

**UNIVERSIDAD AUSTRAL DE CHILE**  
**Facultad de Ciencias de la Ingeniería**  
**Escuela de Ingeniería Civil en Informática**

# **Segmentación y análisis automatizado de objetos en movimiento aplicado al estudio de Sistemas Biológicos**

Tesis para optar al título de  
**Ingeniero Civil en Informática**

Profesor Patrocinante:  
Sr. Jorge Videla Godoy

Profesor Copatrocinante:  
Sra. Eliana Scheihing García

Profesor Colaborador:  
Dr. Steffen Härtel

María Luisa Oyarzo Vergara  
Valdivia Chile 2003

## **Agradecimientos**

A mi Mamita, gracias por haberme dado la posibilidad de llegar a este punto de mi vida, por haberme apoyado y sobre todo por haber confiado en mi.

A mi Tía, mis Hermanas y Hermanitas, gracias por estar siempre conmigo.

A mis Profesores Eliana Scheihing, Jorge Videla y Steffen Härtel, por el tiempo dedicado y especialmente a Steffen por los aportes realizados al presente trabajo.

A la Sra. Juanita, por toda su ayuda.

A Rodrigo Rojas, por haber aportado en el inicio de este trabajo.

A Renan Saldivar, por tu comprensión y apoyo.

A mis amigos, por la ayuda incondicional brindada.

A mi Angel Guardián.

## INDICE

CAPITULO I.....	11
1. INTRODUCCIÓN.....	11
1.1. NIVEL ACTUAL DE PROCESAMIENTO DE IMÁGENES.....	13
1.2. MOTIVACIÓN DEL TRABAJO.....	17
1.3. OBJETIVOS.....	20
1.4. ORGANIZACIÓN DE LA TESIS.....	23
CAPITULO II.....	24
2. ESTADO DEL ARTE.....	24
2.1. PROCESAMIENTO DIGITAL DE IMÁGENES.....	24
2.1.1. Fundamentos en Procesamiento Digital de Imágenes.....	26
2.1.2. Representación Digital de una Imagen.....	28
2.1.3. Relaciones Entre Píxeles.....	31
2.1.4. Operaciones Básicas Entre Píxeles.....	34
2.1.4.1. Medidas de Distancias.....	34
2.1.4.2. Operaciones Lineales y No Lineales.....	36
2.2. TÉCNICAS DE REDUCCIÓN DE RUIDO Y REALCE DE LA IMAGEN.....	37
2.2.1. Introducción.....	38
2.2.2. Mejoras Mediante Procesamiento Puntual.....	41
2.2.2.1. Ampliación de Contraste.....	41
2.2.2.2. Corte y Umbralización.....	42
2.2.2.3. Negativo Digital.....	43
2.2.2.4. Extracción de Bits.....	43
2.2.2.5. Compresión del Rango.....	43
2.2.2.6. Potenciación.....	44
2.2.2.7. Diferencia de Imágenes.....	44
2.2.2.8. Promedio de Imágenes.....	45
2.2.3. Procesamiento por Histograma.....	46
2.2.3.1. Uniformización del Histograma.....	47
2.2.3.2. Especificación del Histograma.....	50
2.2.3.3. Mejora Local.....	51
2.2.4. Filtrado Espacial.....	53
2.2.4.1. Filtros de Suavizado.....	55
2.2.4.2. Filtros de Realce.....	57
2.2.5. Filtrado en el Dominio de la Frecuencia.....	60
2.2.5.1. Filtros de Paso Bajo.....	61
2.2.5.2. Filtros de Paso Alto.....	63
2.2.6. Transformada de Fourier.....	64
2.2.6.1. Definición de la Transformada de Fourier.....	64
2.2.6.2. Extensión a Dos Dimensiones.....	66
2.2.6.3. Transformada Discreta de Fourier.....	67
2.2.6.4. Transformada Discreta de Fourier en Dos Dimensiones.....	68
2.2.6.5. Transformada Rápida de Fourier.....	69
2.3. TÉCNICAS DE SEGMENTACIÓN.....	74
2.3.1. Regiones y Bordos.....	74
2.3.2. Segmentación Basada en Fronteras.....	77
2.3.2.1. Métodos de Gradiente.....	79
2.3.2.2.1. Operadores Isotrópicos.....	79
2.3.2.2.2. Operador de Roberts.....	81
2.3.2.2.3. Operador de Sobel.....	81
2.3.2.2.4. Operador Prewitt.....	82
2.3.2.1. Pasos en la Detección de Bordos.....	82
2.3.2.2. Operadores de Segunda Derivada.....	83
2.3.2.2.1. Operador Laplaciano.....	83

2.3.2.2.2.	Cruces por Cero de la Segunda Derivada. Laplaciana de Gaussiana.....	85
2.3.3.	Segmentación Basada en Bordes.....	87
2.3.3.1.	Relajación de Bordes.....	87
2.3.3.2.	Extracción de Frontera.....	92
2.3.4.	Segmentación Orientada a Regiones.....	94
2.3.4.1.	Unión de Regiones.....	94
2.3.4.2.	Unión de Regiones por Agregación de Píxeles.....	95
2.3.4.3.	Separación y Unión de Regiones.....	96
2.3.5.	Transformada Wavelets Discreta.....	99
2.3.5.1.	Descomposición de Wavelets.....	100
2.3.5.2.	Combinación de Canales.....	101
2.3.5.3.	Cuantización.....	102
2.4.	COMPARANDO WAVELETS VERSUS FOURIER.....	104
CAPITULO III.....		106
3.	METODOLOGÍA.....	106
3.1.	INTRODUCCIÓN.....	106
3.2.	ADQUISICIÓN DE LA IMAGEN.....	109
3.3.	REDUCCIÓN DE RUIDO.....	110
3.4.	CÁLCULO DE LOS IP UTILIZANDO LA TRANSFORMADA DISCRETA DE WAVELETS.....	111
3.4.1.	Introducción.....	111
3.4.2.	Implementación y Código Fuente de la Transformada Discreta de Wavelets.....	115
3.4.3.	Aplicación para la Obtención de los Coeficientes Wavelets Utilizando Transformada Discreta de Wavelets.....	117
3.4.4.	Resultados Obtenidos al Aplicar la Transformada Discreta de Wavelets.....	119
3.4.5.	Implementación y Código Fuente para la Extracción de los Puntos de Interés a partir de los Coeficientes Wavelets.....	122
3.4.6.	Resultados Obtenidos al Aplicar Puntos de Interés.....	127
3.5.	CÁLCULO DE LOS VECTORES DE DESPLAZAMIENTO UTILIZANDO OPTICAL FLOW.....	128
3.5.1.	Introducción.....	129
3.5.2.	Ecuación de Restricción de Optical Flow.....	131
3.5.3.	Implementación y Código Fuente del Algoritmo Optical Flow.....	133
3.5.4.	Aplicación para la Obtención del Algoritmo Optical Flow.....	136
3.5.5.	Resultados Obtenidos al Aplicar el Algoritmo Optical Flow.....	137
3.6.	SEGMENTACIÓN POR MOVIMIENTO.....	138
3.7.	EXTRACCIÓN DE PARÁMETROS DE INTERÉS.....	141
3.8.	ANÁLISIS GRÁFICO DE LA INFORMACIÓN.....	143
3.9.	EXPORTACIÓN DE LA INFORMACIÓN.....	144
3.10.	COMPARACIÓN RESULTADOS.....	145
3.11.	TRABAJOS FUTUROS.....	147
CAPITULO IV.....		148
4.	CONCLUSIONES.....	148
5.	BIBLIOGRAFÍA.....	151
Anexo A: Análisis de Imágenes con el Método Propuesto.....		154
Anexo B: Análisis Interactivo de los Objetos.....		159
Anexo C: Imágenes sección 3.6.....		160
Anexo D: Imágenes sección 3.7.....		164

## INDICE DE IMAGENES

Figura 2.1:	Representación digital de la imagen.....	30
Figura 2.2:	Vecindad de 3x3 alrededor de un punto (x,y) en una imagen. ....	38
Figura 2.3:	Efecto de aplicar Transformaciones. ....	40
Figura 2.4:	Ampliación de contraste. ....	41
Figura 2.5:	Ejemplos de Histogramas.....	46
Figura 2.6:	Función de transformación de niveles de gris que es univaluada y monótonamente creciente. ....	47
Figura 2.7:	Representación general de una máscara 3x3 de filtro espacial, donde $w_5$ es el origen. ....	54
Figura 2.8:	Dos máscaras de filtros de suavizado, la constante multiplicativa es igual a la suma de los valores de sus coeficientes, como se requiere para calcular un promedio. ....	55
Figura 2.9:	Ejemplo filtro de paso alto. ....	57
Figura 2.10:	Ejemplo filtro de paso alto. ....	59
Figura 2.11:	Etiquetado de niveles de gris para explicar los detectores de bordes.....	82
Figura 2.12:	a) Intensidad, b) Primera Derivada, c) Segunda Derivada. ....	85
Figura 2.13:	Borde tipo grieta e con sus vecinos.....	88
Figura 2.14:	Borde tipo grieta e con sus vecinos.....	89
Figura 2.15:	Ejemplo de seguimiento de contorno.....	92
Figura 2.16:	Ejemplo de crecimiento de regiones a partir de puntos semilla.....	96
Figura 2.17:	Ejemplo de quad-tree. ....	97
Figura 2.18:	Proceso de separación y unión usando quad-trees.....	98
Figura 2.19:	Esquema de los coeficientes Wavelet. ....	99
Figura 2.20:	Descomposición Wavelet, primer nivel.....	100
Figura 2.21:	Descomposición Wavelet, Tres niveles. ....	101
Figura 2.22:	Bandas Wavelet.....	102
Figura 2.23:	Extracción de los Patrones de Frecuencia de la señal de la imagen. ....	103
Figura 2.24:	Comparación de transformaciones, a la izquierda Hessian y a la derecha Wavelets [HAR01].....	105
Figura 3.1:	a) IO. b) Resultados Haar Wavelet.....	<b>¡Error! Marcador no definido.</b>
Figura 3.2:	Método de Harris, b) Método de Haar Wavelets, c) Método de Debauchie Wavelets, d) Método Multi Resolución Basado en Contraste [HAR01], [OHM99]. ....	113
Figura 3.3:	Comparación de transformaciones, la imagen izquierda corresponde a Hessian y a la derecha Wavelets [HAR01]. ....	114
Figura 3.4:	Coeficientes Wavelet buscados [OHM99]. ....	115
Figura 3.5:	Convención utilizada para calcular los coeficientes Wavelet [OHM99]. .	115
Figura 3.6:	a) Interfase de ingreso de imágenes. ,b). Menú de ingreso de imágenes y manejo de archivos, c). Menú de análisis que abre la Interfase de Segmentación, d) Interfase de segmentación. e) Ingreso de parámetros para la transformada Wavelets. ....	118
Figura 3.7:	a) Imagen de Lena original, b). Resultado al aplicar Wavelets.....	119

Figura 3.8:	a) Imagen Hela original, b). Resultado al aplicar Wavelets. ....	<b>¡Error! Marcador no definido.</b>
Figura 3.9:	a) Imagen Hela selección, b). Resultado al aplicar Wavelets. ....	<b>¡Error! Marcador no definido.</b>
Figura 3.10:	a) Imagen Hela selección, b). Resultado al aplicar Wavelets, con zoon (3×). ....	121
Figura 3.11:	Sumas sobre las bandas Wavelets Escala invariante [OHM99]. ....	123
Figura 3.12:	Análisis de los IP por medio de la suma de los Coeficientes Wavelet [OHM99]. ....	124
Figura 3.13:	a) Imagen de Lena original, b). Resultado al aplicar IP a partir de los coeficientes Wavelet. ....	<b>¡Error! Marcador no definido.</b>
Figura 3.14:	a) Imagen Hela, b). Resultado al aplicar IP a partir de los coeficientes Wavelet. ....	<b>¡Error! Marcador no definido.</b>
Figura 3.15:	Se muestra un cuadro en su posición 1, luego el cuadro en su posición 2 y finalmente los VD [BEA99]. ....	128
Figura 3.16:	a) Imagen de la red coronaria, b). Resultado de la sustracción de las imágenes. ....	129
Figura 3.17:	a) Primera imagen secuencia 20 del set Hela, b) Segunda imagen secuencia 21 del Hela, c). Resultado de la diferencia de las imágenes a) y b). ....	130
Figura 3.18:	Convención a utilizar en el cálculo del OF. ....	133
Figura 3.19:	a) Se agrega en la interfase de segmentación la transformada OF, b) Ingreso de parámetros para la transformada OF. ....	136
Figura 3.20:	a) Imagen Hela20, b) Imagen Hela21, c) Resultado OF, d) Resultado VD. ....	137
Figura 3.21:	a) Secuencia de Transformaciones aplicadas sobre la imagen, b) Imagen 20 Hela a Segmentar, c) IP, d) Objetos de mayor interés, e) VD. ....	139
Figura 3.22:	Menú interfase de segmentación, Se está mostrando Aplicar las transformaciones a todas las imágenes. ....	140
Figura 3.23:	a) Interfases de ROI, muestran los parámetros que se pueden analizar, a) Parámetros de tamaños y b) Parámetros de perímetro. ....	141
Figura 3.24:	Muestra de la extracción de parámetros de interés, a) Transformaciones aplicadas sobre la IO, b) Resultado de aplicar a), c) Histograma de b). ....	142
Figura 3.25:	a) Transformaciones aplicadas sobre la IO, b) Resultado de aplicar a), c) Histograma de b). ....	143
Figura 3.26:	Muestra el resultado de algunas imágenes de la secuencia analizada. ....	144
Figura 3.27:	Esquema que se utilizará en la comparación de resultados. ....	145
Figura A.1:	a) Muestra Hela21, b) Muestra Hela30. ....	154
Figura A.2:	Fig. A.1: Muestra resultado de S1.1. ....	154
Figura A.3:	Muestra resultado de S1.2. ....	155
Figura A.4:	Muestra resultado de S2.1. ....	155
Figura A.5:	Muestra resultado de S2.2. ....	156
Figura A.6:	Muestra resultado de S2.3. ....	156
Figura A.7:	Muestra resultado de S2.4. ....	157
Figura A.8:	Muestra resultado de S3. ....	157

Figura A.9: Muestra resultado de S4.....	158
Figura A.10: Muestra resultado de S5.....	158
Figura B.1: Análisis interactivo de los resultados.....	159
Figura C.2: a) OF y b) VD Hela20 .....	160
Figura C.3: a) OF y b) VD Hela21 .....	160
Figura C.4: a) OF y b) VD Hela22 .....	161
Figura C.5: a) OF y b) VD Hela23 .....	161
Figura C.6: a) OF y b) VD Hela24 .....	161
Figura C.7: a) OF y b) VD Hela25 .....	162
Figura C.8: a) OF y b) VD Hela26 .....	162
Figura C.9: a) OF y b) VD Hela27 .....	162
Figura C.10: a) OF y b) VD Hela28 .....	163
Figura C.11: a) OF y b) VD Hela29 .....	163
Figura D.1: Transformaciones aplicadas a Hela29.....	164
Figura D.2: Hela29 bajo OF y Threshold.....	164
Figura D.3: Parámetro Object Size (Pixel <sup>2</sup> ) .....	165
Figura D.4: Parámetro Object Perimeter (x).....	165
Figura D.5: Parámetro Object Perimeter (Pixel).....	165

## RESUMEN

El perfeccionamiento de las técnicas de procesamiento de imágenes en las últimas décadas ha permitido extender los métodos tradicionales a todos los ámbitos tecnológicos actuales. En el campo de la investigación biológica el aporte del procesamiento de imágenes a sido particularmente bienvenido como herramienta para el estudio del desarrollo de sistemas biológicos microscópicos en el tiempo, tanto por su habilidad de síntesis como por su capacidad de extraer información compleja a través del procesamiento de parámetros obtenidos directamente de imágenes.

Sin embargo, a pesar de las numerosas ventajas de este tipo de software, existe un vacío de aplicaciones que permitan extraer información acerca del comportamiento de un sistema biológico a partir del grado de actividad de sus componentes (células), reflejado en la magnitud de sus desplazamientos y la relación con sus similares, así como también realizar un estudio particular de individuos.

En éste sentido, el presente trabajo tiene como objetivo abordar el vacío existente a través del desarrollo de un conjunto de herramientas de software que permitan, Individualizar (segmentar) componentes en una secuencia de imágenes utilizando como principal parámetro de discriminación el movimiento que tales individuos presenten, y utilizar las habilidades del procesamiento informático de datos para a partir de tales individuos extraer información particular o general acerca de su actividad.

## **SUMMARY**

The improvement of images processing techniques in the last decades has allowed extending the traditional methods to all the current technological environments. In the field of biological investigation, the contribution of images processing has been particularly received as a tool for the study of the development of microscopic biological systems in the time, so much for their synthesis ability like for their capacity to extract complex information through the processing of parameters obtained directly of images.

However, in spite of the numerous advantages of this kind of software, it exists a hole of applications that allow to extract information about the behavior of a biological system starting from the degree of activity of its components (cells), reflected in the magnitude of its displacements and the relationship with its similar ones, as well as to carry out a particular study of individuals.

In this sense, the present work has as an objective to approach the existent hole through the development of a group of software tools that allow to individualize (to segment) components in a sequence of images using as a main discrimination parameter the movement that such individuals present, and to use the abilities of the computer processing of data for starting from such individuals to extract particular or general information about their activity.

## **NOMENCLATURA**

CECS	:	Centro de Estudios Científicos
EM	:	Electromagnético
FEAF	:	Filtro de Enfasis de Altas Frecuencias
HSV	:	Hue Saturation Value
IDL	:	Interactive Data Language
IO	:	Imagen Original
IP	:	Puntos de Interés o Interest Points
JPEG	:	Join Photographics Experts Group
MPEG	:	Moving Picture Experts Group
PDI	:	Procesamiento Digital de Imágenes
RGB	:	Red Green Blue
TF	:	Transformada de Fourier
FFT	:	Fast Fourier Transform
DWT	:	Discret Wavelets Transform
OF	:	Optical Flow
VD	:	Vectores de Desplazamiento
ROI	:	Region Of Interest

# **CAPITULO I.**

## **1. INTRODUCCIÓN.**

La historia del Procesamiento Digital de Imágenes (PDI) está estrechamente ligada al desarrollo de los computadores, las imágenes digitales requieren gran capacidad de almacenamiento y poder de cálculo, el progreso del PDI ha sido dependiente del desarrollo de los computadores y de las tecnologías asociadas al almacenamiento, despliegue y transmisión de la información.

Las técnicas de PDI se utilizan hoy en día en una variedad de problemas muy diversos, estos comparten una necesidad común, ser capaces de realzar la información ilustrada para la interpretación y el análisis humanos.

El PDI es el mecanismo mediante el cual se puede extraer información del mundo real de manera automática a partir de una imagen o secuencia de imágenes dinámicas, dicha información podría ser el reconocimiento de objetos genéricos y descripciones del mismo. La extracción de información de una imagen se realiza aplicando técnicas de segmentación de imágenes.

La segmentación de imágenes consiste básicamente en agrupar regiones visuales en términos de proximidad, similitud y continuidad, para construir un conjunto de píxeles, regiones o contornos que muestran o disciernen una similitud en cuanto a intensidad, color, textura, movimiento, u otra característica de interés. Al utilizar los conceptos de segmentación lo que se logra es distinguir, separar o extraer objetos de su fondo o background.

Las imágenes usualmente se representan como conjuntos de píxeles que codifican información de color y brillo en forma de una matriz. Esta matriz se obtiene transformando la imagen o frame en tres matrices que la describen (transformación RGB) cada una de éstas matrices contiene el nivel de Rojo,

Verde y Azul presente en la imagen, luego mediante operaciones matemáticas entre éstas matrices se obtiene una matriz que representa a la imagen en sus niveles de grises, ésta matriz es conocida como la Matriz de Intensidades.

## **1.1. Nivel Actual de Procesamiento de Imágenes.**

Desde los años sesenta hasta el presente el campo de PDI ha crecido considerablemente. Las áreas donde se aplica son muy variadas, como por ejemplo en medicina y programas espaciales, las técnicas de PDI son utilizadas en un amplio rango de aplicaciones. Estas técnicas son usadas para realzar el contraste o para cifrar los niveles de intensidad en el color de una imagen facilitando así la interpretación de rayos X u otras imágenes utilizadas en la industria, medicina, y ciencias biológicas. Los Geógrafos utilizan éstas técnicas o similares para estudiar los patrones de contaminación desde imágenes aéreas y satelitales. Procedimientos de realce y restauración de imágenes se utilizan para procesar imágenes degradadas de objetos irrecuperables o de resultados experimentales demasiado costosos de duplicar. En Arqueología, métodos de PDI han restaurado con éxito los cuadros borrosos que eran los únicos expedientes disponibles de artefactos raros perdidos o dañados después de ser fotografiados. En física y campos relacionados, las técnicas por computador realzan rutinariamente imágenes de experimentos en áreas tales como plasmas de alta energía y microscopio electrónico. Análogamente aplicaciones exitosas se pueden encontrar en astronomía, biología, medicina nuclear, defensa, aplicaciones industriales, etc.

En el presente trabajo el interés se centra en los procedimientos para extraer información desde una imagen en forma adecuada para su posterior procesamiento. Frecuentemente ésta información tiene poca semejanza con las características visuales que el humano utiliza para interpretar el contenido de una imagen. Ejemplos del tipo de información usados por la percepción de la máquina son momentos estadísticos, coeficientes de la transformada de Fourier, medidas multidimensionales de distancia, etc. Los problemas típicos de

percepción de la máquina que utiliza rutinariamente técnicas de PDI son reconocimiento automático de caracteres, visión de máquinas industriales para inspección y ensamble de productos, procesamiento automático de huellas digitales, investigación de muestras de sangre y rayos X, procesamiento de imágenes aéreas y satelitales para predicción del tiempo y valoración ambiental. La continua disminución en los precios de los computadores, el alto rendimiento y la expansión de las redes además del ancho de banda de las comunicaciones e Internet han creado oportunidades sin precedentes para incrementar el crecimiento del PDI.

Hoy en día casi no hay áreas de esfuerzo técnico en que no se vean impactadas de alguna forma por el PDI. Como se ha dicho son numerosas y muy variadas las áreas de aplicación del PDI, una de las formas más fáciles de simplificar el desarrollo de un entendimiento hasta el punto de aplicaciones de procesamiento de imágenes es catalogar las imágenes dependiendo de su origen, el principal origen utilizado es el Espectro de Energía Electromagnética (El rango completo de longitudes de onda es conocido como "Espectro Electromagnético"), otras fuentes importantes de energía incluyen la acústica, ultrasónica y electrónica, ésta última se refiere a la forma en que el electrón brilla en el microscopio electrónico.

Las imágenes sintéticas usadas para modelamiento y visualización son generadas por computador. Los métodos que se utilizan para convertir imágenes en sus formas digitales se explican en el siguiente capítulo.

Las imágenes basadas en la radiación del Espectro Electromagnético (EM) son las más familiares, especialmente imágenes en radiografía y bandas visuales del espectro. Las ondas electromagnéticas pueden ser conceptuadas como ondas sinusoidales que se propagan en longitudes de onda que varían, o pueden ser pensadas como un flujo de partículas sin masa que viajan en

mismos patrones de ondas y que se mueven a la velocidad de la luz, cada partícula sin masa contiene una cierta cantidad de energía.

Las aplicaciones más importantes basadas en rayos gamma incluyen medicina nuclear y observaciones astronómicas. El principal uso de los rayos x es aplicado a diagnósticos médicos, pero también son ampliamente usados en la industria y en áreas como astronomía. Las aplicaciones en bandas de Luz Ultravioleta son variadas, incluyen litografía, inspección industrial, microscopia, láser, imágenes biológicas, observaciones astronómicas, la Luz Ultravioleta es usada en microscopia de fluorescencia. La principal aplicación de imágenes en bandas de microondas es en radar, la principal característica de las imágenes de radar es su capacidad de recolectar datos sobre cualquier región virtual en un tiempo dado, sin importar las condiciones del tiempo o las condiciones ambientales de la luz. En el área de bandas de radio son importantes las aplicaciones en medicina y astronomía, en medicina las ondas de radio son utilizadas en imágenes de resonancia magnética. Aplicaciones de bandas infrarrojas son utilizadas en microscopia, astronomía, detección remota e industria. Existen también otras modalidades en las se obtienen imágenes, como por ejemplo, imágenes acústicas, imágenes sintéticas (generadas por computados en forma artificial) e imágenes extraídas por Microscopio Electrónico. Imágenes que se obtienen utilizando sonido pertenecen a campos de exploración tales como geológico, industrial y médico. Las aplicaciones geológicas utilizan el extremo inferior del espectro EM (cientos de hertz) mientras que en otras áreas se utiliza el ultrasonido (millones de hertz). La aplicación más importante de procesamiento de imágenes en geología es la exploración de minerales y petróleo.

Por otro lado las imágenes sintéticas generadas por computador son llamados Fractales. Los fractales son imágenes geométricas. Básicamente, un fractal es

una reproducción iterativa de un patrón básico según algunas reglas matemáticas, un enfoque más estructurado de generación de imágenes por computador “no reales” es en modelamiento en 3-D, ésta es un área que proporciona una importante intersección entre procesamiento de imágenes y gráficos por computador y es la base de muchos de los sistemas de visualización 3-D.

En el área óptica, el Microscopio electrónico funciona como su contraparte óptico, utiliza un haz enfocado de electrones en vez de luz sobre la muestra. El Microscopio Confocal Permite que sólo observemos el plano que está situado en el punto de foco del sistema óptico, eliminando de forma óptica a través de un diafragma la luz proveniente de los planos que están fuera de foco, con esto se reduce el ruido en el momento de la captura de las imágenes, las imágenes utilizadas en el presente trabajo fueron y serán obtenidas por éste medio, en el Centro de Estudios Científicos CECS, y en la Universidad de Córdoba.

En la actualidad, existen pequeños grupos altamente especializados en el tema de la manipulación de imágenes, siendo el de mayor importancia por su aporte a las técnicas de análisis el dirigido por el profesor Ian Young en Holanda. Instituto Max-Planck, Múnich, Alemania. Existen muchos grupos dedicados al tema siendo reconocidos en América Latina dos, uno en Uruguay y otro en Argentina, por lo que el aporte al tema será fundamental para potenciales grupos que se formen en el país [BAR2002].

## 1.2. Motivación del Trabajo.

En el campo de la investigación biológica el aporte del procesamiento de imágenes a sido particularmente bienvenido como herramienta para el estudio del desarrollo de sistemas biológicos microscópicos en el tiempo, tanto por su habilidad de síntesis como por su capacidad de extraer información compleja a través del procesamiento de parámetros obtenidos directamente de imágenes.

Sin embargo, a pesar de las numerosas ventajas de éste tipo de software, existe un vacío de aplicaciones que permitan extraer información acerca del comportamiento de un sistema biológico a partir del grado de actividad de sus componentes (células), reflejado en la magnitud de sus desplazamientos y la relación con sus similares (otras células), así como también realizar un estudio particular de individuos.

Estos vacíos serán abordados a través del desarrollo de un conjunto de herramientas de software que permitan:

- Individualizar (segmentar) componentes en una secuencia de imágenes (células u otros elementos de interés) utilizando como principal parámetro de discriminación el movimiento que tales individuos presenten.
- Utilizar las habilidades del procesamiento informático de datos para a partir de tales individuos extraer información particular o general acerca de su actividad (respecto a si u otros elementos).

Como se dijo antes segmentación de imágenes consiste básicamente en agrupar regiones u objetos contenidos en una imagen, la imagen es transformada en una matriz de Intensidades función  $I(x,y)$  (ver sección 2.1.2). Al utilizar los conceptos de segmentación lo que se logra es distinguir, separar o extraer objetos de su fondo o background. Estos conceptos se encargan de evaluar si cada píxel de la imagen pertenece o no a un objeto de interés. Con lo

que se genera una imagen binaria, donde los píxeles que pertenecen al objeto se representan con un 'uno', mientras que los que no pertenecen se representan con un 'cero'. Las técnicas de segmentación más utilizadas son: Segmentación por Píxeles, Segmentación por Contornos y Segmentación por Regiones (Ver sección 2.3). Una vez identificado el objeto de interés se aplicará la hipótesis de movimiento, cuando se habla de movimiento la primera idea es asociarlo con cambios en la intensidad de la imagen.

La hipótesis de movimiento consiste en que la imagen está compuesta de muchos objetos puestos en la escena, se estudiará cuál es la dirección en que se mueven, la suposición básica es que los objetos se mueven sólo traslacionalmente, se estudiarán entonces los llamados Vectores de Desplazamiento (VD) utilizando técnicas de Flujo Óptico (Optical Flow u OF) para determinar el movimiento de los píxeles de un frame a otro. En cada frame los objetos tienden a confundirse o mezclarse unos con otros, con la ayuda de los VD se puede determinar que uno o más grupos de píxeles se están moviendo en determinada dirección y que cada grupo compone un objeto, en cada frame existirá otro grupo de píxeles que carece de movimiento y que conforman el fondo o background. Con lo anterior el sistema será capaz de realizar el seguimiento de objetos a través de las secuencias de imágenes.

Esto se logrará en tres niveles:

**Nivel Básico:**

- La imagen digital se transforma en una matriz en niveles de grises.
- Se aplican filtros de eliminación de ruido o suavizado de la imagen.

**Nivel Medio:**

- Se aplican los filtros de segmentación, se aplicará Segmentación por Contornos, utilizando Wavelets para la extracción de los puntos de interés (Interest Points o IP).

- Se aplican los métodos para obtener los VD (Flujo Óptico) a los IP obtenidos en la etapa anterior.
- Se realizará la segmentación de objetos aplicando la hipótesis de movimiento.

**Nivel Alto:**

- Se aplicarán filtros para obtener la extracción y análisis de los parámetros de interés contenidos en las secuencias de imágenes.
- Se permitirá la exportación de ésta información en forma de archivos de texto o imágenes.

En el área de procesamiento de imágenes existen muchos grupos de desarrollo, pero las aplicaciones son específicas. Lo interesante de este proyecto es su versatilidad, las imágenes a las que se aplicará el estudio son sistemas biológicos (células), pero será posible analizar cualquier tipo de imagen. Es destacable que más allá de la aplicación particular al ámbito de la biología celular, el software descrito es en gran medida aplicable lo que se conoce como 'seguimiento de objetos' cuyo ámbito de usos es extremadamente amplio.

### 1.3. Objetivos.

Lo que se quiere lograr entonces es desarrollar un sistema de procesamiento y análisis robusto, capaz de reconocer objetos contenidos en secuencias de imágenes digitalizadas, esto es segmentación de objetos de interés, con lo que se podrá llevar a cabo un análisis más completo de las imágenes en estudio y una mejor representación de los datos contenidos en ellas, además se quiere lograr una herramienta genérica capaz de realizar el análisis de cualquier tipo de imágenes, con el fin de estudiar el desarrollo en el tiempo de individuos particulares y del sistema en general.

Esto se llevará a cabo con el desarrollo de módulos que realicen las tareas que nos llevarán a cumplir el objetivo principal, éstas son:

- Desarrollo de algoritmos necesarios para convertir las secuencias de imágenes en estructuras de datos: La imagen ingresará como archivo TIFF (una imagen individual o secuencias de imágenes) en escala de grises, entonces se almacena la imagen digitalizada en una matriz  $M \times N$ , obteniendo así la matriz de datos a la que se denominará Matriz de Intensidades  $I(x,y)$  que representa la imagen en sus niveles de grises.
- Desarrollo de algoritmos que realicen la tarea de eliminar el ruido en la Matriz de Intensidad  $I(x,y)$ : Lo que se requiere es obtener una estructura de datos libre de ruido, que permita diferenciar claramente a cada objeto de la imagen y poder utilizar esta información en las siguientes etapas.
- Desarrollo de algoritmos que lleven a cabo la tarea de obtener los IP desde  $I(x,y)$ : A partir de la estructura de datos libre de ruido se requiere localizar los IP, éstos son conjuntos de píxeles o áreas que conforman los posibles objetos en las imágenes, para esto se implementarán los métodos que permitan detectar los Puntos de mayor Interés contenidos en la imagen, esto

entrega como resultado una imagen que contiene la representación de los contornos o bordes de las áreas de los objetos a segmentar. Se aplicará principalmente el método de los Wavelets, adicionalmente también se implementarán otros métodos para verificar y comparar los resultados.

- Desarrollo de algoritmos que permitan obtener los VD de los IP, aplicando las técnicas de Flujo Óptico: Una vez definidos los IP, se estudiará el movimiento de los píxeles de un frame a otro, en cada frame los objetos tienden a confundirse o mezclarse con otros, con la ayuda de los VD se puede determinar que uno o más grupos de píxeles se están moviendo en determinada dirección y que cada grupo conforma un objeto. Se aplicará el método del Flujo Óptico.
- Desarrollo de algoritmos que permitan realizar la Segmentación de Objetos por Movimiento: Una vez que se obtiene la secuencia de imágenes donde se pueden encontrar las áreas de objetos que componen cada frame y sus VD, se hará uso de la hipótesis de movimiento lo que nos entregará como resultado la segmentación de objetos en el tiempo, es decir, se obtendrán las áreas de datos que se están moviendo en determinada dirección a través de la secuencia de imágenes.
- Desarrollo de algoritmos que permitan la extracción de los parámetros de interés de los Objetos Segmentados: Se necesitará extraer las características de interés de los objetos segmentados en el punto anterior, éstas características pueden ser por ejemplo parámetros de forma, excentricidad, intensidad, color, textura o movimiento.
- Desarrollo de algoritmos que permitan realizar el análisis gráfico y reportes de los parámetros obtenidos: Con los parámetros obtenidos, la forma más sencilla de representar la información es aplicando estadísticas para visualizar los resultados de manera eficiente y rápida, esto se puede realizar

con la generación de gráficos y reportes para un mejor entendimiento de la información.

- Desarrollo de algoritmos que permitan realizar la exportación de la información obtenida: Finalmente a partir de la información obtenida se requerirá realizar otro tipo de análisis, por lo que se permitirá exportar la información de los objetos segmentados en el tiempo en forma de secuencias de imágenes o archivos de texto.

## **1.4. Organización de la Tesis.**

La descripción de las técnicas involucradas y de los resultados obtenidos se ha organizado de la siguiente manera. En el capítulo 2, se revisará el estado del arte de PDI, en la sección 2.1 se introducen los diversos conceptos relacionados con PDI, en la sección 2.2 se describen las técnicas de reducción de ruido y realce de imágenes, en la sección 2.3 se describen las técnicas de segmentación, también se definen las transformaciones más utilizadas en PDI. En el capítulo 3, se describe la metodología que se utilizó para desarrollar el presente trabajo, también se analizan y discuten los resultados obtenidos y se plantean los trabajos futuros. Y por último, en el capítulo 4 están contenidas las conclusiones obtenidas.

## **CAPITULO II.**

### **2. ESTADO DEL ARTE.**

#### **2.1. Procesamiento Digital de Imágenes.**

La visión es el sentido más desarrollado del humano, es por eso que las imágenes juegan un papel importante en la percepción humana, sin embargo, distinto a los humanos, quienes están limitados a las bandas visuales del Espectro Electromagnético, las máquinas cubren casi por completo el Espectro Electromagnético, en el rango de las ondas gamma y de radio. Las máquinas pueden operar sobre imágenes generadas por orígenes que los humanos no están acostumbrados asociar con imágenes. Estas imágenes pueden ser obtenidas por ultrasonido, microscopio electrónico, imágenes generadas por computador entre otras.

El área de PDI se refiere a Procesamiento de Imágenes por medio de un computador.

El límite entre el procesamiento de imágenes y otras áreas como análisis de imágenes y visión por computador no está bien definido por los autores, se dice como definición que el procesamiento de imágenes es una disciplina en la que entrada y la salida de un proceso son imágenes. Se cree que esto es una limitante y que el límite es artificial. El área de análisis de imágenes (llamada también de entendimiento) está entre el procesamiento de imágenes y Visión por Computador.

No hay límites claros en la serie continua desde procesamiento de imágenes por un lado y Visión por Computador por otro. Sin embargo un útil paradigma es considerar tres tipos de procesos en la serie continua: procesos de bajo nivel, medio y alto.

- Nivel Bajo: Se refiere a los procesos que requieren operaciones básicas, tales como preprocesado de una imagen para reducir ruido, realce del contraste, contorneado de una imagen, entre otros. Este nivel se caracteriza porque su entrada y salida son imágenes.
- Nivel Medio: Involucra tareas tales como Segmentación, esto es encontrar las regiones u objetos que conforman la imagen, obtener la descripción de éstos objetos, obtener una forma reducida de la información de la imagen para disminuir así el tiempo de cálculo, además en éste nivel también se obtiene la clasificación o reconocimiento de los objetos individuales. Los procesos de nivel medio se caracterizan porque sus entradas generalmente son imágenes, pero sus salidas son atributos extraídos de las imágenes originales, por ejemplo contornos, bordes, y la identificación de objetos individuales o conjuntos de ellos.
- Nivel Alto: Involucra procesos de análisis de la imagen, extracción de información a partir de los objetos obtenidos, y exportación de información, con ésta información se pueden generar salidas gráficas, visuales (imágenes), y en formato de texto.

Entonces se deduce que el procesamiento de imágenes y análisis de imágenes se traslapan en el área de reconocimiento de regiones individuales u objetos contenidos en una imagen. Entonces se llama PDI a los procesos donde entrada y salida son imágenes, además de los procesos que extraen las cualidades de la imagen, incluyendo el reconocimiento de objetos individuales. PDI como se ha definido se utiliza en una amplia gama de áreas como se vio en la sección 1.1.

### **2.1.1. Fundamentos en Procesamiento Digital de Imágenes.**

En PDI es útil hacer una diferenciación entre el formato de las entradas y salidas de los procesos, por un lado se encuentran los métodos donde las entradas y salidas son imágenes y por otro están los métodos donde las entradas pueden ser imágenes pero las salidas son características extraídas de las imágenes. A continuación se explican brevemente los diferentes procesos utilizados en Procesamiento de imágenes, no necesariamente se aplican todos los procesos a las imágenes, dependiendo de lo que se quiera obtener se aplican los más adecuados de acuerdo a los objetivos planteados.

#### **Las salidas son imágenes:**

- Obtención de la imagen: La obtención de la imagen puede ser algo tan simple como que la imagen ya este en formato digital, usualmente ésta etapa requiere de un preprocesado de la imagen, como por ejemplo el escalamiento.
- Realce de la Imagen: Consiste en destacar o poner en evidencia ciertas características de interés en la imagen, por ejemplo, es lo que se obtiene al incrementar el contraste de una imagen.
- Restauración de la Imagen: También busca mejorar la imagen, éste proceso se basa en modelos matemáticos y probabilísticos de degradación de la imagen.
- Procesamiento de Imágenes por Color: El color se utiliza como base para extraer las características de interés en una imagen.
- Wavelets: Son la base para la representación de imágenes en varios grados de resolución.

- Compresión: Este proceso comprime la imagen, disminuyendo así el espacio requerido para su almacenamiento y posible transmisión, un ejemplo muy utilizado es el estándar en compresión de imágenes JPEG (Join Photographics Experts Group).

**Las salidas son atributos de la imagen:**

- Procesamiento Morfológico: Engloba las herramientas que extraen componentes de la imagen, que son utilizados en la representación y descripción de forma.
- Segmentación: Consiste en agrupar regiones u objetos contenidos en la imagen.
- Descripción y Representación: Convierte los datos (generalmente obtenido de un proceso de segmentación) a una forma adecuada para el procesamiento por computador. Descripción es llamada “selección de características”, se refiere a la extracción de atributos que son básicos para diferenciar a una clase de objetos de otros.
- Reconocimientos: Es el proceso que asigna etiquetas (label) a un objeto basado en su Descripción.

### **2.1.2. Representación Digital de una Imagen.**

Cuando se adquiere una imagen, los tipos de éstas en los que está puesto el interés son generadas por una combinación de fuentes de iluminación y reflexión o absorción de la energía de esas fuentes por los elementos (objetos) de la escena, éstas fuentes de iluminación pueden generarse a partir de energía electromagnética, tales como energía de radares, infrarrojos o rayos X (Ver sección 1.1). Entonces la energía de iluminación se transforma en imágenes digitales, es decir, la energía entra en el mecanismo de captura (sensor) y es transformada en voltaje por medio de la combinación del poder eléctrico y el material del sensor, que responde al tipo de energía que está siendo detectada. El voltaje en forma de onda es la salida del sensor y una cantidad digital es obtenida del sensor digitalizando su respuesta.

Las imágenes pueden ser definidas como una función bi-dimensional,  $I(x,y)$ , donde  $x$  e  $y$  son coordenadas espaciales, y la evaluación de  $I$  en un par de coordenadas  $(x,y)$  representa a Intensidad o niveles de grises de la imagen en el punto elegido. Cuando los valores de  $x$ ,  $y$  y la amplitud de  $I$  son todos finitos, cantidades discretas, se llama a la imagen Imagen Digital.

Como se acaba de definir, las imágenes son representadas por funciones  $I(x,y)$  donde el rango de valores pertenece a la escala de grises. El valor de  $I$  en las coordenadas espaciales  $(x,y)$  es una cantidad escalar positiva, las imágenes que serán utilizadas en el presente trabajo serán imágenes que ya se encuentran en escala de grises (Ver sección 3.2). Cuando se genera una imagen por medio de un proceso físico sus valores son proporcionales a la energía radiada por la fuente física, por ejemplo ondas electromagnéticas, por lo tanto:

$$0 < I(x,y) < \infty \quad (\text{Ec. 2.1})$$

Entonces se definirá el nivel de gris  $I$  como el valor de la intensidad de una imagen monocromática en las coordenadas  $(x_0, y_0)$ :

$$I = I(x_0, y_0) \quad (\text{Ec. 2.2})$$

De (2.1) se tiene que:

$$L_{\min} < I < L_{\max} \quad (\text{Ec. 2.3})$$

Se deduce que  $L_{\min}$  tiene que ser positivo y  $L_{\max}$  tiene que ser finito, al intervalo  $[L_{\min}, L_{\max}]$  se le llama escala de grises. Este intervalo corresponde a  $[0, L-1]$ , donde  $I = 0$  corresponde al color negro y  $I = L-1$  al valor del blanco en la escala de grises.

Existen muchas formas de obtener una imagen, pero el objetivo en todas ellas es generar imágenes digitales desde los datos capturados. La salida de la mayoría de los sensores es una forma de onda continua de voltajes, su comportamiento de amplitud y espacial están relacionados al fenómeno físico asociado a los sensores. Para crear una imagen digital se convierten los datos continuos en su forma digital, esto es, muestrear la función en ambas coordenadas y en amplitud. Digitalizar los valores espaciales es llamado muestreo o sampling y digitalizar los valores de amplitud es llamado cuantización o quantization.

El resultado del muestreo y la cuantización es una matriz de números de  $M$  filas y  $N$  columnas. Los valores de  $(x, y)$  son discretos, los valores para las coordenadas del origen son  $(x, y) = (0, 0)$ , la figura 2.1 lo grafica.

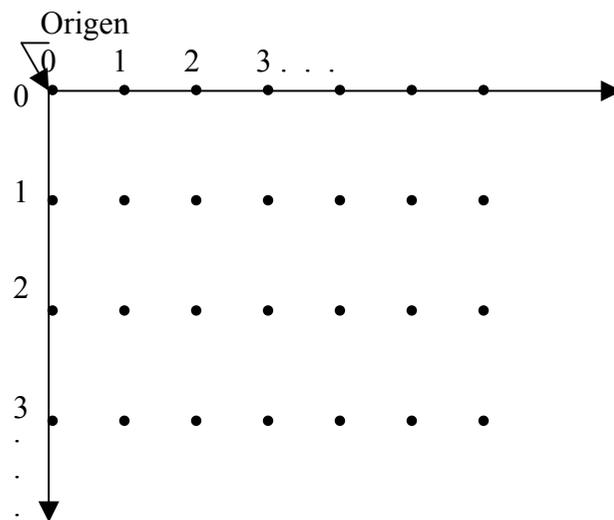


Figura 2.1: Representación digital de la imagen.

La imagen digital  $M \times N$  escrita como matriz es de la forma:

$$I(x,y) = \begin{bmatrix} I(0,0) & I(0,1) & \dots & I(0,N-1) \\ I(1,0) & I(1,1) & \dots & I(1,N-1) \\ \vdots & \vdots & & \vdots \\ I(M,0) & I(M,1) & \dots & I(M,N-1) \end{bmatrix} \quad (\text{Ec.2.4})$$

Entonces  $I(x,y)$  es por definición una imagen digital y cada elemento de la matriz es denominado elemento de la imagen, elemento del cuadro o píxel.

El proceso de digitalización requiere decidir acerca de los valores de  $M$ ,  $N$  y  $L$ , donde  $L$  es el nivel de los valores de grises permitidos para cada píxel. Es importante considerar éste valor porque impacta sobre el proceso de almacenamiento, el número de niveles de grises se define como:

$$L = 2^k \quad (\text{Ec.2.5})$$

La cantidad que se requiere para almacenar una imagen es  $b$ :

$$b = M \times N \times k \quad (\text{Ec.2.6})$$

Cuando  $M = N$ ,

$$b = N^2 k \quad (\text{Ec.2.7})$$

Por ejemplo cuando una imagen puede tener  $2^k$  niveles de grises, quiere decir que la imagen es una "imagen de  $k$ -bit", entonces con  $k = 3$  tenemos una imagen de 8-bit que tiene 256 niveles de grises [GON02].

### 2.1.3. Relaciones Entre Píxeles.

Las principales relaciones que pueden existir entre los píxeles de una imagen digital se denominan Vecindad.

Los vecinos de un píxel  $p$  que tiene coordenadas  $(x,y)$  son 4, dos verticales y dos horizontales y están dados por:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \quad (\text{Ec.2.8})$$

A este conjunto de píxeles se le llama 4-vecinos de  $p$ , se anotará como  $N_4(p)$ .

Cada píxel es una unidad de distancia desde  $(x,y)$ , hay que destacar que algunos de los vecinos de  $p$  podrían no pertenecer a la imagen, es decir, si  $(x,y)$  es un punto del borde, algunos de los vecinos estarán fuera de la imagen.

Además un píxel  $p$  tiene 4 vecinos diagonales  $N_D(p)$  que tienen las coordenadas:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1) \quad (\text{Ec.2.9})$$

$N_4(p)$  junto con  $N_D(p)$  son llamados 8-vecinos de  $p$  o  $N_8(p)$ , como se dijo antes algunos de los puntos de  $N_4(p)$  o  $N_D(p)$  podrían estar fuera de la imagen si  $(x,y)$  pertenece al borde.

La conectividad entre píxeles es un concepto fundamental, para determinar si dos píxeles están o no conectados se debe determinar si ellos son vecinos y si sus niveles de grises satisfacen un criterio de similitud, que indica que sus niveles de grises son iguales.

Tenemos que  $V$  es el conjunto de valores de niveles de gris utilizados para definir adyacencia. En una imagen binaria dos píxeles pueden ser 4-vecinos pero se dice que están conectados si ellos tienen el mismo valor,  $V = 1$  se refiere a la adyacencia de píxeles con valor 1. En una imagen con escalas de grises el principio es el mismo, sólo que el conjunto  $V$  tiene más elementos, por

ejemplo desde 0 a 255 entonces  $V$  es sería un subconjunto de 256 valores. Hay tres tipos de adyacencia:

- a) 4 – adyacencia. Dados dos píxeles  $p$  y  $q$  con valores de  $V$  son 4 – adyacencia si  $q$  está en el conjunto  $N_4(p)$ .
- b) 8 – adyacencia. Dados dos píxeles  $p$  y  $q$  con valores de  $V$  son 8 – adyacencia si  $q$  está en el conjunto  $N_8(p)$ .
- c)  $m$  – adyacencia (adyacencia mezclada). Dados dos píxeles  $p$  y  $q$  con valores de  $V$  son  $m$  – adyacentes si:
  - i)  $q$  está en  $N_4(p)$ , o
  - ii)  $q$  está en  $N_D(p)$  y el conjunto  $N_4(p) \cap N_4(q)$  no tiene píxeles con valores de  $V$ .

Una ruta o path digital desde un píxel  $p = (x,y)$  a un píxel  $q = (s,t)$  es una secuencia de píxeles distintos con coordenadas:

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \quad (\text{Ec.2.10})$$

Donde  $p = (x_0, y_0)$  y  $q = (x_n, y_n)$ , y los píxeles  $(x_i, y_i)$  y  $(x_{i-1}, y_{i-1})$  son adyacentes para  $1 \leq i \leq n$ , donde  $n$  es el largo del path.

Se denotará  $S$  un subconjunto de píxeles en una imagen. Sea  $p$  y  $q$  dos píxeles se dice que están conectados en  $S$  si existe una ruta entre ellos compuesta de píxeles que pertenecen a  $S$ .

Para cualquier píxel  $p$  en  $S$ , el conjunto de píxeles que están conectados a éste en  $S$  es llamado componente conectado de  $S$ .

Y finalmente  $R$  es un subconjunto de píxeles en la imagen. Entonces  $R$  es una región de la imagen si  $R$  es un conjunto conectado. Los límites (bordes o contornos) de una región  $R$  son el conjunto de píxeles en la región que tienen uno o más vecinos que no están en  $R$ .

El concepto de borde, no está claro, el límite de una región forma una ruta cerrada y es un concepto global, los bordes están formados por píxeles con valores derivativos que exceden un Threshold o valor umbral, siendo entonces un concepto local que se basa en la medida de discontinuidad del nivel de gris en un punto. Entonces los bordes son intensidades discontinuas y los límites son rutas cerradas [GON02].

## 2.1.4. Operaciones Básicas Entre Píxeles.

Diversas operaciones entre imágenes serán necesarias a lo largo del presente trabajo, tales como restas, sumas o divisiones de imágenes. En la ecuación 2.4 se representa una imagen en forma de matriz. La división de matrices no está definida, por lo que ésta operación será como una división de una imagen por otra, es decir se divide el primer elemento de una imagen por el primer elemento de la otra, por supuesto que los píxeles de la segunda imagen son distintos de cero. Operaciones aritméticas y lógicas se realizan de la misma forma definida entre los píxeles correspondientes de las imágenes involucradas.

### 2.1.4.1. Medidas de Distancias.

Para los píxeles  $p$ ,  $q$  y  $z$ , con coordenadas  $(x,y)$ ,  $(s,t)$  y  $(v,w)$  respectivamente,  $D$  es una función de distancia o métrico si:

a)  $D(p,q) \geq 0$ , ( $D(p,q) = 0$  si  $p = q$ ),

b)  $D(p,q) = D(q,p)$ , y

c)  $D(p,z) \leq D(p,q) + D(q,z)$ .

La Distancia euclidiana entre los píxeles  $p$  y  $q$  es:

$$D_e(p,q) = [(x,s)^2 + (y,t)^2]^{1/2} \quad (\text{Ec. 2.11})$$

Para ésta medida de distancia, los píxeles que tienen una distancia menor o igual al valor  $r$  de  $(x,y)$  son puntos contenidos en un disco de radio  $r$  con centro en  $(x, y)$ .

La distancia  $D_4$  entre  $p$  y  $q$  es definida como:

$$D_4(p,q) = |x - s| + |y - t| \quad (\text{Ec. 2.12})$$

En éste caso los píxeles tienen una distancia  $D_4$  desde  $(x,y)$  menor o igual que el mismo valor de  $r$  desde un diamante centrado en  $(x,y)$ .

La distancia  $D_8$  entre  $p$  y  $q$  es definida como:

$$D_8(p,q) = \max(|x - s|, |y - t|) \quad (\text{Ec. 2.13})$$

En éste caso los píxeles con distancia  $D_8$  desde  $(x,y)$  menor o igual que el mismo valor de  $r$  desde un cuadrado con centro en  $(x,y)$ .

Las distancias  $D_4$  y  $D_8$  entre  $p$  y  $q$  son independientes de cualquier ruta o path que pudiera existir entre los puntos, porque éstas distancias sólo involucran las coordenadas de los puntos. Si elegimos considerar las  $m$ -adyacencia, la distancia  $D_m$  entre dos puntos se define como la  $m$ -ruta más corta entre los puntos. En éste caso, la distancia entre dos los píxeles dependerá de los valores de los píxeles a lo largo del path, así como también de los valores de sus vecinos.

Suponiendo que se considera la adyacencia de un píxel con valor 1 ( $V = \{1\}$ ). Si  $p_1$  y los  $p_3$  son 0, la longitud del  $m$ -path más corto (distancia  $D_m$ ) entre  $p$  y  $p_4$  es 2. Si el  $p_1$  es 1, entonces el  $p_2$  y  $p$  ya no serán  $m$ -adyacentes (Ver sección 2.1.3-c)) y la longitud del  $m$ -path más corto se vuelve 3 (el path va a través de los puntos  $pp_1p_2p_3p_4$ ). Similares comentarios se aplican si  $p_3$  es 1 (y  $p_1$  es 0); en este caso, la longitud del  $m$ -path más corto también es 3. Finalmente, si  $p_1$  y  $p_3$  son 1 el largo del  $m$ -path más corto entre  $p$  y el  $p_4$  es 4. En este caso, el camino pasa por la secuencia de puntos  $pp_1p_2p_3p_4$ .

#### **2.1.4.2. Operaciones Lineales y No Lineales.**

Se tiene que H es un operador cuya entrada y salida son imágenes, entonces se dice que H es un operador Lineal si, para dos imágenes f y g y cualquier escalar a y b se cumple que:

$$H ( af + bg ) = aH(f) + bH(g) \quad (\text{Ec. 2.14})$$

Es decir el resultado de aplicar a un operador lineal a la suma de dos imágenes es idéntico a aplicar el operador a las imágenes individualmente, multiplicando los resultados por las constantes adecuadas, y sumando los resultados. Un operador que no cumple la ecuación 2.14 se define como No Lineal.

Los operadores lineales son muy importantes en procesamiento de imágenes porque son la base de los resultados teóricos y prácticos.

## **2.2. Técnicas de Reducción de Ruido y Realce de la Imagen.**

El objetivo de hacer una transformación de la imagen (más bien transformaciones) es obtener a partir de los valores de intensidad de los píxeles, características que den una mejor información sobre la física de los objetos en la imagen, y que sean de mayor utilidad para los niveles posteriores. Es decir, procesar una imagen para que el resultado sea más adecuado que la Imagen Original (IO) para su posterior análisis.

Los métodos de realce tienen dos categorías que son los métodos en el dominio espacial y los métodos en el dominio de frecuencia. En el dominio espacial se refiere a la imagen su mismo plano, y los enfoques en ésta categoría están basados en la directa manipulación del píxel en una imagen. Las técnicas de procesado en el dominio de frecuencia están basados en modificar la Transformada de Fourier de una imagen. La evaluación visual de la calidad de una imagen es un proceso subjetivo, incluso en situaciones cuando un criterio bien definido de rendimiento puede ser impuesto en un problema, se requiere de cierta cantidad de prueba y error antes de elegir o seleccionar la técnica de realce que será aplicada a una imagen particular.

### 2.2.1. Introducción.

Previo a la obtención de las características de una imagen, muchas veces, necesario “mejorar” o “Realzar” la calidad de la imagen para resaltar las características deseadas y eliminar las no deseadas, llamadas “Ruido”. Una operación puntual en el dominio espacial transforma una imagen de entrada a una imagen de salida de forma que cada píxel de la imagen de salida sólo depende del correspondiente píxel de la imagen de entrada. Las operaciones en el dominio espacial se rigen por:

$$g(x,y) = T[I(x,y)] \quad (\text{Ec. 2.15})$$

Donde  $I(x,y)$  es la imagen de entrada y  $g(x,y)$  es la imagen procesada, y  $T$  es un operador aplicado sobre  $I$ , definido sobre alguna vecindad de  $(x,y)$ . Además  $T$  puede operar sobre un conjunto de imágenes de entrada, tal como ejecutar la suma píxel a píxel de imágenes para reducir el ruido (ruido promedio de las imágenes). El principal enfoque en definir una vecindad al rededor de un punto  $(x,y)$  es usar un área de subimagen cuadrada o rectangular centrada en  $(x,y)$ , ver Figura 2.2.

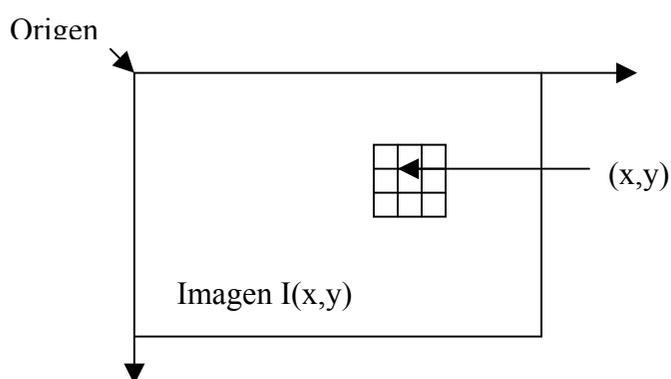


Figura 2.2: Vecindad de 3x3 alrededor de un punto  $(x,y)$  en una imagen.

El centro de la subimagen se mueve de píxel a píxel comenzando en la esquina superior izquierda. El operador  $T$  es aplicado en cada posición  $(x,y)$  para

producir la salida  $g$  en  $(x,y)$ . El proceso utiliza solo los píxeles en el área de la imagen alcanzada por los vecinos. Aunque a veces otras formas de vecinos son utilizadas, tales como círculos, arreglos cuadrados y rectangulares son predominantes por su fácil implementación.

La forma más simple de  $T$  es cuando la vecindad es de tamaño  $1 \times 1$  (un sólo píxel), en éste caso  $g$  depende sólo del valor de  $I$  en  $(x,y)$  y entonces  $T$  es el nivel de gris de la función transformación de la forma:

$$s = T(r) \quad (\text{Ec. 2.16})$$

Donde  $r$  y  $s$  representan a  $g(x,y)$  y  $I(x,y)$  respectivamente. Por ejemplo si  $T(r)$  tiene la forma mostrada en la Fig. 2.3-a), el efecto de ésta transformación será producir una imagen de más alto contraste que la original por medio de oscurecer los niveles inferiores a  $m$  e iluminar los niveles superiores a  $m$  en la IO. Esta técnica se conoce como Contrast Stretching (extender o aumentar el contraste), los valores de  $r$  debajo de  $m$  están comprimidos por la función de transformación en un estrecho rango de  $s$  hacia el negro. El efecto opuesto ocurre con los valores de  $r$  sobre  $m$ , el caso límite se muestra en la Fig. 2.3-b),  $T(r)$  produce una imagen binaria, es decir a dos niveles. Un mapeo de ésta forma es llamado función de Thresholding. El mejoramiento o realce de la imagen en un punto depende sólo del nivel de gris en ese punto, las técnicas en ésta categoría son llamadas Procesamiento de Puntos.

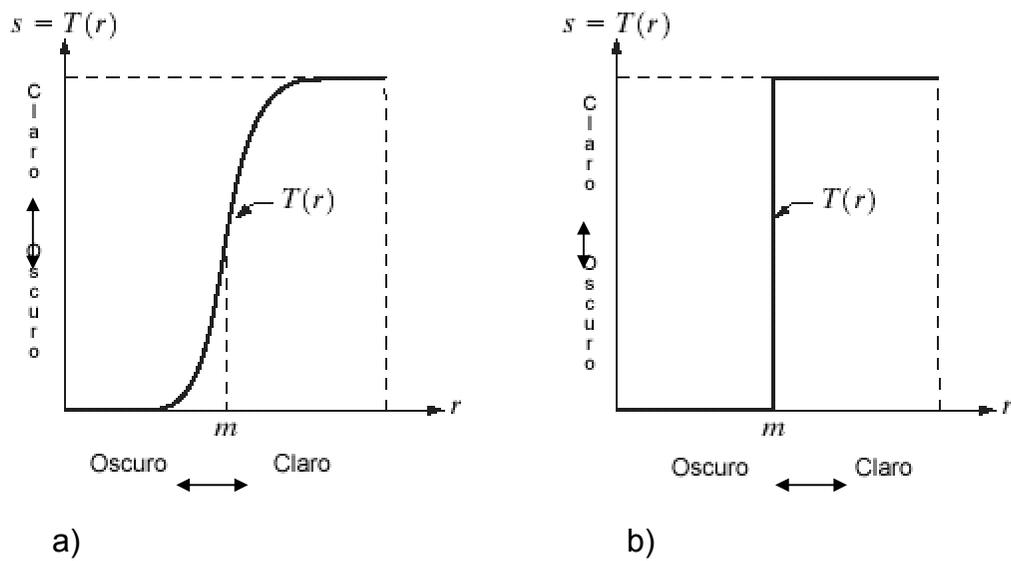


Figura 2.3: Efecto de aplicar Transformaciones.

Las vecindades grandes permiten más flexibilidad. El enfoque general es utilizar una función de valores de  $I$  en una vecindad predefinida de  $(x,y)$  para determinar el valor de  $g$  en  $(x,y)$ . Uno de los principales enfoques se basa en la utilización de Máscaras (también llamadas Filtros, Núcleos o Templates). Básicamente un Filtro es un pequeño arreglo 2D, es decir  $3 \times 3$  por ejemplo, ver Figura 2.2, en que los valores de los coeficientes de los filtros determinan la naturaleza de el proceso, así como también definen la imagen.

## 2.2.2. Mejoras Mediante Procesamiento Puntual.

Se puede decir que éste tipo de transformaciones son operaciones de memoria cero, es decir, no tienen en cuenta información local. Un nivel de gris  $u \in [0,L]$  se transforma en otro nivel de gris  $v$ , usualmente en el mismo rango, mediante:

$$V = T(u) \quad (\text{Ec. 2.17})$$

### 2.2.2.1. Ampliación de Contraste.

Las imágenes con poco contraste se deben a la poca iluminación o no uniformidad o a la no linealidad o pequeño rango dinámico de los sensores de imágenes. La idea de perseguir ampliación de contraste es incrementar el rango dinámico de los niveles de gris en las imágenes que están siendo procesadas. En la Figura 2.4 muestra un ejemplo de transformación de ampliación de contraste.

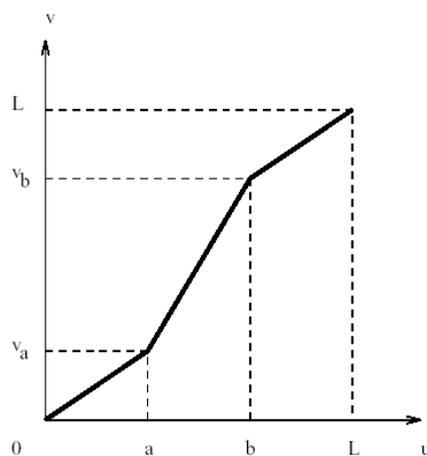


Figura 2.4: Ampliación de contraste.

Mediante:

$$s = \begin{cases} \alpha u & 0 \leq u \leq a \\ \beta(u - a) & a \leq u \leq b \\ \gamma(u - b) & b \leq u \leq L \end{cases} \quad (\text{Ec. 2.18})$$

Las pendientes se toman mayores que uno en las regiones en las que se quiere ampliar el contraste.

Los parámetros  $a$  y  $b$  se pueden estimar examinando el histograma de la imagen. Por ejemplo, los intervalos de nivel de gris donde los píxeles ocurren con mayor frecuencia deben ampliarse para mejorar la visibilidad de la imagen.

### **2.2.2.2. Corte y Umbralización.**

Es un caso particular del anterior, corresponde a tomar  $\alpha = \gamma = 0$  que recibe el nombre de Corte. Este Corte puede ser útil por ejemplo cuando sabemos que la señal está en el rango  $[a,b]$ .

La umbralización es un caso especial del modelo de corte en el que se obtiene una imagen binaria. Esto puede expresarse como:

$$v = \begin{cases} 0 & 0 \leq u \leq a \\ L & a \leq u \leq b \\ 0 & b \leq u \leq L \end{cases} \quad (\text{Ec. 2.19})$$

Este modelo puede ser útil cuando sabemos que la imagen tiene que ser binaria y por el proceso de digitalización no se ha obtenido así. Este método puede ser utilizado en un proceso de segmentación.

### 2.2.2.3. Negativo Digital.

Una imagen negativa puede obtenerse cambiando la escala mediante la transformación:

$$v = L - 1 - u \quad (\text{Ec. 2.20})$$

Invirtiendo los niveles de intensidad de una imagen se produce el equivalente de un negativo fotográfico. Este tipo de procesamiento es adecuado para los detalles blancos o negros de las regiones oscuras de una imagen, especialmente cuando las áreas oscuras son dominantes en tamaño.

### 2.2.2.4. Extracción de Bits.

Supongamos que cada píxel en una imagen es cuantificada uniformemente en B bits, se quiere extraer el n-ésimo bit más significativo y mostrarlo. Se tiene:

$$u = k_1 2^{B-1} + k_2 2^{B-2} + \dots + k_n 2^{B-n} + \dots + k_{B-1} 2^1 + k_B \quad (\text{Ec. 2.21})$$

Se quiere que el resultado sea:

$$v = \begin{cases} L, & \text{Si } k_n = 1 \\ 0, & \text{En caso contrario} \end{cases} \quad (\text{Ec. 2.22})$$

Este tipo de transformación es útil para determinar el número de bits que son visualmente significativos, ya que el despliegue de determinados planos puede probar que ellos no son significativos.

### 2.2.2.5. Compresión del Rango.

A veces el rango dinámico de una imagen puede ser muy grande. Son ejemplos típicos las imágenes astronómicas o las de su función de emborronamiento. El rango dinámico puede comprimirse mediante la transformación.

$$v = c \log_{10}(1+u) \quad (\text{Ec. 2.23})$$

Donde  $c$  es una constante para escalado. Este tipo de transformaciones realiza las pequeñas magnitudes de los píxeles comparada con las grandes magnitudes.

#### **2.2.2.6. Potenciación.**

Otro tipo de transformación que puede ser útil y que algunos casos corresponde a la respuesta de determinados medios de captación de imágenes es la transformación de la forma:

$$v = \delta(u/c)^\lambda \quad (\text{Ec. 2.24})$$

$\delta$  es una constante de escalado mientras que  $c$  lleva el rango inicial al intervalo  $[0,1]$  y  $\lambda \in [0,\infty]$ . Si  $\lambda < 1$  amplía los niveles de gris bajos, mientras que si  $\lambda > 1$  el efecto es el contrario.

#### **2.2.2.7. Diferencia de Imágenes.**

Esta técnica tiene numerosas aplicaciones en segmentación y realce. La diferencia entre dos imágenes  $f(x,y)$  y  $h(x,y)$  se expresa como:

$$g(x,i) = f(x,y) - h(x,y) \quad (\text{Ec. 2.25})$$

Se obtiene calculando la diferencia entre todos los pares de píxeles correspondientes de  $f$  y  $h$ .

### 2.2.2.8. Promedio de Imágenes.

Supongamos que el valor del ruido  $g(x,y)$  está formado por la suma del el ruido  $\eta(x,y)$  de una IO  $f(x,y)$ :

$$g(x,y) = f(x,y) + \eta(x,y) \quad (\text{Ec. 2.26})$$

Donde se supone que en todo par de coordenadas  $(x,y)$  el ruido es no correlacionado y tiene media cero y desviación estándar  $\sigma$ . El objetivo es reducir el ruido agregando imágenes ruidosas,  $g_i(x,y)$ . Si el ruido satisface las condiciones dadas, se puede mostrar que una imagen  $\bar{g}(x,y)$  está formada por el promedio  $K$  de diferentes imágenes ruidosas,

$$\bar{g}(x,y) = \frac{1}{K} \sum_{i=1}^K g_i(x,y) \quad (\text{Ec. 2.27})$$

Por lo tanto si tenemos  $n$  imágenes de la misma escena, podemos obtener una nueva versión de la IO con menos ruido aplicando la ecuación 2.27.

### 2.2.3. Procesamiento por Histograma.

Los histogramas son la base de muchas técnicas de procesamiento en el dominio espacial, la manipulación del histograma puede ser utilizado efectivamente para mejorar o realzar la calidad de la imagen.

El histograma de una imagen digital con niveles de gris en el rango  $(0, L-1)$  es una función discreta con valores de  $p(r_k) = n_k/n$  donde  $r_k$  es el  $k$ -ésimo nivel de gris,  $n_k$  es el número de píxeles en la imagen con ese nivel de gris ( $r_k$ ), y  $n$  es el número total de píxeles en la imagen y  $k = 0, 1, 2, \dots, L-1$ .  $p(r_k)$  entrega un estimador de la probabilidad de ocurrencia del nivel de gris  $r_k$ . La suma de todos los componentes de un histograma normalizado es igual a 1.

En la Figura 2.5 se muestran los histogramas de cuatro tipos básicos de imágenes, el histograma Fig. 2.5-a), muestra que los niveles de grises están concentrados en la zona oscura de la escala. La Fig. 2.5-b) muestra lo contrario, es decir, que los niveles de gris están concentrados en la zona clara o brillante de la escala. En la Fig. 2.5-c) se muestra un histograma de bajo contraste y la Fig. 2.5-d) un histograma de alto contraste. El eje horizontal de cada histograma corresponde a los valores del nivel de gris,  $r_k$ . El eje vertical corresponde a los valores de  $p(r_k) = n_k$  o  $p(r_k) = n_k/n$  si los valores están normalizados.

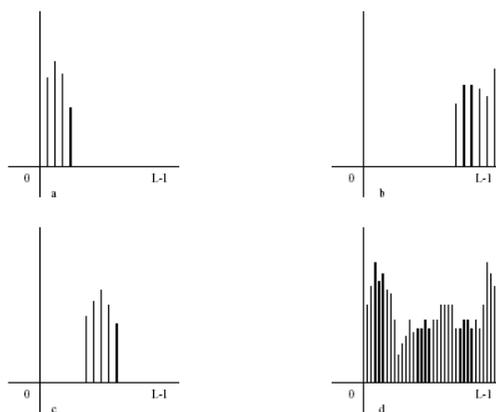


Figura 2.5: Ejemplos de Histogramas.

### 2.2.3.1. Uniformización del Histograma.

Sean  $r$  los niveles de gris en la imagen que queremos mejorar. Supondremos que los valores de los píxeles son cantidades continuas que han sido normalizadas para que se extiendan en el rango  $[0,1]$ , con  $r = 0$  representando al negro y  $r = 1$  al blanco. Posteriormente consideraremos la formulación discreta y permitiremos que los valores de los píxeles estén en el rango  $[0,L-1]$ .

Para cualquier  $r$  en el intervalo  $[0,1]$  nos concentraremos en las transformaciones de la forma:

$$S = T(r), 0 \leq r \leq 1. \quad (\text{Ec. 2.28})$$

Que produce un nivel de gris  $s$  para cada nivel de gris  $r$  de la IO. Se asume que la transformación que se usa en la ecuación 2.28 cumple las siguientes condiciones:

- a)  $T(r)$  es una función univaluada monótonamente creciente en  $0 \leq r \leq 1$  y
- b)  $0 \leq T(r) \leq 1$  para  $0 \leq r \leq 1$ .

El requerimiento que  $T(r)$  sea univaluada es necesario para garantizar que la transformación inversa exista y la condición de monótonamente creciente preserva el orden incremental de negro a blanco en la salida de la imagen, mientras que la segunda condición garantiza que la aplicación es consistente con el rango de niveles de gris permitido. Esto se muestra en la Figura 2.6.

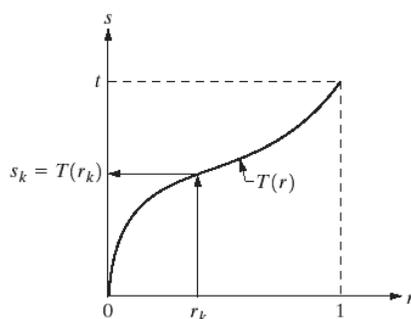


Figura 2.6: Función de transformación de niveles de gris que es univaluada y monótonamente creciente.

La transformación inversa de  $s$  a  $r$  se anota como:

$$r = T^{-1}(s), 0 \leq r \leq 1 \quad (\text{Ec. 2.29})$$

Esta transformación debe cumplir las condiciones (a) y (b) anteriores y en particular ser univaluada.

Los niveles de gris en una función pueden considerarse como cantidades aleatorias en el intervalo  $[0,1]$ . Si son variables continuas, los niveles de gris original y transformado pueden caracterizarse por sus funciones de densidad de probabilidad  $p_r(r)$  y  $p_s(s)$ , respectivamente. Por resultados de la teoría de probabilidades, si  $p_r(r)$  y  $p_s(s)$  son conocidos y  $T_{-1}(s)$  satisface la primera condición anterior (a), la función de densidad de probabilidad  $p_s(s)$  de la variable transformada  $s$  puede ser obtenida usando:

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| \quad (\text{Ec. 2.30})$$

La función de probabilidad de la variable  $s$  transformada, se determina por el nivel de gris de la función de densidad de probabilidad de la imagen de entrada y por la elección de  $T(r)$ .

Consideremos la función de transformación ( $w$  es una variable muda de integración):

$$s = T(r) = \int_0^r p_r(w) dw, 0 \leq r \leq 1 \quad (\text{Ec. 2.31})$$

A ésta ecuación se la llama función de distribución acumulativa, de una variable aleatoria  $r$ , las funciones de densidad de probabilidad son siempre positivas y tomando en cuenta que la integral es el área bajo la función, por consiguiente la función de transformación es univaluada y monótonamente creciente, esto

satisface la condición (a), de forma similar la función de densidad de probabilidad para variables en el rango  $[0,1]$  también está en el rango  $[0,1]$ , ésto satisface la condición (b). Además:

$$\frac{ds}{dr} = p_r(r) \quad (\text{Ec. 2.32})$$

Sustituyendo en la función (2.30):

$$P_s(s) = 1 \quad (\text{Ec. 2.33})$$

Esto es la densidad uniforme en el intervalo de definición de la variable transformada  $s$ . Este resultado es independiente de la transformación inversa, ésto es importante puesto que no siempre es fácil obtener la transformación inversa.

El desarrollo anterior indica que usando una función de transformación igual a la función de distribución de  $r$  produce una imagen cuyos niveles de gris tienen una densidad uniforme. En términos de mejora de la imagen ésto significa un crecimiento del rango dinámico de los píxeles, que puede mejorar considerablemente la apariencia de la imagen [GON02].

### 2.2.3.2. Especificación del Histograma.

Aunque el método descrito en la sección 2.2.3.1 es bastante útil, no conduce a la aplicación de técnicas de mejoras interactivas ya que éste método es sólo capaz de generar un resultado, una aproximación a la distribución uniforme.

Es a veces deseable ser capaces de especificar interactivamente histogramas concretos que resalten determinados rangos de niveles de gris, para ver como ésto puede llevarse a cabo, supongamos, en el caso continuo, que  $p_r(r)$  y  $p_z(z)$  son las densidades de probabilidad original y deseada respectivamente. Supongamos que a una imagen dada se le uniformiza el histograma utilizando:

$$s = T(r) = \int_0^r p_r(w)dw \quad (\text{Ec. 2.34})$$

Obviamente si la imagen deseada estuviera disponible, sus niveles podrían ser uniformizados utilizando la función de transformación:

$$v = G(z) = \int_0^z p_z(w)dw \quad (\text{Ec. 2.35})$$

Entonces si  $G(z) = T(r)$ ,  $z$  satisface:

$$z = G^{-1}(s) = G^{-1}[T(r)] \quad (\text{Ec. 2.36})$$

Lo que necesitamos es construir una distribución uniforme, ésto puede llevarse a cabo utilizando el  $s$  obtenido de la IO, los niveles resultantes  $z = G^{-1}(s)$  tendrían entonces la función de densidad deseada. Suponiendo que  $G^{-1}(s)$  es univaluada, el proceso puede resumirse de la siguiente forma:

1. Uniformizar los niveles de la IO, es decir obtener la función de transformación  $T(r)$ ,
2. Especificar la función de densidad deseada y obtener la transformación  $G(z)$  que la llevaría a una uniforme,

3. Aplicar la transformación inversa  $z = G^{-1}(s)$  a los niveles que hemos obtenido en el paso 1.
4. Obtener la imagen de salida aplicando (2.35) a todos los píxeles de la imagen de entrada.

El resultado de éste proceso será una imagen cuyos niveles de gris,  $z$ , tienen la función de densidad de probabilidad especificada por  $P_z(z)$  [GON02].

### **2.2.3.3. Mejora Local**

Las técnicas de procesamiento del histograma discutidas antes, son globales, en el sentido que los píxeles son modificados por una función de transformación basada en la distribución de los niveles de gris contenidos en la imagen completa. Es obvio que en algunos casos es necesario una mejora local de los detalles de la imagen sobre áreas pequeñas.

Las técnicas de procesamiento del histograma descritas antes son fácilmente adaptables a mejoras locales. El procedimiento es definir una ventana cuadrada o rectangular y centrarla sobre cada píxel aplicando las técnicas del histograma a ésta región.

En lugar de usar el histograma, la mejora local puede basarse en las propiedades de los píxeles en un entorno que se está analizando. La intensidad media y la varianza son dos propiedades que se usan frecuentemente por su relevancia en la apariencia de la imagen. Se podría decir que la media es una medida del brillo mientras que la varianza mide el contraste.

Una transformación local típica basada en éstos conceptos aplica a la intensidad de una imagen  $f(x,y)$  en un nuevo nivel de gris  $g(x,y)$  mediante la realización de la siguiente transformación en cada punto  $(x,y)$ .

$$g(x,y) = A(x,y) [f(x,y) \cdot m(x,y)] + m(x,y) \quad (\text{Ec. 2.37})$$

Donde,

$$A(x,y) = k \frac{M}{\sigma(x,y)}, \quad 0 < k < 1 \quad (\text{Ec. 2.38})$$

$m(x,y)$  y  $\sigma(x,y)$  son los niveles de gris medio y la desviación típica calculada en el entorno del píxel  $(x,y)$ ,  $M$  es la media global de  $f(x,y)$  y  $k$  es una constante de contraste.

Los valores de las variables  $A$ ,  $m$  y  $\sigma$  dependen del entorno del píxel  $(x,y)$ . La aplicación del factor de ganancia local  $A(x,y)$  a la diferencia entre  $f(x,y)$  y su media local amplifica las variables locales. En regiones con bajo contraste  $A(x,y)$  es grande y recíprocamente. La media se vuelve a sumar para mantener el nivel de intensidad [GON02].

#### 2.2.4. Filtrado Espacial.

El uso de máscaras espaciales para procesamiento de imágenes recibe el nombre de Filtrado Espacial (para diferenciarlo del filtrado en el dominio de las frecuencias que utiliza la transformada de Fourier), las máscaras reciben el nombre de filtros espaciales, a continuación se examinarán los filtros espaciales, lineales y no lineales para mejorar la imagen.

Los filtros lineales se basan en transformaciones de la forma:

$$g(i, j) = \sum_{m, n \in \mathcal{D}} h(i - m, j - n) f(m, n) \quad (\text{Ec. 2.39})$$

Donde  $f(i, j)$  es la imagen de entrada,  $g(i, j)$  es la imagen de salida y  $h(i, j)$  es el llamado núcleo de convolución. Se utiliza una malla de vecinos  $\mathcal{D}$  rectangular, con un número impar de filas y columnas asignándole valores también al píxel central.

Este tipo de procesamiento utiliza muy poco conocimiento a priori sobre el contenido de la imagen. Además el tamaño de  $\mathcal{D}$  es algo que puede variar en función del tamaño de los objetos de la imagen.

Los *filtros de paso bajo* atenúan o eliminan componentes de alta frecuencia en el dominio de Fourier mientras que dejan las frecuencias bajas sin alterarlas. Las frecuencias altas en el dominio de Fourier caracterizan las fronteras y detalles con alta variación en la imagen, así los filtros de paso bajo emborronan en algún sentido la imagen. El efecto contrario se obtiene con los *filtros de paso alto*.

Independiente del tipo de filtro lineal usado, la aproximación básica es sumar los productos entre los coeficientes de la plantilla y las intensidades de los píxeles en la plantilla. Un tipo de plantilla o máscara es:

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

Figura 2.7: Representación general de una máscara 3x3 de filtro espacial, donde  $w_5$  es el origen.

Si anotamos los niveles de gris de los píxeles en la máscara como  $z_1, z_2, \dots, z_9$ , la respuesta de la máscara lineal es:

$$R = w_1z_1 + w_2z_2 + \dots + w_9z_9 \quad (\text{Ec. 2.40})$$

Los filtros espaciales no lineales también operan sobre los vecinos, sin embargo, en general sus operaciones están basadas directamente en los valores de los píxeles que se consideran y no usan coeficientes en la forma de los filtros lineales. La reducción espacial también puede ser alcanzada con filtros no lineales que calculan la mediana de los niveles de gris. Otro ejemplo incluye el máximo de los 9 píxeles o el mínimo que pueden usarse para calcularlos píxeles localmente más brillantes u oscuros respectivamente.

### 2.2.4.1. Filtros de Suavizado.

Los filtros de suavizado se utilizan con el objetivo de emborronar o para reducir el ruido. El emborronamiento es usado como pre-procesamiento para suprimir detalles pequeños antes de la extracción de objetos grandes, también puede usarse para unir segmentos. La reducción de ruido puede llevarse a cabo con emborronamiento de un filtro lineal y también por un filtro no lineal.

#### Filtros De Paso Bajo

Aunque el filtro espacial podría ser modelado mediante una gaussiana o en general cualquier función de densidad o su aproximación continua, es usual restringirse a los filtros de tamaño 3x3 o bien máscaras con forma de estrella. Es muy importante que la suma de los coeficientes sea la unidad.

Por ejemplo, se supone que  $h(0) = h(1) = h(-1) = 1/3$ , para una señal de tamaño  $M$ . Su transformada de Fourier es:

$$F(u) = \frac{1}{M} \frac{1}{3} (1 + 2 \cos(2\pi u / M)) \quad (\text{Ec. 2.41})$$

Esta función casi no altera las frecuencias bajas (cercanas a cero y a  $M$ ) y disminuye las cercanas a la mitad ( $M/2$ ).

Un ejemplo es la siguiente máscara:

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \quad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Figura 2.8: Dos máscaras de filtros de suavizado, la constante multiplicativa es igual a la suma de los valores de sus coeficientes, como se requiere para calcular un promedio.

A veces se aumenta la fuerza del píxel central y la de aquellos que están más próximos.

### **Promediando Según El Rango**

El objetivo es promediar sobre aquellos píxeles que cumplen algunos criterios tratando de evitar el emborronamiento. Por ejemplo se puede usar:

$$h(m,n) = \begin{cases} 1 & \text{si } g(i-m, j-n) \in [\min, \max] \\ 0 & \text{en caso contrario} \end{cases} \quad (\text{Ec. 2.42})$$

El rango puede ser incluso dependiente del nivel de imagen en el píxel central.

### **Alisamiento Por Medianas**

En un conjunto de valores ordenados, la mediana es el valor central. El filtrado por mediana es un método de alisamiento que reduce el emborronamiento de las fronteras. La idea es reemplazar el valor de la imagen en un punto por la mediana de los niveles de brillo de sus vecinos. Es obvio que los puntos aislados de ruido son eliminados y que además no emborronan las fronteras y que puede ser aplicado iterativamente.

Uno de sus mayores problemas es que si se usa una malla rectangular se dañará las líneas finas y los bordes de los objetos. Esto podría ser evitado tomando los vecinos sólo en las direcciones verticales y horizontales.

El alisamiento por mediana es un caso particular del filtrado por estadísticas ordenadas, donde los valores de los vecinos se ordenan en una sucesión y un nuevo valor se le asigna al píxel central que es una combinación lineal de éstos valores ordenados.

## Filtros De Medias No Lineales

El filtro no lineal es una generalización de los promedios. Se define como:

$$f(m,n) = u^{-1} \left( \frac{\sum_{i,j \in \mathcal{S}} a(i,j)u(g(m-i,n-j))}{\sum_{i,j \in \mathcal{S}} a(i,j)} \right) \quad (\text{Ec. 2.43})$$

Si los pesos son constantes el filtro recibe el nombre de homomórfico, algunos ejemplos son:

1. Media aritmética,  $u(g) = g$
2. Media armónica,  $u(g) = 1/g$
3. Media Geométrica,  $u(g) = \log g$

### 2.2.4.2. Filtros de Realce.

El objetivo principal del realce es destacar los detalles finos de una imagen o mejorar detalles que han sido emborronados, bien por errores o por problemas en la adquisición de la imagen.

#### Filtrado Espacial de Paso Alto Básico

Es claro que éste tipo de filtro puede obtenerse si tiene coeficientes positivos cerca de su centro y negativos en la periferia. Para una máscara 3x3 si el valor del centro es positivo y negativo en el resto es suficiente.

Una implementación clásica de filtro de paso alto es la siguiente:

-1/8	-1/8	-1/8
-1/8	1	-1/8
-1/8	-1/8	-1/8

Figura 2.9: Ejemplo filtro de paso alto.

Es importante notar que la suma de los coeficientes es cero, así cuando la máscara se encuentra sobre una región constante o que varia muy suavemente el resultado es cero o muy pequeño.

Su significado en el dominio de Fourier es por ejemplo: si  $h(0) = 1$  y  $h(-1) = h(1) = -1/2$  entonces la transformada de Fourier es:

$$F(u) = \frac{1}{M}(1 - \cos(2\pi u / M)) \quad (\text{Ec. 2.44})$$

Que es cero, o próximo a cero, en las frecuencias bajas y no modifica las altas. El problema es que como éste tipo de filtro produce una imagen de media cero, hemos de buscar formas de llevarlo al rango de la IO.

### **Filtros de Énfasis en las Frecuencias Altas**

Una imagen filtrada de paso alto puede ser calculada como la diferencia entre la IO y una versión de paso bajo de la imagen, es decir:

$$\text{Paso Alto} = \text{Original} - \text{Paso Bajo} \quad (\text{Ec. 2.45})$$

Si multiplicamos la imagen por un factor de ampliación, que se notará A, produce lo que recibe el nombre de filtro de énfasis de las altas frecuencias (FEAF).

$$\begin{aligned} \text{FEAF} &= A \text{ Original} - \text{Paso Bajo} \\ &= (A - 1) (\text{Original}) + \text{Paso Alto} \end{aligned}$$

Cuando  $A = 1$  produce el filtro clásico de paso alto. Cuando  $A > 1$  parte de la IO se añade al filtro y por lo tanto se recuperan algunas de las bajas frecuencias. El resultado es que éste tipo de filtro produce imágenes que se parecen más a la original con relativo grado de mejora de bordes que depende del valor de A.

Este tipo de técnica se usa frecuentemente en impresión y publicaciones de documentos. En la misma línea se pueden obtener resultados similares con otros filtros de forma de la Fig. 2.10, con  $w > 0$ :

$-w$	$-w$	$-w$
$-w$	$1 + 8w$	$-w$
$-w$	$-w$	$-w$

Figura 2.10: Ejemplo filtro de paso alto.

### 2.2.5. Filtrado en el Dominio de la Frecuencia.

El teorema de convolución que se basa en las técnicas de dominio de frecuencias dice que si  $g(x,y)$  es la imagen formada por la convolución de la imagen  $f(x,y)$  y un operador lineal invariante  $h(x,y)$ , es decir,

$$g(x,y) = (h * f)(x,y) \quad (\text{Ec. 2.46})$$

Entonces por el teorema de convolución, se verifica la siguiente relación en el dominio de las frecuencias:

$$G(u,v) = H(u,v)F(u,v) \quad (\text{Ec. 2.47})$$

Donde  $G$ ,  $H$  y  $F$  son las transformadas de Fourier de  $g$ ,  $h$  y  $f$  respectivamente.

Dado  $f(x,y)$ , una vez calculado  $F(u,v)$  el objetivo es seleccionar  $H(u,v)$  de forma que la imagen deseada muestre algunos rasgos de interés de  $f(x,y)$ . Muchos problemas de mejora de imágenes pueden ser formulados de la forma:

$$g(x,y) = F^{-1}[H(u,v)F(u,v)] \quad (\text{Ec. 2.48})$$

Por ejemplo, los bordes de la imagen  $f(x,y)$  se pueden acentuar usando una función que enfatice las frecuencias altas de  $F(u,v)$ .  $h(x,y)$  caracteriza un sistema cuyo objetivo es producir una imagen de salida  $g(x,y)$  a partir de una imagen de entrada  $f(x,y)$ . El sistema realiza la convolución de  $h(x,y)$  con la imagen de entrada y produce el resultado. El teorema de convolución permite dar una nueva interpretación al proceso, en el sentido de que se puede alcanzar el mismo resultado si se multiplica  $F(u,v)$  por  $H(u,v)$  para producir  $G(u,v)$ . calculando la transformada de Fourier inversa se obtiene como resultado la imagen deseada.

La mejora en el dominio de las frecuencias es más bien directa, simplemente se calcula la transformada de Fourier de la imagen que se quiere mejorar, se multiplica el resultado por una función de transferencia, un filtro y se calcula la transformada de Fourier inversa para producir la imagen mejorada.

La idea de emborronar mediante la reducción del contenido de las frecuencias altas o de realzar aumentando el contenido de las frecuencias altas en relación a las bajas, viene obviamente del dominio de Fourier. De hecho la idea del filtrado lineal se considera más atrayente e intuitiva en el dominio de las frecuencias. Sin embargo, en la práctica, las máscaras espaciales pequeñas se usan más que la transformada de Fourier por su simplicidad de implementación y velocidad en las operaciones. Sin embargo, la comprensión de los conceptos en el dominio de las frecuencias es esencial para la solución de problemas que no son fácilmente comprensibles en el dominio espacial.

#### **2.2.5.1. Filtros de Paso Bajo.**

Las fronteras y otras transiciones rápidas como el ruido en los niveles de gris de una imagen contribuyen significativamente al contenido de las frecuencias altas en el dominio de Fourier. Por lo tanto, el emborronamiento o alisamiento se alcanza en el dominio de Fourier atenuando un rango específico de componentes de altas frecuencias en la transformada de una imagen dada. En ecuación 2.47 se tiene que  $F(u,v)$  es la transformada de Fourier de la imagen a alisar. El problema es seleccionar una función de transferencia  $H(u,v)$  que produzca  $G(u,v)$  atenuando las componentes de alta frecuencia de  $F(u,v)$ . La transformada inversa producirá entonces la imagen que se desea.

#### **Filtros Ideales**

Un filtro ideal bidimensional es uno cuya función de transferencia cumple la relación:

$$H(u,v) = \begin{cases} 1 & \text{si } D(u,v) \leq D_0 \\ 0 & \text{en caso contrario} \end{cases} \quad (\text{Ec. 2.49})$$

Donde  $D_0$  es una cantidad no negativa y  $D(u,v)$  es la distancia del punto  $(u,v)$  al origen en el plano de las frecuencias, es decir:

$$D(u,v) = (u^2 + v^2)^{1/2} \quad (\text{Ec. 2.50})$$

El nombre *ideal* proviene de todas las frecuencias dentro de un círculo de radio  $D_0$  se pasan sin atenuación, mientras que todas las frecuencias fuera de ese círculo son atenuadas completamente.

Los filtros que aquí se consideran son radialmente simétricos alrededor del origen, todas las frecuencias de ese círculo son atenuadas.

Para un filtro ideal de paso bajo radialmente simétrico, el punto de transición entre  $H(u,v) = 1$  y  $H(u,v) = 0$  recibe el nombre de Frecuencia de Corte.

El funcionamiento de todos los filtros de paso bajo que se introducen a través del dominio de Fourier se puede comparar examinando las frecuencias de corte. Una forma de establecer un conjunto de frecuencias de corte estándar es calcular círculos que incluyan varias cantidades de la potencia de la señal  $P_T$ . Esta cantidad se obtiene como sigue:

$$P_T = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} P(u,v) \quad (\text{Ec. 2.51})$$

Donde  $P(u,v)$  denota el espacio de potencias. Si la transformación ha sido centrada, un círculo de radio  $R$  con origen en el centro de las frecuencias contiene  $\beta$  por ciento de la potencia.

$$\beta = 100 \left[ \sum_u \sum_v P(u,v) / P_T \right] \quad (\text{Ec. 2.52})$$

Donde en la suma se toman los valores que están dentro del círculo o su frontera.

El emborronamiento y lo que se llama *ringing* de éste tipo de filtros puede explicarse fácilmente en base con el teorema de convolución. Puesto que la transformada de Fourier de la IO y la borrosa se relacionan en el dominio de

Fourier por medio de la ecuación (2.46). Donde  $H(x,y)$  es la transformada inversa del filtro de transferencia  $H(u,v)$ . El paso para entender el proceso de emborronamiento como un proceso de convolución en el dominio espacial está en la forma de  $h(x,y)$ .

### **2.2.5.2. Filtros de Paso Alto.**

Una imagen puede ser emborronada atenuando las componentes de frecuencia alta en su transformada de Fourier. Esto es debido, a que los bordes y otros cambios abruptos en los niveles de gris se asocian con frecuencias altas. El realce de imágenes puede alcanzarse mediante el filtrado de paso alto que atenúa las frecuencias bajas sin distorsionar la información en las frecuencias altas en el dominio de Fourier.

#### **Filtro Ideal**

Un filtro bidimensional ideal de paso alto es uno cuya función de transferencia cumple la relación:

$$H(x,y) = \begin{cases} 0 & \text{si } D(u,v) \leq D_0 \\ 1 & \text{en caso contrario} \end{cases} \quad (\text{Ec. 2.53})$$

Donde  $D_0$  es la distancia de corte medida desde el origen del plano de frecuencias y  $D(u,v)$  ya ha sido definida.

## 2.2.6. Transformada de Fourier.

La teoría de las transformaciones ha jugado un papel fundamental en el procesamiento de imágenes por su descripción alternativa de una imagen y por ser una herramienta para realizar operaciones como la resolución de ecuaciones lineales. Es muy importante tener en cuenta la enorme aplicación de las transformaciones de imágenes en campos como restauración, codificación, mejora y segmentación, entre otros.

### 2.2.6.1. Definición de la Transformada de Fourier.

Una función  $f(x)$ , es una función continua de la variable real  $x$ , la Transformada de Fourier (TF) de  $f(x)$ , se anota  $\mathfrak{T}\{f(x)\}$  se define por:

$$\mathfrak{T}\{f(x)\} = F(\mu) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi\mu x} dx \quad (\text{Ec. 2.54})$$

Dada  $F(\mu)$ ,  $f(x)$  puede obtenerse mediante la transformada inversa de Fourier:

$$\mathfrak{T}^{-1}\{f(x)\} = f(x) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu x} dx \quad (\text{Ec. 2.55})$$

Las ecuaciones (2.54) y (2.55) se denominan el par de TF. La TF puede no existir, salvo que  $f$  cumpla las siguientes condiciones suficientes:

1.  $\int_{-\infty}^{\infty} |f(x)| dx < \infty$ ,
2.  $f(x)$  debe tener sólo un número finito de discontinuidades y un número finito de máximos y mínimos en cualquier rectángulo finito.
3.  $f(x)$  no puede tener discontinuidades infinitas.

En éste tema las funciones  $f(x)$  son reales. Sin embargo, la transformada de Fourier de una función real en general será siempre compleja, es decir:

$$F(\mu) = R(\mu) + il(\mu) \quad (\text{Ec. 2.56})$$

Donde  $R(u)$  e  $I(u)$  son respectivamente las partes real e imaginaria de  $F(\mu)$  con  $i = \sqrt{-1}$ . La ecuación anterior puede escribirse en forma exponencial:

$$F(\mu) = |F(\mu)|e^{i\phi(\mu)} \quad (\text{Ec. 2.57})$$

Donde

$$|F(\mu)| = \sqrt{R^2(\mu) + I^2(\mu)} \quad (\text{Ec. 2.58})$$

y

$$\phi(\mu) = \text{arctg} \left[ \frac{I(\mu)}{R(\mu)} \right] \quad (\text{Ec. 2.59})$$

La función de magnitud  $|F(\mu)|$  recibe el nombre de espectro de Fourier de  $f(x)$  y  $\phi(\mu)$  recibe el nombre de fase. El cuadrado del espectro suele recibir el nombre de espectro de potencias.

$$E(\mu) = |F(\mu)|^2 \quad (\text{Ec. 2.60})$$

La variable  $u$  se denomina comúnmente variable frecuencia, hecho que deriva de la fórmula de Euler:

$$e^{-j2\pi\mu x} = \cos(2\pi\mu x) - j\text{sen}(2\pi\mu x) \quad (\text{Ec. 2.61})$$

Si se interpreta la integral de (2.54) como el límite de una sumatoria de términos discretos, resulta que  $F(\mu)$  está compuesta por una sumatoria infinita de términos seno y coseno, y que cada valor de  $\mu$  determina la frecuencia de su correspondiente par seno-coseno [GON02].

### 2.2.6.2. Extensión a Dos Dimensiones.

La transformada de Fourier puede ser extendida a una función  $f(x,y)$  de dos variables.

Si  $f(x,y)$  es continua e integrable y  $F(\mu,\nu)$  es integrable, se tienen las condiciones suficientes para la existencia del siguiente par de transformaciones de Fourier.

$$\mathfrak{F}\{f(x,y)\} = F(\mu,\nu) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) e^{-j2\pi(\mu x + \nu y)} dx dy \quad (\text{Ec. 2.62})$$

y

$$\mathfrak{F}^{-1}\{F(\mu,\nu)\} = f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(\mu,\nu) e^{-j2\pi(\mu x + \nu y)} d\mu d\nu \quad (\text{Ec. 2.63})$$

Donde  $\mu$  y  $\nu$  son las variables de frecuencia [GON02].

Tal como en el caso de una dimensión, el Spectrum de Fourier, y los de fase y energía, están dados, respectivamente por:

$$|F(\mu,\nu)| = \sqrt{R^2(\mu,\nu) + I^2(\mu,\nu)} \quad (\text{Ec. 2.64})$$

$$\phi(\mu,\nu) = \text{arctg} \left[ \frac{I(\mu,\nu)}{R(\mu,\nu)} \right] \quad (\text{Ec. 2.65})$$

$$E(\mu,\nu) = |F(\mu,\nu)|^2 = R^2(\mu,\nu) + I^2(\mu,\nu) \quad (\text{Ec. 2.66})$$

### 2.2.6.3. Transformada Discreta de Fourier.

Suponiendo que una función continua  $f(x,y)$  es discretizada en una secuencia:

$$\{f(x_0), f(x_0+\Delta x), f(x_0+2\Delta x), \dots, f(x_0+[N-1]\Delta x)\} \quad (\text{Ec. 2.67})$$

Tomando  $N$  muestras uniformes espaciadas en  $\Delta x$  unidades entre sí. Por conveniencia  $x$  será utilizada para representar tanto a una variable discreta como una variable continua, lo que dependerá del contexto en que se esté trabajando.

La doble utilización de representación por  $x$  se puede hacer definiendo:

$$f(x) = f(x_0+\Delta x) \quad (\text{Ec. 2.68})$$

Así, la secuencia:

$$\{f(0), f(1), f(2), \dots, f(N-2), f(N-1)\}$$

Será utilizada para denotar el conjunto de  $N$  muestras uniformemente espaciadas de una correspondiente función continua. Al emplear la notación descrita, se tiene que el par de TF discreta que se aplica para funciones muestreadas está dado por:

$$F(\mu) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi\mu x / N} \quad (\text{Ec. 2.69})$$

Donde  $\mu = 0, 1, 2, \dots, N-2, N-1$ .

$$f(x) = \sum_{\mu=0}^{N-1} F(\mu) e^{j2\pi\mu x / N} \quad (\text{Ec. 2.70})$$

Para  $x = 0, 1, 2, \dots, N-2, N-1$ .

Los valores de  $\mu = 0, 1, 2, \dots, N-2, N-1$  en la TF discreta dada en la ec. (2.69) corresponden a muestras de la transformada continua en los valores  $0, \Delta\mu, 2\Delta\mu, \dots, (N-2) \Delta\mu, (N-1) \Delta\mu$ .

Con lo anterior se ha hecho que  $F(\mu)$  represente a  $F(\mu+\Delta\mu)$ , en una notación similar a la utilizada para  $f(x)$  discreta, con la diferencia que las muestras de  $F(\mu)$  comienzan en el origen del eje de frecuencia [GON02].

#### 2.2.6.4. Transformada Discreta de Fourier en Dos Dimensiones.

Para el caso de dos variables, el par de TF discreta está dado por:

$$F(\mu, \nu) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-2\pi j \left( \frac{\mu x}{M} + \frac{\nu y}{N} \right)} \quad (\text{Ec. 2.71})$$

Para  $\mu = 0, 1, 2, \dots, M-2, M-1$  y  $\nu = 0, 1, 2, \dots, N-2, N-1$ .

$$f(x, y) = \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{N-1} F(\mu, \nu) e^{j2\pi \left( \frac{\mu x}{M} + \frac{\nu y}{N} \right)} \quad (\text{Ec. 2.72})$$

Para  $\mu = 0, 1, 2, \dots, M-2, M-1$  y  $\nu = 0, 1, 2, \dots, N-2, N-1$ .

El muestreo de una función continua se realiza ahora con una malla bidimensional con espaciamiento de ancho  $\Delta x$  en el eje  $x$  y de ancho  $\Delta y$  en el eje  $y$ .

Tal como en el caso de una dimensión, la función discreta  $f(x,y)$  representa muestras de la función  $f(x_0 + x\Delta x + y\Delta y)$ , para valores  $x = 0, 1, 2, \dots, M-1$  e  $y = 0, 1, 2, \dots, N-1$ .

Lo mismo es válido para  $F(\mu, \nu)$ .

Los incrementos de muestreo en los dominios espacial y de frecuencia están relacionados por:

$$\Delta \omega = \frac{1}{M\Delta x} \quad (\text{Ec. 2.73})$$

$$\Delta \nu = \frac{1}{N\Delta y} \quad (\text{Ec. 2.74})$$

Cuando el muestreo se realiza mediante un arreglo cuadrado, lo que resulta en significativas simplificaciones o comodidad, se tiene  $M = N$  y:

$$F(\mu, \nu) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\mu x + \nu y)/N} \quad (\text{Ec. 2.75})$$

Para  $\mu = \nu = 0, 1, 2, \dots, N-2, N-1$ .

$$f(x, y) = \frac{1}{N} \sum_{\mu=0}^{N-1} \sum_{\nu=0}^{N-1} F(\mu, \nu) e^{j2\pi(\mu x + \nu y)/N} \quad (\text{Ec. 2.76})$$

Para  $\mu = \nu = 0, 1, 2, \dots, N-2, N-1$ .

El término  $1/N$  ha sido incluido en ambas expresiones, dado que  $F(\mu, \nu)$  y  $f(x, y)$ , constituyen un par de TF, la agrupación de éstos términos constantes multiplicativos es arbitraria.

Dado que en la práctica lo usual es la digitalización en arreglos cuadrados, las expresiones (2.73) y (2.74) se emplean de preferencia frente a las expresiones (2.71) y (2.72).

Los espectros de Fourier, de fase y de energía de funciones discretas uni y bidimensionales están dados por las expresiones (2.63) a (2.60) y (2.64) a (2.66) respectivamente. La única diferencia es que las variables independientes son discretas.

#### **2.2.6.5. Transformada Rápida de Fourier.**

La transformada rápida de Fourier es simplemente un algoritmo rápido para la evaluación numérica de integrales de Fourier, y su importancia radica en la rapidez de cálculo conseguida, importante en aplicaciones como ecualización y filtrado en equipos de audio/vídeo en tiempo real y comunicaciones, entre otros.

La implementación de la Ec. (2.69) involucra un número de sumas y multiplicaciones complejas que es proporcional a  $N^2$ .

Lo anterior se puede apreciar fácilmente ya que para cada uno de los  $N$  valores de  $\mu$ , la expansión de la sumatoria requiere  $N$  multiplicaciones complejas de  $f(x)$  por  $e^{-j2\pi\mu x / N}$  y  $(N-1)$  sumas de resultados.

El término  $e^{-j2\pi\mu x / N}$  puede ser calculado de una vez y almacenado en una tabla para las aplicaciones subsecuentes, por tal razón, la multiplicación de  $\mu$  por  $x$  en éste término no se contabiliza normalmente como parte de la implementación.

Se demuestra en lo que sigue que la descomposición de la Ec. (2.69), permite reducir el número de sumas y multiplicaciones a un valor proporcional a  $N \log_2 N$ .

El procedimiento de descomposición se denomina Algoritmo de Transformada Rápida de Fourier (FFT).

El ahorro o reducción en el número de operaciones es significativo para valores de  $N$  como lo que es doble esperar en imágenes prácticas, por ejemplo, para una imagen de  $1024 \times 1024$  píxeles,  $N = 1024$ , se tendría:

$$N^2 = 1.048.576 \text{ operaciones complejas}$$

Con FFT:

$$N2\log_2 N = 10.240 \text{ operaciones complejas}$$

Con una reducción de 102.4:1, el tiempo de cálculo, empleando máquinas equivalentes, se reduce a menos del 1%.

La descripción que sigue se refiere al desarrollo de un algoritmo de FFT para una variable. Una TF de dos variables puede ser calculada por aplicación sucesiva de la TF de una variable [GON02].

### Algoritmo para FFT

El algoritmo que se plantea está basado en el método denominado Doblamiento Sucesivo.

Para simplificar las expresiones, la Ec. (2.69) se rescribe:

$$F(\mu) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) W_N^{\mu x} \quad (\text{Ec. 2.77})$$

Donde,

$$W_N = e^{-j2\pi/N} \quad (\text{Ec. 2.78})$$

y N se supone de la forma:

$$N = 2^n \quad (\text{Ec. 2.79})$$

Con N entero positivo. Entonces puede expresarse como:

$$N = 2M \quad (\text{Ec. 2.80})$$

Con M también entero positivo. Sustituyendo (2.80) en (2.77) se obtiene:

$$F(\mu) = \frac{1}{2M} \sum_{x=0}^{2M-1} f(x) W_{2M}^{\mu x} = \frac{1}{2} \left\{ \frac{1}{M} \sum_{x=0}^{M-1} f(x) W_{2M}^{\mu(2x)} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_{2M}^{\mu(2x+1)} \right\} \quad (\text{Ec. 2.81})$$

Puesto que, de ec. (3.78),  $W_{2M}^{2\mu x} = W_M^{\mu x}$ , la ec. (3.81) se expresa como:

$$F(\mu) = \frac{1}{2} \left\{ \frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{\mu x} + \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_M^{\mu x} W_{2M}^{\mu} \right\} \quad (\text{Ec. 2.82})$$

Se define:

$$F_{par}(\mu) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x) W_M^{\mu x} \quad (\text{Ec. 2.83})$$

Para  $\mu = 0, 1, 2, \dots, M-1$  y :

$$F_{impar}(\mu) = \frac{1}{M} \sum_{x=0}^{M-1} f(2x+1) W_M^{\mu x} \quad (\text{Ec. 2.84})$$

Para  $\mu = 0, 1, 2, \dots, M-1$ , entonces la ec. (3.82) se define:

$$F(\mu) = \frac{1}{2} \left\{ F_{par}(\mu) + F_{impar}(\mu) W_{2M}^{\mu} \right\} \quad (\text{Ec. 2.85})$$

También, dado que  $W_M^{\mu+M} = W_M^\mu$  y  $W_{2M}^{\mu+M} = W_{2M}^\mu$ ,

$$F(\mu + M) = \frac{1}{2} \{ F_{par}(\mu) - F_{impar}(\mu) W_{2M}^\mu \} \quad (\text{Ec. 2.86})$$

Un análisis cuidadoso de las ecuaciones (2.83) y (2.84) muestran algunas propiedades interesantes de dichas expresiones. Nótese que una transformada de N-puntos puede ser calculada dividiendo la expresión original en dos partes, como se indica en las ecuaciones (2.85) y (2.86).

El cálculo de la primera mitad de  $F(\mu)$  requiere de la evaluación de las dos transformada de N/2 puntos según las ecuaciones (2.83) y (2.84). Los valores resultantes de  $F_{impar}(\mu)$  y  $F_{par}(\mu)$  se sustituyen en la ecuación (2.85) para obtener  $F(\mu)$  para  $\mu = 0, 1, 2, \dots, (N/2-1)$ . La otra mitad se obtiene mediante la ecuación (2.86) sin requerir evaluaciones adicionales de la transformada.

Considerando un número de muestras igual a  $2^n$ , con n entero positivo, se puede demostrar que el número de operaciones complejas (multiplicaciones y sumas) está dado por:

$$m(n) = 2m(n-1) + 2^{n-1}, \quad n \geq 1 \quad (\text{Ec. 2.87})$$

$$a(n) = 2a(n-1) + 2^n, \quad n \geq 1 \quad (\text{Ec. 2.88})$$

Expresiones recursivas que indican el número de multiplicaciones (Ec. (2.87)) y de sumas (Ec. (2.88)). Para las que  $m(0)$  y  $a(0)$  son iguales a cero, puesto que la transformada de un punto no requiere operación alguna [GON02].

### La FFT Inversa

Resulta que todo algoritmo que se implemente para calcular la FFT discreta con modificaciones simples en sus entradas, puede ser utilizado para el cálculo de la inversa.

La ecuación de la directa:

$$F(\mu) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi\mu x / N} \quad (\text{Ec. 2.89})$$

y

$$f(x) = \sum_{\mu=0}^{N-1} F(\mu) e^{j2\pi\mu x / N} \quad (\text{Ec. 2.90})$$

Para la inversa, se permite el siguiente procedimiento:

-Tomando al Ec. (2.90) en su conjugada y dividiendo ambos lados por N, resulta,

$$\frac{1}{N} f^*(x) = \frac{1}{N} \sum_{\mu=0}^{N-1} F^*(\mu) e^{-j2\pi\mu x / N} \quad (\text{Ec. 2.91})$$

Al comparar se aprecia que el lado derecho tiene la misma forma que la ecuación (2.89).

Entonces, usando  $F^*(\mu)$  como entrada para el algoritmo empleado en el cálculo de la FFT directa, el resultado que se obtiene es  $f^*(x) / N$ .

Al resultado obtenido se le conjuga (se obtiene su complejo conjugado) y se multiplica por N, resultando la inversa deseada  $f(x)$ .

Para el caso bidimensional corresponde obtener el complejo conjugado de la ec. (2.76),

resultando:

$$f^*(x, y) = \frac{1}{N} \sum_{\mu=0}^{N-1} \sum_{\nu=0}^{N-1} F^*(\mu, \nu) e^{-j2\pi(\mu x + \nu y) / N} \quad (\text{Ec. 2.92})$$

Que tiene la misma forma que la FFT directa para dos dimensiones de la ec. (2.75). Entonces, aplicando  $F^*(\mu, \nu)$  a un algoritmo desarrollado para el cálculo de la transformada directa, el resultado obtenido será  $f^*(x, y)$ , tomando el complejo conjugado de éste resultado se obtendrá  $f(x, y)$ . Naturalmente si  $f(x)$  o  $f(x, y)$  son reales, la operación de complejo conjugado es innecesaria [GON02].

## **2.3. Técnicas de Segmentación.**

Las técnicas de mejora de la imagen reciben en algunos casos el nombre de técnicas de preprocesamiento. Ahora se analizará en la línea de la extracción de información de una imagen, a esto se le denomina Análisis de Imágenes.

El primer paso de análisis de imágenes es generalmente segmentar la imagen en regiones. Una Región, en una imagen, es un grupo de píxeles conectados, que tienen propiedades similares. Es obvio que las regiones son importantes para la interpretación de las imágenes, porque pueden corresponder a objetos en la escena. Una imagen puede contener varios objetos y además cada objeto puede contener varias regiones que corresponden a partes del mismo.

Para que una imagen pueda ser interpretada correctamente, tiene que ser dividida en regiones que correspondan a objetos o partes de ellos. Sin embargo, debido a los errores de segmentación, la correspondencia entre regiones y objetos es perfecta y se debe utilizar conocimiento específico en etapas posteriores de la interpretación de las imágenes.

### **2.3.1.Regiones y Bordes.**

Se considerará una imagen simple que contenga diferentes objetos. El primer paso en el análisis para entender una imagen es particionar la imagen de forma que las regiones que representan diferentes objetos son marcadas explícitamente. Estas particiones pueden ser obtenidas a partir de las características de los niveles de gris en la imagen, o utilizando alguna otra. Así podrá decirse que los valores de gris en las posiciones de la imagen son las observaciones y que los otros atributos, como la pertenencia a regiones, deben obtenerse a partir de los niveles de gris. Existen dos aproximaciones para

particionar una imagen en regiones: Segmentación basada en regiones y estimación de la frontera utilizando detección de bordes.

En la aproximación basada en regiones, todos los píxeles que corresponden a un objeto se agrupan juntos y son marcados para indicar que pertenecen a una región. Este proceso recibe el nombre de Segmentación. Los píxeles son asignados a regiones según algún criterio que los distingue del resto de la imagen. Dos principios muy importantes en la segmentación son valor de similitud y proximidad espacial. Dos píxeles pueden ser asignados a la misma región si tienen características similares de intensidad y si están próximos. La varianza de los niveles de gris en una región y la compactificación de una región pueden ser también usadas como valores de similitud y proximidad entre píxeles respectivamente.

Los principios de similitud y proximidad aparecen por la hipótesis de que los puntos de un mismo objeto producirán en la misma imagen píxeles con niveles de gris similares y próximos espacialmente. Se podría agrupar píxeles en una imagen utilizando estos principios y usar conocimiento que depende del dominio para realizar la correspondencia objeto - región.

Al igual que se ha abordado el problema de la segmentación por similitud entre los niveles, es posible realizar este proceso mediante lo que se denomina *disparidad*. El objetivo es buscar los píxeles que se encuentran en las fronteras de las regiones. Estos píxeles, llamados *Bordes*, pueden localizarse examinando los píxeles vecinos. Puesto que los píxeles llamados bordes están en la frontera de las regiones y éstas suelen tener diferentes niveles de gris a cada lado, lo que necesitamos es medir las diferencias entre los píxeles vecinos. La mayoría de los detectores de bordes usan sólo las características de la intensidad para su detección, aunque algunos aspectos como la textura y el movimiento también aportan información.

En imágenes reales, una región estará acotada por un contorno cerrado. En principio, la segmentación de regiones y detección de bordes debería producir los mismos resultados. Las fronteras (cerradas) podrían obtenerse a partir de las regiones utilizando un algoritmo de seguimiento de fronteras. Análogamente, las regiones pueden obtenerse a partir de los bordes usando un algoritmo de relleno. Por desgracia, en las imágenes reales es raro obtener tanto bordes a partir de regiones como regiones a partir de bordes, debido al ruido y otros factores, ni la segmentación de regiones ni la detección de bordes proporcionan información perfecta.

Se realizará una subdivisión adicional, los métodos de segmentación serán divididos en tres grupos en función de los rasgos dominantes que emplean. Primero *segmentación basada en fronteras*, para continuar con la *segmentación basada en bordes* y la *segmentación basada en regiones*.

### 2.3.2. Segmentación Basada en Fronteras.

Los primeros pasos tratan de identificar rasgos en la imagen que son relevantes para estimar la estructura y propiedades de los objetos en la escena. Los bordes corresponden a cambios locales significativos en una imagen y son probablemente los rasgos más importantes para el análisis de la misma. Estos se encuentran en la frontera entre dos regiones diferentes, siendo su detección un paso importante para la recuperación de información.

Un borde es una discontinuidad de algún tipo en la función de intensidad de la imagen. Pero existen muchos tipos de bordes diferentes, algunos son debido a las sombras que se producen sobre los objetos, otros a la variación de la reflectancia de los objetos o incluso por su textura.

Un borde se puede definir como un cambio local significativo en la intensidad de la imagen que usualmente está asociado con una discontinuidad de la imagen o su derivada. Las discontinuidades pueden ser de dos tipos: 1. de *Salto* donde la intensidad cambia abruptamente de un valor en un lado de la discontinuidad a otro valor diferente en el lado opuesto, 2. discontinuidades de *línea* donde las intensidades cambian abruptamente volviendo en poco espacio a su valor inicial. Es claro que los saltos y pasos son raros en las imágenes, debido en parte al proceso de digitalización.

**Definición. 1.** Un punto de borde es un punto geométrico con coordenadas  $[i,j]$  en la posición de un cambio de intensidad local significativo en la imagen.

**Definición. 2.** Un fragmento de borde es la  $i$  y  $j$  coordenada de un borde y la orientación  $\theta$  que puede ser el ángulo del gradiente.

**Definición. 3.** Un detector de bordes es un algoritmo que produce un conjunto de bordes a partir de la imagen.

**Definición. 4.** Un contorno es una lista de bordes, o la curva matemática que modela la lista de bordes.

**Definición. 5.** La unión de los bordes es el proceso de formar una lista ordenada de bordes a partir de una no ordenada.

**Definición. 6.** El seguimiento de bordes es el proceso de búsqueda en la imagen (filtrada) para determinar los contornos.

Las coordenadas de un punto borde pueden ser los índices de la fila y la columna donde el borde fue detectado o bien las coordenadas a otra resolución. Las coordenadas pueden venir dadas en el sistema de coordenadas de la imagen, pero más probablemente vendrán dadas en el sistema de coordenadas producido por el filtro de detección. El término borde se usa tanto para puntos como para fragmentos.

El conjunto de bordes que produce un detector de fronteras puede dividirse en dos clases: *correctos* y los *falsos*. Un tercer conjunto recibe el nombre de *falsos negativos* mientras que los *falsos* corresponden a *falsos positivos*.

La diferencia entre la unión de fronteras y el seguimiento de fronteras es que la unión tiene como entrada un conjunto desordenado de bordes producido por un detector de bordes y el resultado es una lista ordenada. El seguimiento de bordes toma como entrada una imagen y produce una lista ordenada de bordes, este proceso puede utilizar información global.

### 2.3.2.1. Métodos de Gradiente.

La detección de bordes es una operación de detección de cambios de intensidad locales significativos. En una dimensión éstos cambios, los de salto, están asociados con un máximo local de la primera derivada. Se verá a continuación su equivalente bidimensional. En las últimas décadas se han desarrollado muchos detectores de fronteras. Es fundamental entender ahora que puesto que de lo que se dispone es una imagen y no la escena real, se deben hacer aproximaciones de las derivadas basadas en diferencias finitas.

#### 2.3.2.2.1. Operadores Isotrópicos.

Un operador isotrópico tiene como objetivo determinar la fuerza de un borde con independencia de su dirección en la imagen. Es obvio que ésta propiedad es deseable entre los detectores de bordes ya que no sería bueno que un detector sólo fuera sensible a bordes en determinadas direcciones. Este tipo de detectores es aplicado a cada punto de la imagen para calcular la fuerza del borde independientemente de su orientación.

Se considerará una imagen de intensidades  $f(x,y)$  con coordenadas  $x$  e  $y$ . Se supondrá que la rotamos. Se considerará un punto  $(x,y)$  y su rotación está dada por  $\theta$  que produce  $(x',y')$  según:

$$x' = x\cos\theta + y\sin\theta \quad (\text{Ec. 2.93})$$

$$y' = -x\sin\theta + y\cos\theta \quad (\text{Ec. 2.94})$$

Las derivadas de los dos sistemas se relacionan mediante:

$$\frac{\partial f}{\partial x'} = \frac{\partial f}{\partial x} \cos\theta + \frac{\partial f}{\partial y} \sin\theta \quad (\text{Ec. 2.95})$$

y

$$\frac{\partial f}{\partial y'} = -\frac{\partial f}{\partial x} \operatorname{sen} \theta + \frac{\partial f}{\partial y} \cos \theta \quad (\text{Ec. 2.96})$$

Si se está interesado en operadores isotrópicos, entonces un análisis matemático simple prueba que derivadas de orden par o de orden impar elevadas a potencias pares pueden dar lugar a operadores isotrópicos. Puede probarse que un operador isotrópico es la suma de los cuadrados de las derivadas parciales en las direcciones  $x$  e  $y$  ya que:

$$\left(\frac{\partial f}{\partial x'}\right)^2 + \left(\frac{\partial f}{\partial y'}\right)^2 = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \quad (\text{Ec. 2.97})$$

Para encontrar la dirección en la que se produce el cambio, la dirección del borde, se considera la ecuación para la derivada parcial en dirección  $x'$ . Se puede entonces obtener el  $\theta$  para el que esta derivada parcial se maximiza.

Para esto se deberá resolver:

$$d\left(\frac{\partial f}{\partial x'}\right) / d\theta = 0 \quad (\text{Ec. 2.98})$$

Donde

$$\theta = \operatorname{arctg} \left( \frac{\partial f}{\partial y} \right) / \left( \frac{\partial f}{\partial x} \right) \quad (\text{Ec. 2.99})$$

Sustituyendo éste  $\theta$  en la fórmula de la derivada parcial tendremos que el máximo cambio vale:

$$M = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2} \quad (\text{Ec. 2.100})$$

Las expresiones anteriores tienen operaciones de raíces cuadradas y cuadrados que pueden ser computacionalmente lentos. Es por ello que se realizan aproximaciones de éstas funciones que son costosas en cuanto a tiempo de ejecución.

### 2.3.2.2.2. Operador de Roberts.

El operador de Roberts es una de los más antiguos. Es muy fácil de calcular ya que usa sólo una ventana o máscara de tamaño 2x2. sus máscaras de convolución son:

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (\text{Ec. 2.101})$$

La magnitud del gradiente se calcula entonces mediante la siguiente aproximación

$$G(f(i, j)) = |f(i, j) - f(i+1, j+1)| + |f(i, j+1) - f(i+1, j)| \quad (\text{Ec. 2.102})$$

Puesto que  $G_x$  y  $G_y$  son aproximaciones de las derivadas parciales con respecto a  $x$  e  $y$  respectivamente.

Una gran desventaja del operador de Roberts es que es muy sensible al ruido, ya que se usan muy pocos píxeles para aproximar el gradiente, además, la aproximación está desplazada del punto  $(i, j)$  al  $(i+1/2, j+1/2)$ .

### 2.3.2.2.3. Operador de Sobel.

Una forma de evitar que el gradiente se calcule en un punto intermedio, tal y como ocurre en el operador de Roberts, es usar entornos de tamaño 3x3. Considerando la distribución de píxeles que se muestra alrededor del  $(i, j)$  en la Figura 2.11. El operador de Sobel es la magnitud del gradiente calculado mediante:

$$M = \sqrt{s_x^2 + s_y^2} \quad (\text{Ec. 2.103})$$

Donde las derivadas parciales se calculan mediante:

$$s_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6) \quad (\text{Ec. 2.104})$$

$$s_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4) \quad (\text{Ec. 2.105})$$

Donde la constante  $c = 2$ .

Es importante notar que este operador le da más peso a los píxeles más cercanos al centro de la máscara. Este operador es uno de los más usados.

$$\begin{array}{ccc} a_0 & a_1 & a_2 \\ a_7 & (i,j) & a_3 \\ a_6 & a_5 & a_4 \end{array}$$

Figura 2.11: Etiquetado de niveles de gris para explicar los detectores de bordes.

#### 2.3.2.2.4. Operador Prewitt.

Este operador es similar al de Sobel, reemplazando la constante  $c$  por 1.

#### 2.3.2.1. Pasos en la Detección de Bordes.

Se resume en tres pasos que son los siguientes:

**1. Filtrado.** Puesto que el cálculo del gradiente basado en los valores de intensidad de un número finito de puntos es muy susceptible al ruido, normalmente se filtra la imagen inicialmente para mejorar el rendimiento del detector con respecto al ruido. Es obvio que éste proceso ha de aplicarse con cuidado, puesto que el emborronamiento oculta los bordes.

**2. Realce.** Este paso es esencial para determinar cambios en la intensidad de los vecinos de un punto. El proceso es básicamente el cálculo del gradiente.

**3. Detección.** Muchos puntos en una imagen tienen un gradiente no nulo. Obviamente no todos ellos son bordes para todas las aplicaciones. Es necesario algún método para determinar que puntos son bordes. Un criterio ampliamente utilizado es la umbralización.

**4. Estimación.** Esta deberá llevarse a cabo en la resolución apropiada y en algunos casos será necesario también estimar la dirección del gradiente.

### 2.3.2.2. Operadores de Segunda Derivada.

Los detectores de fronteras calculan el gradiente y si su valor está por encima de un umbral, se supone la existencia de un punto de borde. El resultado es demasiados puntos de borde. Una mejor aproximación sería encontrar sólo los puntos que tienen máximos locales en el gradiente y considerar éstos como puntos de borde. Esto significa que en los puntos de borde existe un máximo en la primera derivada y equivalentemente un cruce por cero en la segunda. Así los puntos de borde se detectan encontrando los cruces por cero de la segunda derivada en las intensidades de la imagen.

#### 2.3.2.2.1. Operador Laplaciano.

El Laplaciano es el equivalente bidimensional de la segunda derivada. La fórmula para el Laplaciano de una función  $f(x,y)$  es:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (\text{Ec. 2.106})$$

La segunda derivada en las direcciones  $x$  e  $y$  se aproximan utilizando las ecuaciones en diferencias:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial f}{\partial x} = \frac{\partial(f(i+1, j)) - f(i, j)}{\partial x} = f(i+2, j) - 2f(i+1, j) + f(i, j) \quad (\text{Ec. 2.107})$$

Análogamente:

$$\frac{\partial^2 f}{\partial y^2} = f(i, j+2) - 2f(i, j+1) + f(i, j) \quad (\text{Ec. 2.108})$$

Centrando estas aproximaciones en la posición (i,j), entonces más razonable sería:

$$\frac{\partial^2 f}{\partial x^2} = f(i+1, j) - 2f(i, j) + f(i-1, j) \quad (\text{Ec. 2.109})$$

$$\frac{\partial^2 f}{\partial y^2} = f(i, j+1) - 2f(i, j) + f(i, j-1) \quad (\text{Ec. 2.110})$$

Combinando éstas dos ecuaciones en un único operador, podemos usar la siguiente máscara para aproximar el Laplaciano.

$$\nabla^2 \approx \begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix} \quad (\text{Ec. 2.111})$$

Algunas veces es deseable darle más peso a los píxeles centrales del entorno.

Una aproximación del Laplaciano que hace esto es:

$$\nabla^2 \approx \begin{matrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{matrix} \quad (\text{Ec. 2.112})$$

El operador Laplaciano marca la presencia de un borde cuando la salida del operador realiza una transición por cero. Los ceros triviales, es decir, las regiones de cero uniforme tienen que ser ignoradas. En principio, las localizaciones de los cruces por cero pueden ser estimadas a nivel de subpíxeles usando interpolación lineal, pero los resultados pueden ser inexactos.

### 2.3.2.2. Cruces por Cero de la Segunda Derivada. Laplaciana de Gaussiana.

Una técnica de detección de bordes basada en los cruces por cero de la segunda derivada, explora el hecho de que un borde de salto corresponde a un cambio brusco de la función de la imagen. La primera derivada de la función de la imagen debe tener un extremo en la posición correspondiente a un borde en la imagen, y la segunda derivada debe ser cero en la misma posición. Sin embargo, es mucho más fácil y preciso encontrar un cruce por cero que un extremo. Ver Figura (2.12), aquí se ilustra el problema en el caso unidimensional.

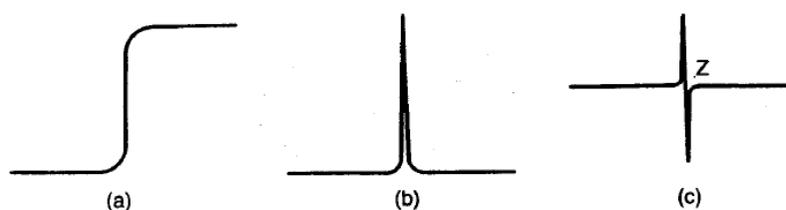


Figura 2.12: a) Intensidad, b) Primera Derivada, c) Segunda Derivada.

El punto es cómo calcular la segunda derivada de la forma más robusta. Una posibilidad es suavizar la imagen primero, y después calcular la segunda derivada. Cuando se selecciona un filtro de alisamiento se deben tener en cuenta dos criterios: debe ser suave y de banda limitada para reducir el número de frecuencias en las que puede producir el cambio y segundo que la respuesta sea casi puntual. Aunque éstos criterios entran en conflicto, la distribución que optimiza simultáneamente ambos es la distribución Gaussiana.

El filtro de alisamiento Gaussiano 2D,  $G(x,y)$  está dado por:

$$G(x, y) = c \exp\left[-\frac{x^2 + y^2}{2\sigma^2}\right] \quad (\text{Ec. 2.113})$$

Donde  $x$  e  $y$  son las coordenadas de la imagen,  $\sigma$  es la desviación típica de la distribución de probabilidad y  $c$  es una constante de normalización.

Se quiere obtener la segunda derivada en una versión más suave, de la IO  $f(x,y)$ . Se sabe que el operador Laplaciano  $\nabla^2$  produce la segunda derivada y además es isotrópico. Entonces el Laplaciano de una imagen  $f(x,y)$  suavizada por una Gaussiana es :

$$\nabla^2(G(x,y,\sigma) * f(x,y)) \quad (\text{Ec. 2.114})$$

El orden en el que se realiza la convolución y diferenciación puede ser intercambiado debido a la linealidad de los operadores, se tiene:

$$(\nabla^2 * G(x,y,\sigma)) * f(x,y) \quad (\text{Ec. 2.115})$$

Si el filtro de convolución se denota  $h(x,y,\sigma)$ , entonces se tiene:

$$h(x,y,\sigma) = \frac{c}{\sigma^2} \left( \frac{x^2 + y^2}{\sigma^2} - 2 \right) \exp \left[ -\frac{x^2 + y^2}{2\sigma^2} \right] \quad (\text{Ec. 2.116})$$

Que recibe el nombre de filtro de *sombrero mejicano*.

Este proceso de búsqueda de la segunda derivada es muy robusto. El aislamiento Gaussiano suprime la influencia de píxeles lejanos y el Laplaciano es una medida eficiente y estable de los cambios en una imagen.

La localización en la imagen  $\nabla^2 G$  donde se cruza por cero corresponde a la posición de los bordes. La ventaja de ésta aproximación comparada con los clásicos operadores de bordes es que se cubre una mayor área que rodea al píxel. La influencia de los píxeles distantes está determinada por  $\sigma$ .

### **2.3.3. Segmentación Basada en Bordes.**

Históricamente el primer grupo de métodos de segmentación que es aún muy utilizado es el basado en información sobre bordes en la imagen. Las técnicas de segmentación basadas en bordes se basan en los bordes encontrados en la imagen por los detectores. Los métodos de extracción de fronteras que no pueden ser utilizados como resultado de segmentación por sí solos, otros pasos posteriores se han de seguir para combinar los bordes en cadenas de bordes que corresponden con fronteras de objetos en la imagen. El objetivo final es alcanzar al menos una segmentación parcial, es decir, agrupar bordes locales en una imagen donde sólo cadenas de bordes con una correspondencia con objetos en la imagen o partes de imágenes están presentes.

Existen diferentes métodos para la segmentación basada en bordes que difieren en la estrategia para la construcción de la frontera, así como en la cantidad de información previa que se incorpora en el método. Es obvio que a mayor información mejor es la segmentación.

#### **2.3.3.1. Relajación de Bordes.**

Las fronteras que se obtienen de los métodos descritos en las secciones anteriores están usualmente muy afectados tanto por el ruido de la imagen como por la no presencia de partes que deberían estar presentes. Si en el proceso de paso de borde a frontera se tiene en cuenta las propiedades de los bordes vecinos presentes, es posible aumentar la calidad de los resultados. Todas las propiedades de una imagen junto con la posibilidad de existencia de más bordes, se evalúan iterativamente con más precisión hasta que el contexto de los bordes queda claro. Basado en la fuerza de bordes en una vecindad local especificada, la confianza de cada borde se aumenta o disminuye. Un borde

débil posicionado entre dos bordes fuertes proporciona un ejemplo de contexto, es altamente probable que éste borde débil entre ellos sea parte de una frontera. Si, por otra parte, un borde, incluso fuerte no tiene contexto que lo soporte, puede que no sea parte de ninguna frontera.

Este método es un ejemplo clásico de evaluación de contexto de bordes. El método utiliza bordes tipo grieta (entre píxeles). Contexto del borde se considera a ambos lados de un borde, e, con el sistema de vecinos que se muestra en la figura 2.13.

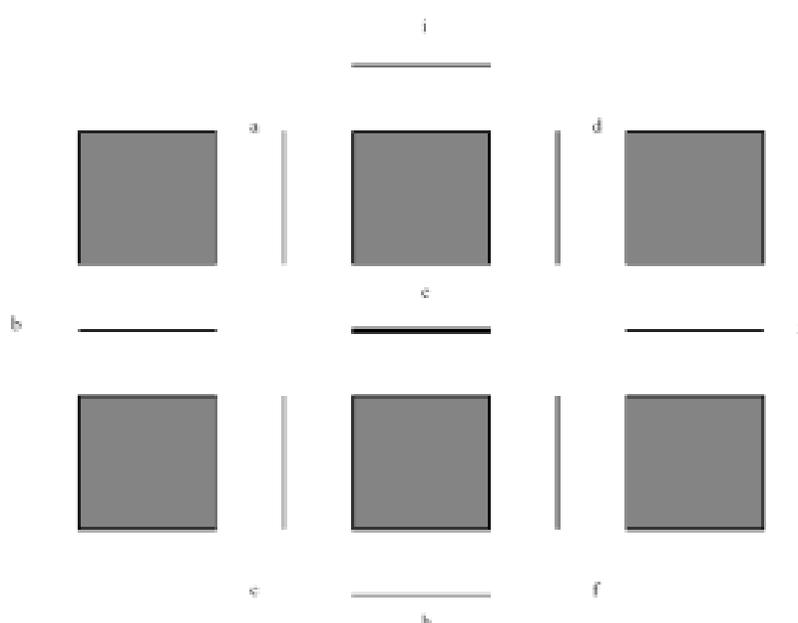


Figura 2.13: Borde tipo grieta e con sus vecinos.

Todas las posibles posiciones de los vecinos de e deben ser incluidas para cubrir la forma en la que se continua en el entorno. Además dos posiciones paralelas con e pueden ser incluidas en el sistema de vecinos, éstas posiciones paralelas compiten con e en la colocación del borde. La relajación de bordes busca la construcción de fronteras continuas de forma que se discutirán los patrones que se pueden encontrar en el entorno. Dado un borde central e, éste tiene un vértice a cada uno de sus lados y cada uno de éstos vértices tiene tres posibles formas de continuar. La forma de continuar de un vértice se evalúa de

acuerdo al número de bordes que de él emanan, sin contar e, y éste número define el tipo de vértice. Entonces el tipo de borde e puede ser representado mediante  $i - j$ , donde i notará la configuración de bordes que se produce a su izquierda (en un vértice izquierdo) y j el que hay a la derecha. Por ejemplo se asignará el tipo 0 – 0 la configuración que se muestra en la Figura 2.14-a), el 2 – 0 para la figura 2.14-c). Es obvio que por simetría se debe considerar sólo el caso  $i \leq j$ . Tenemos las siguientes situaciones de contexto:

- 0 – 0. borde aislado, influencia negativa en la confianza del borde.
- 0 – 2, 0 – 3. final sin salida, influencia negativa en la confianza del borde.
- 0 – 1. incierto, poco positivo o no influencia.
- 1 – 1. continuación, influencia fuertemente positiva en la confianza del borde.
- 1 – 2, 1 – 3. continuación a la intersección de bordes, influencia medio positiva sobre la confianza del borde.
- 2 – 2, 2 – 3, 3 – 3. puente entre fronteras, no necesario para la segmentación no tiene influencia en la confianza del borde.

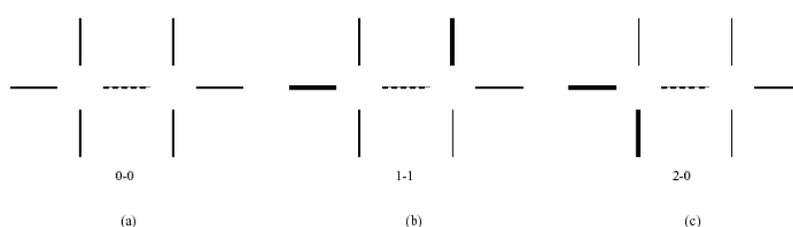


Figura 2.14: Borde tipo grieta e con sus vecinos.

Una relajación de bordes es un método iterativo, en la que la confianza de los bordes tiende a cero (desaparición del borde) o a uno (los bordes forman una frontera). La confianza  $c^{(1)}(e)$  de cada borde e en la primera iteración puede definirse como una magnitud normalizada del borde tipo grieta, con una

normalización basada en el máximo global de los bordes tipo grieta de toda la imagen, o en el máximo local de algún entorno grande del borde, en éste último caso el objetivo es disminuir la influencia de unos pocos valores muy altos de la magnitud del borde en la imagen.

El algoritmo funcionaría como se describe a continuación: **Relajación de Bordes:**

1. Evaluar la confianza  $c^{(1)}(e)$  para todos los bordes tipo grieta de la imagen,
2. Encontrar el tipo de borde de cada borde basado en la confianza de los bordes de sus vecinos,
3. Actualizar la confianza  $c^{k+1}(e)$  de cada borde  $e$  de acuerdo con su tipo y su confianza previa,
4. Parar si todas las confianzas de los bordes han tendido a cero o uno. Repetir los pasos 2 y 3 en caso contrario.

Los pasos principales del algoritmo anterior son la evaluación de los tipos de vértices seguido de la evaluación del tipo de borde y la forma en que se modifica la confianza en el borde. Un vértice es considerado de tipo  $i$  si:

$$\text{tipo}(i) = \text{máx} (\text{tipo}(k)), k=0,1,2,3$$

$$\text{tipo}(0) = (m - a) (m - b) (m - c)$$

$$\text{tipo}(1) = a(m - b) (m - c)$$

$$\text{tipo}(2) = ab(m - c)$$

$$\text{tipo}(3) = abc$$

Donde  $a, b, c$  son los valores normalizados de los otros bordes y sin pérdida de generalidad podemos suponer que  $a \leq b \leq c$ ,  $q$  es una constante para la que un valor de  $0, 1$  parece apropiado y  $m = \text{máx}(a, b, c, q)$ . Se observa que la introducción de la cantidad  $q$  asegura que  $\text{tipo}(0)$  es no cero para pequeños valores de  $a$ .

Por ejemplo si elegimos  $q = 0, 1$  un vértice  $(a, b, c) = (0,5, 0,5, 0,5)$  es un vértice de tipo uno, mientras que un vértice  $(0,3, 0,2, 0,2)$  es un vértice tipo tres. Los mismos resultados se pueden obtener observando al número de bordes de un vértice que están por encima de un umbral. El tipo de borde se encuentra como la concatenación del tipo de vértices y la confianza de bordes se modifica como sigue:

$$\text{Aumento de confianza } c^{k+1}(e) = \min(1, c^k(e) + \delta)$$

$$\text{Disminución de confianza } c^{k+1}(e) = \max(0, c^k(e) - \delta)$$

Donde  $\delta$  es una constante apropiada, usualmente en el rango  $[0,1, 0,3]$ .

Las reglas para la modificación de la confianza del borde pueden simplificarse y usar un único  $\delta$ , sin incluir el crecimiento / decrecimiento débil, moderado o fuerte. Además, los vértices tipo 2 y 3 pueden considerarse del mismo tipo de implementación.

La relajación de bordes, tal y como ha sido descrita, mejora rápidamente el etiquetado original de bordes en unas pocas iteraciones, sin embargo con muchas iteraciones empeora el resultado. La razón para ésta conducta extraña está en la búsqueda de un máximo global del criterio de consistencia sobre toda la imagen, lo cual puede no ser óptimo localmente. Una solución es asignar a la confianza de un borde cero si está por debajo de un umbral y uno si está por encima de otro. Por tanto se agrega un paso adicional al cálculo de la confianza:

$$\text{Si } c^{k+1}(e) > T_1 \text{ asignar entonces } c^{k+1}(e) = 1 \quad (\text{Ec. 2.117})$$

$$\text{Si } c^{k+1}(e) < T_2 \text{ asignar entonces } c^{k+1}(e) = 0 \quad (\text{Ec. 2.118})$$

Donde  $T_1$  y  $T_2$  son parámetros que controlan la convergencia del algoritmo.

### 2.3.3.2. Extracción de Frontera.

Las fronteras son bordes unidos que caracterizan la forma de un objeto. Son por lo tanto, útiles para calcular rasgos geométricos como tamaño u orientación.

#### Conectividad.

Conceptualmente, las fronteras pueden encontrarse trazando los bordes conectados (4 u 8 conectados,  $N_4(p)$  o  $N_8(p)$ ) (Ver sección 2.1.3). Sin embargo hay dificultades asociadas con éstas definiciones de conectividad, como muestra la Figura 2.15. si utilizamos la 4 – adyacencia, los segmentos 1,2,3 y 4 serían clasificados como disjuntos, aunque obviamente parecen formar un anillo. Bajo la 8 – adyacencia éstos segmentos estarían conectados, pero también lo estaría el interior con el exterior del anillo. En principio, ésto podría resolverse utilizando tipos de conectividad distinta para los objetos y para el fondo, pero obviamente hay otros problemas.

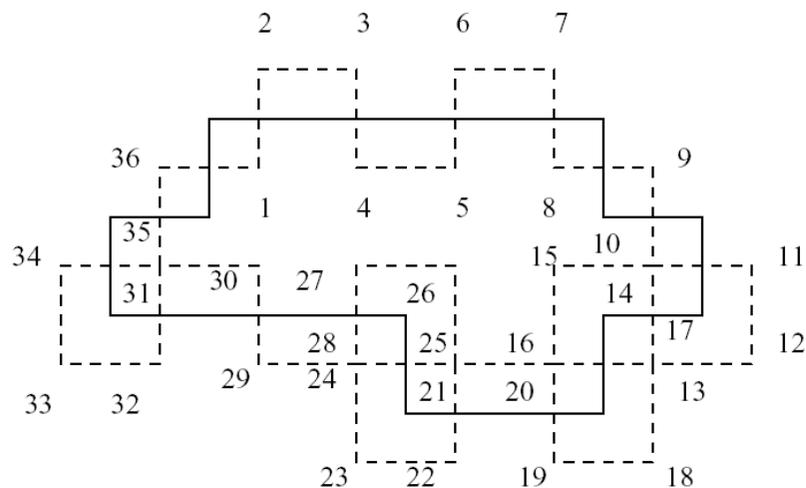


Figura 2.15: Ejemplo de seguimiento de contorno.

### **Seguimiento del Contorno.**

Como el nombre lo indica, los algoritmos de seguimiento de contorno trazan las fronteras ordenando los puntos de tipo borde sucesivos. Un algoritmo simple para trazar fronteras cerradas en imágenes binarias es el que sigue:

1. Comenzar dentro de la región A, (por ejemplo el primer píxel que encontramos de la región cuando hacemos un rastreo por filas),
2. Girar a la izquierda y pasar al píxel siguiente si estamos dentro de la región A, en caso contrario girar a la derecha y pasar al píxel siguiente,
3. Continuar hasta que se llegue al punto del que se partió.

### **Unión de Bordos y Búsqueda Heurística en Grafos.**

Una frontera puede ser también vista como un camino que se forma a través de un grafo construido uniendo los elementos de los bordes. Las reglas de unión dan la forma para conectar los bordes. Suponiendo que un grafo con nodos en las posiciones  $x_k$ ,  $k = 1, 2, \dots$  se forma desde el nodo A al B, suponiendo que se tiene una función de evaluación  $\phi(x_k)$  que da el valor del camino de A a B cuando pasa por el nodo  $x_k$ . En algoritmos de búsqueda heurística, se examinan los sucesores del nodo de salida A y se selecciona el que maximiza  $\phi(x_k)$ . El nodo seleccionado se convierte ahora en el nodo de salida y se repite el proceso hasta que se alcance B. La sucesión de nodos seleccionada constituye la frontera. Obsérvese que el algoritmo no produce un óptimo global.

### **2.3.4. Segmentación Orientada a Regiones.**

El objetivo de los métodos de segmentación descritos antes fue encontrar bordes entre regiones. Los métodos que se analizarán involucran directamente a las regiones.

Las regiones resultantes de los procesos de segmentación que ahora se examinarán deberán ser homogéneas y maximales, donde maximal significa que los criterios de homogeneidad no se cumplirán tras la unión de una región con alguna de sus adyacentes.

El criterio de homogeneidad utiliza el nivel de gris medio de la región, algunas propiedades simples o un vector de medias para imágenes multibandas.

#### **2.3.4.1. Unión de Regiones.**

El método más natural para el crecimiento de las regiones es comenzar considerando cada píxel de la imagen como una región, obviamente en éste caso cada región es homogénea, pero no necesariamente cumple el criterio de ser las regiones maximales, el proceso habrá de repetirse hasta que las regiones sean maximales. El algoritmo se describe a continuación:

1. Definir una segmentación inicial que cumpla el criterio de homogeneidad,
2. Definir un criterio para unir regiones adyacentes,
3. Unir las regiones sin romper el criterio de homogeneidad.

La implementación más simple de éste algoritmo empieza con la unión de regiones comenzando la segmentación con regiones 2x2, 4x4 u 8x8. las descripciones de las regiones se basan entonces en propiedades estadísticas de los niveles de gris, como por ejemplo en el histograma, la descripción de una región se compara con la de otra adyacente, si conciben se unen las regiones y

se calcula el nuevo descriptor de la región. En caso contrario las regiones se marcan como que no se pueden unir. El proceso continua.

En general podría decirse que lo más importante es seleccionar el criterio para realizar la unión. Un criterio de tipo heurístico es:

Unir dos regiones  $R_i$  y  $R_j$  si  $w/P_m > \theta$  donde  $P_m = \min(P_i, P_j)$   $P_i$  y  $P_j$  son los perímetros de  $R_i$  y  $R_j$  y  $W$  es el número de localizaciones de fronteras débiles, (píxeles que a ambos lados tienen una diferencia menor que un cierto umbral).

#### **2.3.4.2. Unión de Regiones por Agregación de Píxeles.**

Como es obvio el crecimiento de regiones es un procedimiento que agrupa píxeles o subregiones en regiones mayores. La forma más sencilla de éste proceso es la llamada agregación de píxeles que comienza con un conjunto de píxeles semilla y a partir de ellos hace crecer la región agregando a dichos píxeles semilla aquellos vecinos que tienen propiedades similares (nivel de gris, por ejemplo).

Para ilustrar ese procedimiento consideremos la Figura 2.16, donde los números dentro de las celdas representan niveles de gris. Sean los puntos (3,2) y (3,4) las semillas a considerar. Si se utilizan éstos dos puntos semilla se tendrá una segmentación que consiste en dos regiones a los más: una región  $R_1$  asociada con la semilla (3,2) y otra  $R_2$  asociada con (3,4). La propiedad  $P$  que se utilizará para incluir un píxel en una región es que la diferencia de un valor absoluto entre los niveles de gris del píxel y la semilla sea menor que un umbral  $T$ . Cualquier píxel que cumpla la propiedad dada con relación a los píxeles es asignado arbitrariamente a una de las regiones. La Figura 2.16-b) muestra el resultado para  $T = 3$ . las dos regiones segmentadas se notan a y b.

Con  $T = 8$  se tendría una única segmentación como se muestra en la Figura 2.16-c).

Obviamente el método descrito presenta algunos inconvenientes adicionales es la determinación de un criterio de parada. No sólo de debería tener en cuenta en el criterio de parada el nivel de gris, conceptos como forma, tamaño, entre otros son también importantes.

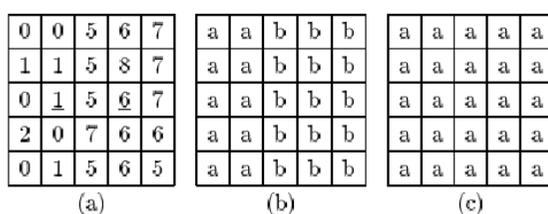


Figura 2.16: Ejemplo de crecimiento de regiones a partir de puntos semilla.

### 2.3.4.3. Separación y Unión de Regiones.

El procedimiento que se acaba de describir hace crecer las regiones a partir de un conjunto de puntos semilla. Una alternativa es subdividir una imagen inicialmente en un conjunto de regiones disjunta y luego unir las o separarlas en un intento de que todas ellas:

1. Cubran la imagen completa.
2. Cada una de ellas esté conectada.
3. Sus intersecciones dos a dos sea el conjunto vacío.
4. Que algún predicando de homogeneidad sea verdadero en esa región. Y
5. Que la unión de dos cualquiera de ellas no cumpla algún predicando de homogeneidad.

Sea  $R$  la imagen entera y  $P$  un predicando de homogeneidad. Suponiendo que se tiene una imagen cuadrada, una aproximación para segmentar  $R$  es subdivide la imagen sucesivamente en cuadrantes menores, de forma que se

tendrá que para cada región  $R_i$ ,  $P(R_i) = V$ . ésto es, si  $P(R) = F$ , se dividirá el cuadrante en subcuadrantes y se repetirá el proceso con los subcuadrantes obtenidos. Esta técnica de división puede representarse muy bien a través de los llamados *quad-trees*, esto es, un árbol en el que cada nodo tiene cuatro descendientes, ver Figura 2.17. La raíz del árbol corresponde a la IO y cada nodo es una subdivisión. En éste caso, sólo  $R_4$  se dividió más.

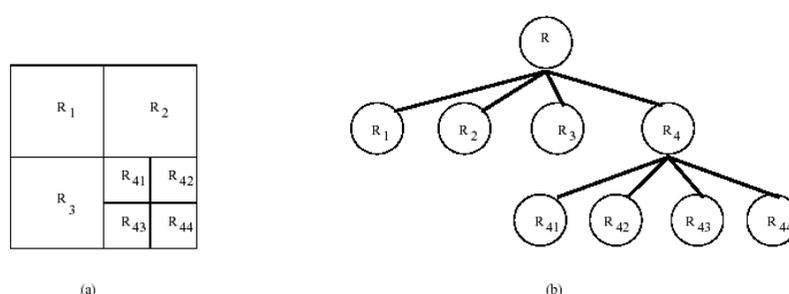


Figura 2.17: Ejemplo de quad-tree.

Si se realizan procesos de separación, se podría terminar con regiones idénticas que fueran adyacentes pero no identificadas como una misma región. Este inconveniente puede resolverse si además de la separación se permite la unión. La idea es unir ahora regiones adyacentes cuyos píxeles combinados satisfagan el predicado  $P$ , esto es, dos regiones adyacentes  $R_j$  y  $R_k$  son unidas cuando y sólo cuando  $P(R_j \cup R_k) = V$ .

Lo anterior se puede resumir con el procedimiento siguiente:

1. Dividir en cuatro regiones disjuntas cualquier región  $R_i$  para la que  $P(R_i) = F$ ,
2. Unir cualesquiera de dos regiones adyacentes  $R_j$  y  $R_k$  para las que  $P(R_j \cup R_k) = V$ ,
3. Parar cuando no sea posible realizar más uniones o divisiones.

La Figura 2.17 ilustra el algoritmo de división y unión. La imagen consiste en un objeto único con fondo, con nivel de gris constante en el fondo como en el objeto. El predicando utilizado es  $P(R_i) = V$  si todos los píxeles en la región  $R_i$  tienen la misma intensidad.

Como  $P(R) = F$  para la imagen completa, se dividió, como se muestra en la Figura 2.18-a). En el paso siguiente, sólo la región en la zona superior izquierda cumple el predicado y por lo tanto no es cambiada. Las otras tres sí lo son, ver Figura 2.18-b). En éste momento se puede realizar un proceso de unión que no puede llevarse a cabo con los dos subcuadrantes que contienen la parte baja del objeto. La Figura 2.18-c) muestra el resultado de la operación de partición y unión. En éste punto todas las regiones satisfacen  $P$  y la unión de las regiones obtenidas por la última división produce el resultado final que se muestra en la Figura 2.18-d).

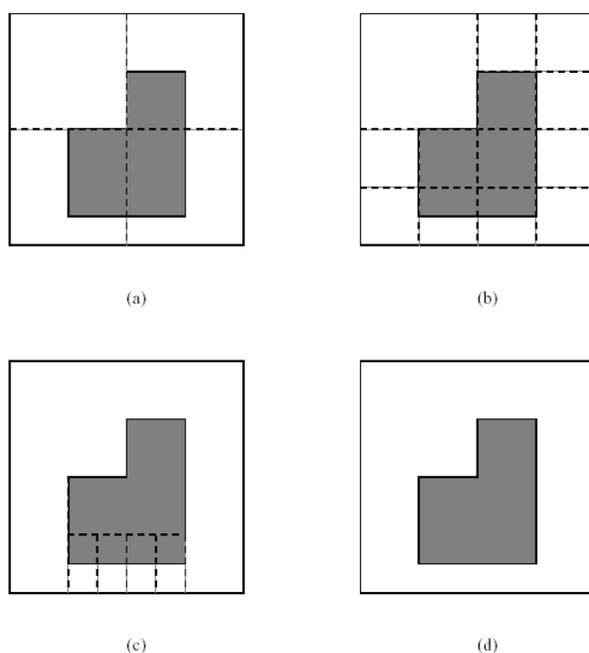


Figura 2.18: Proceso de separación y unión usando quad-trees.

### 2.3.5. Transformada Wavelets Discreta.

La Transformada Wavelets Discreta (DWT) (también conocida como Haar Wavelets) aplicada a imágenes proporciona una matriz de coeficientes, conocidos como coeficientes Wavelets. Si a una imagen le aplicamos la DWT obtenemos cuatro tipos de coeficientes: aproximaciones (A), detalles horizontales (H), detalles verticales (V) y detalles diagonales (D), ver Figura 2.19. La aproximación contiene la mayor parte de la energía de la imagen, es decir, la información más importante, mientras que los detalles tienen valores próximos a cero [OHM99].

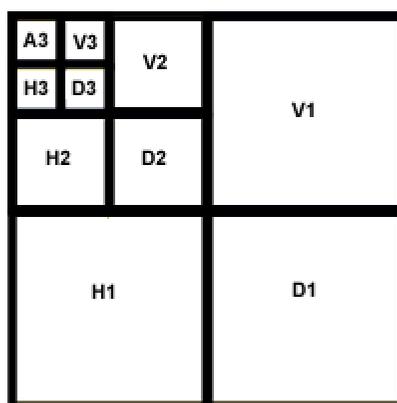


Figura 2.19: Esquema de los coeficientes Wavelet.

Generalmente, la energía de las imágenes se concentra en las frecuencias bajas. Una imagen tiene un espectro que se reduce con el incremento de las frecuencias. Estas propiedades de las imágenes quedan reflejadas en la transformada Wavelets discreta de la imagen. Los niveles más bajos de compresión se corresponden con las bandas de alta frecuencia. En particular, el primer nivel representa la banda de más alta frecuencia y el nivel más fino de resolución. A la inversa, el último nivel (n) de descomposición corresponde con la banda de frecuencia más baja y la resolución más tosca. Así, al desplazarse

de los niveles más altos a los más bajos, o sea, de baja resolución a alta resolución, se observa una disminución de la energía contenida en las subbandas recorridas.

### 2.3.5.1. Descomposición de Wavelets.

Es la descomposición 2D de Wavelets, donde se debe especificar el valor del número de niveles (es igual a un tercio de la banda de Wavelets Figura 2.22.

Este método es un muy utilizado enfoque de descomposición, una estructura no separable con  $L = 3$ , es decir tres niveles ( $L$  debe ser  $> 0$ ). El primer nivel de descomposición se muestra en la Figura 2.20, y todo el sistema con los tres niveles en la Figura 2.21.

El resultado de ésta descomposición es un canal a escala llamado  $S$ , el cual representa una subimagen muestreada, y nueve canales Wavelets (1,...,9), que representan los niveles de frecuencia de la imagen. Diferentes métodos de análisis estadístico se aplican a  $S$  y a 1 – 9, la implementación será descrita en el capítulo siguiente.

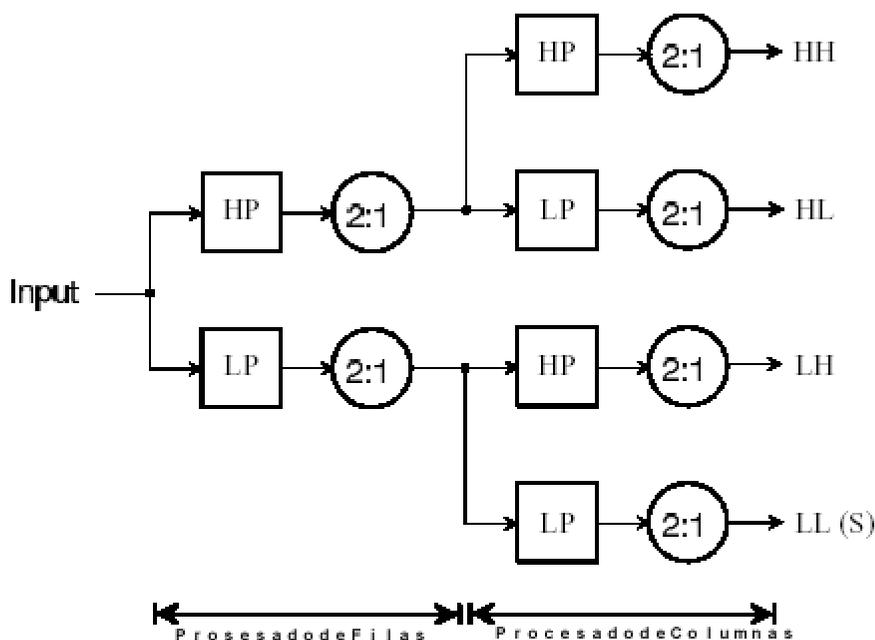


Figura 2.20: Descomposición Wavelet, primer nivel.

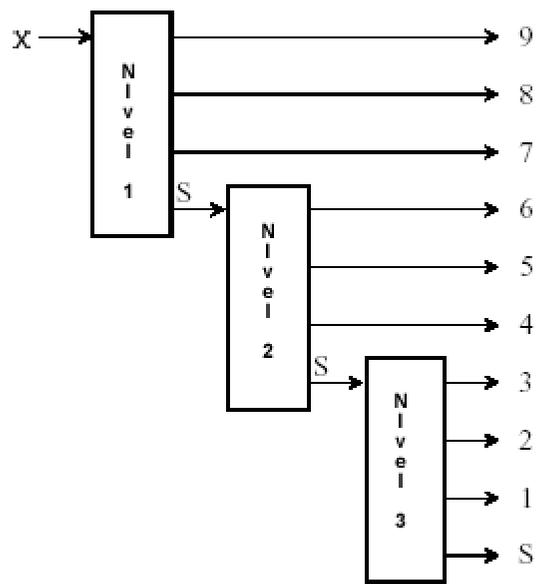


Figura 2.21: Descomposición Wavelet, Tres niveles.

### 2.3.5.2. Combinación de Canales.

Desde la naturaleza de la Transformada Wavelets, la energía en ciertas bandas se mantiene constante si se ejecuta el escalado o rotación de una imagen, esto se denomina invariante al escalamiento e invariante a la rotación respectivamente. Por medio del uso de un descriptor de combinación de canal, sumas lineales de los canales de energía se puede definir una descripción para combinaciones de canales Wavelets. Para cada combinación de canales, el número de combinación de  $K$  canales, y sus índices deben ser especificados, lo cual es un total de  $k + 1$  valores de descriptores. Se propone la siguiente combinación de canales Wavelets  $1+2+3$ ,  $4+5+6$ ,  $7+8+9$ , los cuales son de criterio rotación-invariante. Y los canales Wavelets  $1+4+7$ ,  $2+5+8$ ,  $3+6+9$ , los cuales son de criterio escala-invariante, ver Figura 2.22. En lugar de 9 canales se puede representar la imagen en una combinación de 6 canales o bandas, lo cual es más compacto y resume la información más importante.

### 2.3.5.3. Cuantización.

Los coeficientes de los canales Wavelets o combinaciones de canales serán ahora cuantizadas. Esto se puede lograr con uno de los cuantizadores, es un operador binario denominado *Thresholding* o *valor umbral*. El valor óptimo del umbral se debe determinar tal que aproximadamente la mitad de las muestras queden bajo este valor, y la otra mitad sobre el umbral. Porque esto aumenta al máximo la información contenida sobre la imagen dentro de la representación binaria. Esto puede lograrse por medio del análisis del histograma. Sin embargo una función de matching puede comparar sólo aquellos ítems donde la característica es extraída con el mismo o valores muy aproximados el umbral.

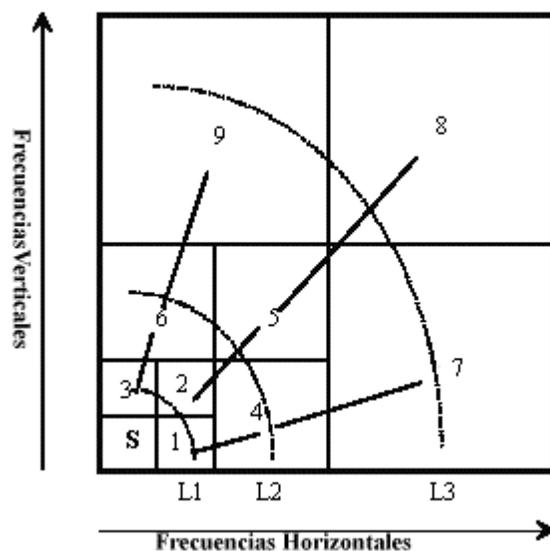


Figura 2.22: Bandas Wavelet.

Entonces, desde los coeficientes Wavelets o coeficientes de combinación, después se construyen los patrones de frecuencia a partir del umbral en cada posición de muestreo de la imagen. Para éste propósito es necesario explotar los canales de Wavelets de más baja frecuencia. Tal que todos los coeficientes son combinados en imágenes de igual tamaño (ver lado izquierdo de la Figura 2.23). Para cada posición de éstas imágenes, los valores binarios se componen

para formar un patrón de frecuencia (Figura 2.23 lado derecho). El número total de patrones de frecuencia diferentes es  $2^M$ , siendo M el número de canales Wavelets o combinación de canales.

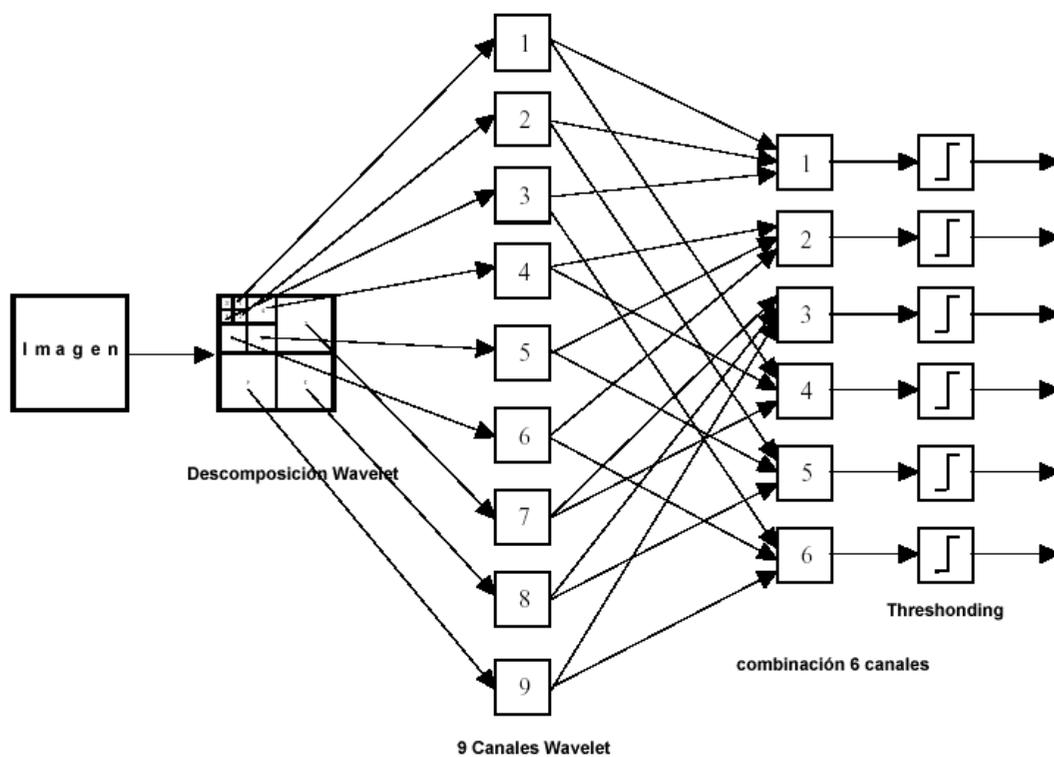


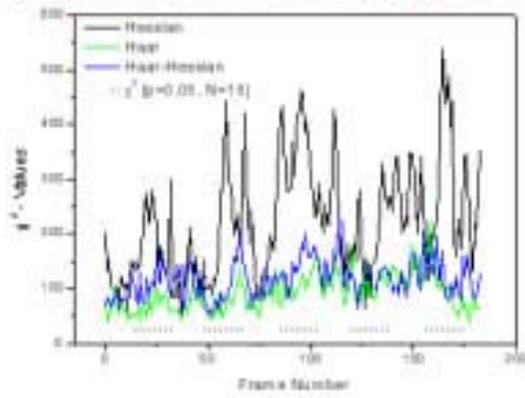
Figura 2.23: Extracción de los Patrones de Frecuencia de la señal de la imagen.

## 2.4. Comparando Wavelets versus Fourier.

FFT y DWT se basan ambas en operaciones lineales, como resultado, ambas transformadas pueden verse como una rotación en el espacio de la función a un dominio diferente. Para FFT, éste nuevo dominio contiene funciones en base a senos y cosenos. Para DWT, éste nuevo dominio contiene las funciones en base a las llamadas Wavelets.

Una cosa para importante es que DWT no tenga un solo juego de funciones base como FFT que utiliza simplemente funciones seno y coseno, DWT tiene un conjunto infinito de posibles funciones de la base. Así el análisis DWT proporciona acceso inmediato a información que puede disimularse por otros métodos de tiempo-frecuencia como el análisis de Fourier. Como conclusión DWT es más rápida y más exacta de FFT. A continuación se muestran imágenes analizadas por diversas transformaciones (Figura 2.24) entre ellas Hessian (basada en las derivadas parciales) y DWT, se nota que a simple vista que para la selección de IP con ambos métodos DWT es mucho mejor ya que entrega como resultado puntos a través de toda la imagen, en cambio FFT se aglomera en estructuras sin importancia real. En los gráficos, queda demostrado que FFT es más sensible a cambios buscos en la imagen, tiene más variaciones, es decir los detalles finos en la imagen influyen de forma negativa en los resultados, no así en DWT en donde los detalles finos tienden a desaparecer a través de los niveles, entregando mejores los resultados.

-> Hess



-> Haar

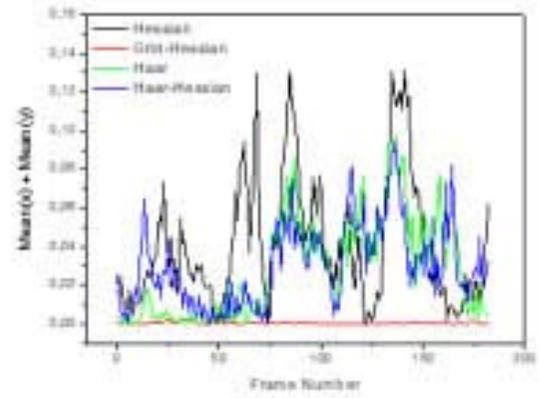


Figura 2.24: Comparación de transformaciones, a la izquierda Hessian y a la derecha Wavelets [HAR01].

## **CAPITULO III.**

### **3. METODOLOGÍA.**

#### **3.1. Introducción.**

Como se describió en el Capítulo I lo que se quiere llegar a obtener es un sistema de análisis robusto de segmentación por movimiento. En éste capítulo se describirá la implementación de los algoritmos que se utilizarán para llevar a cabo los objetivos planteados.

Las imágenes que serán analizadas fueron proporcionadas por el Centro de Estudios Científicos CECS, éstas imágenes corresponden a células de iones de sodio, blebbing, que corresponde a una transformación de la célula al nivel de la pared celular y se manifiesta con erupciones de dicha capa en forma de ampollas y fueron extraídas por medio de un microscopio electrónico. También se tiene una serie de imagen proporcionadas por la Universidad de Córdoba (Dr. Härtel), que fueron extraídas por el mismo medio.

Ambas instituciones proporcionaron videos con las muestras celulares, los que serán separados en sets de imágenes o frames con el software MediaEditor, y luego almacenados como secuencias de .tif a las que se denominarán “Set de Imágenes”, con lo que se genera una entrada valida para ser analizada por los métodos de PDI que serán implementados.

Se analizarán también diversos videos proporcionados por el Dr. Härtel, con la finalidad de probar que el sistema funciona tanto para imágenes específicas (sistemas biológicos) como para imágenes en general.

La implementación está basada en los siguientes supuestos:

- Las imágenes están compuestas de Objetos y Fondo.
- Los Objetos cumplen con la hipótesis de movimiento.

- El Fondo carece de movimiento.

El Software base fue proporcionado por el Dr. Steffen Härtel, es una aplicación desarrollada en IDL para procesamiento de imágenes celulares, también proporcionó el Software de desarrollo IDL y los manuales relacionados [RES03]. Para satisfacer los requerimientos se construye un sistema en el lenguaje IDL que posee las características siguientes [BAR2002]:

- Es un Lenguaje Procedural, es decir permite generar módulos que se ajustan a modelos que están gobernados por una secuencia de procesos que dependen de una entrada.
- Posee una Interfase GUI que permite generar módulos amigables.
- Posee una amplia Biblioteca de Funciones Gráficas, especiales para el manejo de imágenes.
- Es muy robusto en cuanto al manejo matricial se refiere.

El sistema es capaz de:

- Permite el análisis de cualquier tipo de imágenes, desde células hasta imágenes de diversa índole.
- Permite ingresar una imagen o un conjunto de imágenes. Estas pueden ser almacenadas como un Stack (archivo .sav) (Ver sección 3.2).
- Permite cargar un .sav o conjunto de imágenes almacenado.
- Permite seleccionar una parte IO o la imagen completa.
- Permite ingresar los parámetros de entrada necesarios para cada transformación definida en el sistema.
- Permite aplicar más de una transformación a la imagen seleccionada.
- Permite aumentar el tamaño de la imagen resultante con una opción de zoom (8x).

- Permite seleccionar transformaciones para una imagen y luego reflejarlas o copiarlas al conjunto de imágenes, es decir al Stack completo.
- Permite realizar el análisis de los coeficientes Wavelets (Ver sección 3.4).
- Permite realizar la extracción de los IP a partir de los coeficientes Wavelets (Ver sección 3.4.5).
- Permite obtener los VD con la transformación OF (Ver sección 3.5).
- Permite llevar a cabo Segmentación de objetos por movimiento, entregando imágenes con los objetos segmentados a través del tiempo (Ver sección 3.6).
- Permite extraer características de interés de los objetos obtenidos (Ver sección 3.7).
- Permite extraer la información en forma gráfica (Ver sección 3.8).
- Permite guardar los resultados de las transformaciones como imágenes o conjuntos de imágenes de tipo .tif y Permite exportar datos en formato texto (Ver sección 3.9).

## 3.2. Adquisición de la Imagen.

El objetivo es obtener la imagen digital de la representación de la imagen, es decir, obtener la matriz de Intensidad  $I(x,y)$  (Ver sección 2.1.2.).

La imagen que recibirá el sistema debe estar en formato .tif (TIFF, sin compresión LZW - es un método de compresión que no ocasiona pérdidas, ya que no elimina detalles de la imagen - debido que la licencia de las librerías para IDL se deben adquirir en forma adicional) y además debe estar en formato monocromático (HSV de las siglas en inglés Hue, Saturation, Value) éste fue el estándar elegido según los requerimientos, debido a que la implementación monocromática implica menos costo y un procesamiento más rápido.

Para transformar una imagen a su forma digital se requiere obtener la matriz  $I(x,y)$  (Ver Figura 2.1). A su vez se debe definir el tamaño de la imagen y los niveles de intensidad permitidos, el tamaño de la imagen  $M$  y  $N$  se definen según la IO, y el rango de intensidades  $L$  será  $[0,255]$  (Ver sección 2.1.2).

La matriz de Intensidades se obtiene de la directa manipulación del píxel en la IO, es decir, el valor de salida es absolutamente dependiente del valor de la entrada en la misma posición. La cantidad de píxeles utilizada será de  $256 \times 256$ , debido a que con imágenes más grandes la complejidad de cálculo y el tiempo requerido es mayor.

Las imágenes serán ingresadas al sistema por la Interfase de Stack, Figura 3.6-a), y el menú, Figura 3.6-b), con la opción *Load Images*. Se puede ingresar tanto una imagen como un conjunto de ellas.

Estas imágenes pueden ser guardadas en conjunto en el formato definido como .sav, éste archivo .sav contiene el Stack, pila o conjunto de imágenes ingresadas al sistema y posteriormente éste archivo puede ser utilizado como entrada, en lugar de ingresar nuevamente las imágenes .tif.

### 3.3. Reducción de Ruido.

El objetivo (como se mencionó en la sección 2.2) de hacer una transformación de la imagen (más bien transformaciones) es obtener a partir de los valores de intensidad de los píxeles, características que den una mejor información sobre la física de los objetos en la imagen, y que sean de mayor utilidad en los niveles posteriores. Es decir, procesar una imagen para que el resultado sea más adecuado que la IO para su posterior análisis.

La eliminación del ruido de las imágenes será opcional y se podrá aplicar en cualquier momento del proceso. El ruido depende de muchos factores, que no necesariamente están presentes en todas las imágenes, por lo que la persona que está analizando un set de imágenes tendrá la libertad de usar o no éstos métodos según estime conveniente.

En la sección 2.2. se describieron en forma teórica las técnicas para reducción de ruido y realce de la imagen, las transformaciones más utilizadas son:

- Median: ésta transformación suaviza la imagen, quitando el ruido tipo *sal* y *pimienta*, que son los valores aislados, ya sean altos o bajos.
- OpenClose: ésta transformación al igual que la anterior consigue eliminar el ruido *sal* y *pimienta*, limpia la imagen de los puntos aislados. Open realiza la limpieza en forma de apertura, desde los límites del objeto hacia afuera, y Close desde el objeto hacia adentro.
- Histograma: éste método es útil cuando se necesita resaltar un determinado rango de niveles de gris, eliminando los niveles no deseados, ver sección 2.2.3, en la Figura 2.5 se muestran algunos ejemplos.

### **3.4. Cálculo de los IP Utilizando la Transformada Discreta de Wavelets.**

El objetivo de realizar la transformada de Wavelets sobre la imagen o más bien sobre la Matriz de Intensidades  $I(x,y)$  es con el fin de obtener los IP de dicha imagen. El reconocimientos de IP sirve para simplificar el análisis de imágenes, realizando una minimización de la cantidad de puntos a ser procesados, mientras que al mismo tiempo se conserva la información estructural de la imagen.

Con los IP se realiza el cálculo sobre la información de interés de la imagen con lo que se disminuye el tiempo de cálculo en las etapas posteriores.

#### **3.4.1. Introducción.**

La Transformada de Wavelets es ampliamente utilizada en compresión de imagen y señales [MPE02], además es considerada un algoritmo robusto dentro de las técnicas de estimación de movimiento de objetos en imágenes.

Un ejemplo del resultado de aplicar éste método es el que se muestra en la Figura 3.1. En Figura 3.1-a) se tiene la IO de Lena. 3.1-b) se tiene el resultado de los IP utilizando el algoritmo de Wavelets o Haar Wavelet. Con lo que se demuestra que se reduce la cantidad de píxeles a sólo un conjunto de IP, lo que facilitará el análisis posterior de la imagen en estudio. El único inconveniente que presenta éste algoritmo es que en imágenes donde los puntos que representan un objeto son de poco peso o son puntos del borde y éste no está bien definido, dichos puntos desaparecen al realizar el cálculo de los coeficientes Wavelets, es decir, los detalles más finos desaparecen. Sin embargo ésta es una perdida mínima en comparación a resultados obtenidos.

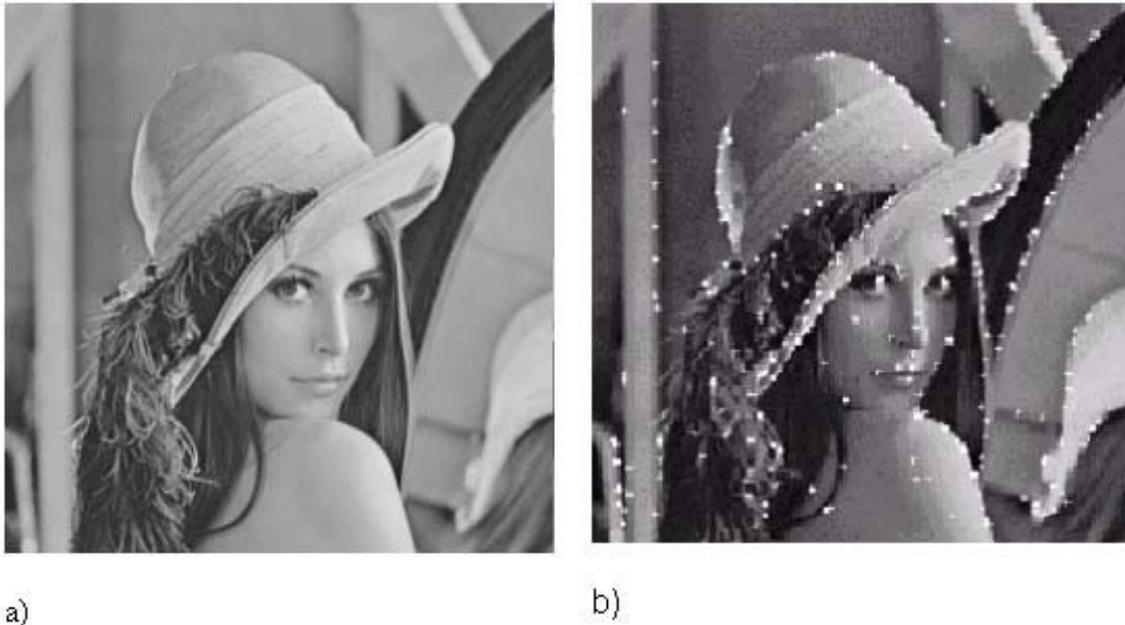


Figura 3.1: a) IO. b) Resultados Haar Wavelet.

Por ejemplo cuando se quiere aplicar sobre la IO una transformación como OF, el algoritmo se demora bastante en procesar la respuesta, y esto se debe a que está aplicando las operaciones matemáticas sobre la matriz completa, es decir sobre  $256 \times 256 = 65536$  píxeles, en cambio, si se hace uso de la extracción de IP, existe la posibilidad de elegir cuántos puntos se requiere obtener, por ejemplo la mitad de los píxeles de la imagen, o sea, 32768, con esto el tiempo de cálculo del OF se reduce a la mitad que en el caso anterior.

A continuación se muestran los resultados de cuatro métodos utilizados en PDI, aplicados a la imagen de Lena, de los cuatro métodos los dos basados en Wavelets visualmente son mejores, toman en cuenta la escena completa, los IP representan a todos los objetos contenidos en la imagen, a diferencia de los otros dos que se concentran en estructuras sin mucha importancia como por ejemplo el cabello.



a)



b)



c)

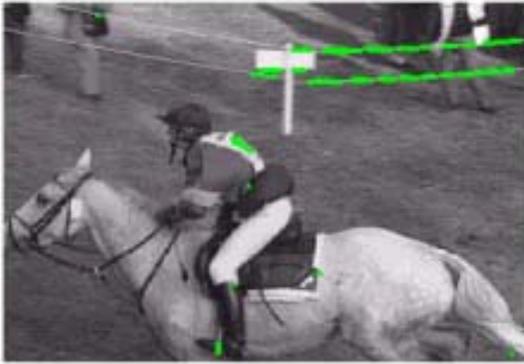


d)

Figura 3.2: Método de Harris, b) Método de Haar Wavelets, c) Método de Debauchie Wavelets, d) Método Multi Resolución Basado en Contraste [HAR01], [OHM99].

Otro ejemplo, al igual que la Figura 2.23 es la figura 3.2-1 (Ver sección 2.4).

-> Hess



-> Haar

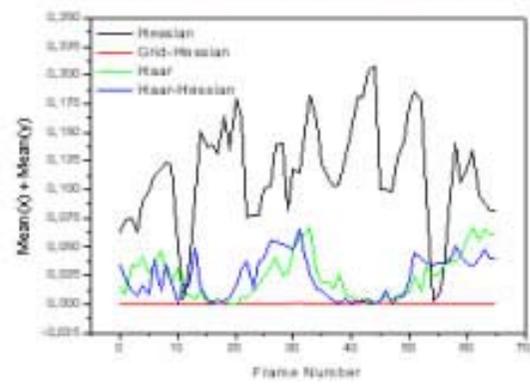
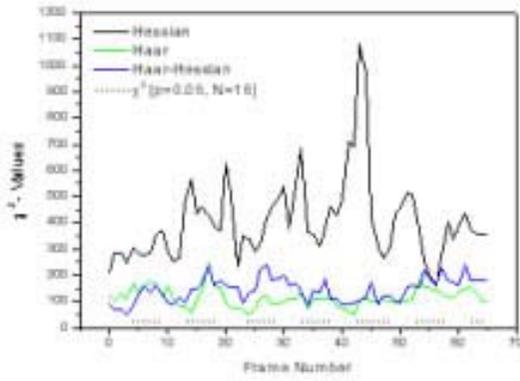


Figura 3.3: Comparación de transformaciones, la imagen izquierda corresponde a Hessian y a la derecha Wavelets [HAR01].

### 3.4.2. Implementación y Código Fuente de la Transformada Discreta de Wavelets.

A continuación se describirá la implementación de la transformada de Wavelets o Haar Wavelets definida en la sección 2.3.5. Entonces se requiere calcular los coeficientes Wavelets (Ver Fig. 2.19), para ello se creó el procedimiento llamado `C_slmageFilter_HaarWavelet`, éste procedimiento recibe como entrada la matriz de intensidades, y devuelve una matriz que contiene los coeficientes Wavelets como se muestra en la Fig. 3.4.

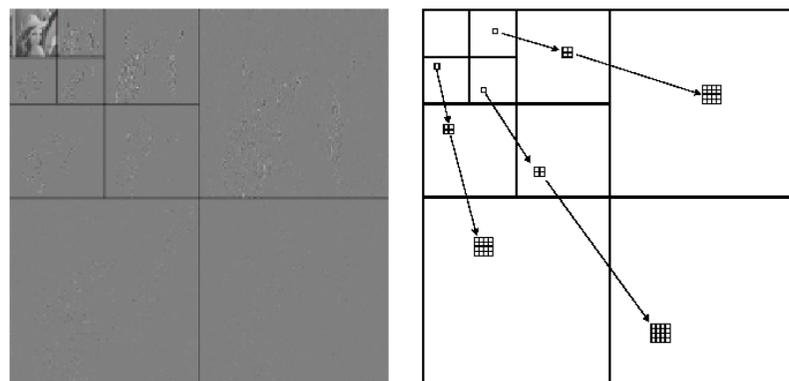


Figura 3.4: Coeficientes Wavelet buscados [OHM99].

La Figura 3.5, muestra la convención utilizada para calcular los coeficientes Wavelets.

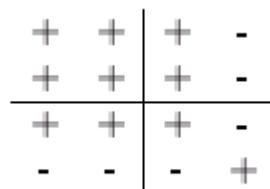


Figura 3.5: Convención utilizada para calcular los coeficientes Wavelet [OHM99].

El siguiente es el código fuente de la implementación de la transformación de Wavelets.

```

function C_ImageFilter_HaarWavelet::apply, image = image
for i = 0,level-1 do begin ;l = nivel de profundidad
  xSizeRed = floor( xSize * 0.5) ;la dim x se divide en dos
  ySizeRed = floor( ySize * 0.5) ;la dim y se divide en dos
  if ( floor(xSize mod 2.) ) then xBroken = 1 else xBroken = 0 ;control de bordes
  if ( floor(ySize mod 2.) ) then yBroken = 1 else yBroken = 0 ;control de bordes

  for x=0, xSizeRed-1 do begin
    for y=0, ySizeRed-1 do begin
      ij1 = 1. * image[ 2 * x , 2 * y ]
      ij2 = 1. * image[ 2 * x + 1 , 2 * y ]
      ij3 = 1. * image[ 2 * x , 2 * y + 1 ]
      ij4 = 1. * image[ 2 * x + 1 , 2 * y + 1 ]
      un[ x , y ] = (ij1 + ij2 + ij3 + ij4) * 0.25 ;la vecindad es de 4 =0.25 ;calcula coef S
      un[ x + xSizeRed + xBroken , y ] = ij1 - ij2 + ij3 - ij4 ;calcula coef H
      un[ x , (y + ySizeRed + yBroken) ] = ij1 + ij2 - ij3 - ij4 ;calcula coef V
      un[ x + xSizeRed + xBroken , (y + ySizeRed + yBroken) ] = ij1 - ij2 - ij3 + ij4 ;calcula coef D
    endfor
  endfor

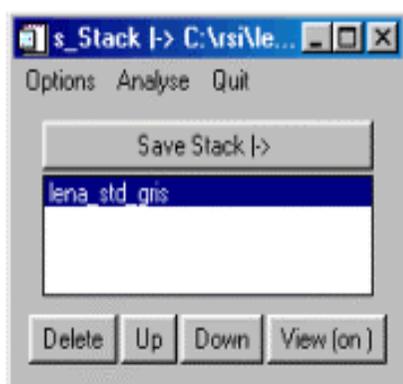
  if (yBroken) then begin ;control de bordes
    for x=0, xSizeRed-1 do begin
      un[ x , ySizeRed] = 0.25 * (un[x , y] + un[(x+1) , y] + un[x , (y+1)] + un[(x+1) , (y+1)])
    endfor
  endif
  if (xBroken) then begin ;control de bordes
    for y=0, ySizeRed-1 do begin
      un[xSizeRed , y] = 0.25 * (un[x , y] + un[(x+1) , y] + un[x , (y+1)] + un[(x+1) , (y+1)])
    endfor
  endif
  image = un[0:xSizeRed-1, 0:ySizeRed-1] ;se guarda en la imagen original los coeficientes calculados
  xSize = xSizeRed
  ySize = ySizeRed
endfor
return un
end

```

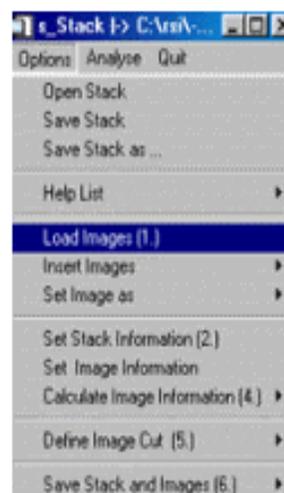
### 3.4.3. Aplicación para la Obtención de los Coeficientes Wavelets

#### Utilizando Transformada Discreta de Wavelets.

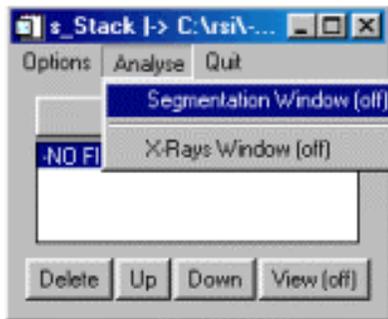
Las Interfases que se muestran a continuación corresponden al Software implementado. La Figura 3.6-a) muestra la interfase que permite ver la lista de imágenes ingresadas al sistema, al elegir el botón “View (on)” se abre otra ventana que contiene la imagen seleccionada, la Figura 3.6-b) muestra el menú que permite el manejo de los archivos en el sistema, aquí se encuentra la opción “Load Images” que permite ingresar las imágenes al sistema, la Figura 3.6-c) muestra un menú de análisis que permite abrir la interfase de segmentación mostrada en la Figura 3.6-d), ésta interfase permite elegir la transformación a utilizar, dentro de ellas está la de Wavelets, luego se debe hacer clic sobre “Pass Segmentation Method” y la transformación quedará seleccionada en la parte inferior, se puede elegir más de una, esto depende del análisis que requieran las imágenes que están siendo estudiadas.



a)



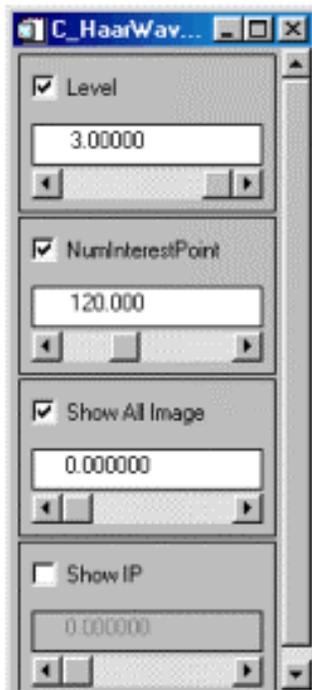
b)



c)



d)



e)

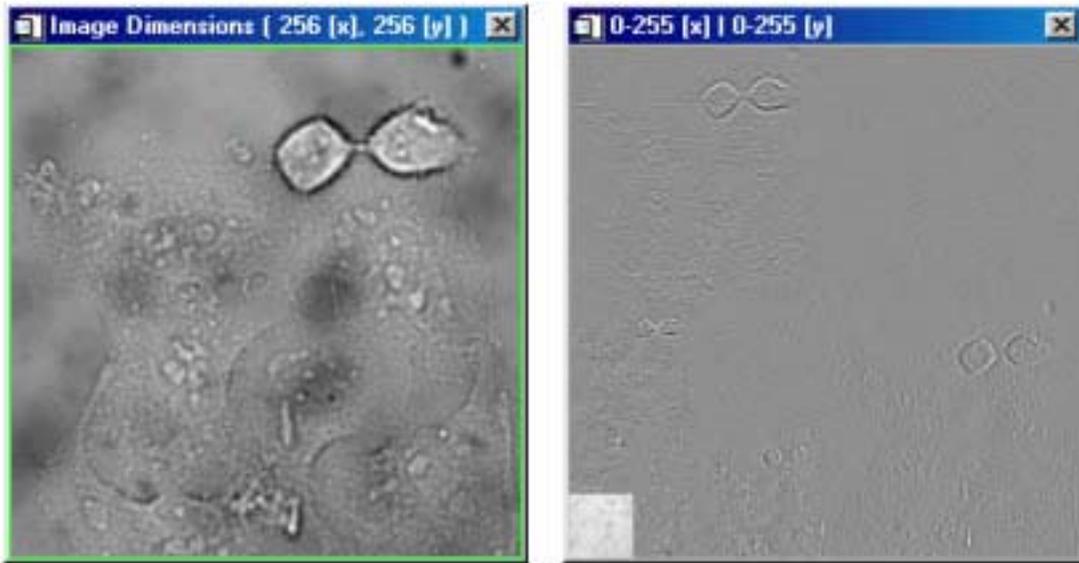
Figura 3.6: a) Interfase de ingreso de imágenes. ,b). Menú de ingreso de imágenes y manejo de archivos, c). Menú de análisis que abre la Interfase de Segmentación, d) Interfase de segmentación. e) Ingreso de parámetros para la transformada Wavelets.

### 3.4.4. Resultados Obtenidos al Aplicar la Transformada Discreta de Wavelets.

Las imágenes originales utilizadas tienen una dimensión de 256×256, se tomaron dos ejemplos para mostrar los resultados: la imagen de Lena (Figura 3.7-a)) [COM02] muy utilizada en PDI y una imagen del set proporcionado por el CECS, Hela (Figura 3.8-a)). En la interfase de ingreso de parámetros (Ver Figura 3.6-e)) debe estar seleccionado el ítem “Show All Image” con lo que se despliegan los coeficientes Wavelets, los siguientes son los resultados:



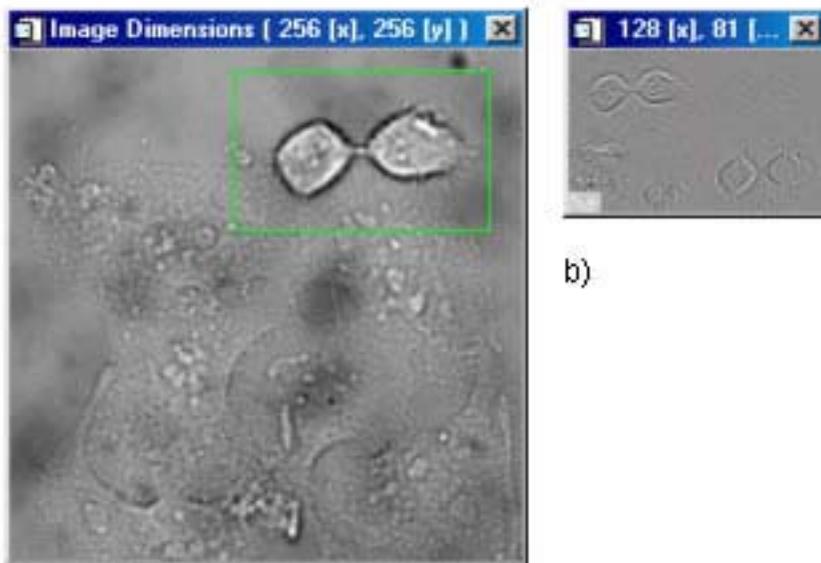
Figura 3.7: a) Imagen de Lena original, b). Resultado al aplicar Wavelets.



a)

b)

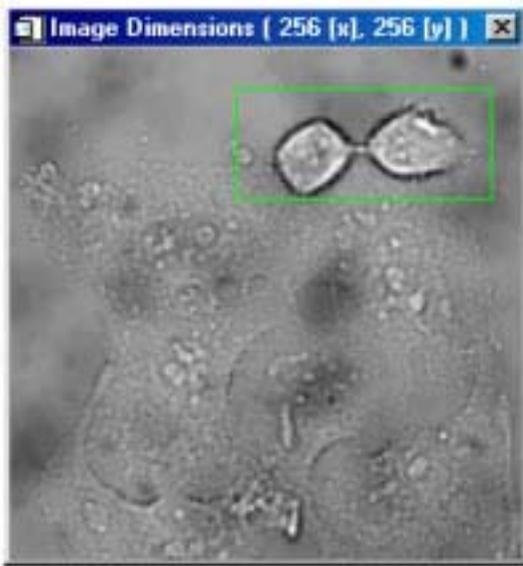
Figura 3.8: a) Imagen Hela original, b). Resultado al aplicar Wavelets.



a)

b)

Figura 3.9: a) Imagen Hela selección, b). Resultado al aplicar Wavelets.



a)



b)

Figura 3.10: a) Imagen Hela selección, b). Resultado al aplicar Wavelets, con zoon (3×).

### **3.4.5. Implementación y Código Fuente para la Extracción de los Puntos de Interés a partir de los Coeficientes Wavelets.**

La extracción de los IP se llevará a cabo siguiendo los pasos que se describen a continuación, una vez que se tienen los coeficientes Wavelets se debe:

Paso 1) Calcular las sumas de los grupos de canales para los criterios de Rotación Invariante (alteración de la dirección) y Escala invariante (alteración de los tamaños) (Ver Figura 2.21) (Ver sección 2.3.5.2):

Rotación Invariante	Escala Invariante
SUM1=SUM1(1,2,3)	SUM4=SUM4(1,4,7)
SUM2=SUM2(4,5,6)	SUM5=SUM5(2,5,8)
SUM3=SUM3(7,8,9)	SUM6=SUM6(3,6,9)

En las sumas de Rotación Invariante no hay problema al obtener los IP, Los tamaños de las matrices a sumar son iguales. Entonces se calcula S1, S2 y S3 y en cada suma (en cada nivel) se eligen NumIntPoint/6 IP, si el número de IP NumIntPoint es 120, en cada suma se elegirán 20 IP.

En las sumas de Escala Invariante hay que escalar los puntos de nivel (Ver Figura 3.11), por lo que en el nivel L1 se elegirán NumIntPoint/6 IP en las direcciones vertical, horizontal y diagonal, para ello se eligen los 20 puntos máximos, luego en el nivel L2 se escalan los puntos elegidos en cada dirección de L1 a L2, aquí se utilizará la siguiente convención: de los cuatro puntos en L2 se elige el punto más significativo que se encuentre más a la izquierda y más arriba. Nuevamente en cada dirección se tendrán 20 IP, en el nivel L3 se

escalamos nuevamente los IP de la misma forma descrita, con lo que se tendrán 20 IP en cada dirección.

Por lo tanto tenemos que en cada suma hay 20 IP  $\Rightarrow$  IP = 120.

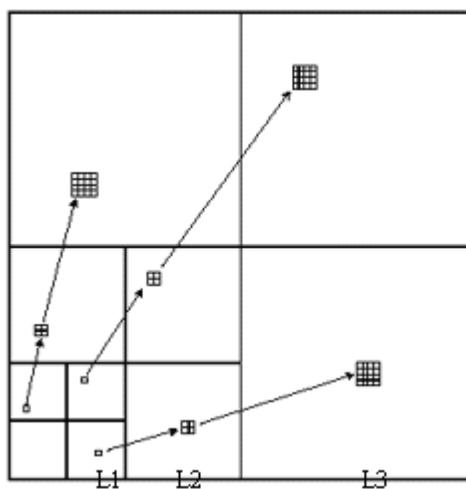


Figura 3.11: Sumas sobre las bandas Wavelets Escala invariante [OHM99].

Paso 2) Finalmente se deben escalar los IP elegidos hacia su posición en la imagen original.

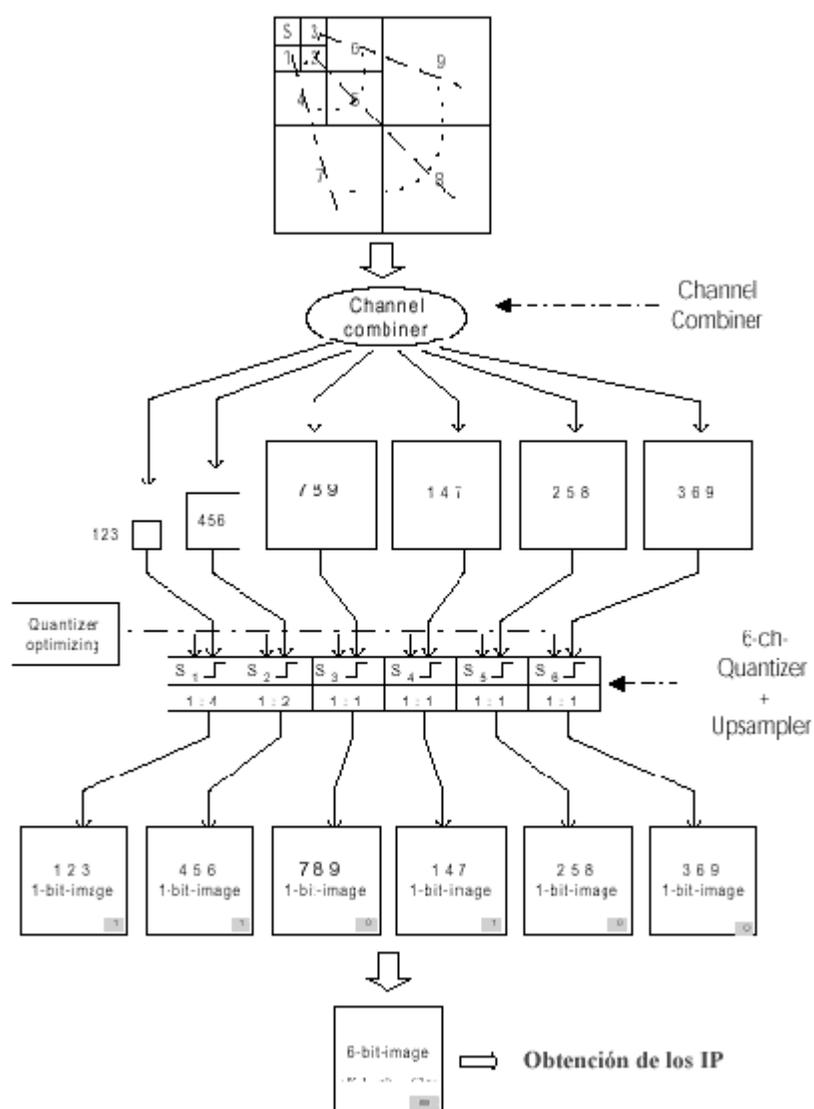


Figura 3.12: Análisis de los IP por medio de la suma de los Coeficientes Wavelet [OHM99].

El siguiente es el código fuente de la implementación de la extracción de los IP a partir de los coeficientes Wavelets. s\_GetInterestPointsSumHaar.pro:

```

for i = 0,level-1 do begin ;Search size of image S
    xSizeRed = floor(xSizeRed * 0.5)
    ySizeRed = floor(ySizeRed * 0.5)
endfor
for i = 0,level-1 do begin ;search Interest Point
    xSize = (xSizeRed * 2)
    ySize = (unsigned int)(ySizeRed * 2)
    if ( floor(xSize mod 2.) ) xBroken = 1; else xBroken = 0
    if ( floor(ySize mod 2.) ) yBroken = 1; else yBroken = 0
    NPoint = xSizeRed * ySizeRed
    sum = fltArr( xSizeRed , ySizeRed )
    for x=0, xSizeRed-1 do begin ;calcula s1,s2,s3
        for y=0, ySizeRed-1 do begin
            sum[xSizeRed:x + xSizeRed, y] = 0.
            sum[x + xSizeRed , y] = (fabs (un[(x+xSizeRed+xBroken) , y]) + $
                (un[ x , (y+ySizeRed+yBroken)]) + (un[(x+xSizeRed+xBroken) , (y+ySizeRed+yBroken)]))
        endfor
    endfor
endfor
NumInterestPointSum = floor(NumInterestPoint / 6 )
if (k eq 0) then ;begin get interest point level 1

    for j=0, NumInterestPointSum-1 do begin
        maxSum = max(sum)
        where_ = where(sum eq maxSum)
        IntPoints1[i,j] = where_[0] ;IP[k,*]
        if ( IntPoints1[i,j] ne -1 ) then begin
            sum[IntPoints1[i,j]] = 0
        endif
    endfor

endif ;end get interest point level 1
if (k eq 1) then ;begin get interest point level 2
    IntPoints = s_GetInterestPointsoflevelHaar(sum, xSizeRed, ySizeRed, IntPoints1, NumIntPoints)
    for i=0, NumIntPoints-1 do begin
        IntPoints1[i] = IntPoints[i]
    endfor ;end get interest point level 2
if (k == 2) then ;begin get interest point level 3
    IntPoints = s_GetInterestPointsoflevelHaar(sum, xSizeRed, xSizeRed, ySizeRed, IntPoints1, NumIntPoints)
    for i=0, NumIntPoints-1 do begin
        IntPoints1[i] = IntPoints[i]
    endfor ;end get interest point level 3
endif ;end get interest point level 3
xSizeRed = xSize
ySizeRed = ySize
endfor
return, IntPoints ;return Interest Point

```

### s\_GetInterestPointsoflevelHaar.pro:

```
xsize = (xSizeRed*0.5)
for i=0, NumIntPoints-1 do begin
  ;Get position in (x,y)
  posx = (IntPoints1[i] mod xsize) * 2
  posy = (IntPoints1[i] / xsize) * 2
  max = -100
  pos = 0
  for x=0, xSizeRed-1 do begin
    for y=0, ySizeRed-1 do begin
      if (posx eq x) then
        if (posy eq y)
          max = sum[x , y]
          pos = x + xSizeRed * y
          if ((sum[(x+1) , y]) gt max) then
            max = (sum[(x+1) , y])
            pos = (x+1) + xSizeRed * y
          endif
          if ((sum[x , (y+1)]) gt max) then
            max = (sum[x , (y+1)])
            pos = x + xSizeRed * (y+1)
          endif
          if ((sum[(x+1) , (y+1)]) gt max) then
            max = (sum[(x+1) , (y+1)])
            pos = (x+1) + xSizeRed * (y+1)
          endif
          IntPoints[i] = pos
        endif
      endif
    endfor
  endfor
endfor
return, IntPoints ;return Interest Point
```

### s\_s\_GetInterestPointsofendHaar.pro:

```
xsizeRed = (xSize*0.5)
for i=0, NumIntPoints-1 do begin
  posx = (IntPoints1[i] mod xsizeRed) * 2 ; Get position in (x,y)
  posy = (IntPoints1[i] / xsizeRed) * 2
  max = -100
  pos = 0
  for x=0, xSize-1 do begin
    for y=0, ySize-1 do begin
      if (posx eq x) then
        if (posy eq y) then
          max = pBWImage[x , y]
          pos = x + xSize * y
          if ((pBWImage[(x+1) , y]) gt max) then
            max = (pBWImage[(x+1) , y])
            pos = (x+1) + xSize * y
          endif
          if ((pBWImage[x , (y+1)]) gt max) then
            max = (pBWImage[x , (y+1)])
            pos = x + xSize * (y+1)
          endif
          if ((pBWImage[(x+1) , (y+1)]) gt max) then
            max = (pBWImage[(x+1) , (y+1)])
            pos = (x+1) + (xSize+xInterleave) * (y+1)
          endif
          IntPoints[i] = pos
        endif
      endif
    endfor
  endfor
endfor
return, IntPoints ;return Interest Point
```

### 3.4.6. Resultados Obtenidos al Aplicar Puntos de Interés.

Para ver los resultados de los IP se debe seleccionar en la interfase de ingreso de parámetros (Ver Figura 3.5-e)) el ítem “Show IP”, con lo que se obtendrán como resultado visual la IO con los IP (Ver Figura 3.13-b) y 3.14-b)) y como resultado interno se obtiene sólo la matriz con los IP.



Figura 3.13: a) Imagen de Lena original, b). Resultado al aplicar IP a partir de los coeficientes Wavelet.

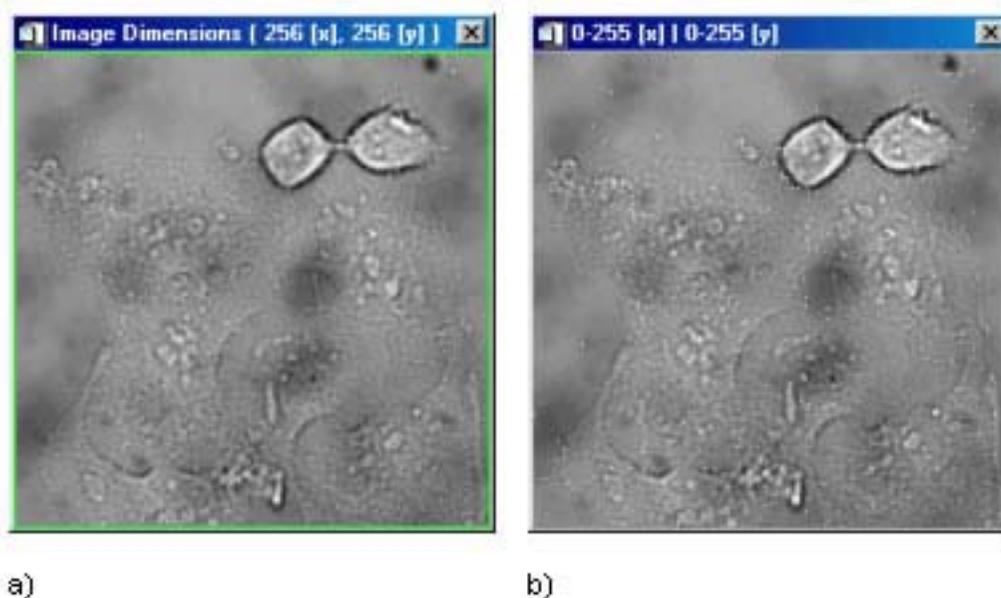


Figura 3.14: a) Imagen Hela, b). Resultado al aplicar IP a partir de los coeficientes Wavelet.

### 3.5. Cálculo de los Vectores de Desplazamiento Utilizando Optical Flow.

OF utiliza el gradiente de primer de las intensidades de la imagen para determinar los puntos de inflexión y clasificar estos puntos como de interés, en base a, la suposición que tal comportamiento en la distribución de las intensidades describe la estructura de la imagen.

Para una secuencia de imágenes se calculará la transformación llamada OF a cada píxel, es decir sobre los IP escogidos en la etapa anterior. Entonces se calcula el vector de velocidad  $V(u,v)$  (Ver sección 3.5.2), que nos dice cuán rápidamente se está moviendo el píxel a través de la imagen y en que dirección se está moviendo. El objetivo entonces es obtener la velocidad en cada píxel para determinar si el píxel se trasladó o no. OF es de gran apoyo esencialmente en imágenes donde no hay movimiento aparente de los objetos, donde es muy difícil distinguirlos o separarlos del fondo y sólo se tiene el factor tiempo para discriminar. En la Figura 3.15 se muestra un ejemplo de los VD.

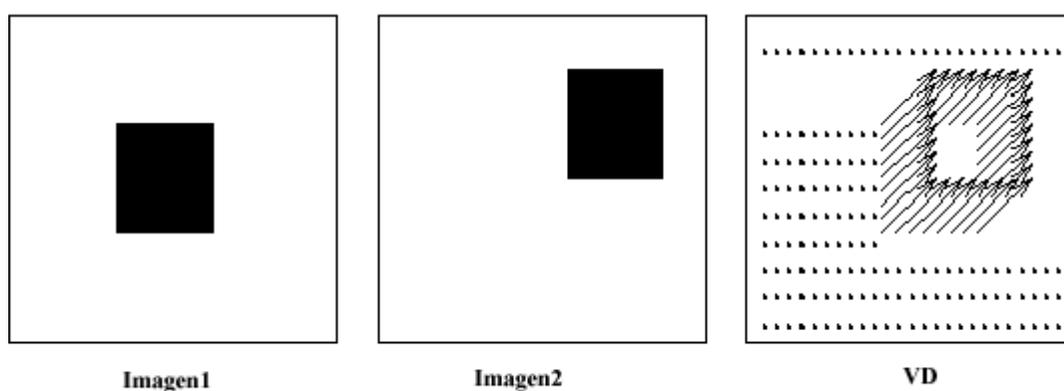


Figura 3.15: Se muestra un cuadro en su posición 1, luego el cuadro en su posición 2 y finalmente los VD [BEA99].

### 3.5.1. Introducción.

Cuando se habla de movimiento, la primera idea es asociarlo con cambios. En imágenes a niveles de gris estos cambios inicialmente se manifiestan como variaciones de las intensidades de una imagen a la imagen siguiente. Las Figuras 3.16 a) y b), muestran dos imágenes que a simple vista parecen idénticas. Si se realiza la simple sustracción de éstas dos imágenes se puede observar que uno de los objetos de la misma a sufrido un desplazamiento, como lo muestra la Figura 3.16-c). En tal sentido, se puede afirmar entonces que los cambios en los niveles de gris de las imágenes analizadas se encuentran asociados a movimiento de algunos de los objetos de la escena.

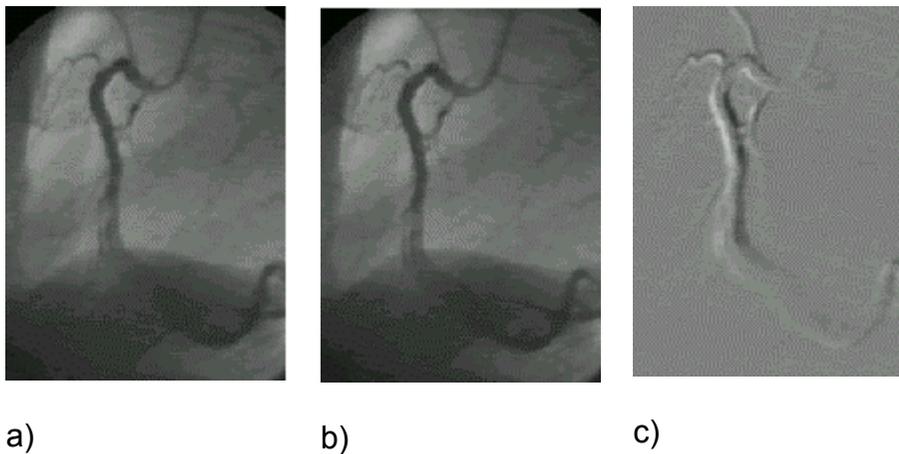
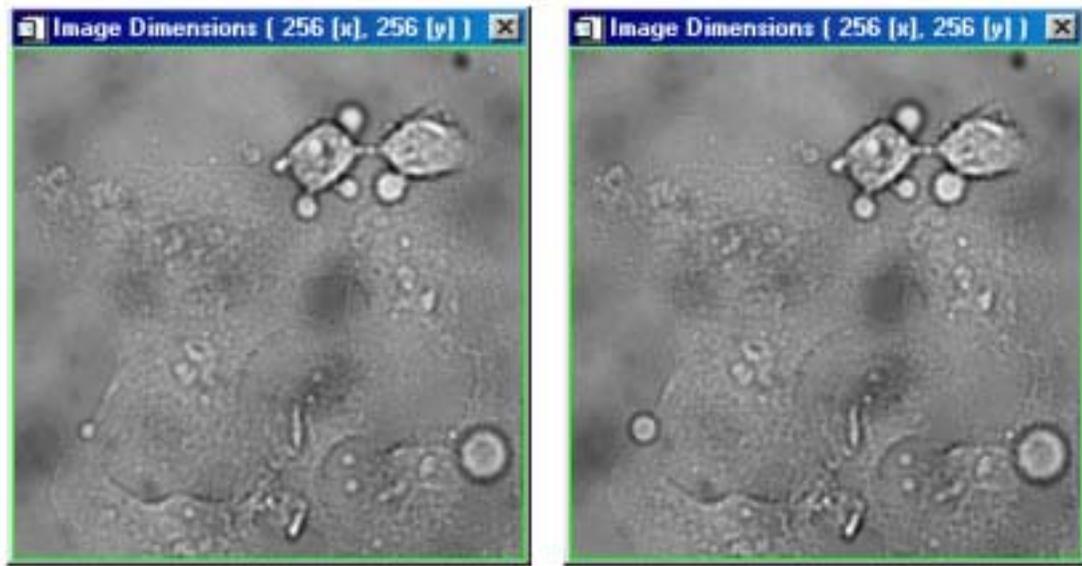
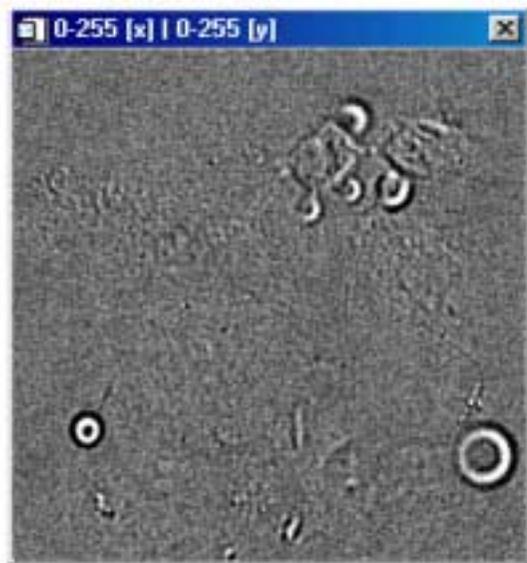


Figura 3.16: a) Imagen de la red coronaria, b). Resultado de la sustracción de las imágenes.



a)

b)



c)

Figura 3.17: a) Primera imagen secuencia 20 del set Hela, b) Segunda imagen secuencia 21 del Hela, c). Resultado de la diferencia de las imágenes a) y b).

De las Figuras 3.16-c) y 3.17.c) se puede concluir que el movimiento se manifiesta como cambios temporales en los valores de gris, pero no se puede afirmar que todo cambio temporal en los niveles de gris, se encuentra asociado a movimiento. Debido a que es posible encontrar variaciones en los niveles de gris, simplemente con cambiar el nivel de iluminación de la escena en estudio.

El problema ahora se debe centrar en cómo distinguir los cambios en los valores de gris causados directamente por el movimiento, de aquellos cambios que resultan de la variación del nivel de iluminación.

La estimación del movimiento se encuentra estrechamente relacionada con los cambios temporales y espaciales de los valores de gris, ambas cantidades pueden ser encontradas con ayuda de operadores locales, los cuales permiten calcular las derivadas espaciales y temporales.

### 3.5.2. Ecuación de Restricción de Optical Flow.

La matriz de intensidad está dada por  $I(x, y, t)$  en  $(x, y)$  (se tendrán que aproximar los cálculos a partir de las secuencias de imágenes, discretas en el espacio y el tiempo). Ésta es una función de tres variables debido a que se incorpora la variación espacio temporal al análisis de las imágenes. Para ver cómo cambia en el tiempo, se aplica la derivada con respecto a  $t$  considerando la siguiente Hipótesis: la intensidad de cada objeto permanece constante durante el movimiento.

$u(x, y)$  y  $v(x, y)$  son las componentes en  $x$  e  $y$  del OF en ese punto.

Después de un pequeño intervalo de tiempo  $\partial t$ , éste punto estará en  $(x + u\partial t, y + v\partial t)$ , conservando su intensidad. Se Tendrá, por tanto:

$$I(x, y, t) = I(x + u\partial t, y + v\partial t, t + \partial t) \quad (\text{Ec. 3.1})$$

Usando el desarrollo en serie de Taylor se tiene:

$$I(x, y, t) = I(x, y, t) + u\partial t \frac{\partial I}{\partial x} + v\partial t \frac{\partial I}{\partial y} + \partial t \frac{\partial I}{\partial t} + \varepsilon \quad (\text{Ec. 3.2})$$

Dividiendo por  $\partial t$  tendremos:

$$\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} \quad (\text{Ec. 3.3})$$

Ahora se asume que la intensidad de la imagen en cada píxel es invariable en el tiempo (por ejemplo, las sombras e iluminaciones no cambian debido al movimiento), es decir, cuando  $\partial t \rightarrow 0$ , entonces se tiene:

$$\frac{\partial I(x, y, t)}{\partial t} = 0 \quad (\text{Ec. 3.4})$$

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0 \quad (\text{Ec. 3.5})$$

Utilizando las abreviaciones,  $u = \frac{\partial x}{\partial t}$ ,  $v = \frac{\partial y}{\partial t}$ ,  $I_x = \frac{\partial I}{\partial x}$ ,  $I_y = \frac{\partial I}{\partial y}$ ,  $I_t = \frac{\partial I}{\partial t}$

$$I_x \cdot u + I_y \cdot v + I_t = 0 \quad (\text{Ec. 3.6})$$

Aquí,  $I_x$  e  $I_y$  corresponden a las derivadas parciales,  $I_t$  corresponde a la derivada temporal de la intensidad de la imagen,  $u$  y  $v$  corresponden a los VD vertical y horizontal del OF.

Luego se debe evaluar la velocidad para cada punto de la imagen, a ésta ecuación (Ec. 3.6) se le conoce como *Ecuación de Restricción de OF* [FIS02].

El objetivo es estimar el movimiento 2-D, bien sea la velocidad instantánea o el desplazamiento, a partir de la secuencia de imágenes de que disponemos.

El problema de la estimación del flujo óptico consiste en la estimación de la velocidad. Es decir,

$$V = \sqrt{u^2 + v^2} \quad (\text{Ec. 3.7})$$

Se estudia entonces el movimiento aparente (flujo óptico o correspondencia) de los puntos de la imagen, se trata entonces de evaluar la velocidad de cada uno de los puntos de la imagen, en el caso ideal, en donde la iluminación se mantiene constante a lo largo de la secuencia (un punto de la imagen conserva su intensidad al moverse), el OF corresponde al campo de velocidades.

### 3.5.3. Implementación y Código Fuente del Algoritmo Optical

#### Flow.

Si se utilizara la matriz intensidad de la imagen para calcular el OF, el tiempo de cálculo sería considerable, por lo que se aplicará el OF sobre la matriz resultante de la obtención de los IP a partir del algoritmo Wavelets.

Para ello se toman dos imágenes del set I1 y I2 (Ver Figura 3-18), aquí se muestra la convención utilizada en el algoritmo.

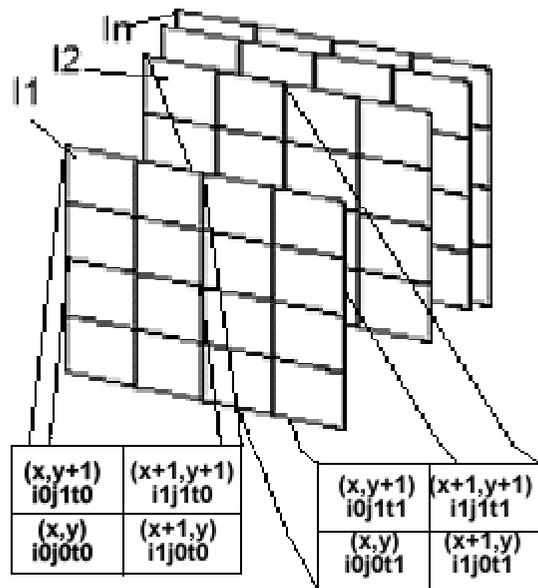


Figura 3.18: Convención a utilizar en el cálculo del OF.

El procedimiento a utilizar es iterativo, en el algoritmo las variables  $i_{0j0k0}$ ,  $i_{1j0k0}$ ,  $i_{0j1k0}$ ,  $i_{1j1k0}$ ,  $i_{0j0k1}$ ,  $i_{1j0k1}$ ,  $i_{0j1k1}$ ,  $i_{1j1k1}$ , contienen los valores de la intensidad de las imágenes (Ver Figura 3.18). Luego se calculan las derivadas parciales,  $image\_Ex$ ,  $image\_Ey$ ,  $image\_Et$ . Se utilizan dos parámetros adicionales  $alpha$  y  $Iteration$ , donde  $alpha$  es una constante de regularización, e

*Iteration*, corresponde al número de iteraciones a realizar, se permite ingresar ambos parámetros.

Se quiere obtener la aproximación de  $un$  y  $vn$ . Se calcula entonces la variable  $a$  que corresponde al error:

$$a = \frac{\partial I}{\partial x} * um + \frac{\partial I}{\partial y} * vm + \frac{\partial I}{\partial t} \quad (\text{Ec. 3.8})$$

Se calcula  $b$  y  $c$  que es el balance del error:

$$b = \alpha^2 + \left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2 \quad (\text{Ec. 3.9})$$

$$c = a/b \quad (\text{Ec. 3.10})$$

y finalmente se aproximan los valores de  $un$  y  $vn$ :

$$un = um - \frac{\partial I}{\partial x} * c \quad (\text{Ec. 3.11})$$

$$vn = vm - \frac{\partial I}{\partial y} * c \quad (\text{Ec. 3.12})$$

Con lo que se obtiene la velocidad de la ecuación 3.7, donde  $u = un$  y  $v = vn$ .

El siguiente es el código fuente de la implementación del OF a partir de los PI.

C\_sImageFilter\_OpticalFlow\_\_define.pro:

```

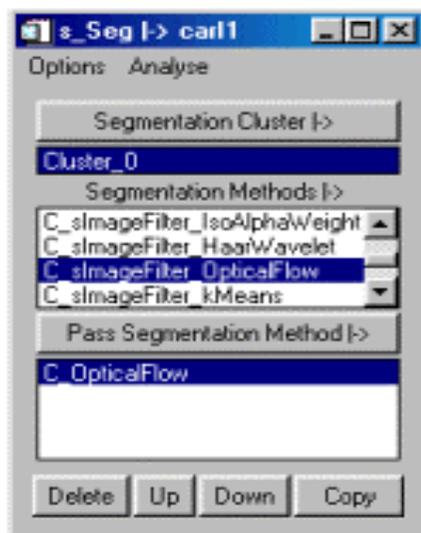
imageDim = size (image, /dim) ; get dim image
xSize = imageDim[0]
ySize = imageDim[1]
vn = fltArr( imageDim[0], imageDim[1] ) ; crea matrices a utilizar
un = fltArr( imageDim[0], imageDim[1] )
vn1 = fltArr( imageDim[0], imageDim[1] )
un1 = fltArr( imageDim[0], imageDim[1] )
image_Ex = fltArr( imageDim[0], imageDim[1] )
image_Ey = fltArr( imageDim[0], imageDim[1] )
image_Et = fltArr( imageDim[0], imageDim[1] )
;obtiene parametros de entrada
Alpha = ((*self.pParamStruct).pValues)[(where((*self.pParamStruct).pNames) eq 'Alpha')][0]
Iteration = ((*self.pParamStruct).pValues)[(where((*self.pParamStruct).pNames) eq 'Iteration')][0]

for x=0, xSize-2 do begin
  for y=0, ySize-2 do begin
    i0j0k0= 1. * image[ x , y ] ;posicion (x,y) mascara imagen1
    ilj0k0= 1. * image[ x , (y+1)] ;posicion (x,y+1) mascara imagen1
    i0j1k0= 1. * image[(x+1) , y ] ;posicion (x+1,y) mascara imagen1
    ilj1k0= 1. * image[(x+1) , (y+1)] ;posicion (x+1,y+1) mascara imagen1
    i0j0k1= 1. * image_2[ x , y ] ;posicion (x,y) mascara imagen2
    ilj0k1= 1. * image_2[ x , (y+1)] ;posicion (x,y+1) mascara imagen2
    i0j1k1= 1. * image_2[(x+1) , y ] ;posicion (x+1,y) mascara imagen2
    ilj1k1= 1. * image_2[(x+1) , (y+1)] ;posicion (x+1,y+1) mascara imagen2
    image_Ex[ x , y ] = ((i0j1k0 - i0j0k0 + ilj1k0 - ilj0k0 + i0j1k1 - i0j0k1 + ilj1k1 - ilj0k1)/4.0)
    image_Ey[ x , y ] = ((ilj0k0 - i0j0k0 + ilj1k0 - i0j1k0 + ilj0k1 - i0j0k1 + ilj1k1 - i0j1k1)/4.0)
    image_Et[ x , y ] = ((i0j0k1 - i0j0k0 + ilj0k1 - ilj0k0 + i0j1k1 - i0j1k0 + ilj1k1 - ilj1k0)/4.0)
  endfor
endfor
for i=0, Iteration-1 do begin
  alpha2 = alpha * alpha
  for x=1, xSize-2 do begin
    for y=1, ySize-2 do begin
      um = (( un[ x , (y-1)] + un[ x+1 , y ] + un[ x , (y+1)] + un[ x-1 , y ] ) / 6.0 + $
        ( un[ x-1 , (y-1)] + un[ x+1 , (y-1)] + un[ x+1 , (y+1)] + un[ x-1 , (y+1)] ) / 12.0)
      vm = (( vn[ x , (y-1)] + vn[ x+1 , y ] + vn[ x , (y+1)] + vn[ x-1 , y ] ) / 6.0 + $
        ( vn[ x-1 , (y-1)] + vn[ x+1 , (y-1)] + vn[ x+1 , (y+1)] + vn[ x-1 , (y+1)] ) / 12.0)
      a = image_Ex[ x , y ] * um + image_Ey[ x , y ] * vm + image_Et[ x , y ]
      b = (alpha2 + image_Ex[ x , y ] * image_Ex[ x , y ] + image_Ey[ x , y ] * image_Ey[ x , y ])
      u = um - (image_Ex[ x , y ] * a) / b ;dx/dt
      v = vm - (image_Ey[ x , y ] * a) / b ;dy/dt
      un1[ x , y ] = u
      vn1[ x , y ] = v
    endfor
  endfor
  un=un1
  vn=vn1
return, sqrt(un*un+vn*vn)
endfor

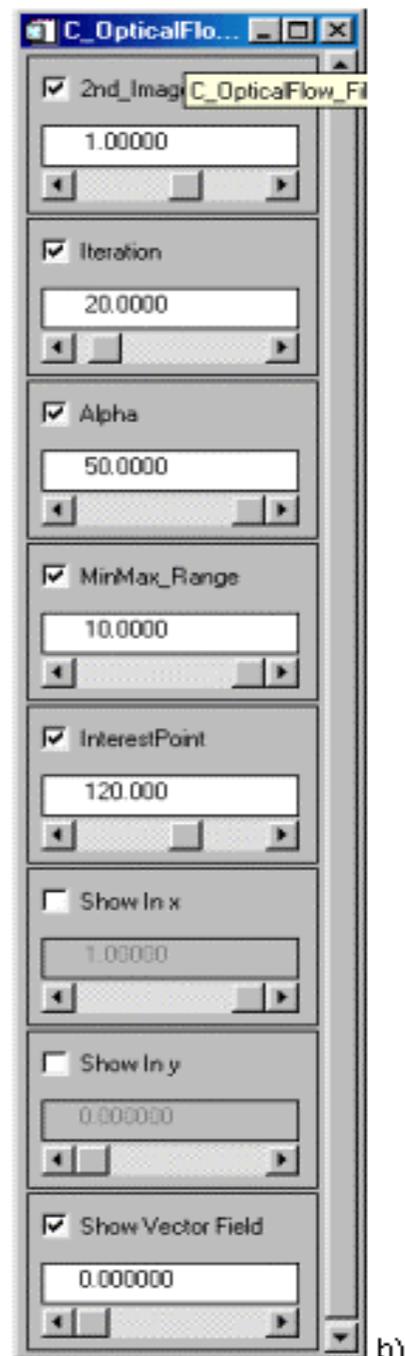
```

### 3.5.4. Aplicación para la Obtención del Algoritmo Optical Flow.

El OF es una transformación que se aplicará sobre la imagen o parte de ella, por lo que se incorporó en la interfase de segmentación, ver Figura 3.19-a).



a)



b)

Figura 3.19: a) Se agrega en la interfase de segmentación la transformada OF, b) Ingreso de parámetros para la transformada OF.

### 3.5.5. Resultados Obtenidos al Aplicar el Algoritmo Optical Flow.

Las imágenes originales utilizadas tienen una dimensión de  $256 \times 256$ , se utilizó la imagen la imagen Hela (Ver Figura 3.20). En la Figura 3.20-a) y 3.20-b) se encuentran las imágenes originales sobre las que se aplicó OF. Y las Imágenes 3.20-c) y d) corresponden a los resultados obtenidos, OF y VD respectivamente.

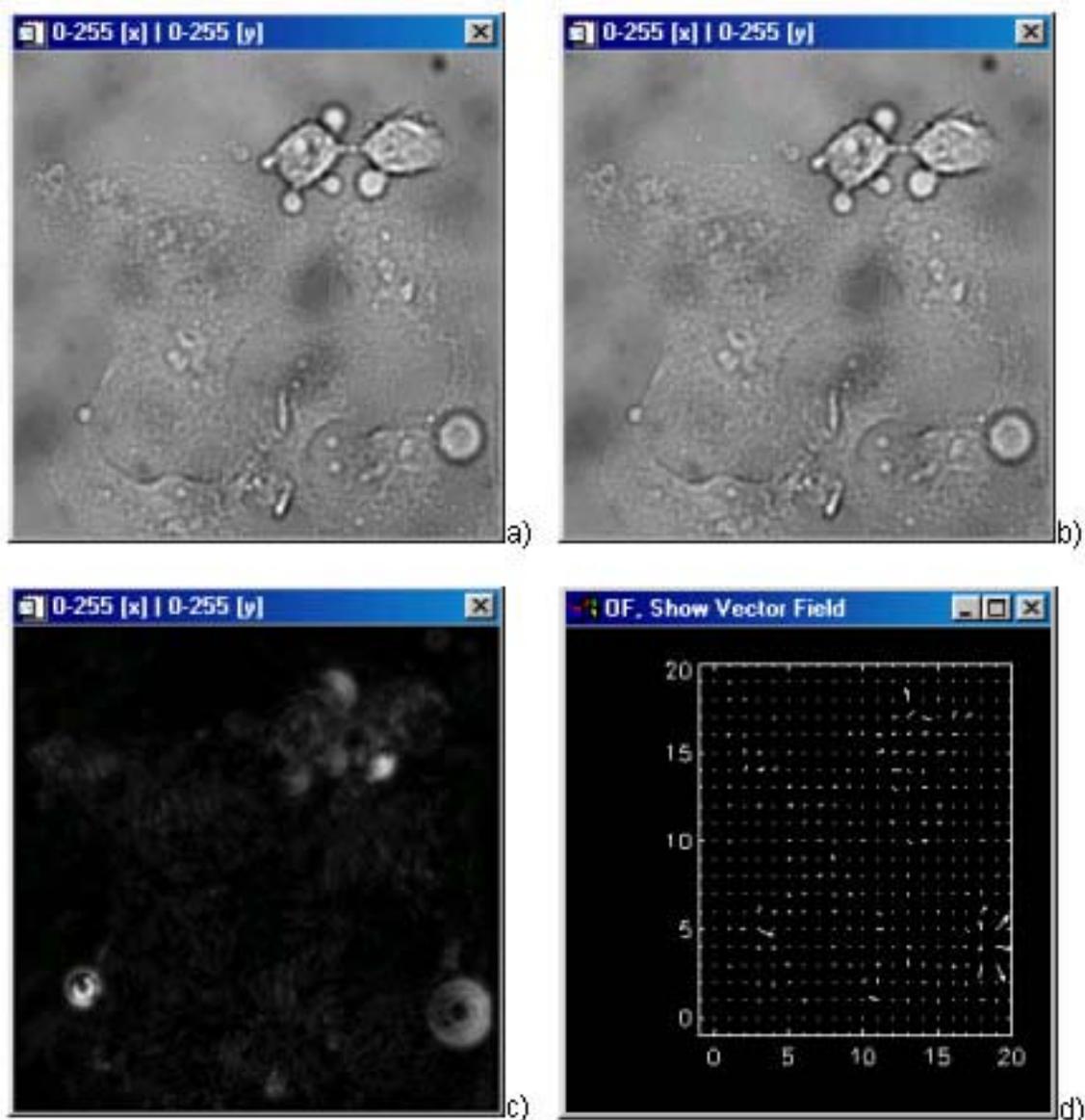


Figura 3.20: a) Imagen Hela20, b) Imagen Hela21, c) Resultado OF, d) Resultado VD.

### **3.6. Segmentación por Movimiento.**

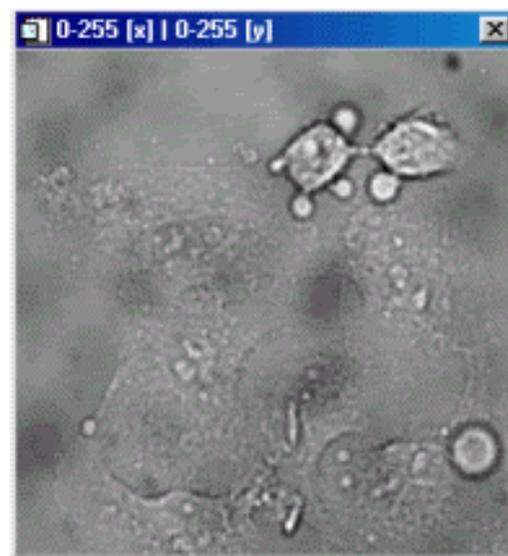
El proceso de marcar los píxeles que pertenecen a un objeto y que tienen alguna característica en común se llama segmentación y la característica en estudio en el presente trabajo es el movimiento. Como resultado de aplicar Wavelets (IP) y OF (VD) sobre las imágenes, se obtuvieron imágenes que contienen los objetos de mayor interés de las escenas, por cada escena se obtuvo una imagen .tif con los objetos segmentados (Ver Figura 3.20). El objetivo de la segmentación por movimiento es la extracción de información basada en los objetos presentes en las imágenes, en el caso de las células, es poder separar o segmentar a las células que presentan movimiento.

Los objetos serán almacenados en ROI (Region Of Interest) (una ROI almacena los objetos de mayor interés, es una estructura de almacenamiento de los parámetros que se necesita calcular) y luego pueden ser grabados como set de imágenes (.tif) para luego construir un video con las imágenes resultantes.

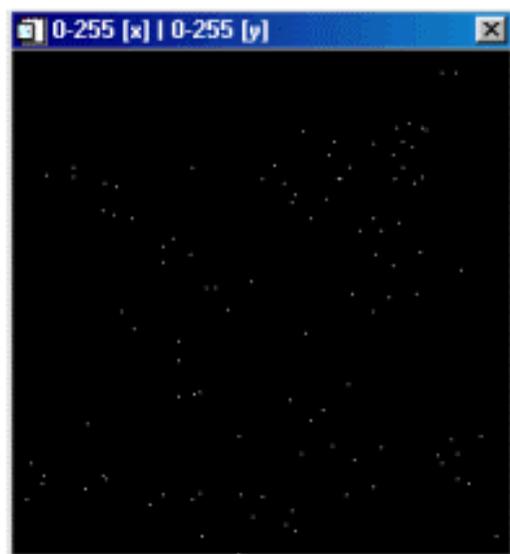
Como se mencionó antes, la entrada al algoritmo OF son los IP, por lo que primero se debe aplicar Wavelets sobre la imagen y luego calcular el OF (Ver Figura 3.21-a)).



a)



b)



c)



d)

Figura 3.21: a) Secuencia de Transformaciones aplicadas sobre la imagen, b) Imagen 20 HeLa a Segmentar, c) IP, d) Objetos de mayor interés, e) VD.

En la Figura 3.21-d) se muestran los objetos de mayor interés segmentados de la IO, se recuerda que los VD fueron calculados sobre los IP de la IO.

Esta secuencia de Transformaciones se debe aplicar a todas las imágenes del Stack, esto se logra con el Menú de la Figura 3.22.

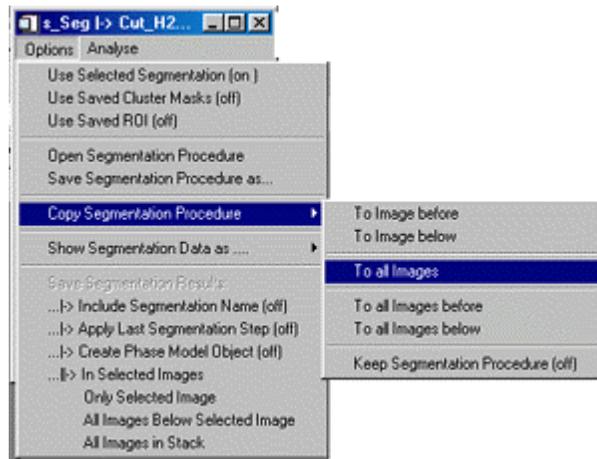


Figura 3.22: Menú interfase de segmentación, Se está mostrando Aplicar las transformaciones a todas las imágenes.

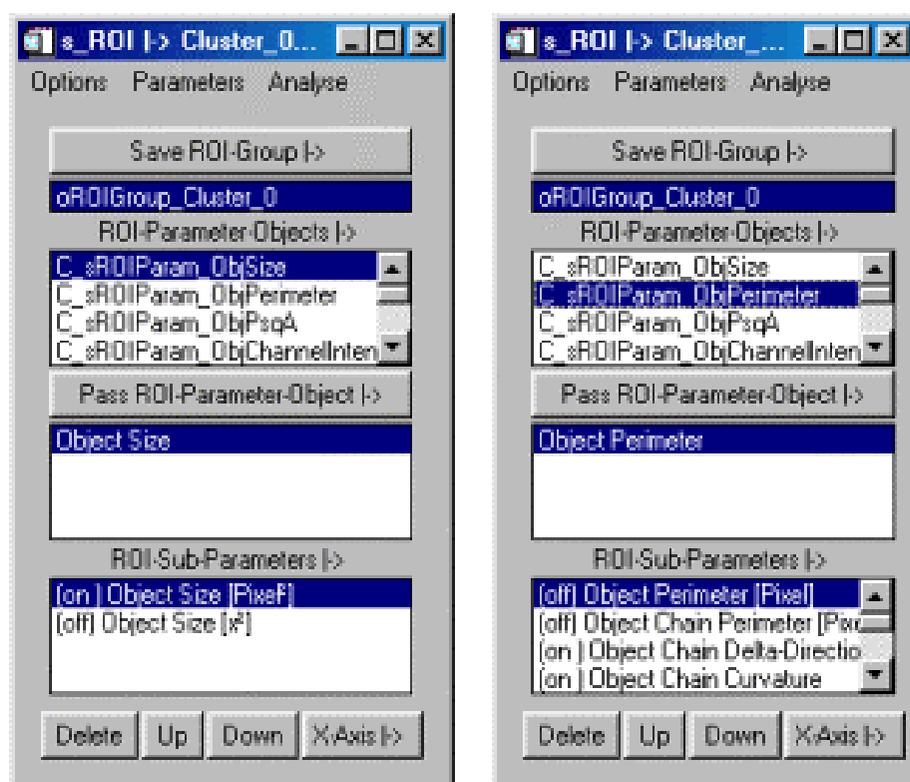
Con ésta secuencia de imágenes segmentadas por movimiento, se puede generar una película para ver el actuar de los objetos a través del tiempo. (Ver sección 3.9).

También se probó el set de imágenes sólo con la transformación OF y los resultados con bastante aceptables (Ver Anexo C.).

### 3.7. Extracción de Parámetros de Interés.

Una vez que se tiene la ROI de la secuencia de imágenes es posible calcular a partir de éstas los parámetros que se indican a continuación: en la Figura 3.23-a) muestra los parámetros de tamaños permitidos, y la Figura 3.23-b) muestra los parámetros de perímetro permitidos. El análisis se realiza a través de los histogramas de las información, (Ver Anexo D).

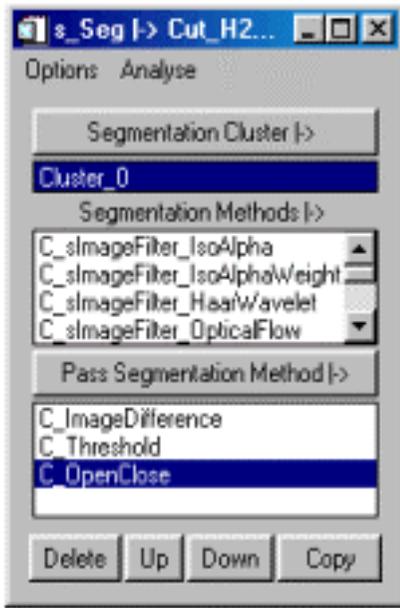
Un ejemplo es la Figura 3.24-a) corresponde a la combinación de filtros aplicados a las imágenes, La Figura 3.24-b) corresponde al resultado de aplicar los filtros sobre la imagen y La Figura 3.24-c) corresponde al Histograma de los píxeles.



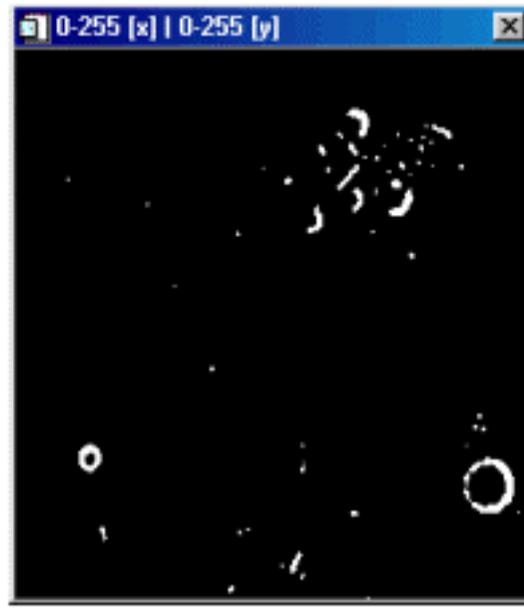
a)

b)

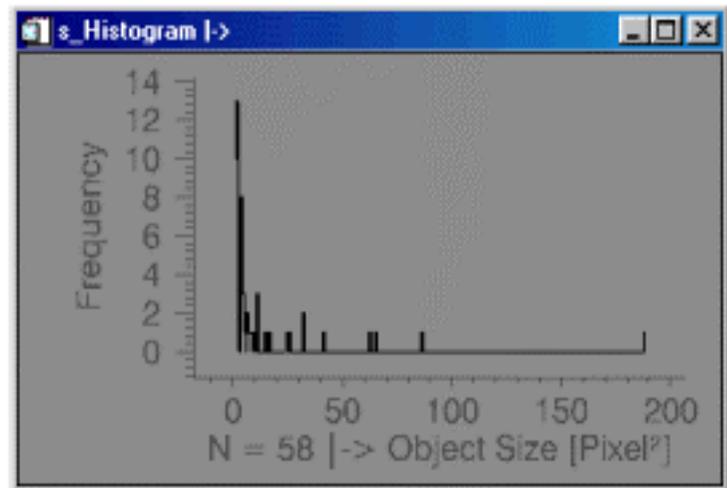
Figura 3.23: a) Interfases de ROI, muestran los parámetros que se pueden analizar, a) Parámetros de tamaños y b) Parámetros de perímetro.



a)



b)



c)

Figura 3.24: Muestra de la extracción de parámetros de interés, a) Transformaciones aplicadas sobre la IO, b) Resultado de aplicar a), c) Histograma de b).

### 3.8. Análisis Gráfico de la Información.

A partir de la información obtenida de la ROI se puede analizar gráficamente los datos. El Menú en la parte inferior de la Figura 3.25-a), permite graficar la información obtenida de los parámetros extraídos. Además en cualquier etapa de las transformaciones se puede hacer uso del Histograma de la imagen (Ver Figura 3.25-c)).

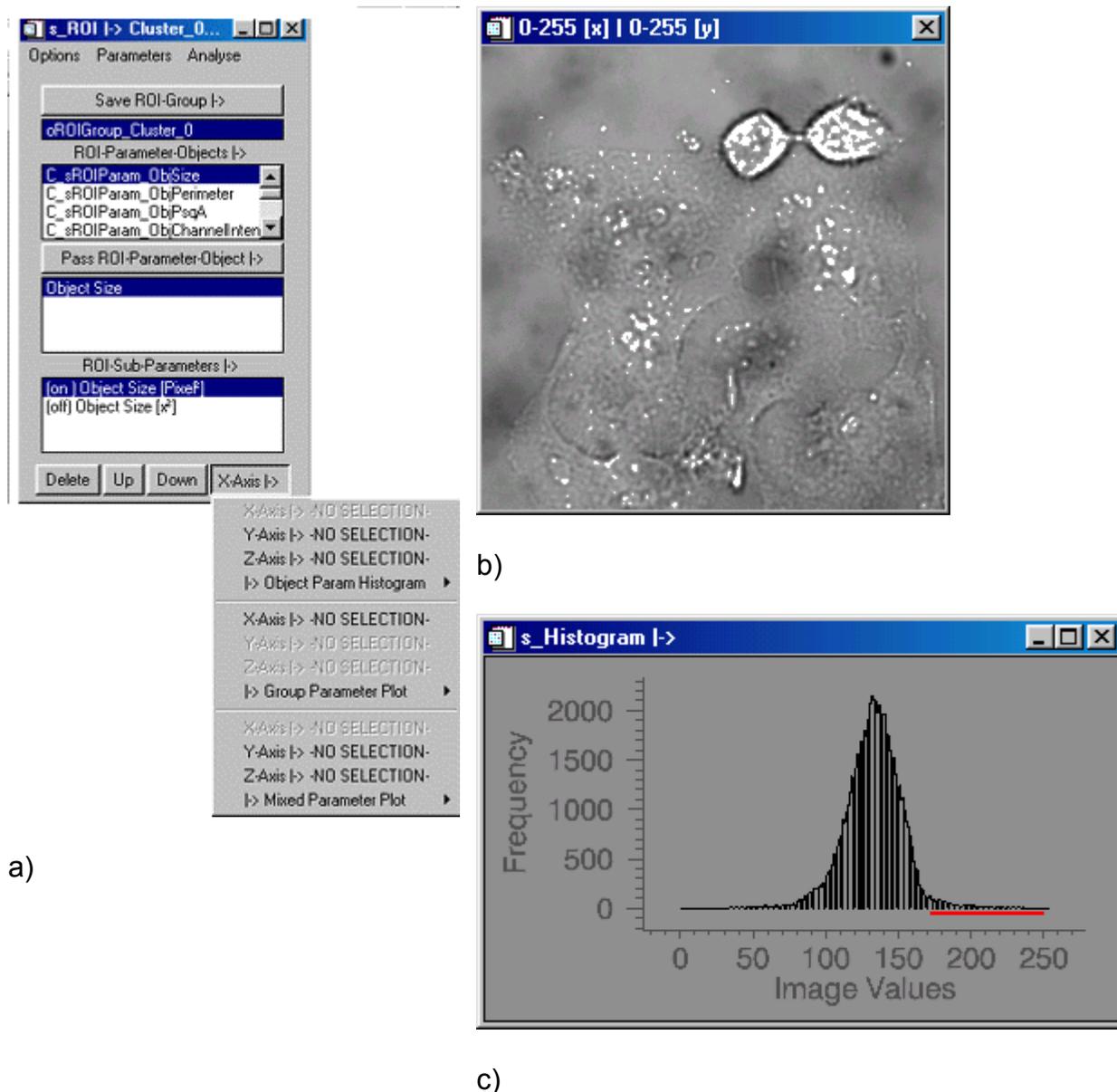


Figura 3.25: a) Transformaciones aplicadas sobre la IO, b) Resultado de aplicar a), c) Histograma de b).

### 3.9. Exportación de la Información.

Exportación de las imágenes de las ROI y posterior generación de un avi o mpeg. El sistema permite exportar o grabar el conjunto de imágenes con los objetos segmentado en formato .tif, con lo que se puede generar una película con el software MediaEditor, también es posible exportar los datos a un .txt, es decir posición y valor de los resultados.

En la Figura 3.26 se muestran dos imágenes extraídas de un ROI analizado con el sistema, y se muestran los objetos contenidos, en éste caso células, como integrante principal de la escena. Otro ejemplo son las imágenes del Anexo C y Anexo D, donde en este último sólo se muestra la Imagen Hela29 de la secuencia.

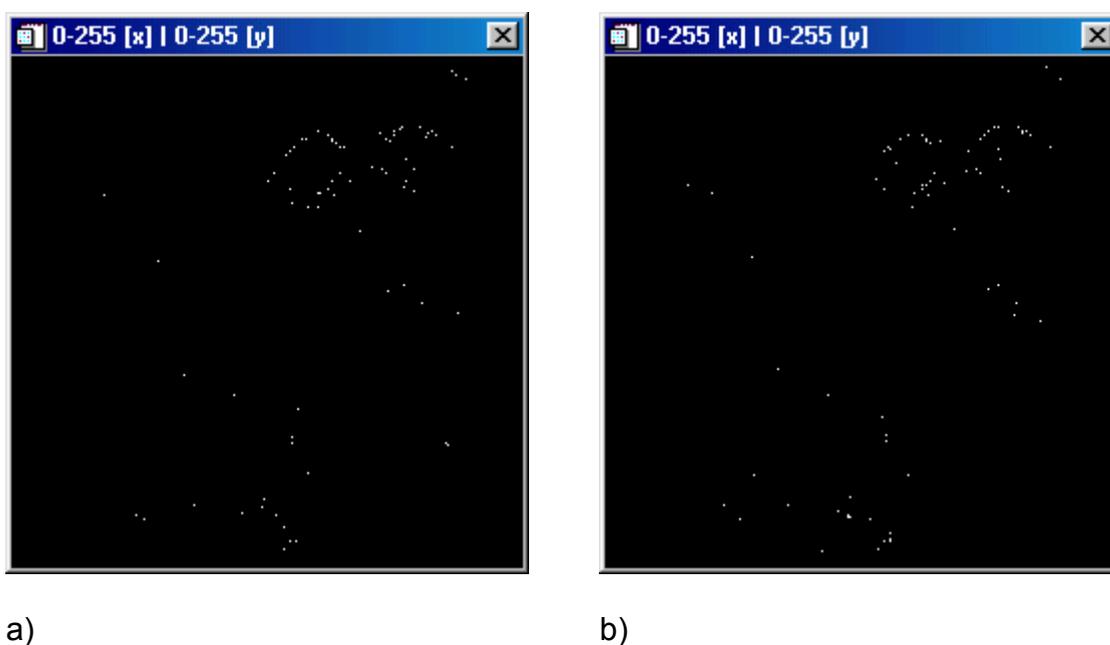


Figura 3.26: Muestra el resultado de algunas imágenes de la secuencia analizada.

### 3.10. Comparación Resultados.

En éste capítulo se compararán los resultados prácticos obtenidos utilizando la aplicación desarrollada, se analizarán dos secuencias de transformaciones sobre las mismas imágenes, las secuencias se denominaron S1, S2, S3, S4 y S5, y el orden de las secuencias es como se muestra en la Figura 3.27.

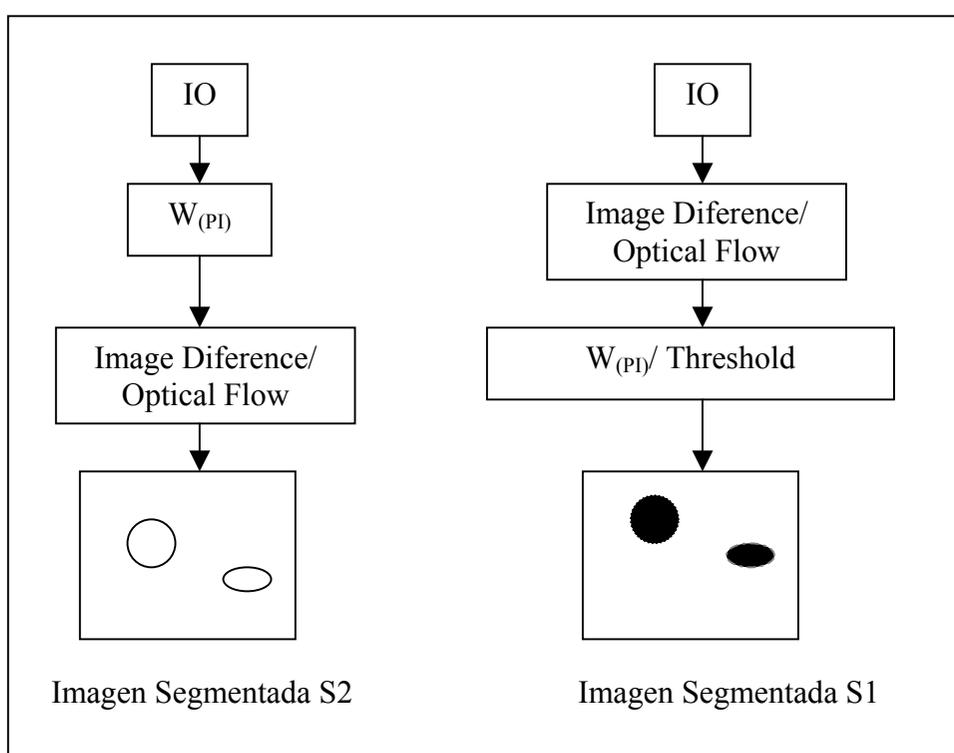


Figura 3.27: Esquema que se utilizará en la comparación de resultados.

Donde cada secuencia tiene las siguientes transformaciones:

S1.1: Recibe la IO, luego se aplica la transformada de Wavelets para obtener los IP, luego se utiliza la transformada OF, con lo que se obtiene la secuencia de imágenes segmentada. Ver los resultados en el Anexo A, Figura A.1.

S1.2: Recibe la IO, luego se aplica la transformada de Wavelets para obtener los IP, luego se utiliza la transformada Diferencia de Imágenes, con lo que se obtiene la secuencia de imágenes segmentada. Ver los resultados en el Anexo A, Figura A.2.

S2.1: Recibe la IO, luego se aplica la transformada Diferencia de Imágenes, luego se utiliza la transformada de Wavelets para obtener los IP, con lo que se obtiene la secuencia de imágenes segmentada. Ver los resultados en el Anexo A, Figura A.3.

S2.2: Recibe la IO, luego se aplica la transformada OF, luego se utiliza la transformada de Wavelets para obtener los IP, con lo que se obtiene la secuencia de imágenes segmentada. Ver los resultados en el Anexo A, Figura A.4.

S2.3: Recibe la IO, luego se aplica la transformada Diferencia de Imágenes, luego se utiliza la transformada de Threshold para obtener los IP, con lo que se obtiene la secuencia de imágenes segmentada. Ver los resultados en el Anexo A, Figura A.5.

S2.4: Recibe la IO, luego se aplica la transformada OF, luego se utiliza la transformada de Threshold para obtener los IP, con lo que se obtiene la secuencia de imágenes segmentada. Ver los resultados en el Anexo A, Figura A.6.

S3: Recibe la IO, Luego se aplica la transformada Diferencia de Imágenes. Ver los resultados en el Anexo A, Figura A.7.

S4: Recibe la IO, Luego se aplica la transformada OF. Ver los resultados en el Anexo A, Figura A.8.

S5: Recibe la IO, Luego se aplica la transformada Wavelets. Ver los resultados en el Anexo A, Figura A.9.

### Comentarios

En resumen la implementación y uso del sistema de Segmentación por Movimiento resultó ser completamente exitosa.

### **3.11.Trabajos Futuros.**

El desarrollo descrito en éste trabajo abarca sólo el ámbito de la segmentación por movimiento, sería interesante plantear la inquietud de llevar a cabo una técnica llamada Object Tracking, así como también la implementación de un Sprite del video analizado, (esto consiste en generar un análisis a largo plazo, es decir, generar una memoria especial que contiene toda la información de la imagen que fue visible a través de la secuencia), esto se consigue por medio de deformación controlada de la imagen. La generación del Sprite pertenece a las técnicas para la estimación de movimiento global a largo término.

## **CAPITULO IV.**

### **4. CONCLUSIONES.**

PDI ha sido un gran aporte en el análisis de imágenes de diversas áreas como se describió en la sección 1.1, esencial en la extracción de información de las imágenes aplicando las diversas técnicas de segmentación y transformaciones sobre las imágenes para la interpretación y análisis humano.

Se eligió el formato monocromático (HSV de las siglas en inglés Hue, Saturation, Value) éste fue el estándar elegido según los requerimientos, debido a que la implementación monocromática implica menos costo y un procesamiento más rápido, como se comprobó en la implementación.

Usualmente las imágenes a estudiar tienen una gran cantidad de ruido, producto de diversas fuentes externas a la escena, éste ruido dificulta los procesos de análisis y obtención de información como por ejemplo la segmentación. Para reducir la cantidad de ruido en las imágenes, existe la posibilidad de utilizar los métodos descritos en la sección 3.3, eliminando de ésta forma la influencia de los puntos muy altos o muy bajos que provocan distorsión en el análisis.

Se construyó un conjunto de herramientas que permitió extraer la información de interés a través de los IP. En particular en las imágenes de Hela permitió diferenciar a las células como objetos de interés, y en las demás imágenes permitió diferenciar las estructuras de los objetos presentes en las secuencias de su fondo, permitiendo de ésta obtener la información de interés presente en

las secuencias, lo que permite concluir que los métodos utilizados arrojaran información con un alto grado de certeza, esto se puede comprobar a simple vista con los resultados de los IP, con lo que se comprobó que el método de Wavelets es muy robusto.

Al aplicar IP a la secuencia de imágenes y perseguir los IP en el tiempo, se ve a simple vista que la información representa satisfactoriamente a los puntos de mayor consideración o interés dentro de las imágenes (Ver Figura 3.13).

Reconocimientos de IP sirve para simplificar el análisis de imágenes, realizando una minimización de la cantidad de puntos a ser procesados, mientras que al mismo tiempo se preserva la información estructural de la imagen. Esta reducción de la información se logra en base a alguna propiedad característica, el principal parámetro de discriminación utilizado en éste trabajo fue el movimiento.

El análisis de los IP permitió capturar y perseguir el comportamiento de cada elemento de interés, como el aumento o disminución de los tamaños en las células. Y en el aspecto de la reducción de la cantidad de puntos a procesar, se tiene que, se logra reducir considerablemente el tiempo de cálculo necesario para procesar las secuencias de imágenes.

De las Figuras 3.16-c) y 3.17.c) se puede concluir que el movimiento se manifiesta como cambios temporales en los valores de gris, pero no se puede afirmar que todo cambio temporal en los niveles de gris, se encuentra asociado a movimiento.

La aplicación del OF ayudó en el reconocimiento de objetos diferentes, debido a sus distintas velocidades, conociendo la velocidad real de los objetos se puede reconocer la forma de los mismos, la estructura y su orientación.

La implementación del presente trabajo permitió el estudio de secuencias de objetos en movimiento, si bien el enfoque estaba centrado en las muestras celulares es posible estudiar también imágenes provenientes de otras áreas y los resultados se consideran exitosos.

El objetivo final, segmentación por movimiento, se cumplió, es decir, el objetivo de poder segmentar células que presentan movimiento en la secuencia de imágenes, fue posible con el desarrollo aquí descrito.

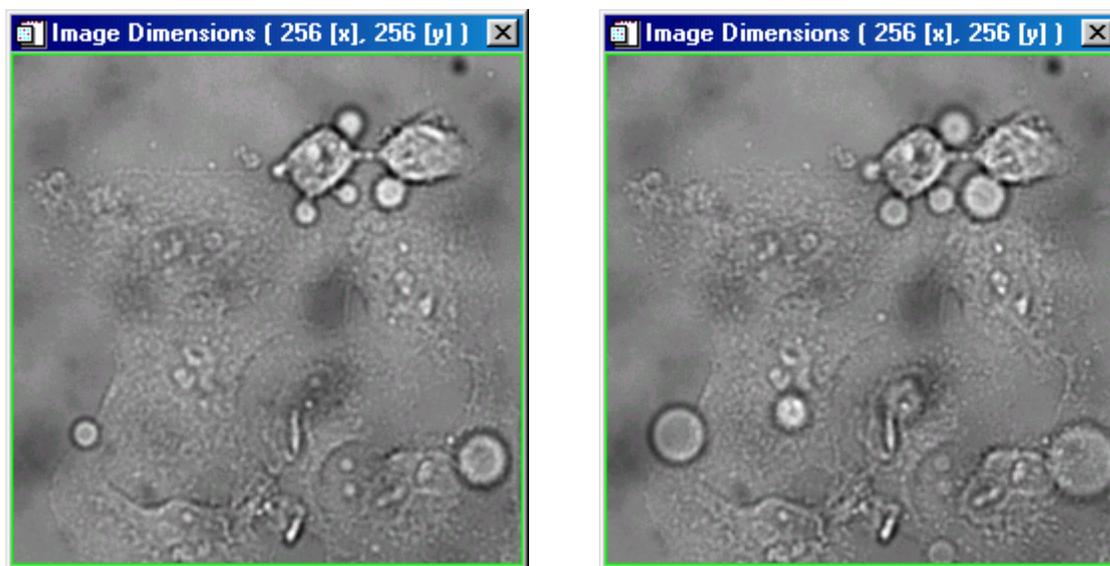
## 5. BIBLIOGRAFÍA.

- [BAR2002] Pamela L. Barría, Rodrigo P. Rojas, Víctor H. Espinoza.  
2002  
“Plataforma de Identificación, Clasificación y Análisis de Imágenes, obtenidas a partir de Muestras Celulares”  
Universidad Austral de Chile
- [BEA99] Steven S. Beauchemin  
1999  
<http://www.scs.carleton.ca/~beau/home.html>
- [BRA98] Antonio J. Bravo  
1998  
“Procesamiento Digital de Imágenes”  
<http://www.ing.ula.ve/~abravo/document/tutorial/imagenes/indice.html>
- [CAS96 ] K. R. Castleman  
1996  
“Digital Image Processing”  
Prentice hall
- [COM02] “Computer Vision Test Images”  
2002  
<http://www-2.cs.cmu.edu/~cil/v-images.html>
- [FIS02] Robert Fisher  
2002  
“Motion and Time Sequence Analysis”  
<http://www.dai.ed.ac.uk/CVonline/motion.htm>

- [GON02] Rafael C. Gonzalez, Richard E. Woods  
2002  
“Digital Image Proccesing Second Edition”  
Prentice Hall
- [HAR02 ] PhD. Steffen Härtel  
2002  
“Métodos de procesamiento de imágenes y su aplicación en el análisis de estructuras, de eventos dinámicos y de cinéticas en sistemas biológicos”
- [HAR01] PhD. Steffen Härtel, M. Luisa Oyarzo, Rodrigo P. Rojas.  
2001  
“Representation and analisys of visual motion”
- [MPE02] MPEG7  
2002  
“MPEG-7 Overview (version 8)”  
<http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>
- [OHM99] Jens-Rainer Ohm  
1999  
“Bildverarbeitung” (“Procesamiento de Imágenes”)  
<http://bs.hhi.de/users/ohm/>
- [RES03] Research Systems, Inc.  
2003  
<http://www.rsinc.com/>
- [SON98] Milan Sonka, Vaclav Hlavac, Roger Boyle  
1998  
“Image Processing, Analysis, and Machine Vision” (2nd Edition)

<http://www.icaen.uiowa.edu/~dip/LECTURE/lecture.html>

## Anexo A: Análisis de Imágenes con el Método Propuesto.



a)

b)

Figura A.1: a) Muestra Hela21, b) Muestra Hela30.

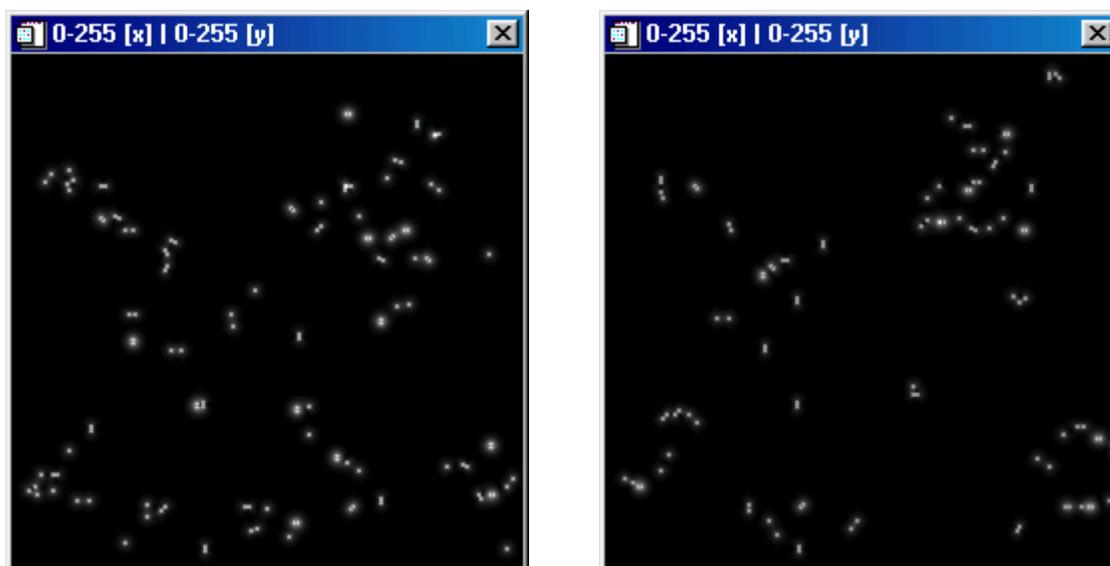


Figura A.2: Fig. A.1: Muestra resultado de S1.1.

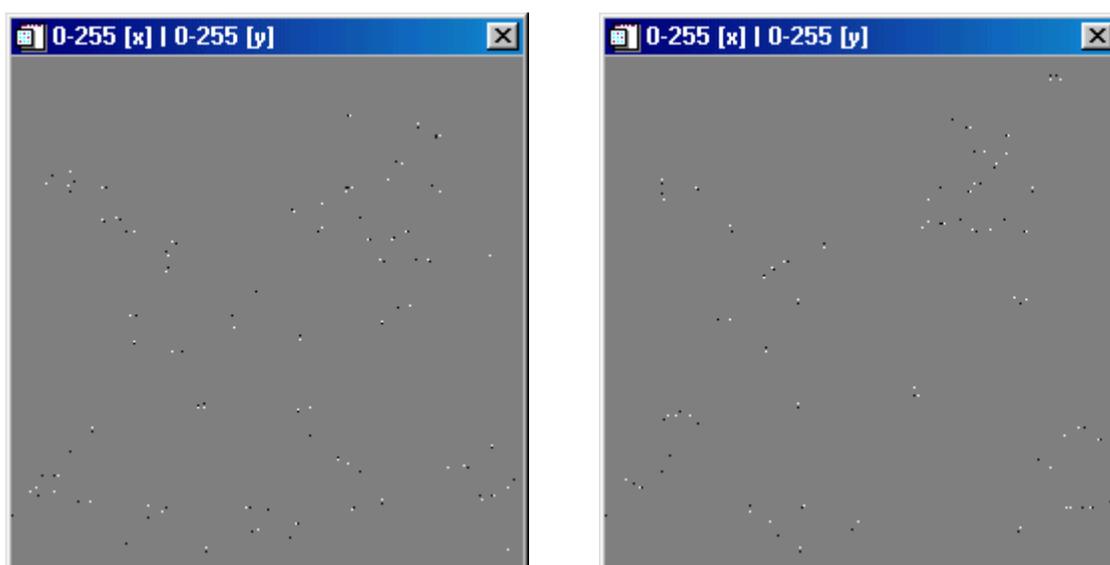


Figura A.3: Muestra resultado de S1.2.



Figura A.4: Muestra resultado de S2.1.



Figura A.5: Muestra resultado de S2.2.

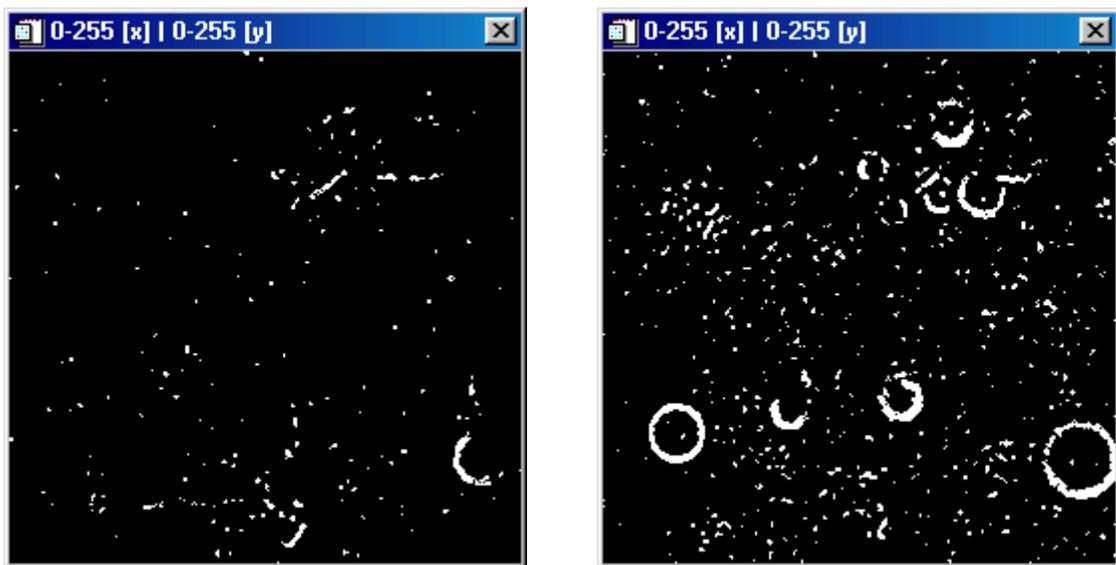


Figura A.6: Muestra resultado de S2.3.

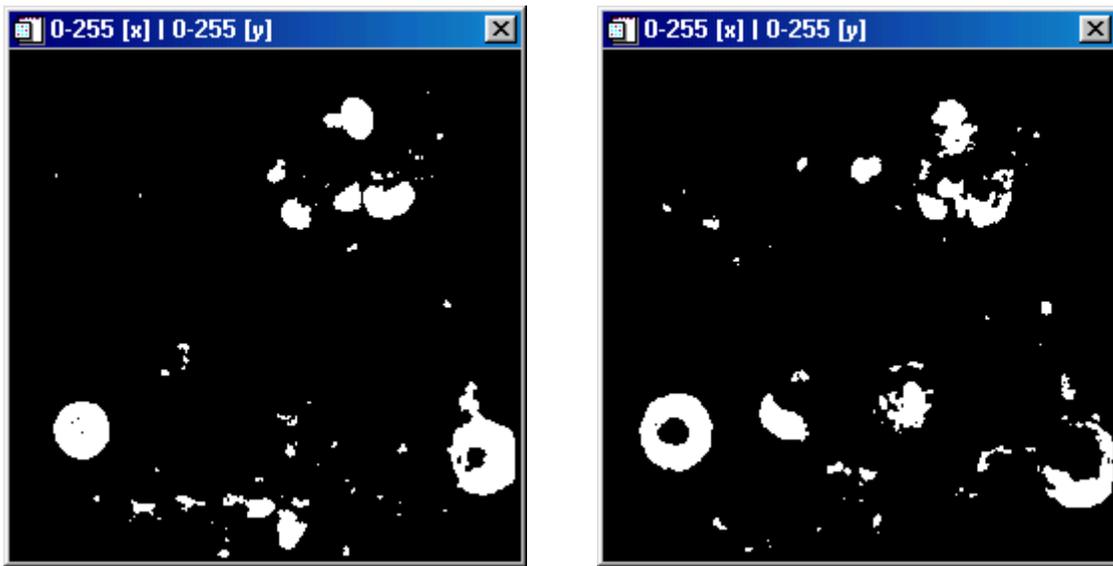


Figura A.7: Muestra resultado de S2.4.

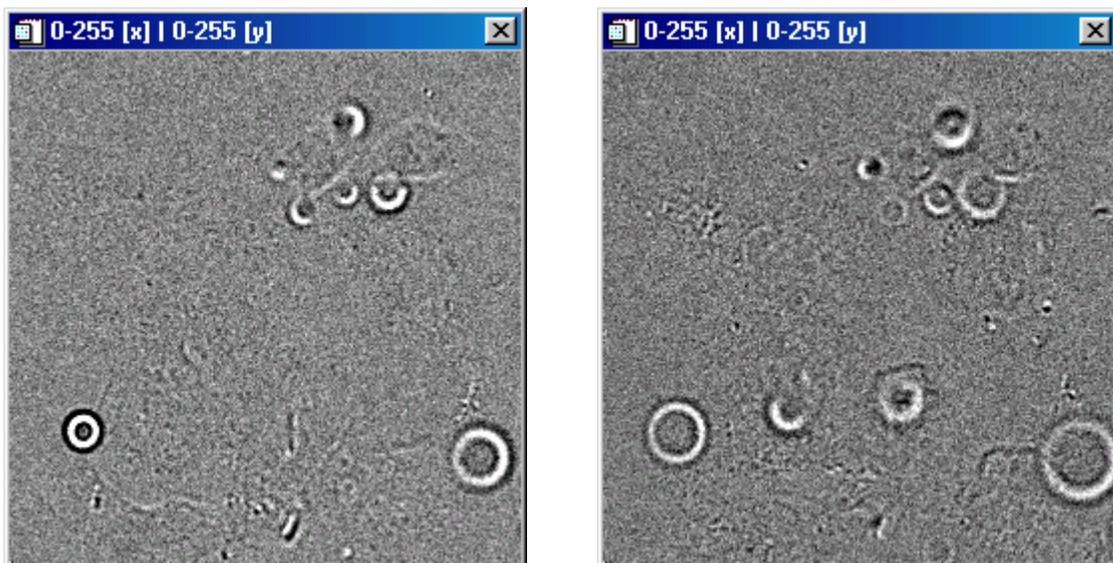


Figura A.8: Muestra resultado de S3.

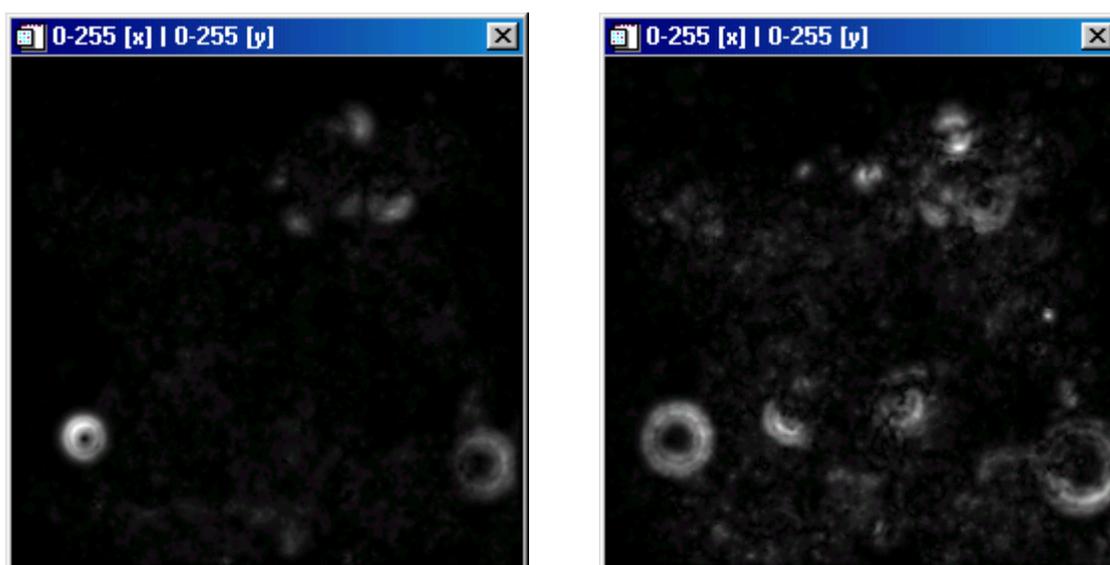


Figura A.9: Muestra resultado de S4.



Figura A.10: Muestra resultado de S5.

## Anexo B: Análisis Interactivo de los Objetos.

Es posible realizar un análisis interactivo de los objetos obtenidos, a partir de la ROI, se pueden analizar por ejemplos los bordes de los objetos segmentados, y realizar a la vez un escalamiento de dichos bordes, tal como lo muestra la figura B.1.

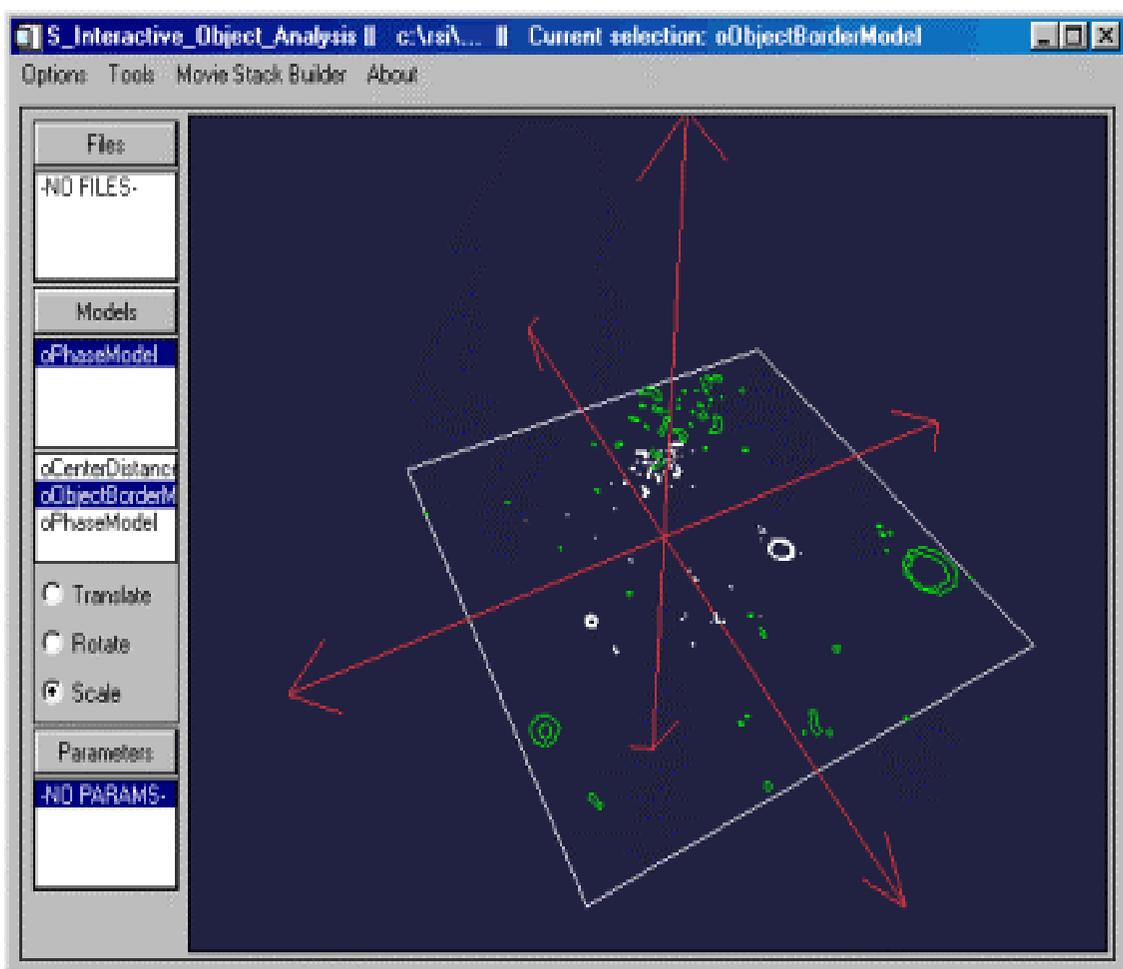


Figura B.1: Análisis interactivo de los resultados.

## Anexo C: Imágenes sección 3.6.

Imágenes obtenidas al aplicar el filtro “Optical Flow” a la secuencia de imágenes Hela, se probó con las imágenes Hela20 al Hela30 del set. A continuación se muestran los resultados y sus respectivos VD.

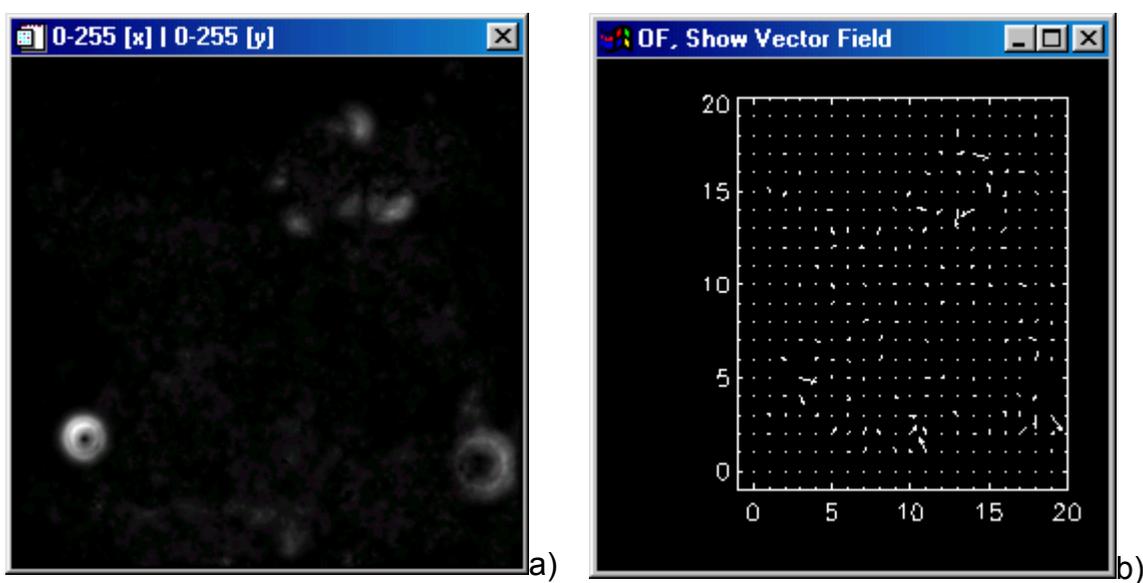


Figura C.2: a) OF y b) VD Hela20

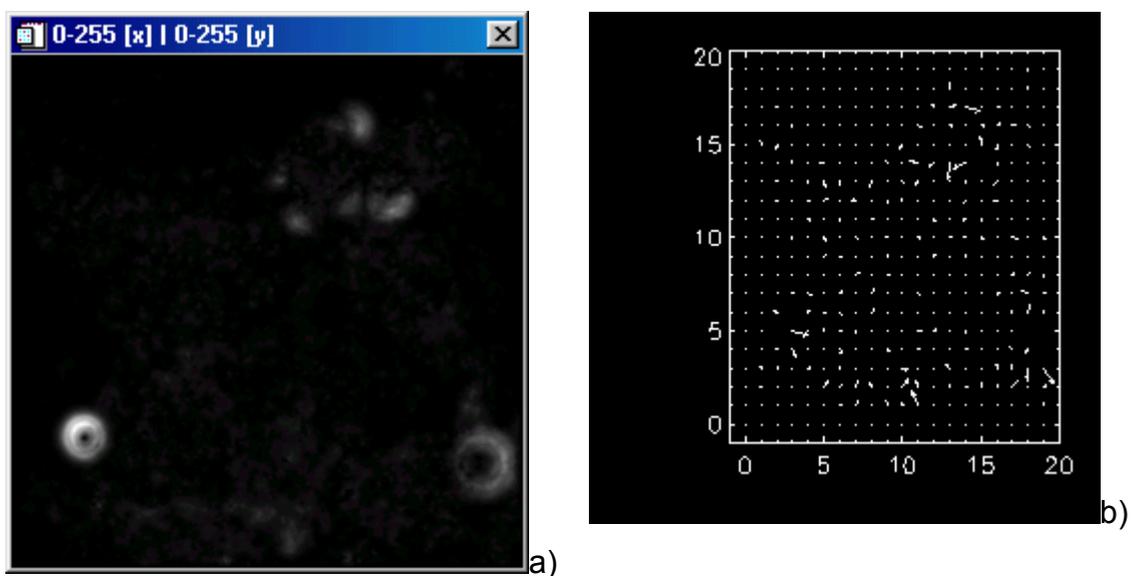


Figura C.3: a) OF y b) VD Hela21

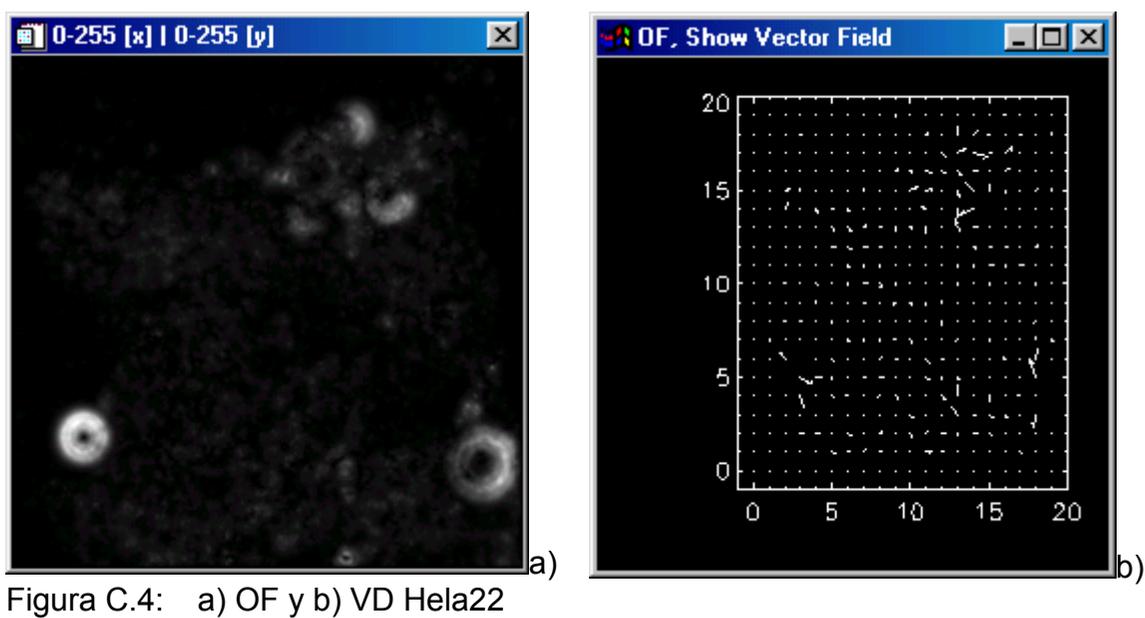


Figura C.4: a) OF y b) VD Hela22

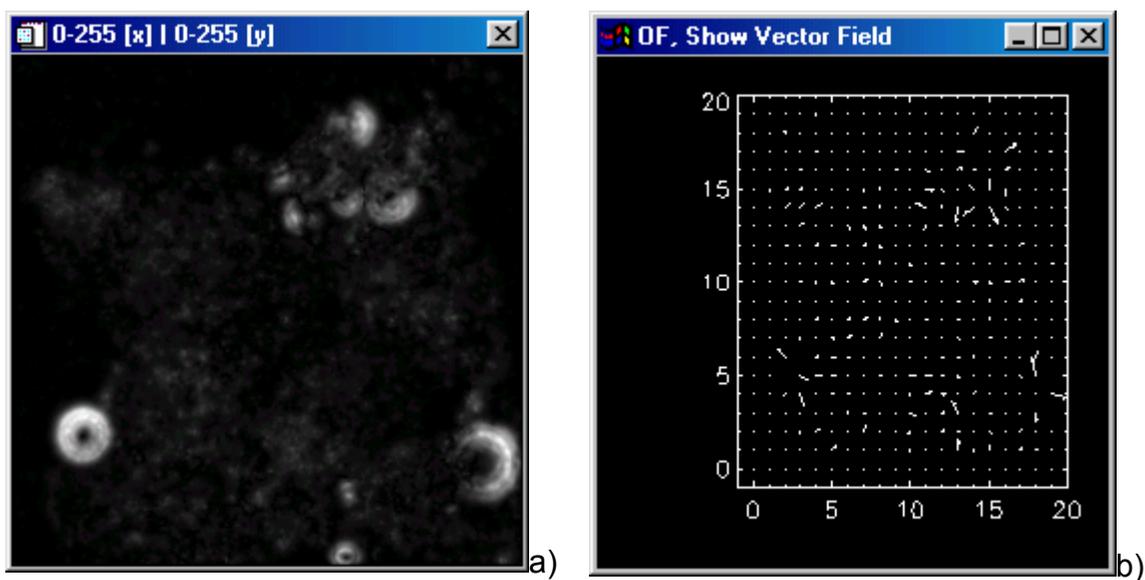


Figura C.5: a) OF y b) VD Hela23

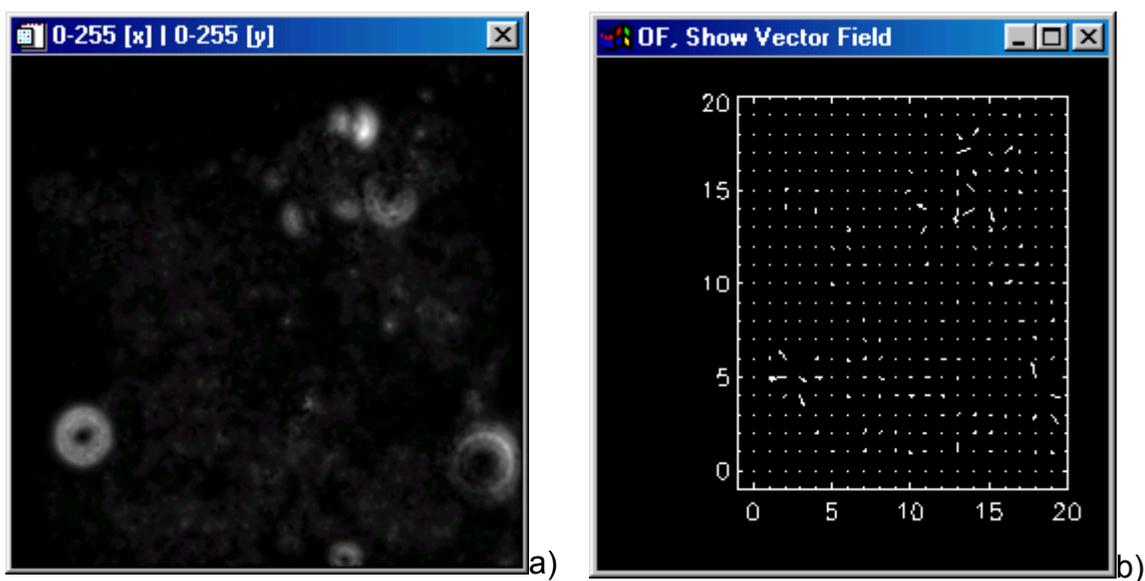
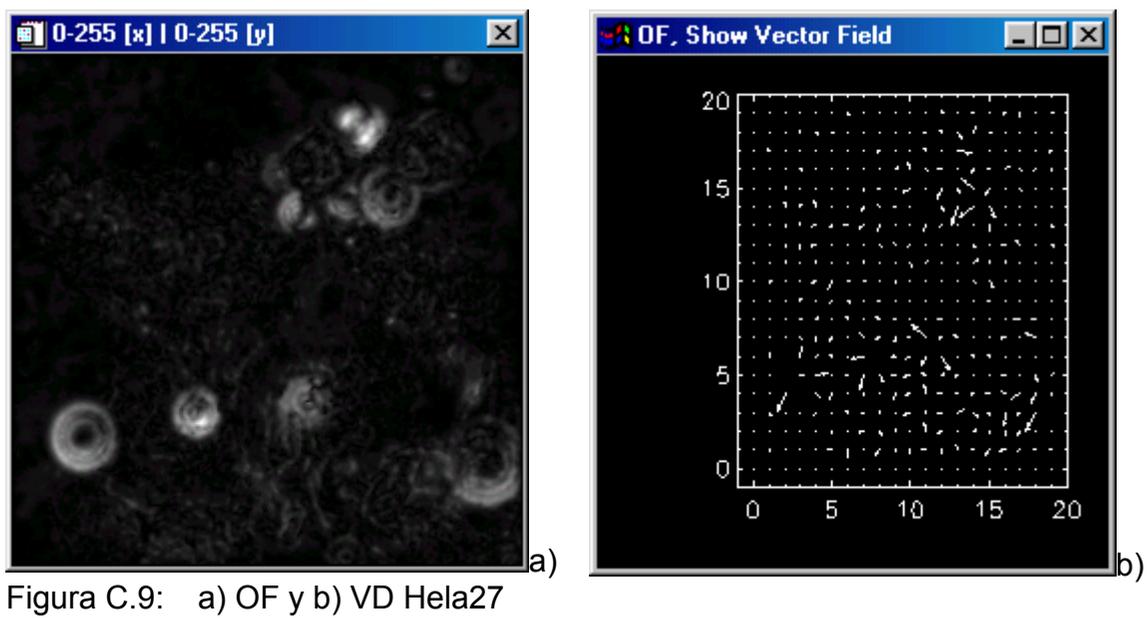
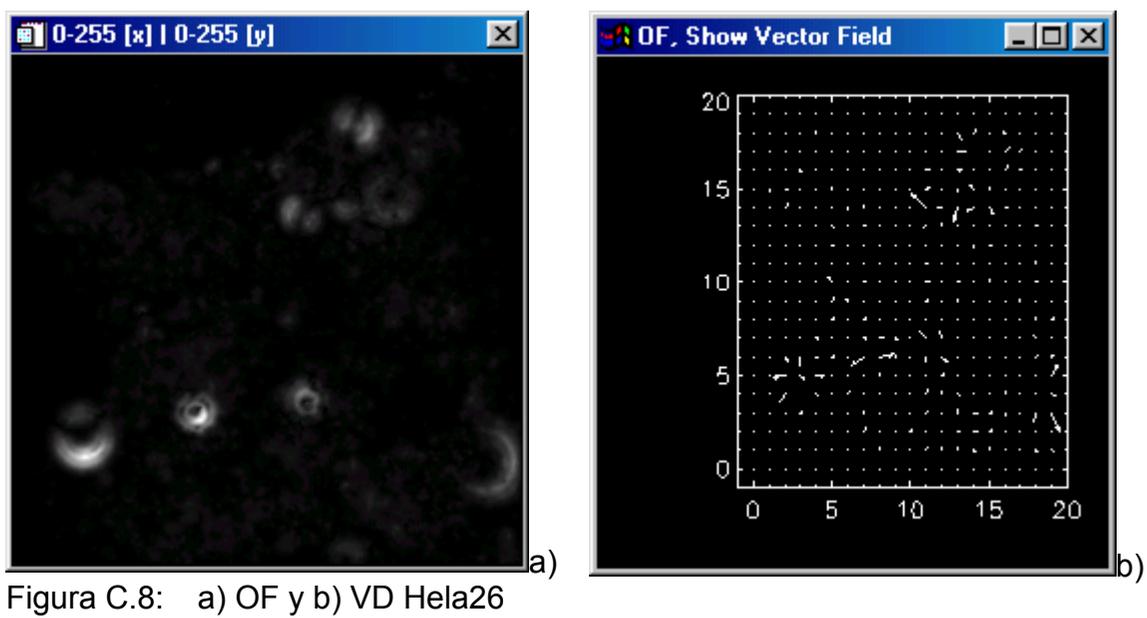
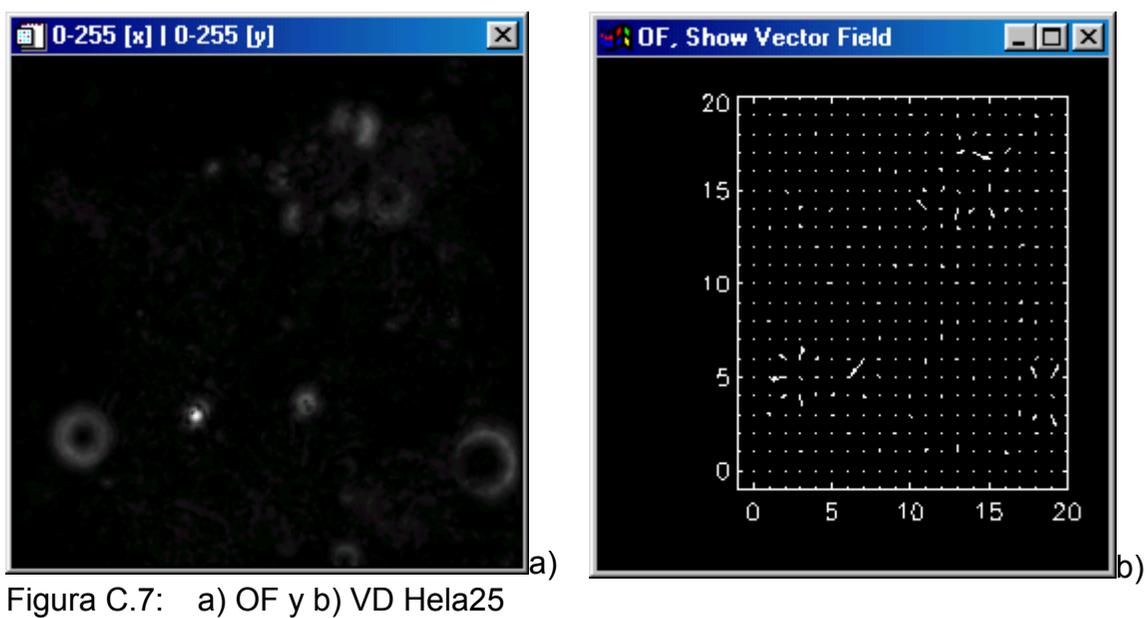


Figura C.6: a) OF y b) VD Hela24



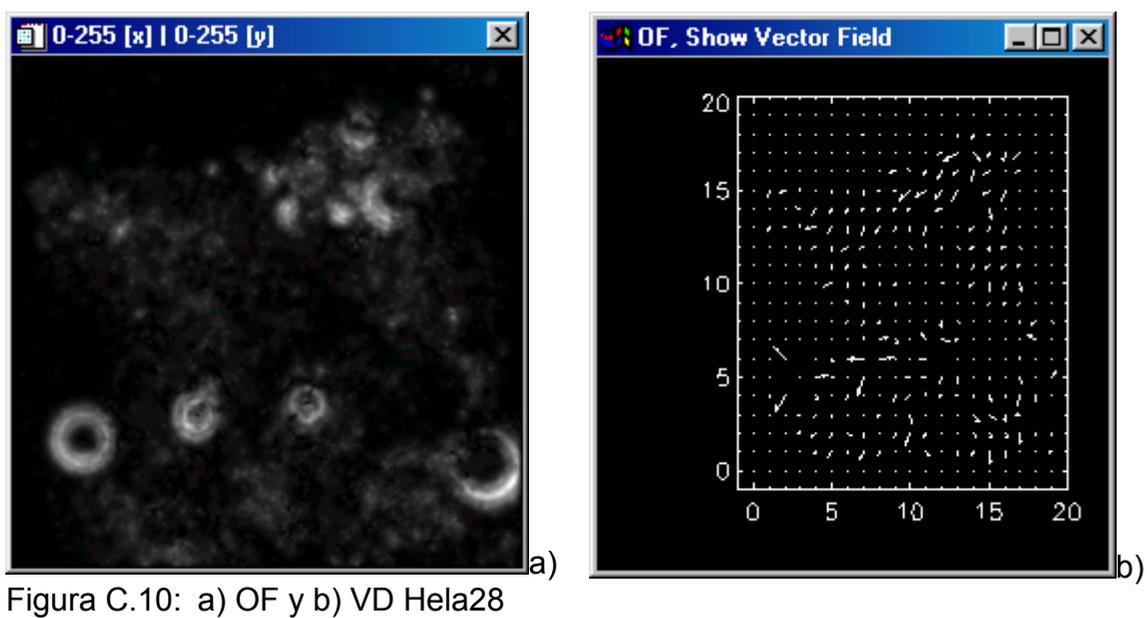


Figura C.10: a) OF y b) VD Hela28

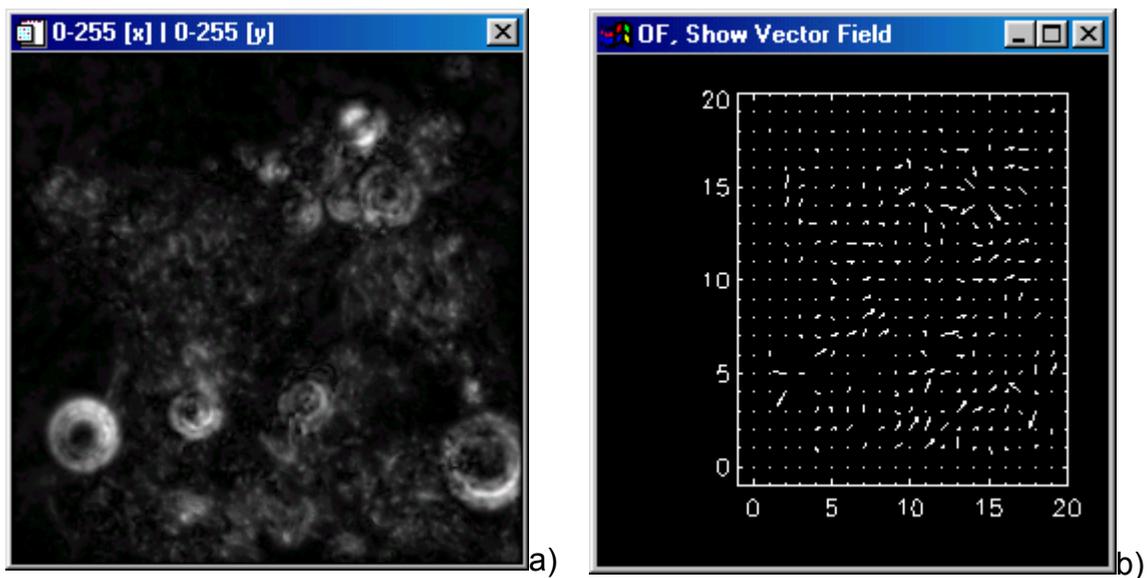


Figura C.11: a) OF y b) VD Hela29

## Anexo D: Imágenes sección 3.7.

La imagen que se utilizó para obtener los resultados que se muestran a continuación corresponde a la imagen Hela29 a la que se le aplicaron los filtros OF y Threshold (similar a la obtención del resultado S2.4 de la sección 3.10).

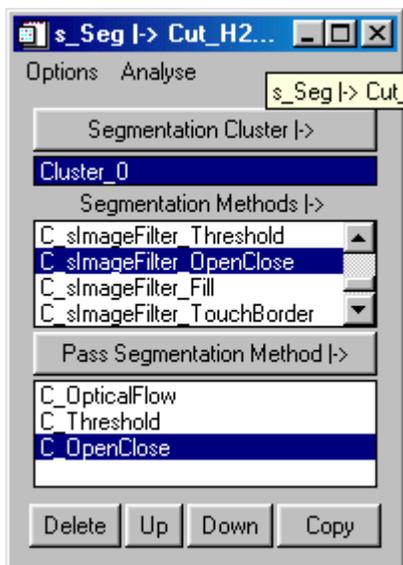


Figura D.1: Transformaciones aplicadas a Hela29



Figura D.2: Hela29 bajo OF y Threshold

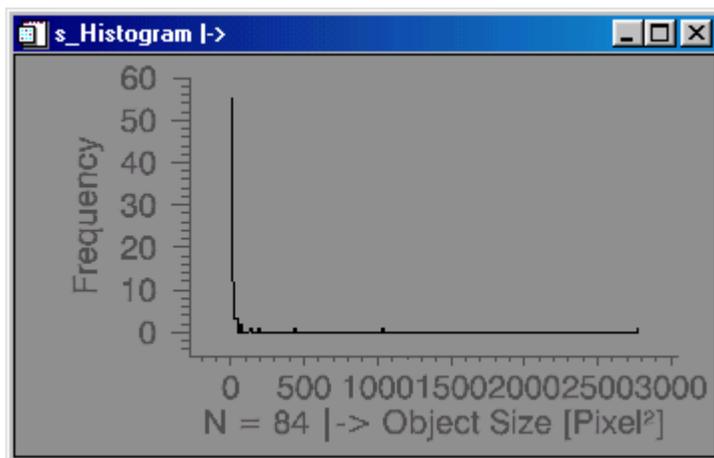


Figura D.3: Parámetro Object Size (Pixel<sup>2</sup>)

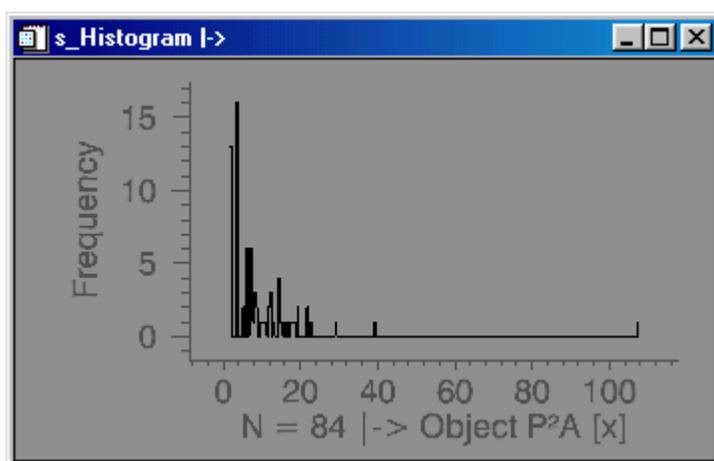


Figura D.4: Parámetro Object Perimeter (x)

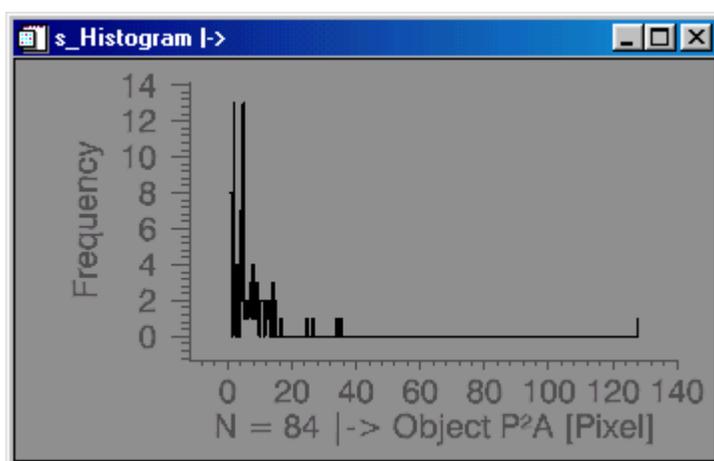


Figura D.5: Parámetro Object Perimeter (Pixel)