Antecedentes Generales	3
1.1 Introducción	3
1.2 Microscopia Confocal	4
2 Segmentación y Contornos Activos	8
2.1 Visión Computacional en Biología	8
2.2 Segmentación	10
2.3 Contornos Activos	12
2.3.1 Balance de Fuerzas y Mínima Energía	13
2.3.2 Fuerzas de Imagen	13
2.3.3 Formulación	14
2.3.4 Teoría y Aplicaciones	15
3 Antecedentes Específicos	18
3.1 Contexto Biológico	18
3.2 Plataformas de Software	19
3.2.1 IDL	19
3.2.2 SCIAN	20
3.3 Descripción del Proyecto	22
3.3.1 Nombre	22
3.3.2 Motivación	22
3.3.3 Objetivo General	22
3.3.4 Objetivos Específicos	22
3.3.5 Resultados Esperados	22
4 Contornos Activos en Dos Dimensiones	24
4.1 Formulación	24
4.2 Campos de Fuerzas Externas	27
4.2.1 Detectores de Aristas	27
4.2.2 Flujo de Vectores Gradiente – GVF	29
4.2.3 Flujo de Vectores Gradiente Generalizado - GGVF	32

4.3	Parametrización de Fuerzas	34
4.3.1	Coeficiente de Elasticidad	35
4.3.2	Coeficiente de Rigidez	35
4.3.3	Coeficiente de Fuerzas Externas	35
4.3.4	Coeficiente de Viscosidad	36
4.4	Refinamiento y Caracterización	36
4.4.1	Interpolación de la Curva de Contorno	36
4.4.2	Estimación de Curvatura en Dos Dimensiones	37
4.4.3	Experimento de Validación	37
5	Aplicación	39
5.1	Dominio de Aplicación	39
5.2	Segmentación por Filtros	39
5.3	Interpolación de Contornos	40
5.4	Segmentación por Contornos Activos	40
5.5	Cálculo de Parámetros	41
6	Contornos Activos en Tres Dimensiones	42
6.1	Modelos Discretos de Superficies	43
6.1.1	Modelo de Vóxeles	43
6.1.2	Superficie Poligonal	44
6.1.3	Malla Triangular	45
6.1.4	Comparación Experimental de los Modelos	46
6.1.5	Experimento Complementario de Validación	48
6.1.6	Conclusiones	50
6.2	Formulación de Contornos Activos en Tres Dimensiones	50
6.2.1	Fuerzas Internas	51
6.2.2	Implementación	54
6.3	Caracterización de Superficies Tridimensionales	58
7	Conclusiones	58
7.1	Cumplimiento de Objetivos	58
7.1.1	Objetivo General	58
7.1.2	Objetivos Específicos	58
7.2	Aspectos Notables, Ventajas y Limitaciones	59

7.3	Trabajos Futuros	60
7.4	Difusión y Aplicaciones	60
8	Referencias	63
8.1	Artículos	63
8.2	Libros	64
8.3	Charlas de Conferencia	65
8.4	Presentaciones de Póster	68
8.5	Tesis	68
8.6	Direcciones Web	69

Anexos

1	Cálculo Variacional	72
2	Clases y Métodos Implementados	77
3	Color y Opacidad en Imágenes Digitales	109
4	<i>Scripting</i> para Deconvolución de Imágenes	116
5	Publicaciones	124

Índice de Tablas y Figuras

Tabla/Figura	Nombre	Pág.
Fig. 1.2-1	Sistema de microscopia confocal.	5
Fig. 1.2-2	Imagen de microscopia confocal de fluorescencia.	5
Fig. 1.2-3	Visualización de un <i>z-stack</i> en tres dimensiones.	6
Fig. 2.1-1	Diagrama de flujo de datos para el procesamiento de imágenes.	9
Fig. 2.1-2	Análisis de volumen en estructuras celulares mediante reconstrucción en tres dimensiones.	10
Fig. 2.2-1	Comparación entre filtros de umbral y contornos activos.	12
Tabla 2.2.3-1	Resumen comparativo de contornos activos paramétricos y geométricos.	15
Tabla 3.3.5-1	Objetivos específicos y resultados del Proyecto.	23
Fig. 4.1-1	Elasticidad y rigidez en una curva discreta.	25
Fig. 4.2.1-1	Niveles de intensidad y gradientes asociados a una imagen.	28
Fig. 4.2.2-1	Campos de gradientes.	29
Fig. 4.2.3-1	Campos de gradiente y flujos de gradiente GVF/GGVF.	33
Fig. 4.3-1	Ajuste de contornos activos en dos dimensiones.	34
Fig. 4.3.1-1	Efectos de la elasticidad de contornos.	35
Fig. 4.3.2-1	Efectos de la rigidez de contornos.	35
Fig. 4.3.3-1	Efectos de la ponderación de fuerzas externas.	35
Fig. 4.4.3-1	Aumento en resolución de contornos mediante índices de curvatura.	37
Fig. 5.1-1	Imagen de microscopia.	38
Fig. 5.2-1	Segmentación por umbrales de intensidad.	38
Fig. 5.2-2	Ajuste de contornos activos e interpolación de puntos.	40
Fig. 6.1-1	Apilamiento de imágenes en un <i>Z-stack</i> .	42
Fig. 6.1.1-1	Modelo de vóxeles.	43
Fig. 6.1.2-1	Superficie poligonal.	44
Fig. 6.1.3-1	Malla triangular.	44
Fig. 6.1.4-1	Valores de área calculados para modelos de superficies esféricas.	46
Fig. 6.1.5-1	Comparación de modelos de superficies cilíndricas.	48
Fig. 6.2.1-1	Elasticidad en superficies activas.	52
Fig. 6.3-1	Evolución de curvatura en superficies activas	57

Resumen

En el campo de procesamiento de imágenes, las técnicas de segmentación permiten identificar y definir regiones de interés (*regions of interest*, ROIs) dentro de una imagen. Aplicada a la biología y la biofísica, la segmentación de estructuras en imágenes microscópicas es indispensable para su posterior análisis morfológico.

Este trabajo aplica métodos para la segmentación y caracterización morfológica de estructuras biológicas, mediante contornos activos paramétricos que interactúan con campos vectoriales de flujo de gradientes (*gradient vector flow*, GVF, *generalized gradient vector flow*, GGVF). Los contornos activos son modelos matemáticos que controlan la deformación de curvas (dos dimensiones) o superficies (tres dimensiones), mediante la acción de parámetros intrínsecos (fuerzas internas), y campos vectoriales calculados en función las variaciones de intensidades en las imágenes (fuerzas externas). Se define un funcional de energía interna y externa que se busca minimizar mediante la ecuación de Euler-Lagrange; añadiendo un parámetro de tiempo al modelo se aproxima una solución con métodos iterativos. Se implementaron contornos activos en dos dimensiones, para luego extender su formulación y aplicación como superficies activas en ROIs tridimensionales.

Se estableció un criterio para optimizar la resolución de contornos en dos dimensiones, en base a la evaluación de curvatura, puntos de inflexión y perímetro. Para tres dimensiones se evaluaron mallas de superficie, en función de la precisión en el cálculo de área y regularidad de curvatura.

Los métodos desarrollados fueron integrados a una aplicación computacional para procesamiento de imágenes, combinándose con otras técnicas de segmentación, reconstrucción, visualización y parametrización de estructuras biológicas.

Summary

In the field of image processing, the segmentation techniques allow to identify and to define regions of interest (ROIs) within an image. Applied to Biology and Biophysics, the segmentation of structures in microscopic images is indispensable for its later morphologic analysis.

This work applies to methods for the segmentation and morphologic characterization of biological structures, by means of parametric active contours that interact with gradient vector flow fields (GVF, generalized gradient vector flow fields, GGVF). The active contours are mathematical models that control the deformation of curves (two dimensions) or surfaces (three dimensions), by means of the action of intrinsic parameters (internal forces), and vector fields calculated as a function of the variations of intensities in the images (external forces). A functional of internal and external energy is defined that is aimed to be minimized by means of the equation of Euler-Lagrange; adding a time parameter to the model a solution with iterative methods can be approximated. Active contours in two dimensions were implemented first, and then extended in their formulation and application as active surfaces in three-dimensional ROIs.

A criterion was settled down to optimize the resolution of contours in two dimensions, on the basis of the evaluation of curvature, inflection points and perimeter. For three dimensions surface meshes were evaluated, based on the precision in the calculation of area and regularity of curvature.

The developed methods were integrated to a computational application for image processing, being combined with other techniques of segmentation, reconstruction, visualization and parametrization of biological structures.

Capítulo 1: Antecedentes Generales

1.1 Introducción

En las últimas décadas, el procesamiento computacional en imágenes ha cobrado relevancia dentro de diversas áreas del quehacer científico. Imágenes de microscopia, rayos X, resonancia magnética, ultrasonido, entre otras, requieren de técnicas que permitan extraer de ellas información significativa e interpretable por los profesionales del área, con el mayor nivel de precisión y detalle que sea posible lograr. Se han desarrollado distintos métodos para clasificar, visualizar, realizar seguimientos y cuantificar una variedad de estructuras anatómicas en dos y tres dimensiones [Kim03], desde niveles macroscópicos hasta microscópicos, con diversos ejemplos de aplicación [Caz95, Mci96, Whi01, Lin03, Liu03].

En el campo de la biofísica y la biología celular, el procesamiento de imágenes ha permitido profundizar el estudio de diversas estructuras biológicas [Boi96], entregando información que de otro modo sería más difícil o costosa, si no imposible de obtener. Ejemplo de esto lo constituyen los análisis de morfología, textura, topología o movimiento de células, tejidos y órganos; además de reacciones químicas en sus distintos niveles de organización.

La biología celular concentra su estudio en estructuras micrométricas, las que son observadas mediante sistemas de microscopia, bajo condiciones experimentales controladas. A este nivel de resolución, cobran relevancia las características e interacciones físicas micro y macro moleculares, asociadas a las distintas estructuras y al sistema mismo de observación. Los modelos y técnicas experimentales derivadas de la física aparecen como un enfoque que permite elucidar información sobre las estructuras y sus interacciones, como partes de sistemas más complejos.

Como en todo experimento de observación, se requiere de procedimientos que introduzcan el menor nivel posible de alteraciones en las estructuras celulares bajo estudio. Para los sistemas de microscopia, resulta útil aplicar técnicas de marcado sobre las estructuras que se han de observar: mediante la aplicación de compuestos específicos, estas estructuras se hacen distinguibles al microscopio. En este contexto aparece la **microscopia confocal de fluorescencia**, capaz de medir concentraciones de moléculas fluorescentes al interior de las células, como una alternativa viable para estudios de morfología y dinámica de comportamiento.

1.2 Microscopia Confocal

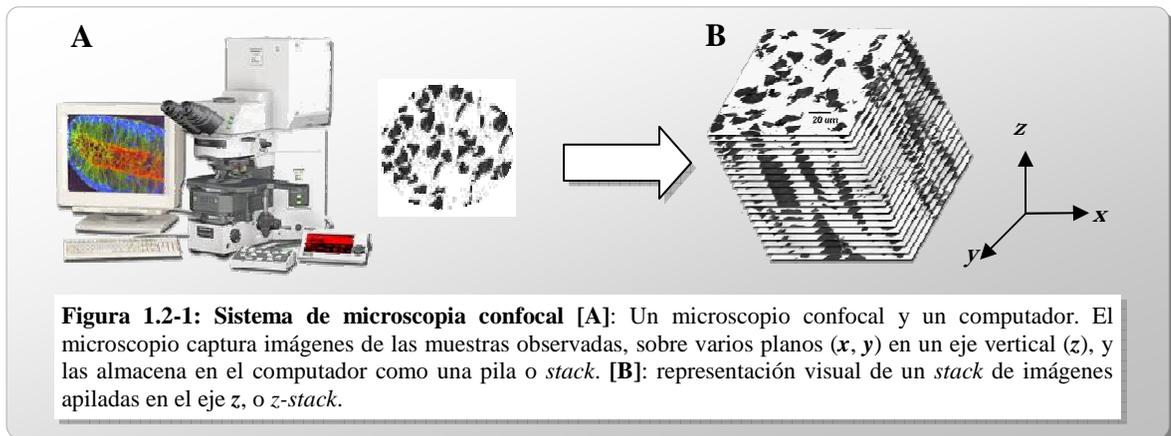
El estudio de la dinámica del comportamiento celular, requiere de procedimientos que permitan observar a las estructuras de interés en instantes de tiempo dados. Además, para el análisis morfológico se hace necesario disponer de herramientas que permitan capturar información en distintos puntos dentro del volumen o superficie observada, durante los distintos instantes de observación. Si bien la microscopia electrónica ofrece imágenes de muy alta resolución (a nivel atómico, $\sim 10^{-10}$ m), requiere fijar las estructuras a ser observadas sobre un medio, causando su muerte y, en consecuencia, impidiendo un seguimiento temporal. Es por esto que la microscopia confocal aparece como una alternativa para seguir la dinámica del comportamiento celular, sin causar daño letal, puesto que no requiere la fijación de la muestra.

Aunque sus fundamentos se establecieron en 1953 [Min88], sólo a finales de la década de 1980 la tecnología láser permitió masificar el uso de los microscopios confocales; desde entonces la microscopia confocal se ha convertido en un estándar, debido tanto a la relativa facilidad en la preparación de muestras, como al desarrollo de técnicas para el refinamiento de las imágenes adquiridas [URL1, URL2]. Prueba de esto es la extensa aplicación de este tipo de sistemas en el área de la biología celular, en líneas de investigación tales como el estudio de procesos en membranas, cambios de morfología y organización de estructuras intracelulares e intercelulares [Alv03, Alv04a/b, Fan02/06, Har03, Har04a/b/c/d, Har05a/b, Har06a/b, Roj04, Oya03, Tul06, Ver05]. Además, recientes trabajos han extendido el concepto fundamental de la microscopia confocal, desarrollando mejoras en la resolución [Hel04].

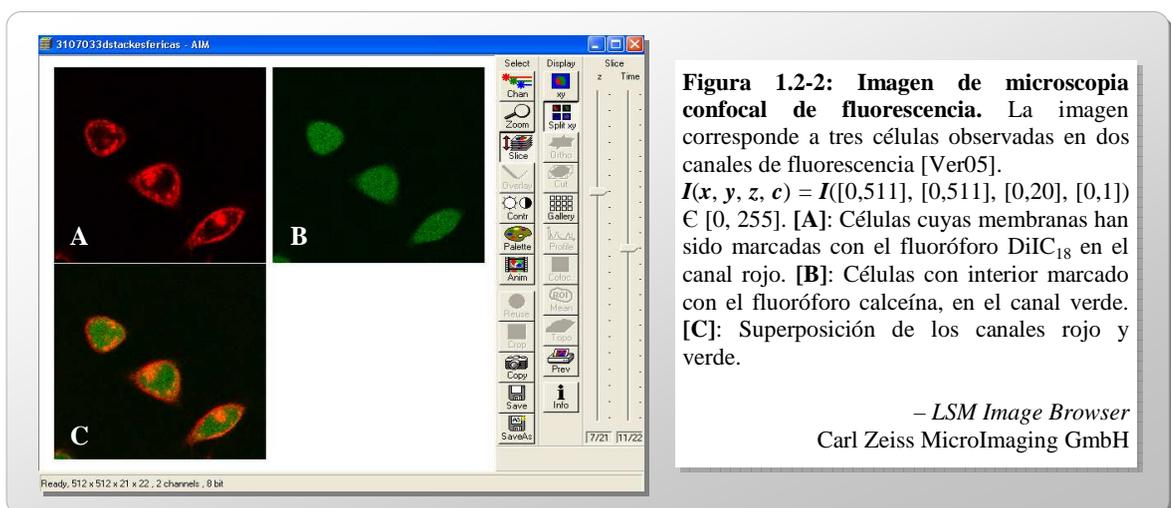
La microscopia confocal, si bien presenta una resolución inferior a la de la microscopia electrónica, tiene la ventaja de que permite tomar imágenes de muestras en cortes tanto verticales (z) como horizontales (x , y). La resolución de la microscopia confocal es del orden de nanómetros (10^{-9} m), con cortes verticales relativamente finos en z (~ 500 nm) y con una mejor resolución en el plano horizontal (~ 150 nm), debido a la difracción de las ondas lumínicas en el sistema [Hel92, Egn98, Sch98]. Aún así, el nivel de resolución de los microscopios confocales ha resultado ser suficiente para el estudio dinámico del comportamiento celular.

La microscopia confocal se clasifica en la categoría de sistemas de microscopia óptica, lo que significa que basa su funcionamiento en la emisión y captación de luz. Los microscopios confocales emplean un sistema de iluminación con barrido de rayo láser de

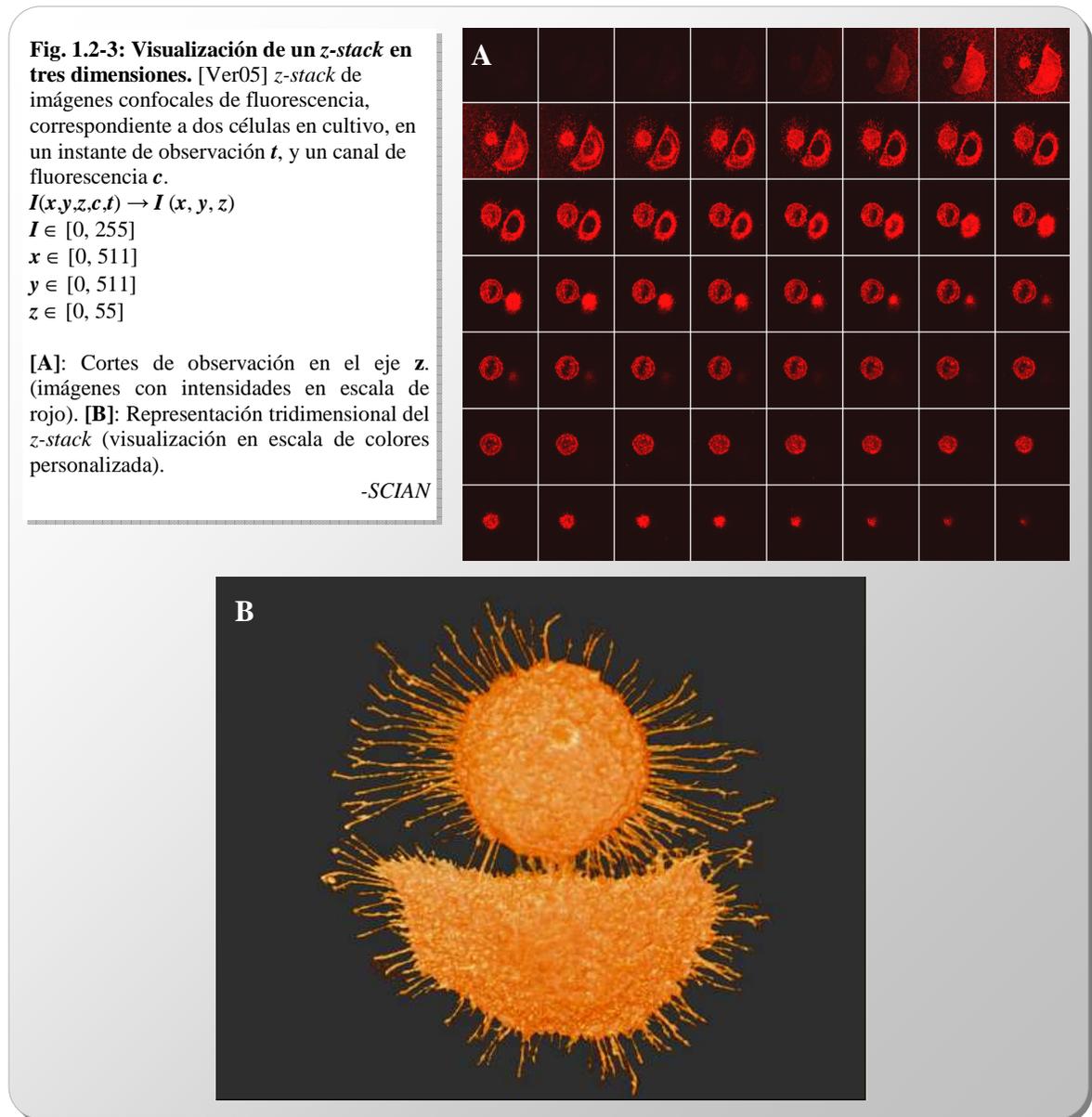
alta precisión (LSM, *Laser Scanning Microscopy*), que les permite capturar información en distintos puntos asociados espacialmente al objeto que se observa [Wil03]. Un sistema de espejos mueve al láser a través del objeto, iluminando un solo punto por vez. Se registran los datos de cada punto de la muestra recorrida con este rayo móvil y se guardan como una pila o *stack* de imágenes en un computador [URL3].



Para poder identificar a los elementos o estructuras de interés en las imágenes del microscopio, se les aplican marcadores fluorescentes denominados fluoróforos. Los fluoróforos son compuestos químicos que absorben ondas de luz de una cierta longitud de onda, para luego emitir ondas de otra longitud, en una banda de frecuencia (canal de color) específica y reconocible dentro del espectro; la frecuencia de la onda emitida es propia de cada compuesto, permitiendo identificar a los elementos al interior del volumen celular, discriminando entre éstos por el canal de fluorescencia aplicado a cada uno [Wil03]. Además, la intensidad de fluorescencia observada (en un rango determinado) será proporcional a la concentración del fluoróforo [Fin98]; esto permite observar y analizar las distribuciones de concentración y/o gradientes de sustancias dentro de las estructuras bajo observación, además de la morfología de las propias estructuras.



La detección de intensidades de fluorescencia no sólo se realiza en un plano de observación (x, y), también se aplica a distintos niveles de altura espacial en la muestra (z). Mediante el sistema de control del láser es posible generar imágenes múltiples (para los distintos canales de fluorescencia) a diferentes profundidades, permitiendo luego reconstruir estructuras tridimensionales, como se ilustra en la figura 1.2-3.



Debido a restricciones propias del sistema físico de observación, debe eliminarse la influencia de regiones fuera de foco en cada punto, restándolas de cada imagen para obtener una definición óptima¹ [Fin98, Kem99]. Posteriormente, diversos filtros permiten recuperar y mejorar información de la imagen que ha sido distorsionada por otros factores durante la adquisición de datos [Ver05].

¹ Este tipo de procesos debe ser realizado mediante software, dada la gran cantidad de datos de cada imagen requeridos para los cálculos.

El proceso de adquisición de datos en microscopia confocal puede entenderse como la conversión de información desde un espacio continuo (real) a uno discreto (digital), que captura y cuantifica datos asociados a características físicas de una o más estructuras biológicas en observación. Como resultado del proceso se tiene un *stack* de imágenes, es decir, un conjunto finito de estructuras de datos, que constituye una representación discreta de las estructuras observadas.

La microscopia confocal entrega imágenes digitales de las muestras biológicas, cuantificando la intensidad de fluorescencia (I) en tres dimensiones espaciales (x, y, z), múltiples canales de fluorescencia (c), y a intervalos de tiempo (t). Un *stack* se representa por un arreglo multidimensional, con una función de intensidad asociada $I = I(x, y, z, c, t)$, siendo

I : intensidad $\in [0, 2^D]$ (D = número de bits de profundidad, en escala de grises)²

c : canal de fluorescencia

t : instante de tiempo

$(x, y, z) \in [0, dim_x-1] \times [0, dim_y-1] \times [0, dim_z-1]$: dimensiones espaciales en píxeles

² En el presente Proyecto se trabajó sobre imágenes en escalas de grises, en que la cantidad de bits para cada píxel determina los niveles de gris posibles de representar.

Capítulo 2: Segmentación y Contornos Activos

En este capítulo se introducen los conceptos fundamentales de la llamada segmentación de regiones de interés (*regions of interest*, ROIs), con aplicaciones a imágenes de microscopía. Se presentan los contornos activos como modelos para la representación de tales regiones, y que incorporan información morfológica, mediante parámetros asociados a su geometría.

2.1 Visión Computacional en Biología

Entre las diversas aplicaciones de la informática a la biología, la visión computacional constituye una herramienta de importancia para obtener información sobre características físicas de las estructuras bajo estudio.

La visión computacional puede definirse como “la extracción de conocimiento involucrado en una realidad de tres dimensiones, desde una o más proyecciones de esa realidad” [Wat98].

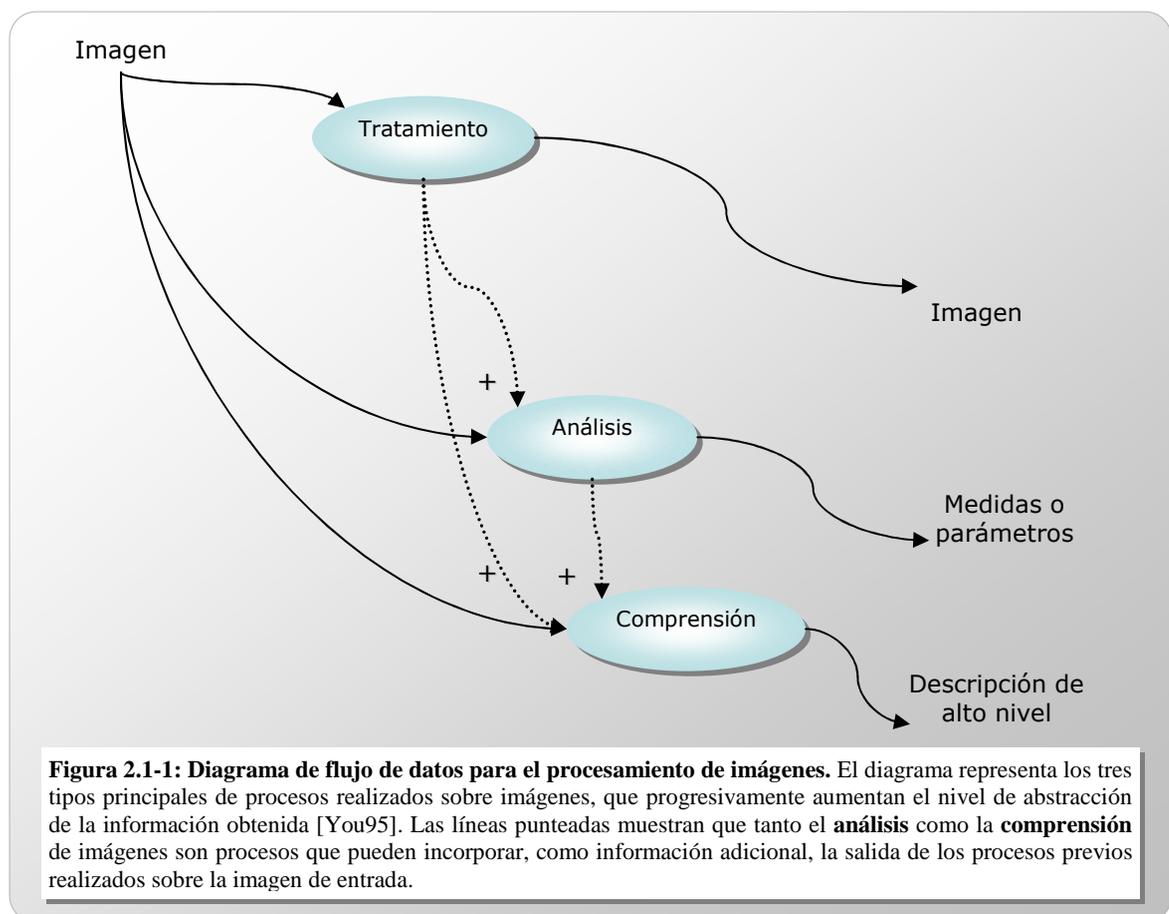
En el estudio morfológico de estructuras celulares, se requiere en primer lugar definir las e identificarlas, sean éstas células o bien algunos de sus componentes, como membranas u organelos internos. A continuación, es necesario describir o caracterizar a estas estructuras, según su forma de representación, posición y orientación espacial, etc. En base a esta información se pueden determinar propiedades asociadas, por ejemplo, al volumen y/o la superficie. Finalmente, en algunos casos se habrán de detectar y seguir los cambios que puedan experimentar las estructuras observadas, en intervalos de tiempo determinados.

Para trabajar con problemas de este tipo, la tecnología digital existente permite el procesamiento de señales multidimensionales (como las imágenes) desde circuitos digitales simples hasta sistemas de cómputo múltiple. Los procesos de manipulación y tratamiento para esta clase de información corresponden al **campo de procesamiento de imágenes**. Dentro de este campo es posible establecer una clasificación funcional de los distintos tipos de tareas o procesos, según su nivel de abstracción [You95]:

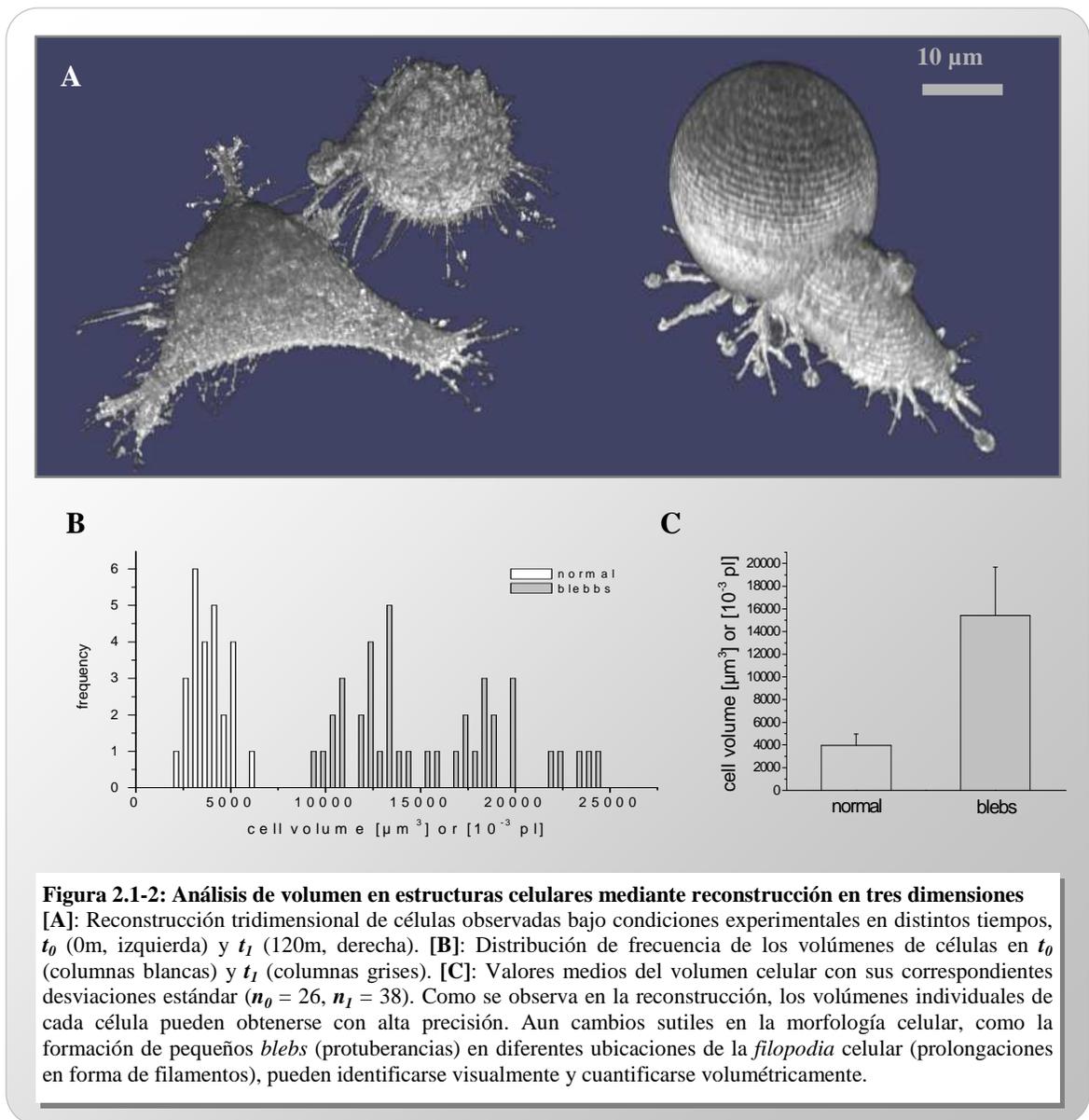
- Tratamiento de imágenes: busca recrear una imagen lo más fiel a la realidad que representa, quitando la influencia de elementos ajenos introducidos en procesos como adquisición o almacenamiento.

- Análisis de imágenes: apunta a obtener, a partir de una imagen, indicadores cuantitativos o cualitativos, que describan sus características de interés.
- Comprensión de imágenes: corresponde a un nivel de abstracción mayor, con la inferencia de conocimiento a partir de la información disponible sobre una imagen, para describirla e interpretarla en del contexto del problema que lo requiere.

El diagrama de la figura siguiente muestra cómo estos procesos se relacionan con el fin de caracterizar e interpretar información, a partir de una o más imágenes de entrada. Cada proceso permite sucesivamente aumentar en calidad, cantidad y complejidad, el nivel de información posible de extraer de los datos existentes.



La obtención y cuantificación de datos a nivel científico, requiere de técnicas que permitan identificar y parametrizar a las estructuras de interés a partir de las imágenes obtenidas. Aplicado a la microscopia confocal, el proceso involucra el análisis de imágenes en dos o tres dimensiones espaciales (x, y, z), además de los distintos canales (c) y tiempos (t) [Ver05]. La figura 2.1-2 ilustra una aplicación típica en tres dimensiones espaciales para el cálculo del volumen celular, para un estudio de cambios morfológicos en células de cultivo [Har04d].



2.2 Segmentación

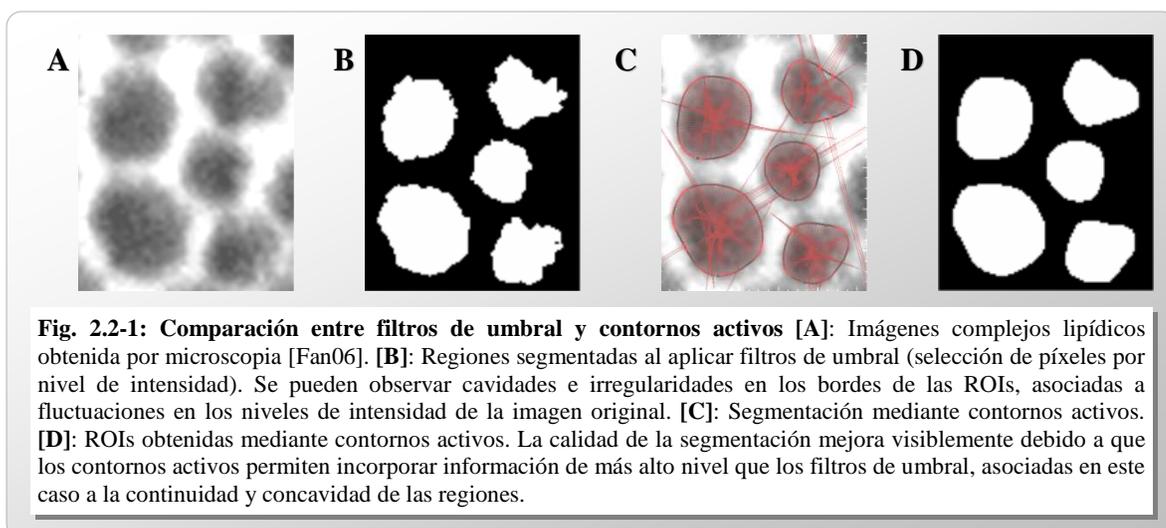
La segmentación es el proceso que separa objetos o regiones de interés que tengan algún significado dentro de una imagen [Wat98, Bar02, Oya03, Gon02]. Mediante criterios de proximidad, similitud o continuidad se busca construir un conjunto de píxeles, regiones o contornos similares en intensidad, color, textura, u otra característica de interés [Oya03].

La segmentación llega a definir un conjunto de estructuras o descriptores de las ROIs segmentadas, que permite diferenciarlas del resto de la imagen. En computación, las ROIs deberán de representarse por estructuras de datos accesibles para las aplicaciones, de modo que sea posible almacenarlas y analizarlas en forma posterior.

Según su formulación [Mar02a], se pueden distinguir entre métodos de segmentación basados en:

1. Agrupamiento de píxeles: separan una imagen en dos conjuntos de píxeles (fondo/ROIs), distinguiendo entre criterios de
 - Transición: operadores matemáticos que destacan los cambios en las intensidades de los píxeles en una imagen, como gradientes de intensidad o los operadores de Sobel y de Laplace.
 - Homogeneidad: agrupan píxeles buscando características comunes, como la afinidad en los niveles de intensidad.
2. Morfología: buscan en la imagen formas de un modelo geométrico previamente definido, como segmentos de rectas, circunferencias, etc.
3. Modelos deformables o contornos activos [Kas88, Cas97]: consisten en el ajuste iterativo de funciones que delimitan o demarcan las regiones de interés. Las curvas en evolución (*snakes*) y los métodos basados en conjuntos de nivel (*level sets*) corresponden a este tipo de métodos.

En general, las técnicas de segmentación 1 y 2 son útiles cuando el objeto y el fondo son relativamente uniformes, y la presencia de ruido u otros factores de distorsión ajenos son marginales [Mar02b]. En escenarios más complejos se requiere realizar un proceso adicional, que **incorpore información de un nivel más alto, para interpretar, reconocer y/o clasificar los objetos segmentados previamente**, y así darles conectividad, o eliminar falsos positivos y negativos, además de **definir e identificar a cada región con una frontera definida** (la segmentación por morfología requiere determinar *a priori* la geometría las ROIs a detectar). Se trata de técnicas de aparición más reciente, más robustas, pero que requieren de un mayor tiempo de cálculo: son los llamados **modelos deformables o contornos activos**.



2.3 Contornos Activos

“Las bases matemáticas de los modelos deformables representan la confluencia de geometría, física y teoría de aproximaciones. La geometría sirve para representar la forma del objeto, la física impone restricciones sobre cómo la forma puede variar en el espacio y el tiempo, y la teoría de aproximación óptima proporciona bases formales de mecanismos para ajustar los modelos a los datos medidos” [Mci96].

Los modelos de contornos activos proporcionan una representación matemática para cada región de interés, que se ajusta dinámicamente sobre la imagen. Este ajuste se determina por fuerzas y restricciones que determinan la evolución de los contornos, en función de las características la imagen [Kas88].

Los contornos activos modelan las fronteras entre una región de interés, el fondo y las demás regiones en la imagen [Mar02b]. Permiten extraer los contornos de las regiones de interés a partir de modelos que utilizan información *a priori* de su forma. Estas técnicas resultan más robustas frente a la presencia de ruido y otros factores externos, y permiten segmentar imágenes más complejas, en comparación con los métodos de segmentación por filtros. Esto porque el enfoque de los modelos deformables permite incorporar un nivel mayor de conocimiento para guiar la detección y caracterizar a las estructuras segmentadas (Capítulo 4).

El nombre de modelos deformables se debe primero a la teoría física de la elasticidad: los contornos se consideran como cuerpos elásticos que responden a fuerzas aplicadas, estando sujetos a ciertas restricciones.

La geometría de modelos deformables permite una amplia cobertura de formas, empleando diversas representaciones geométricas que involucran varios grados de libertad (como las curvas o superficies *spline*, por ejemplo). El modelo permanece controlable, sin embargo, porque los grados de libertad se restringen por principios físicos, que le otorgan un comportamiento intuitivamente significativo sobre el sustrato geométrico.

2.3.1 Balance de Fuerzas y Mínima Energía

El ajuste de los contornos activos se basa en un criterio de **balance de energía**, en términos de las distintas propiedades definidas, tanto del contorno como de la imagen. Se busca **minimizar la energía total en el contorno**, al deformarlo desde un estado inicial, llegando a un nivel límite mínimo, y que corresponderá al ajuste óptimo de la curva [Kas88]. En la búsqueda de este ajuste se derivan los términos del funcional de energía, para pasar (en el equivalente físico del modelo) desde un balance de energías a un **balance de fuerzas**. El enfoque del balance de fuerzas permitirá aproximar la solución óptima por la vía iterativa, a través de métodos de cálculo numérico (Capítulo 4).

En primer lugar, es necesario definir las energías que afectarán al contorno, como restricciones de forma, suavidad, concavidad o simetría. Estos términos se ponderan para formar el denominado **funcional de energía del contorno**. Este funcional consta de términos de **energía interna**, o propiedades intrínsecas del contorno, y de **energía externa**, o propiedades de la imagen. De acuerdo a la ponderación de estos términos es que el modelo deberá ajustarse dinámicamente, partiendo desde una forma o estado inicial. La energía del contorno variará monótonicamente según éste se deforma, evaluando los términos de energía interna y externa definidos por el funcional.

Los contornos activos son modelos cuya formulación se puede extender a espacios de distinta dimensión. Así, mientras un contorno es definido por una curva en el espacio bidimensional (Capítulo 4), lo será por una superficie para el tridimensional (Capítulo 6).

2.3.2 Fuerzas de Imagen

El ajuste de un contorno activo, además de considerar parámetros de morfología (intrínsecos a la formulación), debe ponderar los indicadores de transición entre ROIs y fondo de una imagen. Es por esto que se definen funciones **detectores de bordes**, que cuantifican las transiciones o cambios entre los niveles de intensidad para los píxeles de una imagen. Los gradientes de intensidad son el tipo de detector más simple y frecuentemente empleado.

Para el caso de contornos activos paramétricos con balance de fuerzas, se emplean campos de vectores para representar a las fuerzas externas, que ponderan los datos de la imagen en el ajuste. Estos campos se definen de modo tal que la deformación iterativa de los contornos ocurra en dirección de los bordes de ROIs (transiciones de intensidad). Recurriendo a los niveles de intensidad en la imagen, los vectores gradiente se pueden emplear como campos de fuerzas externas, y más aun, conformar un punto de partida para la generación de campos vectoriales más complejos, que permitan optimizar el ajuste de los contornos activos (Capítulo 4).

2.3.3 Formulación

Desde [Kas88], diversos enfoques han ampliado la visión y aproximación a los modelos de contornos activos, extendiendo su definición tanto en las funciones de contorno como las fuerzas que actúan sobre éstas, además de las condiciones que determinan su convergencia.

Partiendo desde su formulación matemática, se distinguen principalmente los contornos **activos paramétricos** y los **contornos activos geométricos**.

- **Contornos activos paramétricos:** son representados en forma explícita por funciones de contorno parametrizadas (curvas o superficies); su deformación se rige por propiedades intrínsecas, formuladas explícitamente [Kas88, Xu98a].
- **Contornos activos geométricos:** son representados implícitamente como conjuntos de nivel (*level sets*) de una función dinámica con una dimensión más, y cuya deformación depende explícitamente de sus propiedades geométricas [Osh88, Cas97].

La tabla siguiente resume los principales aspectos que permiten distinguir ambos tipos de formulación de los contornos activos.

Tabla 2.3.3-1: Resumen comparativo de contornos activos paramétricos y geométricos.

Contornos Activos Paramétricos	Contornos Activos Geométricos
<p>a. Función de contorno explícita.</p> <p>Las ROIs resultantes son directamente interpretables (no requieren de procesamiento posterior).</p>	<p>a. Función de contorno implícita como conjunto de nivel de una función envolvente, de dimensión superior.</p> <p>Las ROIs y funciones de contorno resultantes deben recuperarse a partir de la función envolvente, en un proceso posterior.</p>
<p>b. Cada contorno permite definir sólo una ROI (se requiere un contorno inicial por cada ROI).</p>	<p>b. Flexibilidad topológica (segmentación indistinta para una o varias ROIs, no existen problemas de separación entre puntos de contorno).</p>
<p>c. La estabilidad numérica depende de la forma del contorno (cruces sobre sí mismo, ponderación de fuerzas).</p> <p>El contorno debe ajustarse para mantener regularidad (espaciado entre puntos) en cada iteración.</p>	<p>c. Estabilidad numérica. Requiere de un parámetro de regularización cada cierto número de iteraciones.</p>
<p>d. Implementación varía según número de dimensiones.</p>	<p>d. Extensión directa a varias dimensiones.</p>
<p>e. Tiempo de cómputo depende directamente de los números de punto del contorno.</p>	<p>e. Tiempo de cómputo depende directamente del tamaño de la imagen de entrada.</p>

2.3.4 Teoría y Aplicaciones

Distintos trabajos se han realizado en áreas como la inteligencia artificial, reconocimiento de patrones y tratamiento de imágenes médicas, para desarrollar y aplicar mecanismos de segmentación por contornos activos. A continuación se resumen las principales ideas desarrolladas en la literatura, revisadas en el presente Proyecto.

Contornos Activos Paramétricos

- [Kas88] sienta las bases teóricas de los modelos deformables y realiza una formulación paramétrica para las funciones de contorno. Se trata de curvas que se ajustan dinámicamente a fuerzas definidas sobre una imagen, a la vez que ponderan parámetros de comportamiento propios; este ajuste se determina por un criterio de minimización de energía, traducido en ecuaciones de balance de fuerzas. Los enfoques derivados de esta formulación son conocidos posteriormente como **contornos activos paramétricos**.
- [Coh91] incorporó el concepto de fuerzas externas de presión, que permiten mejorar la convergencia del proceso de ajuste.

- [Xu98a] introdujo los campos de vectores gradiente, un tipo particular de fuerzas externas para los contornos activos paramétricos. Una formulación más generalizada es propuesta en [Xu98b]. Los campos vectoriales son una alternativa a las funciones de gradientes, que mejoran la convergencia para contornos que no se inicializan próximos a la solución óptima. Posteriormente, [Xu00] prueba la optimalidad global de su formulación para el campo de vectores.

Contornos Activos Geométricos

- [Osh88] presentó un modelo físico aplicable a fenómenos tales como la propagación del fuego, el crecimiento de cristales, y otros que involucran el desplazamiento de superficies, basándose en su curvatura. Este trabajo plantea los fundamentos matemáticos de la llamada teoría de evolución de frentes.
- [Cas93] y [Mal95] presentan una formulación de contornos activos con ecuaciones diferenciales parciales, basadas en parámetros de geometría y la teoría de evolución de frentes, que permite encontrar varios contornos simultáneamente. Posteriormente en [Cas97] se introduce la noción de **contornos activos geodésicos o geométricos** como una generalización de los modelos deformables.

Contornos Activos Paramétricos y Geométricos

- La formulación de los contornos activos geodésicos de [Cas97] relaciona a los contornos basados en energía (paramétricos) con los contornos activos geométricos, basados en la teoría de evolución de frentes. La solución para el problema de minimización de energía de [Kas88] se corresponde con la de curvas de mínima distancia, dentro de un espacio inducido a partir de una imagen (modelado a partir de la formulación de [Osh88]).
- [Xu01] aborda distintos principios y parámetros involucrados para formas equivalentes de contornos activos paramétricos y geométricos, apuntando restricciones en la formulación de los modelos paramétricos al momento de establecer su correspondiente modelo geométrico. Estas restricciones están dadas principalmente por el carácter explícito de parámetros o propiedades físicas asociadas a los contornos, además de la importancia de mantener fuerzas asociadas directamente con la función de contorno.
- En [Osh06] se recopilan diversas revisiones para los principales modelos desarrollados hasta la fecha, en los tópicos de modelos deformables (clásicos, de topología adaptativa,

formulaciones generalizadas implícitas y explícitas) y detección de bordes (eliminación de ruido e implementación numérica).

Otros Enfoques

- [Jal04] presenta un modelo de aplicación de contornos activos basado en sistemas de partículas. Este tipo de modelo, derivado de los modelos de optimización de enjambres de partículas [Ken95], presenta una gran rapidez de cómputo, pero carece de mecanismos para manejar las propiedades geométricas de los contornos, siendo necesario aplicar algoritmos de reconstrucción en forma posterior.

Desarrollos de software

La aplicación de los contornos activos se traduce en algoritmos computacionales para realizar segmentación sobre imágenes de todo tipo. En el marco del presente Proyecto, las siguientes aplicaciones de software proporcionaron una base para el desarrollo en IDL de rutinas de segmentación por contornos activos para dos dimensiones:

- La formulación de contornos activos de [Xu98a, Xu98b, URL6], aplicada en MATLAB, sirvió como base para desarrollar e implementar los métodos de segmentación en dos dimensiones, así como para identificar las condiciones elementales de la implementación y los algoritmos de cálculo.
- En [URL7/8], a partir de la versión desarrollada en MATLAB [Xu98a], se implementó una versión orientada a objetos de los contornos activos en dos dimensiones, con el lenguaje IDL. Esta versión se empleó en el presente Proyecto para construir la primera versión de un filtro de segmentación de contornos activos, y fue el punto de partida para el desarrollo de una versión orientada a objetos.

Capítulo 3: Antecedentes específicos

3.1 Contexto Biológico

El presente Proyecto se desarrolló en el área de Biofísica y Fisiología Molecular del Centro de Estudios Científicos de Valdivia (CECS). En esta área se estudian fenómenos que ocurren en la membrana celular, frontera que separa a la célula de su medio ambiente. Se experimenta con proteínas que permiten a la célula transferir información a través de esta membrana. Mediante diversas técnicas, entre las que se encuentra la microscopia confocal de fluorescencia, se estudia cómo la estructura y función de estas proteínas afectan la fisiología celular y la del organismo [URL4].

Dentro de las líneas de investigación existentes en esta área aparece el estudio de la muerte celular. Aquí interesa conocer cómo los mecanismos de transporte de iones y sustratos metabólicos a través de la membrana celular son afectados e inciden en la muerte patológica y fisiológica de células de mamífero. En este marco, el Proyecto se aplicó al estudio de cambios morfológicos durante la muerte celular, en el proyecto FONDECYT post-doctoral 3030065 “Mechanisms of apoptotic and necrotic cell death: time resolved multi-parameter analysis of membrane related process”³.

Posteriormente, el Proyecto se enmarcó dentro del proyecto FONDECYT regular 1060890 “Development and application of scaling index methods for the automated quantification of colocalization in fluorescence microscopy”⁴, desarrollado en Programa de Anatomía y Biología del Desarrollo, en el Instituto de Ciencias Biomédicas (ICBM) de la Facultad de Medicina de la Universidad de Chile. El marco de aplicación en esta etapa estuvo dado por el análisis morfológico de estructuras cerebrales, durante el desarrollo embrionario de ejemplares de pez cebra; en este ámbito, la aplicación de las herramientas implementadas en el Proyecto permitieron resolver estructuras y complejos de nivel subcelular, celular e intracelular.

Investigaciones recientes en ambas líneas de investigación se encuentran en diversas publicaciones. La segmentación de estructuras y cuantificación de características morfológicas se han aplicado en casos de dos dimensiones [Fan02/06, Alv03, Har04a, Har05a/b, Har06a, Tul06], además de incluir análisis de superficie para estructuras tridimensionales [Alv04a/b, Har04a/b/c, Har06b/c].

³ <http://www.conicyt.cl/bases/fondecyt/proyectos/03/2003/3030065.html>

⁴ <http://www.conicyt.cl/bases/fondecyt/proyectos/01/2006/1060890.html>

3.2 Plataformas de Software

Los proyectos previos de software se han desarrollado sobre la plataforma de programación IDL, conformando una aplicación particular para el procesamiento de imágenes de microscopia, denominada SCIAN (*Scientific Image Analysis*).

3.2.1 IDL

IDL⁵ es un ambiente computacional (entorno integrado de desarrollo y lenguaje de programación) para el análisis y la visualización de datos; integra un lenguaje orientado a arreglos con varias técnicas de análisis matemático y despliegue de gráficos. IDL es ampliamente utilizado en el ámbito científico, que requiere el uso intensivo de estas funcionalidades.

IDL presenta las siguientes características distintivas:

- Es un lenguaje estructurado, aunque en sus versiones más recientes incorpora soporte básico para la programación orientada a objetos.
- Posee operadores y funciones que trabajan sobre arreglos, encapsulando el uso de ciclos iterativos.
- La compilación y ejecución de comandos proporcionan realimentación inmediata y permiten interacción en tiempo de ejecución (depuración).
- Funcionalidades y herramientas para gráficos multidimensionales, despliegue de imágenes y animaciones.
- Soporta aceleración de gráficos por hardware basada en el estándar OpenGL.
- Incorpora rutinas numéricas y estadísticas para análisis y simulación.
- Manipula nativamente distintos formatos de archivos científicos (CDF, HDF, NetCDF), gráficos (BMP, TIFF, PNG, JPEG, JPEG2000, entre otros) y de texto (ASCII, WAV y XDR, entre otros).
- Permite la programación con componentes de tipo *widget* para entornos gráficos basados en ventanas.
- Está disponible para plataformas UNIX, Macintosh y Windows.

⁵ <http://www.itvis.com/idl/>

Para poder ejecutar aplicaciones desarrolladas en IDL se requiere disponer una máquina virtual. Esta máquina virtual puede encontrarse en el entorno integrado de desarrollo, o bien instalarse como un programa independiente.

El código fuente, así como la versión ejecutable de los programas escritos en IDL se almacenan en archivos de distinto tipo. Una aplicación en IDL se compone de los siguientes archivos:

- Proyecto (***.prj**): Existe uno por aplicación. Es un archivo en formato propietario que contiene información sobre la ubicación y organización de los demás archivos del proyecto (que pueden asociarse en grupos), además de un conjunto de opciones para la compilación, generación de código ejecutable y ejecución de la aplicación.
- Código fuente (***.pro**): en formato ASCII, contienen la codificación de los procedimientos o funciones de la aplicación; también pueden emplearse para definir clases y tipos de datos.
- Almacenamiento (***.sav**): archivos binarios que permiten tanto almacenar programas y procedimientos compilados, como guardar información sobre las variables y recursos manejados por una aplicación, para ser restaurados en distintas sesiones de trabajo.

3.2.2 SCIAN

Como resultado de varios proyectos de software enmarcados en la investigación de la biofísica se ha desarrollado SCIAN⁶: software *ad hoc* desarrollado en IDL, con el propósito de efectuar tareas de análisis sobre imágenes celulares [URL5]. Para esto, SCIAN incorpora varias funcionalidades de:

- Restauración de imágenes.
- Segmentación de objetos.
- Visualización de objetos en dos y tres dimensiones.
- Clasificación de objetos.
- Análisis temporal de objetos.

⁶ <http://www.scian.cl>. El nombre original, CIAN (*Cellular Image Analysis*), fue reemplazado por SCIAN debido a un conflicto de nombres de dominio Internet, el año 2005.

Diversos proyectos de investigación científica se han llevado a cabo gracias a las funcionalidades proporcionadas por SCIAN.

SCIAN es un proyecto de software en continuo desarrollo. Si bien no se inició como una aplicación orientada a objetos, ha incorporado algunas características para el diseño de filtros de imágenes y modelos de ROIs. Su desarrollo, que obedece a los requerimientos de investigación en biología, ha dado lugar previamente a varios Proyectos de Título en la carrera de Ingeniería Civil en Informática de la Universidad Austral de Chile. Es así que se han generado colaboraciones con la Universidad de Bremen (Alemania) [Bar02], el Departamento de Química Biológica de la Universidad Nacional de Córdoba (Argentina) [Oy03], y el Centro de Estudios Científicos (Valdivia, Chile) [Ver05]:

- “Plataforma de Identificación, Clasificación y Análisis de Imágenes, obtenidas a partir de Muestras Celulares” [Bar02] abordó métodos para segmentación y clasificación automatizadas para estructuras biológicas en movimiento en dos dimensiones.
- “Segmentación y análisis automatizado de objetos en movimiento aplicado al estudio de sistemas biológicos” [Oya03] incluyó herramientas para el análisis de estructuras celulares en movimiento.
- “Reconstrucción, visualización y parametrización tridimensional de estructuras biológicas a través de imágenes confocales de fluorescencia” [Ver05] desarrolló métodos para visualización y análisis de estructuras celulares en tres dimensiones.

3.3 Descripción del Proyecto

3.3.1 Nombre

Contornos activos en tres dimensiones para la segmentación y parametrización de estructuras biológicas.

3.3.2 Motivación

La motivación principal de este Proyecto es contribuir a la aplicación de métodos computacionales en el ámbito de la investigación científica, en el marco de los proyectos de investigación en biología celular que lo patrocinan.

Lo anterior, sumado a un entorno de trabajo que involucra a distintas disciplinas, conforma un Proyecto que busca contribuir al desarrollo local de iniciativas en ciencia y tecnología, a la vez que se integra a perspectivas de uso más amplio, con la difusión de SCIAN y su aplicación a diversas investigaciones científicas.

3.3.3 Objetivo General

Desarrollar e implementar algoritmos de contornos activos para la segmentación y parametrización tridimensional de estructuras biológicas.

3.3.4 Objetivos Específicos

- a) Reconocer e identificar los métodos de segmentación de ROIs a través de contornos activos, en su formulación y aplicación.
- b) Implementar algoritmos de contornos activos para la segmentación de ROIs en dos dimensiones, sobre la base de SCIAN existente.
- c) Extender la funcionalidad de los contornos activos desarrollada en dos dimensiones para la segmentación de ROIs tridimensionales.
- d) Combinar, ajustar y optimizar la funcionalidad de los contornos activos con la biblioteca de filtros existentes en SCIAN para la segmentación de estructuras biológicas.
- e) Extender la funcionalidad de los contornos activos desarrollada en dos y tres dimensiones para la reconstrucción multidimensional y la parametrización de contornos y superficies de ROIs para estructuras biológicas.

3.3.5 Resultados Esperados

Derivado de la motivación para desarrollar el presente Proyecto, se espera contribuir con el desarrollo herramientas computacionales que puedan ser aplicadas en el estudio de la morfología celular. Se espera que la implementación de los contornos activos permita:

- i. mejorar la representación espacial de las estructuras segmentadas, y
- ii. caracterizar cuantitativamente curvas y superficies de contorno complejas, tanto en dos como en tres dimensiones.

En función de los objetivos específicos, se definieron los siguientes ítems para la concreción del Proyecto:

3.3.6 Tabla 3.3.5-1: Objetivos específicos y resultados del Proyecto.

Objetivo Específico	Resultado
a) Reconocer e identificar los métodos de segmentación de ROIs a través de contornos activos, en su formulación y aplicación.	a) Descripción de los contornos activos para la segmentación y caracterización de ROIs en dos y tres dimensiones.
b) Implementar contornos activos para la segmentación de ROIs en dos dimensiones, sobre la base de SCIAN existente.	b) Implementación por software para segmentación por contornos activos en dos dimensiones sobre IDL.
c) Extender la funcionalidad de los contornos activos desarrollada en dos dimensiones para segmentar ROIs tridimensionales.	c) Implementación por software para segmentación por contornos activos en tres dimensiones sobre IDL.
d) Combinar, ajustar y optimizar la funcionalidad de los contornos activos con la biblioteca de filtros existentes en SCIAN para la segmentación de estructuras biológicas.	d) Integración de módulos de software para segmentación por contornos activos en dos y tres dimensiones en SCIAN.
e) Extender la funcionalidad de los contornos activos desarrollada en dos y tres dimensiones para la reconstrucción multidimensional y la parametrización de estructuras biológicas.	e) Desarrollo de métodos de software para la caracterización de ROIs en dos y tres dimensiones, a partir de la segmentación por contornos activos.

Capítulo 4: Contornos Activos en Dos Dimensiones

En este capítulo se presenta la base teórica de los contornos activos, formulada para dos dimensiones, junto con las funciones matemáticas que permiten la detección de bordes en imágenes. Luego se desarrolla el problema de interpolación de puntos para los contornos de estructuras biológicas, dado el carácter discreto de su representación: se presenta una aproximación experimental para determinar un nivel de discretización (resolución, i.e. número de puntos) óptimo. Lo anterior se efectúa en conjunto con una caracterización mediante parámetros geométricos, con el fin de cuantificar características sutiles en la morfología de los contornos. Finalmente se presenta una aplicación sobre estructuras de lípidos en células de cultivo.

4.1 Formulación

Un contorno activo (*snake*, en inglés) se define como una curva plana que busca minimizar su energía al deformarse iterativamente, guiado por fuerzas restrictivas internas e influenciado por fuerzas de la imagen (externas), para localizarse en características de ésta como líneas y bordes [Xu98a]. Mediante la definición de un funcional de energía sobre la curva, se derivan las fuerzas internas y externas, que la deforman desde una posición inicial hasta una que minimice su energía. La deformación se realiza considerando a la variación del contorno como una función del tiempo, en la búsqueda de una solución óptima. Un *snake* es, por lo tanto, una curva que evoluciona de forma dinámica hacia los contornos relevantes de la imagen. Las fuerzas internas permiten imponer restricciones de suavidad para regularizar la forma de la curva final. Las fuerzas externas empujan al *snake* hacia características de la imagen como líneas y bordes, o bien hacia contornos subjetivos. También es posible establecer una clasificación de las fuerzas externas: las que dependen sólo de la imagen, y las que permiten añadir información de alto nivel para hacer que el *snake* se desplace hacia un mínimo local deseado, como en [Coh91] y [Xu99].

En este modelo, tanto la conectividad del contorno como la presencia de los bordes en la imagen, afectarán al funcional de energía y, por ende, a los detalles de la estructura del contorno localmente óptimo [Kas88, Cas97, Mar02]. Además, pueden incluirse mecanismos de alto nivel introducidos por el usuario con el modelo de contorno, para ajustarlo según los criterios de optimalidad especificados.

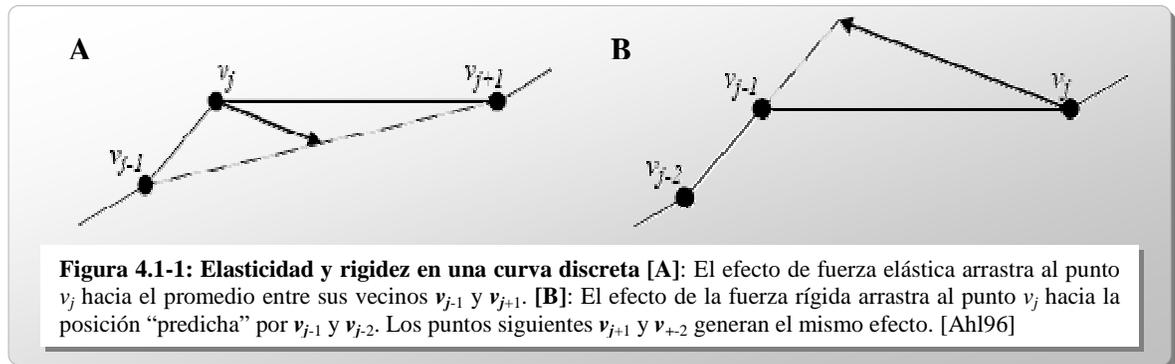
Un *snake* se define formalmente como una curva paramétrica, $C = C(s)$, que en dos dimensiones se representa por

$$C(s) = [x(s), y(s)], s \in [0,1] \quad [4.1-1]$$

El funcional integral de energía asociado, E , es definido en [Kas88] a partir de propiedades de continuidad (primera derivada) y concavidad (segunda derivada):

$$E = \int_0^1 \frac{1}{2} \left[\alpha \left| \frac{\partial C(s)}{\partial s} \right|^2 + \beta \left| \frac{\partial^2 C(s)}{\partial s^2} \right|^2 \right] + E_{ext}(C(s)) ds \quad (7) \quad [4.1-2]$$

α pondera la primera derivada de la curva, actuando como parámetro de tensión o fuerza contractiva del contorno; β (segunda derivada) se asocia a la rigidez de la curva, esto es, a su capacidad para deformarse y formar ángulos [Kas88]. La figura siguiente ilustra este concepto en una curva representada por un conjunto discreto de puntos.



La minimización del funcional E se realiza por medio del cálculo variacional (ver Anexo 1), y da lugar a una ecuación diferencial de Euler-Lagrange, de la forma

$$\alpha \frac{\partial^2 C(s)}{\partial s^2} - \beta \frac{\partial^4 C(s)}{\partial s^4} - \nabla E_{ext} = 0 \quad [4.1-3]$$

Para poder tener una solución computacional de esta ecuación se define un sistema dinámico; este sistema deberá estar gobernado por el funcional de energía, a la vez que **evolucione** hacia un estado de equilibrio. Esto se logra considerando un contorno variante en el tiempo: se extiende la formulación de C , incluyendo un parámetro temporal, t , haciendo

$$C = C(s, t) = [x(s, t), y(s, t)], t \in \mathbb{Z}_0^+ \quad [4.1-4]$$

⁷ La expresión " $|\cdot|$ " denota la norma euclidiana de \cdot .

Incorporando la ecuación 6.2.2-3, el criterio de minimización queda descrito por

$$\frac{\partial C(s,t)}{\partial t} = \alpha \frac{\partial^2 C(s,t)}{\partial s^2} - \beta \frac{\partial^4 C(s,t)}{\partial s^4} - \nabla E_{ext} \quad [4.1-5]$$

Así, cuando el contorno converge hacia un estado estable, su variación en el tiempo $\partial C/\partial t$ se anula, satisfaciendo la condición de la ecuación [4.1-3].

Para la solución numérica se aplican diferencias finitas. Siendo $C_i = (x_i, y_i)$ el contorno cerrado de $n+1$ puntos, $C(0) = C(n)$, $\partial^2 E_{ext}/\partial x^2 = -f_x$ y $\partial^2 E_{ext}/\partial y^2 = -f_y$ las fuerzas externas, la ecuación se reescribe como

$$\begin{aligned} & \alpha(C_{i+1} - C_{i-1}) \\ & + \beta(C_{i+2}) - 4\beta(C_{i+1}) + 6\beta(C_i) - 4\beta(C_{i-1}) + \beta(C_{i-2}) \\ & + (f_x(i) - f_y(i)) = 0 \end{aligned} \quad [4.1-6]$$

Ensamblando para todos los puntos de C , se formula el sistema de ecuaciones

$$\begin{aligned} Ax + f_x(x, y) &= 0 \\ Ay + f_y(x, y) &= 0 \end{aligned} \quad [4.1-7]$$

en que A es una matriz de bandas pentadiagonal con coeficientes de tipo:

$$A = \left\{ \begin{array}{cccccccc} a_0 & b_0 & c_0 & & & & c_{n-2} & b_{n-1} \\ b_0 & a_1 & b_1 & c_1 & & & & c_{n-1} \\ c_0 & b_1 & a_2 & b_2 & c_2 & & & \\ & c_1 & b_2 & a_3 & b_3 & c_3 & & \\ & & \dots & \dots & \dots & \dots & \dots & \\ & & & c_{n-5} & b_{n-4} & a_{n-3} & b_{n-3} & c_{n-3} \\ c_{n-2} & & & & c_{n-4} & b_{n-3} & a_{n-2} & b_{n-2} \\ b_{n-1} & c_{n-1} & & & & c_{n-3} & b_{n-2} & a_{n-1} \end{array} \right\}$$

Al incorporar la derivada C_t (en diferencias finitas atrasadas), según [4.1-5], se tienen las ecuaciones de Euler

$$\begin{aligned} Ax_t + f_x(x_{t-1}, y_{t-1}) &= -\gamma \frac{(x_t - x_{t-1})}{\Delta t} \\ Ay_t + f_y(x_{t-1}, y_{t-1}) &= -\gamma \frac{(y_t - y_{t-1})}{\Delta t} \end{aligned} \quad [4.1-8]$$

En este caso $\Delta t = 1$, tomando en cuenta que el contorno se ajusta progresivamente de una iteración (t_{i-1}) a la siguiente (t_i).

Se incluye el parámetro γ como un tamaño de paso, que permite ponderar la respuesta de la curva a la aplicación de fuerzas. γ se puede asociar físicamente con la propiedad de viscosidad de un fluido, dando cuenta de la rapidez con que éste se deforma al ser sometido a fuerzas mecánicas.

Finalmente, la inversión de matrices permite obtener el esquema iterativo final para obtener la solución del contorno (\mathbf{x}, \mathbf{y})

$$\begin{aligned} x_t &= (A + \gamma \mathcal{A})^{-1} - (x_{t-1} - f_x(x_{t-1}, y_{t-1})) \\ y_t &= (A + \gamma \mathcal{A})^{-1} - (y_{t-1} - f_y(x_{t-1}, y_{t-1})) \end{aligned} \quad [4.1-9]$$

4.2 Campos de Fuerzas Externas

Las fuerzas externas, definidas a partir propiedades de una imagen, son las que guían a un *snake* en su deformación hacia los contornos en la misma. Los contornos activos basados en energía [Kass88, Cas97, Xu98a/b], utilizan funcionales de energía (similares a los de las fuerzas internas), de los que derivan las fuerzas externas. Estos funcionales se implementan a partir de cierto tipo de funciones llamadas **detectores de aristas**.

4.2.1 Detectores de aristas

Sea I una imagen digital en dos dimensiones, de $m \times n$ píxeles, con intensidades en escala de grises en el rango $[0, I_{max}]$ ⁸. Definida como una función espacial de intensidad sobre sus coordenadas rectangulares:

$$\begin{aligned} I &= I(i, j); \\ i &\in [0, m-1], \quad j \in [0, n-1], \quad I \in [0, I_{max}]; \quad i, j, I \in \mathbb{Z}_0^+ \end{aligned} \quad [4.2.1-1]$$

Un detector de bordes o aristas es una función espacial, f , definida sobre el dominio de I , que toma valores máximos en las aristas de los objetos dentro de la imagen. Un ejemplo simple de detector se basa en los **gradientes de intensidad**. Se trata de una función vectorial que cuantifica los cambios de intensidad para resaltar las aristas de objetos en una imagen.

La función gradiente se denota por ∇I , y se define por

$$\begin{aligned} \nabla I &= [I_x, I_y] \\ I_x &= \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y} \end{aligned} \quad [4.2.1-2]$$

⁸ En imágenes en escala de grises el valor de I_{max} depende de la profundidad en bits d , siendo $I_{max} = 2^d$

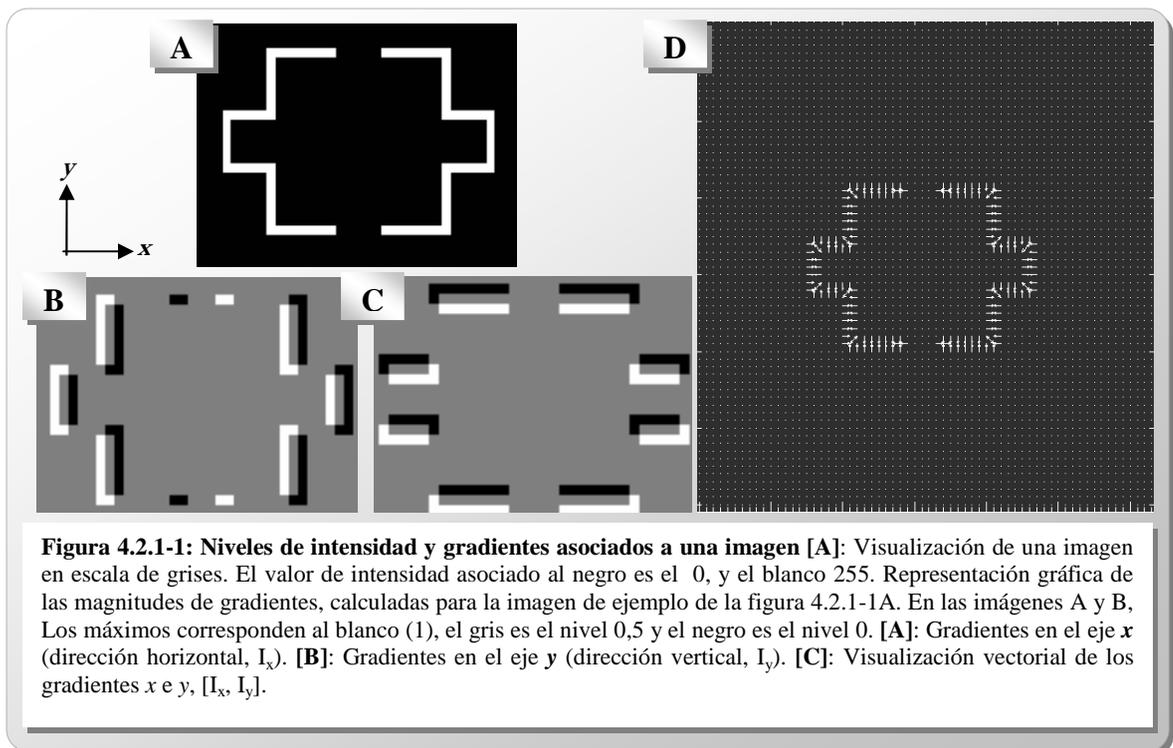
El detector de bordes que se construye sobre ∇I es

$$f(x, y) = f(\nabla I) = |\nabla I| = |I_x| + |I_y| \quad [4.2.1-3]$$

Aplicado sobre una imagen bidimensional $I = \{i_{ij}\}$, $[i, j] \in [0, m-1] \times [0, n-1]$, el gradiente en cada dirección se aproxima para cada píxel como el promedio de las intensidades de sus vecinos en los dos sentidos, siendo

$$\begin{aligned} I_x(i, j) &= 0.5I(i+1, j) - 0.5I(i-1, j) \\ I_y(i, j) &= 0.5I(i, j+1) - 0.5I(i, j-1) \end{aligned} \quad [6.2.1-3]$$

La siguiente figura muestra una imagen y sus gradientes de intensidad asociados.



4.2.2 Flujo de Vectores Gradiente – GVF

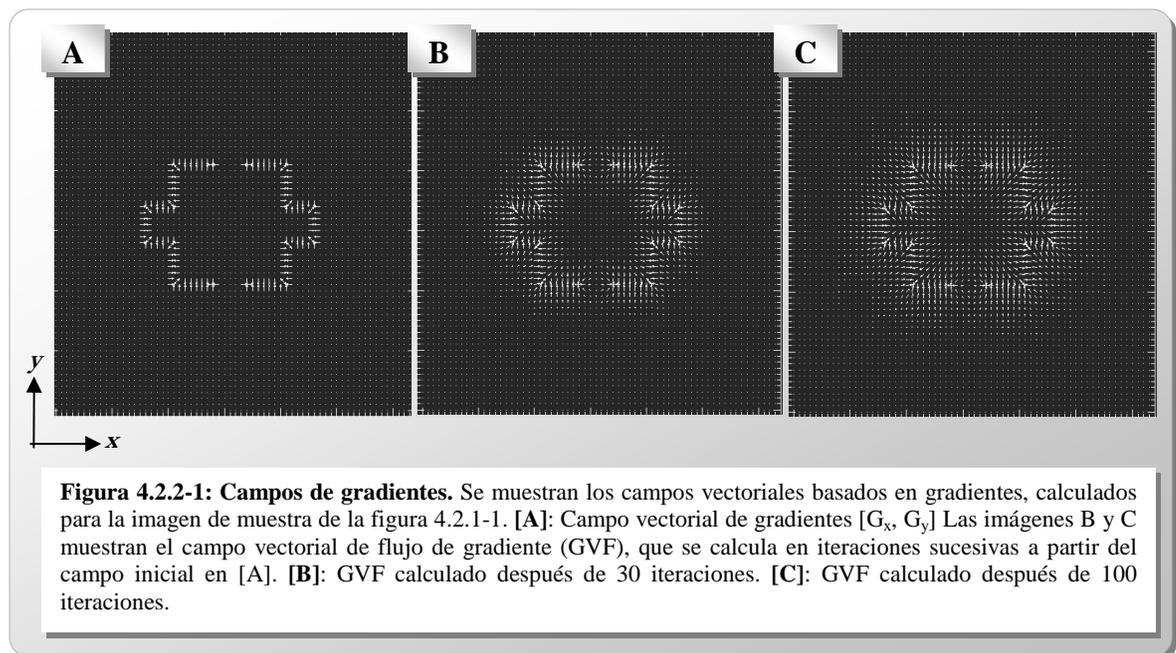
[Xu98a] propuso el uso de un campo de fuerzas externas estático, derivado del detector de gradientes de intensidad. La idea es aplicar una difusión sobre el campo de vectores, en el llamado flujo de vectores gradiente (*Gradient Vector Flow, GVF*).

Sean $I(x, y)$ la función de intensidad de una imagen, y ∇I su respectivo detector de aristas. El campo de vectores $V = V(x, y)$ se puede derivar del funcional de energía de la forma

$$\min(\mathcal{E}) = \iint g(|\nabla I|) \nabla^2 V + h(|\nabla I|) |V - \nabla I|^2 dx dy \quad [4.2.2-1]$$

h es una función que controla la orientación del campo de vectores en dirección a las aristas (valores máximos de ∇I), y g pondera el grado de difusión del campo (dado por el operador de Laplace, $\nabla^2 V$) en zonas de bajo gradiente. Así, el campo de vectores será suave en zonas alejadas de los bordes, orientado en dirección hacia éstos por ∇I , y aumentando progresivamente de magnitud hacia las aristas, donde tomará valores máximos.

Como las funciones g y h dependen de los gradientes (que varían espacialmente), para lograr que el campo tienda hacia los bordes, éstas deben ser monótonas no-creciente y no-decreciente, respectivamente. Así, V tomará valores similares a los del mapa de aristas en los bordes, y difundirá en el sentido de los gradientes en las zonas suaves de la imagen. La figura 4.2.2-1 muestra la aplicación de este tipo de campo vectorial sobre la imagen de ejemplo de la figura 4.2.1-1A.



Para encontrar una solución iterativa del campo de vectores, a partir de una aproximación inicial (en este caso, el mapa de aristas), se incluye en la formulación la variación temporal de V . De forma análoga a las ecuaciones de mínima energía en un *snake* se el esquema iterativo se formula según

$$V_t = g(|\nabla I|)\nabla^2 V - h(|\nabla I|)(V - \nabla I) \quad [4.2.2-2]$$

La formulación original [Xu98a] emplea $g = \mu$ (coeficiente de difusión constante), y $h = |\nabla I|^2$ (detector de bordes), dando lugar al sistema de ecuaciones diferenciales con variables no acopladas de la ecuación 4.2.2-3.

$$\begin{aligned}\mu\nabla^2 u - (u - I_x)(I_x^2 + I_y^2) &= 0 \\ \mu\nabla^2 v - (v - I_y)(I_x^2 + I_y^2) &= 0\end{aligned}\quad [4.2.2-3]$$

Para la implementación y solución numérica del sistema, primero se debe considerar al campo como función del tiempo

$$\begin{aligned}u_t(x, y, t) &= \mu\nabla^2 u(x, y, t) - [u(x, y, t) - I_x(x, y)] * [I_x(x, y)^2 + I_y(x, y)^2] \\ v_t(x, y, t) &= \mu\nabla^2 v(x, y, t) - [v(x, y, t) - I_y(x, y)] * [I_x(x, y)^2 + I_y(x, y)^2] \\ t &\in \mathbb{Z}_0^+\end{aligned}\quad [4.2.2-4]^9$$

Por conveniencia se rescribe el sistema como

$$\begin{aligned}u_t(x, y, t) &= \mu\nabla^2 u(x, y, t) - b(x, y) * u(x, y, t) + c^1(x, y) \\ v_t(x, y, t) &= \mu\nabla^2 v(x, y, t) - b(x, y) * v(x, y, t) + c^2(x, y)\end{aligned}\quad [4.2.2-5]$$

con

$$\begin{aligned}b(x, y) &= I_x(x, y)^2 + I_y(x, y)^2 \\ c^1(x, y) &= b(x, y)I_x(x, y)^2 \\ c^2(x, y) &= b(x, y)I_y(x, y)^2\end{aligned}\quad [4.2.2-6]$$

Así, b , c^1 y c^2 son términos constantes para el proceso iterativo.

⁹ Estas ecuaciones se conocen como ecuaciones de difusión generalizadas, y suelen aplicarse a fenómenos físicos de transporte tales como conducción de calor y dinámica de fluidos.

El campo de vectores V en un espacio discreto de coordenadas rectangulares x e y , será

$$V = V(x_i, y_j) = [u_i, v_j] \quad [4.2.2-7]$$

Reemplazando las derivadas por diferencias finitas centradas se tiene un sistema discreto de ecuaciones

$$\begin{aligned} u_{i,j}^{t+1} &= (1 - b_{i,j}\Delta t)u_{i,j}^t + r(u_{i+1,j}^t + u_{i,j+1}^t + u_{i-1,j}^t + u_{i,j-1}^t - 4u_{i,j}^t) + c_{i,j}^1\Delta t \\ v_{i,j}^{t+1} &= (1 - b_{i,j}\Delta t)v_{i,j}^t + r(v_{i+1,j}^t + v_{i,j+1}^t + v_{i-1,j}^t + v_{i,j-1}^t - 4v_{i,j}^t) + c_{i,j}^2\Delta t \end{aligned} \quad [4.2.2-8]$$

La solución para este sistema se obtiene de modo iterativo, con Δt como paso de tiempo. La convergencia del proceso depende de la restricción de Courant-Friedrichs-Lewy [Ame92]

$$\Delta t \leq \frac{1}{4\mu} \Delta x \Delta y \quad [4.2.2-9]$$

Esta restricción se aplica a la velocidad de convergencia con que el contorno puede deformarse, según las condiciones de la imagen. Si ésta tiene estructuras gruesas, Δx y Δy pueden hacerse más grandes. En tanto, para valores altos de μ (alto nivel de difusión en el campo, para imágenes con ruido, por ejemplo), la convergencia se hace más lenta, ya que Δt deberá mantenerse bajo (en este caso $\Delta t = 1$).

Extensión a Tres Dimensiones

La formulación del campo vectorial de la ecuación [4.2.2-1] permite su extensión de modo directo a tres dimensiones, considerando a la imagen una función definida para tres variables, $\mathbf{I} = \mathbf{I}(x, y, z)$, e incorporando una tercera componente al campo vectorial, \mathbf{w} , siendo $\mathbf{V} = [\mathbf{u}, \mathbf{v}, \mathbf{w}]$. De este modo el esquema iterativo base para el cálculo del campo será

$$\begin{aligned} u_t(x, y, z, t) &= \mu \nabla^2 u(x, y, z, t) - [u(x, y, z, t) - I_x(x, y, z)] * [I_x(x, y, z)^2 + I_y(x, y, z)^2 + I_z(x, y, z)^2] \\ v_t(x, y, z, t) &= \mu \nabla^2 v(x, y, z, t) - [v(x, y, z, t) - I_y(x, y, z)] * [I_x(x, y, z)^2 + I_y(x, y, z)^2 + I_z(x, y, z)^2] \\ w_t(x, y, z, t) &= \mu \nabla^2 w(x, y, z, t) - [w(x, y, z, t) - I_z(x, y, z)] * [I_x(x, y, z)^2 + I_y(x, y, z)^2 + I_z(x, y, z)^2] \\ t &\in \mathbf{Z}_0^+ \end{aligned} \quad [4.2.2-10]$$

4.2.3 Flujo de Vectores Gradiente Generalizado –GGVF

[Xu98b] analiza un caso general de funciones de ponderación para el funcional de energía del campo vectorial. Si la función de suavidad asociada a la difusión, g , es constante, en zonas de bordes muy próximos su efecto se traslapará con el de la función detectora de bordes h , haciendo perder fuerza al campo que guía al contorno en su deformación.

Al escoger un par de funciones g y h que aumenten y disminuyan en forma recíproca, se puede obtener un campo que permita acercarse mejor al contorno hacia bordes pronunciados. En particular [Xu98b], se emplean

$$\begin{aligned} g(|\nabla I|) &= e^{-(|\nabla I|/K)} \\ h(|\nabla I|) &= 1 - g(|\nabla I|) \end{aligned} \tag{4.2.3-1}$$

La restricción de convergencia en este caso será

$$\Delta t \leq \frac{\Delta x \Delta y}{4g_{\max}} \tag{4.2.3-2}$$

La figura siguiente muestra el campo vectorial GGVF obtenido para la imagen de muestra de la sección 6.2.1.

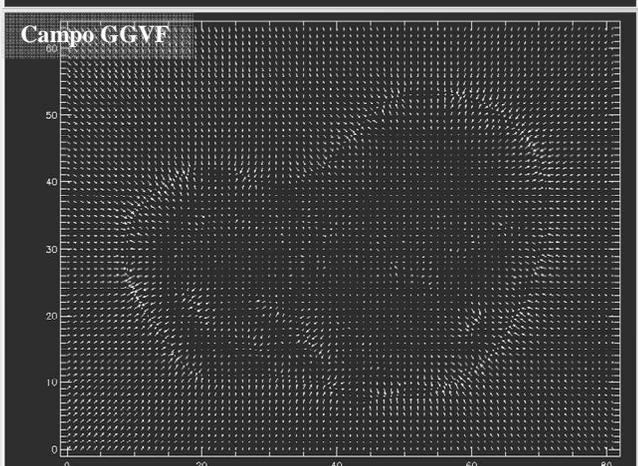
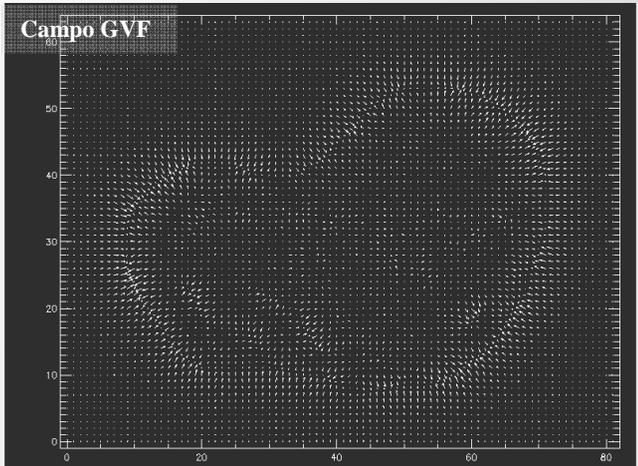
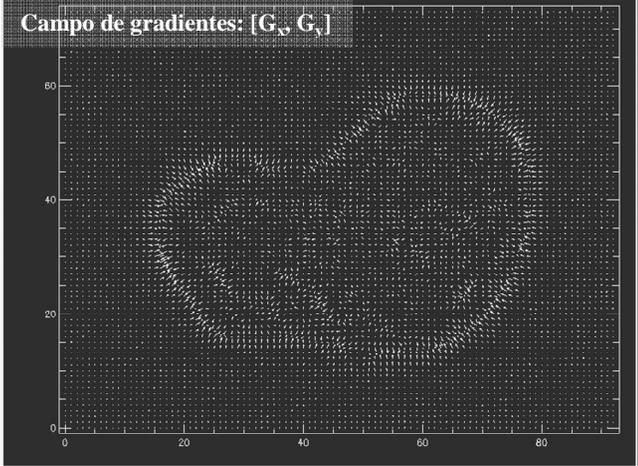
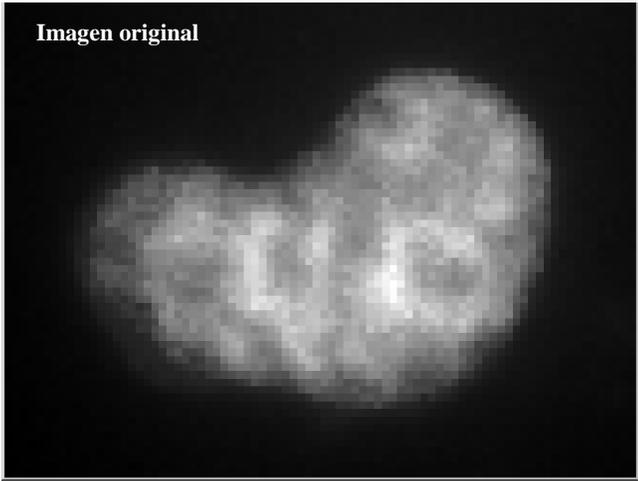
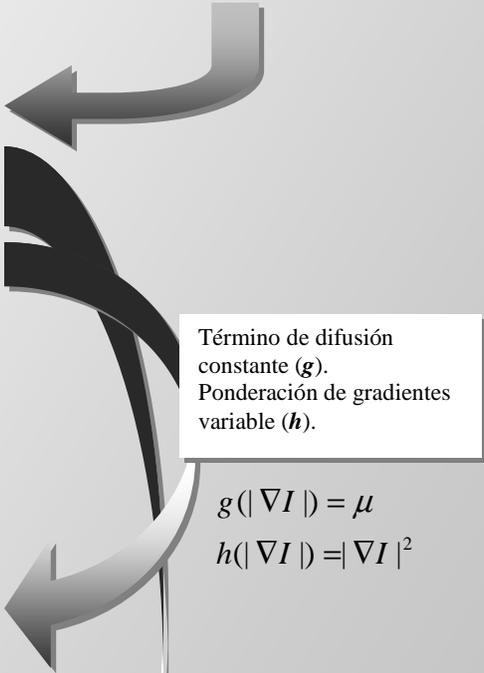
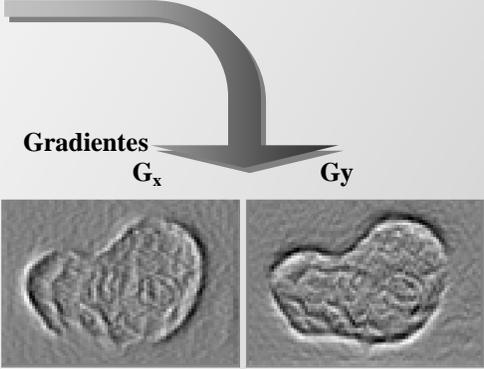


Figura 4.2.3-1: Campos de gradiente y flujos de gradiente GVF/GGVF.
Campos vectoriales basados en gradientes, calculados a partir de la imagen de muestra de la figura 6.2.1-1.



Término de difusión constante (g).
Ponderación de gradientes variable (h).

$$g(|\nabla I|) = \mu$$

$$h(|\nabla I|) = |\nabla I|^2$$

Ponderación de gradientes variables, y recíprocas en variación.

$$g(|\nabla I|) = e^{-(|\nabla I|/K)}$$

$$h(|\nabla I|) = 1 - g(|\nabla I|)$$

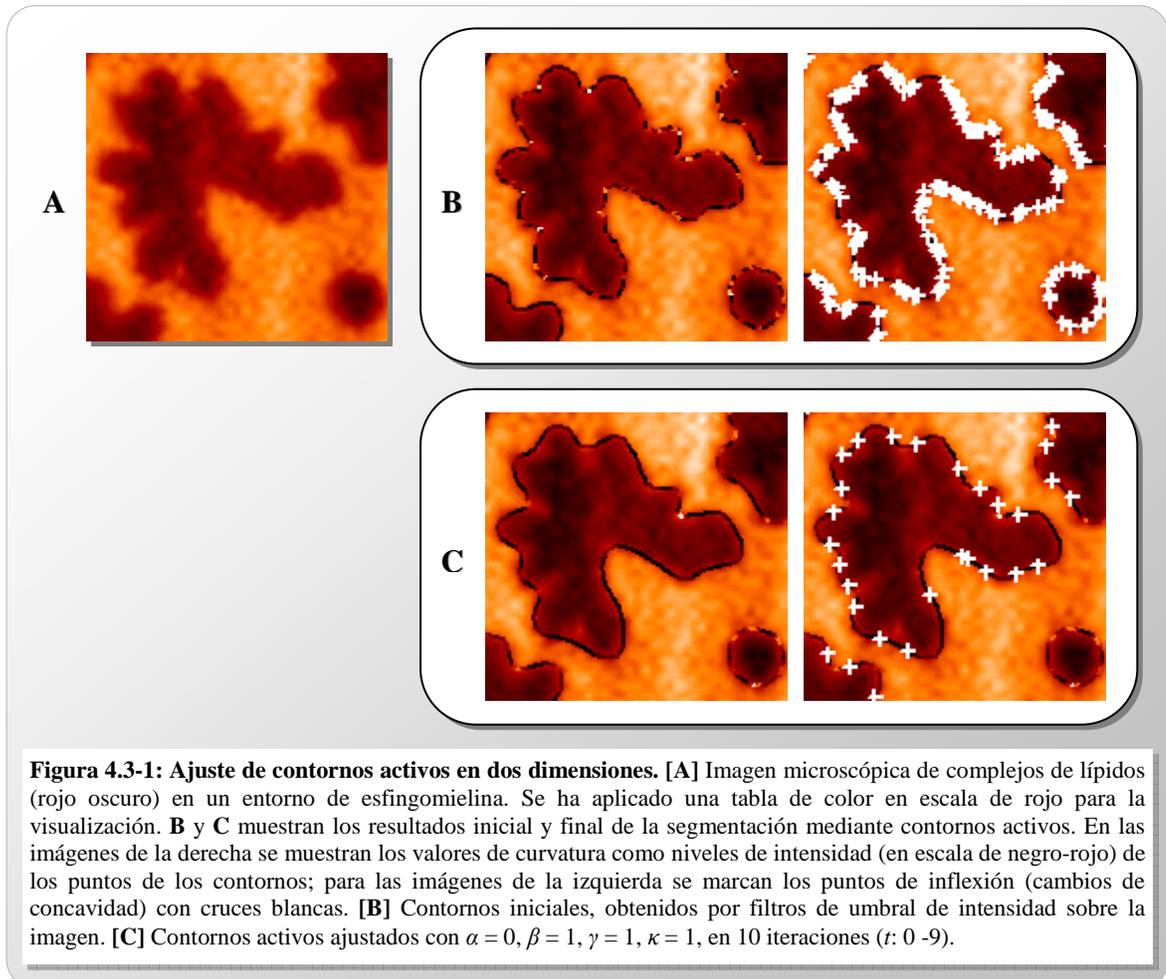
4.3 Parametrización de Fuerzas

La dinámica de ajuste de los contornos activos paramétricos se deriva de un criterio de mínima energía (sección 4.1). A partir de un funcional de energía se definen fuerzas ($F_{elást}$, $F_{rígida}$, y F_{ext}) con sus respectivos coeficientes (α , β , κ), más un parámetro de viscosidad (γ), que actúan sobre el contorno. De este modo se genera el siguiente esquema iterativo para la deformación del contorno en el tiempo, (C_{t+1}), desde un estado inicial ($C_{t=0}$):

$$\gamma(C_{t+1} - C_t) = \alpha F_{elást}(C_t) + \beta F_{rígida}(C_t) + \kappa F_{ext}(C_t) \quad [4.3-1]$$

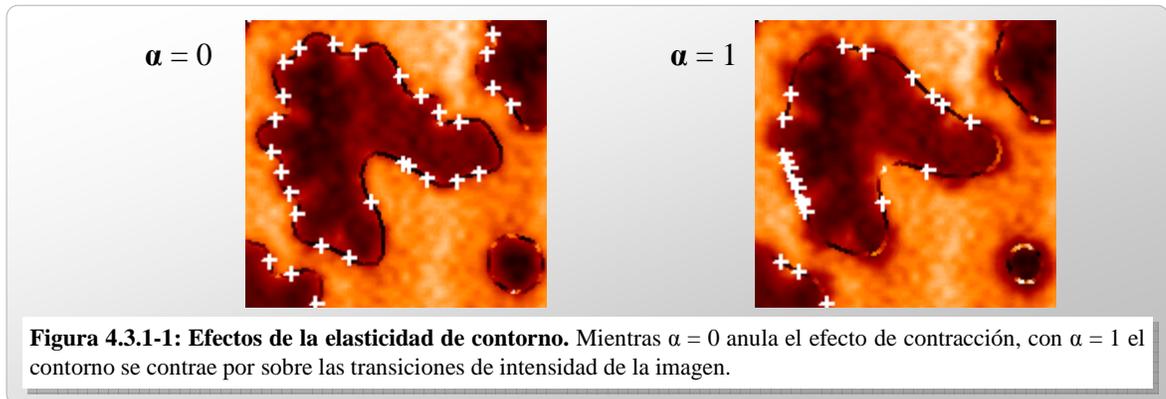
$$C_{t+1} = C_t + \gamma^{-1}[\alpha F_{elást}(C_t) + \beta F_{rígida}(C_t) + \kappa F_{ext}(C_t)] \quad [4.3-2]$$

A continuación se presentan y se discuten los efectos de los coeficientes α , β , γ , κ sobre la deformación de un contorno inicial. El ejemplo representativo escogido presenta dominios lipídicos enriquecidos con ceramida (oscuro), en un entorno consistente de esfingomielinea [Har04a/05b/06]. Se busca encontrar las mejores aproximaciones de sus contornos para obtener una precisa parametrización morfológica.



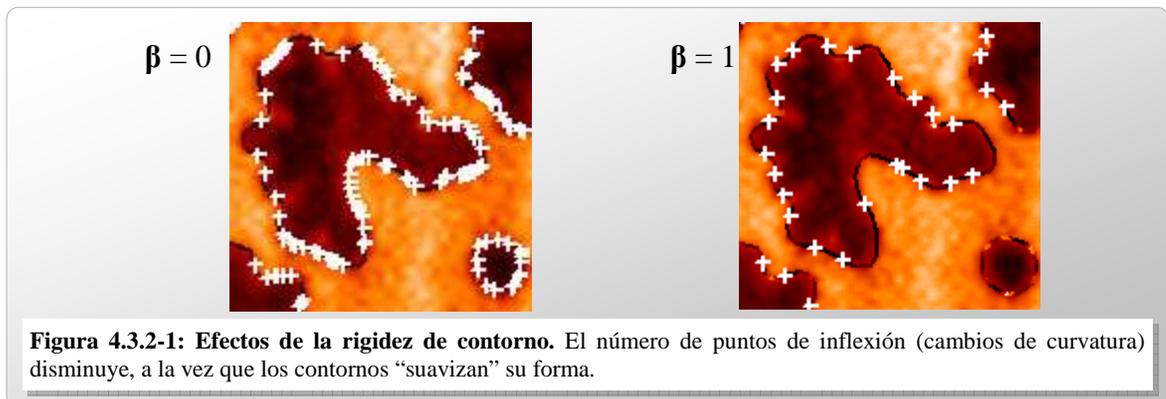
4.3.1 Coeficiente de Elasticidad (α)

La fuerza elástica actúa contrayendo al contorno, disminuyendo la distancia entre sus puntos; y su ponderación está dada por el coeficiente α .



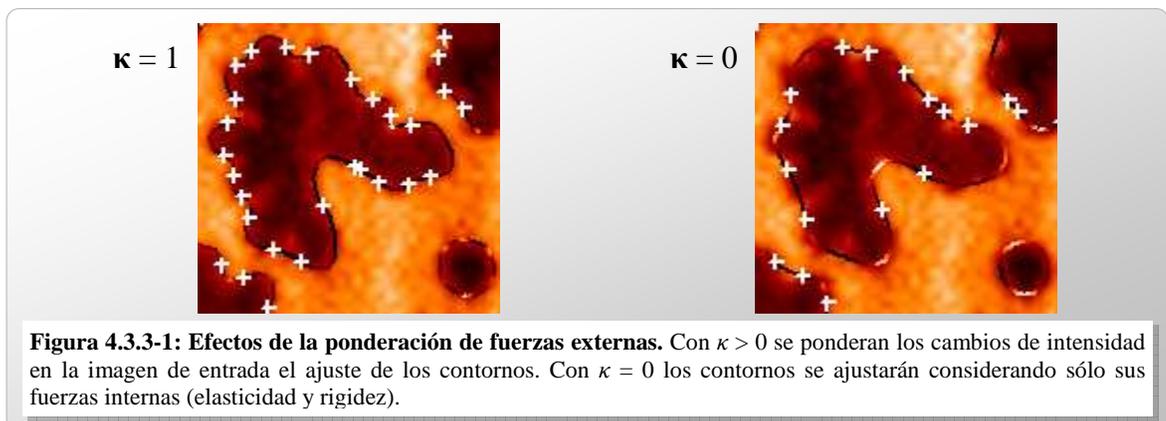
4.3.2 Coeficiente de Rigidez (β)

Pondera el grado en que el contorno presenta cambios de curvatura pronunciados (fuerza rígida), como esquinas o salientes, que disminuyen con valores altos de β .



4.3.3 Coeficiente de Fuerzas Externas (κ)

El factor κ pondera las fuerzas de campo calculadas sobre la imagen.



4.3.4 Coeficiente de Viscosidad (γ)

Controla la magnitud de la deformación del contorno entre iteraciones, y equivale a una ponderación conjunta de todas las fuerzas anteriores. Menores valores de γ aumentan el grado de sensibilidad del contorno a la acción de las fuerzas, mientras que al incrementarse disminuye la velocidad de deformación del contorno.

4.4 Refinamiento y Caracterización

Las estructuras segmentadas con los contornos activos paramétricos no requieren procesamiento posterior, por lo que son directamente interpretables en términos de su morfología y topología. Aprovechando esta propiedad, se busca incrementar la resolución de los contornos, con el fin de mejorar la calidad de su caracterización morfológica y topológica.

4.4.1 Interpolación de la Curva de Contorno

El proceso de segmentación presentado se aplica sobre un dominio discreto de la solución, es decir, representando a cada contorno por un conjunto discreto de puntos. Al representar estructuras físicas continuas, se requiere saber cuántos puntos son necesarios para no perder ni distorsionar información. El número óptimo de puntos equidistantes para un contorno dado será el del mínimo conjunto que conserve la información relativa a la estructura, sin redundar en información para cómputo y almacenamiento posterior. Si bien los contornos que se definen a partir de la imagen son de coordenadas enteras (píxeles), la interpolación permite generar valores intermedios a partir de criterios de continuidad, generando un **modelo de resolución aumentada**.

Con el objetivo de caracterizar morfológicamente a las regiones segmentadas, el cálculo de parámetros sirvió de base para evaluar la relación entre el número de puntos de los contornos y grado de información que aportan. Los parámetros de morfología deben ser cuantificables, y comparables entre las estructuras segmentadas. Por esta razón se empleó la **curvatura geométrica** como criterio para definir el grado de información representado por un contorno, en relación al número de puntos de éste.

4.4.2 Estimación de Curvatura en Dos Dimensiones

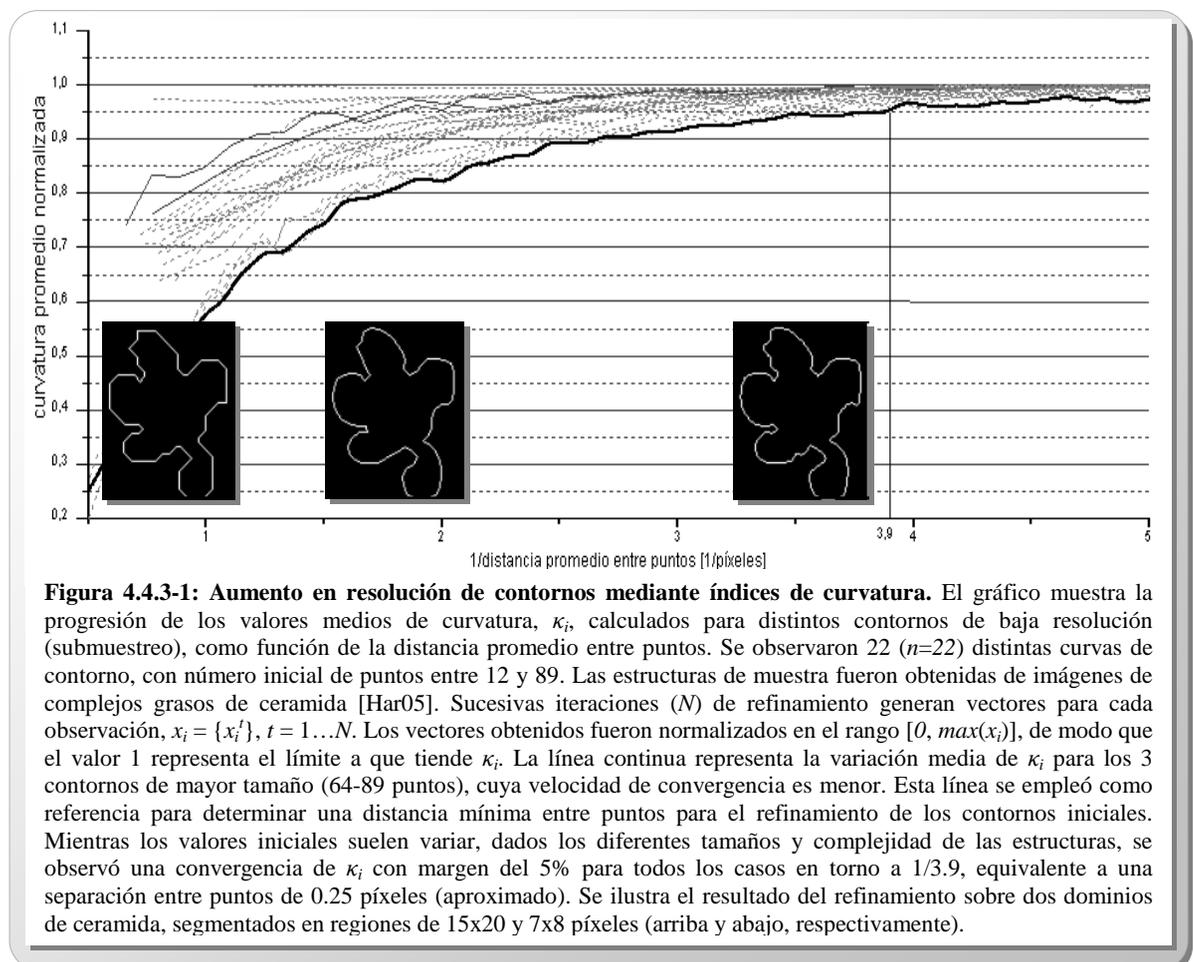
Para un punto cualquiera, p_j , de una curva C_i , la curvatura, κ_j , se define por una función de dicho punto y sus dos adyacentes (p_{j-1}, p_{j+1}). El valor de κ_j será igual a 1 sobre el radio de la circunferencia que pasa por esos tres puntos [Ebe04]. La curvatura promedio estará dada por $\kappa_i = (\sum \kappa_j) / n$, sumando sobre todos los puntos de C_i .

Con el objetivo de determinar la resolución (grado de discretización) suficiente para representar a las estructuras segmentadas por contornos activos, se realizaron experimentos de prueba con curvas de contorno típicas.

4.4.3 Experimento de Validación

Para una curva de contorno discreta, al aumentar sucesivamente el número de puntos (interpolados) y calcular el valor de curvatura κ correspondiente, se puede observar la variación de κ , hasta encontrar un rango en que converge o se estabiliza. Luego, esta cantidad debe transformarse en una medida que sea aplicable a contornos de distinto tamaño y complejidad (igual o menor a la de las estructuras de prueba).

Dado un contorno inicial, C_i^0 , con n^0 puntos y curvatura media κ_i^0 , se itera aumentando el número de puntos, generando una serie de curvas por interpolación, C^* , que se aproximan a un valor límite de curvatura promedio κ_{i_lim} . La curva C_i^* de curvatura κ_{i_lim} y cuyo número de puntos sea mínimo para C^* será el contorno óptimo. Al relacionar los valores de curvatura promedio normalizados con la distancia media entre puntos (d_i) sobre varios contornos, se busca obtener un valor d^* que sea aplicable para estructuras de igual o menor complejidad. La figura 4.4.3-1 muestra los resultados obtenidos.



Capítulo 5: Aplicación

A continuación se presenta una aplicación conjunta de las herramientas descritas en las secciones previas, implementadas en software mediante IDL e integradas en SCIAN. [Har05b].

5.1 Dominio de Aplicación

La siguiente aplicación se enmarca en un estudio de los cambios en morfología que experimentan complejos de lípidos, observados mediante imágenes de microscopía en una serie temporal.

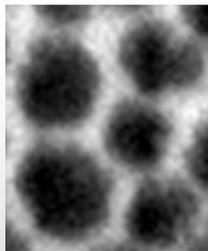


Figura 5.1-1: Imagen de microscopía. Estructuras en una monocapa lipídica de la membrana celular, consistente de ceramida y esfingomiolina.

5.2 Segmentación por Filtros

El primer paso de segmentación consistió en aplicar filtros de umbrales a las imágenes de entrada. Estos filtros permiten cuantificar una característica de una imagen para cada píxel que la compone, y definirle valores de límite para discriminar entre regiones de interés y fondo. Una vez aplicados los filtros, se obtiene una imagen binaria en que es posible distinguir a las ROIs del fondo.

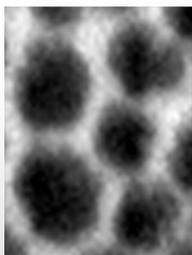


Figura 5.2-1: Segmentación por umbrales de intensidad. La imagen binaria de la derecha muestra la máscara resultante de aplicar un filtro de umbral a la imagen original (izquierda), con valores de intensidad entre 0 y 255. En la imagen máscara, cada píxel del fondo se identifica en por tener un valor de intensidad 0 (negro), mientras que los píxeles de las regiones segmentadas tienen un valor de intensidad 1 (blanco).

De las máscaras binarias se obtuvieron los polígonos de contorno iniciales para cada región, como una curva cerrada C_i de p píxeles ordenados por sus posiciones adyacentes¹⁰

$$C_i = \{ (x_j, y_j) \mid (x_j, y_j) \in \mathbb{Z}_{0+}^2; j \in \mathbb{Z}_{0+}, j \in [0, p]; x_j \in [0, m-1]; y_j \in [0, n-1] \}$$

¹⁰ En el espacio de coordenadas rectangulares de la imagen, se consideran adyacentes a un píxel a aquellos ubicados arriba, abajo, a la derecha y a la izquierda, así como a los de las cuatro diagonales que se forman.

5.3 Interpolación de Contornos

Para aumentar el nivel de resolución de las estructuras, se incrementa el número de puntos de los contornos generados por los filtros. Esto se hizo interpolando sobre los puntos de las curvas para obtener contornos mejorados C_i^* de p^* puntos equidistantes, según el criterio determinado en la sección 4.4.

La interpolación se realizó con curvas *spline* cúbicas, añadiendo como información de entrada las derivadas de la curva original. Los algoritmos empleados corresponden a las funciones predefinidas de IDL **spl_init** (cálculo de derivadas) y **spl_interp** (interpolación).

5.4 Segmentación por Contornos Activos

A partir de cada imagen I_i , se aplica una función de cálculo del campo GGVF, para luego iterar deformando los contornos interpolados C_i^* .

Se determinaron los siguientes parámetros para el ajuste de los contornos:

$\alpha = 0.1$ Coeficiente de elasticidad bajo, ya que los contornos iniciales no requieren contraerse.

$\beta = 0.2$ Permite que la curva tenga variaciones suaves.

$\gamma = 0.5$ Deformación (viscosidad) suave de la curva.

$\kappa = 0.1$ Las variaciones de intensidad en la imagen son suaves, haciendo que el campo vectorial no sea muy pronunciado en los bordes.

$f. \text{perímetro} = 0.5-1.3$ Separación máxima entre los puntos de la curva (en píxeles).

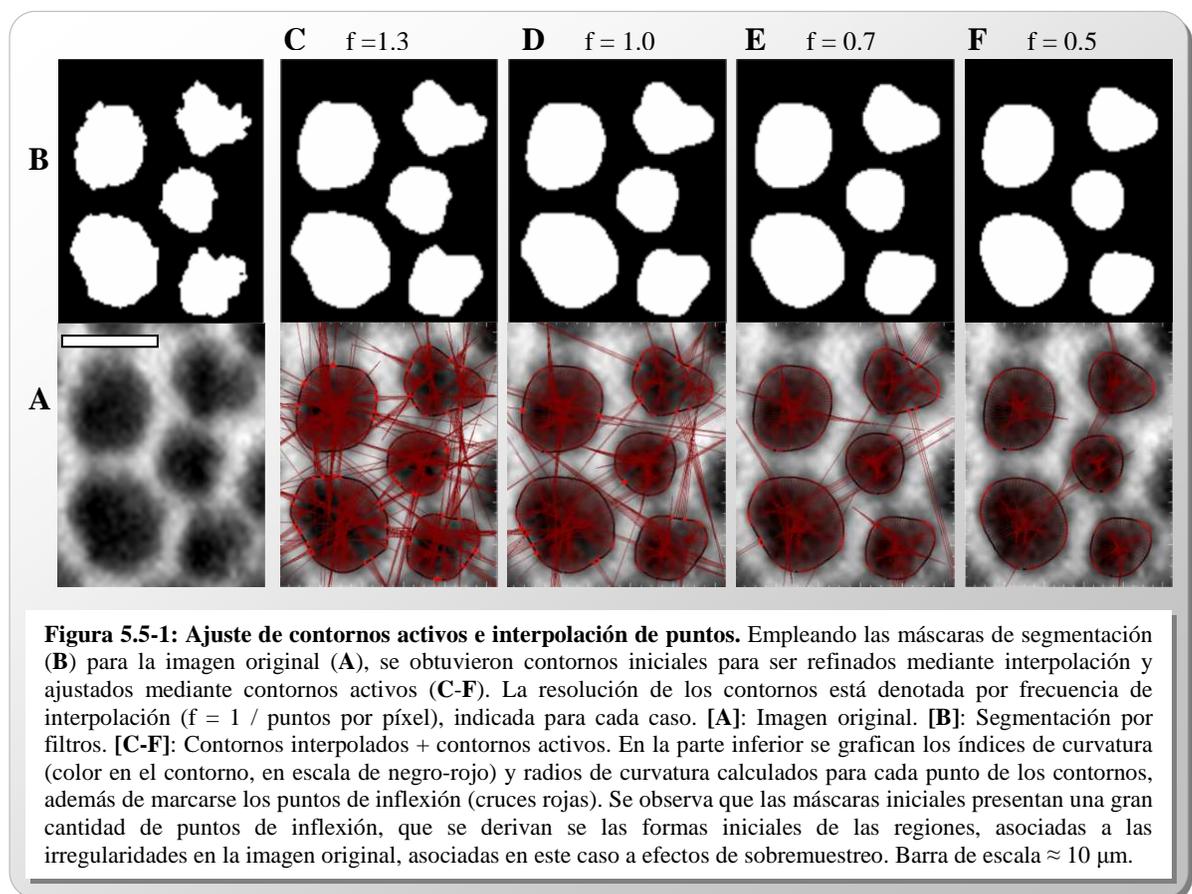
$k = 0.05$ Ponderación de gradiente $|\nabla I|$ para las funciones de control g y h .

El cálculo del campo vectorial se realizó con 20 iteraciones, mientras que se realizaron 5 iteraciones para el ajuste de los contornos, considerando la cercanía de los contornos iniciales a las transiciones de intensidad en la imagen.

5.5 Cálculo de Parámetros

Una vez segmentadas las regiones, se aplican funciones para el cálculo de parámetros geométricos sobre cada contorno C_i^{*f} :

- **Perímetro:** calculado como la suma de las distancias euclidianas entre cada par de puntos contiguos en la curva.
- **Radio de curvatura (suma y valor medio):** dependiente del tamaño de las estructuras, estimado por la interpolación de un arco circular para tríadas de puntos.
- **Índice de curvatura (suma y valor medio) extrínseca:** evaluada sobre la tangente del arco, en función de las derivadas de x e y de la curva [Gra98]; entrega un valor con un signo que permite reconocer cambios de concavidad.
- **Número de puntos de inflexión:** contando los cambios de signo en los índices de curvatura, permite reconocer los cambios de concavidad, correspondientes a salientes o hendiduras en los dominios segmentados.



Capítulo 6: Contornos Activos en Tres Dimensiones

Este capítulo aborda la representación de estructuras espaciales como modelos de superficies en tres dimensiones, además de implementar los contornos activos sobre dichas superficies. Se presentan diferentes alternativas existentes en el contexto de la aplicación, y se validan con casos de solución teórica. Posteriormente se determina la forma de representación que mejor capture la información estructural, requerida para la aplicación de los contornos activos en tres dimensiones.

La extensión de los contornos activos, desde su formulación original en dos dimensiones, hacia un espacio tridimensional, implica cambios en la representación de las regiones segmentadas (ROIs). Para el caso bidimensional, cada ROI fue definida por una curva cerrada de contorno (Capítulo 4); análogamente, para tres dimensiones, la función de contorno corresponderá a una superficie cerrada. La expresión “superficies deformables” o “superficies activas” denota genéricamente a los distintos modelos de volumen tridimensionales para contornos activos.

En el presente Proyecto, la elección de un adecuado modelo para la representación de superficies en tres dimensiones obedece a dos requisitos esenciales:

- la aplicación de los contornos activos, y
- reflejar adecuadamente la morfología de las estructuras, en función de los datos de entrada.

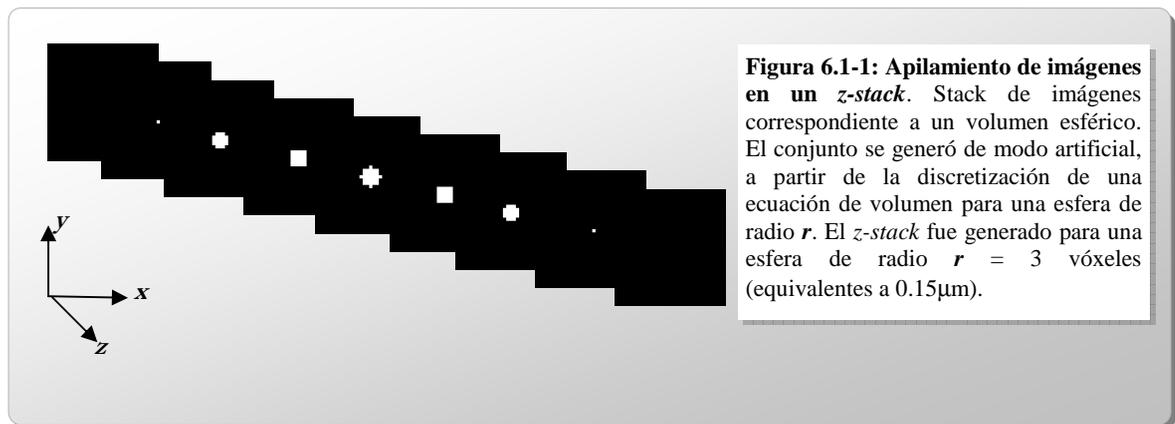
Considerando la forma de los datos de entrada, se opta por un modelo de contorno discreto. Para representar estructuras tridimensionales a partir de un *z-stack* de imágenes (datos discretos), se recurre a un modelo de **malla** de superficie discreta.

Para las ROIs segmentadas desde un *z-stack*, es posible definir una malla o conjunto de puntos, conectados entre sí por un conjunto de polígonos asociado. La superficie resultante es una aproximación a la estructura real que representa, permitiendo su visualización y análisis por medios computacionales.

6.1 Modelos Discretos de Superficies

A partir en las funcionalidades de segmentación desarrolladas para dos dimensiones, los modelos de volumen son generados desde las imágenes binarias (máscaras), en un proceso de **apilamiento**. Considerando a cada imagen como un plano (x, y) , se le ubica en un z -stack según su altura z_i , y se establece la conectividad de los puntos que definen a cada ROI de su plano (contorno bidimensional) con los de sus vecinos z_i y z_{i+1} .

En el proceso de apilamiento, cada punto en una imagen pasa de su representación en píxeles, $P_{ij} = (x_i, y_j)$, a una de vóxeles (elementos discretos de volumen) $V_{ijk} = (x_i, y_j, z_k)$. La lista de polígonos permite asociar a los distintos vértices que conforman los contornos de una ROI tridimensional, conformando su superficie de contorno.



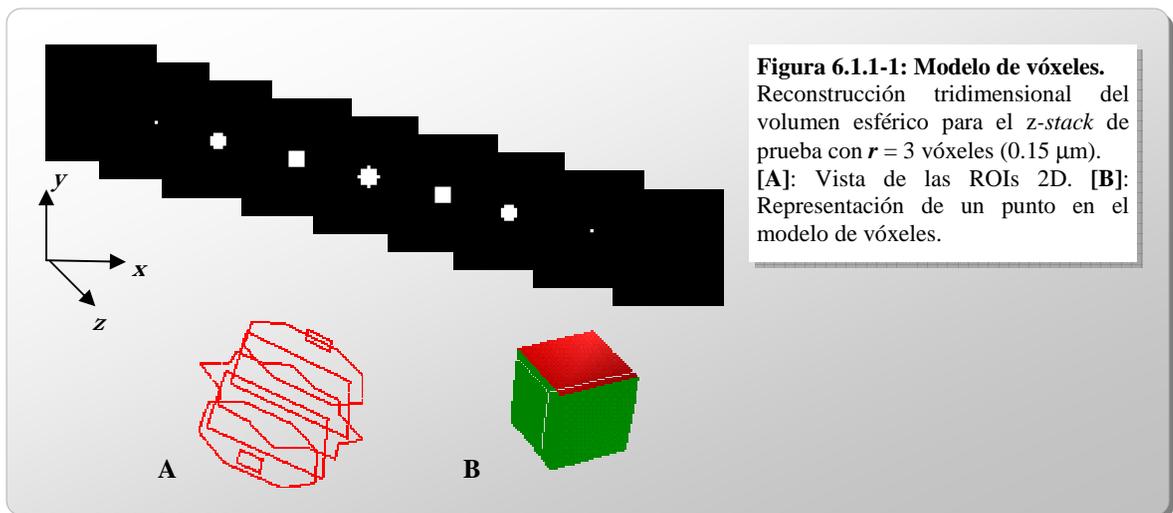
Una vez obtenida la malla de puntos, es posible calcular el área superficial como la suma de las áreas de sus polígonos componentes. Con el fin de comparar las distintas alternativas de generación de mallas, se incluyó una función para el cálculo de área, en la definición de la clase para ROIs tridimensionales.

Tomando en cuenta las herramientas y tipos de datos disponibles en el entorno de desarrollo, se evaluaron tres enfoques para la generación de mallas poligonales, descritos a continuación. Los modelos de volumen, además de las rutinas para procesamiento y visualización, fueron implementados sobre SCIAN, y programados en IDL¹¹.

6.1.1 Modelo de Vóxeles

Se construye a partir de las ROIs segmentadas en dos dimensiones, representadas por imágenes (máscaras) binarias. Cada píxel se transforma en un vóxel de acuerdo a su posición en el z -stack, y se agrupa con sus vecinos en una ROI tridimensional.

¹¹ Anexo 2



6.1.2 Superficie Poligonal

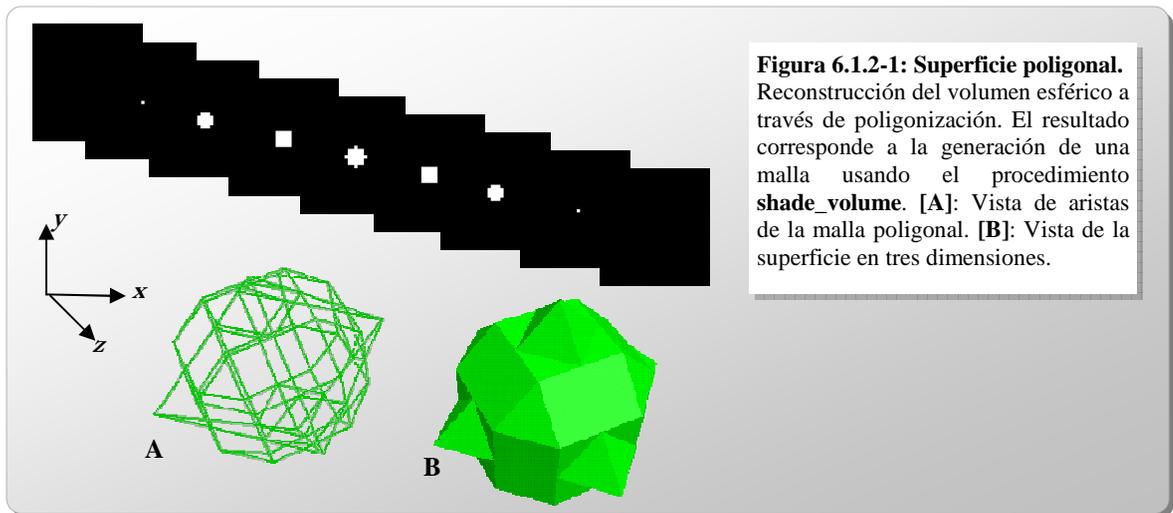
Empleando el *z-stack* de imágenes binarias, se establece un conjunto de vóxeles que corresponde a la superficie de contorno para una región. Esta región se encuentra asociada a un valor de intensidad de referencia, i^* , dentro del *stack* de imágenes.

Este proceso equivale a la detección de isocurvas en dos dimensiones, y se implementa para generar superficies de nivel (isosuperficies en tres dimensiones), típicamente mediante el algoritmo de *Marching Cubes* [Lor87].

Notas sobre la implementación

Dadas las máscaras binarias del *z-stack* (datos de volumen), se deben asociar los vóxeles con un valor de intensidad de contorno (i^*) que separan a una ROI ($i = 255$) del fondo ($i = 0$). Empleando la función **shade_volume**, se obtiene una lista de vértices y polígonos que describen a la superficie de contorno.

shade_volume es una función para obtener datos de volumen en IDL, optimizados para su visualización. Se basa en el algoritmo *PolyPaint* [Kle90], y opera de modo similar a *Marching Cubes*. A diferencia de este último algoritmo, que genera una malla triangular, **shade_volume** genera una estructura que no lo es necesariamente. Aplicaciones con los *z-stacks* de prueba mostraron mallas poligonales compuestas de triángulos y cuadriláteros.

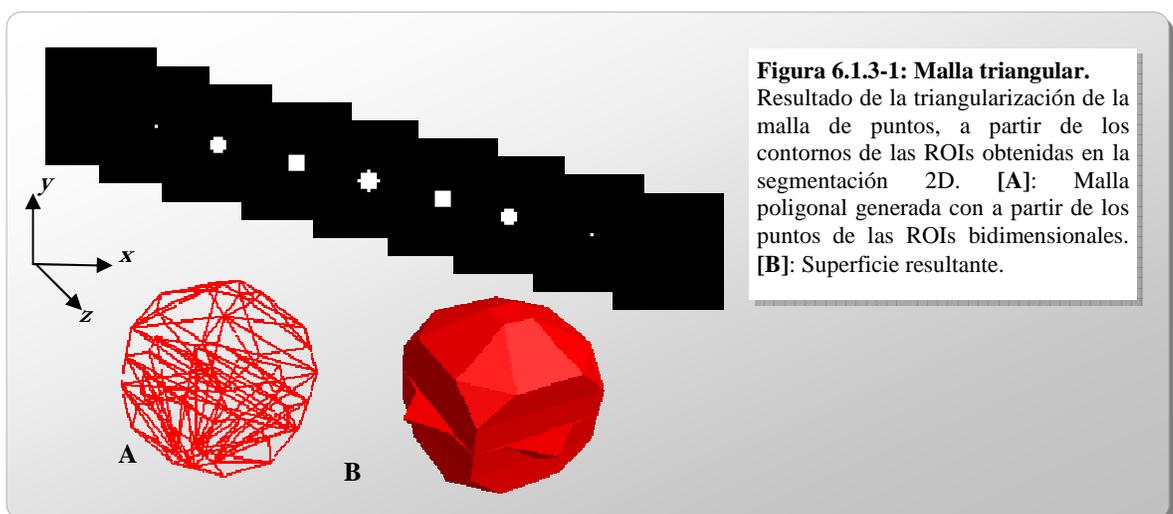


6.1.3 Malla Triangular

Se genera a partir de los datos de cada ROI 2D en un *z-stack*. Uniendo los puntos de contorno (conjunto de vértices) de las ROIs para planos adyacentes en *z*, se forma una serie de triángulos (conjunto de polígonos) que dan conectividad a la malla.

Notas sobre la implementación

La implementación en IDL se realizó a través de objetos de tipo **IDLanROI**, agregados en una instancia de la clase contenedora **IDLanROIgroup**. Estas clases permiten generar ROIs y efectuar análisis a través de los métodos que encapsulan. Así, en cada instancia de **IDLanROI** se capturan los datos para una ROI (x, y) en un plano z_i dado; luego, el método **IDLanROIgroup::ComputeMesh** permite obtener la malla (lista de vértices y triángulos), a partir de la serie de ROIs en la clase contenedora.



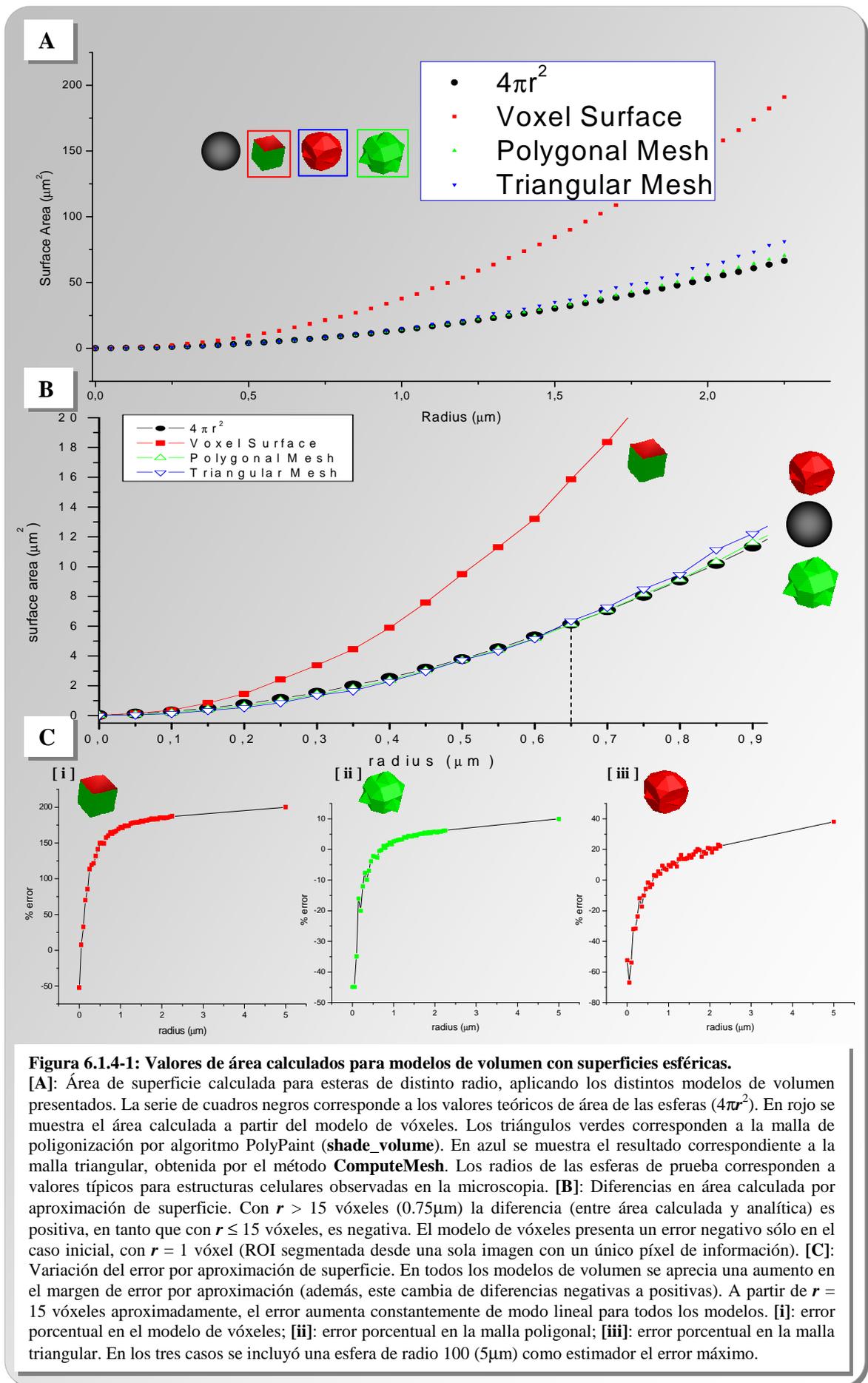
6.1.4 Comparación Experimental de los Modelos

Se compararon los valores de área superficial, obtenidos experimentalmente, a partir de cada modelo de volumen presentado. Se generó una serie de *z-stacks* con imágenes de esferas de distinto tamaño, cuya área de superficie (valor teórico, $4\pi r^2$) se comparó con la obtenida de cada modelo (valor empírico).

Empleando el conjunto de vértices y polígonos resultantes, se calculó el área total de la superficie sumando en todos los polígonos, con la implementación del método **getObjectSurfaceFromPolygons**, definido dentro de la clase contenedora de ROIs tridimensionales, **C_sROI3DGroupObject**.

Resultados

El gráfico de la figura 6.1.4-1 muestra la comparación entre los resultados obtenidos, a partir de los *z-stacks* generados. El radio de las esferas representadas se hizo variar entre 1 y 100 vóxeles (equivalentes al rango de 0.05–5 μ m, para estructuras celulares típicas). Se efectuó la segmentación en dos dimensiones, para luego obtener los modelos de volumen con los métodos presentados.



Discusión

En general, se observó que los dos modelos de malla aplicados lograron una aproximación mucho mejor que el volumen de vóxeles, para determinar el área superficial total. Para el caso de mallas sobre esferas con $r \leq 15$ vóxeles ($0.75\mu\text{m}$), el error al aproximar el área corresponde a una subestimación (valores menores que lo real); en tanto, para $r \geq 15$ vóxeles, se produce un error por sobreestimación de la superficie. A partir de este valor, el margen de error aumenta de modo lineal, llegando al orden del 10% para la malla poligonal, y del 38% para la malla triangular.

El modelo de vóxeles sólo muestra un error negativo (-52%) para $r = 1$ vóxel ($0.05\mu\text{m}$). Posteriormente, la variación del error aumenta linealmente en el rango de 7% y 200%, aproximadamente. La causa está en la diferencia introducida por el modelo de superficie asociado a un conjunto de cubos de igual dimensión (caras cuadradas), que aumenta proporcionalmente al tamaño de la superficie esférica (curva) original.

Por otra parte, los modelos de malla introducen un menor nivel de distorsión, ya que permiten definir polígonos de forma y tamaño variable. Los efectos de rugosidad inducidos por la malla triangular en la generación de la superficie, producen un efecto de distorsión mayor que el de la malla poligonal, que representa de mejor manera la morfología de las esferas, visual y cuantitativamente.

6.1.5 Experimento Complementario de Validación

Se realizó un experimento adicional para observar el margen de error inducido por los modelos, al aproximar una serie de cilindros de altura fija y radio variable. Esto, con el fin de probar el ajuste de los modelos de volumen a este tipo de morfología, similar a la de filamentos en estructuras microscópicas.

El experimento consistió en generar un *z-stack* de imágenes para representar una serie de cilindros de distinto radio. Esto se hizo replicando una imagen con varios círculos (de distinto radio) dentro del *stack* de imágenes, totalizando 90 cortes (*z-slices*). De manera análoga al caso de las esferas, se realizaron los procesos de segmentación 2D y generación de modelos en 3D, para luego obtener el área superficial de cada cilindro, y compararla con su respectivo valor teórico. Se emplearon los mismos modelos de superficie y métodos de cálculo que en el caso anterior. La figura 6.1.5-1 muestra los resultados obtenidos.

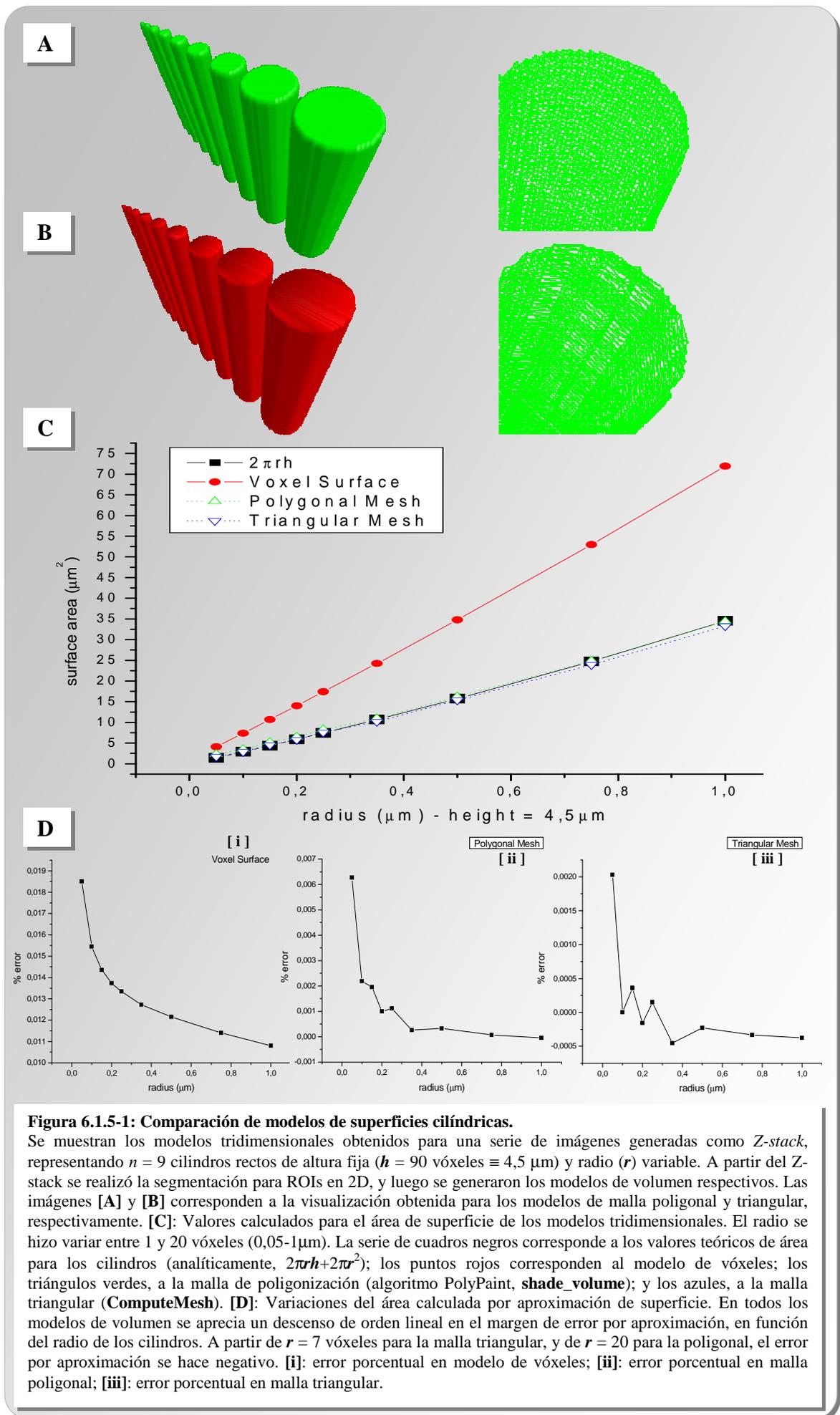


Figura 6.1.5-1: Comparación de modelos de superficies cilíndricas.

Se muestran los modelos tridimensionales obtenidos para una serie de imágenes generadas como *Z-stack*, representando $n = 9$ cilindros rectos de altura fija ($h = 90$ vóxeles $\equiv 4,5 \mu\text{m}$) y radio (r) variable. A partir del *Z-stack* se realizó la segmentación para ROIs en 2D, y luego se generaron los modelos de volumen respectivos. Las imágenes [A] y [B] corresponden a la visualización obtenida para los modelos de malla poligonal y triangular, respectivamente. [C]: Valores calculados para el área de superficie de los modelos tridimensionales. El radio se hizo variar entre 1 y 20 vóxeles ($0,05\text{-}1\mu\text{m}$). La serie de cuadros negros corresponde a los valores teóricos de área para los cilindros (analíticamente, $2\pi rh + 2\pi r^2$); los puntos rojos corresponden al modelo de vóxeles; los triángulos verdes, a la malla de poligonización (algoritmo PolyPaint, *shade_volume*); y los azules, a la malla triangular (*ComputeMesh*). [D]: Variaciones del área calculada por aproximación de superficie. En todos los modelos de volumen se aprecia un descenso de orden lineal en el margen de error por aproximación, en función del radio de los cilindros. A partir de $r = 7$ vóxeles para la malla triangular, y de $r = 20$ para la poligonal, el error por aproximación se hace negativo. [i]: error porcentual en modelo de vóxeles; [ii]: error porcentual en malla poligonal; [iii]: error porcentual en malla triangular.

6.1.6 Conclusiones

De las pruebas realizadas con los distintos modelos para representación de volúmenes y superficies en 3D, la malla poligonal es la que evidenció el menor margen de error en el cálculo de área, respecto de los valores teóricos. Lo anterior permite considerar a este modelo de aproximación, para ser integrado en la implementación de los contornos activos en tres dimensiones.

Como un recurso adicional para mejorar la aproximación de estructuras 3D por modelos discretos, es posible considerar el aumento en la resolución de los contornos (número de puntos o vértices de la malla, vóxeles), en la búsqueda de una mejor representación y caracterización de estructuras complejas a partir los datos originales.

6.2 Formulación de Contornos Activos en Tres Dimensiones

Un contorno en tres dimensiones se define por una función paramétrica de una dimensión más que en el caso bidimensional (ecuación 4.1-1), con la adición de un parámetro r , según

$$\begin{aligned} C &= C(s, r) = [x(s, r), y(s, r), z(s, r)] \\ (s, r) &\in [0,1] \times [0,1] \\ C &: [0,1] \times [0,1] \rightarrow \mathfrak{R}^3 \end{aligned} \quad [6.2-1]$$

sobre una imagen $I : \mathfrak{R}^3 \rightarrow \mathfrak{R}$. En este caso, el funcional de energía interna toma la forma

$$E_{\text{int}} = \int \left(\frac{1}{2} \alpha [|C'_s|^2 + |C'_r|^2] + \frac{1}{2} \beta [|C''_{ss}|^2 + |C''_{rr}|^2] + \frac{1}{2} \beta_{sr} |C''_{sr}|^2 \right) ds dr \quad [6.2-2]$$

$$C'_s = \frac{\partial C}{\partial s}; C'_r = \frac{\partial C}{\partial r}; C''_{ss} = \frac{\partial^2 C}{\partial s^2}; C''_{rr} = \frac{\partial^2 C}{\partial r^2}$$

α y β corresponden a los coeficientes de elasticidad y rigidez, de modo análogo al caso bidimensional, pero ahora ponderando las derivadas del contorno con respecto a dos parámetros, s y r . El término β_{sr} pondera la resistencia a la torsión del contorno, dada por

$$C''_{sr} = \frac{\partial^2 C}{\partial s \partial r} \quad [6.2-3]$$

Para encontrar un contorno, C , que minimice su energía, E , la ecuación de Euler-Lagrange se formula de modo análogo a la ecuación 4.1-3 (caso bidimensional), como

$$\alpha [C''_{ss} + C''_{rr}] + \beta [C^{(4)}_{ssss} + C^{(4)}_{rrrr}] + \beta_{sr} C^{(4)}_{ssrr} = 0 \quad [6.2-4]$$

Incorporando las fuerzas externas (imagen), ∇E_{ext} , se obtiene

$$\nabla E_{ext} + \alpha [C''_{ss} + C''_{rr}] + \beta [C^{(4)}_{ssss} + C^{(4)}_{rrrr}] + \beta_{sr} C^{(4)}_{ssrr} = 0 \quad [6.2-5]$$

Al añadir el término para la evolución dinámica del contorno en el tiempo, haciendo $C = C(s, r, t)$,

$$\begin{aligned} \nabla E_{ext} + \alpha [C''_{ss} + C''_{rr}] + \beta [C^{(4)}_{ssss} + C^{(4)}_{rrrr}] + \beta_{sr} C^{(4)}_{ssrr} &= C'_t \\ C'_t &= \frac{\partial C}{\partial t} \end{aligned} \quad [6.2-6]$$

Esta expresión difiere de la formulación bidimensional en términos de la parametrización de la superficie $C = C(s, r)$, con lo que la elasticidad y la rigidez aparecen separadas en las direcciones determinadas por s y r . Así, siendo

$$\begin{aligned} F_{elást}(C) &= [C''_{ss} + C''_{rr}] \\ F_{rígida}(C) &= [C^{(4)}_{ssss} + C^{(4)}_{rrrr}] \end{aligned} \quad [6.2-7]$$

$$\alpha * F_{elást}(C) + \beta * F_{rígida}(C) + \nabla E_{ext} = 0 \quad [6.2-8]$$

El ajuste del contorno queda formulado de modo análogo al caso bidimensional (ecuación [4.1-5]), expresado como

$$\alpha * F_{elást}(C) + \beta * F_{rígida}(C) + \nabla E_{ext} = \frac{\gamma(C^{t+1} - C^t)}{\Delta t} \quad [6.2-9]$$

6.2.1 Fuerzas Internas

A diferencia del caso bidimensional, en que cada iteración de ajuste del contorno se puede realizar mediante inversión de matrices, para tres dimensiones se requiere proceder de forma particular en cada nodo, identificando y evaluando las vecindades y fuerzas de deformación correspondientes. Esto porque en un contorno bidimensional cada nodo tiene conectividad con sólo dos vecinos (ubicados de forma adyacente en el vector de nodos), mientras que en el caso tridimensional, la malla generada no otorga regularidad en este aspecto: un nodo puede tener tres o más vecinos, que no necesariamente sean adyacentes en el vector de vértices. Se desarrolló entonces el enfoque de [Ahl96] para trabajar con este tipo de modelos, conservando los principios para la definición de las fuerzas internas descritas en la sección 4.1.

Las siguientes definiciones permiten formular de modo general las fuerzas internas para una malla con cualquier tipo de conectividad.

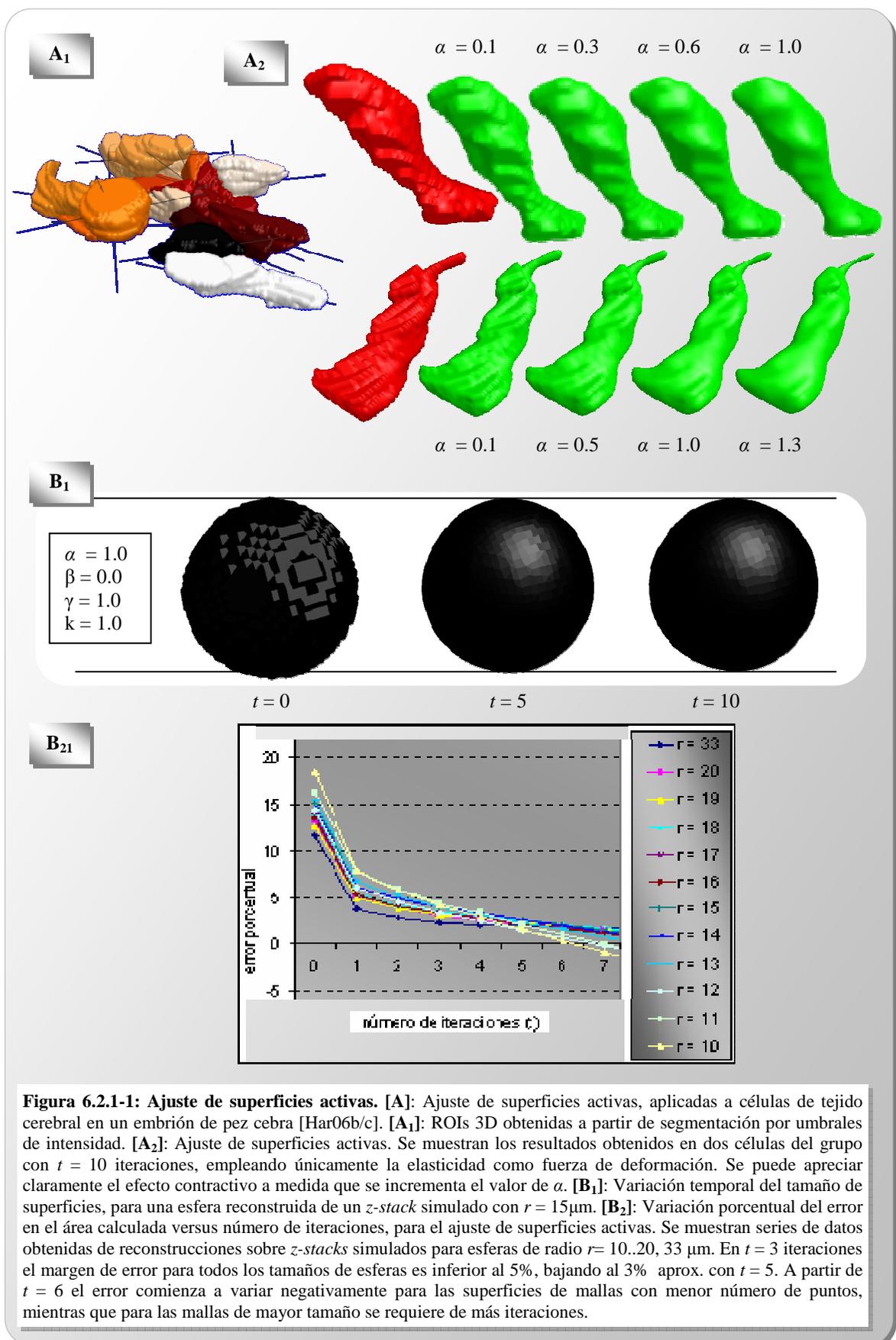
1. Sea una malla de superficie compuesta por:
 - un vector de n puntos de control o vértices, V , con coordenadas (x,y,z)
 - una lista de n_p polígonos, P , que representa a cada polígono, p_i , por un vector de vértices, $V(p_i) \subset V$ en que dos vértices consecutivos dentro del vector (además del último con el primero) definen una arista.
2. Dos puntos distintos, c_i y c_j , se dirán **vecinos a una distancia de un (1) paso** si forman una arista en un polígono cualquiera de la malla. A partir de lo anterior, si un tercer nodo c_k es vecino de c_j , pero no de c_i , se dirá que c_i y c_k son **vecinos a una distancia de dos pasos**.
3. Sea $P_{j,d} = \{p_{j,d,m}\}$, $m = 1 \dots M$, el conjunto de M puntos a una distancia de d pasos desde un punto de control c_j . Se define por $c_{j,d}^*$ al promedio de todos los puntos d -distantes a c_j , tal que

$$c_{j,d}^* = \sum_{m=1}^M \frac{p_{j,d,m}}{|P_{j,d}|} \quad [6.2.1-1]$$

La fuerza de elasticidad para un punto cualquiera c_j se puede expresar por

$$F_{elást}(c_j) = c_{j,1}^* - c_j \quad [6.2.1-2]$$

En esta expresión, los puntos en la vecindad $P_{j,1}$ ejercen una fuerza de arrastre sobre c_j , con un vector resultante igual al promedio de todos ellos, $c_{j,1}^*$. El efecto observado es la contracción de los puntos de la malla, de modo análogo al caso de dos dimensiones, como se muestra en la figura 6.2.1-1.



Para el caso de la rigidez, primero se considera el efecto de “predicción” para la posición de un nodo determinado, dada por sus vecinos a distancia 1 y 2: $2c_{j,1}^* - c_{j,2}^* - c_j$. Este efecto no regulariza la posición de los nodos salientes de la malla, sino que desplaza a los vecinos hacia fuera para nivelarse; al añadir dos veces la elasticidad se suprime este efecto. Se formula entonces

$$\begin{aligned} F_{\text{rígido}}(c_j) &= 2c_{j,1}^* - c_{j,2}^* - c_j - 2F_{\text{elást}}(c_j) \\ F_{\text{rígido}}(c_j) &= 4c_{j,1}^* - c_{j,2}^* - 3c_j \end{aligned} \quad [6.2.1-3]$$

Aplicando las ecuaciones [6.2.1-2] y [6.2.1-3] sobre una curva bidimensional se llega al mismo esquema iterativo desarrollado en la sección 4.1. El esquema presentado permite aplicarse sobre mallas o curvas en las que no es posible obtener una parametrización para calcular las derivadas, o bien su conectividad no permite obtener una forma directa para su resolución (como la inversión de matrices para dos dimensiones).

6.2.2 Implementación

A continuación se presentan los algoritmos implementados para identificar y evaluar las vecindades y fuerzas de deformación, dada una malla de superficie C , para cada nodo, C_j , a través de una lista de polígonos asociada.

Para $i = 1, \text{número_de_iteraciones}$ realizar {

- Determinar para cada punto C_j^i del vector de puntos de control C^i , los conjuntos de puntos en las vecindades $C_{j,1}^i$ y $C_{j,2}^i$
- Determinar para cada punto C_j^i del vector de puntos de control C^i , el promedio de las vecindades $C_{j,1}^i$ y $C_{j,2}^i$
- Calcular el vector

$$C^{i+1} = C^i + [\alpha \cdot (C_{*,1}^i - C_{*,*}^i) + \beta \cdot (C_{*,2}^i - C_{*,*}^i)] / \gamma + F_{\text{ext}}(C^i)$$

} Siguiente, i

La evaluación de $C^{i,*}_1$ (nodos vecinos a distancia 1) para todos los nodos del vector C^i se implementó mediante el siguiente algoritmo:

```
punteroVecinos1: arreglo [0..número_de_puntos-1] de puntero  
Para j = 0, número_de_puntos-1 inicializar punteroVecinos[j]  
Para k = 0, número_de_polígonos-1 realizar {  
    • Para cada arista del polígono [ka, kb] realizar {  
        o añadir el vértice ka a *punteroVecinos1[kb]  
        o añadir el vértice kb a *punteroVecinos1[ka]  
    • Siguiente arista  
    } Siguiente, k  
Para j = 0, número_de_puntos-1 realizar {  
    • Ordenar elementos de *punteroVecinos1[j] por número  
    • Eliminar elementos duplicados de *punteroVecinos1[j]  
    •  $C^{i,*}_1 = [$   
        promedio (vectorPuntosCoordenadaX[*punteroVecinos1[j]]),  
        promedio (vectorPuntosCoordenadaY[*punteroVecinos1[j]]),  
        promedio (vectorPuntosCoordenadaZ[*punteroVecinos1[j]])]  
    } Siguiente, j
```

Para $C_{*,2}^i$ (nodos vecinos a distancia 2) el algoritmo emplea la interpolación de esferas con los nodos vecinos para aproximar una posición en cada nodo de la malla.

```

punteroVecinosCurvatura: arreglo [0..número_de_puntos-1] de puntero
puntos_esfera: arreglo [4] de coma_flotante | doble_precision
para i = 0, número_de_puntos inicializar punteroVecinos2[i]
para i = 0, número_de_puntos realizar {
    radio_suma = 0
    centro_x_suma = 0
    centro_y_suma = 0
    centro_z_suma = 0
    para cada 4_combinacion_puntos(*punteroVecinos1[i]) realizar {
        puntos_esfera = generar_4_combinacion(*punteroVecinos1[i])
        interpolar_esfera(puntos_esfera, radio, cenX, cenY, cenZ)
        radio_suma = radio_suma + radio
        centro_x_suma = centro_x_suma + cenX
        centro_y_suma = centro_y_suma + cenY
        centro_z_suma = centro_z_suma + cenZ
    } Siguiente, 4_combinacion_puntos
    centroPromX = centro_x_suma / j
    centroPromY = centro_y_suma / j
    centroPromZ = centro_z_suma / j
    radioProm = radio_suma / j
    angulo = atan((vectorPuntosCoordenadaX[i]-centroPromX)/
        raiz((vectorPuntosCoordenadaY[i]-centroPromY)^2+
        (vectorPuntosCoordenadaZ[i]-centroPromZ)^2))

```

```

si no(valor_finito(angulo)) entonces angulo = 0

desplazaX = radioProm * seno(angulo)

angulo = atan((vectorPuntosCoordenadaY[i]-centroPromY)/
raiz((vectorPuntosCoordenadaX[i]-centroPromX)^2+
(vectorPuntosCoordenadaZ[i]-centroPromZ)^2))

si no(valor_finito(angulo)) entonces angulo = 0

desplazaY = radioProm * seno(angulo)

angulo = atan((vectorPuntosCoordenadaZ[i]-centroPromZ)/
raiz((vectorPuntosCoordenadaX[i]-centroPromX)^2+
(vectorPuntosCoordenadaY[i]-centroPromY)^2))

si no(valor_finito(angulo)) entonces angulo = 0

desplazaZ = radioProm * seno(angulo)

*punteroVecinosCurvatura[i] = [centroPromX + desplazaX, centroPromY
+ desplazaY, centroPromY + desplazaY]

} Siguiente, i

```

Al término del presente Proyecto se encuentra pendiente la implementación de la rigidez en el ajuste de las superficies.

6.3 Caracterización de Superficies Tridimensionales

Además del área superficial (introducida en la sección 6.1.4), se buscó caracterizar la complejidad de las superficies mediante su curvatura.

Para una superficie tridimensional, la curvatura es una medida de desviación de la superficie con respecto a un plano. Existen varias definiciones de curvatura, asociadas

La curvatura superficial (κ_s) fue calculada para cada nodo de la malla con respecto a sus vecinos, mediante el radio (r) de la esfera cuya ecuación determinan cada nodo y tres de sus vecinos, siendo $\kappa_s = r^{-1}$; para nodos con más de tres vecinos se calculó la curvatura de todas las posibles 3-combinaciones y se obtuvo un valor promedio.

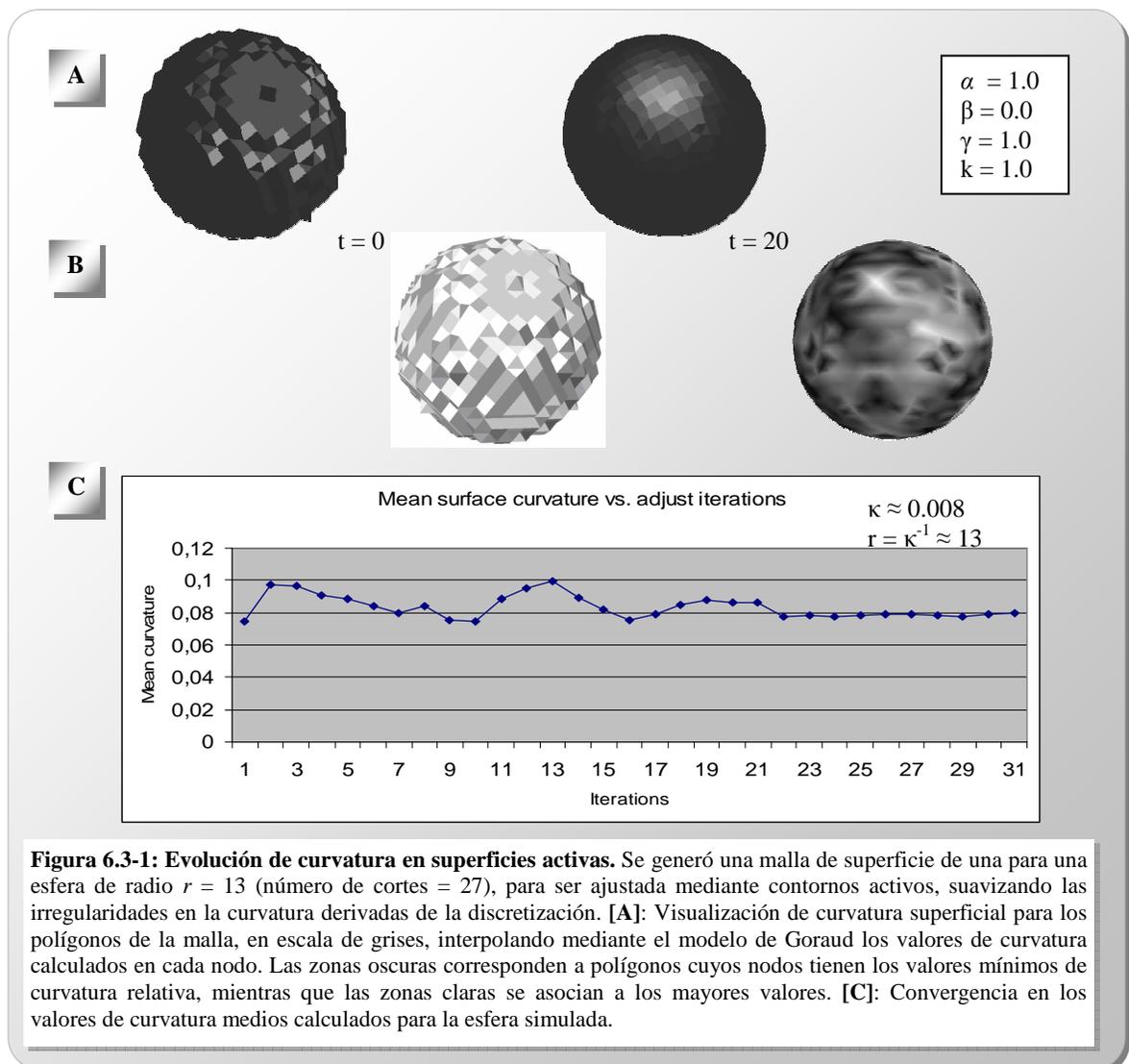


Figura 6.3-1: Evolución de curvatura en superficies activas. Se generó una malla de superficie de una para una esfera de radio $r = 13$ (número de cortes = 27), para ser ajustada mediante contornos activos, suavizando las irregularidades en la curvatura derivadas de la discretización. [A]: Visualización de curvatura superficial para los polígonos de la malla, en escala de grises, interpolando mediante el modelo de Goraud los valores de curvatura calculados en cada nodo. Las zonas oscuras corresponden a polígonos cuyos nodos tienen los valores mínimos de curvatura relativa, mientras que las zonas claras se asocian a los mayores valores. [C]: Convergencia en los valores de curvatura medios calculados para la esfera simulada.

Capítulo 7: Conclusiones

7.1 Cumplimiento de Objetivos

7.1.1 Objetivo General

Planteado como “Desarrollar e implementar algoritmos de contornos activos para la segmentación y parametrización tridimensional de estructuras biológicas”, apunta a:

- i. mejorar la representación espacial de las estructuras segmentadas, y
- ii. caracterizar cuantitativamente las curvas y superficies de contorno de dichas estructuras.

Ambas metas se consideran alcanzadas mediante el cumplimiento de los objetivos específicos del Proyecto.

7.1.2 Objetivos Específicos

- a) *Reconocer e identificar los métodos de segmentación de ROIs a través de contornos activos, en su formulación y aplicación.*

Cumplido a través de la revisión bibliográfica y de la descripción realizada para el proceso de segmentación mediante contornos activos, desarrollada en el presente informe.

- b) *Implementar algoritmos de contornos activos para la segmentación de ROIs en dos dimensiones, sobre la base de SCIAN existente.*

Cumplido mediante una implementación en software (elaborada sobre la plataforma IDL) de los contornos activos en dos dimensiones.

- c) *Extender la funcionalidad de los contornos activos desarrollada en dos dimensiones para la segmentación de ROIs tridimensionales.*

Cumplido parcialmente. Se consiguió modelar el campo de fuerzas externas y la fuerza elástica para contornos en tres dimensiones o superficies activas, faltando la formulación e implementación para la rigidez.

- d) *Combinar, ajustar y optimizar la funcionalidad de los contornos activos con la biblioteca de filtros existentes en SCIAN para la segmentación de estructuras biológicas.*

Cumplido. Las clases implementadas para contornos activos fueron incorporadas al conjunto de herramientas para segmentación de estructuras en SCIAN.

- e) *Extender la funcionalidad de los contornos activos desarrollada en dos y tres dimensiones para la reconstrucción multidimensional y la parametrización de contornos y superficies de ROIs para estructuras biológicas.*

Cumplido. Se implementaron clases y métodos para la caracterización de los modelos de contornos activos (dos dimensiones) y superficies activas (tres dimensiones).

7.2 Aspectos Notables, Ventajas y Limitaciones

La implementación del Proyecto ha contribuido a mejorar los resultados del proceso análisis de morfología celular, además de la visualización de estructuras. Se integraron los modelos de contornos activos al sistema de objetos gráficos de IDL/SCIAN para la reconstrucción y visualización de estructuras a nivel intracelular, celular e intercelular [Har06b/c]. El ajuste de resolución en dos dimensiones, combinado con el balance de fuerzas para el ajuste de los contornos, mejoró en forma significativa la caracterización morfológica de las estructuras segmentadas.

Como limitante destaca el consumo de recursos computacionales para los algoritmos implementados. El tiempo de cálculo para los campos vectoriales se relaciona proporcionalmente con el tamaño de las imágenes de entrada; de igual modo ocurre con el ajuste iterativo de los contornos, relacionado con el número y tamaño de las ROIs. En el caso de las superficies activas, el uso de memoria aumenta considerablemente, ya que las imágenes y los campos vectoriales son arreglos tridimensionales que deben almacenarse en memoria durante el proceso de ajuste, así como las estructuras de datos para cada superficie: vértices y polígonos, más los arreglos de curvatura y vecindades requeridos para el ajuste dinámico. En casos extremos, el procesamiento demanda casi la totalidad de memoria de una estación de trabajo personal (~3GB de RAM), haciendo necesario disminuir la precisión de los cálculos numéricos para poder realizarse (notación de coma flotante de precisión doble a simple).

Para los contornos de tres dimensiones el número de nodos de la malla es fijo, a diferencia de dos dimensiones, en que la interpolación de puntos en las curvas permite variar según sea necesario. Si bien se pueden refinar las máscaras de dos dimensiones para la subsecuente reconstrucción tridimensional, en el eje vertical (z) la distancia entre cortes (z -slices) se mantiene igual, siendo varias veces mayor que en el plano (x, y).

7.3 Futuros Trabajos

Apuntando a las limitaciones y restricciones de la implementación del Proyecto, se plantean los siguientes desarrollos a futuro:

- Aplicación de fuerza rígida en los modelos de superficies activas.
- Ajustes en la distribución de los puntos en mallas de superficie tridimensionales, de modo de regularizar el tamaño de los polígonos en función de su curvatura, variando la resolución de la malla, y compensando la diferencia de resolución en el eje z con respecto al plano (x, y) .
- Determinación de una parametrización óptima para el balance entre resolución y/o distribución de nodos de mallas y las fuerzas para superficies activas, de modo análogo al desarrollo de la sección 4.4.3.
- Herramientas para la visualización de campos vectoriales de fuerzas sobre contornos en tres dimensiones.

7.4 Difusión y Aplicaciones

El cumplimiento de los objetivos del Proyecto se refleja en las herramientas de software desarrolladas para la segmentación y análisis de estructuras biológicas, aplicadas en varios trabajos de investigación (ver Anexo 5 con publicaciones):

- Organización de estructuras monocapas de lípidos en la membrana celular [Fan06, Har05b, Tul06], desarrollados en la Unidad de Biofísica y Fisiología Molecular del CECS, Valdivia, y el Programa de Anatomía y Biología del Desarrollo (ICBM, Facultad de Medicina) de la Universidad de Chile; en colaboración con el Departamento de Química Biológica, Facultad de Ciencias Químicas de la Universidad de Córdoba, Argentina.
- Desarrollo de estructuras celulares en embriones de pez cebra (Har06b/c), en el Programa de Anatomía y Biología del Desarrollo (ICBM, Facultad de Medicina) de la Universidad de Chile.

Adicionalmente se presentaron los contornos activos como una aplicación de modelos matemáticos en biología [Jar05], y como una herramienta de software para la aplicación de herramientas computacionales en el procesamiento de imágenes para la biología [Har06a].

Capítulo 8: Referencias

8.1 Artículos

- [Alv03] Alvarez M., Härtel S., Godoy R., Heyser W. New perspectives in the determination of phosphatase activity in ectomycorrhizae of *Nothofagus obliqua* in forests of southern Chile. *Gayana Botanica* 60(1): 41-46. 2003.
- [Alv04a] Alvarez M., Godoy R., Heyser, W., Härtel S. Surface bound phosphatase activity in living hyphae of ectomycorrhizal fungi of *Nothofagus obliqua*. *Mycologia* 96(3): 479-487. 2004.
- [Alv04b] Alvarez M., Härtel S., Godoy R., Heyser W. Anatomical-physiological determination of surface bound phosphatase activity in ectomycorrhiza of *Nothofagus obliqua* based on image processed confocal fluorescence microscopy. *Soil Biology and Biochemistry* 37(1): 125-132. 2004.
- [Cas93] Caselles V., Catta F., Coll T., Dibos F. A geometric model for active contours. *Numerische Mathematik* 66: 1-31. 1993.
- [Cas97] Caselles V., Kimmel R., Sapiro G. Geodesic active contours. *International Journal of Computer Vision* 22(1): 61-79. 1997.
- [Coh91] Cohen L. On active contour models and balloons. *Computer Vision, Graphics and Image Processing: Image Understanding* 53(2): 211-218. 1991.
- [Egn98] Egner A., Schrader M., Hell S. Refractive index mismatch induced intensity and phase variations in fluorescence confocal, multiphoton and 4Pi-microscopy. *Optics Communications* 153: 211-217. 1998.
- [Fan02] Fanani M., Härtel S., Oliveira R., Maggio B. Bidirectional control of sphingomyelinase activity and surface topography in lipid monolayers. *Biophysical Journal* 83(6): 3416-24. 2002.

- [Fin98] Fink C., Morgan F., Loew L. Intracellular fluorescent probe concentrations by confocal microscopy. *Biophysical Journal* 75:1648-1658. 1998.
- [Har03] Härtel S., Zorn-Kruppa M., Tikhonova S., Heino P., Engelke M., Diehl H. Staurosporine-induced apoptosis in human cornea epithelial cells in vitro. *Cytometry* 8:15-23. 2003.
- [Har04a] Härtel S., Fanani M., Maggio B. Shape transitions and lattice structuring of ceramide-enriched domains generated by sphingomyelinase in lipid monolayers. *Biophysical Journal* 88:287-304. 2004.
- [Har05a] Härtel S., Rojas R., R ath C., Guarda M., Goicoechea O. Identification and classification of di- and triploid erythrocytes by multiparameter image analysis: a new method for the quantification of triploidization rates in rainbow trout. *Archivos de Medicina Veterinaria* 37(2):147-154. 2005.
- [Hel92] Hell S., Stelzer E. Fundamental improvement of resolution with a 4Pi-confocal fluorescence microscope using two-photon excitation. *Optics Communications* 93:277-282. 1992.
- [Hel04] Hell S., Dyba M., Jakobs S. Concepts for nanoscale resolution in fluorescence microscopy. *Neurobiology* 14: 599-609. 2004.
- [Jal04] Jalba A., Wilkinson M., Roerdink J. CPM: a deformable for shape recovery and segmentation based on charged particles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(10):1320-1335. 2004.
- [Kas88] Kass M., Witkin A., Terzopoulos D. Snakes: active contour models. *International Journal of Computer Vision* 1:321-331. 1988.
- [Kim03] Kimmel R. Geometric segmentation of 3D structures. *Image Processing* 2-3: 639-42. 2003.

- [Lin03] Lin G., Adiga U., Olson K., Guzowski J., Barnes C., Roysam B. A hybrid 3D watershed algorithm incorporating gradient cues and object models for automatic segmentation on nuclei in confocal image stacks. *Cytometry* 56(1): 23-36. 2003.
- [Liu03] Liu R., Shang Y., Sachse F., Dössel O. 3D active surface method for segmentation of medical image data: assessment of different image forces. *Biomedizinische Technik* 48(1): 28-29. 2003.
- [Lor87] Lorensen W., Cline H. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics* 21: 163-169. 1987.
- [Mal95] Malladi R., Sethian J. Vemuri B. Shape modeling with front propagation: a level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 17: 158-175. 1995.
- [Mar02a] Martín M. Técnicas clásicas de segmentación de imagen. Laboratory of Mathematics in Imaging, Brigham and Women's Hospital, Harvard Medical School. 2002.
- [Mar02b] Martín M. Contornos activos. Laboratory of Mathematics in Imaging, Brigham and Women's Hospital, Harvard Medical School. 2002.
- [Mci96] McInerney T., Terzopoulos D. Deformable models in medical image analysis: a survey. *Medical Image Analysis* 1(2): 91-108. 1996.
- [Min88] Minsky M. Memoir on inventing the confocal scanning microscope. *Scanning* 10: 128-138. 1988.
<http://web.media.mit.edu/~minsky/papers/ConfocalMemoir.html>
- [Mon01] Montagnat J., Delingette H., Ayache N. A review of deformable surfaces: topology, geometry and deformation. *Image and Vision Computing* 19: 1023-1040. 2001.

- [Osh88] Osher S., Sethian J. A. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulation. *Journal of Computational Physics* 79: 12–49. 1988.
- [Sch98] Schrader M., Bahlmann K., Giese G., Hell S. 4Pi-confocal imaging in fixed biological specimens. *Biophysical Journal* 75: 1659-1668. 1998.
- [Xu98a] Xu Ch., Prince J. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing* 3(7): 359-369. 1998.
- [Xu98b] Xu Ch., Prince J. Generalized gradient vector flow external forces for active contours. *Signal Processing* 71: 131-139. 1998.
- [Xu99] Xu Ch., Pham D., Rettmann M., Yu D., Prince J. Reconstruction of the human cerebral cortex from magnetic resonance images. *IEEE Transactions on Medical Imaging* 18(6): 467-480. 1999.

8.2 Libros

- [Ame92] Ames W.F. (1992) *Numerical Methods for Partial Differential Equations, 3rd Edition*. New York Academic Press.
- [Ang05] Angelini E., Jin Y., Laine A. (2005) *State of the Art of Level Set Methods in Segmentation and Registration of Medical Imaging Modalities. Handbook of Biomedical Image Analysis*: 47-102. Springer Verlag.
- [Ebe04] Eberhard Z. (2004) *Oxford User's Guide to Mathematics*. Oxford University Press.
- [Gon02] González R. y Woods RE. (2002) *Digital Image Processing, Second Edition*. Prentice Hall.
- [Gra98] Gray A. (1998) *Modern Differential Geometry of Curves and Surfaces with Mathematica, 2nd Edition*. CRC Press.

- [Lav50] Lavrentiev M. y Liusternik L. (1950) *Curs de Calcul Variational*. Rumania: Editura Tehnica, 1955.
- [Mys71] Myškis A. (1971) *Advanced Mathematics for Engineers*. MIR Publishers, Moscow, 1975.
- [Osh06] Osher S. y Paragios D. (2006) *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer.
- [Sap01] Sapiro G. (2001) *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press.
- [Vol30] Volterra V. (1930) *Theory of Functionals and of Integral and Integro-differential Equations*. New York: Dover Publications, 1959.
- [Wat98] Watt A. y Policarpo F. (1998) *The Computer Image*. Addison-Wesley.
- [Whi01] Whitaker R., Breen D., Museth K., Soni N. (2001) *Segmentation of Biological Volume Datasets Using a Level-Set Framework*. Volume Graphics 2001:249-263. Vienna: Springer.
- [Wil03] Wilhelm S., Gröbler B., Gluch M., Heinz H. (2003) *Microscopy from Carl Zeiss – Principles - Confocal Laser Scanning Microscopy*. Carl Zeiss Advanced Imaging Microscopy.
- [You95] Young I., Gerbrands J., van Vliet L. (1995) *Fundamentals of Image Processing*. Delft: PH publications, 1995.

8.3 Charlas de Conferencia

- [Caz95] Cazorla M., Cáceres P., Escolano F., Gallardo D., Rizo R. Detección Automática con Snakes y Representación 3D sobre Imágenes Cerebrales. VI Conferencia de la Asociación Española Para la Inteligencia Artificial. Alicante, España, Noviembre 1995.

- [Fio03] Fiorentini S., Larrabide I., Vénere M. J. A Simple 3D Image Segmentation Technique over Medical Data. Simposio de Informática y Salud, Jornadas Argentinas de Informática e Investigación Operativa, Buenos Aires, Argentina, 1-5 de Septiembre, 2003.
- [Har04d] Härtel S., Verdugo E., Montenegro C., Niemeyer M., Fanani M., Barros L., Time Resolved 3-Dimensional Image Processing of Microscopical Data and its Application in Membrane Related Cellular Processes. XVIII Reunión anual, Sociedad de Biología Celular de Chile. Pucón, 13-17 de Octubre, 2004.
- [Har04e] Härtel S., Rojas R., Verdugo E., Montenegro C., Niemeyer M., Cárdenas A., Cornejo I., Álvarez M., Fanani M., Maggio B., Barros L. Deconvolution of Microscopical Data for Time Resolved 3-Dimensional Image Processing for the Study of Membrane Related Processes in Cell Biology. Representation of reality by brain and machines - Crossed views from neurosciences and computer vision. Montevideo, Uruguay, 8-12 November, 2004.
- [Har06a] Härtel S., Jara J., Barros L. F. Time Resolved Image Processing Routines for the Study of Membrana Related Processes with Fluorescence Microscopy. 13th International Symposium on Chromaffin Cell Biology, Pucón, Chile, 7-12 January, 2006.
- [Har06b] Härtel S., Lemus C.G., Jara J., Cabrejos M.E., Concha M.L. Análisis de la morfogénesis asimétrica del órgano parapineal de pez cebra en tres niveles de organización. XX Reunión anual, Sociedad de Biología Celular de Chile, Pucón, Chile, 8-12 de Octubre, 2006.
- [Har06c] Härtel S., Jara J., Lemus C.G., Concha M.L. 3D Morpho-Topological Analysis of the Asymmetric Morphogenesis of the Parapineal Organ. Computational Modelling of Objects represented in Images (CompIMAGE). Coimbra, Portugal, 20-21 October, 2006.
- [Hur99] Hurley D., Nixon M. Carter J. Force Field Energy Functionals for Image Feature Extraction. Proceedings of the British Machine Vision Conference (BMVC99): 604-613. Nottingham, 13-16 September, 1999.

- [Jar05] Jara J., Härtel S. Contornos Activos y Flujo de Vectores de Gradientes Generalizados para la Segmentación y Parametrización de Estructuras Biológicas. Sexto Encuentro Chileno de Biomatemática. Valdivia, Chile, 19-20 de Diciembre, 2005. ¹²
- [Ken95] Kennedy J., Eberhart R. Particle swarm optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks: 1942-1948. Perth, Australia, 27 November -1 December 1995.
- [Kle90] Klemp J., McIrvin M., Boyd W. PolyPaint-A Three-Dimensional Rendering Package. American Meteorology Society Proceedings, 6th International Conference on Interactive Information and Processing Systems. Boston, Massachusetts, 1990.
- [Set87] Sethian J. Numerical Methods for Propagating Fronts. Variational methods for free surface interfaces. Proceedings of the Sept. 1985 Vallambrosa Conference. New York, 1987.
- [Tul06] De Tullio L., Maggio B., Härtel S., Jara J., Fanani ML. The initial surface composition and topography modulate sphingomyelinase-driven sphingomyelin to ceramide conversion in lipid monolayers. Enviado para publicación en *Cell Biochemistry and Biophysics*.
- [Xu00] Xu Ch., Prince J. Global Optimality of Gradient Vector Flow. Proceedings of 34th Annual Conferences on Information Sciences and Systems. Princeton University, 15-17 March, 2000.
- [Xu01] Xu Ch., Yezzi A. Jr., Prince J. A Summary of Geometric Level-Set Analogues for a General Class of Parametric Active Contour and Surface Models. Proceedings of 2001 IEEE Workshop on Variational and Level Set Methods in Computer Vision: 104-111. Vancouver, Canada, July 13, 2001.

¹² Trabajo aceptado para su presentación. Exposición y participación en el Encuentro canceladas por el autor.

8.4 Presentaciones de Póster

- [Har04c] Härtel S., Procesamiento de Imágenes y sus Aplicaciones en el Análisis de Patrones en Sistemas Biológicos. Hospital Regional, Nefrología, Valdivia, Chile, 6 de Agosto, 2004.
- [Har05b] Härtel S., Fanani ML., Jara J., Maggio B. Active Contours Quantify Shape Transitions and Multi-Scale Organization of Sphingomyelinase Generated Ceramide-Enriched Domains in Lipid Monolayers Physics of Life: Dynamics of single molecules and dynamics of biological networks. Humlebæk, Denmark, 21-27 August, 2005.
- [Fan06] Fanani ML., Härtel S., Jara J., Maggio B. Sphingomyelinase-induced non-equilibrium phenomena in biointerfaces. VI Congreso Iberoamericano de Biofísica. Madrid, España, 24-27 Septiembre, 2006.
- [Roj04] Rojas R., Härtel S., Identification and Classification of Di and Triploid Erythrocytes by Multi-parameter Image Analysis: A New Method for the Quantification of Triploidization Rates in Rainbow Trout. Representation of reality by brain and machines. Crossed views from neurosciences and computer vision. Montevideo, Uruguay, 8-12 November, 2004.

8.5 Tesis

- [Ahl96] Ahlberg J. (1996) Active contours in Three Dimensions. Computer Vision Laboratory, Linköping University, Sweden.
- [Bar02] Barría P., Rojas R., Espinoza V. (2002) Plataforma de Identificación, Clasificación y Análisis de Imágenes, obtenidas a partir de Muestras Celulares. Universidad Austral de Chile.
- [Boi96] Boier I. (1996) Scientific Data Visualization and digital Image Processing for Structural Biology. PhD Thesis, Department of Computer Science, Purdue University.

- [Kem99] van Kempen G. (1999) Image Restoration in Fluorescence Microscopy. Technische Universiteit Delft, Netherlands.
- [Oya03] Oyarzo M. (2003) Segmentación y Análisis Automatizado de Objetos en Movimiento Aplicado al Estudio de Sistemas Biológicos. Universidad Austral de Chile.
- [Ver05] Verdugo E. (2005) Reconstrucción, Visualización y Parametrización Tridimensional de Estructuras Biológicas a través de Imágenes Confocales de Fluorescencia. Universidad Austral de Chile.

8.6 Direcciones Web

- [URL1] Molecular Expressions (2006) Molecular Expressions Microscopy Primer: Specialized Microscopy Techniques - Confocal Microscopy. Florida State University.
<http://micro.magnet.fsu.edu/primer/techniques/confocal/index.html>
- [URL2] Confocal Laser Scanning Microscopy (2006) Wikipedia.
http://en.wikipedia.org/wiki/Confocal_laser_scanning_microscopy
- [URL3] Microscopia – Monografías.com (2004).
<http://www.monografias.com/trabajos7/micro/micro.shtml>
- [URL4] Centro de Estudios Científicos.
<http://www.cecs.cl>
- [URL5] SCIAN - Scientific Image Analysis.
<http://www.scian.cl>
- [URL 6] Prince J., Xu Ch. (2004) Active Contours and Gradient Vector Flow.
<http://iacl.ece.jhu.edu/projects/gvf/>

- [URL 7] Fanning D. (2004) Coyote's Guide to IDL Programming.
<http://www.dfanning.com>
- [URL 8] Fanning D. (2004) Active Contouring (Snakes) in IDL.
http://www.dfanning.com/ip_tips/snakes.html
- [URL9] Weisstein E. (1999) "Calculus of Variations -- From MathWorld".
<http://mathworld.wolfram.com/CalculusofVariations.html>
- [URL10] Weisstein E. (1999) "Functional Derivative -- From MathWorld".
<http://mathworld.wolfram.com/FunctionalDerivative.html>
- [URL11] Weisstein E. (1999) "Euler-Lagrange Differential Equation -- From MathWorld".
<http://mathworld.wolfram.com/Euler-LagrangeDifferentialEquation.html>

Anexos

Anexo 1: Cálculo Variacional

Este anexo presenta los conceptos elementales del cálculo variacional, relativos al problema de minimización de energía de los contornos activos expuesto en el Capítulo 2.

El cálculo variacional fue fundado en el siglo XVIII por L. Euler y J. L. Lagrange, y fue desarrollado extensivamente por otros matemáticos durante el siglo XIX. Los principios variacionales son de gran importancia a nivel científico, puesto que proporcionan una aproximación general a varios problemas de física aplicada. Los métodos del cálculo variacional han probado ser efectivos en la resolución práctica de problemas, tanto en sus aspectos cuantitativos como cualitativos [Mys71].

El cálculo variacional tiene mucho en común con la teoría de extremos (máximos y mínimos) de funciones, y constituye en cierto modo una extensión de ésta. “Es una rama de la matemática que constituye un tipo de generalización del cálculo; el cálculo de variaciones busca encontrar la ruta, curva, superficie, etc., para la cual una función dada tiene un valor estacionario (...un mínimo o un máximo)” [URL9].

1.1 El Concepto de Funcional

“Una cantidad z es un funcional de la función $y(x)$ en el intervalo $[a, b]$ ($x \in [a, b]$) cuando ella depende de todos los valores que toma $y(x)$ cuando x varía en $[a, b]$ ” [Vol30].

Las formas de notación para representar esta relación son:

$$\begin{aligned} z &= F\left[\left(y(x)\right)\right]; z = F\left[\left(y(x)\right)\right]; \\ z &= F[y(x)]; \\ z &= F(x, y) \end{aligned} \tag{A1.1-1}^{13}$$

En los problemas de optimización de funcionales se buscan funciones en lugar de valores puntuales; en otras palabras, se deben determinar una o varias funciones para extremar una cantidad o condición, dada por el funcional, satisfaciendo ciertas condiciones de borde.

¹³ Por simplicidad se emplea la forma de notación $z = F[y(x)]$. Para no perder especificidad se debe tener en cuenta que siempre los funcionales son evaluados en un intervalo $[a, b]$ dado.

La solución a los problemas de optimización en el cálculo de variaciones se apoya en la extensión de varios conceptos ya establecidos para el cálculo sobre una o varias variables.

1.2 Derivación de un Funcional

Extendiendo las propiedades de continuidad de las funciones al campo de los funcionales, se extiende también el concepto de derivadas, necesario para determinar analíticamente formas de optimización.

Asumiendo que un funcional $z = F[y(x)]$ es continuo sobre el intervalo $[a, b]$, se requiere describir la forma en que varía z , según lo hace la función y . Para el caso del funcional F sobre la función variable y , su derivada, F'_y , esto se puede expresar de la forma

$$F'_y[y(x)] = \frac{F[y(x) + \delta y(x)] - F[y(x)]}{\delta y(x)} \quad [\text{A1.2-1}]$$

en que $\delta y(x)$ es una **variación** o **variacional** de F , entendiéndose como una función de variación arbitraria de $y(x)$, y cuya desviación con respecto a cero sea relativamente pequeña¹⁴.

1.3 Funcionales Integrales y Optimización: Ecuación de Euler-Lagrange

Se define el funcional integral I , como

$$I\{y(x)\} = \int_a^b F[y(x)] dx \quad [\text{A1.3-1}]$$

Para que exista un valor extremo en este tipo de funcionales, la condición necesaria está dada por la llamada ecuación de Euler-Lagrange¹⁵ (ver derivación en [Lav50, Mys71, Vol30, URL10/11]). Esta ecuación establece que

$$F'_y[y(x)] = 0 \quad [\text{A1.3-2}]$$

¹⁴ La desviación de una función con respecto a otra (ya se trate de una constante, como cero, o bien de otra función definida) debe entenderse dentro de un espacio normado de funciones que satisfaga ciertas condiciones. Del mismo modo, deberá tenerse en cuenta (según se requiera) cuando las funciones sobre las que se define F están en un espacio de funciones continuas en $[a, b]$, como también si son diferenciables en todo el intervalo. Una descripción formal de los constructos matemáticos necesarios, dada su extensión, escapa al alcance del presente texto, por lo que se remite al lector a la bibliografía para mayor detalle.

¹⁵ En varios textos esta ecuación es llamada simplemente ecuación de Euler.

En términos generales, esta ecuación se deriva de considerar una variación δI sobre I , [Mys71], definida como

$$\delta I\{\bar{y}, \delta y\} = 0 \quad [A1.3-3]$$

La interpretación de esta ecuación, para asociarla a la existencia de un valor extremo (mínimo o máximo) de I es análoga a la de la derivada de una función analítica [Lav50].

1.4 Aplicación a Contornos Activos

1.4.1 Funcional de Energía de un *Snake* en Dos Dimensiones

Se considera una función genérica $C = C(s)$, con $s \in [0, 1]$ ¹⁶; en analogía a $y(x)$, y con $x \in [a, b]$. El funcional de energía de un *snake* se define por

$$E = E(s) = \int_0^1 \frac{1}{2} \left[\alpha \left| \frac{dC}{ds} \right|^2 + \beta \left| \frac{d^2C}{ds^2} \right|^2 \right] + E_{ext}(C) ds \quad [A1.4.1-1]$$

E es un funcional de la función $C(s)$, que depende de su primera y su segunda derivada, dC/ds y d^2C/ds^2 , respectivamente. La forma general de este tipo de funcional [Mys71] corresponde a

$$I\{y\} = \int_z^b F[y(x), y'(x), y''(x)] dx \quad [A1.4.1-2]$$

con condiciones de borde

$$y(a) = y_a, y(b) = y_b, y'(a) = y'_a, y'(b) = y'_b \quad [A1.4.1-3]$$

Reemplazando los términos de [A1.4.1-1] en [A1.4.1-2] se tiene

$$\begin{aligned} y(x) &= C(s); \\ y'(x) &= \frac{dC}{ds} = C'(s); \\ y''(x) &= \frac{d^2C}{ds^2} = C''(s) \end{aligned} \quad [A1.4.1-4]$$

¹⁶ X puede ser una función escalar, o bien un vector de 2 más dimensiones, como es el caso de las curvas $X = (x, y)$ o superficies $X = (x, y, z)$ activas, presentadas en los capítulos 2 y 3, respectivamente.

Así, las condiciones de borde de [A1.4.1-3] quedan como

$$\begin{aligned} y(a) = E_{ext}(C(a)) = y(b) = E_{ext}(C(b)); \\ y'(a) = C'(a) = y'(b) = C'(b) \end{aligned} \quad [A1.4.1-5]$$

La variación un funcional, δI , incorporando las derivadas, queda expresada por

$$\delta I = \int_a^b (F'_{y'} \delta y + F'_{y''} \delta y' + F'_{y'''} \delta y'') dx \quad [A1.4.1-6]$$

Luego, el valor extremo de I se determina al minimizar la variación de todas las funciones que lo definen.

La condición necesaria para extremar el valor de E , mediante la ecuación de Euler-Lagrange [Mys71, Vol30], se traduce en

$$F'_{y'} - \frac{d}{dx} F'_{y''} + \frac{d^2}{dx^2} F'_{y'''} = 0 \quad [A1.4.1-7]$$

Reemplazando para el funcional de energía E , de la ecuación [A1.4.1-1], se llega a

$$\nabla E_{ext}(C(s)) - \alpha C'''(s) + \beta C''''(s) = 0 \quad [A1.4.1-8]$$

Ordenando términos se tiene

$$\alpha C'''(s) - \beta C''''(s) - \nabla E_{ext}(C(s)) = 0 \quad [A1.4.1-9]$$

Para encontrar la solución a esta ecuación, se recurre a los métodos numéricos, que permiten aproximarla por la vía iterativa. Además, la forma general de las funciones $C(s)$ y E_{ext} permiten extender su aplicación de magnitudes escalares a funciones vectoriales sobre dominios de distinta dimensión, como es el caso de las abordadas por el presente Proyecto.

1.4.2 Funcional de Energía de un Contorno en Tres Dimensiones

Un contorno activo en tres dimensiones, o superficie activa, se define por una superficie paramétrica, $C = C(s, \mathbf{r})$. El funcional de energía correspondiente (Capítulo 6) está dado por

$$E = \int_0^1 \frac{1}{2} \left[\alpha_s \left| \frac{\partial C}{\partial s} \right|^2 + \alpha_r \left| \frac{\partial C}{\partial \mathbf{r}} \right|^2 + \beta_s \left| \frac{\partial^2 C}{\partial s^2} \right|^2 + \beta_r \left| \frac{\partial^2 C}{\partial \mathbf{r}^2} \right|^2 + \beta_{sr} \left| \frac{\partial C}{\partial s \partial \mathbf{r}} \right|^2 \right] + E_{ext}(C) ds dr \quad [A1.4.2-1]$$

La forma general asociada [Mys71] es

$$I\{C(s, r)\} = \int_0^1 F\left(C, \frac{\partial C}{\partial s}, \frac{\partial C}{\partial r}, \frac{\partial^2 C}{\partial s^2}, \frac{\partial^2 C}{\partial r^2}, \frac{\partial^2 C}{\partial s \partial r}\right) ds dr \quad [\text{A1.4.2-2}]$$

La minimización se realiza a partir de la ecuación de Euler-Lagrange, en la llamada ecuación de Euler-Ostrogradsky [Mys71]

$$F'_z - \frac{\partial}{\partial s} F'_{\frac{\partial C}{\partial s}} - \frac{\partial}{\partial r} F'_{\frac{\partial C}{\partial r}} = 0 \quad [\text{A1.4.2-3}]$$

Incluyendo las segundas derivadas en la formulación, como en [1.4.1-7], se llega a

$$F'_z - \frac{\partial}{\partial s} F'_{\frac{\partial C}{\partial s}} - \frac{\partial}{\partial r} F'_{\frac{\partial C}{\partial r}} + \frac{\partial^2}{\partial s^2} F'_{\frac{\partial^2 C}{\partial s^2}} + \frac{\partial^2}{\partial r^2} F'_{\frac{\partial^2 C}{\partial r^2}} + \frac{\partial^2}{\partial r^2} F'_{\frac{\partial^2 C}{\partial s \partial r}} = 0 \quad [\text{A1.4.2-4}]$$

Así, la implementación en términos de [A1.4.2-1] toma la forma de

$$\nabla E_{ext}(C) + \alpha_s \frac{\partial^2 C}{\partial s^2} + \alpha_r \frac{\partial^2 C}{\partial r^2} + \beta_s \frac{\partial^4 C}{\partial s^4} + \beta_r \frac{\partial^4 C}{\partial r^4} + \beta_{sr} \frac{\partial^2 C}{\partial s^2 \partial r^2} = 0 \quad [\text{A1.4.2-5}]$$

Anexo 2: Clases y Métodos Implementados

En esta sección se presentan los tipos básicos de datos proporcionados por IDL, en base a los cuales se diseñaron e implementaron las estructuras.

float: número en coma flotante de precisión simple, representado por 32 bits con 6 a 7 decimales significativos, en el rango de $\pm 10^{38}$.

double: número en coma flotante de precisión doble, representado por 64 bits con 14 decimales significativos, en el rango de $\pm 10^{308}$.

integer: número entero con signo representado por:

integer (16 bits): rango [-32168, 32767]

unsigned integer (16 bits): rango [0, 65535]

long integer (64 bits): rango [-9.223.372.036.854.775.808,
+9.223.372.036.854.775.807]

unsigned long integer (64 bits): rango [0, 18446744073709551615]

puntero: definido mediante la instrucción **ptr_new()**, permite asignar memoria dinámicamente para almacenamiento de datos o estructuras de datos. Un arreglo de punteros puede definirse mediante la instrucción **ptrArr(dim)**, en que **dim** permite especificar las dimensiones del arreglo.

2.1 Clase C_sActiveContour

2.1.1 Propósito

Implementación de los contornos activos en dos dimensiones.

2.1.2 Atributos

pImage

- Tipo: puntero.
- Propósito: puntero a la imagen 2D para la cual el contorno es ajustado. La imagen 2D debe ser representada por un arreglo bidimensional $\{x_i, y_j\}$ con valores de intensidad I_{ij} enteros sin signo.
- Valor por defecto: no aplicable; debe especificarse una imagen al crear una instancia de la clase.

pU, pV

- Tipo: puntero.
- Propósito: punteros a las componentes x e y del campo vectorial GVF (arreglos de dos dimensiones).
- Valor por defecto: nulo; los vectores del campo son calculados mediante los métodos **calcGVF** o **calcGGVF**.

pX, pY

- Tipo: puntero.
- Propósito: punteros a los arreglos de coordenadas x e y del contorno.
- Valor por defecto: especificado al crear una instancia de la clase, o bien mediante el método **setContour**.

npts

- Tipo: **unsigned integer**.
- Propósito: número de puntos del contorno.

- Valor por defecto: 0 si no se especifican las coordenadas (x, y) del contorno al crear una instancia de la clase, de lo contrario corresponderá al número de componentes del vector de coordenadas. Este valor puede modificarse de modo indirecto al cambiar la resolución del contorno mediante el método `arcSample`.

alpha

- Tipo: `float`.
- Propósito: coeficiente de elasticidad (tensión) del contorno. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

beta

- Tipo: `float`.
- Propósito: coeficiente de rigidez del contorno. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

gamma

- Tipo: `float`.
- Propósito: coeficiente de viscosidad del contorno. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

kappa

- Tipo: `float`.
- Propósito: coeficiente para las fuerzas externas. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

mu

- Tipo: `float`.
- Propósito: parámetro de regularización para la función detector de bordes. Su valor debe ser mayor que 0.
- Valor por defecto: 1.0

gvf_iterations

- Tipo: **unsigned integer**.
- Propósito: número de iteraciones para el cálculo del campo GVF. Este valor debe ser mayor que 0.
- Valor por defecto: 10

iterations

- Tipo: **unsigned integer**.
- Propósito: número de iteraciones para el cálculo del ajuste del *snake*. Este valor debe ser mayor que 0.
- Valor por defecto: 10

2.1.3 Métodos

init

- Propósito: constructor de la clase.
- Parámetros:
 - **image**: arreglo bidimensional $\{x_i, y_j\}$ de intensidades I_{ij}
 - **x**: vector de coordenadas en x del contorno; debe especificarse junto con **y**.
 - **y**: vector de coordenadas en y del contorno; debe especificarse junto con **x**.
 - **alpha**: coeficiente de elasticidad del contorno.
 - **beta**: coeficiente de rigidez del contorno.
 - **gamma**: coeficiente de viscosidad del contorno.
 - **kappa**: factor de ponderación para el campo vectorial GVF/GGVF.
 - **mu**: parámetro de regularización para la función detector de bordes.
 - **iterations**: número de iteraciones de ajuste del contorno.
 - **gvf_iterations**: número de iteraciones para el cálculo del campo GVF/GGVF.
- Valor de retorno:

- 1, si el método se ejecutó correctamente y se creó una instancia de la clase;
- -1, en caso de encontrar un error en los parámetros de entrada;
- 0, en caso de producirse una excepción en tiempo de ejecución.

cleanup

- Propósito: destructor de la clase. Libera los punteros definidos en ésta.
- Parámetros: ninguno.
- Valor de retorno: ninguno.

getCoords

- Propósito: función que retorna las coordenadas (x, y) correspondientes a los puntos que conforman la curva de contorno.
- Parámetros: ninguno.
- Valor de retorno: arreglo bidimensional con las coordenadas x e y del contorno.

getCir3, xyRes

- Propósito: calcular la curvatura para cada punto del contorno, como el inverso del radio del círculo que describe cada punto y sus 2 adyacentes.
- Parámetros:
 - **xyRes**, factores de escala (x, y) en forma de par ordenado [escalaX, escalaY].
- Valor de retorno: arreglo unidimensional con los valores de curvatura calculados para cada punto del contorno.

getCurvD, xyRes

- Propósito: calcular la curvatura extrínseca para cada punto del contorno, mediante aproximación de derivadas.
- Parámetros:
 - **xyRes**, factores de escala (x, y) en forma de par ordenado [escalaX, escalaY].

- Valor de retorno: arreglo unidimensional con los valores de curvatura calculados para cada punto del contorno.

getDistance, xyRes

- Propósito: calcular las distancias euclidianas entre cada par de puntos contiguos del contorno.
- Parámetros:
 - **xyRes**, factores de escala (x , y) en forma de par ordenado [escalaX, escalaY].
- Valor de retorno: arreglo bidimensional con los valores de distancia euclidiana calculados.

getGradients, xyRes

- Propósito: calcular los gradientes para cada punto del contorno, contando su diferencia de posición con respecto al punto que lo precede (en sentido antihorario).
- Parámetros:
 - **xyRes**, factores de escala (x , y) en forma de par ordenado [escalaX, escalaY].
- Valor de retorno: arreglo bidimensional que contiene los vectores gradiente para las coordenadas x e y del contorno.

getInflexPoints, curvArray

- Propósito: evaluar los cambios de signo en la **curvatura extrínseca** de los puntos del contorno, recorriéndolo en sentido horario.
- Parámetros:
 - **curvArray**: arreglo de dimensión igual al número de puntos de contorno, conteniendo los valores de curvatura en correspondencia uno a uno.
- Valor de retorno: arreglo de dimensión igual al número de puntos del contorno, con valor
 - 0 si en el punto no existe curvatura (segmento de recta).

1 si en el punto existe un cambio de curvatura en sentido positivo.

2 si en el punto existe un cambio de curvatura en sentido negativo.

getNpts

- Propósito: función que retorna el número de puntos del contorno.
- Valor de retorno: número de puntos del contorno.

getPerimeter

- Propósito: función que retorna el perímetro total del contorno, calculado como la suma de las distancias euclidianas para cada segmento de éste, definido por dos puntos consecutivos.
- Parámetros: ninguno.
- Valor de retorno: perímetro del contorno.

arcSample, points

- Propósito: distribuir o interpolar los puntos del contorno en forma equidistante.
- Parámetros:
 - **points** (opcional): número de puntos que interpolan el contorno.
- Valor de retorno: ninguno.

adjustContour

- Propósito: Aplicar el ajuste iterativo del contorno, en función de las fuerzas calculadas para cada punto, ponderadas por los parámetros respectivos.
- Parámetros: ninguno.
- Valor de retorno: arreglo de coordenadas $\{x_i, y_j\}$ del contorno ajustado.

calcGVF

- Propósito: Cálculo del campo vectorial de flujo de vectores de gradiente (GVF).
- Parámetros: ninguno.
- Valor de retorno: ninguno.

calcGGVF

- Propósito: Cálculo del campo vectorial de flujo de vectores de gradiente generalizado (GGVF).
- Valor de retorno: ninguno.

edgemap

- Propósito: Método para calcular el mapa de aristas sobre la imagen (**pImage**).
- Parámetros: ninguno.
- Valor de retorno: ninguno.

gradient, image, direction

- Propósito: Función que retorna el gradiente de **image** en la dirección especificada por **direction**, para matrices de imagen tridimensionales.
- Parámetros
 - **image**: imagen de entrada, arreglo bidimensional con valores numéricos de intensidad.
 - **direction**: indicador de dirección para calcular la imagen gradiente, pudiendo ser 0 (**x**) o 1 (**y**); valor por defecto: 0.
- Valor de retorno: arreglo de igual dimensión y tamaño que **image**, conteniendo los valores del gradiente calculado para la dirección especificada.

laplacian, image

- Propósito: Función que retorna el laplaciano de una matriz (imagen) bidimensional.
- Parámetros
 - **image**: imagen de entrada, arreglo bidimensional con valores numéricos de intensidad.
- Valor de retorno: arreglo de igual dimensión y tamaño que **image**, resultante de aplicar sobre ésta el operador laplaciano.

setContour, x, y

- Propósito: Método para definir las coordenadas **x** e **y** del contorno.

- Parámetros:
 - **x, y**: vectores de coordenadas que definen los puntos del contorno.
- Valor de retorno: ninguno.

setParams, alpha, beta, gamma, kappa, mu, iterations, gvf_iterations

- Propósito: Modificar valores de los atributos de la clase.
- Parámetros:
 - **alpha** (opcional): coeficiente de elasticidad.
 - **beta** (opcional): coeficiente de elasticidad.
 - **gamma** (opcional): coeficiente de elasticidad.
 - **kappa** (opcional): coeficiente de elasticidad.
 - **mu** (opcional): coeficiente de elasticidad.
 - **iterations** (opcional): coeficiente de elasticidad.
 - **gvf_iterations** (opcional): coeficiente de elasticidad.
- Valor de retorno: ninguno.

getParam, paramName

- Propósito: Método para obtener el valor de un atributo de la clase.
- Parámetros:
 - **paramName**: nombre del atributo cuyo valor es requerido; puede indicar a cualquiera de los atributos susceptibles de modificación en el método **setParams**.
- Valor de retorno: valor del atributo indicado por **paramName**.

2.2 Clase C_sActiveContour3D

2.2.1 Propósito

Implementación de contornos activos en tres dimensiones.

2.2.2 Atributos

pImage

- Tipo: puntero.
- Propósito: puntero a la imagen 3D para la cual el contorno es ajustado. La imagen 3D debe ser representada por un arreglo tridimensional $\{x_i, y_j, z_k\}$ con valores de intensidad I_{ijk} enteros sin signo.
- Valor por defecto: no aplicable; debe especificarse una imagen al crear una instancia de la clase.

pU, pV, pW

- Tipo: puntero.
- Propósito: punteros a las componentes x, y y z del campo vectorial GVF (arreglos de tres dimensiones).
- Valor por defecto: nulo; los vectores del campo son calculados mediante los métodos `calcGVF` o `calcGGVF`.

pX, pY, pZ

- Tipo: puntero.
- Propósito: punteros a los arreglos de coordenadas x, y y z del contorno.
- Valor por defecto: especificado al crear una instancia de la clase, o bien mediante el método `setContour`.

pPolygons

- Tipo: puntero
- Propósito: Puntero a la lista de polígonos de la superficie de contorno.
- Valor por defecto: especificado al crear una instancia de la clase, o bien mediante el método `setContour`.

npts

- Tipo: **unsigned integer**.
- Propósito: número de puntos del contorno.
- Valor por defecto: 0 si no se especifican las coordenadas (x , y , z) del contorno al crear una instancia de la clase, de lo contrario corresponderá al número de componentes del vector de coordenadas.

alpha

- Tipo: **float**.
- Propósito: coeficiente de elasticidad. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

beta

- Tipo: **float**.
- Propósito: coeficiente de rigidez. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

gamma

- Tipo: **float**.
- Propósito: coeficiente de viscosidad. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

kappa

- Tipo: **float**.
- Propósito: coeficiente para las fuerzas externas. Su valor debe ser mayor o igual a 0.
- Valor por defecto: 1.0

mu

- Tipo: **float**.
- Propósito: parámetro de regularización para la función detector de bordes. Su valor debe ser mayor que 0.
- Valor por defecto: 1.0

gvf_iterations

- Tipo: **unsigned integer**.
- Propósito: número de iteraciones para el cálculo del campo GVF. Este valor debe ser mayor que 0.
- Valor por defecto: 10

iterations

- Tipo: **unsigned integer**.
- Propósito: número de iteraciones para el cálculo del ajuste del *snake*. Este valor debe ser mayor que 0.
- Valor por defecto: 10

2.2.3 Métodos

init

- Propósito: constructor de la clase.
- Parámetros:
 - **image**: arreglo tridimensional $\{x_i, y_j, z_k\}$ de intensidades I_{ikj}
 - **x**: vector de coordenadas en x del contorno; debe especificarse junto con **y**, **z** y **polygons**.

- **y**: vector de coordenadas en y del contorno; debe especificarse junto con **x**, **z** y **polygons**.
- **z**: vector de coordenadas en z del contorno; debe especificarse junto con **x**, **y** y **polygons**.
- **polygons**: lista de polígonos del contorno; debe especificarse junto con **x**, **y** y **z**.
- **alpha**: coeficiente de elasticidad del contorno.
- **beta**: coeficiente de rigidez del contorno.
- **gamma**: coeficiente de viscosidad del contorno.
- **kappa**: factor de ponderación para el campo vectorial GVF/GGVF.
- **mu**: parámetro de regularización para la función detector de bordes.
- **iterations**: número de iteraciones de ajuste del contorno.
- **gvf_iterations**: número de iteraciones para el cálculo del campo GVF/GGVF.
- Valor de retorno:
 - 1, si el método se ejecutó correctamente y se creó una instancia de la clase;
 - -1, en caso de encontrar un error en los parámetros de entrada;
 - 0, en caso de producirse una excepción en tiempo de ejecución.

cleanup

- Propósito: destructor de la clase. Libera los punteros definidos en ésta.
- Parámetros: ninguno.
- Valor de retorno: ninguno.

getCoords

- Propósito: función que retorna las coordenadas (x, y, z) correspondientes a los puntos que conforman la curva de contorno.
- Parámetros: ninguno.
- Valor de retorno: arreglo tridimensional con las coordenadas x, y y z del contorno.

computeCurv4, x, y, z, xC, yC, zC, r

- Propósito: calcular la curvatura para cada punto del contorno mediante la ecuación de esfera que determinen grupos de cuatro nodos conectados entre sí.
- Parámetros:
 - x, y, z - parámetros de entrada: vectores de coordenadas que definen los puntos del contorno.
 - xC, yC, zC, r - parámetros variables: almacenan las coordenadas con el radio de curvatura y posiciones (x,y,z) del centro de la esfera asociada.
- Valor de retorno: contenido en los parámetros variables xC, yC, zC, r .

getCurv4

- Propósito: función que retorna la curvatura media para cada nodo del contorno y sus adyacentes, empleando el método **computeCurv4** para calcular la curvatura en cada 4-combinación de puntos.
- Parámetros: ninguno.
- Valor de retorno: arreglo unidimensional con los valores de curvatura calculados para cada punto del contorno.

getCurvVect

- Propósito: función que retorna la curvatura extrínseca para cada punto del contorno, mediante aproximación de derivadas.
- Parámetros: ninguno.
- Valor de retorno: arreglo unidimensional con los valores de curvatura calculados para cada punto del contorno.

getNpts

- Propósito: función que retorna el número de puntos del contorno.
- Parámetros: ninguno.
- Valor de retorno: número de puntos del contorno.

adjustContour

- Propósito: Aplicar el ajuste iterativo del contorno, en función de las fuerzas calculadas para cada punto, ponderadas por los parámetros respectivos.
- Parámetros: ninguno.
- Valor de retorno: arreglo de coordenadas $\{x_i, y_j, z_k\}$ del contorno ajustado.

getPressureForces

- Propósito: Calcular las fuerzas internas del contorno sobre cada nodo de éste.
- Parámetros: ninguno.
- Valor de retorno: arreglo de coordenadas $\{x_i, y_j, z_k\}$ del contorno ajustado.

calcGVF

- Propósito: Cálculo del campo vectorial de flujo de vectores de gradiente (GVF).
- Parámetros: ninguno.
- Valor de retorno: ninguno.

calcGGVF

- Propósito: Cálculo del campo vectorial de flujo de vectores de gradiente generalizado (GGVF).
- Parámetros: ninguno.
- Valor de retorno: ninguno.

edgemap

- Propósito: Método para calcular el mapa de aristas sobre la imagen (**pImage**).
- Parámetros: ninguno.
- Valor de retorno: ninguno.

gradient, image, direction

- Propósito: Función que retorna el gradiente de **image** en la dirección especificada por **direction**, para matrices de imagen tridimensionales.
- Parámetros
 - **image**: imagen de entrada, arreglo bidimensional con valores de intensidad.
 - **direction**: indicador de dirección para calcular la imagen gradiente, pudiendo ser 0 (**x**), 1 (**y**) o 2 (**z**); valor por defecto: 0.
- Valor de retorno: arreglo de igual dimensión y tamaño que **image**, conteniendo los valores del gradiente calculado para la dirección especificada.

laplacian, image

- Propósito: Función que retorna el laplaciano de una matriz (imagen) tridimensional.
- Parámetros
 - **image**: imagen de entrada, arreglo tridimensional con valores numéricos de intensidad.
- Valor de retorno: arreglo de igual dimensión y tamaño que **image**, resultante de aplicar sobre ésta el operador laplaciano.

setContour, x, y, z, polygons

- Propósito: Método para definir las coordenadas **x**, **y** y **z** del contorno.
- Parámetros:
 - **x, y, z**: vectores de coordenadas que definen los puntos del contorno.
 - **polygons**: lista de polígonos que asociada a los puntos del contorno.
- Valor de retorno: ninguno.

**setParams, alpha, beta, gamma, kappa, mu, iterations,
gvf_iterations**

- Propósito: Modificar valores de los atributos de la clase.
- Parámetros:
 - **alpha** (opcional): coeficiente de elasticidad.
 - **beta** (opcional): coeficiente de rigidez.
 - **gamma** (opcional): coeficiente de viscosidad.
 - **kappa** (opcional): factor de ponderación del campo GVF/GGVF.
 - **mu** (opcional): parámetro de regularización del detector de bordes.
 - **iterations** (opcional): número de iteraciones de ajuste del contorno.
 - **gvf_iterations** (opcional): número de iteraciones para cálculo del campo GVF/GGVF.
- Valor de retorno: ninguno.

getParam, paramName

- Propósito: Método para obtener el valor de un atributo de la clase.
- Parámetros:
 - **paramName**: nombre del atributo cuyo valor es requerido; puede indicar a cualquiera de los atributos susceptibles de modificación en el método **setParams**.
- Valor de retorno: valor del atributo indicado por **paramName**.

2.3 Clase C_sImageFilter_ActiveContours

2.3.1 Propósito

Filtro de segmentación por contornos activos en dos dimensiones.

2.3.2 Herencia

Clase genérica para filtros sobre ROIs bidimensionales: **C_sImageFilter**.

2.3.3 Atributos

pParamStruct

- Tipo: puntero.
- Propósito: puntero a las estructuras de parámetros que caracterizan a cada ROI.
- Valor por defecto: puntero a arreglo con la estructura de parámetros.

pValueStruct

- Tipo: puntero.
- Propósito: puntero a matrices con valores de parámetros calculados en cada ROI.
- Valor por defecto: puntero a arreglos con los valores de cada parámetro calculado:
 - Número de puntos de inflexión
 - Curvatura
 - Perímetro
 - Area

pParamApplied

- Tipo: puntero.
- Propósito: puntero que almacena los valores de parámetros de la última aplicación del filtro, con el fin de evitar repetir el procesamiento de modo redundante.
- Valor por defecto: puntero a un vector de valores **float**.

pImage

- Tipo: puntero.

- Propósito: puntero a la imagen de máscara (binaria) para la cual el filtro es aplicado. A partir de esta imagen se obtienen los contornos iniciales para las ROIs.
- Valor por defecto: nulo.

pIntImage

- Tipo: puntero.
- Propósito: puntero a la imagen original (intensidades) para la cual el filtro es aplicado. A partir de esta imagen se calculan los campos de fuerzas externas para el ajuste de los contornos activos.
- Valor por defecto: nulo.

pSegImage

- Tipo: puntero.
- Propósito: puntero a la imagen segmentada por contornos activos.
- Valor por defecto: nulo.

2.3.4 Métodos

init

- Propósito: constructor de la clase. Heredado de **C_sROIParam**.
- Valor de retorno:
 - 1, si el método se ejecutó correctamente y se creó una instancia de la clase;
 - -1, en caso de encontrar un error en los parámetros de entrada;
 - 0, en caso de producirse una excepción en tiempo de ejecución.

cleanup

- Propósito: destructor de la clase. Heredado de **C_sROIParam**.
- Parámetros: ninguno.
- Valor de retorno: ninguno.

`apply, image, selectedStackObject, timePosition,`
`channelPosition, zSlicePosition, clusterPosition,`
`segPosition, cut_x, cut_y, stack_tlb`

- Propósito: calcular los parámetros característicos sobre un grupo de ROIs, creando una instancia de la clase `C_sActiveContour` para obtener de ella los contornos ajustados y generar una máscara binaria de segmentación.
- Parámetros:
 - `image`: imagen de máscara sobre la que se aplica la segmentación.
 - `selectedStackObject`: puntero al *stack* de imágenes sin segmentar.
 - `timePosition`: índice de tiempo para identificar a la imagen dentro del *stack*.
 - `channelPosition`: índice de canal para identificar a la imagen dentro del *stack*.
 - `zSlicePosition`: índice de ubicación en el eje z para identificar a la imagen dentro del *stack*.
 - `clusterPosition`: no aplicable; empleado por otros filtros de la misma categoría.
 - `segPosition`: no aplicable; empleado por otros filtros de la misma categoría.
 - `cut_x, cut_y`: especifican las coordenadas de un rectángulo de recorte sobre la imagen, para aplicar la segmentación sobre éste, o bien pueden corresponder a las dimensiones para la imagen completa.
 - `stack_tlb`: puntero de referencia a componente de interfaz gráfica (*widget*), que permite especificar la ubicación de la imagen asociada a las ROIs dentro del *stack*.
- Valor de retorno: imagen segmentada por contornos activos.

getImageFilterType

- Propósito: función que retorna el tipo de filtro, que lo identifica en una categoría específica, para determinar los parámetros que habrán de especificarse en su aplicación sobre una máscara.
- Parámetros: ninguno.
- Valor de retorno: 'Multiple_Image_Filter_Method'.

checkParamsApplied

- Propósito: función que compara los parámetros actuales del filtro con los especificados durante la última aplicación de éste. Se emplea por el método **apply** para determinar si es necesario recalcular los contornos y campos de fuerzas para el ajuste.
- Parámetros: ninguno.
- Valor de retorno:
 - 1, si se ha producido una modificación en los parámetros desde la última aplicación del filtro,
 - 0 en caso contrario.

2.4 Clase C_sROIParam_ActiveContour

2.4.1 Propósito

Caracterización de ROIs bidimensionales a través de los contornos activos.

2.4.2 Herencia

Clase genérica para parámetros de ROIs bidimensionales: **C_sROIParam**.

2.4.3 Atributos

pParamStruct

- Tipo: puntero.
- Propósito: puntero a las estructuras de parámetros que caracterizan a cada ROI.
- Valor por defecto: puntero a arreglo de la forma

pValueStruct

- Tipo: puntero.
- Propósito: puntero a las matrices de valores para los parámetros calculados en cada ROI.
- Valor por defecto: puntero a arreglos de la forma

2.4.4 Métodos

init

- Propósito: constructor de la clase. Heredado de **C_sROIParam**.
- Valor de retorno:
 - 1, si el método se ejecutó correctamente y se creó una instancia de la clase;
 - -1, en caso de encontrar un error en los parámetros de entrada;
 - 0, en caso de producirse una excepción en tiempo de ejecución.

cleanup

- Propósito: destructor de la clase. Heredado de **C_sROIParam**.
- Parámetros: ninguno.
- Valor de retorno: ninguno.

apply

- Propósito: calcular los parámetros característicos sobre un grupo de ROIs.
- Parámetros:
 - **C_sROIGroupObject**: clase contenedora de ROIs bidimensionales.
 - **stack_tlb**: puntero de referencia a componente de interfaz gráfica (widget), que permite especificar la ubicación de la imagen asociada a las ROIs dentro del *stack*.
- Valor de retorno: ninguno.

2.5 Diagramas de Clases Implementadas

C_sActiveContour
alpha: float ≥ 0 (*) beta: float ≥ 0 (*) gamma: float ≥ 0 kappa: float ≥ 0 mu: float > 0 iterations: unsigned integer ≥ 0 gvf_iterations: unsigned integer > 0 pImage: 2D array pointer npts: unsigned integer pX: (float double) array pointer pY: (float double) array pointer pU: 2D (float double) array pointer pV: 2D (float double) array pointer
setParams, alpha, beta, gamma, kappa, mu, iterations, gvf_iterations getParam, paramName: float getXcoords: (float double) array getYcoords: (float double) array getCoords: (float double) array getCir3, xyRes: (float double) array getCurvD, xyRes: (float double) array getDistance, xyRes: (float double) array getGradients, xyRes: (float double) array getInflexPoints, curvArray: array getPerimeter, xyRes: float arcSample, points adjustContour: {x, y: (float double) array} calcGVF calcGGVF edgeMap gradient: 2D (float double) array laplacian: 2D (float double) array defaultContour setContourCoords, xCoords, yCoords cleanup init: integer

C_sActiveContour3D
alpha: float ≥ 0 beta: float ≥ 0 (alpha $\neq 0$ beta $\neq 0$) gamma: float ≥ 0 kappa: float ≥ 0 mu: float > 0 iterations: unsigned integer ≥ 0 gvf_iterations: unsigned integer > 0 pImage: 2D byte array pointer npts: unsigned long integer pX: (float double) array pointer pY: (float double) array pointer pZ: (float double) array pointer pPolygons: unsigned long integer array pointer pNeighbors: unsigned long integer array pointer pCurv: (float double) array pointer pU: 3D (float double) array pointer pV: 3D (float double) array pointer pW: 3D (float double) array pointer
setParams, alpha, beta, gamma, kappa, mu, iterations, gvf_iterations getParam, paramName: float getXcoords: (float double) array getYcoords: (float double) array getZcoords: (float double) array getCoords: (float double) array getPolygonList: unsigned long integer array getCurvVect: (float double) array computeCurv4, x, y, z, xC, yC, zC, R getCurv4, (float double) array getInflexPoints, curvArray: unsigned long integer array getNpts: unsigned long integer adjustContour: {x, y, z: (float double) array } getAvgNeighborNodes {x1, x2, y1, y2, z1, z2: (float double) array } calcGVF calcGGVF edgeMap gradient: 3D (float double) array laplacian: 3D (float double) array defaultContour setContourCoords, xCoords, yCoords, zCoords cleanup init: integer

2.6 Métodos Implementados en Otras Clases

2.6.1 Clase C_sROI3DGroupObject

- Propósito de la clase: contenedor para los modelos de ROIs en tres dimensiones.
- Nombre del método: `getoSurfaceModel`.
- Propósito del método: generar modelos de superficie poligonal de las ROIs, a partir de las máscaras binarias de segmentación (imágenes binarias tridimensionales).
- Parámetros:
 - `oObjectModel` (parámetro de valor): objeto contenedor de los modelos de ROIs para su visualización
- Valor de retorno: modelos de superficie agregados en `oObjectModel`.

```
pro C_sROI3DGroupObject::getoSurfaceModel, oObjectModel
```

```
self->IDLgrROIGroup::getProperty,   xCoord_conv = xCoord_conv,  
                                     yCoord_conv = yCoord_conv,  
                                     zCoord_conv = zCoord_conv
```

```
rgb_table0 = bytArr(256,3)+ transpose((*self.pVolState).rgbValues[0,*,*])  
opacVect = self->getOpacVect('1st Volume Opacity')
```

```
for i = 0, self->count()-1 do begin
```

```
  obj = (self->get(position = i))->makePixelObjectInVoxel(/all)  
  shade_Volume, obj.obj, 0, vertices, polygons, /low
```

```
  vertices[0,*] += (obj.minX - obj.pixelFrame)  
  vertices[1,*] += (obj.minY - obj.pixelFrame)  
  vertices[2,*] += (obj.minZ - obj.pixelFrame)
```

```
  if (i eq 0) $
```

```
  then color = [rgb_table0[round(255./(self->count()-1)),0], $
```

```

        rgb_table0[round(255./(self->count()-1)),1],$
        rgb_table0[round(255./(self->count()-1)),2]]$
else color = [rgb_table0[round(255./(self->count()-1)*i),0],$
             rgb_table0[round(255./(self->count()-1)*i),1],$
             rgb_table0[round(255./(self->count()-1)*i),2]]

opacVal = opacVect[round(255./(self->count()*(i+1)))]
oObjectModel->add, obj_new('IDLgrPolygon', $
                        alpha_channel = opacVal/255., $
                        ambient = [0,0,0], $
                        data = vertices, poly = polygons, $
                        bottom = color, color = color, shading = 1, $
                        xCoord_conv = xCoord_conv, $
                        yCoord_conv = yCoord_conv, $
                        zCoord_conv = zCoord_conv, $
                        name = strCompress('3DSurfaceModel:'+string(i),/rem))
endfor
end

```

- Nombre del método: **getoACModel**.
- Propósito del método: generar modelos de superficie poligonal de las ROIs, a partir de las máscaras binarias de segmentación (imágenes binarias tridimensionales), y ajustados mediante contornos activos en tres dimensiones.
- Parámetros:
 - **oObjectModel** (parámetro de valor): objeto contenedor de los modelos de ROIs para su visualización
- Valor de retorno: modelos de superficie agregados en **oObjectModel**.

```

pro C_sROI3DGroupObject::getoACModel, oObjectModel, stack_tlb = stack_tlb

self->IDLgrROIGroup::getProperty,   xCoord_conv = xCoord_conv,
                                     yCoord_conv = yCoord_conv,
                                     zCoord_conv = zCoord_conv

rgb_table0 = bytArr(256,3)+ transpose((*self.pvolState).rgbValues[0,*,*])
opacVect = self->getOpacVect('1st Volume Opacity')

    ; define initial surfaces
nObjects = self->count()
pVertices = ptrArr(nObjects)
pPolygons = ptrArr(nObjects)
for i = 0, nObjects-1 do begin
    obj = (self->get(position = i))->makePixelObjectInVoxel(/all)
    shade_Volume, obj.obj, 0, vertices, polygons, /low
    vertices[0,*] += (obj.minX - obj.pixelFrame)
    vertices[1,*] += (obj.minY - obj.pixelFrame)
    vertices[2,*] += (obj.minZ - obj.pixelFrame)
    pVertices[i] = ptr_new(vertices, /no_copy)
    pPolygons[i] = ptr_new(polygons, /no_copy)
endfor

iterations = 5

oActiveContour3D = obj_new('C_sActiveContour3D', $
                           make_array(3, 3, 3, /float), $;dummy test image
                           iterations = iterations, $
                           alpha = 1., beta = 1.)

for i = 0, nObjects-1 do begin
    oActiveContour3D->setContour, transpose((*pVertices[i])[0,*]), $
                                transpose((*pVertices[i])[1,*]), $
                                transpose((*pVertices[i])[2,*]), $
                                *(pPolygons[i])

    if iterations eq 0 then res = oActiveContour3D->getAvgNeighborNodes()

```

```

if oActiveContour3D->adjustContour() then begin
    vertices = oActiveContour3D->getCoords()
    polygons = oActiveContour3D->getPolygonList()
    colorVect = oActiveContour3D->getCurv4()
    maxCV = max(colorVect, min = minCV)
    if (maxCV gt minCV) then begin
        colorVect -= minCV
        colorVect /= ((maxCV - minCV) / 255.)
    endif
    colorVect = floor(colorVect)
    if (i eq 0) then $
        color = [rgb_table0[round(255./(self->count()-1)),0],$
                rgb_table0[round(255./(self->count()-1)),1],$
                rgb_table0[round(255./(self->count()-1)),2]]$
    else color = [rgb_table0[round(255./(self->count()-1)*i),0],$
                rgb_table0[round(255./(self->count()-1)*i),1],$
                rgb_table0[round(255./(self->count()-1)*i),2]]
    opacVal = 255.
    oObjectModel->add, obj_new('IDLgrPolygon',
                                alpha_channel = opacVal/255.,
                                ambient = [0,0,0],
                                data = vertices, poly = polygons,$
                                shading = 1,$
                                xCoord_conv = xCoord_conv,$
                                yCoord_conv = yCoord_conv,$
                                zCoord_conv = zCoord_conv,$
                                name = strCompress('3DSurfaceModel:'+string(i),/rem),$
                                vert_colors = transpose(rgb_table0[colorVect]))
    endif
endfor

obj_destroy, oActiveContour3D

```

```

for i = nObjects-1, 0, -1 do begin
    ptr_free, pVertices[i]
    ptr_free, pPolygons[i]
endfor

ptr_free, pVertices
ptr_free, pPolygons
end

```

2.6.2 Clase C_sROI3DObject

- Propósito de la clase: modelo de ROI en tres dimensiones.
- Nombre del método: **getObjectSurfaceFromPolygons**.
- Propósito del método: calcular el área de superficie para la ROI a partir de los polígonos que la conforman.
- Parámetros:
 - **polygons**: arreglo con la lista de polígonos.
 - **vertices**: arreglo con las coordenadas de los vértices de la ROI.
- Valor de retorno: área de superficie calculada para la ROI.

```

function C_sROI3DObject::getObjectSurfaceFromPolygons, polygons =
polygons, vertices = vertices

if (n_elements(realSize) gt 0) then begin
    vertices[0,*] *= ((*self.pEigenSys).sizePerXYZ)[0]
    vertices[1,*] *= ((*self.pEigenSys).sizePerXYZ)[1]
    vertices[2,*] *= ((*self.pEigenSys).sizePerXYZ)[2]
endif

pos = 0l
triPos = [0]

```

```

repeat begin
  case polygons[pos] of
    3: begin
      triPos = [triPos, polygons[pos+1:pos+3]]
      pos+=4
    endcase
    4: begin
      triPos = [triPos, polygons[[pos+1, pos+3, pos+2]],$
                polygons[[pos+1, pos+4, pos+3]]]
      pos+=5
    endcase
  else: begin
    print, 'Polygon of higher order than 4'
    pos += polygons[pos]
  endcase
endcase
endRep until (pos ge n_elements(polygons))

triPos = triPos[1:*]
triTri = make_array(n_elements(triPos)/3, /index) * 3

a = sqrt((vertices[0,triPos[triTri+1]]-$
          vertices[0,triPos[triTri]])^2+$
          (vertices[1,triPos[triTri+1]]-$
          vertices[1,triPos[triTri]])^2+$
          (vertices[2,triPos[triTri+1]]-$
          vertices[2,triPos[triTri]])^2)
b = sqrt((vertices[0,triPos[triTri+2]]-$
          vertices[0,triPos[triTri+1]])^2+$
          (vertices[1,triPos[triTri+2]]-$
          vertices[1,triPos[triTri+1]])^2+$
          (vertices[2,triPos[triTri+2]]-$
          vertices[2,triPos[triTri+1]])^2)

```

```
c = sqrt((vertices[0,triPos[triTri+2]]-$
          vertices[0,triPos[triTri]])^2+$
          (vertices[1,triPos[triTri+2]]-$
          vertices[1,triPos[triTri]])^2+$
          (vertices[2,triPos[triTri+2]]-$
          vertices[2,triPos[triTri]])^2)
```

```
s = (a+b+c)/2.
```

```
a = sqrt(s*(s-a)*(s-b)*(s-c))
```

```
return, total(a)
```

```
end
```

Anexo 3: Color y Opacidad en Imágenes Digitales

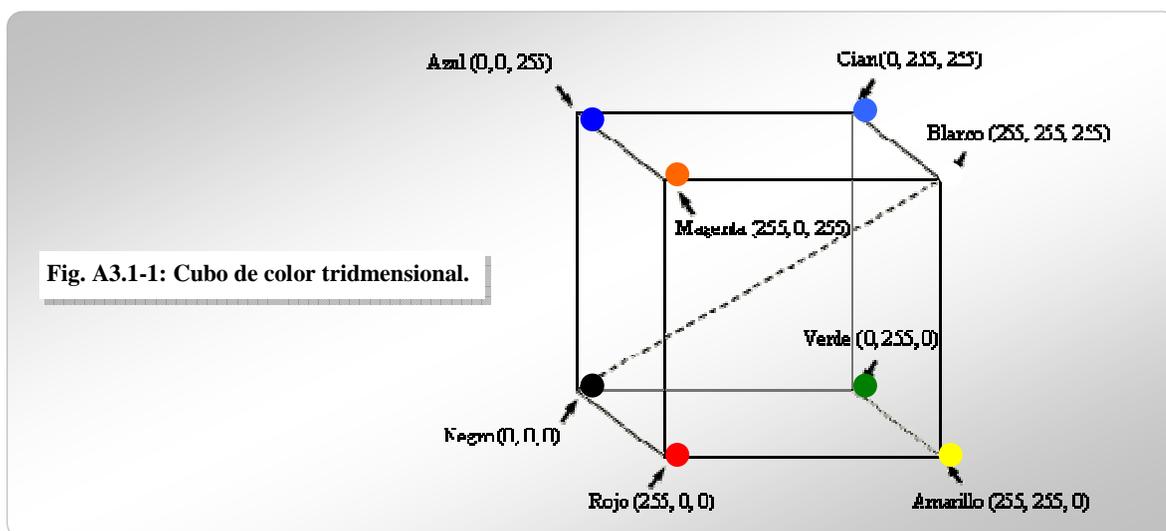
Este anexo resume los conceptos relativos a los modelos y tablas de color digital, aplicados en IDL a las imágenes de microscopia confocal.

3.1 Sistemas Digitales de Color

Para su representación digital, el color puede ser codificado usando distintos esquemas [Gon02], utilizando en la mayoría de los casos 3 valores para representar un color como una posición dentro de un espacio tridimensional. IDL proporciona las mesas de color RGB (rojo-verde-azul: *Red-Green-Blue*), HSV (*Hue-Saturation-Value*), HLS (*Hue-Lightness-Saturation*), y CMY (cian-magenta-amarillo: *Cyan-Magenta-Yellow*).

Adicionalmente, la **profundidad de color** está dada por la cantidad de bits que codifican un determinado color. Una imagen digital de n bits de profundidad es capaz de desplegar 2^n colores en total; así, $n = 8$ bits permiten desplegar $2^8 = 256$ colores (*PseudoColor*), y con $n = 24$ se tienen $2^{24} = 16.777.216$ colores (*TrueColor*). IDL puede manejar 8, 16 o 24 bits.

En IDL, el espacio de color RGB es representado como un sistema de coordenadas rectangulares tridimensionales, cuyos ejes corresponden a los colores primarios rojo, verde y azul respectivamente, y con rango de valores [0, 255]. La combinación de distintos niveles de cada color primario determina un color individual, codificado como una coordenada (x, y, z) . El origen $(0, 0, 0)$ corresponde al negro, y el extremo $(255, 255, 255)$ corresponde al blanco, representando una mezcla aditiva del total de las intensidades de los 3 colores primarios. Los puntos en la diagonal (con igual valor de intensidad para cada color primario) representan la escala de grises. Así, el color amarillo se representa por la coordenada $(255, 255, 0)$, o la mezcla del rojo al 100% más verde al 100%, etc. La figura A3.1-1 muestra las combinaciones posibles de colores, como posiciones dentro de un cubo de color tridimensional para 256 colores (8 bits).



3.2 Modelos de Color en Objetos Gráficos de IDL

Los objetos para despliegue gráfico en IDL permiten aplicar los siguientes modelos de color:

- Indexado: se especifica usando un índice dentro de una **tabla de búsqueda** (*Look-up Table*, LUT) o paleta de color, con 256 colores distintos (Fig. A3.4-1). Cada entrada de la LUT corresponde a un color individual que consiste de una tripleta de valores RGB, que se asocia a cada nivel de intensidad en el objeto.
- RGB: se especifica usando una tríada de valores enteros de 8 o 12 bits para las componentes RGB.

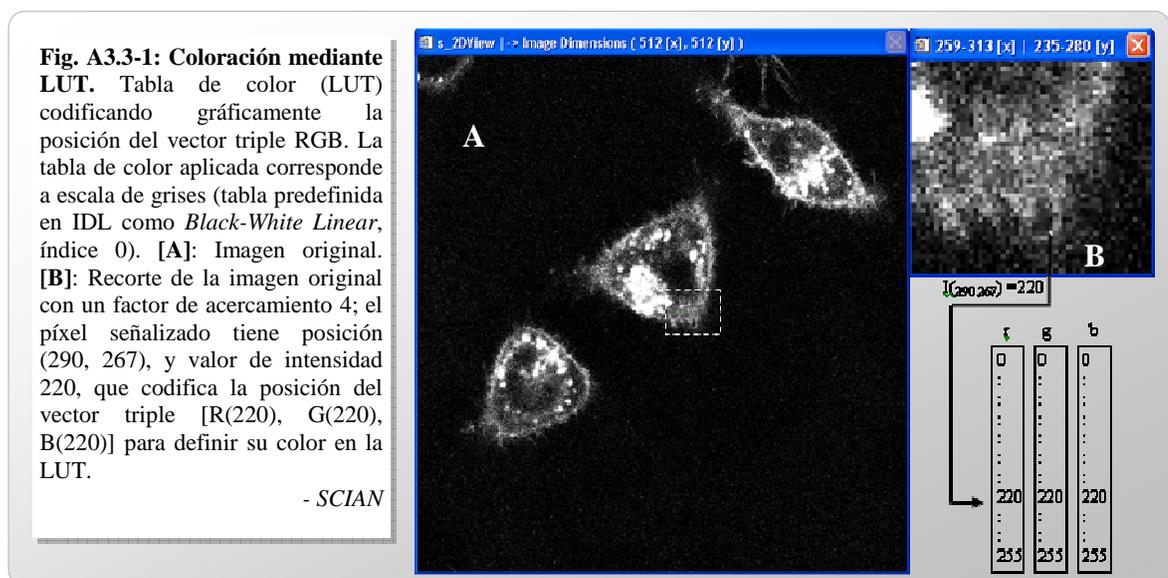
Para asociar el color a una imagen en IDL, se utilizan instancias de la clase **IDLgrImage**, que almacenan la información de cada imagen con un tipo de dato entero sin signo (*byte*), en alguna de las siguientes formas:

- Un arreglo de dos dimensiones [x, y]. El color para cada píxel se tiene a partir de las entradas dentro de una LUT especificada, o como un valor en escala de grises (3.4).
- Un arreglo con dimensiones [2, x, y]. Para cada píxel se almacena de un valor de intensidad en escala de grises [I, x, y], más un valor de canal alfa [α , x, y] asociado. α es un valor equivalente al de un canal RGB, $\alpha \in [0, 255]$, que codifica la opacidad o transparencia de cada píxel (3.5).

- Un arreglo con dimensiones [3, x, y]. A cada píxel se asocia una tripleta de intensidades RGB: [R, x, y], [G, x, y], [B, x, y].
- Un arreglo con dimensiones [4, x, y]. Cada píxel consiste de una tríada RGB y un valor de canal α ¹⁷ asociado ([R, x, y], [G, x, y], [B, x, y], [α , x, y]).

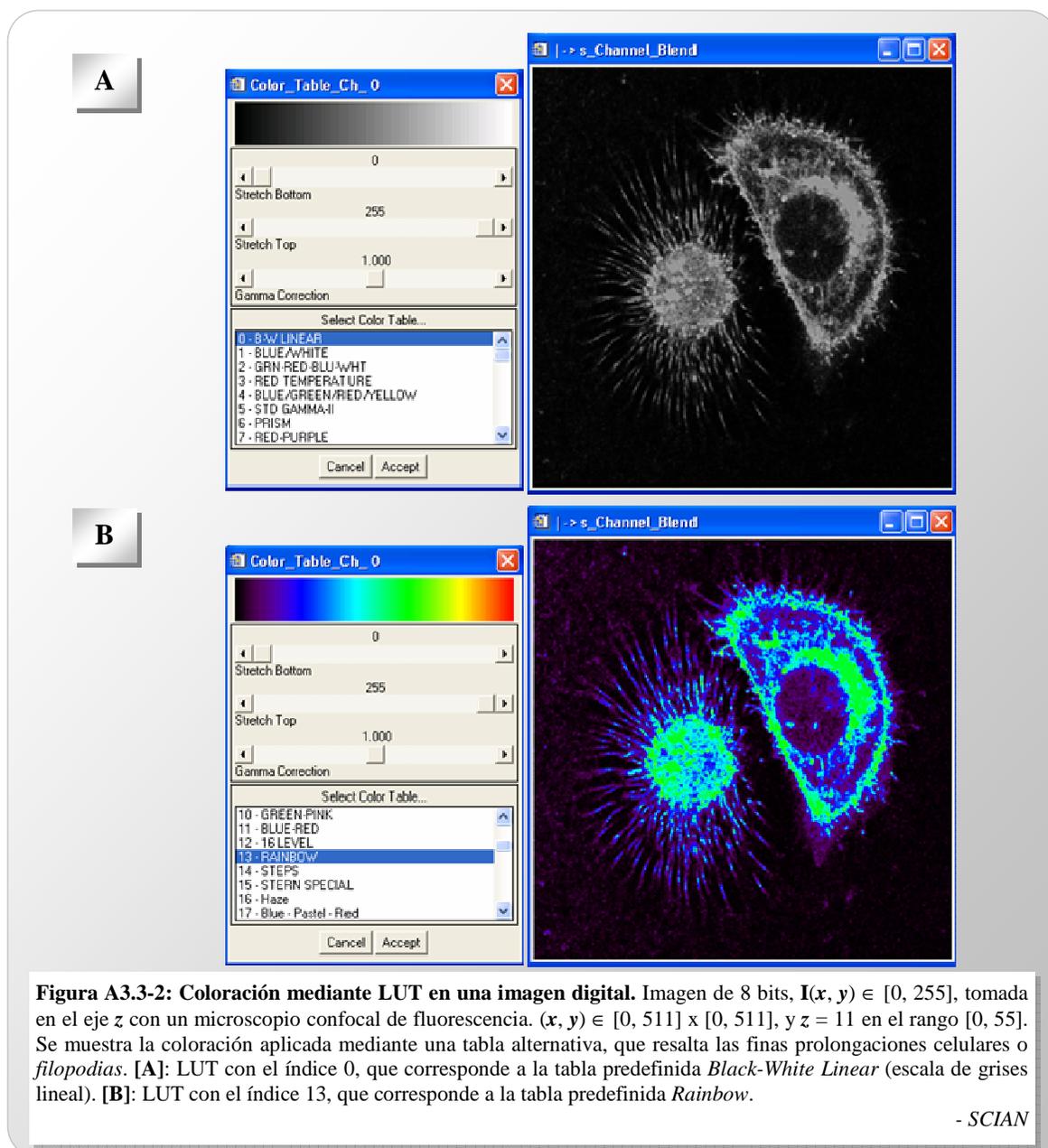
3.3 LUT

El uso de tablas de color, o LUTs, permite resaltar características dentro de una imagen que no se observan a simple vista con la escala de grises. IDL proporciona distintas tablas, a las que se accede especificando su número de índice; este índice es un valor entero sin signo (*byte*) entre cero y el número total de tablas definidas menos uno¹⁸. La intensidad cuyos valores definen una LUT es codificada en las posiciones de un vector triple RGB (figura A3.3-1); de esta forma se puede asignar color a imágenes con un solo canal de color o intensidades (escala de grises). La figura A3.3-2 muestra una tabla de color aplicada para realzar detalles en imágenes de células en cultivo.



¹⁷ opacidad, ver sección 3.5

¹⁸ Existen 41 LUT predefinidas por IDL, además de 12 tablas personalizadas elaboradas para SCIAN [Ver05].

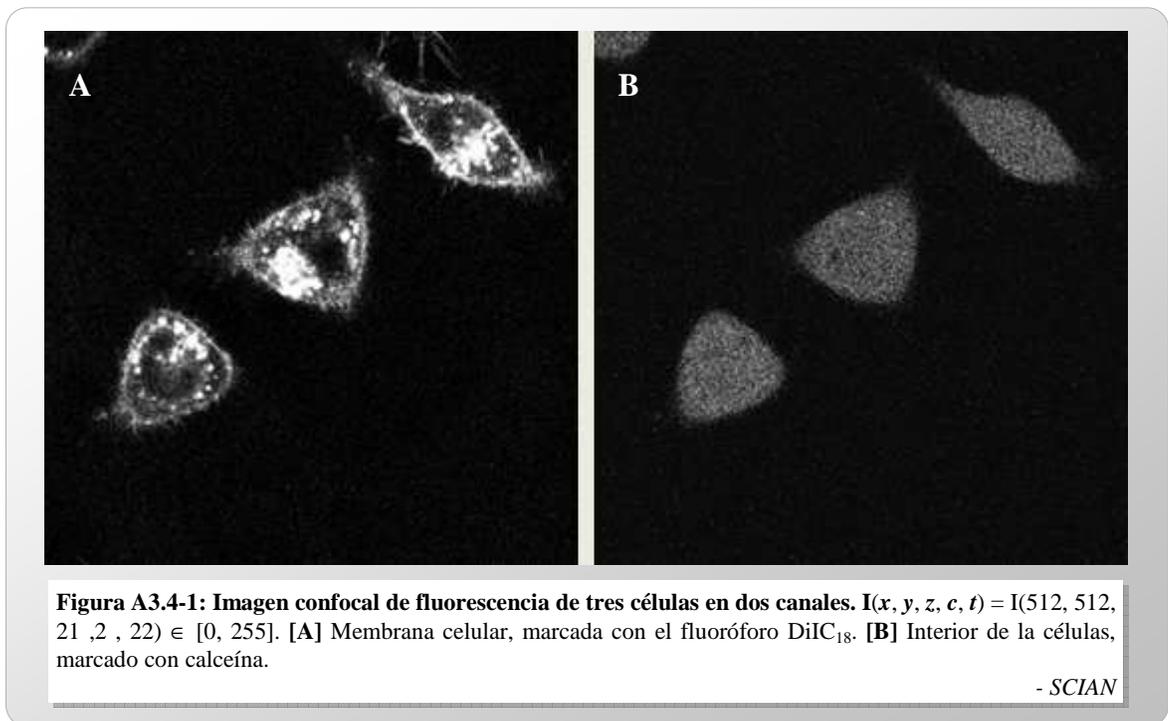


3.4 Canales de Color en Imágenes de Microscopía Confocal

Si los canales de color almacenan la información de los colores de la imagen según las talas de color, la cantidad de canales definidos para una imagen dependerá del **modo de color**. Por ejemplo, una imagen RGB tiene 3 canales (rojo, verde y azul). Las imágenes en escala de grises, mapas de bits y de colores indexados tienen un solo canal.

En la microscopía confocal una muestra celular se marca con fluoróforos, colorantes fluorescentes que se unen a estructuras específicas. El término **canal** en microscopía se utiliza para distinguir en una misma muestra biológica a las distintas longitudes de onda

que excitan y emiten los diferentes fluoróforos. La imagen de la muestra es capturada en un canal de fluorescencia con valores de intensidad $I(x, y) \in [0, 255]$, asociados a la cantidad de fluoróforo detectada en la banda de frecuencia correspondiente (Fig. A3.4-1). Al asignar colores a cada canal de fluorescencia se consigue distinguir uno de otro, formando tantas imágenes como canales.



Una forma de asignar color a un canal de una imagen es asociándole una LUT, con lo que es posible es posible realizar una superposición de los colores de cada canal para visualizar características de ambos (Fig. A3.4-2).

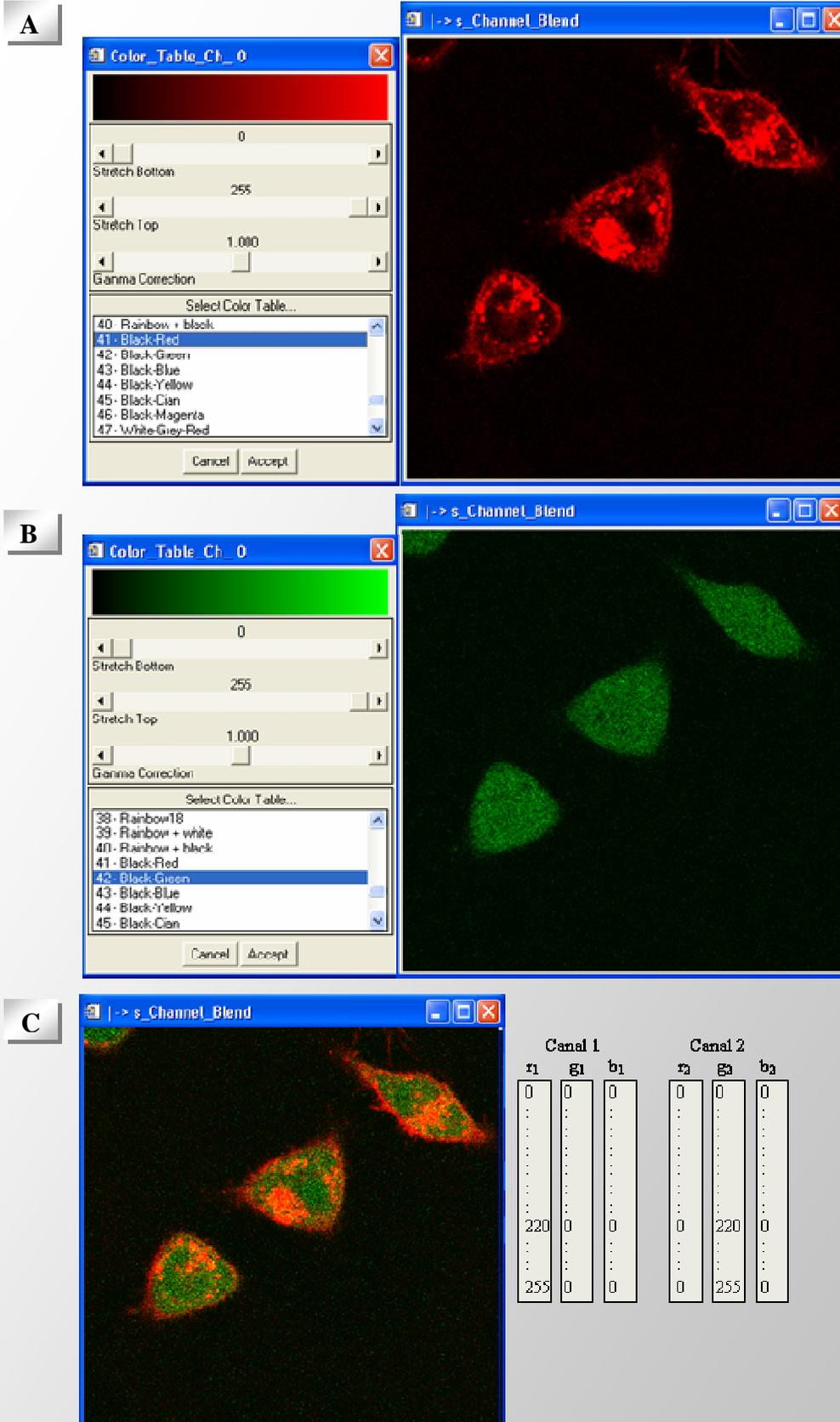
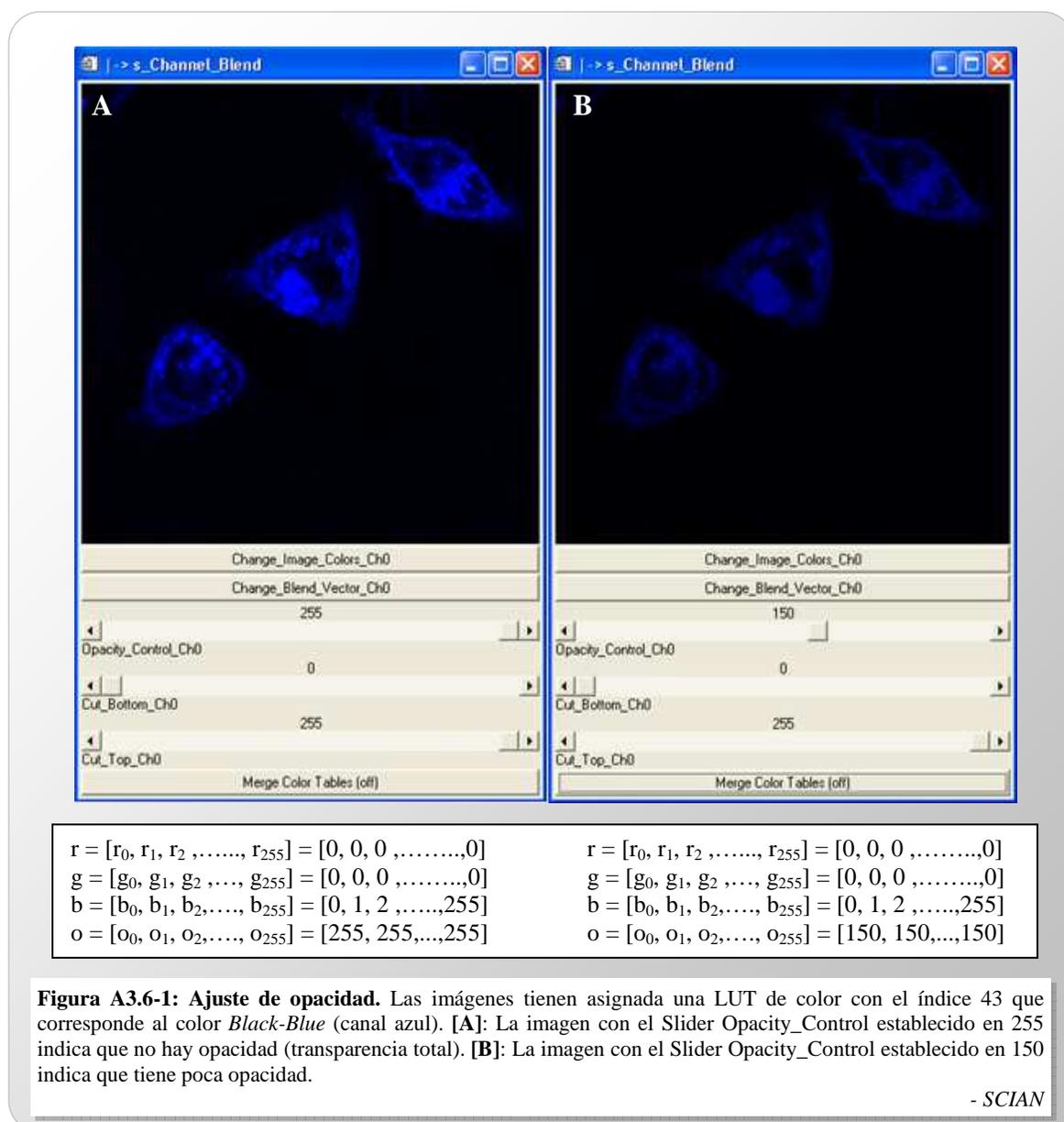


Figura A3.4-2: Representación de imágenes que describen 2 canales de fluorescencia a través de una LUT. [A]: Imagen confocal de canal rojo, luego de aplicar una LUT con el índice 41 (*Black-Red*). [B]: Aplicación de una LUT con el índice 42 (*Black-Green*) a la imagen confocal del canal verde. [C]: Superposición entre un canal rojo y un canal verde representando gráficamente el vector triple RGB. Cada canal está descrito por una LUT de color (Fig. 3.6). La superposición se produce por la suma de las intensidades de los vectores RGB de ambos canales y debe cumplirse que $[r_1 + r_2, g_1 + g_2, b_1 + b_2] < 255$.

3.5 Opacidad o Canal α

La **opacidad** es una propiedad de transparencia que se asocia a cada píxel en una imagen, y se codifica en el denominado canal alfa (α), que actúa de forma independiente al color establecido. La opacidad toma valores enteros en un rango de $[0, 2^n-1]$; así, si el valor de α es máximo de la escala (para 8 bits = $2^8-1 = 255$), no hay transparencia, y $\alpha = 0$ determina opacidad mínima o transparencia total. La figura A3.6-1 ilustra el efecto de la aplicación del canal α en una imagen.



Anexo 4: Scripting para Deconvolución de Imágenes

Los métodos de segmentación abordados en el presente Proyecto se pueden clasificar, dentro del área de procesamiento de imágenes, en la etapa llamada **análisis de imágenes** [You95]. En esta etapa se busca obtener información cuantitativa sobre características de interés, desde un conjunto de imágenes de entrada. Previo al **análisis** se encuentra el **tratamiento de imágenes**, cuya finalidad es la de mejorar la calidad de las imágenes de entrada.

Aplicado a las imágenes de microscopia confocal, el tratamiento de imágenes se aboca a eliminar de éstas los efectos de distorsión introducidos por el sistema físico de observación (óptica del microscopio confocal) [Wil03], contribuyendo a que el proceso de segmentación (análisis de imágenes) se efectúe sobre datos de entrada más fieles a la realidad de las imágenes capturadas.

La **deconvolución** es una operación matemática de transformación, inversa a la inducida por el sistema físico de captura, que se efectúa sobre las imágenes de la microscopia confocal de fluorescencia. La distorsión inducida por los microscopios confocales ha sido estudiada y modelada matemáticamente, en función de los ajustes o parámetros específicos de cada tipo de microscopio, tales como la longitud de onda de excitación y emisión de los fluoróforos, las dimensiones de la muestra, tipo de lente y apertura empleados, y el medio de inmersión de la muestra [Kem99, Wil03]. De este modo es posible aplicar procedimientos de deconvolución a un conjunto de imágenes dado, si se conoce la configuración del sistema con que se adquirieron.

A continuación se presenta el código fuente de uno de los *scripts*, desarrollado en el lenguaje TCL, para deconvolución de *stacks* de imágenes en un servidor de red mediante procesamiento por lotes (*batch processing*). El *script* se ejecuta en un software específico para realizar deconvolución en imágenes de microscopia¹⁹.

¹⁹ Huygens Scripting, <http://www.svi.nl>

```
# Script name: deconvolution.tcl

# Purpose: Huygens deconvolution script

## Info Section

# About this script

#

# This script will allow to you to perform a deconvolution process using
# the default configuration for your input image and microscope settings,
# using the Huygens software (www.svi.nl).

# Running this script

#

# You must have the following Huygens software (Linux only):
#   Huygens Professional / Huygens / Huygens Scripting
# Locate the input image in your local file system (path)
# Locate the path for this script
# In the Linux shell, change your current path to the script path
# Execute the command "huscript" (without the quotes)
# The Huygens Scripting program will load and display some information in
# the shell, after which a prompt will appear. Use this prompt to enter
# the command for running this script in Huygens Scripting
# (the command prompt is a '%' character).
# When the prompt appears, enter the command "source deconvolution.tcl"
# Now the scripting will be running and you must take a seat while the
# deconvolution process is performed.

## End of Info section
```

```

## Input Parameters Section

# +
# The purpose of this section is to provide the Huygens software with all
# the information required to perform a deconvolution process, according
# to the input parameters specified here.
# Specify the parameters according to your original data, by setting the
#input variables defined here.

# -
# Input image filename.
# You must specify the complete filename of the image, including its
# extension (*.lsm, *.ics, etc.)
set inputImageDir {/home/shaertel/Images}
set inputImageName sampleVol.lsm

# Output (deconvolved) image filename.
set outputImageDir {/home/shaertel/Images/sampleVol_deconvolved}
set outputImageName regVol2deconvolved

# Output image format
# Available formats: ics | ics2 | imaris | tiff | tiff16 | tiffrgb | ppm
set outputImageFormat tiffrgb

# Maximum number of iterations for the deconvolution process
set deconvolutionIterations 140

# - - - Microscope settings - - -

set numericalAperture 1.3
set lensRefractiveIndex 1.515
set mediumRefractiveIndex 1.44

```

```

# a scalar or a vector like {500.0 520.0}
set excitationWaveLength 520.0

set emissionWaveLength 520.0
# the same as previous {500.0 520.0}

set photonCount 1

# sample distances in micron/sec in x, y, z, t
set sampleDistances {50 50 50 1}

set pinholeRadius 250.0
set pinholeSpacing 2.53

# Microscope type
# Available types confocal | widefield | nipkow | 4Pi | generic
set microscopeType confocal

# - - - CMLE settings - - -

# Signal noise ratio
set signalToNoiseRatios (20 20 20 20)

# Background mode
# Available options: auto | manual | lowest | object | wf
set backgroundMode auto

# Estimated backgrounds
set estimatedBackgrounds (0 0 0 0)

# Bleaching correction
# Available options: auto | off
set bleaching off

```

```
# Stopping criterion
set qualityChangeThreshold 0.1

# Iteration mode
# Available options: fast | highQ
set iterationMode fast

# Padding mode
# Available options: auto | parent | padpar | fullpadpar
set paddingMode auto

## End of Input Parameters Section
```

```

## Execute Section

# +

# This section performs all the program calls for the deconvolution
# process, according to the parameters specified in the previous section.
# You should not have to modify any part of this section.

# -

# Get the input image variable name from the input image filename
set inputImagefNameList [split $inputImagefName .]
set inputImage [lindex $inputImagefNameList 0]

# Open the input image
cd $inputImageDir
puts "<script>: Loading image: $inputImageDir/$inputImagefName"
img open $inputImageName
puts "<script>: Image loaded as $inputImage"

# Image settings
# Enable this lines to apply custom settings to the image

#$inputImage setp -na $numericalAperture
#$inputImage setp -ril $lensRefractiveIndex
#$inputImage setp -i $mediumRefractiveIndex
#$inputImage setp -ex $excitationWaveLength
#$inputImage setp -em $emissionWaveLength
#$inputImage setp -pcnt $photonCount
#$inputImage setp -s $sampleDistances
#$inputImage setp -pr $pinholeRadius
#$inputImage setp -ps $pinholeSpacing
#$inputImage setp -micro $microscopeType

# Check the sizes
set maxSize 200000

```

```

set xSize [$inputImage getdims -mode x]
set ySize [$inputImage getdims -mode y]
set zSize [$inputImage getdims -mode z]
set tSize [$inputImage getdims -mode t]
set imgSize [expr ($xSize * $ySize * $zSize * $tSize)]

# PSF computation
puts "computing PSF"
$inputImage genpsf -> psf

# Deconvolution process
if {$imgSize > $maxSize} then {
    puts "<script>: processing the time series frame by frame"
    for {set count 0} {$count < $tSize} {incr count} {
        $inputImage getframe -> a -frame $count
        puts "<script>: processing frame $count"
        b del
        img create b
        a cmle psf -> b -blMode off
        # a cmle psf -> outputImage -it $deconvolutionIterations -sn
        $signalToNoiseRatios -bgMode $backgroundMode -bg $estimatedBackgrounds
        -blMode $bleaching -q $qualityChangeThreshold -q $iterationMode -pad
        $paddingMode

        cd $outputImageDir
        if {$count < 100} then {
            set sframe _t0$count
        } else {
            set sframe _t$count
        }
        if {$count < 10} then {
            set sframe _t00$count
        }
    }
}

```

```

        set outputImageFname $outputImageName$sframe
        puts "<script>: saving deconvolved frame to $outputImageFname
with $outputImageFormat format"
        cd $outputImageDir
        b save $outputImageFname\_ -type $outputImageFormat
        puts "<script>: frame $count saved"
    }
} else {
    puts "\n<script>: Starting the deconvolution process with the CMLE
method"

    $inputImage cmle psf -> b -blMode off -it $deconvolutionIterations -
sn $signalNoiseRatio

    # $inputImage cmle psf -> b -it $deconvolutionIterations -sn
$signalToNoiseRatios -bgMode $backgroundMode -bg $estimatedBackgrounds
-blMode $bleaching -q $qualityChangeThreshold -q $iterationMode -pad
$paddingMode

    #Save the deconvolved image
    puts "\n<script>: Saving deconvolved image to $outputImagefName with
$outputImageFormat format"
    cd $outputImageDir
    b save $outputImageName -type $outputImageFormat
    puts "<script>: Image saved"
}
puts "<script>: CMLE done"
b del

$inputImage del
puts "<script>: Process finished succesfully"

## End of Execute Section

```

Anexo 5: Publicaciones

A continuación se presentan los trabajos publicados en congresos y un artículo (hasta la fecha en proceso de revisión) a una revista, en su mayoría del área científica, en los que se muestran las aplicaciones a distintos trabajos de investigación de las herramientas de software desarrolladas en el presente Proyecto.

Active Contours Quantify Shape Transitions and Multi-Scale Organization of Sphingomyelinase Generated Ceramide-Enriched Domains in Lipid Monolayers

Steffen Härtel, Laura Fanani, Jorge Jara & Bruno Maggio

Physics of Life: Dynamics of single molecules and dynamics of biological networks.

Humlebæk, Denmark, 21-27 August, 2005.

ABSTRACT: Sphingomyelinases (SMases) hydrolyze the membrane constituent sphingomyelin (SM) to phosphocholine and ceramide (Cer). Growing evidence supports that SMase-induced SM->Cer conversion leads to the formation of lateral Cer-enriched domains which drive structural reorganization in lipid membranes. Previously, we provided visual evidence in real-time for the formation of Cer-enriched domains in SM monolayers through the action of the neutral *Bacillus cereus* Smase [1]. In this work [2], we apply image processing procedures to disclose a succession of discrete morphologic transitions and lateral organization of Cer-enriched domains that underlay the SMase-generated surface topography. Our computer based analysis further reveals how these structural parameters couple to the generation of two-dimensional electrostatic fields, based upon the specific orientation of the lipid dipole moments in the Cer-enriched domains. In our quest to improve the quantification of the morphology of Cer-enriched domains, we recently adapted a segmentation procedure based on the equilibrium between virtual active contours and generalized gradient vector flow [5]. Active contours lead to an improved segmentation and quantification of the domain morphology due to a pixel-independent parameterization which permits to sculpture subtle characteristics of the domain morphology as a result of different initial conditions of the catalysis (Cer concentration).

**Contornos Activos y Flujo de Vectores de Gradientes Generalizados:
Modelos Deformables para la Segmentación y Parametrización de Estructuras
Biológicas**

Jorge Jara & Steffen Härtel

BIOMAT VI, 6° Encuentro Chileno de Biomatemática, 19-20 de Diciembre, 2005,
Universidad Austral de Chile, Valdivia, Chile.

En el campo de procesamiento de imágenes digitales, diferentes técnicas de segmentación permiten identificar estructuras dentro de una imagen y definirlas como regiones de interés (*regions of interest*, ROI). La segmentación de estructuras microscópicas en ROIs es indispensable para un posterior análisis paramétrico dentro de diversos contextos científicos.

Este trabajo aplica métodos para la segmentación y parametrización morfológica de estructuras biológicas, basada en contornos activos paramétricos que interactúan con flujos de vectores de gradientes generalizados (*generalized gradient vector flow*, GGVF) [1,2]. En base a la parametrización obtenida, se determina el grado de discretización de las regiones, suficiente para su representación y análisis computacional. Contornos activos son modelos matemáticos que controlan la deformación de curvas a través de una serie de parámetros intrínsecos (fuerzas internas). Estas curvas se ajustan dinámicamente a características determinadas por los GGVF (fuerzas externas) que se calculan en base a los relieves de intensidades dentro de imágenes. Los contornos activos paramétricos definen un funcional de energía interna y externa que se minimiza a través de la ecuación de Euler-Lagrange. Se añade un parámetro de tiempo al modelo para aproximar la solución de mínima energía por la vía iterativa.

En base a la evaluación de curvatura y puntos de inflexión de las curvas de contorno, se estableció un criterio de evaluación para definir un nivel de discretización de los contornos, relacionando el número de puntos para cada curva con el perímetro de la

ROI que describe. La implementación de contornos activos fue integrada a una aplicación computacional para procesamiento de imágenes, combinándose con otras técnicas de segmentación, parametrización y reconstrucción de modelos espaciales.

La técnica desarrollada fue aplicada al estudio de la formación de dominios laterales en monocapas de lípidos observadas por epifluorescencia [3] y al estudio de la condensación de la cromatina nuclear durante la muerte apoptótica y necrótica de células de cultivo (HeLa). Para ambas aplicaciones, contornos activos lograron abrir el acceso a informaciones estructurales novedosas dentro de los respectivos campos de investigación.

- [1] Kass M., Witkin A., Terzopoulos D. (1988) Snakes: active contour models. International Journal of Computer Vision 1:321-331
- [2] Xu Ch., Prince J. (1998) Generalized gradient vector flow external forces for active contours. Signal Processing 78: 131-139
- [3] Härtel S., Fanani L., Jara J., & B. Maggio. Active Contours Quantify Shape Transitions and Multi-Scale Organization of Sphingomyelinase Generated Ceramide-Enriched Domains in Lipid Monolayers Physics of Life: Dynamics of single molecules and dynamics of biological networks Humlebæk, Denmark, 21-27 August 2005

**Time resolved image processing routines for the study of membrane related processes
with fluorescence microscopy**

Steffen Härtel, Jorge Jara & Felipe Barros

13th International Symposium on Chromaffin Cell Biology, January 7-12, Pucón, Chile.

ABSTRACT: Image processing routines continuously help to reveal new features in biological specimen which are analyzed by optical microscopy. Beyond the quantification of biochemical properties which depend on the intrinsic characteristics of the fluorescent probes themselves, more complex, structure related information can be approached by diverse mathematical concepts. This presentation will focus on a series of strategies which have been developed during the last years in order to quantify morphology, texture, topology, and colocalization of biological objects in digital image series. We will present that: (i) texture sensitive descriptors identify apoptotic chromatin condensation and reveal diverse subjacent kinetics in human cornea epithelial cells (Härtel et al. 2003), (ii) morphologic and topology dependent descriptors unmask Sphingomyelinase induced lateral Ceramide domain formation as a transient, non-equilibrium process which suggests a new level of signal transduction from short-range (nm) to long-range (μm) scales (Härtel et al. 2005a), (iii) time dependent cell volume and surface related data contribute valuable information to the mechanisms of cell death (Castro et al. 2005), and (iv) local scaling index methods not only segment heterogeneous regions of interest from the image background (Härtel et al. 2005b), but also lead to a reliable quantification of colocalization in fluorescent image pairs. We guarantee the access to the latter procedure through the Online Processing Section on our web-site for Scientific Image Processing <http://www.scian.cl>.

- Härtel, S. et al. (2003) Staurosporine-induced Apoptosis in Human Cornea Epithelial Cells in Vitro. *Cytometry*, 08, 15-23.
- Härtel, S. et al. (2005a) Shape transitions and lattice structuring of ceramide-enriched domains generated by sphingomyelinase in lipid monolayers. *Biophysical Journal*, 88, 287-304.
- Härtel, S. et al. (2005b) Identification and Classification of Di- and Triploid Erythrocytes by Multi-parameter Image Analysis: A New Method for the Quantification of Triploidization Rates in Rainbow Trout (*Oncorhynchus mykiss*). *Archivos de Medicina Veterinaria XXXVII* (2), 147-154.
- Castro, J. et al. (2005) ATP steal: a novel mechanism linking Na⁺ with the onset of necrotic Ca²⁺ overload. In press. *Cell Death & Differentiation*

Sphingomyelinase-induced non-equilibrium phenomena in biointerfaces

María Laura Fanani, Steffen Härtel, Jorge Jara, Bruno Maggio

VI Congreso Iberoamericano de Biofísica. Madrid, España, 24-27 Septiembre, 2006.

ABSTRACT: Sphingomyelinase (SMase) is a membrane associated enzyme that hydrolyses Sphingomyelin (SM) to Ceramide (Cer) and Phosphocholine. The real time visualization by epifluorescence microscopy of its action against air/buffer SM monolayer shows the generation of laterally segregated Cer-rich domains. The morphology, dipolar properties and lattice organization of the enzyme-generated domains appear as a non-equilibrium phenomenon with a slow relaxation process. The transition from fractal-like far-from-equilibrium domain topography to compact morphology is studied. These results introduce a new perspective for focusing on crosstalk between the dynamics of lateral structure in biointerfaces and biocatalytic activity relevant to cell signaling events.



Sphingomyelinase-induced non-equilibrium phenomena in biointerfaces

Maria Laura Fanani¹, Steffen Härtef, Jorge Jara², Bruno Maggio¹

¹Departamento de Química Biológica - CIQUIBIC, Facultad de Ciencias Químicas - CONICET, Univ. Nacional de Córdoba.

²Anatomy and Development Biology Program, ICBM, Faculty of Medicine, Universidad de Chile.

Supported by: SECYT-UNC, CONICET, FONCYT, FONDECYT-Chile: 1060890



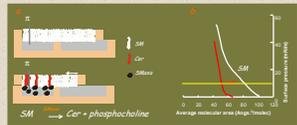
Summary

Sphingomyelinase (SMase) is a membrane associated enzyme that hydrolyses Sphingomyelin (SM) to Ceramide (Cer) and Phosphocholine. The real time visualization by epifluorescence microscopy of its action against an/over SM monolayer shows the generation of laterally segregated Cer-rich domains. The morphology, dipolar properties and lattice organization of the enzyme-generated domains appear as a non-equilibrium phenomenon with a slow relaxation process. The transition from fractal-like far-from-equilibrium domain topology to compact morphology is studied. These results introduce a new perspective for focusing on crosstalk between the dynamics of lateral structure in biointerfaces and biocatalytic activity relevant to cell signaling events.

Materials and Methods

Real time hydrolysis of SM monolayers by *B. cereus* SMase was visualized by epifluorescence microscopy using DiI-C₁₂ (1 mol%) for dyeing lipid phase. 10mM Tris-HCl, 125 mM NaCl, 3 mM MgCl₂, pH 8, was used as subphase and the reaction took place at 25°C and a surface pressure of 7 mN/m in a zero order trough (Kroon μ Trough S, Helsinki, Finland) (Fig. 1a). Since the product Cer has a molecular area smaller than SM (Fig. 1b), the reaction course can be followed as total area reduction at constant surface pressure after SMase injection in the subphase. As the reaction reaches pseudo zero-order regime, SMase action was quenched by an injection of EDTA in the subphase (final concentration 12mM) (Fig. 3b). All images were 175 x 225 μ m real size. Surface topography analysis was performed using Iterative Image Analysis developed through IIA (ITT, www.ittvis.com/idi_CO_USA).

Figure 1



Introduction

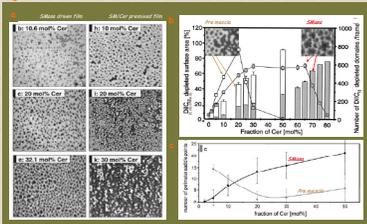
SMase action on SM monolayers produce laterally segregated domains

In previous works we described the spontaneous generation of laterally segregated Cer-enriched domains after the enzymatic breakdown of SM by *B. cereus* SMase (Fanani et al. 2002, Härtef et al. 2005). Several morphologic transitions of the condensed phase domains occur with changes of the kinetic regime. The long range structure is also affected by the enzymatic SM \rightarrow Cer breakdown. Hexagonal lattice structuring is achieved when the enzymatic reaction reaches the pseudo zero-order regime. These phenomena can be explained by dipolar domain repulsion and electric field effect.

SMase-generated surface topography differs of SM/Cer premixed films

At the same lipid composition, SM/Cer premixed monolayers show higher proportion of condensed phase area in comparison with enzymatically generated SM/Cer monolayers (Fig. 2a,b, Fanani et al. 2002). Premixed monolayers also show more relaxed condensed-liquid crystalline phase boundary (Fig. 2c), and form a more heterogeneous lattice (Härtef et al., 2005). These differences can only be explained on the basis of a far-from-equilibrium situation for the monolayer generated by SMase.

Figure 2



References

- Fanani, M.L., Härtef, S., Oliveira, R.G. and Maggio, B. (2002) Bidirectional control of Sphingomyelinase activity and surface topography in lipid monolayers. *Biolophy*, 51, 83, 3416-24.
- Härtef, S., Fanani, M.L. and Maggio, B. (2005) Shape transitions and lattice structuring of ceramide-enriched domains generated by Sphingomyelinase in lipid monolayers. *Biolophy*, 52, 181, 357-364.
- Jørgensen, K., Klapper, A. and Bittman, R.L. (2000) Nonequilibrium Lipid Domain Growth in the Gel-Fluid Two-Phase Region of a DiI₁₂-DPPC/Lipid Mixture Investigated by Monte Carlo Computer Simulation, FT-IR, and Fluorescence Spectroscopy. *J. Phys. Chem. B*, 104, 11763-11773.
- Xu, C. & Prince, J.L. 1998. Generalized gradient vector flow external forces for active contours. *Signal Processing* 71: 131-139.

Results

SMase-generated domains show increase of condensed phase proportions after enzyme activity halting, denoting enrichment in SM content.

Fig 3 shows changes after EDTA injection (SMase stop) in SM/Cer monolayer topology. An increase in the condensed phase fraction is observed and quantitatively confirmed by Image Analysis (Fig. 4a shows the fraction of remaining SM-enriched, fluid, phase). Decrease of fluid phase follows an exponential decay with half lifetime of 4.2 min.

After activity halting, the total Cer content remains constant. In these conditions the increase in condensed phase fraction can only be explained as an increased partitioning of SM into the Cer-enriched domains.

The proportion of Cer in the condensed phase can be calculated assuming that the fluid phase can only accept about 2 mole % of Cer, with the remaining Cer in the condensed phase.

Fig 4b shows the percentage of condensed phase as a function of total Cer mol fraction in both, SMase generated conditions (open circles and solid line) and SM-Cer premixed conditions (gray circles and gray line).

Figure 3

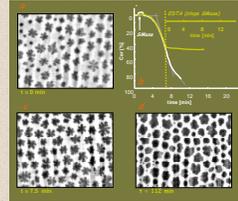
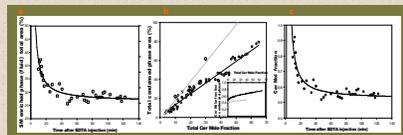


Figure 4

Insert shows the calculated Cer mole fraction in the condensed phase in each condition. The enzymatically-generated condensed domains shows a higher content of Cer than the SM/Cer premixed condensed domains. Fig. 4c shows the exponential decay (half lifetime 4.4 min) of condensed domain Cer content after SMase halt with EDTA.



Active contour models improve border parameterization of under- and oversampled Cer-enriched domains

A threshold based segmentation of lipid domains only provides a gross estimation of the undersampled domain border (Fig. 5a) whereas active contours successively sculpture the original domain shape through the variation of intrinsic active contour parameters (Fig. 5a). We use a modified version of an active contour model originally described in Xu & Prince (1998). The plot shows the variation of domain border curvatures in dependence of one intrinsic contour parameter (f = frequency of interpolation points in number per pixel). The insets visualize how the segmentation of the domain borders improve with increasing f . For this experiment, $f = 3.9$ was considered to represent the border trajectory with sufficient detail (95 % of the saturation value). For Fig. 5b, $f = 0.7$ was considered to yield the best results for domain border segmentation and parameterization (perimeter curvature and number of saddle points).

SMase-generated domains shows morphological relaxation (fractal like to compact) after enzymatic halt

As can be seen in Fig. 3, fractal like highly curved structures relax to almost circular structures after 112 min. Although the average number of saddle points per domain shows a wide dispersion an exponential decay with a half lifetime of 36 min is observed as a function of time (Fig. 6a). The histogram of number of saddle point per domain at two times of the relaxation evolution (Fig. 6b). This clearly shows that immediately after SMase stop most domains present 6 saddle point while after 112 min most domains are circular. Domain border circularity is calculated as perimeter²/area (Fig. 7a). The average circularity shows high dispersion. Nevertheless an quick exponential decay can be seen with a half lifetime of 1.1 min. This parameter appears to be highly sensitive to morphology change and start a sequence of temporal events as: Decrease in circularity ($t < 1$ min) \rightarrow enrichment of SM in condensed phase domains ($t < 4$ min) \rightarrow decrease saddle point number per domain ($t < 36$ min). Fig. 7b shows histogram of frequency: time relaxation induce shift of circularity peak to lowers values.

Figure 5

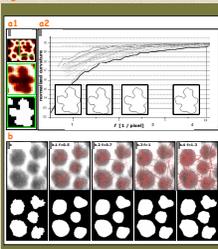


Figure 6

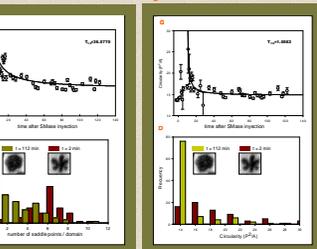
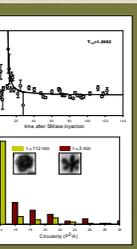


Figure 7

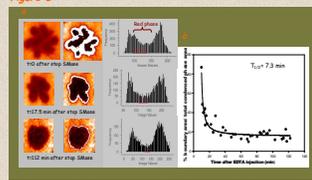


A third phase is present as a wetting layer at the domain boundary and could explain a surfactant-induced slow relaxation dynamic

Enzymatically-generated, far-from-equilibrium domains shows a third phase of the domain boundary, where DiI-C₁₂ probe show an intermediate solubility (red phase, Fig. 8a), that narrows with time.

This 'intermediate' phase represents a high proportion of the total domain area at short times after SMase halting and decays exponentially with time with a half lifetime of 7.3 min (Fig. 8b). Previous studies were made in more simple non-equilibrium, two-component lipid bilayer systems (Jørgensen et al. 2000). Theoretical investigation of the topography of the interface by Monte Carlo Simulation led to the proposal that a wetting layer by a third phase present at the nanoscale domain boundary could act as surfactant slowing down the equilibration dynamic. In this work the presence of a 'wetting layer' on the microscale range topography, studied by epifluorescence, supports the idea of a surfactant-induced slow relaxation dynamic.

Figure 8



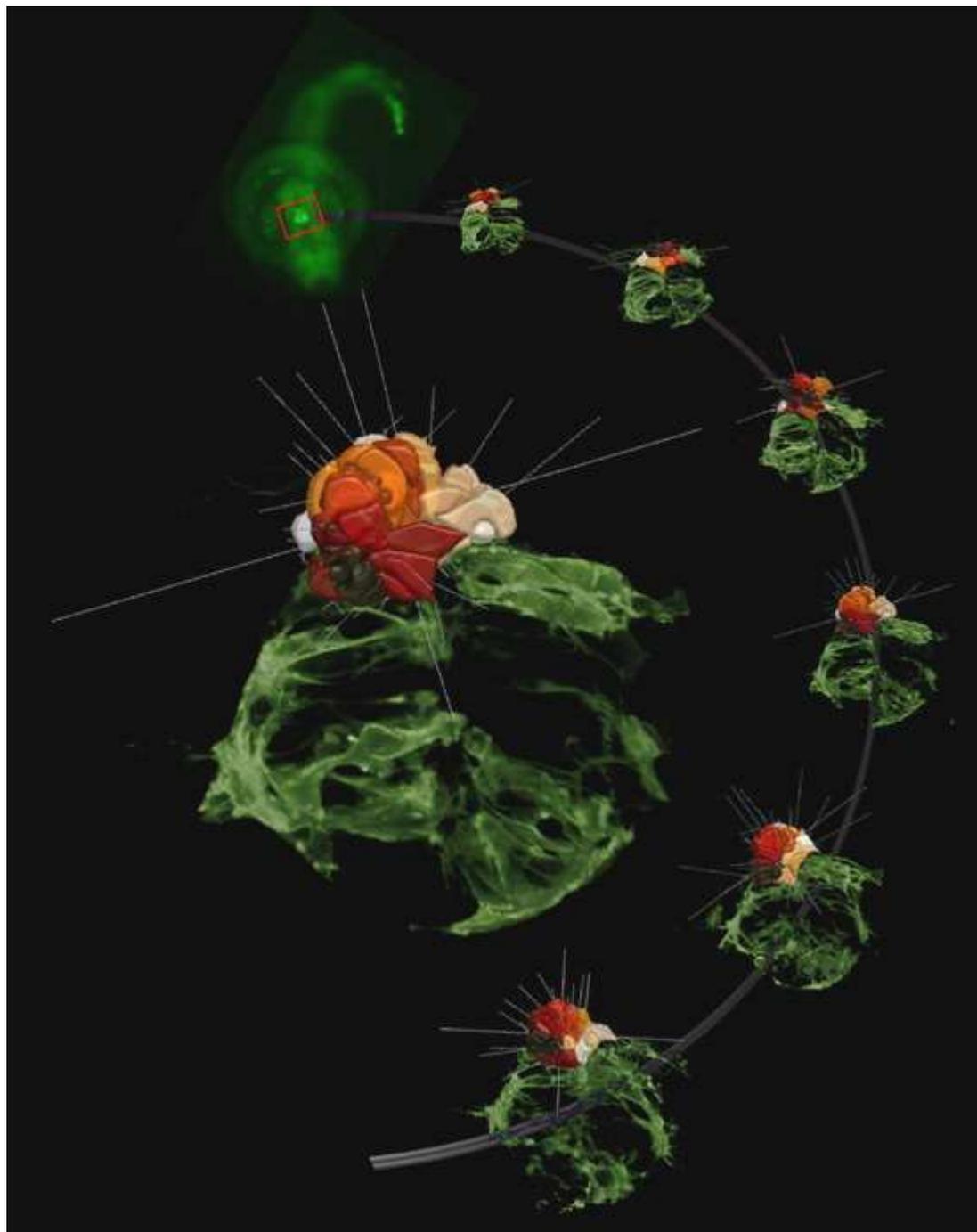
Morfogénesis Asimétrica del Organo Parapineal de Pez Cebra

Steffen Härtel, Carmen Lemus, Jorge Jara y Miguel Concha

Premio Nikon-Ivens a la Mejor Imagen en Biología Celular

Sociedad de Biología Celular de Chile, XX Reunión Anual, 8-12 Octubre 2006

Pucón, Chile



Summary

We applied *in vivo* confocal microscopy of GFP-transgenic zebrafish (upper left) in combination with 3D image analyses to study the asymmetric morphogenesis of the diencephalic parapineal organ (red square) on a supra-cellular, cellular, and sub-cellular level. The image shows the entire pineal complex (green fluorescence) and emphasizes the emerging diencephalic parapineal organ by colour coded mathematical surface models.

Morphology of the segmented parapineal cells is refined by an active surface model which iteratively adjusts the surface mesh towards the precise morphology of the cellular structures. This procedure is essential for a precise mathematical morpho-topological analysis, mostly because of the adversarial diffraction limited resolution in the z-dimension of confocal image stacks. 3D Morphology and topology of the reconstructed cellular and supra-cellular structures during morphogenesis was quantified by principal axis transformations (grey axis) and 3D moment invariants.

The image series focuses on the asymmetric migration and formation of the parapineal organ between 26 – 38 h post fertilization. The combination of biological and mathematical tools highlights diverse hidden aspects during this process that could not have been detected otherwise. For example a rapid transition between predominantly parallel cell orientations towards predominantly perpendicular orientations, a phenomenon which requires a precise control of cell shape and polarity [1, 2]. The orientational transition is followed by a phase of polarized cell motility (formation of blebs and filopodia).

The combination of biological and mathematical approaches becomes essential to access morphogenetic mechanisms which control asymmetry and migration.

1 S. Härtel, C.G. Lemus, J. Jara, M.E. Cabrejos, M.L. Concha. Análisis de la morfogénesis asimétrica del órgano parapineal de pez cebra en tres niveles de organización. XX Reunión anual, Sociedad de Biología Celular de Chile, Pucón, Chile, 8-12 de Octubre, 2006.

2 S. Härtel, J. Jara, C.G. Lemus, M.L. Concha 3D Morpho-Topological Analysis of Asymmetric Neuronal Morphogenesis in Developing Zebrafish. CompIMAGE Computational Modelling of Objects Represented in Images, 20-21 October, Coimbra, Portugal.

3D Morpho-Topological Analysis of Asymmetric Neuronal Morphogenesis in Developing Zebrafish

S. Härtel, J. Jara, C.G. Lemus, M.L. Concha

Computational Modelling of Objects represented in Images (CompIMAGE). Coimbra,
Portugal, 20-21 October, 2006.

ABSTRACT: We applied in vivo confocal microscopy of GFP-transgenic zebrafish in combination with 3D image analyses to study the asymmetric morphogenesis of the diencephalic parapineal organ on a supra-cellular, cellular, and sub-cellular level. Following a rough manual segmentation of the respective regions of interest (ROIs), the morphology of generated surface meshes was refined by an active surface model which iteratively adjusts the mesh towards the morphology of the cellular structures. This procedure is essential for a precise morpho-topological analysis, mostly because of the adversarial diffraction limited resolution in the z-dimension of confocal image stacks. 3D Morphology and topology of the reconstructed cellular and supra-cellular structures during morphogenesis was quantified by principal axis transformations and 3D moment invariants. Our data indicates that migration of the parapineal organ is accompanied by a rapid transition between predominantly parallel cell orientations towards predominantly perpendicular orientations, a phenomenon which requires a precise control of cell shape and polarity. The orientational transition is followed by a phase of polarized cell motility in which membrane protrusions in the form of blebs and filopodia become oriented in the direction of the asymmetric migration. The morpho-topological descriptors unveil information that is not perceptible for a direct visual analysis of the microscopical data sets. This approach becomes essential to access morphogenetic mechanisms which control asymmetry and migration.

1 INTRODUCTION

1.1 Left-right asymmetry in the zebrafish brain

Despite our increasing understanding of the mechanisms that control left-right asymmetry in the heart, little is known of the morphogenetic mechanisms that establish lateralized circuitry in the vertebrate brain. In recent years we have studied the mechanisms that control the asymmetric development of neuronal nuclei in the zebrafish brain (Concha et al. 2000/2003; Concha 2004). Development of brain asymmetry is preceded by left-sided expression of a number of genes, and is characterized by the allocation of a single midline nucleus on the left side (the parapineal organ), and by left-right differences in the pattern of neuronal differentiation within a paired nucleus (the habenulae). Asymmetric morphogenesis of the parapineal organ is a key event as it induces further asymmetric morphogenesis, and involves the migration of parapineal precursors from their place of origin at the midline towards the left side of the brain (Concha et al. 2003; Gamse et al. 2005). To understand the morphogenetic mechanisms responsible of this phenomenon, we have recently initiated a 3D morphological and topological description of the cell behaviors underlying the asymmetric positioning of the parapineal organ.

1.2 Morpho-topological analysis of the parapineal organ

Image processing routines have become indispensable for the reliable detection and quantification of light-microscopic data. Sophisticated algorithms not only restore optical information on the level of data acquisition (see 2.3 below), they become increasingly important for the characterization and quantification of morpho-topological information in biological structures. In the past, we have developed diverse approaches to access biochemical and structure related information on a 2D level (e.g. Carrer et al. 2003; Härtel et al. 2003/2005a/b; Alvarez et al. 2005). In this contribution we continue our approach to reveal biologically relevant information on a 3D level. As shown recently, regulation of cellular volume and morphology can be monitored with great precision in cultured HeLa cells (Castro et al. 2006). In order to derive morpho-topological information from the

parapineal cells and the parapineal organ during the asymmetric morphogenesis in zebrafish embryos, we present an approach based on principle axis transformation and the calculation of rotation, translation, and scale invariant moments. This method has recently been suggested for morphometry of cortical sulci by Mangin et al. (2004). For 3D presentation of cell surfaces and calculation of surface related parameters, we adopt an active surface model introduced by Ahlberg (1996) based on the formulation from Kass et al. (1988). Analytical characterizations are performed on different scales of cellular organization: a supra-cellular scale (pineal and parapineal organ), a cellular scale (parapineal cells), and a sub-cellular scale (blebs and filopodia).

2 MATERIAL & METHODS

2.1 Manipulation of GFP-transgenic zebrafish

Transgenic zebrafish expressing green fluorescent protein (GFP) in the dorsal diencephalon were generated previously (Concha et al. 2003), and kept under standard laboratory conditions in the Fish Facility of the Faculty of Medicine, Universidad de Chile. Embryos between 24 and 38 hours post-fertilisation were manually dechorionated, anesthetized with Tricaine (3-amino benzoic acid ethyl ester, Sigma), and mounted in a custom-made chamber in 1% agarose dissolved in embryo medium (Concha et al. 2003). After the end of the experiment, embryos were removed from the chamber and sacrificed with a Tricaine overdose.

2.2 Acquisition of confocal microscopic images

The morphogenesis of the parapineal organ in living GFP-transgenic zebrafish embryos was followed by confocal epifluorescence microscopy (Leica TCS SP) under controlled temperature conditions (28°C). 3D Image stacks, intensity $I \in [0, 255]$, were captured with a 63x (NA 0.9) water-immersion objective, excitation/emission at 488/505–560 nm, with typical xyz stack [512×512×70 voxels] and voxel [0.116×0.116×0.5 μm/voxel] dimensions. The selected vertical sampling distances guaranteed reliable volume determinations and shape analysis.

2.3 Deconvolution and image analysis

Huygens Scripting (Scientific Volume Imaging BV, www.svi.vl, Hilversum, Netherlands), using a Maximum Likelihood Estimator (MLE) algorithm was used to deconvolve 3D confocal data in order to reveal hidden details of the biological structures and to improve the signal to noise ratio significantly. All further image processing routines for visualization and morpho-topological analysis were written in our laboratory, using Interactive Data Language (IDL, ITT, www.itvis.com/idl/, CO, USA).

2.4 Cell segmentation and surface reconstruction

A first approximation of ROIs including sub-cellular structures (blebs or filopodia) and parapineal cells was obtained manually by drawing object contours into the xy-frames of the deconvolved image stacks, using an Interactive Pen Display (Cintiq-15X, Wacom). Binary masks of the ROI were generated with a custom-made macro written for the public domain image analysis software Image-SXM (<http://www.liv.ac.uk/~sdb/imageSXM>). The definition of cellular contours in 3D was improved significantly by the application of a modified version of an active surface model that was originally presented by Ahlberg (1996) expanding the 2D active contour model developed by Kass et al. (1988). The active surface model parameterizes internal forces like elasticity (α) or rigidity (β), which counteract line tension or curvature and mimic intrinsic physical properties of a deformable surface.

Following Xu & Prince (1998), surface tension and curvature are induced by external force fields, which are derived from the intensity gradients and laplacians of the image data by an iterative algorithm. The external forces are parameterized by so-called Generalized Gradient Vector Flows (GGVF) and consist of 3D vector fields, which attract surface points towards the object borders. Force balance between internal and external forces is solved by the Euler-Lagrange condition for the minimization of an energy functional E for a parametric surface $C(s) = [x(s), y(s), z(s)]$, $s \in [0,1]$.

$$E = \int_0^1 0.5 \cdot [\alpha |C'(s)|^2 + \beta |C''(s)|^2] + E_{ext}(C(s)) ds \quad [1]$$

For the cellular structures presented in this work, an initial surface mesh is derived from the manually approximated ROI contours.

Precise assimilation of the active surface mesh towards the morphology of the cellular structures was supervised interactively by setting the appropriate parameter combination to the following coefficients: α , β , viscosity (γ), external force (k) and iterations (t).

2.5 Calculation of surface curvature

Surface curvature (κ) was calculated for each mesh node in respect to its adjacent neighbours. For this purpose spheres with radius ($r = \kappa^{-1}$) were fitted to each node and 3 of its neighbours. For nodes with more than 3 neighbours, we calculate κ for all possible 3 neighbour combinations and derived a mean value for κ .

2.6 Invariant moments

Eigen vectors ($\underline{e}_{1,2,3}$) and Eigen values ($\lambda_{1,2,3}$) of segmented objects were calculated on three levels of spatial organization: (i) on the sub-cellular level (blebs and filopodia, $\varnothing = 1-3 \mu\text{m}$), (ii) on the cellular level (individual parapineal cells, $\varnothing = 5-15 \mu\text{m}$), and (iii) on the supra-cellular level (parapineal organ, $\varnothing = 20-100 \mu\text{m}$). Eigen vectors and values were derived from the inertia tensor T (or covariance matrix) by Householder reduction and the QL method (based on the routine `tqli` described by Press et al. 1992). T is required in order to determine the rotation of a rigid body around its centre of mass. In analogy to mechanical physics, the segmented structures were considered to have a uniform mass distribution $\rho(x,y,z) = 1$ in each segmented voxel. The Eigen values and vectors were ordered in respect to the size of λ_i , which directly represents the rotational inertia in respect to each axis e_i . The Eigen systems $\underline{E} = [\lambda_1 \cdot \underline{e}_1, \lambda_2 \cdot \underline{e}_2, \lambda_3 \cdot \underline{e}_3]$ were used to access object morphologies (morphometry), spatial orientation between objects of the same level of organization (Fig. 1), and spatial organization between the different levels of organization (Fig. 2). In addition, segmented parapineal cells were transformed into their corresponding

Eigen systems $\rho(x,y,z) \rightarrow \rho(x',y',z')$ and aligned in box pattern in order to improve the visual perception of more subtle geometric features.

Object morphometry was parameterized by translation-, rotation-, and scale-invariant descriptors (μ'_{ijk} invariant moments of order ijk), which were calculated according to Castleman (1996):

$$\mu_{ijk} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x' - \bar{x})^i (y' - \bar{y})^j (z' - \bar{z})^k \rho(x', y', z') dx' dy' dz' \quad [2]$$

$$\mu'_{ijk} = \mu_{ijk} / \mu_{000}^{((i+j+k)/3)+1} \quad [3]$$

3 RESULTS

Fig. 1 shows a temporal series of 3D reconstructions in which the topological reorganization of the parapineal organ is depicted at supra-cellular and cellular levels. The initial phase of asymmetric migration is revealed as a gradual movement of the parapineal organ to the left in respect to the mayor symmetry axis of the entire pineal complex. This phenomenon is accompanied by oscillations in the orientation of the 1st principal axis of the parapineal organ and by changes in the orientation of the 1st principal axes of individual parapineal cells: at early stages principal axes show a predominantly parallel alignment, which is gradually lost during morphogenesis.

Fig. 2 shows a quantitative analysis of the temporal reorganization of the alignments of the 1st principal axes of the parapineal cells in respect to each other (red circles) and in respect to the 1st principal axes of the entire parapineal organ (white squares), which accompanies the asymmetric cell migration described in Fig. 1. Both parameters indicate a reorganization of a predominantly parallel axes alignment (26-28 h post-fertilization) towards a predominantly perpendicular alignment (32-38 h), undergoing an intermediate phase of random distribution (30 h). The transition between the parallel alignment toward the predominantly perpendicular orientation between the principal axes of individual cells and the axis of the entire parapineal organ occurs within a time span of 2-4 h.

Further information about the process of reorganization during the orientational transition presented in Fig. 2 was obtained by a more detailed analysis of the distribution of the angles between the 1st principal axes of individual parapineal cells. Fig. 3 connects the spatial allocation within the 3D reconstructions with the parameter distribution of a sample population that enters the phase of orientational transition (28 h).

As can be observed, cells with a predominantly parallel axes alignment form the left wing of the parapineal organ (green), while cells that diverge from the parallel alignment constitute the right wing of the organ (red). In addition, cells with an intermediate degree of alignment are located close to the centre (yellow).

On a final level of 3D topological analysis, we analysed the distribution of sub-cellular structures (membrane protrusions) within cells of the parapineal organ. Fig. 4 visualizes the sub-cellular structures for a wild type and a mutant embryo and shows the spatial distribution of membrane protrusions in the form of blebs (red) and filopodia (yellow). In the wild type, parapineal membrane protrusions show a polarised behaviour during the asymmetric morphogenesis as they concentrate on the left side to which the parapineal organ migrates. In contrast to the wild type, protrusion polarisation is lost in a mutant embryo that is characterized by an impaired parapineal migration.

4 DISCUSSION

Principal axes transformation and invariant moments have been introduced a few decades ago for the characterisation and recognition of pattern in 2D images by Hu (1962), and for the analysis of 3D data sets by Lo & Don (1989). The method presents a standard mathematical tool that is applied in a wide range of disciplines including statistics, quantum mechanics, or classical mechanics. In biological and medical science however, the application of this attractive tool is rather sparse.

Recently, 3D invariant moments have been reported to provide good discriminatory power to identify handedness- and sex-correlated shapes of the cortical sulci in brain morphometry (Mangin et al. 2004). The general low acceptance of invariant moments in biological and

medical science might be caused by the slightly abstract formalism (see [2,3]). Nevertheless, the invariants up to the power of 2 have a direct physical relevance: they define the principal rotational axis of a solid body of homogeneous or heterogeneous mass distribution ($\rho(x,y,z) = 1$ or $\rho(x,y,z) = f(x,y,z)$).

The calculated Eigen system $E = [\lambda_1 \cdot \underline{e}_1, \lambda_2 \cdot \underline{e}_2, \lambda_3 \cdot \underline{e}_3]$ directly presents the rotational symmetry axes (\underline{e}_{123} , axes where the angular momentum is un-changed, unless an external torque is applied) in combination with its respective rotational inertia (λ_{123}). Length and orientation of projected axes ($\lambda_i \cdot e_i$) therefore represent and quantify an essential visual characteristic of a solid body or cellular sur-face (see Fig. 1-3): the longest principal axis ($\lambda_1 \cdot \underline{e}_1$) generally aligns with the elongated axis of a cellular body.

Principal axes become explicitly powerful when it comes to characterize the orientation of individual cells which form a complex supra-cellular structure. In case of the developing parapineal organ, an agglomeration of 15-20 cells already makes it difficult to visually explore the orientation of each individual cell (see Fig. 1-3). Therefore the morphotopological descriptors unveil information that is not perceptible for a direct visual analysis of the microscopical data sets. For example, the transition of the orientational organisation within the parapineal organ could not be perceived visually without the projection of the principal axes (Fig. 1).

The rapid access to statistical properties of the axes components on a cellular and a supra-cellular level permitted to determine the temporal interval for the orthogonal transition in parapineal cell organisation (Fig. 2). Finally, the back projection of colour coded parameter intervals revealed a supra-cellular pattern inside the parapineal organ (Fig. 3). Altogether, the analysis allowed us to detect a morphogenetic asymmetry prior to the lateral migration of the organ (compare 28 h and 30 h in Fig. 1 with Fig. 3). The detected internal cellular pattern directly leads to the question of the underlying biological mechanisms.

In conclusion, a relatively simple mathematical analysis unveils new biological questions in respect to the first phenomena of asymmetry during parapineal morphogenesis. We are

presently expanding the analysis in all levels of organization and include mutant zebrafish embryos (Fig. 4) to address the biological implication of the observed phenomena.

5 REFERENCES

Ahlberg, J. 1996. Active Contours in Three Dimensions. Thesis project done at Computer Vision Laboratory, Linköping University, Sweden.

Alvarez, M.; Godoy, R.; Heyser, W. & Härtel, S. 2005. Anatomical-physiological determination of surface bound phosphatase activity in ectomycorrhiza of *Nothofagus obliqua* based on image processed confocal fluorescence microscopy. *Soil Biology and Biochemistry* 37(1): 125-132.

Castleman, K.R. 1996. Digital Image Processing. Prentice Hall, Englewood Cliffs, NJ, USA.

Castro, J.; Ruminot, I.; Porras, O.; Flores, C.; Hermosilla, T.; Verdugo, E.; Härtel, S. & Barros, L.F. 2006. ATP steal: a novel mechanism linking Na⁺ with the onset of necrotic Ca²⁺ overload. In press: *Cell Death & Differentiation*.

Carrer, D.; Härtel, S. & Maggio, B. 2003. Ceramide Modulates the Lipid Membrane Organization at Molecular and Supramolecular Levels. *Chemistry and Physics of Lipids* 122: 147-152.

Concha, M.L.; Burdine, R.D.; Russel, C.; Schier, A.F. & Wilson, S.W. 2000. A Nodal signalling pathway regulates the laterality of neuroanatomical asymmetries in the zebrafish forebrain. *Neuron* 28: 399-409.

Concha, M.L.; Russell, C.; Regan, J.C.; Tawk, M.; Sidi, S.; Gilmour, D.; Kapsimali, M.; Sumoy, L.; Goldstone, K.; Amaya, E.; Kimelman, D.; Nicolson, T.; Grunder, S.; Gomperts, M.; Clarke, J.D. & Wilson, S.W. 2003. Local tissue interactions across the dorsal midline of the forebrain establish CNS laterality. *Neuron* 39: 423-438.

Concha, M.L. 2004. The dorsal diencephalic conduction system of zebrafish as a model of vertebrate brain lateralisation. *Neuroreport* 15: 1843-1846.

- Gamse, J.T.; Kuan, Y.S.; Macurak, M.; Brosamle, C.; Thisse, B.; Thisse, C.; & Halpern, M.E. 2005. Directional asymmetry of the zebrafish epithalamus guides dorsoventral innervation of the midbrain target. *Development* 132: 4869-4881.
- Härtel, S.; Zorn-Kruppa, M.; Tykhonova, S.; Alajuuma, P.; Engelke, M. & Diehl, H. 2003. Staurosporine-Induced Apoptosis in Human Cornea Epithelial Cells In Vitro. *Cytometry* 08, 15-23.
- Härtel, S.; Fanani, M.L.; & Maggio, B. 2005a. Shape transitions and lattice structuring of ceramide-enriched domains generated by sphingomyelinase in lipid monolayers. *Biophysical Journal* 88: 287-304.
- Härtel, S.; Rojas, R.; Räth, C.; Guarda, M.I. & Goicoechea, O. 2005b. Identification and Classification of Di and Triploid Erythrocytes by Multiparameter Image Analysis: A New Method for the Quantification of Triploidization Rates in Rainbow Trout (*Oncorhynchus mykiss*). *Archivos de Medicina Veterinaria* 37(2): 147-154.
- Hu, M.K. 1962. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory* 8 (February): 179–187.
- Kass, M.; Witkin, A. & Terzopoulos, D. 1988. Snakes: Active Contour Models. *International Journal of Computer Vision* 1: 321-331.
- Lo, C.H. & Don, H.S. 1989. 3D moment forms: their construction and application to object identification and positioning. *IEEE PAMI* 11 (October): 1053–1064.
- Mangin J.F.; Poupon, F.; Duchesnay, E.; Riviere, D.; Cachia A.; Collins, D.L.; Evans, A.C. & Regis J. 2004. Brain morphometry using 3D moment invariants. *Medical Image Analysis* 8: 187–196.
- Press, W.H.; Teukolsky, S.A.; Vetterling, W. & Flannery, B.P. 1992. *Numerical Recipes in C: The Art of Scientific Computing* (2nd edition), section 11.3, Cambridge University Press.
- Xu, C. & Prince, J.L. 1998. Generalized gradient vector flow external forces for active contours. *Signal Processing* 71: 131-139.

6 ACKNOWLEDGEMENTS

This work was supported by FONDECYT 1060890, PBCT ACT 47, and ICM P04-068-F.

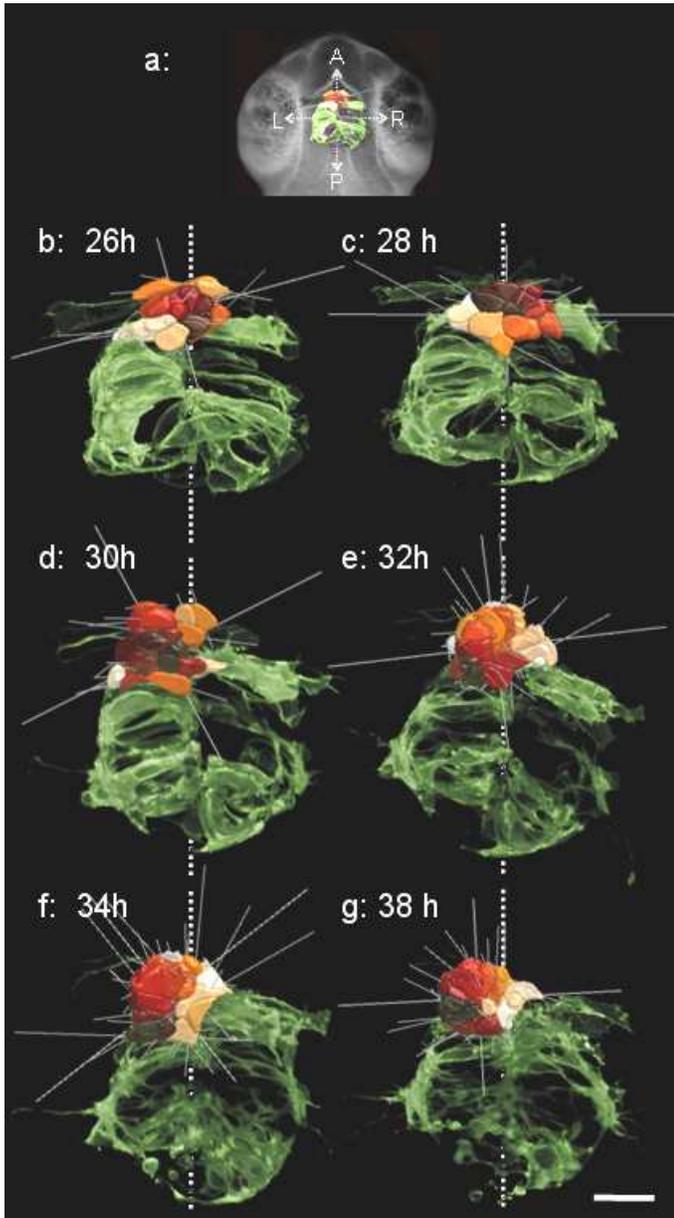


Figure 1. Topological reorganization of the parapineal organ during asymmetric morphogenesis. [a] Scheme shows the transparent head of a developing fish embryo and the fluorescence of the GFP-transgenic pineal complex (green). The parapineal organ is located at the anterior-most end of this complex. [b-g] Pineal complex (green) and colour-coded active surface models of individual cells of the parapineal organ at 26, 28, 30, 32, 34, and 38 hours post-fertilisation. Left-right symmetry axes of the pineal complex are shown by dotted, vertical lines. Thick and thin grey lines depict principal axes of the entire parapineal organ and of individual reconstructed cells, respectively. Abbreviations: anterior (A), posterior (P), left (L), right (R). Scale bar = 20 μ m.

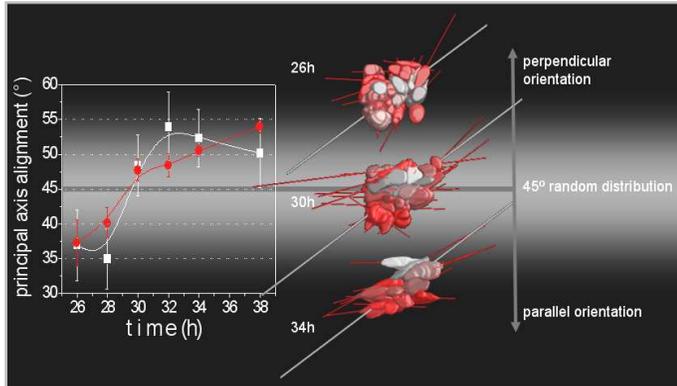


Figure 2. Principal axes alignment during parapineal morphogenesis. [Left] Alignment between the 1st principal axis of individual parapineal cells (red circles) is compared to the alignment between the 1st principal axis of the individual cells with the 1st principal axis of the entire parapineal organ (white squares). The parameter ‘principal axis alignment’ represents the mean values calculated from the respective angular distributions in three independent experiments. Mean values were connected by β -spline curves. Error bars represent standard deviations. [Centre-Right] Representative surface reconstructions of the parapineal organ based on active contours in combination with the respective 1st principal axis of individual parapineal cells (red lines) and of the entire parapineal organ (grey lines) at 26, 30, and 34 hours post-fertilisation. The reconstructions open a direct visual access to the data presented in the plot. The data reveals a transition of the organization of parapineal cells from a predominantly parallel orientation towards predominantly perpendicular orientation.

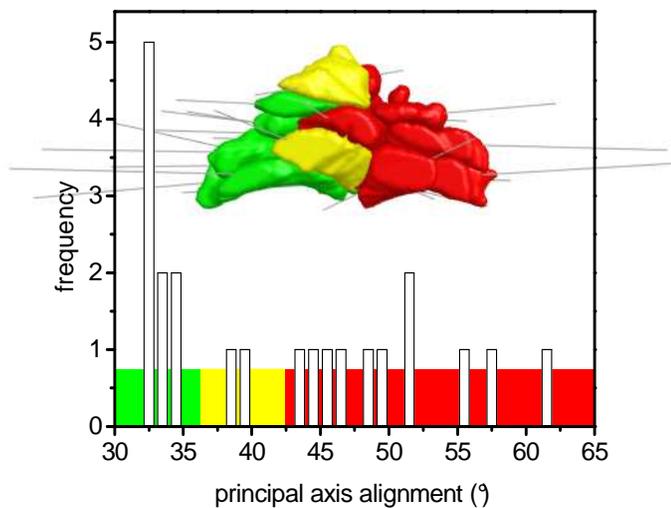


Figure 3. Asymmetric distribution of the axial alignment of parapineal cells during morphogenesis. The histogram plots the frequency distribution of the parameter ‘inter cellular alignment of the 1st principal axis (e_1)’ for a representative parapineal cell population at 28 h post fertilisation. Three intervals of the angular alignment were colour coded according to the predominant orientation: parallel (green), random (yellow), and perpendicular (red) (see Fig. 2). Colours were projected to the surface of the corresponding parapineal cells.

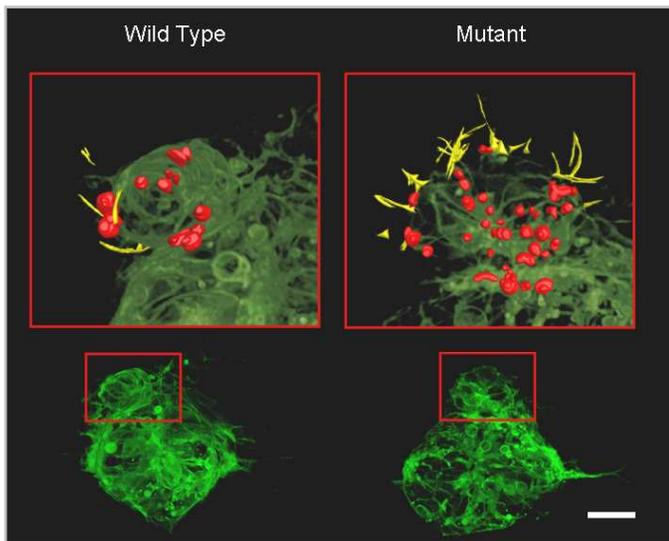


Figure 4. Spatial distribution of sub-cellular structures within the parapineal organ of wild type (left) and mutant (right) zebrafish embryos. As can be observed in the upper 3D reconstructions, membrane protrusions in the form of blebs (red), and filopodia (yellow) are oriented in the direction of the asymmetric migration toward the left side of the wild type pineal complex. In contrast, blebs and filopodia are symmetrically oriented in the mutant. Scale bar = 20 μ m.

Steffen Härtel
Jorge Jara
Carmen Lemus
Miguel Concha

shartel@med.uchile.cl



Anatomy & Developmental
Biology Program
Faculty of Medicine
Universidad de Chile
Santiago

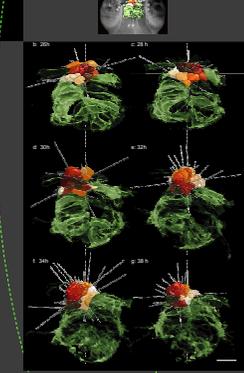
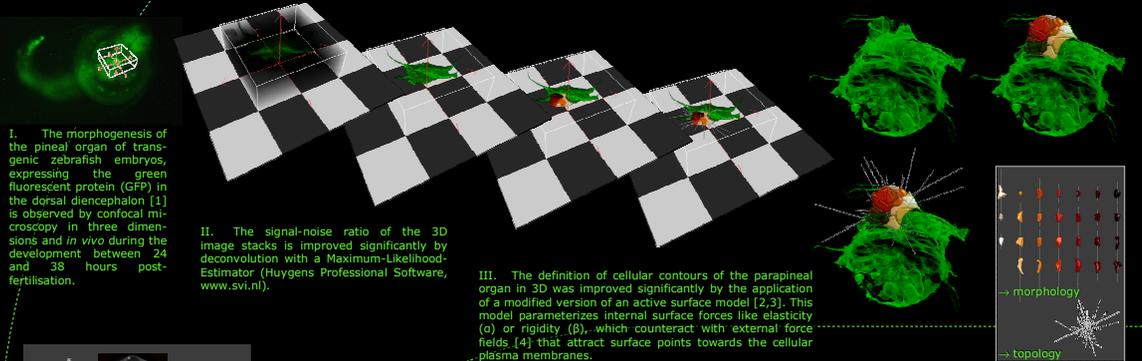
Financed by:
FONDECYT 1060899
FICET ACT 47
ICM P04-068-F

3D Morpho-Topological Analysis of Asymmetric Neuronal Morphogenesis in Developing Zebrafish

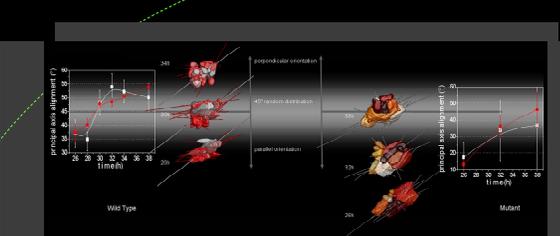
We applied *in vivo* confocal microscopy of GFP-transgenic zebrafish in combination with 3D image analyses to study the asymmetric morphogenesis of the diencephalic parapineal organ on a supra-cellular, cellular, and sub-cellular level. Following a rough manual segmentation of the respective regions of interest (ROIs), the morphology of generated surface meshes was refined by an active surface model which iteratively adjusts the mesh towards the morphology of the cellular structures. This procedure is essential for a precise morpho-topological analysis, mostly because of the adversarial diffraction limited resolution in the z-dimension of confocal image stacks. 3D Morphology and topology of the reconstructed cellular and supra-cellular structures during morphogenesis was quantified by principal axis transformations and 3D moment invariants.

Our data indicates that migration of the parapineal organ is accompanied by a rapid transition between predominantly parallel cell orientations towards predominantly perpendicular orientations, a phenomenon which requires a precise control of cell shape and polarity. The orientational transition is followed by a phase of polarized cell motility in which membrane protrusions in the form of blebs and filopodia become oriented in the direction of the asymmetric migration.

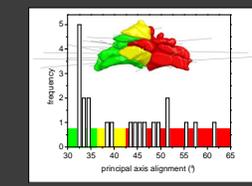
The morpho-topological descriptors unveil information that is not perceptible for a direct visual analysis of the microscopical data sets. This approach becomes essential to access morphogenetic mechanisms which control asymmetry and migration.



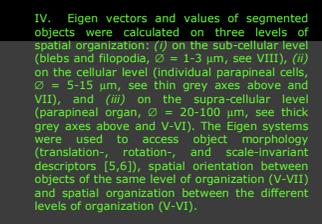
V. Topological reorganization of the parapineal organ during asymmetric morphogenesis. [a-g] Pineal complex (green) and color-coded active surface models of individual cells of the parapineal organ at 26, 28, 30, 32, 34, and 38 hours post-fertilisation. Left-right symmetry axes of the pineal complex are shown by dotted, vertical lines. Thick and thin grey lines depict principal axes of the entire parapineal organ and of individual reconstructed cells, respectively. Abbreviations: anterior (A), posterior (P), left (L), right (R). Scale bar = 20µm.



VI. Principal axes alignment during parapineal morphogenesis of wild type (left) and mutant (right) zebrafish embryos. Alignment between the principal axes of individual parapineal cells (red circles) is compared to the alignment between the 1st principal axis of the individual cells with the 1st principal axis of the entire parapineal organs (white squares). The parameter 'principal axis alignment' represents the mean values calculated from the respective angular distributions in three independent experiments. Mean values were connected by B-spline curves. Error bars represent standard deviations. Representative surface reconstructions of the parapineal organ based on active contours in combination with the respective 1st principal axis of individual parapineal cells and of the entire parapineal organ at 26/30/34 hours (wild type) and 26/32/38 hours (mutant) post-fertilisation open a direct visual access to the data presented in the plots. The data reveals a transition of the organization of parapineal cells from a predominantly parallel orientation towards predominantly perpendicular orientation.



VII. Asymmetric distribution of the axial alignment of parapineal cells during morphogenesis. The histogram plots the frequency distribution of the parameter 'inter cellular alignment of the 1st principal axis' for a representative parapineal cell population at 28 h post fertilisation. Three intervals of the angular alignment were colour coded according to the predominant orientation: parallel (green), random (yellow), and perpendicular (red) (see VI). Colours were projected to the surface of the corresponding parapineal cells.



VIII. Spatial distribution of sub-cellular structures within the parapineal organ of wild type (left) and mutant (right) zebrafish embryos. As can be observed in the upper 3D reconstructions, membrane protrusions in the form of blebs (red), and filopodia (yellow) are oriented in the direction of the asymmetric migration toward the left side of the wild type pineal complex. In contrast, blebs and filopodia are symmetrically oriented in the mutant. Scale bar = 20µm.

Literature

[1] Concha, M.L., Reyes, C., Rojas, J.C., Tami, M., Sol, E., Gribudo, D., Argente, J.C., Gomez, C., Gonzalez, A., Ortega, S., Fernandez, G., Reyes, C., Concha, M., Gonzalez, M., Rojas, J., and Concha, M. (2012). The zebrafish parapineal organ: A novel structure of the diencephalon. *Development* 141, 423-438.

[2] Hirtel, S. (2010). Active Contours in Three Dimensions: From Project Alpha to Computer Vision Laboratory. Leibniz Universität Hannover, 100.

[3] Xu, C., Wang, A., and Teramachi, O. (1995). Surface Active Contour Models. *International Journal of Computer Vision* 5, 215-231.

[4] Xu, C. & Prince, J.L. (1998). Gradient descent vector flow external forces for active contours. *Proceedings of CVPR* 98, 359-366.

[5] Hirtel, S., Reyes, P., Gochtmann, E., Reyes, D., Concha, A., Colla, D.L., Reyes, A.D., & Rojas, J. (2010). Bone morphogenetic protein 30 receptor mutants. *Medical Research Archives* 4, 147-150.

[6] Hirtel, S., Rojas, J., Roth, C., Guzmán, M.L., & Gonzalez, O. (2010). Identification and characterization of the active site of the zebrafish parapineal organ. *Development* 137, 147-150.

[7] Hirtel, S. (2010). A new method for the quantification of topological data in Rainbow Trout (*Oncorhynchus mykiss*). *Revista de Medicina Veterinaria* 51(2): 147-154.

Conclusions

A relatively simple mathematical analysis unveils new biological questions in respect to the first phenomena of asymmetry during parapineal morphogenesis. The rapid access to statistical properties of the axes components on a cellular and a supra-cellular level determine the temporal interval for the orthogonal transition in parapineal cell organization. The back projection of colour coded parameter intervals reveals supra-cellular pattern inside the parapineal organ. Altogether, the analysis detects a morphogenetic asymmetry prior to the lateral migration of the organ (compare 28 & 30 h in V & VI). The detected internal cellular pattern directly leads to the question of the underlying biological mechanisms. We are presently expanding the analysis in all levels of organization and include mutant zebrafish embryos to address the biological implication of the observed phenomena.

**The Initial Surface Composition and Topography Modulate Shpingomyelinase-driven
Shpingomyelin to Ceramide Conversion in Lipid Monolayers**

Luisina De Tullio, Bruno Maggio, Steffen Hartel, Jorge Jara, Maria Laura Fanani

Submitted to Cell Biochemistry and Biophysics in September 2006.

ABSTRACT: Changes of the initial composition and topography of mixed monolayers of Sphingomyelin and Ceramide modulate the degradation of Sphingomyelin by Sphingomyelinase. The presence of initial lateral phase boundary due to coexisting condensed and expanded phase domains favors the precatalytic steps of the reaction. The amount and quality of the domain lateral interface, defined by the type of boundary undulation, appears as a modulatory supramolecular code which regulates the catalytic efficiency of the enzyme. The long range domain lattice structuring is determined by the Sphingomyelinase activity.

Keywords: Phospholipid monolayers; Sphingomyelinase; Epifluorescence microscopy; Sphingomyelin; Ceramide; phase coexistence; laterally segregated domains.