

**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**



**PROCESAMIENTO DIGITAL DE IMÁGENES:  
METODOLOGÍA DE TONOS DE GRISES Y PSEUDO-  
COLOR**

**INFORME DE SUFICIENCIA**

**PARA OPTAR EL TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO**

**PRESENTADO POR:**

**PEDRO SAMUEL ARCE CHIRINOS  
PROMOCIÓN  
1979 - I**

**LIMA – PERÚ  
2006**

**PROCESAMIENTO DIGITAL DE IMÁGENES:  
METODOLOGÍA DE TONOS DE GRISES Y PSEUDO-COLOR**

Dedico este trabajo a:

A mis padres Doña Luzmila y Don Pedro Javier por la dedicación, esfuerzo  
y enseñanzas en mi formación como persona,

A mi esposa Elena del Pilar por su comprensión, apoyo y por constituir la  
razón principal de mi superación.

## SUMARIO

Este Informe de Suficiencia trata de explicar el tratamiento de imágenes, mediante el procesamiento digital de las mismas utilizando la metodología conocida como Tonos de Grises, así como los procesos denominados Pseudo-color, mediante la utilización de procedimientos y programas desarrollados en MATLAB, y la utilización de funciones de lectura o carga, filtrado, descomposición, escritura o grabación e impresión gráfica de los resultados alcanzados, luego de la aplicación de las metodologías utilizadas.

En los Capítulos I al IV se presenta los antecedentes e historia de la visión artificial, bases teóricas de la formación de las imágenes, desarrollado en los temas de origen de los datos, el muestreo y la cuantificación, formación y transformación, modelos de ruido, relación entre píxeles, entre otros, así como los pasos fundamentales en el procesamiento de imágenes, la descripción de transformada de Fourier como herramienta matemática para el tratamiento de imágenes y la descripción de los procedimientos para la obtención de niveles de gris para el caso de las imágenes a color, así como el proceso de pseudocolor para colorear imágenes en niveles de gris.

Finalmente se presentan los casos prácticos de estudio mediante la utilización de las metodologías de tonalidades de grises y pseudo-color en el procesamiento de imágenes, y los resultados obtenidos con la aplicación de los programas desarrollados para aplicar las metodologías mencionadas, incluyendo algunas aplicaciones prácticas en las cuales la utilización de las metodologías presentadas resulta o podría ser muy relevante.

## ÍNDICE

<b>INDICE GENERAL</b>	VI
<b>ÍNDICE DE FIGURAS Y TABLAS</b>	VIII
<b>PROLOGO</b>	1
<b>CAPÍTULO I</b>	
<b>ANTECEDENTES E HISTORIA</b>	3
1.1 Modelos de la Percepción Visual	3
1.2 Visión Artificial	5
1.3 Problemas de la Percepción	7
1.4 Métodos de Visión por Computadora	12
1.5 Antecedentes y Desarrollo del Procesamiento de Imágenes Digitales	13
<b>CAPÍTULO II</b>	
<b>FORMACIÓN DE LA IMAGEN</b>	16
2.1 Origen de los Datos	16
2.2 El Ojo como cámara	17
2.3 Tipos de Sensores e Imágenes	20
2.4 Efectos del Muestreo y Cuantificación	20
2.5 Formación de la Imagen	21
2.6 Transformación	25
2.7 Modelos de Ruido	25
2.8 Relaciones entre Píxeles	26
2.9 Pasos Fundamentales en el. Procesamiento de Imágenes	28
<b>CAPÍTULO III</b>	
<b>TRANSFORMADA DE FOURIER</b>	32
3.1 Concepto y Representación de Señales Continuas	32
3.2 La Imagen como Señal	32

3.3 La Transformada de Fourier	33
3.4 La Transformada de Fourier (TFD) de una Imagen Digital	34
<b>CAPITULO IV</b>	
<b>OBTENCIÓN DE GRISES Y PSEUDOCOLOR</b>	38
4.1 Conversión a Escala de Grises	38
4.2 Pseudo-Color y Color Falso	39
<b>CAPITULO V</b>	
<b>CASOS PRACTICOS DE ESTUDIO</b>	44
5.1 Obtención de Tonos de Grises y Filtrado	44
5.2 Pseudo Color – Color Falso	54
<b>CONCLUSIONES</b>	61
<b>ANEXO A</b>	66
<b>ANEXO B</b>	79
<b>BIBLIOGRAFÍA</b>	91

## INDICE DE FIGURAS Y TABLAS

Figura N° 1.1 - Sistema de Visión de Propósito General	7
Figura N° 1.2.- Espiral de Fraser o Ilusión de Espirales	8
Figura N° 1.3.- Ilusiones Ópticas	9
Figura N° 1.4.- Poseidón o Sirena	10
Figura N° 1.5.- Ejemplos de Ambigüedad de Imágenes	10
Figura N° 1.6.- Litografía de Escher – Consistencias Geométricas	11
Figura N° 2.1 – Ojo Humano	17
Figura N° 2.2 – La Retina.	19
Figura N° 2.3 – Caminos Visuales del Ojo al Cerebro	19
Figura N° 2.4 – Muestreo de Imágenes	20
Figura N° 2.5 – Efecto de la Cuantificación (8 y 1 bits)	21
Figura N° 2.6.- Formación de Imagen 2D	22
Figura N° 2.7.- Imagen Original y Desenfocada	24
Figura N° 2.8.- Imagen Original y con Ruido Gaussiano	26
Figura N° 2.9 – Esquema de Procesamiento de Imágenes	29
Figura N° 3.1 – Muestreo	35
Figura N° 4.1 – Asignación de Color	41
Figura N° 4.2 – Imagen en Niveles de Gris de Precipitación Fluvial	42
Figura N° 4.3 – Imagen Pseudo-color de Precipitación Fluvial	42
Figura N° 5.1 – Imágenes Originales (4 Casos)	45
Figura N° 5.2 – Pantalla para seleccionar Archivo Original (*.jpg)	46

Figura N° 5.3 - Pantalla para seleccionar Archivo Original (*.tif)	47
Figura N° 5.4 - A: Original y Tonos de Grises (Caso 1)	48
Figura N° 5.4 - B: Resultados (Caso 1)	49
Figura N° 5.5 - A: Imagen Original y Tonos de Grises (Caso 2)	50
Figura N° 5.5 - B: Resultados (Caso 2)	51
Figura N° 5.6 - A: Imagen Original y Tonos de Grises (Caso 3)	52
Figura N° 5.6 - B: Resultados (Caso 3)	52
Figura N° 5.7 - A: Imagen Original y Tonos de Grises (Caso 4)	53
Figura N° 5.7 - B: Resultados (Caso 4)	53
Figura N° 5.8: Imagen Original – Caso “A”	56
Figura N° 5.9: Imagen Original – Caso “B”	56
Figura N° 5.10: Menú Principal	57
Figura N° 5.11: Resultados Pseudo-color Caso “A”	58
Figura N° 5.12: Imagen hsv 0.00 – 0,60 - Caso “B”	59
Figura N° 5.13: Imagen hot 0.00 – 0.80 - Caso “B”	60
Figura N° 5.14: Imagen flag 0.00 – 0,80 - Caso “B”	60
Tabla N° A.1: Principales Funciones y Comandos Utilizados	72
Figura N° B.1: Imagen Original – Caso “A”	85
Figura N° B.2: Imagen “hot” 0.00 – 0.80 – Caso “A”	85
Figura N° B.3: Imagen “default” 0.00 – 0.80 – Caso “A”	85
Figura N° B.4: Imagen “cool” 0.00 – 0.80 – Caso “A”	86
Figura N° B.5: Imagen “bone” 0.00 – 0.80 – Caso “A”	86
Figura N° B.6: Imagen “flan” 0.00 – 0.80 – Caso “A”	86
Figura N° B.7: Imagen “pink” 0.00 – 0.80 – Caso “A”	87
Figura N° B.8: Imagen “hsv” 0.00 – 0.80 – Caso “A”	87
Figura N° B.9: Imagen “gris” 0.00 – 0.80 – Caso “A”	87
Figura N° B.10 Imagen Original – Caso “B”	88
Figura N° B.11: Imagen “hot” 0.00 – 0.80 – Caso “B”	88



Figura N° B.12: Imagen “default” 0.00 – 0.80 – Caso “B”	88
Figura N° B.13: Imagen “cool” 0.00 – 0.80 – Caso “B”	89
Figura N° B.14: Imagen “bone” 0.00 – 0.80 – Caso “B”	89
Figura N° B.15: Imagen “flan” 0.00 – 0.80 – Caso “B”	89
Figura N° B.16: Imagen “pink” 0.00 – 0.80 – Caso “B”	90
Figura N° B.17: Imagen “hsv” 0.00 – 0.80 – Caso “B”	90
Figura N° B.18: Imagen “gris” 0.00 – 0.80 – Caso “B”	90

## PRÓLOGO

En el marco del procesamiento digital de señales en general, el procesamiento digital de imágenes ha sido de especial importancia en los últimos años, en especial por el desafío de simular el proceso de la visión humana, así como de la necesidad de lograr la representación digital de una imagen, con el objeto de obtener su almacenamiento y preservación, para un posterior análisis por parte de algún usuario (humano) o un sistema de visión artificial. El procesamiento digital de imágenes incluye un conjunto de técnicas que permiten la representación digital de una imagen, las que son aplicadas en función de la necesidad de modificar la imagen o alguna parte de ella, para su almacenamiento y preservación, para realzar, medir contrastar o reconocer y clasificar dicha imagen o algún elemento contenida en la misma.

El tema se abordó primero estableciendo de manera general el significado y alcance del Procesamiento de Imágenes, definiendo algunas características y propiedades importantes en el procesamiento de imágenes, como son la captura, discretización, cuantificación, filtrado, etc., llegando finalmente a implementar y probar algunos algoritmos aplicables según el tipo de tratamiento que se tiene que hacer a las imágenes, para determinados fines, como por ejemplo en el caso de señales biomédicas, o ante la necesidad de almacenar o transferir importantes volúmenes de información correspondiente a imágenes.

En la actualidad el avance y desarrollo tecnológico alcanzado en lo que respecta a la creación de software, su utilización de forma masiva, la concepción de código libre y estructura abierta, en adición a la evolución de los lenguajes de programación, permite reemplazar algunas implementaciones que anteriormente solo era posible realizar a nivel de hardware. En el caso específico del tratamiento de imágenes, se cuenta con herramientas de tecnología de información como el MATLAB y otros, que permiten realizar procesos de tratamiento, reconocimiento y transferencia de información, de manera más eficiente, con mucha versatilidad y seguridad, en adición a las capacidades de proceso

en términos de velocidad, volumen de información y precisión, entre otros, con que en la actualidad se disponen en los equipos de cómputo.

En el presente informe, las técnicas utilizadas abarcan la transformación de formatos, el mejoramiento de imágenes, tratamiento para fines de almacenamiento, preservación y análisis, aplicables en diversos campos de la gestión de información, que facilitan los procedimientos actuales, economizando recursos y tiempo para el análisis y evaluación de la información de los elementos presentes en cada imagen.

Aunque existen una serie de procesos que se pueden aplicar al procesamiento de imágenes, se ha creído conveniente incluir una relación de funciones que MATLAB tiene incluido para este fin, indicando sus funcionalidades y descripción de las mismas, así como otras desarrolladas para utilizar con MATLAB.

## **CAPÍTULO I**

### **ANTECEDENTES E HISTORIA**

El interés en estudiar y llegar a reproducir los mecanismos de procesamiento y representación de la información visual que percibe un ser vivo se remontan a los orígenes de la ciencia y la filosofía. Desde la época de Platon y Aristoteles hasta nuestros días se han formulado diversas teorías sobre como percibe el ser humano su realidad exterior y de que manera usa y representa la información captada por sus sentidos. Más recientes en la historia están los estudios de Kepler sobre el análisis geométrico de la formación de la imagen en el ojo, los de Newton sobre visión en color y dentro ya de este siglo los de Helmholtz (1910) sobre óptica fisiológica y Wertheimer (1912) sobre el movimiento aparente de agrupaciones de puntos o campos. Estos trabajos, entre otros, han establecido las bases de las actuales teorías de percepción visual.

Sin embargo a principios del siglo XX se dio un gran paso, con la aparición de la escuela de psicología de la Gestalt a partir de los trabajos de Wertheimer. Esta escuela estableció que los criterios de similaridad, homogeneidad, cercanía, entre otros, eran suficientes para tratar de explicar las propiedades de los mecanismos psicofísicos de la visión humana. Sin embargo, más tarde esto no fue suficiente para explicar nuevas interrogantes dentro del proceso de percepción visual.

Luego del fracaso de las teorías de la Gestalt, se originó una nueva corriente de teorías que con base en las potencialidades de las neuronas como células individuales trataron de explicar los mecanismos y propiedades de la visión humana, Hubel & Weisel (1962, 1968) y Barlow (1972). Esta aproximación tampoco pudo dar respuesta a los problemas de mayor detalle, de como nuestro mecanismo visual codifica, representa y reconoce distintos tipos de informaciones espaciales.

#### **1.1 Modelos de la Percepción Visual**

Para una mejor comprensión de la Visión Artificial, se analizará de manera muy general los modelos de percepción visual, para lo cual lo primero que debemos conocer es que

significa percepción. Según la Real Academia de la Lengua, percepción es: *“acción y efecto de percibir y también sensación interior que resulta de la impresión material hecha en nuestros sentidos”*.

Existen diversas teorías sobre la percepción visual, las cuales no son motivo del presente informe, sin embargo las que nos interesan son las de punto de vista de la Inteligencia Artificial y la Visión Artificial.

Ante la aparición de los ordenadores o computadores muchos investigadores se abocaron a realizar trabajos sobre percepción visual, siendo uno de los pioneros el grupo de Azriel Rosenfeld, mediante la aplicación de criterios matemáticos para aproximar dichos procesos mediante la estimación de bordes y líneas, cuya desarrollo de técnicas de aproximación ha continuado hasta nuestros días con gran certeza en ciertos casos.

Otro grupo idealizó el estudio a bloques blancos mate iluminados sobre fondo negro, los que tenían cualquier forma, siempre que las superficies fuesen planas y todos los bordes rectos. Esta simplificación fue para facilitar el estudio y evaluación, para luego extrapolar dichos conocimientos a figuras más complejas. Lamentablemente esto no ocurrió así, requiriéndose estudiar un nuevo enfoque.

Bajo el pensamiento: *“debe existir un nivel adicional de comprensión, en el que el carácter de las tareas del procesamiento de la información llevadas a cabo durante la percepción, se analice y comprenda de modo independiente a los mecanismos y estructuras particulares de los que se implementan en nuestro cerebro”*, Horn establece modelos de cálculo que expresan la formación de la imagen a través de ecuaciones diferenciales que relacionan los valores de la intensidad de ésta con la geometría de la superficie.

Posteriormente, David Marr propone la siguiente aproximación; dividir la extracción de la información sobre la forma (3-D) en tres etapas de representación; i) representación de las propiedades de la imagen bidimensional (cambios de intensidad y geometría bidimensional local), denominado como “esbozo primitivo”; ii) representación de las propiedades de las superficies visibles en un sistema de coordenadas centrado en el observador (orientación de la superficie, distancia al observador, descripción primaria de la iluminación), denominado como “el esbozo 2 ½ D”, y; iii) representación centrada en el objeto de la estructura tridimensional y de la organización de la forma observada (descripción de las propiedades de la superficie), denominada “representación de modelo 3-D”.

Sin embargo la comunidad de investigadores de visión artificial no acepta dicha teoría de percepción visual, dado que no se incluye dentro de ella un aspecto fundamental conocido

como los sistemas activos visuales. Todos los sistemas visuales existentes, desde los que corresponden a insectos, ranas, peces, serpientes, pájaros y hasta los de humanos, son sistemas visuales activos, por lo que controlan el proceso de adquisición de las imágenes y por lo tanto introducen restricciones, que facilitan la recuperación de información sobre la escena 3-D.

En la actualidad el término visión activa, se usa tanto para definir las operaciones activas sobre el mundo, con el objetivo de cambiar las imágenes que se están captando para mejorar las posibilidades de éxito de una tarea, como en el sentido de procesos autónomos (serpientes, por ejemplo) que explotan la coherencia de una sucesión de imágenes para realizar un seguimiento en el tiempo, que sea fiable y eficiente.

## 1.2 Visión Artificial

Se puede decir que la *“visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales”*. Las estructuras y propiedades del mundo tridimensional que se quiere deducir en “visión artificial” incluyen no solo propiedades geométricas, sino también propiedades materiales. Ejemplos de propiedades geométricas son la forma, tamaño y localización de los objetos, mientras que las materiales se refieren a color, iluminación, textura y composición. Si el mundo se modifica en el proceso de formación de la imagen, se requerirá también inferir la naturaleza del cambio, e incluso predecir el futuro.

A estas alturas debemos comprender que la formación de imágenes es la aplicación de muchas y diferentes escenas tridimensionales en una imagen bidimensional., lo que hace de hecho que este proceso sea bastante complejo. A la inversa, dada una imagen bidimensional para determinar la escena tridimensional de la que procede, no tiene una única solución. Para romper la ambigüedad hay varias soluciones: i) tomar más imágenes; y ii) hacer más hipótesis sobre el mundo. En este último caso, es importante que cualquier hipótesis que se haga sea mantenible y que se entienda exactamente su papel.

Aún teniendo el problema de inversión de imagen suficientemente restringido de forma tal que casi se tenga una solución única, queda por resolver el hecho de su computabilidad y robustez. Es decir, si la solución es computable utilizando capacidades de procesamiento razonable y si esta admite o “resiste” pequeños errores o cambios en los datos o en el proceso de la propia computación (tratamiento o cálculo).

El “procesamiento de imágenes” es un término genérico para *la manipulación de imágenes con el objeto de producir nuevas imágenes que sean mejores, en algún sentido*. El “procesamiento de imágenes” comprende: i) la mejora de imágenes, que modifica una imagen para mejorar su apariencia o calidad a nivel humano; ii) la restauración de imágenes, que corrige la degradación de imágenes, por ejemplo por emborronamiento por movimiento; y iii) compresión de imágenes, que pretende la representación de imágenes de forma compacta manteniendo un aceptable nivel de calidad. Por otro lado, el “reconocimiento o clasificación de patrones”, clasifica una imagen o el patrón de la misma en un conjunto finito de posibles categorías. Por último, la “Visión Artificial” se dedica más a la generación de descripciones de escenas tridimensionales –que no están, en principio, restringidas a pertenecer a una clase predeterminada- a partir de imágenes bidimensionales.

La visión artificial busca inferir el estado del mundo físico a partir de imágenes ruidosas y ambiguas de mundo. De las diversas fuentes de información en una escena, sólo unas pocas están presentes en la imagen. Es importante poder tener conocimiento sobre los objetos, su conducta y el contexto, en adición a la información proporcionada por las propias imágenes, así como no influir sobre las condiciones de luminosidad, o lo que conoce como captación pasiva de la imagen.

La aproximación a la visión artificial que cuenta con una mayor aceptación, es la que se conoce como “modular”. En comparación con la alternativa que tiene por objetivo realizar una optimización global, en la aproximación modular, el objetivo es aislar las diferentes fuentes de información en una imagen, estas fuentes pueden ser por ejemplo, discontinuidades, sombreado y movimiento. Es obvio, que un sistema modular es más fácil de controlar, monitorizar y mejorar, así como de comprender.

Se debe destacar que la Visión Artificial, tiene varias aplicaciones adicionales a la construcción de robots autónomos, como son la automatización de líneas de ensamblaje, inspección de circuitos para detectar defectos, teledetección para construir mapas de regiones poco accesibles, comunicación mediante gestos con el computador y ayuda a los discapacitados mediante perros guía robots, entre otros.

Un esquema simple de un sistema de visión de propósito general se muestra en la Figura N° 1.1. En este esquema, una escena tridimensional es observada primero por una o varias cámaras para producir una o varias imágenes monocromáticas o multibanda, detectándose luego las discontinuidades en la luminosidad, a las que se denomina bordes. Puede ocurrir

que la segmentación se produzca también en regiones relativamente homogéneas. Estas informaciones se utilizan para restringir las posibles interpretaciones de la escena. La información sobre la escena se restringe en base a características como, mapa de bordes, sombreado, textura y movimiento. A su vez con estas características dan lugar a restricciones locales (orientación, reflectancia de puntos ) o globales (por ejemplo el objeto es de tal tipo).

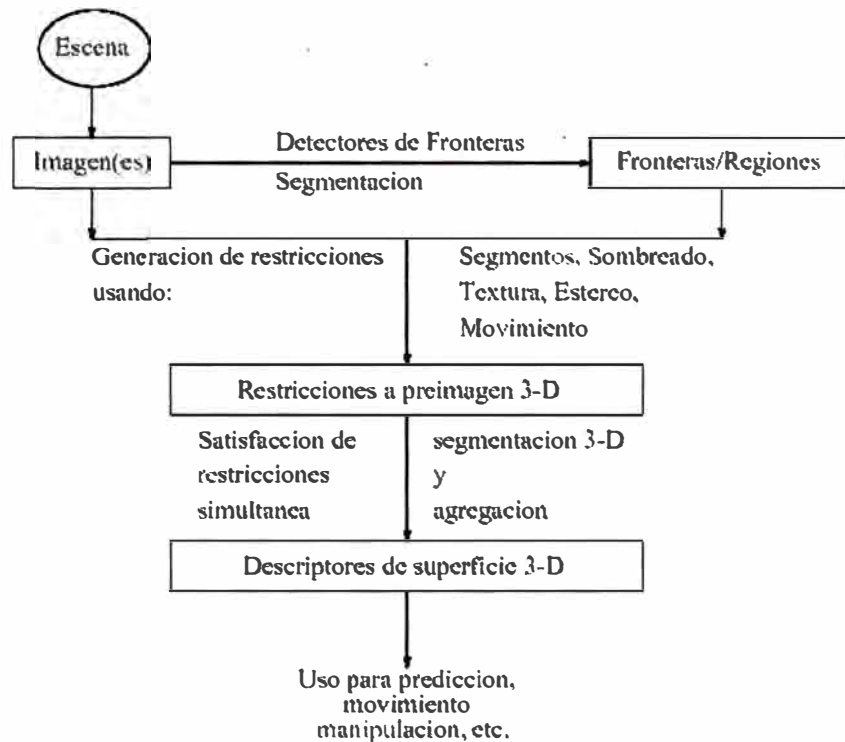
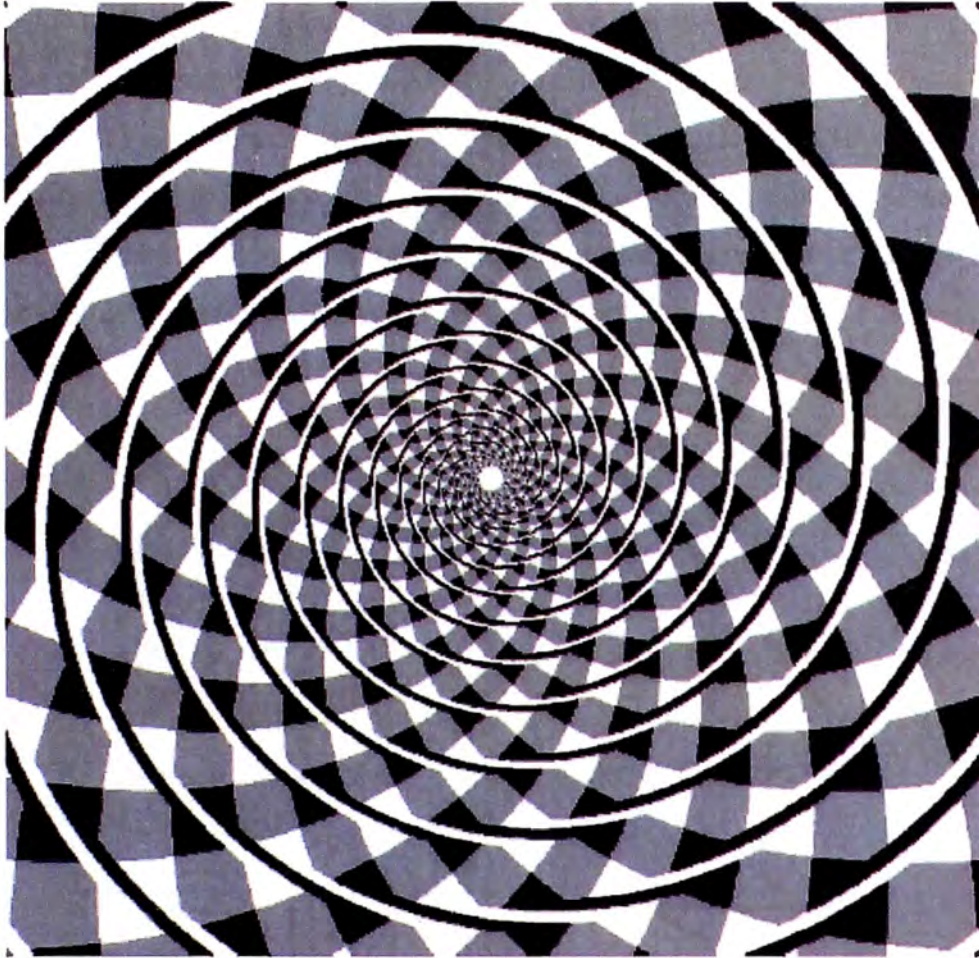


Figura N° 1.1 – Sistema de Visión de Propósito General

### 1.3 Problemas de la Percepción

Una de las preguntas que uno puede hacerse, es por qué los investigadores en visión artificial no construyen simplemente un sistema que emule el sistema visual humano, teniendo en cuenta la enorme cantidad de publicaciones en neurofisiología, psicología y psicofísica. Una buena razón por la que los investigadores en visión artificial no emulan la visión humana, es porque lo que se conoce del sistema de visión humano más allá del propio ojo, es principalmente disjunto, especulativo y escaso. Pero hay más, aunque el sistema de visión humano es adecuado para muchas tareas, es obvio, como veremos a continuación, que ser adecuado no equivale a ser infalible. La no infalibilidad del sistema, es ampliamente demostrada por la existencia de ilusiones visuales. Una de ellas se ilustra con la espiral de Fraser que se muestra en la Figura N° 1.2.





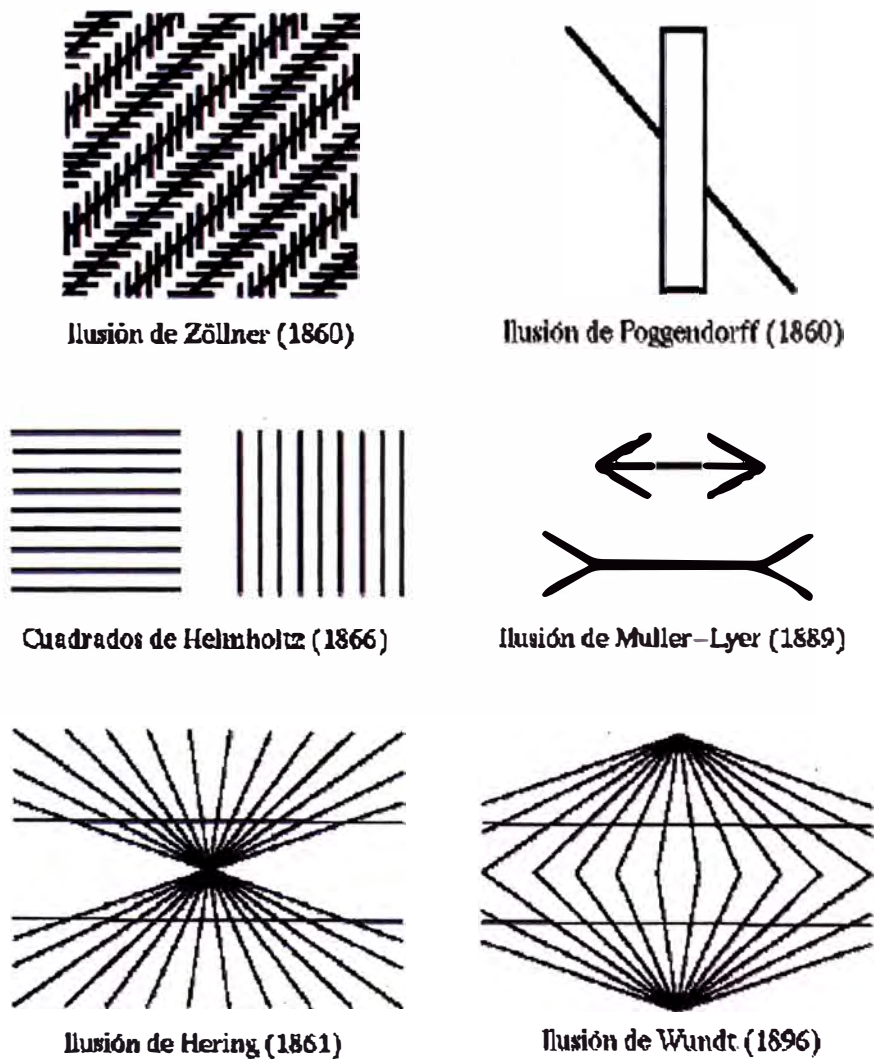
**Figura N° 1.2 – Espiral de Fraser ó Ilusión de Espirales**

Aunque es obvio que nadie puede negar el papel fundamental que juega la percepción en la adquisición de información para los humanos, podría alguien argumentar que “ver” es un acto mecánico y que no genera nada. Todo lo que el acto de ver hace, es inferir el estado del mundo en la medida de lo que es permitido por los datos sentidos. Es decir produce alimento para el pensamiento, para conceptualizar y clasificar -asignar pertenencia a una clase de equivalencia basada en formas o funciones-, por tanto adscribiendo propiedades que no son percibidas y tan solo postuladas.

**Ilusiones Visuales, Ambigüedades e Inconsistencias.** El sistema de visión humano como otras partes de la anatomía del cuerpo humano, es fruto de la evolución y retos a que se ha visto sometido el ser humano en su supervivencia. Como consecuencia de esto, el sistema de visión humano responde a unos patrones mejor que otros, y además, puede auto-engañarse al aplicar pautas de interpretación en situaciones ambiguas, por la existencia de ilusiones visuales, ambigüedades e inconsistencias, y como anteriormente hemos podido presentar, resulta que no es un sistema infalible.

Además de la ilusión en la espiral de Fraser que ya hemos visto, existen otros ejemplos, como los que se muestran en la Figura N° 1.3 – Ilusiones Ópticas.

En la Figura N° 1.3 se presentan seis ilusiones ópticas clásicas. En la Ilusión de Zöllner las diagonales son paralelas pero no lo parecen, en la de Poggendorf, la línea recta no lo parece, en la de Helmholtz los cuadrados parecen rectangulares, en la de Müller-Lyer las líneas horizontales parecen de diferente longitud y en las de Hering y Wundt las líneas rectas parecen curvadas.



**Figura N° 1.3 – Ilusiones Ópticas**

Otro problema es que una figura puede tener más de una posible interpretación, en otras palabras la figura puede ser ambigua. Múltiples interpretaciones pueden coexistir o una puede dominar a la(s) otra(s). Ejemplos de ambigüedades se muestran en la Figura N° 1.4.- Poseidón o Sirena, así como en la Figura N° 1.5.



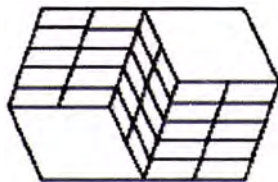
**Figura N° 1.4 – Poseidón o Sirena**



Mujer joven / Anciana



Jarrón / Caras



Hacia dentro / Hacia fuera



Seis cubos / Siete cubos

**Figura N° 1.5 – Ejemplos de Ambigüedad de Imágenes**

Finalmente, es posible que, aunque percibida como una figura que ni es ambigua ni corresponde a una ilusión óptica, la figura sea completamente irrealizable en el sentido de que no podemos construir el objeto 3-D percibido en el espacio real 3-D. Esta posibilidad es ilustrada bellamente por Belvedere, la famosa litografía de Escher que se reproduce en la Figura N° 1.6. Aunque la litografía se percibe al principio como una escena normal, un examen más detallado de la misma revela que la escena percibida es físicamente irrealizable.

Las ilusiones, ambigüedades e inconsistencias presentadas son mucho más que



curiosidades, pues nos llevan a la pregunta importante, en el sentido de que si inferimos de nuestras imágenes en la retina, más de lo que en verdad está soportado por la geometría y física de la formación de imágenes. Helmholtz en su libro *Handbook of Physiological Optics* publicado a mediados del siglo pasado expresó que “*cada imagen es una imagen de algo sólo para aquel que conoce como leerla, y que está capacitado con la ayuda de la imagen a formar una idea de la cosa*”. La implicación de esta afirmación es que el sistema visual humano no estuviera haciendo inferencias precisas y exactas basadas en la física de la formación de imágenes en el ojo, sino que el sistema visual invoca reglas que se obtienen y están sesgadas por la experiencia previa del individuo y tal vez por la especie. Como resultado: “*los humanos podrían ver lo que no hay y no ver lo que hay*”.

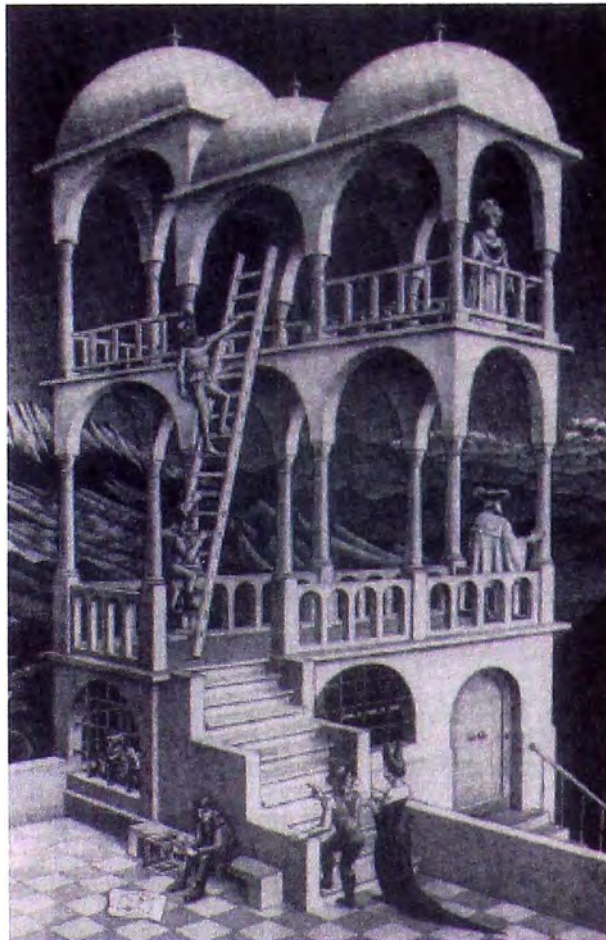


Figura N° 1.6 – Litografía de Escher – Consistencias Geométricas

El cuestionamiento inmediato es si se debe permitir a una máquina hacer lo mismo, y como consecuencia de ello, se tengan interpretaciones ambiguas y erradas con relación a la escena real.

#### 1.4 Métodos de Visión por Computadora

Los trabajos desarrollados en el MIT (Massachusetts Institute of Technology) por el equipo de David Marr, durante los 70s, sobre el sistema de visión humano así como su monografía póstuma *VISION: A "Computational Investigation into Human Representation and Processing of Visual Information"* marcan uno de los hitos más importantes en el desarrollo de una metodología con la que abordar soluciones a los complejos problemas que presentan tanto la visión humana como la visión a través de mecanismos artificiales. Para esto, Marr estableció una metodología modular de tres grandes bloques para el procesamiento de la información:

- Cálculo del esbozo primitivo como estructura que caracteriza los distintos rasgos básicos ( bordes, líneas , arcos, manchas, etc) presentes en la imagen de intensidades (nivel 2D).
- Construcción de las superficies presentes en la imagen desde el punto de vista del observador, a partir de la informaciones suministradas por el esbozo primitivo y las informaciones tridimensionales proporcionadas por las estereoscopia, sombreado, reflectancia, iluminación, etc (nivel 2 ½ D).
- Construcción del modelo que representa a los objetos en el espacio, permite catalogarlos y compararlos con información previamente almacenada (nivel 3D).

Estas ideas sobre la descomposición de la información en niveles 2D, 2½D y 3D, así como sus esquemas de representación de la información en cada uno de estos niveles, han marcado una profunda influencia no tan solo en el campo del estudio de los mecanismos de la visión humana sino también en el estudio y el análisis de las imágenes digitales. Según Nishihara (1978) *"la información acerca de la geometría y la reflectancia de las superficies visibles se codifica en la imagen de varias formas y puede ser decodificada por procesos casi independientes"*.

Este hecho, propició la aparición de otras teorías sobre los procesos de decodificación, muchos de los cuales permanecen hoy en día como áreas activas de investigación. Los más importantes son: la estereoscopia, la derivación de la estructura a partir del movimiento aparente, la selectividad de dirección, el cálculo de la profundidad a partir del flujo óptico, el cálculo de la orientación de la superficie a partir de su textura, el cálculo de la forma a partir de la sombra, la estereoscopia fotométrica, etc.

La influencia de Marr no ha sido la única, también ha habido contribuciones de Tenenbaum & Barrow (1976), al considerar que el proceso de transformación de la

información esta definido por un ciclo en el que intervienen por un lado los datos observados (similar al esbozo primitivo), y de otro lado, un proceso dirigido desde la información ya conocida tratando de ajustar las observaciones a esta información. Otros investigadores como Terzopoulos (1983,1988) y Blake & Zisserman (1987) han hecho aportes sobre regularización y reconstrucción de superficies y modelos de segmentación basados en campos aleatorios.

A la fecha, y en base a los trabajos desarrollados a partir de las teorías de Marr y el resto de investigadores, se ha dado inicio a una nueva especialidad con el nombre de “*Visión Computacional*” que trata de explicar los mecanismos de cálculo de la visión humana, usando para ello procedimientos y métodos de las teorías matemáticas de procesamiento de información.

El término *Visión Artificial* nace pues con la aparición de los ordenadores, los que han permitido su estudio de una forma más profunda y continuada. Por otra parte, la gran simbiosis que ha existido y existe entre los desarrollos en el campo de los modelos de percepción visual y los modelos para el análisis de imágenes digitales, ha originado que hoy en día no exista realmente una línea nítida entre los modelos de la percepción visual y los de la visión artificial.

Como consecuencia de esto, el término *Visión Artificial* dentro del campo de la Inteligencia Artificial puede considerarse como el *conjunto de todas aquellas técnicas y modelos que nos permitan el procesamiento, análisis y explicación de cualquier tipo de información espacial obtenida a través de imágenes digitales*. Por ejemplo, como hacemos para identificar las letras del abecedario independientemente de la familia que estemos usando (courier, helvética, románica, germánica, etc), ó como hacemos para identificar la forma de una nube, que casi todo el mundo sabe identificar pero casi nadie sabe explicar

### **1.5 Antecedentes y Desarrollo del Procesamiento de Imágenes Digitales**

El procesamiento y análisis de imágenes digitales nace en el momento en que se dispone de recursos tecnológicos para captar y manipular grandes cantidades de información espacial en forma de matrices de valores, a diferencia del estudio de los mecanismos de la visión humana. Esta distinción situó al procesamiento y análisis de imágenes digitales como una tecnología asociada a las Ciencias de la Computación y por tanto cabe pensar de ella como una proyección del término *Visión Artificial* dentro del ámbito de la Inteligencia Artificial.

Históricamente la primera vez que se hizo uso de las técnicas de imágenes digitales, fue en los años veinte, en la transmisión de imágenes de fotografías periodísticas a través de cable submarino entre Londres y New York, lo que significó que el tiempo entre emisión y recepción de las imágenes pasase de una semana (por barco) a tres horas (por cable). A partir de esto y durante algún tiempo, se desarrollaron las técnicas de codificación y reproducción de las imágenes logrando pasar en la representación de imágenes, de cinco (5) a quince (25) tonos de gris.

Después de ello, se han dado avances que van desde el hecho de disponer de técnicas para la transmisión y procesamiento de imágenes desde satélite, técnicas de transmisión, realce y restauración de imágenes (primeros satélites espaciales), estudio de imágenes de rayos X, microscopía óptica y electrónica, visión aplicada a robótica bajo el proyecto *Hand-Eye*, geometría integral y el cálculo de probabilidades para la caracterización y medición de formas (Morfología Matemática), hasta observar cierta especialización y catalogación de las técnicas existentes, donde conceptos de *técnicas para el procesamiento de imágenes digitales* como el conjunto de todas aquellas técnicas asociadas a la captura, codificación y representación de las imágenes (no introducen sobre las mismas ningún tipo de interpretación), y *técnicas para el análisis de imágenes digitales, técnicas de visión por computador o visión mediante robot*, representan acepciones que se refieren a aquellas técnicas que tratan de extraer la información presente en la imagen con el fin último de hacer una interpretación de la escena representada por dicha imagen.

Adicionalmente se han hecho contribuciones en el desarrollo de algoritmos para la detección de características (bordes, líneas, texturas, etc.) que ayudan a definir el denominado *esbozo primitivo*, así como en el desarrollo de técnicas globales de segmentación de una imagen en regiones, uso de técnicas de representación del conocimiento para los problemas de interpretación de imágenes, en relación con aplicaciones de tipo industrial, iniciando una fuerte hibridación entre las técnicas de la Inteligencia Artificial para la representación del conocimiento y las técnicas de interpretación de escenas a partir de imágenes digitales, como son la interpretación de la imagen 2D como proyecciones de escenas 3D, técnicas de representación y búsqueda de objetos, el estudio de la orientación a partir de texturas, etc.

Este interés por el estudio de problemas de mayor complejidad ha sido paralelo al desarrollo de nuevas generaciones de ordenadores, los cuales permiten la experimentación con imágenes cada vez más complejas en su interpretación.

El concepto de arquitectura paralela SIMD (Single Instruction Multiple Data) fue sin lugar a duda la más usada en este tipo de aplicaciones. Los conceptos de procesadores con pipeline y procesadores vectoriales fueron también de amplio uso. En cualquier caso cabe destacar que el rápido aumento de las funcionalidades de los computadores estándar, ha desechado el uso de este tipo de arquitecturas en muchas aplicaciones actuales.



## **CAPÍTULO II**

### **FORMACIÓN DE LA IMAGEN**

Con el objeto de entender como los seres vivos ven y asimilan la información que les llega a través de sus sensores, debemos estudiar las tres partes que conforman el proceso: i) las señales que percibimos; ii) el medio en que se transmiten, y; iii) los mecanismos de decodificación del sistema receptor y/o cerebro. Una parte de las técnicas de procesamiento de imágenes están asociadas a problemas en la adquisición de señales, tipo de sensor utilizado y al medio a través del cual se han obtenido.

El ojo humano, como sensor, es un mecanismo demasiado especializado, sin embargo el proceso de formación de la imagen que se lleva a cabo en su interior puede considerarse como genérico para cualquier otro sensor de tipo visual. Es por ello que el estudio del funcionamiento del ojo como sensor mejor adaptado al procesamiento de señales visuales es de gran interés para conocer aspectos básicos de los métodos de captación de imágenes.

#### **2.1 Origen de los Datos**

En el modelo visual humano se asocia el concepto de ver, con el de percibir una señal luminosa con una intensidad mínima y en un rango de frecuencia espectral dado, aunque hoy en día se conocen las posibilidades de obtener imágenes a partir de sensores que trabajen en condiciones muy distintas de iluminación a las que es sensible el ojo humano, p.e. el infrarrojo, rayos X, etc. Queda claro, que la posibilidad de formar imágenes esta asociada al tipo de sensor usado y a las posibilidades de que dicho sensor sea capaz de captar y decodificar la información que le llega. Este punto de vista nos sitúa en una posición más general de lo que sería el simple estudio de las imágenes en color captadas por nuestros ojos.

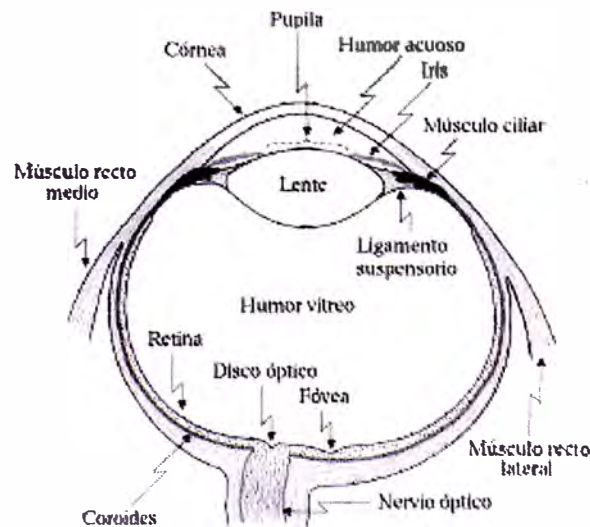
Un tema muy importante es conocer "cuanta información se puede obtener de una escena cuando usamos un determinado sistema sensorial", y si esta información es suficiente para los propósitos de tratamiento que se requiere, o en todo caso, especificar y utilizar otros tipos de sensores, si es que los algoritmos de reconocimiento y /o proceso así lo exigen.

## 2.2 El Ojo como Cámara

Independientemente de lograr emular la visión humana sea en la forma o en su funcionamiento, es importante conocer el sistema de visión humana, lo cual se describe a continuación, de una manera muy genérica.

El ojo humano es aproximadamente una esfera de 20mm de diámetro, formada por un conjunto de membranas denominadas cornea, esclera, coroides y retina, y que puede rotar en su órbita bajo el control de seis músculos extrínsecos. La Figura N° 2.1 muestra una sección del ojo humano. La luz entra a través de la cornea, pasa a través del humor acuoso que rellena la cámara interior, llegando al cristalino y entonces mediante el humor vítreo forma una imagen invertida en la retina. Delante del Cristalino existe una membrana opaca denominada Iris, que es al que da el color al ojo. La luz entra al ojo mediante una apertura circular del Iris que se denomina pupila, cuyo tamaño se controla por la expansión y contracción del Iris.

La acomodación (ajuste de curvatura del cristalinos por medio del músculo filiar) adapta el ojo para obtener la mejor visión a un distancia determinada (en los ojos sanos, los “objetos” se forman en la retina). Cuando los objetos se forman delante o detrás de la retina, aparecen como imágenes borrosas al observador. La miopía es la incapacidad del ojo para enfocar en la retina objetos distantes del ojo, en este caso los “objetos” se forman delante de la retina. La hipermetropía, es la incapacidad de un ojo para enfocar en la retina los objetos próximos al ojo (los “objetos” se forman detrás de la retina).



**Figura N° 2.1 – Ojo Humano**

**La Retina.** La membrana más interna del ojo es la Retina que cubre toda la pared interna del ojo, con un mosaico de fotorreceptores que cuando es estimulada por la luz, produce señales eléctricas nerviosas. Estas señales se transmiten al cerebro a través del nervio óptico, evocando la experiencia de la visión. El lugar de la retina donde se juntan todas las fibras nerviosas individuales que constituyen el nervio óptico, recibe el nombre de disco óptico y es una región que no tiene fotorreceptores y por ello a menudo se le denomina “punto ciego”. Muy cerca del “punto ciego” existe una zona con una alta concentración de fotorreceptores llamada la fovea, con un diámetro menor de un milímetro, la fovea tiene una importancia enorme ya que proporciona la agudeza visual en escenas fuertemente iluminadas.

La retina, con menos de un milímetro de grosor, ha sido objeto de diversas investigaciones. Un punto importante es que las fibras del nervio óptico se colocan cerca del vítreo y sin embargo, los bastones y conos (los dos tipos de receptores que deben su nombre a su forma) están localizados cerca de la coroides, esta disposición es obviamente contra intuitiva ya que obliga a que la luz pase a través de casi toda la profundidad del tejido de la retina (lleno de capilares sanguíneos, células nerviosas y fibras nerviosas) antes de que pueda ser sensada. Sin embargo, en la fovea, al contrario que en el resto de la retina, mucho del tejido retinal entre los fotorreceptores y el humor vítreo está desplazado hacia un lado creando una zona con acceso directo a la luz.

En la Figura N° 2.2 se muestra como la luz llega al retina hasta ser detectado por los fotorreceptores. Los bastones son capaces de detectar luz mucho más débil, es decir facilitan la visión en la oscuridad, mientras que los conos que pueden ser de diferentes variedades (cada variedad con diferente sensibilidad preferente hacia la luz verde, roja o azul) proporcionan la visión en color.

En la Figura N° 2.3 se muestran los caminos visuales del ojo al cerebro, si bien la primera parte referida al proceso ocular es bastante entendida y conocida en la actualidad, aún falta un mayor detalle en lo que respecta a la parte cerebral.

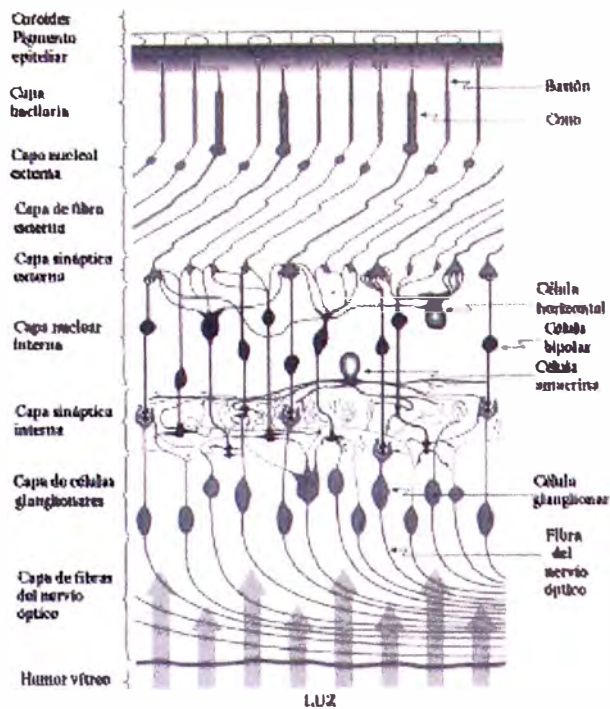


Figura N° 2.2 –La Retina

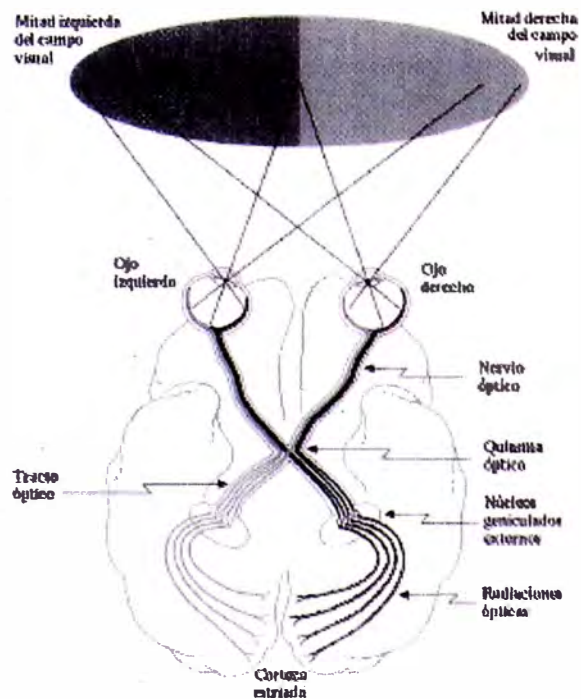


Figura N° 2.3 – Caminos Visuales del Ojo al Cerebro

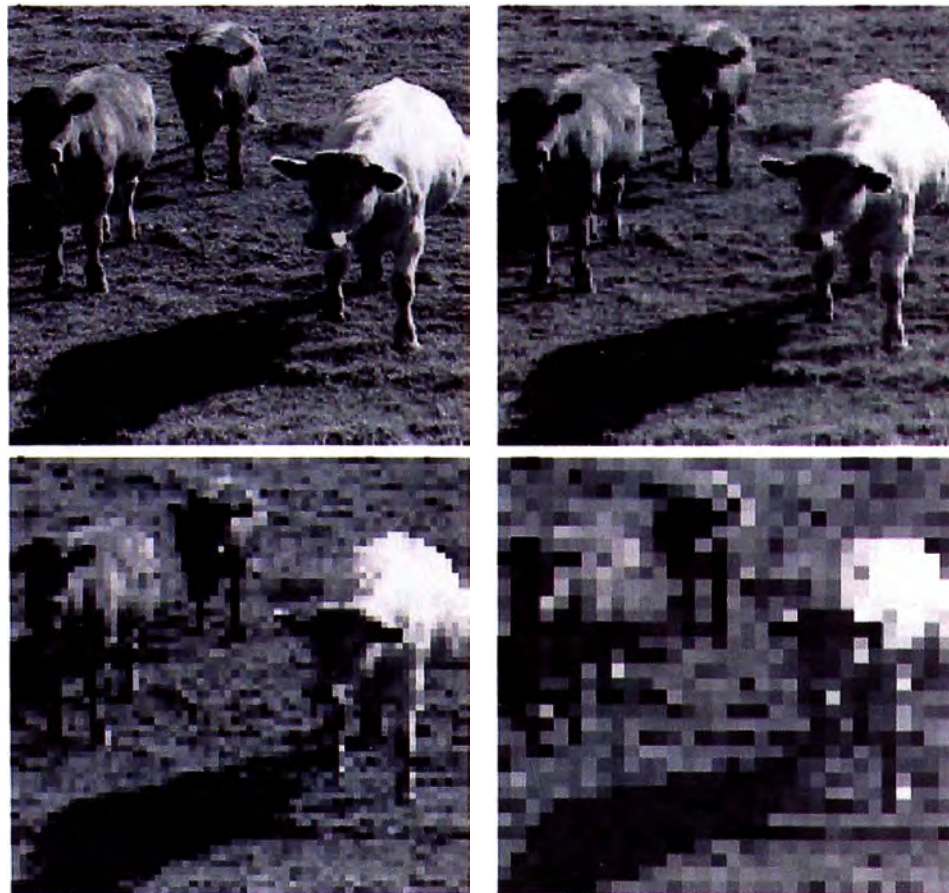
### 2.3 Tipos de Sensores e Imágenes

Hoy en día existen diferentes sensores de tipo opto-electrónico para la adquisición de imágenes en los distintos campos de aplicación científica y técnica. Cada tipo de sensor responde en su diseño a la necesidad de adquirir una determinada información acerca de los objetos que se observan.

### 2.4 Efectos del Muestreo y Cuantificación

Los dos principales causas que producen pérdida de información cuando capturamos una imagen, son la naturaleza discreta de los píxeles de la imagen y el rango limitado de valores de intensidad luminosa que somos capaces de medir en cada píxel.

**Efectos del muestreo.** El muestreo de una imagen tiene el efecto de reducir la resolución espacial de la misma. La Figura N° 2.4 se muestra el efecto de captar una misma imagen a distintas resoluciones.



**Figura N° 2.4 – Muestreo de Imágenes. De izquierda a derecha y de arriba abajo. Muestreo a 256x256, 128x128, 64x64 y 32x32 píxeles.**



Puede observarse la pérdida de información introducida con el aumento del paso de muestreo, así como el ruido que se va introduciendo en forma de patrones rectangulares sobre la imagen.

**Efecto de la cuantificación.** El efecto de cuantificación viene dado por la imposibilidad de tener un rango infinito de valores de medida para la intensidad de brillo de los píxeles. La tecnología actual permite en casos extras llegar hasta rangos de 10-16 bits de información, dependiendo la aplicación, aunque lo general y normal es tener 8 bit, o equivalentemente 256 niveles de gris para codificar este valor lumínico. La Figura N° 2.5 muestra el efecto de representar una imagen con 8 y 1 bits, respectivamente.



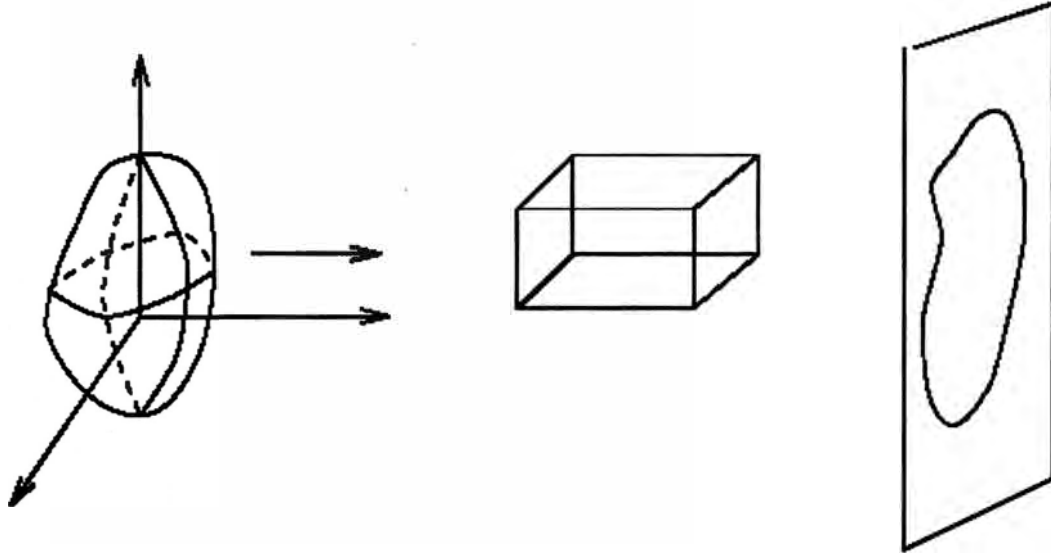
**Figura N° 2.5 – Efecto de la Cuantificación. De izquierda a derecha (8 y 1 bits).**

No existen criterios que nos permitan decidir el número óptimo de píxeles y de bits con los que se debe muestrear una determinada imagen. Sin embargo, se ha llegado a la conclusión de que existen determinados valores de umbral por encima de los cuales no se aprecia una significativa ganancia pero por debajo de ellos si se aprecia una pérdida efectiva en la imagen. En cualquier caso, dado el estado actual de los métodos automáticos de análisis e interpretación de imágenes, el obtener una imagen con la mayor resolución y calidad posible es por ahora lo recomendable.

## **2.5 Formación de la Imagen**

Asumiendo que una escena u objeto tridimensional es convertido a una imagen 2D mediante un sistema de grabación de una cámara tal como se muestra en la Figura N° 2.6.

Cualquier sistema de grabación al no ser perfecto, introduce distorsiones estadísticas y determinísticas. Representando como  $f(i, j)$  a la imagen sin distorsiones y como  $g(i, j)$  a la imagen observada, ruidosa y emborronada



**Figura N° 2.6 – Formación o Captación de Imagen 2D**

Función de Emborronamiento o psf

$$.....u(i, j) = \sum_m \sum_n d(i, j, m, n) f(m, n) \quad (2.1)$$

$$\text{Transformación} \quad v(i, j) = S(u(i, j)) \quad (2.2)$$

$$\text{Proceso de Ruido} \quad g(i, j) = v(i, j) \oplus \varepsilon(i, j) \quad (2.3)$$

Donde  $S(\dots)$  es una función no lineal ,  $d(i, j, m, n)$  es la respuesta del sistema de emborronamiento a un impulso bidimensional en el punto  $(i, j)$ ,  $\varepsilon(i, j)$  es el proceso de ruido y  $\oplus$  representa una operación punto a punto.

**Función de Emborronamiento.** La función o modelo de emborronamiento descrito anteriormente es de uso limitado principalmente por la dificultad para estimar  $d(i, j, m, n)$ , así como al esfuerzo computacional que requieren las técnicas de restauración a las que dan origen. Normalmente el emborronamiento se considera (localmente) como invariante en el espacio, pudiendo representarse como:

$$u(i, j) = \sum_m \sum_n d(i - m, j - n) f(m, n) \quad (2.4)$$

Asumiendo que el proceso de emborronamiento es un proceso continuo y debiendo encontrar  $d(m,n)$  a partir de la versión continua  $d(s,t;\psi)$  donde  $\psi$  es un vector que representa la fuerza del emborronamiento. Si se usa la siguiente aproximación:

$$d(m,n) = \int_{\Omega(m,n)} d(s,t;\psi) ds dt$$

Donde:

$$\Omega(m,n) = (s,t \left| m - \frac{1}{2} \leq s \leq m + \frac{1}{2}, n - \frac{1}{2} \leq t \leq n + \frac{1}{2} \right.) \quad (2.5)$$

Adicionalmente se debe precisar que la función de emborronamiento no puede tomar cualquier valor, ni valores negativos y que la suma de coeficientes sea igual a 1. Los emborronamientos más comunes son por movimiento, por desenfoque y por turbulencia atmosférica.

**Emborronamiento por Movimiento.** Es posible distinguir varios tipos o formas de emborronamiento por movimiento, debido principalmente al movimiento relativo entre el aparato que carga la imagen. Dichos movimientos pueden ser traslaciones, rotaciones, cambios de escala o incluso combinaciones de estos casos. Si el objeto es trasladado a una velocidad constante en un ángulo de  $\phi$  radianes con el eje horizontal durante un intervalo de exposición  $[0,T]$ , entonces la distorsión es unidimensional. Definiendo la longitud del movimiento mediante  $L = VT$ , la psf está dada por  $d(s,t;L,\psi) = \{1/L$  si  $\sqrt{s^2 + t^2} \leq L/2$  y  $t/s = \tan(\psi)$ , en otro caso el valor es cero.

**Emborronamiento por Desenfoque.** Cuando a partir de una escena 3D produce una imagen 2D, algunas partes de la escena están enfocadas, mientras que otras no lo estarán. Si la apertura de la cámara es circular, la imagen de un punto fuente es un disco pequeño conocido como el “círculo de confusión”. El grado de desenfoque (diámetro del centro de confusión) depende de la longitud focal  $F$ , la apertura y la distancia  $P$  entre la cámara y el objeto. La Figura N° 2.7 muestra un ejemplo de emborronamiento por desenfoque.

Si el grado de desenfoques es muy grande, se puede usar una función de emborronamiento uniforme dentro del círculo de confusión. La pfs de este emborronamiento con un radio  $R$  igual al círculo de confusión dividido por dos, viene dado por:



$$d(s,t;R) = \begin{cases} \frac{1}{\pi R^2} \\ 0 \end{cases} \begin{cases} \text{si } \sqrt{s^2 + t^2} \leq R \\ 0 \end{cases} \quad (2.6)$$



**Figura N° 2.7 – Imagen Original y Desenfocada (borrosa)**

**Turbulencia Atmosférica.** No obstante que el emborronamiento introducido por turbulencia atmosférica depende de varios factores como temperatura, velocidad del viento y tiempo de exposición, para exposiciones largas, la función de emborronamiento puede ser aproximada por una función gaussiana:

$$d(s,t;\sigma) = C \exp\left\{-\frac{s^2 + t^2}{2\sigma^2}\right\} \quad (2.7)$$

donde  $\sigma^2$  determina la severidad del emborronamiento ( $C=\text{cte}$ ).

**Emborronamiento de Dispersión.** Las imágenes de Rayos X muestran los detalles mediante la cantidad de radiación que fue absorbida por el objeto que fue irradiado. Lamentablemente, la radiación se dispersa de su camino de incidencia lo que produce una radiación alrededor de un punto. Existiendo muchos factores que influyen en la correspondiente psf, una aproximación válida viene dada por:

$$d(s,t;\beta) = \frac{C}{(1 + (s^2 + t^2)/R^2)^\beta} \quad (2.8)$$

donde  $\beta$  y  $R$  determinan la severidad del emborronamiento y son una función de la distancia entre el detector y el objeto radiado.

## 2.6 Transformación

Los detectores y almacenadores de imágenes tienen de manera general respuestas no lineales, de tal manera que la respuesta de las películas fotográficas, imágenes de escáner y en general de los visualizadores puede aproximarse a una representación  $v(i, j) = \alpha u(i, j)^\beta$ , donde  $\alpha$  y  $\beta$  dependen del equipo utilizado. Para ejemplificar ello, veamos el siguiente ejemplo.

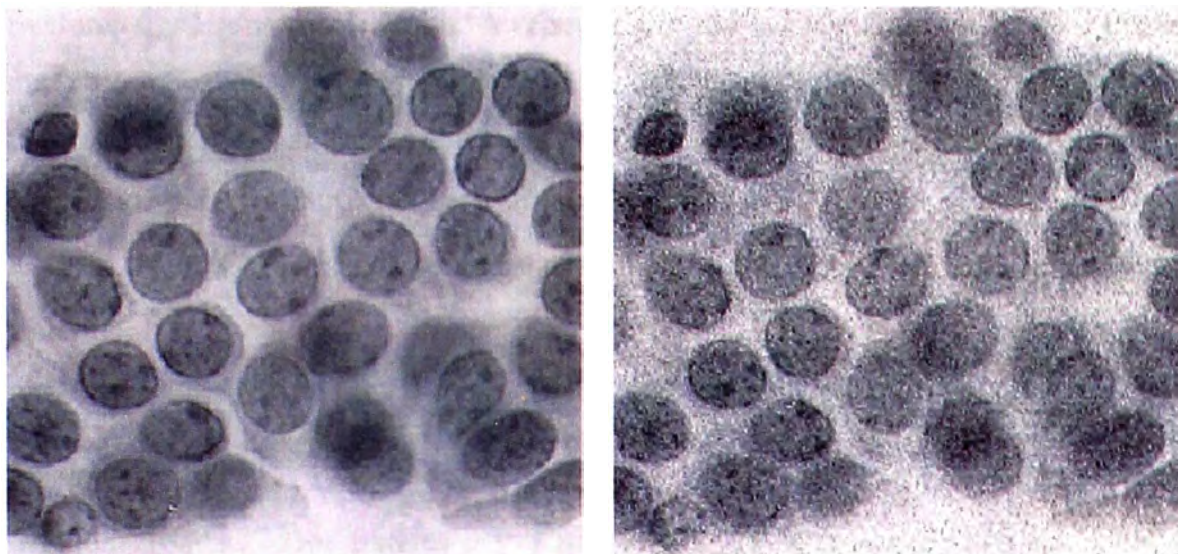
Supongamos en el caso de una señal de video, las componentes de color en el caso del modelo RGB para televisión, aparecen como voltajes como cuyos valores instantáneos son proporcionales a la iluminación que cae en el correspondiente punto de la pantalla (tubo). En consecuencia la señal original de video que llega a un monitor CRT no es apropiada, dado que los tubos de rayos catódicos son “no lineales”, por lo que la función de transferencia de un CRT produce una intensidad  $I$ , proporcional a alguna potencia del voltaje  $V$ , lo cual se puede expresar como:  $I \propto \alpha V^\gamma$ . Donde  $\gamma$  se encuentra entre 2 y 3 dependiendo del tipo de tubo. Son factores gamma típicos 2.2 para el sistema NSTC y 2.8 para PAL/SECAM.

Como solución para compensar el procesamiento no lineal en el destino (monitor), los datos RGB son corregidos antes de la transmisión, mediante la ecuación:  $I_{transmisor} = I_{receptor}^{0.45}$

En muchos casos, para fines prácticos las transformaciones no lineales no son consideradas.

## 2.7 Modelos de Ruido

El ruido expresado como un modelo general no es sino la suma de una señal no deseada sobre la señal limpia (por decirlo de alguna manera) y que matemáticamente se expresa como  $\varepsilon(i, j) = \sqrt{v(i, j)}u_1(i, j) + u_2(i, j)$  donde  $u_1$  y  $u_2$  son ruidos blancos gaussianos diferente de media cero. En varios casos, la fórmula anterior es una aproximación a la expresión  $g(i, j) = \rho(v(i, j) + u_2(i, j))$  donde  $\rho$  corresponde a la distribución de Poisson. La Figura N° 2.8, muestra una imagen ruidosa con efectos de ruido gaussiano aditivo.



**Figura N° 2.8 – Imagen Original y con Ruido Gaussiano**

## 2.8 Relaciones entre Píxeles

Para un mejor entendimiento del procesamiento o tratamiento de imágenes es necesario conocer las relaciones con respecto los píxeles en las imágenes digitales. Normalmente la notación utilizada es a través de letras minúsculas como  $p$  y  $q$ , adicionalmente conociendo que  $f(x, y)$  o  $g(x, y)$  representa una imagen, al conjunto de píxeles de  $f(x, y)$  se denota como  $S$ .

**Vecinos de un píxel.** Un píxel  $p$  de coordenadas  $(x, y)$  tiene cuatro vecinos horizontales y verticales, cuyas coordenadas son:  $(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$ . Este conjunto de píxeles de nombre “4-vecinos de  $p$ ” se nombra como  $N_4(p)$ , donde cada píxel está a una distancia unitaria de  $(x, y)$ .

Los píxeles diagonales vecinos de  $p$ , tienen coordenadas:  $(x + 1, y + 1), (x - 1, y - 1), (x - 1, y + 1), (x + 1, y - 1)$ . En este caso se definen como  $N_D(p)$ . En general a todos los píxeles vecinos, se conocen como “8 vecinos de  $p$ ” y se define o denomina como  $N_8(p)$ .

**Conectividad.** El concepto de conectividad entre píxeles es importante para delimitar o establecer las fronteras de objetos, así como las regiones que conforman una imagen. El establecimiento de si dos píxeles están conectados, se debe establecer si son adyacentes en algún sentido (por ejemplo si son “4 vecinos” y si sus niveles de gris cumplen algún criterio de similaridad –iguales por ejemplo-). Puede darse el caso que una imagen binaria

con valores 0 y 1, dos píxeles sean “4 vecinos” y no estar conectados, salvo que tengan el mismo valor.

Si tenemos a  $V$  como el conjunto de valores de niveles de grises que se usa para definir conectividad, por ejemplo en el caso de imágenes binarias,  $V = \{1\}$  o un rango tal como  $V = \{33,34,\dots,36,\dots,50\}$  para el caso de tonos o niveles de grises, se consideran tres tipos de conectividad.

- “4-conectividad”. Dos píxeles  $p$  y  $q$  con valores en  $V$  se dicen “4-conectados” si  $q$  pertenece a  $N_4(p)$ .
- “8-conectividad”. Dos píxeles  $p$  y  $q$  con valores en  $V$  se dicen “8-conectados” si  $q$  pertenece a  $N_8(p)$ .
- “m-conectividad”. Dos píxeles  $p$  y  $q$  con valores en  $V$  se dicen “m-conectados” si  $q \in N_4(p)$  o  $q \in N_D(p)$  y  $N_4(p) \cap N_4(q)$  es vacío. La “m-conectividad” se introduce para eliminar la ambigüedad en los posibles caminos de unen dos píxeles.

Por lo tanto ahora se puede decir que dos píxeles son adyacentes, si están conectados y en consecuencia dos subconjuntos de  $S$ , como  $S_1$  y  $S_2$  son adyacentes, si existe un píxel en  $S_1$  adyacente a alguno en  $S_2$ .

Un camino del píxel  $p$  con coordenadas  $(x, y)$  a un píxel  $q$  con coordenadas  $(s, t)$  es una sucesión distinta de píxeles con coordenadas  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$  donde  $(x_0, y_0) = (x, y)$  y  $(x_n, y_n) = (s, t)$  siendo  $(x_i, y_i)$  adyacente a  $(x_{i-1}, y_{i-1})$  para  $1 \leq i \leq n$ , donde  $n$  es la longitud del camino. El tipo de camino depende del tipo de adyacencia utilizada.

**Medidas de Distancia.** Para los píxeles  $p, q$  y  $z$  con coordenadas  $(x, y), (s, t), (u, v)$  respectivamente, entonces  $D$  es una función de distancia o métrica, si cumple con:

- $D(p, q) \geq 0, \dots, D(p, q) = 0 \dots si \dots p = q$
- $D(p, q) = D(q, p), y$
- $D(p, z) \leq D(p, q) + D(q, z)$  La distancia entre  $p$  y  $q$  se define como  $D_c(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$ . para esta medida de distancia, los píxeles a distancia menor o igual a “ $r$ ” dado, definen un disco de radio “ $r$ ” centrado en dicho píxel.

La distancia  $D_4$  (denominada de bloques –por la estructura que genera-) se define mediante  $D_4(p, q) = |x - s| + |y - t|$ . Por ejemplo, los píxeles a distancia menor o igual a 2 de un píxel dado serán los siguientes:

$$\begin{array}{ccccc} & & 2 & & \\ & & 2 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ & & 2 & 1 & 2 \\ & & 2 & & \end{array}$$

La distancia  $D_8$  (denominada de ajedrez) entre  $p$  y  $q$ , se define como  $D_8(p, q) = \max(|x - s|, |y - t|)$ , por ejemplo los píxeles con distancia  $D_8$  menor o igual a 2 son los que se muestran:

$$\begin{array}{ccccc} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{array}$$

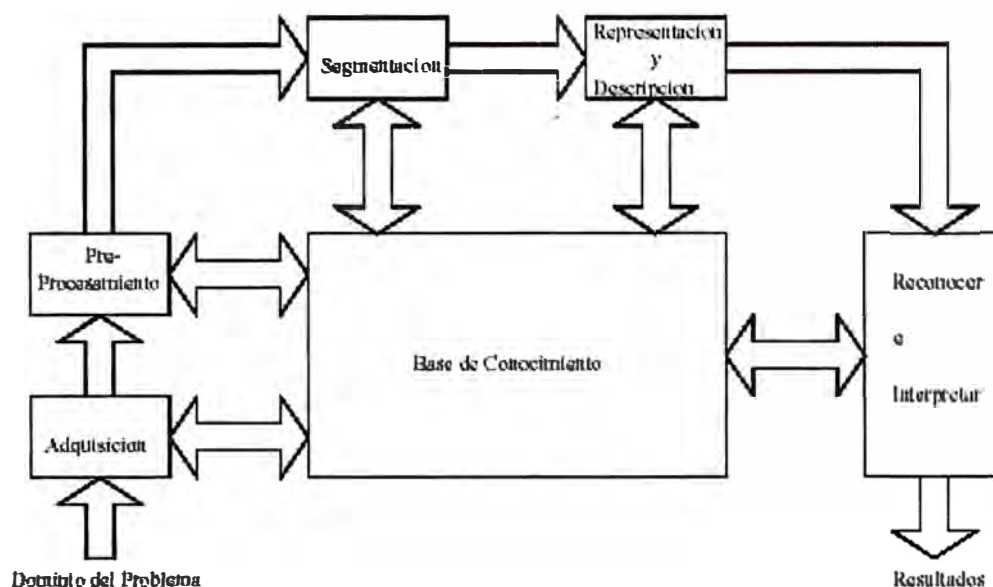
**Operaciones Aritmético / Lógicas.** Las operaciones de carácter aritmético y lógico entre píxeles son muy frecuentes. Las operaciones aritméticas están referidas a la suma ( $p+q$ ), diferencia ( $p-q$ ), multiplicación ( $p*q$  ó  $pq$  ó  $pxq$ ) y división ( $p/p$ ), mientras que las operaciones lógicas son: AND ( $p$  AND  $q$ ), OR ( $p$  OR  $q$ ) y COMPLEMENTO (NOT  $p$ ). En este punto es importante precisar que las operaciones aritméticas se realizan en el caso de imágenes multivaluadas (cuyos píxeles pueden tener diferentes calores), y las lógicas son referidas a imágenes bivaluadas (0 ó 1).

## 2.9 Pasos Fundamentales en el Procesamiento de Imágenes

El tema de Procesamiento de Imágenes tiene asociada un gran cantidad de conceptos propios, algunos de los cuales los hemos podido describir y conocer en los párrafos anteriores, sin embargo hay otros que no han sido tocados y que en algunos casos son de aplicación y utilización en determinados campos, como son en el caso de procesamiento de señales biomédicas. Adicionalmente, existen consideraciones y conceptos relacionados al software, hardware entre otros, que deben ser tomados en cuenta.

Como una forma de entender y conocer el procedimiento para Procesamiento de Imágenes, utilizaremos el problema de reconocimiento de las direcciones de un sobre de correos.

En primer lugar tratemos de esquematizar el proceso, para ello nos ayudaremos de la Figura N° 2.9, teniendo presente que el objetivo final es obtener la dirección establecida en la carta. Este esquema es sin embargo aplicable a cualquier procesamiento de imágenes en general.



**Figura N° 2.9 – Esquema de Procesamiento de Imágenes**

Como primer paso del proceso, se debe de “adquirir” u obtener la imagen digital, lo cual se puede hacer por medio de sensores y posteriormente contar con la capacidad de digitalizar la señal producida por el sensor. Evidentemente, hoy en día contamos con una variedad de sensores, como que podría ser una cámara a color o monocromática, que produce imágenes completas del dominios del problema cada 1/30 de segundo. También podría ser otro dispositivo como un escáner que produce una línea en cada momento, sólo con la diferencia que en este último caso la imagen bidimensional se produce por el desplazamiento del objeto (o del escáner).

Si la salida de la cámara no esta en formato digital se requeriría utilizar un conversor Analógico/Digital. En general, el sensor y la imagen que produce el mismo, viene dada en muchos casos por la propia aplicación sobre la que se actúa.

Luego que se cuenta con la imagen digitalizada, el siguiente paso es un preprocesamiento, con el objetivo principal de mejorar dicha imagen de acuerdo con el objetivo, es decir de forma tal que el objetivo pueda ser cumplido con éxito. Para el caso de nuestro ejemplo

(reconocimiento de la dirección del sobre de correo), normalmente, lo que se persigue es la utilización de técnicas que permitan mejorar el contraste, suprimir o reducir el ruido existente y aislar regiones que por su naturaleza pueden contener información alfanumérica.

El siguiente paso es la segmentación, que tiene por objeto dividir la imagen en las partes que la constituyen o los objetos que la conforman. En general, la segmentación autónoma es uno de los problemas más críticos en el procesamiento de imágenes. Por un lado, una buena segmentación nos facilitará mucho la solución de problema planteado, de otro lado, una mala segmentación o una segmentación errónea nos llevara con mayor seguridad a un hecho fallido. Para el caso del ejemplo propuesto, se debe segmentar con el propósito de lograr el reconocimiento de caracteres de manera individual y que posteriormente representen las palabras correspondientes.

Los resultados de un proceso de segmentación es una imagen de datos que pueden contener la frontera de la región o los puntos de la misma. El siguiente paso, es convertir los datos a una forma que sea apropiada para el computador, siendo para ello importante definir si se utilizará la representación por frontera o por región completa.

La representación por frontera es recomendable cuando el objetivo se centra en las características de la forma externa, como contorno, esquinas, concavidades o convexidades. De otro lado, la representación por región completa es utilizada si el interés esta dado en las propiedades internas como la textura. En muchas aplicaciones, ambas representaciones son necesarias o pueden coexistir, tal como es el caso del ejemplo planteado, que requiere reconocer el contorno como en la textura de la imagen a procesar.

Adicionalmente, debemos tener presente que la representación es sólo una primera parte de la transformación de los datos de entrada, pues la otra parte es la especificación de un método que permita extraer los datos de interés, a este proceso se llama también como “descripción” o “selección de rasgos” y se dedica a obtener o extraer “rasgos” que producen alguna información cuantitativa que es básica para diferenciar una clase de objetos de otra. En el ejemplo que nos sirve de guía, los descriptores serían las concavidades y agujeros que permiten diferenciar una letra de otra o de algún otro símbolo. El último paso, es el reconocimiento y la interpretación. Mientras que el reconocimiento es el proceso que asigna una etiqueta a un objeto en base a la información de los descriptores, la interpretación permite asignar un significado al conjunto de objetos reconocidos. En el ejemplo que venimos comentando, para lograr identificar el carácter “x” se requiere

asociar los descriptores para dicho carácter con la etiqueta “x”, luego de ello la interpretación permite dar un significado a un conjunto de ellos, por ejemplo “x22” podría ser un código postal.

Ahora queda por comentar el tema de la Base del Conocimiento, el que participa en cada actividad del proceso, en ciertos casos sobre la Base del Conocimiento se harán las realimentaciones para cada actividad, o por el grado de entendimiento del problema se podrá seleccionar las áreas de la imagen sobre las cuales se ha de actuar. Con un mayor grado de conocimiento de los procesos en cada aplicación, podrán gestionarse diversas formas o procedimientos para llevar a cabo de manera más eficiente cada paso de los necesarios en el Procesamiento de Imágenes, teniendo presente que no todas las aplicaciones o problemas tendrán que aplicar con la misma rigurosidad cada paso.



## **CAPÍTULO III**

### **TRANSFORMADA DE FOURIER**

#### **3.1 Concepto y Representación de Señales Continuas**

Por señal continua se entiende una función continua de una o varias dimensiones. Existen diferentes tipos de señales en los diversos aparatos o equipos de medición, asociados al estudio de la física, química, biología, medicina, etc., como son los distintos tipos de electrogramas utilizados en medicina, los cuales son señales unidimensionales, ya que se representan por una o varias curvas en función del tiempo. Sin embargo, en el caso de las radiografías son señales bidimensionales y los resultados de la tomografía axial computarizada o la resonancia magnética son señales tridimensionales.

Mediante la utilización del lenguaje matemático se puede afirmar que toda señal es una función matemática que toma un valor en cada punto del espacio en el que esta definida, por consecuencia, los resultados matemáticos sobre aproximación de funciones, permiten establecer que cualquier función continua y periódica definida sobre una región finita del espacio puede ser aproximada por una suma infinita de términos, en donde cada término tiene una contribución a la formación de la señal que es independiente y ortogonal a cualquier otro término del desarrollo.

Las posibilidades a la hora de construir este tipo de aproximaciones pueden ser muy diferentes, pero por distintos motivos la más usada ha sido aquella en que los términos tienen una significación como señales ondulatorias puras. Es decir, cada término es la contribución de una determinada frecuencia/longitud de onda a la formación de la señal. Por lo tanto, toda señal puede ser analizada desde dos perspectivas, como una función continua sobre un espacio de valores de medida o como una función definida sobre un espacio de frecuencias.

#### **3.2 La Imagen como Señal**

Una imagen digital puede establecerse como el resultado de efectuar sobre una señal continua un proceso de muestreo en dos direcciones perpendiculares, con un paso o

distancia entre muestras que se fija para cada dirección.

Una manera bastante intuitiva de entender este efecto, es el considerar a la imagen digital como el resultado de superponer una malla rectangular sobre la función y considerar que los valores de los píxeles de la imagen son una función de los valores de la señal en el interior de cada rectángulo definido por la malla. Funciones típicas del caso bidimensional, son aquellas que asignan a cada píxel la media de los valores de la señal en el interior de cada rectángulo o aquellas que asignan como valor representativo de cada rectángulo el valor en su centro. Con cualesquiera de los criterios anteriores es evidente que se producirá una pérdida de información ya que resumimos en un punto el comportamiento de la señal en una región del espacio.

### 3.3 La Transformada de Fourier

La transformada Fourier de una señal unidimensional o función continua  $f(x)$  es una transformación de dicha señal que nos permite calcular la contribución de cada valor de frecuencia a la formación de la señal. La expresión matemática de dicho cálculo es:

$$F(u) = \int_{-\infty}^{\infty} f(x) \exp[-i2\pi ux] dx \quad (3.1)$$

donde  $i = \sqrt{-1}$ , y la variable  $u$  que aparece en la función  $F(u)$  representa a las frecuencias. Puede demostrarse además que esta transformación tiene una función inversa, es decir que dada la función  $F(u)$  podemos a partir de ella calcular la función  $f(x)$ . La expresión matemática de dicha transformada inversa es

$$f(x) = \int_{-\infty}^{\infty} F(u) \exp[i2\pi ux] du \quad (3.2)$$

Estas dos funciones  $f(x)$  y  $F(u)$  se denominan “par de transformadas de Fourier”. En general las funciones con las que se que tratan problemas reales, deben verificar las condiciones que son necesarias establecer o asumir para que las expresiones anteriores puedan calcularse.

Es importante señalar que aunque las funciones que definen a las imágenes son funciones reales, sus transformadas de Fourier son funciones complejas con parte real y parte imaginaria. Así pues  $F(u)$  se expresará de forma general como  $F(u) = R(u) + iI(u)$ , donde  $R(u)$  denota la parte real y  $I(u)$  la parte imaginaria. Como todo número complejo para

cada valor de  $\mathbf{u}$ ,  $F(\mathbf{u})$  puede expresarse en términos de su módulo y de su ángulo de fase. Es decir,  $F(\mathbf{u})$  también puede expresarse como  $F(\mathbf{u}) = |F(\mathbf{u})| \exp[i\phi(\mathbf{u})]$ , donde:

$$|F(\mathbf{u})| = \sqrt{R^2(\mathbf{u}) + I^2(\mathbf{u})}, \text{ y } \phi(\mathbf{u}) = \tan^{-1} \frac{I(\mathbf{u})}{R(\mathbf{u})} \quad (3.3)$$

A la función  $|F(\mathbf{u})|$  se le denomina espectro de Fourier de la señal  $f(x)$ , y al cuadrado de dicha función,  $|F(\mathbf{u})|^2$  se le denomina espectro de potencias de  $f(x)$ .

**Transformada de Fourier en Imágenes bidimensionales (2D).** En el caso de señales bidimensionales  $f(x, y)$ , los conceptos del caso unidimensional se generalizan de forma directa. Así el par de transformadas de Fourier en notación matemática se expresan como:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-2\pi i(ux + vy)] dx dy \quad (3.4)$$

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[2\pi i(ux + vy)] du dv \quad (3.5)$$

donde  $\mathbf{u}$  y  $\mathbf{v}$  son variables de frecuencia

### 3.4 La Transformada de Fourier de una Imagen Digital

En este punto veremos las consideraciones necesarias para poder utilizar la transformada de Fourier sobre una imagen digital en lugar de hacerlo sobre una función continua. En primer lugar se debe tener en cuenta la relación existente entre una función continua y una imagen digital. Como ya se ha mencionado anteriormente, una imagen digital se obtiene a partir de una señal continua mediante un proceso de muestreo, por lo tanto, habrá que conocer el efecto que este proceso de muestreo tiene sobre el cálculo de la transformada de Fourier.

**Caso Unidimensional.** Supongamos que una señal continua  $f(x)$  es muestreada tomando  $N$  muestras de la misma a una distancia  $\Delta x$  a partir de un valor  $x_0$ . La sucesión de valores que obtendremos serán:

$$f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + (N - 1)\Delta x)$$

Con objeto de poder seguir usando la notación  $f(x)$  tanto para el caso continuo como el discreto, adoptaremos la notación  $f(x) = f(x_0 + x\Delta x)$  donde  $\mathbf{x}$  tomará los valores  $0, 1, 2, \dots, N - 1$ . Dado que el punto de comienzo del muestreo no puede ser cualquiera a

partir de ahora tan solo consideraremos el estudio de cualquier sucesión  $f(0), f(1), \dots, f(N-1)$  de valores de la función  $f(x)$ . La Figura N° 3.1 muestra una representación gráfica del proceso de muestreo descrito.

De acuerdo con la notación utilizada, el par de Transformadas de Fourier Discretas (TFD) asociadas a una sucesión finita de valores vienen dadas por las siguientes expresiones;

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-2i\pi ux/N] \quad (3.6)$$

para  $u = 0, 1, \dots, N-1$ .

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[2i\pi ux/N] \quad (3.7)$$

para  $x = 0, 1, \dots, N-1$ .

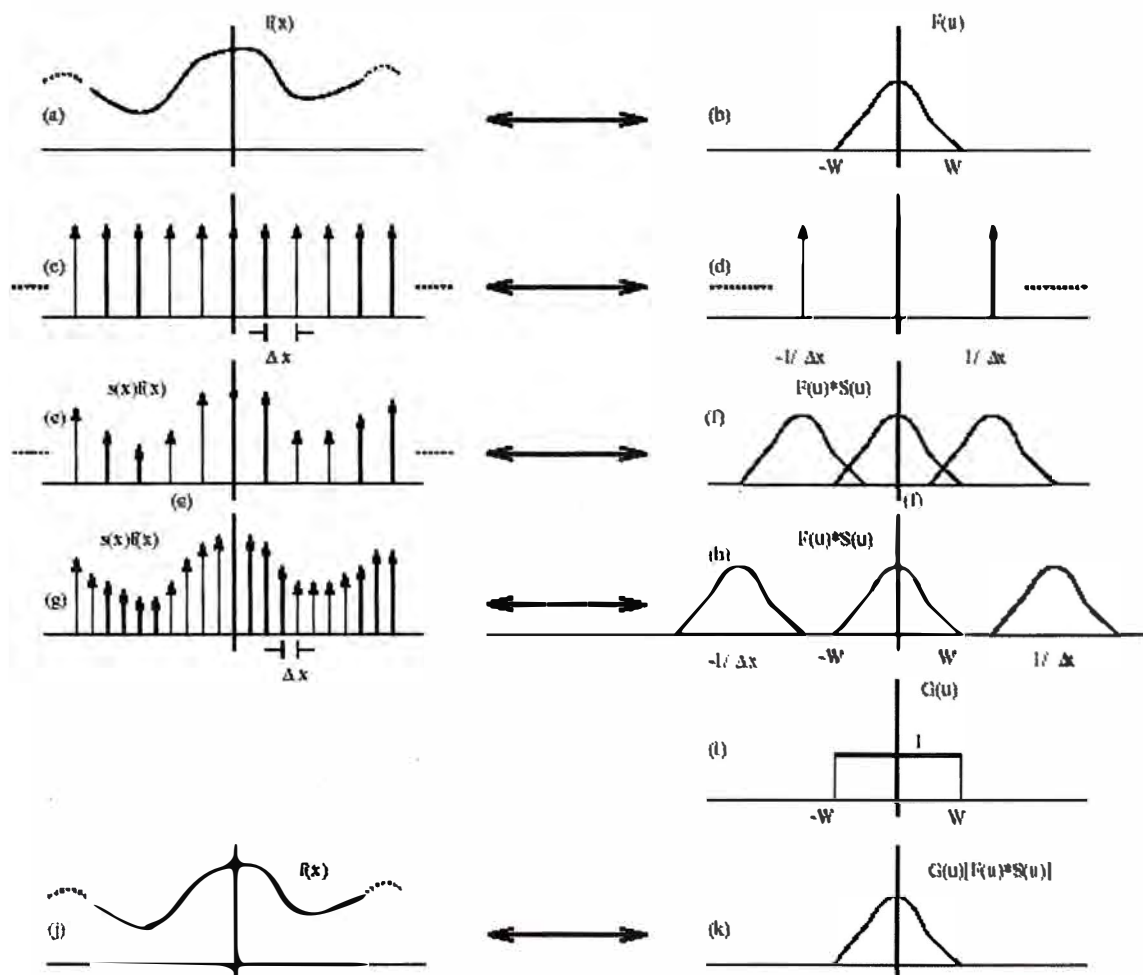


Figura N° 3.1 - Muestreo

Donde los valores de  $u$  en la función  $F(*)$  corresponden a muestras de dicha función en los puntos  $0, \Delta u, 2\Delta u, \dots, (N-1)\Delta u$ , es decir  $F(u) = F(u\Delta u)$ , siendo distinto en el caso de la

función  $f(x)$ , donde el origen de las muestras puede ser cualquier valor  $x_0$ , en el caso de la función de frecuencias el origen siempre estar en el origen del eje de frecuencias. Una explicación intuitiva a este hecho puede ser obtenida si tenemos en cuenta la propiedad, ya antes comentada, de que la aportación de cada frecuencia, en la transformada de Fourier, a la formación de la imagen es ortogonal a la contribución de las demás frecuencias. Entonces podemos mirar los valores de  $|F(u)|$  como si fueran las coordenadas de un vector en un espacio euclídeo de dimensión  $N$ . Para definir un vector, lo único necesario es dar todas sus coordenadas, y por tanto el concepto de origen no tiene significado alguno. Puede demostrarse que la relación que liga los valores de los incrementos  $\Delta x$  y  $\Delta u$  esta dada por:

$$\Delta u = \frac{1}{N\Delta x} \quad (3.8)$$

Esta relación nos indica que el incremento en frecuencias es inversamente proporcional al número de muestras tomadas de la función  $f(x)$  y del paso utilizado para muestrear dicha función. Así pues, cuando se requiere conocer cual es la contribución de un determinado rango de frecuencias a la formación de una señal, es necesario asegurarse que el muestreo realizado sobre la señal original, permita tener alguna muestra dentro del rango de interés.

**Caso Bidimensional.** Para funciones de dos variables, el par de transformadas de Fourier discretas, vendrán dadas por las siguientes expresiones:

$$F(u, v) = \frac{1}{NM} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-2i\pi(ux/M + vy/N)] \quad (3.9)$$

para  $u = 0, 1, \dots, M-1$ ;  $v = 0, 1, \dots, N-1$ , y

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp[2i\pi(ux/M + vy/N)] \quad (3.10)$$

para  $x = 0, 1, \dots, M-1$  y  $y = 0, 1, \dots, N-1$ .

Las expresiones anteriores consideran el uso de un muestreo con distinto paso en la dirección de  $x, \Delta x$ , y en la dirección de  $y, \Delta y$ . Asimismo, el número de muestras tomadas según la dirección de  $x$  es  $M$ , y el número siguiendo la dirección de  $y$  es  $N$ . Similares consideraciones son de aplicación a la función  $F(u, v)$ .

Ahora las relaciones entre los incrementos de muestreo en el dominios del espacio y las

frecuencia serán

$$\Delta u = \frac{1}{M\Delta x} \text{ y } \Delta v = \frac{1}{N\Delta y} \quad (3.11)$$

Cuando  $M = N$ , algunas de las expresiones anteriores pueden resultar se más simple. En particular el par de transformadas de Fourier tendrían las siguientes expresiones,

$$F(u, v) = \frac{1}{N^2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-2i\pi(ux + vy)/N] \quad (3.12)$$

para  $u = 0, 1, \dots, M-1$ ; y  $v = 0, 1, \dots, N-1$ .

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp[2i\pi(ux + vy)/N] \quad (3.13)$$

para  $x = 0, 1, \dots, M-1$  y  $y = 0, 1, \dots, N-1$ .

## CAPÍTULO IV

### OBTENCIÓN DE GRISES Y PSEUDOCOLOR

#### 4.1 Conversión a Escala de Grises

En esta parte se estudiará la conversión de una imagen en color a escala de grises, sin embargo para una mejor comprensión del tema es necesario que previamente se aclaren algunos conceptos referidos a la imágenes en color, sus características principales y la capacidad del el ojo humano en reconocer los diferentes tipos de imágenes.

En primer lugar, se debe precisar que la formación de imágenes tiene mucho que ver con el tipo de luz acromática o cromática, bajo la cual se establece la imagen. La luz acromática (sin color), tiene como único atributo la intensidad o cantidad de luz, por lo que el nivel de gris es la medida escalar de la intensidad de una imagen acromática, que varia del negro al blanco, pasando por toda la escala de grises.

En el caso de la luz cromática, se tienen otras características propias como son la Radiancia -cantidad total de energía que sale de la fuente luminosa-, la Luminancia -cantidad de energía que percibe un observador procedente de la fuente luminosa-, como ejemplo podemos citar una fuente infrarroja que tiene alta radiancia y muy baja luminancia. Adicionalmente otra característica es el brillo, que es un descriptor subjetivo.

En consecuencia la obtención de una imagen en niveles de grises es equivalente a la determinación de la luminancia de la imagen. Como se sabe el ojo humano, percibe distintas intensidades de luz en función del color que se observe, esto es debido a la respuesta del ojo al espectro visible, por esa razón el cálculo del equivalente blanco y negro (escala de grises o luminancia) de la imagen debe realizarse como una media ponderada de las distintas componentes de color de cada píxel.

Una ecuación de la luminancia que permite calcular el nivel de gris, se muestra en la siguiente expresión matemática, con los factores de ponderación de cada componente de color según la sensibilidad del ojo humano a las frecuencias del espectro cercanas al rojo, verde y azul.

$$Y = R*0.3+G*0.59+B*0.11 \quad (4.1)$$

De la aplicación de la fórmula anterior a cada píxel de una imagen ‘true-color’ o color verdadero, se determina una nueva matriz que daría la información de luminancia o tonos de gris. En el caso de convertir a escala de grises una imagen de paleta, bastaría con convertir cada color de la paleta a su gris correspondiente según la ecuación anterior, creando una nueva paleta de grises.

En matlab se cuenta con la función `rgb2gray (I)`, que permite la obtención de tonos o niveles de gris de una imagen a color. En el Capítulo V se muestra en detalle la aplicación de esta función de matlab, para cuatro imágenes a color diferentes, así como los resultados obtenidos en su procesamiento.

Un algoritmo muy utilizado para la determinación de los niveles de gris es la de promediar cada una de las tres matrices que define a una imagen a color, de acuerdo con la siguiente ecuación:

$$I(x,y) = (R(x,y) + G(x,y) + B(x,y)) / 3 \quad (4.2)$$

Adicionalmente, la formula general para la determinación de la intensidad se puede logra a partir de la aplicación de la siguiente fórmula, donde los factores de ponderación dependerán de cada tipo de proceso o aplicación en particular.

$$I(x,y) = k_1 R(x,y) + k_2 G(x,y) + k_3 B(x,y) \quad (4.3)$$

## 4.2 Pseudo-color y Falso Color

Como es de conocimiento general, el ojo humano no es capaz de distinguir entre niveles de gris próximos, pudiendo discernir solo entre 2 a 3 docenas de niveles de gris. En cambio, si somos capaces de distinguir entre un gran numero de colores, que se estima pueden llegar a ser miles de colores y tonos diferentes. Para diferenciar objetos muy próximos en una imagen o para describir o resaltar ciertas características de un objeto es muy útil emplear colores, de donde se desprende que la utilización del color como descriptor de las características de una imagen es parte importante en el procesamiento digital de imágenes.

El Pseudo-color (también denominado falso color) consiste en asignar colores a valores de gris de una imagen monocromática siguiendo algún criterio definido. La calificación de *pseudo* o *falso* color se utiliza para diferenciar el proceso de asignar colores a un imagen monocromática de un imagen de color verdadero (true color).

Existen circunstancias en las que se cuenta con imágenes en escala de grises y por diferentes motivos se requiere tener la representación en color de dichas imágenes.



Algunos ejemplos muy conocidos se dan en medicina, en el caso de una tomografía axial o una radiografía, las cuales se obtienen en principio en escala de grises debido a que no se utiliza el espectro visible para la obtención de dichos exámenes. Otro ejemplo típico es en radioastronomía, que para obtener imágenes se usan toda una gama diversa del espectro radioeléctrico aparte de la luz visible. La necesidad de tener las imágenes en color es como consecuencia de que es más sencillo para la visualización humana, el encontrar ciertas zonas con información importante si hay un cambio brusco de color que si simplemente hay algún cambio de intensidad.

En principio los colores a utilizar en las imágenes son del todo arbitrarios, es decir que no tienen ninguna relación con el color que tendría en la vida real, puesto que lo que se busca, principalmente es establecer los cambios de tonos de color, por lo que se denominan procedimientos de pseudo-color a estos procesos o técnicas. Sin embargo, en ciertos casos la referencia de falso color es utilizada para dar color a imágenes en niveles de gris pero con una relación directa con la vida real, como en el caso de colorear fotos o película antiguas.

Una de las técnicas más utilizadas para obtener la imagen en pseudocolor es la de Rebanada o Rebanado de Intensidad, la cual se describe a continuación:

**Rebanado de intensidad (Density Slicing – DS).** La técnica de “Rebanado de Intensidad” (también denominada “Rebanado de Densidad”), que permite obtener una codificación del color se basa en el principio de asignar un color a un nivel o conjunto de niveles de gris, según los requerimientos propios de cada proceso.

Consideremos la imagen como la interpretación de una función 3D (intensidad contra un espacio coordenado), el método puede verse como si pusiéramos planos paralelos al plano coordenado de la imagen, cada plano rebana a la función en el área de intersección.

En general la técnica puede resumirse como sigue: sea la escala de gris  $[0, L-1]$ , sea el nivel  $l_0$  representado por el negro  $[f(x,y)=0]$ , y sea el nivel  $l_{L-1}$  representado por el blanco  $[f(x,y)=L-1]$ . Suponga que se definen  $P$  planos perpendiculares al eje de intensidad en los niveles  $l_1, l_2, \dots, l_p$ . Suponiendo que  $0 < P < L-1$ , los  $P$  planos parten la escala de grises en  $P+1$  intervalos  $V_1, V_2, \dots, V_{P+1}$ . La asignación de color a los niveles de gris de hacen acorde a la relación:

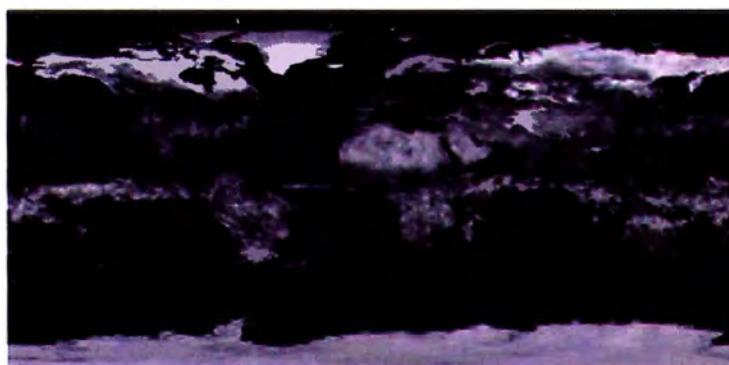
$$F(x,y)=c_k \quad \text{si } f(x,y) \in V_k \quad (4.4)$$

Donde  $c_k$  es el color asociado con el  $k$ -ésimo intervalo de intensidad  $V_k$  definido por la partición de planos  $l=k-1$  y  $l=k$



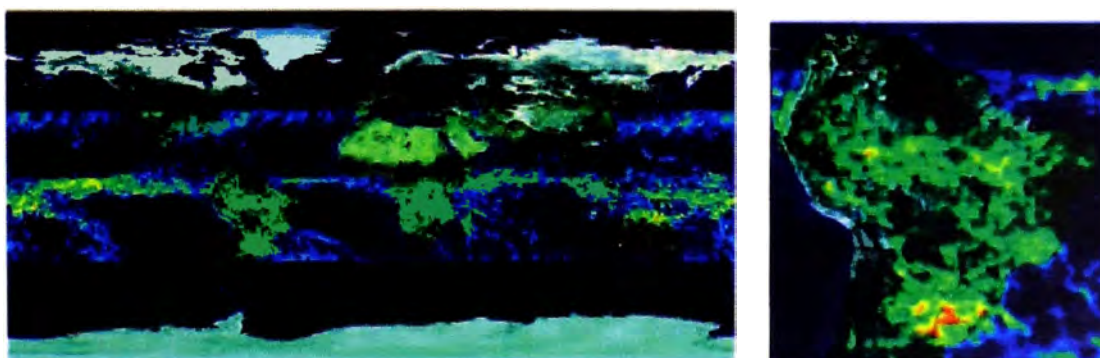
parte de ellas ocurre en los océanos. Un método para obtener los datos de precipitación pluvial es utilizar la información satelital del TRMM (Tropical Rainfall Measuring Misión), que utiliza tres sensores espaciales para detectar lluvia: un radar de precipitación, un visualizador de microondas y un escáner de luz visible e infrarroja.

El resultado de los varios sensores de lluvia son procesados, resultando en valores estimados promedio de lluvia sobre un periodo de tiempo dado en el área monitoreada, a partir de lo cual se generan imágenes en escala de grises cuyos valores de intensidad correspondan directamente a la cantidad de lluvia, con cada uno de los píxeles representando áreas de tierra física cuyos tamaños dependen de la resolución de los sensores, tal como se muestra en la Figura N° 4.2.



**Figura N° 4.2 – Imagen en Niveles de Gris de Precipitación Fluvial**

El análisis visual de la imagen en tonos de grises mostrada en la Figura N° 4.2 anterior es bastante difícil. Sin embargo, si utilizamos colores, y se logra construir una imagen como la que se muestra en la Figura N° 4.3, se puede ver con más facilidad que los valores cercanos al azul significan valores bajos de precipitación mientras que los rojos corresponden a valores altos.



**Figura N° 4.3 – Imagen en Pseudo-color de Precipitación Fluvial**

En el Capítulo V se presentan dos ejemplos de utilización práctica de esta técnica, a partir de imágenes en niveles de gris, procesadas con programas desarrollados especialmente para este Informe de Suficiencia, utilizando algunas de las funciones propias de matlab.

Existen otros tipos de transformaciones que son más generales y por lo tanto son capaces de obtener rangos mayores en los resultados de realce de pseudo-color, que el simple rebajado de intensidades. Uno de ellos es el de realizar tres transformaciones independientes de los niveles de gris de cualquier píxel de entrada, los cuales se utilizan como entrada separadamente en los canales rojo, verde y azul por ejemplo para un monitores de televisión a color. Este método produce una imagen compuesta, cuyo contenido de color está modulado por la naturaleza de las funciones de transformación.

## **CAPÍTULO V**

### **CASOS PRACTICOS DE ESTUDIO**

En este capítulo del Informe de Suficiencia, mostraremos los casos prácticos de Procesamiento de Imágenes, con la utilización de las metodologías de Tonalidad de Grises y Pseudo-color, para lo cual en algunos casos se ha tenido que desarrollar los códigos necesarios de los programas en MATLAB, habiéndose obtenido las imágenes a procesar, así como la realización de varias pruebas de dichos programas, con el fin de determinar los resultados con diferentes imágenes, o con la utilización de diferentes parámetros y condiciones. En total para el caso de Tonos de Grises se han utilizado cuatro imágenes a color, dos en formato \*.jpg y dos en formato \*.tif como imágenes de prueba, mientras en el caso de pseudo-color se han realizado las evaluaciones y pruebas sobre dos imágenes originales en tonos de grises o blanco y negro.

#### **5.1.- Obtención de Tonos de Grises y Filtrado**

En este caso, se desarrolló el programa para Matlab denominado “Obtener\_Grises”, cuyo listado se presenta en el ANEXO A, en el cual se utilizó algunas de las funcionalidades de las herramientas y rutinas existentes en Matlab. Mediante la utilización del Programa “Obtener\_Grises”, se logró obtener la imagen en Tonos de Grises a partir de una imagen original en color.

La primera parte del programa desarrollado es para poder obtener la imagen de origen (color) a partir de cualquier archivo en formato \*.jpg o \*.tif existente en el directorio defendido para trabajar con el programa Matlab. En nuestro caso hemos utilizado cuatro ejemplos de imágenes a color, las cuales se muestran en la Figura N° 5.1, donde dos de ellas corresponden a formatos \*.jpg y las otras dos a formatos \*.tif.



**Caso 1**



**Caso 2**



**Caso 3**

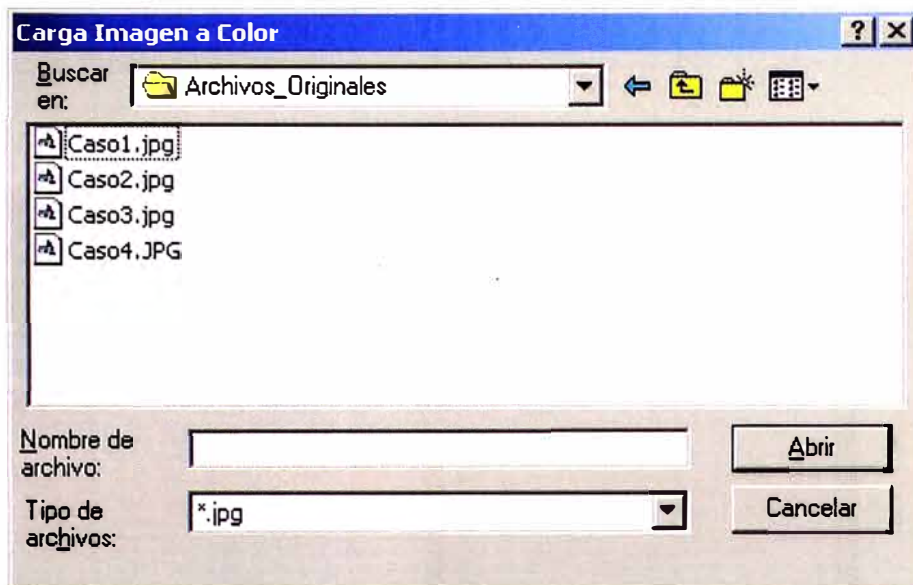


**Caso 4**

**Figura N° 5.1 - Imágenes Originales (4 Casos)**



Para estos efectos se utiliza la función `uigetfile`, que permite la presentación de una pantalla (ventana) para seleccionar el archivo a leer, tal como se muestra en la Figura N° 5.2.



**Figura N° 5.2 – Pantalla para seleccionar Archivo Original (\*.jpg)**

La función `uigetfile` permite realizar una primera selección del tipo de archivos a leer, según el parámetro de tipo de extensión utilizados en la invocación de dicha función. En el siguiente ejemplo:

```
uigetfile('*.*jpg','Carga Imagen a Color');
```

Se realizara la lectura de un archivo con extensión `*.jpg`, cuya relación es mostrada en una ventana (Figura N° 5.2), con el objeto que el usuario pueda realizar la selección de la imagen a ser la procesada. En el caso de tratarse de una imagen con extensión `*.tif`, la estructura de la instrucción será la siguiente:

```
uigetfile('*.*tif','Carga Imagen a Color');
```

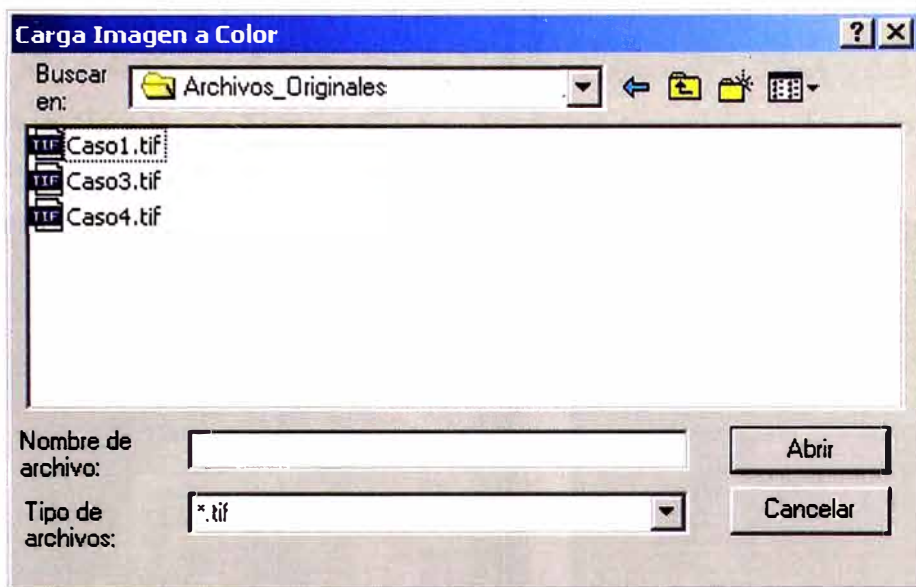
El resultado será la presentación de una ventana, tal como se presenta en la Figura N° 5.3.

Mediante la función `imread`(‘archivo de entrada’), se lee la imagen a color a la cual se ha de convertir en tonos de grises. La descripción detallada de la función `imread`, se presenta en el ANEXO A.

La segunda parte importante, es convertir la imagen leída a su representación en tonos de grises, para lo cual nos apoyamos en la función matlab `rgb2gray`, tal como se muestra en la siguiente instrucción:

```
Imag_gris = rgb2gray(origen)
```

Donde “origen” representa la información de la imagen original leída, que mediante la función **rgb2gray** es convertida a tonalidad de grises, la misma que en forma de valores de intensidad se almacenan en la matriz *Ima\_gris*. En el ANEXO A, se presenta una descripción de la función **rg2bgray**.



**Figura N° 5.3 – Pantalla para seleccionar Archivo Original (\*.tif)**

Los comandos o instrucciones **subplot** e **imshow**, permiten mostrar las imágenes original y en tonos de grises como resultado de los pasos anteriores.

Como tercera parte se ha procedido a aplicar diferentes técnicas de filtrado a la señal obtenida en “tonos de grises”, con la finalidad de realizar una comparación visual de la presentación de la imagen procesada bajo estos tipos de filtros. Para ello se hace uso de la función matlab **fspecial**, a fin de generar la función filtro específico a aplicar, por ejemplo:

```
hgauss = fspecial('gaussian'); ó hlaplac = fspecial('laplacian');
```

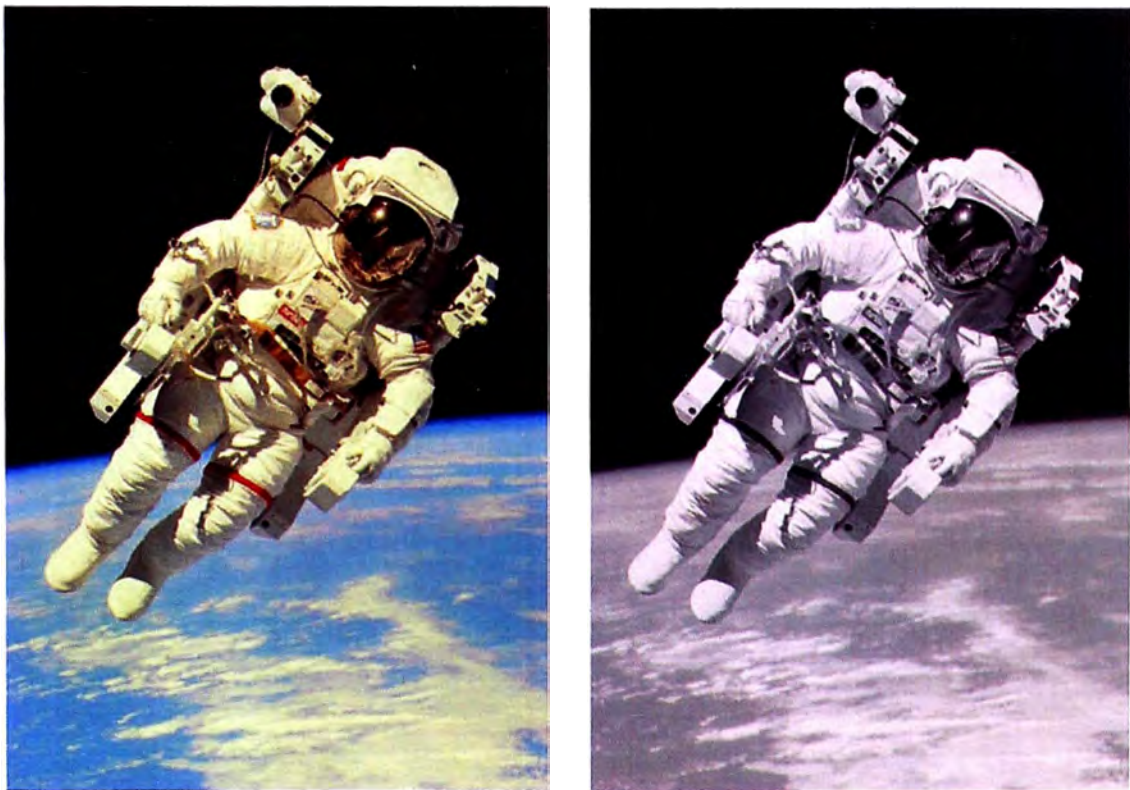
La función. **fspecial** crea un filtro en dos dimensiones de acuerdo a la especificación ingresada, la que puede ser:

- 'average' promedio
- 'disk' promedio circular
- 'gaussian' gaussiano pasa bajo
- 'laplacian' aproximación al operador Laplaciano 2D
- 'log' filtro Laplaciano Gaussiano



- 'motion' de movimiento
- 'prewitt' prewitt con énfasis horizontal
- 'sobel' sobel con énfasis horizontal
- 'unsharp' realce de contraste

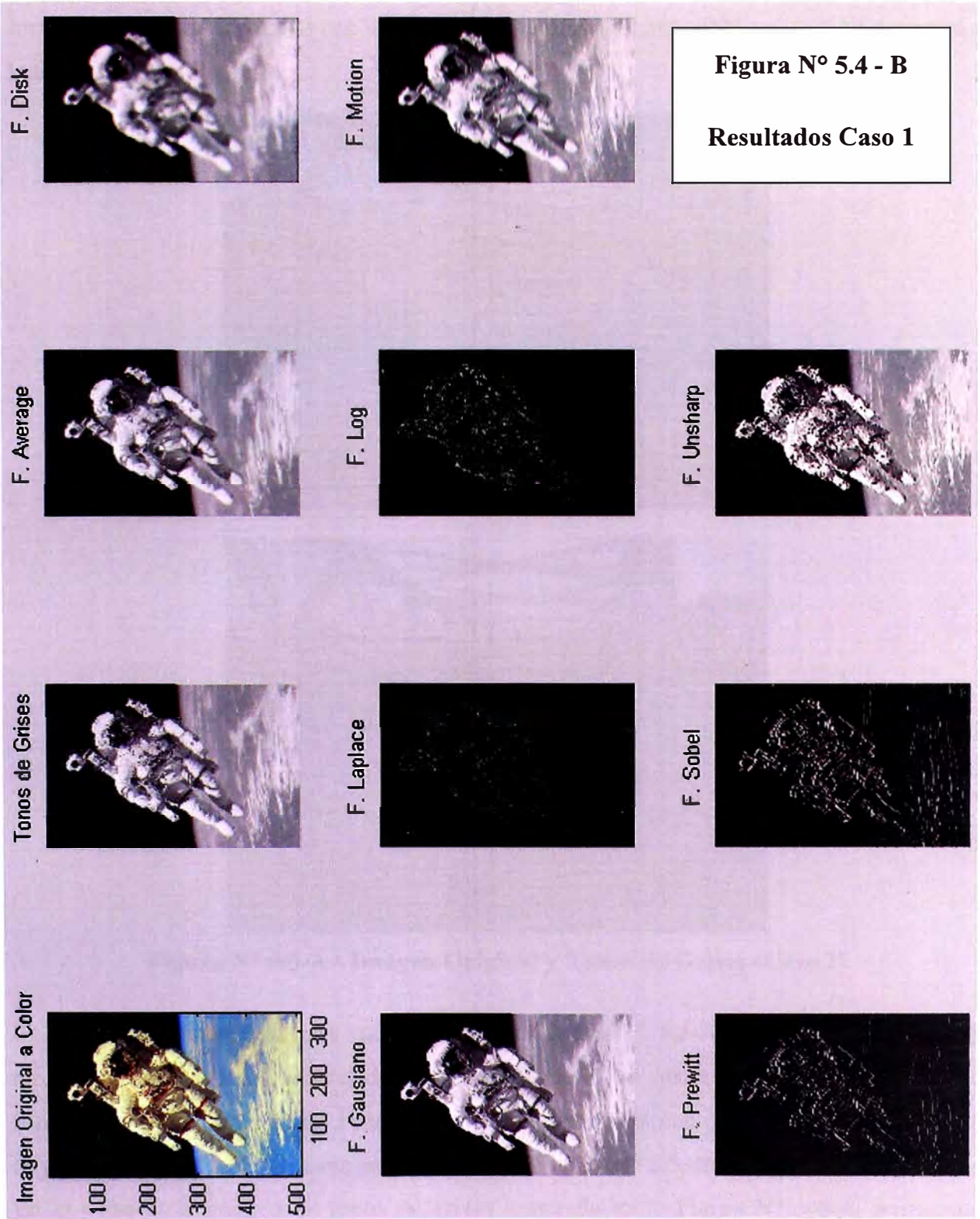
Los resultados de la representación de la imagen original y tonos de grises, así como los que se obtiene por la aplicación de las diferentes “mascaras” o filtros, se presentan en la Figuras N°s 5.4-A y 5.4-B para el Caso 1, en las Figuras N°s 5.5-A y 5.5-B para el Caso 2 , respectivamente. En las Figuras N°s 5.6-A y 5.6-B para el Caso 3 y en las Figuras N°s 5.7-A y 5.7-B para el Caso 4.



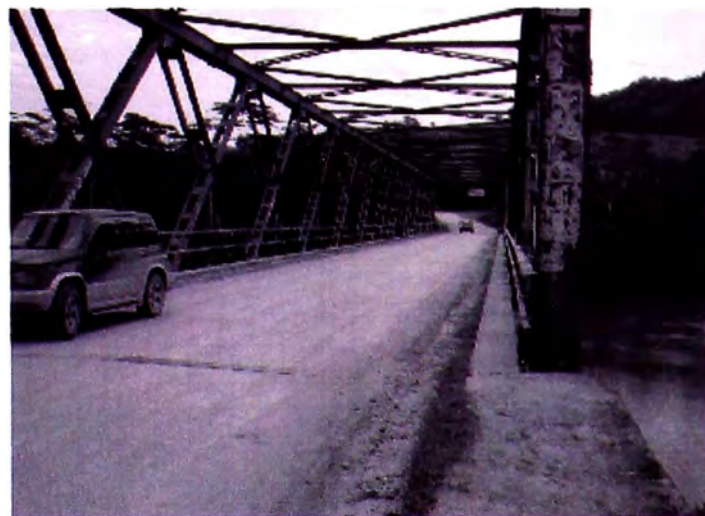
**Figura N° 5.4-A – Original y Tonos Grises (Caso 1)**

En la Figura N° 5.4-A se puede observar la imagen original y la resultante de aplicar la función de tonos grises, de donde se puede en primer lugar concluir que la conversión a tonos de grises da como resultado una imagen de buena calidad para una representación gráfica u observación visual.

En la Figura N° 5.4-B, se presenta los resultados de la imagen filtrada con varios modelos o alternativas de filtrado, siendo la resultante de los filtros promedio y gaussiano, los filtros que permiten obtener un mejor resultado, en términos de representación para una observación visual.



En la imagen del Caso 2, la Figura N° 5.5-A corresponde a las imágenes original y en tono de grises, como resultado del programa “Obtener\_grises.m”. La imagen original está en formato \*.jpg y fue obtenida con una cámara fotográfica el año 2003 en un viaje personal a la localidad de Tarapoto.

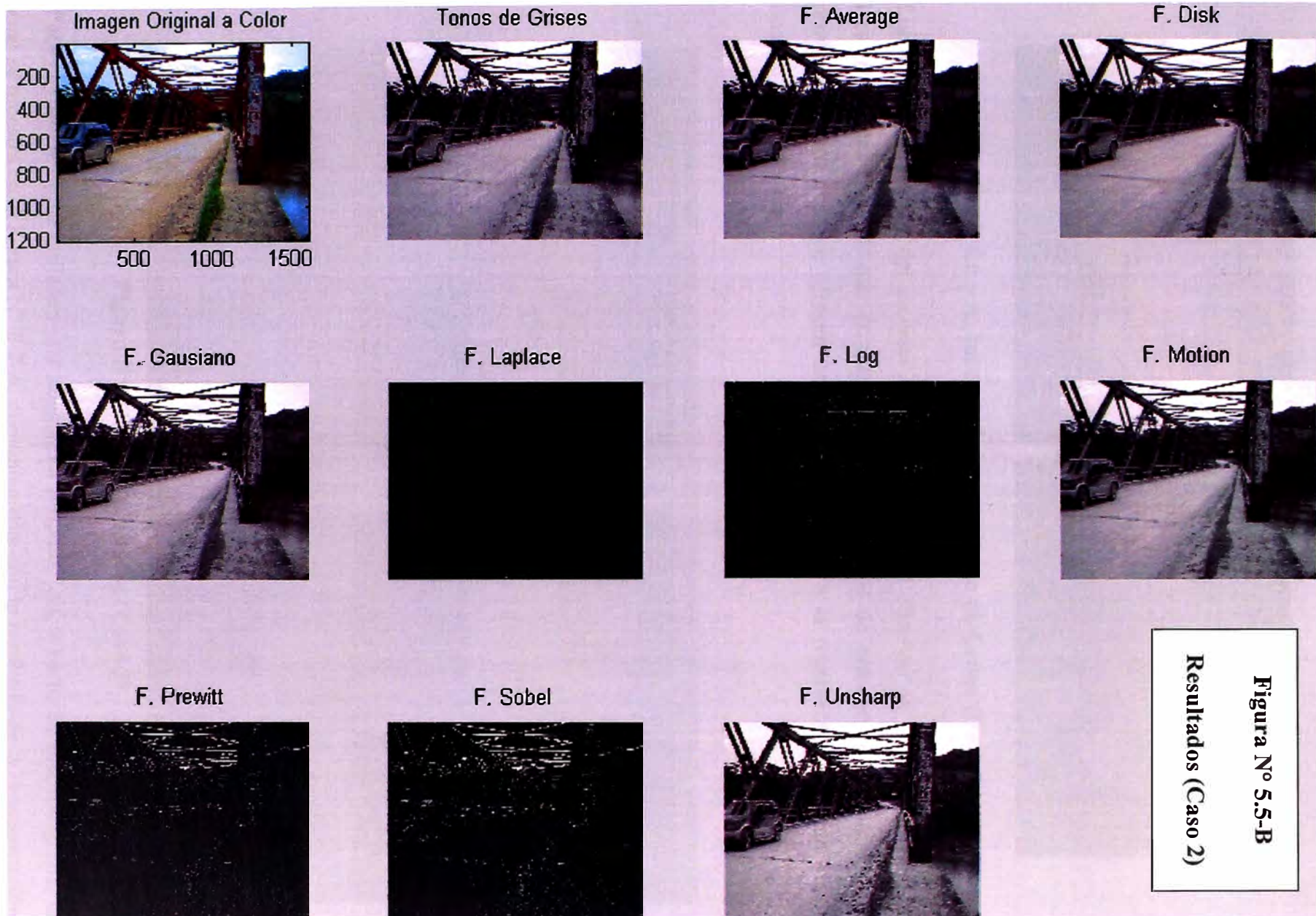


**Figura N° 5.5-A.- Imagen Original y Tonos de Grises (Caso 2)**

La imagen de tono de grises mostrado en la Figura N° 5.5-A es de buena calidad y resolución, sin embargo a nivel de imágenes filtradas, las obtenidas con la aplicación de filtros con las opciones Disk, Promedio y Movimiento, ofrecen una buena calidad gráfica de presentación visual, tal como se observa en la Figura N° 5.5.-B.

En el Caso 3, la imagen en tonos de grises mostrada en la Figura N° 5.6-A, representa bastante bien a la imagen original, la que corresponde a una imagen en formato \*.tif, obtenida de internet.

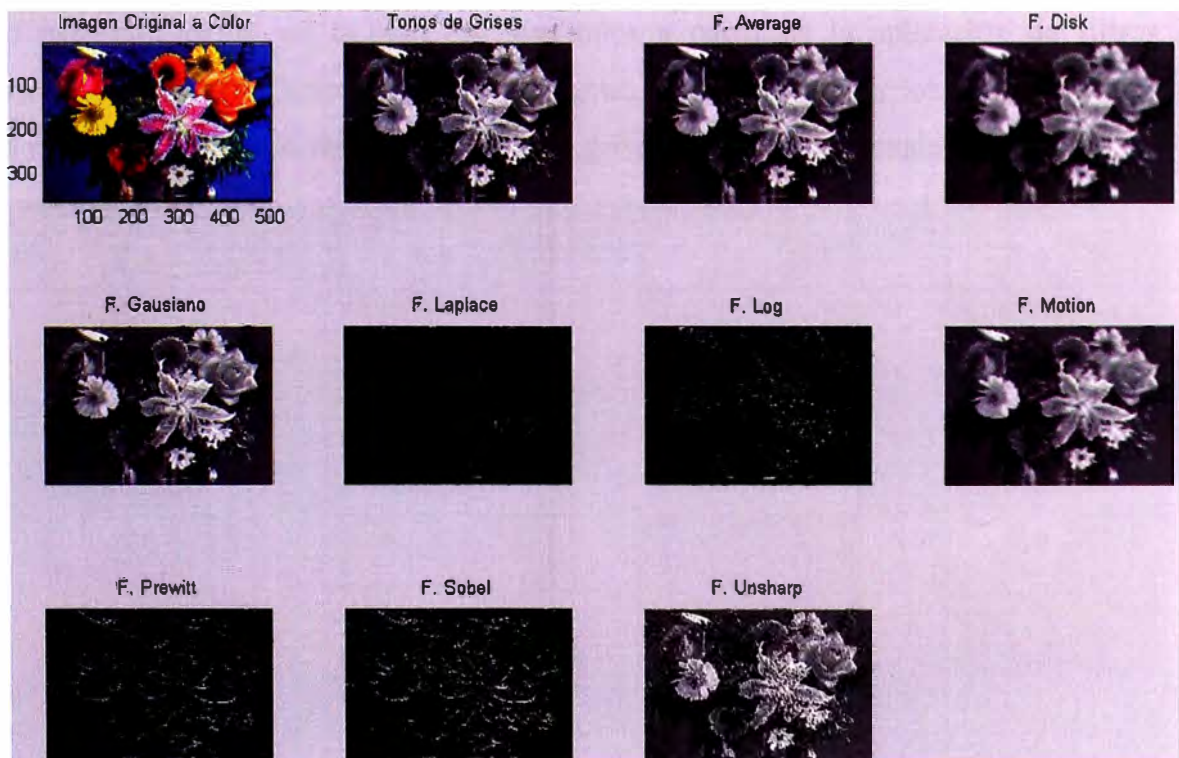






**Figura N° 5.6 – A: Imagen Original y Tono de Grises (Caso 3)**

Luego de la aplicación del proceso de filtrado, se puede observar en la Figura N° 5.6-B: Resultados del Caso 3, que la aplicación de los filtros de promedio, gaussiano y de movimiento son los que ofrecen una mejor respuesta a la representación visual de la imagen en tonos grises.



**Figura N° 5.6 – B: Resultados (Caso 3)**

Para el Caso 4, se ha utilizado una imagen en formato \*.tif, a partir de una fotografía, la cual se muestra en la Figura N° 5-7-A, conjuntamente con le resultado de la representación



de dicha imagen en tonos de grises. En este Caso 4, la representación de la imagen en tonos de grises mostrada en la Figura N° 5.7-A, también representa aceptablemente a la imagen en color, siendo una imagen fácil de reconocer o entender por simple apreciación visual, sin mayor pérdida de información para su reconocimiento visual por parte del ojo humano.



**Figura N° 5.7-A: Imagen Original y Tono de Grises (Caso 4)**

En la Figura N° 5.7-B se observan los resultados de la aplicación de los diversos tipos de filtro para el Caso 4, en la cual los resultados a partir de la aplicación de filtros de promedio, de promedio circular y de movimiento, permiten mostrar los mejores resultados en la representación de la imagen en tonos de grises para efectos visuales.



**Figura N° 5.7 - B: Resultados (Caso 4)**

A partir de las evaluaciones realizadas en los cuatro casos detallados anteriormente, se pueden arribar a las primeras conclusiones como son: i) una representación en tonos de grises es una buena representación gráfica y visual de las imágenes a colores, la cual es de utilidad para diferentes fines o usos, según cada caso que más adelante se detallaran; ii) el tratamiento y proceso de imágenes en tonos de grises, requiere de menos recursos en términos de tiempo, memoria, etc. de los dispositivos informáticos utilizados para tal fin; iii) el almacenamiento de imágenes en tonos de grises requiere de menor espacio en los medios de almacenamiento de información digital; iv) la aplicación de diversos tipos de filtros a las imágenes en tonos de grises tienen diferentes respuestas según las características propias de cada imagen, con lo cual no se puede establecer una única regla, en el sentido que algún tipo de filtro puede ser el que de mejor respuesta con cierta imagen. Finalmente, se puede también establecer que las funciones existentes en la herramienta tecnológica MATLAB, permite realizar estas evaluaciones de manera eficiente, siendo necesario el desarrollo de los scripts o programas necesarios para la utilización de las diversas funciones, comandos, sentencias, etc. que permitan modelar y procesar la información correspondiente. En el ANEXO A, se ha incluido adicionalmente, el programa `Obtener_grises_tif.m`, que permite la lectura, proceso y grabación de imágenes en formato \*.tif.

## **5.2.- Pseudo Color – Color Falso**

Otra de las técnicas en el procesamiento de imágenes, es la “colorear” o asignar colores a imágenes en blanco y negro o que se encuentren representadas en tonos de grises, lo cual de alguna manera resulta ser contraria a la técnica descrita utilizada en el acápite 5.1 anterior, referida justamente a la representación de imágenes de color en tonos grises.

En este caso y a efectos de la evaluación del caso práctico, se ha escrito diferentes “scripts” o programas que utilizando algunas de las funciones de matlab, han de permitir la obtención de imágenes coloradas a partir de imágenes en blanco y negro y/o tonos de grises.

Los “scripts” se han integrado en un módulo que bajo una representación gráfica permite presentar un menú con las diversas funcionalidades desarrolladas, a fin de que el usuario pueda interactuar con la imagen de entrada o de análisis, a la cual en principio hay que leerla y asignarla a una arreglo tipo matriz, y luego poderle aplicar diferentes patrones de color (`colormap`), así como ir modificando los umbrales (valor mínimo y máximo) a

considerarse para la asignación de un color determinado en función del nivel de gris que tenga la imagen original.

La primera parte consistió en la escritura de los programas o “scripts” para las funciones principales de lectura o carga de la información correspondiente a la imagen a ser procesada, la asignación de un determinado patrón o gama de colores a ser utilizados como “tabla de búsqueda de color”, para la asignación de un color, según el nivel o tono de gris que tenga cada píxel, en una imagen en blanco y negro o en tonos de grises. Finalmente fue necesario desarrollar los programas que permitan reasignar los diferentes colores a cada píxel, teniendo en consideración la modificación de los valores umbrales (mínimo y máximo). Los programas desarrollados son los siguientes:

- carga\_imagen.m
- selec\_patron.m
- deter\_valmin.m
- deter\_valmax.m
- principal.m

En las Figuras N°s 5.8 y 5.9 se presentan los casos considerados para la aplicación de pseudo-color o color falso, denominados Caso “A” y “B”, respectivamente. En este punto, es importante precisar que en diversos documentos técnicos que tratan sobre estas metodologías, establecen una diferencia entre el caso cuyo fin es de carácter estético o de presentación visual, y el caso en que se trata de reconocer alguna información relevante que en una representación de tonos de grises no es posible observarla, estableciendo que en el primer caso se trata de pseudo-color y en el segundo se denomina color falso. Sin embargo, para fines prácticos y en opinión nuestra, metodológicamente se trata de lo mismo, dado que lo que se pretende es la asignación de un color determinado según cada fin, a un nivel dado de tono de gris o blanco y negro, de las imágenes monocromáticas.

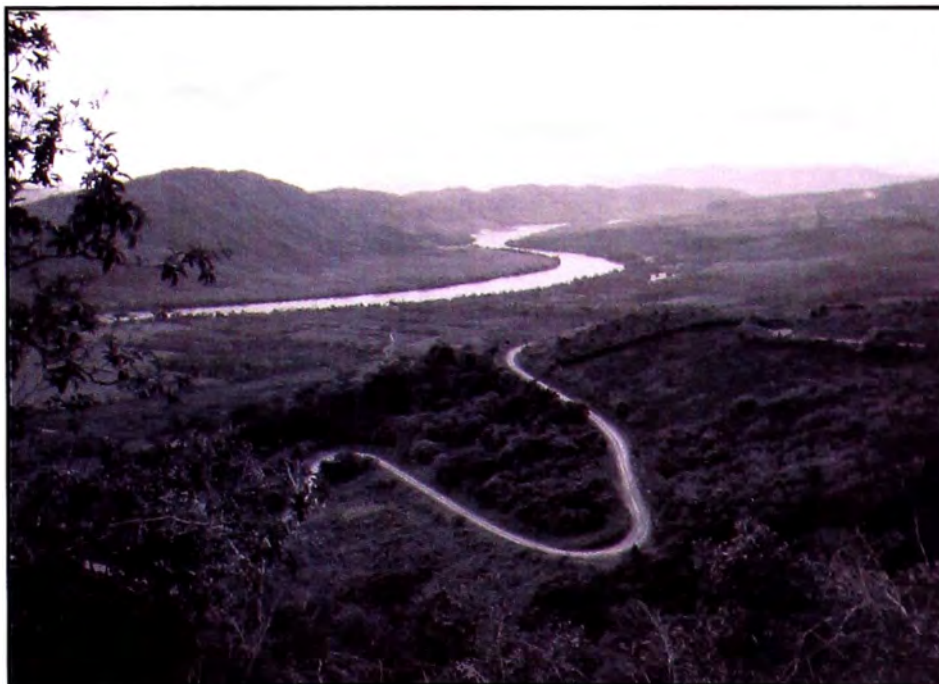
A estas alturas, es obvio que si se tiene una imagen en blanco y negro puros, solo sería posible dos colores, uno para el blanco y otro para el negro, en el caso de requerir asignar colores a la imagen original. En consecuencia, es importante resaltar que un mayor nivel de tonos de grises, es posible asignarle una mayor gama de colores.

Por otro lado, la capacidad visual del ojo humano, permite un mayor reconocimiento de diferentes colores, en comparación con los tonos de grises, en donde la capacidad del ojo humano se encuentra bastante restringido.





**Figura N° 5.8: Imagen Original – Caso “A”**



**Figura N° 5.9: Imagen Original – Caso “B”**

Luego de la escritura de los scripts respectivos, a fin de obtener un modulo integrado, se utilizó la funcionalidad de matlab, para crear “menús” mediante la funcionalidad GUI o GUIDE, asignando en cada caso un script a un botón o representación gráfica, tal como se muestra en la Figura N° 5.10.



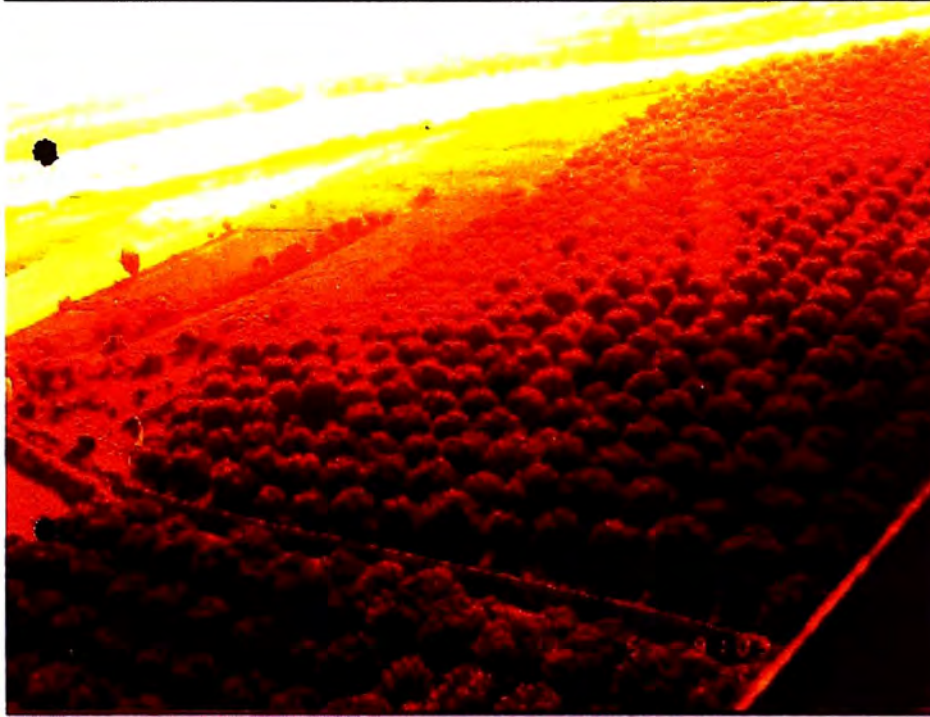
**Figura N° 5.10: Menú Principal**

En el ANEXO B, se presenta el listado completo de cada uno de los programas desarrollados para las evaluaciones de este numeral 5.2, así como una descripción de algunas de las características propias y las principales funciones matlab utilizadas, las que han sido documentadas en cada script, mediante la escritura de los comentarios respectivos. Para efectos de estas evaluaciones se ha utilizado 6 diferentes patrones o mapas de colores, así como diferentes valores umbrales para la asignación de los mapas de color a cada nivel de gris de la imagen a procesar. Los patrones utilizados son: 'hot', 'hsv', 'gray', 'pink', 'cool', 'bone' y 'flag'.

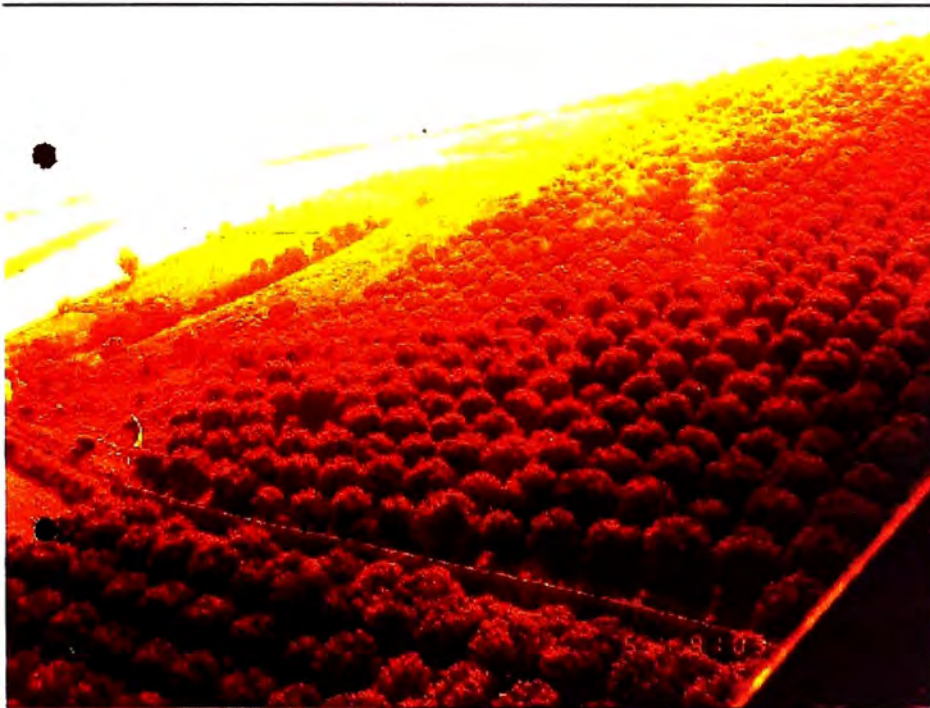
La primera imagen a evaluar corresponde a una fotografía en blanco y negro de una vista aérea de un fundo en la localidad de Ica-Perú, imagen a la cual se le han aplicado diversos patrones de color con distintos niveles para el valor mínimo y valor máximo. En la Figura N° 5.11 se presentan los resultados de la utilización de un patrón de color 'hot' con dos diferentes niveles de umbral, en el primer caso valor mínimo = 0.00 y valor máximo = 0.82 y en el segundo caso con un valor mínimo = 0.00 y un valor máximo = 0.60.

Tal como se puede observar en el segundo caso puede verse de mejor forma el área de cultivo, incluyendo que a partir de ello puede ser determinada dicha extensión, ya que la

zona en color blanco corresponde a el área que no tiene ningún tipo de cultivo.  
En el ANEXO B, se presenta la totalidad de resultados obtenidos por aplicación de diferentes patrones de color y con dos niveles de valores máximos,



Valor mínimo = 0.00 - Valor Máximo = 0.82



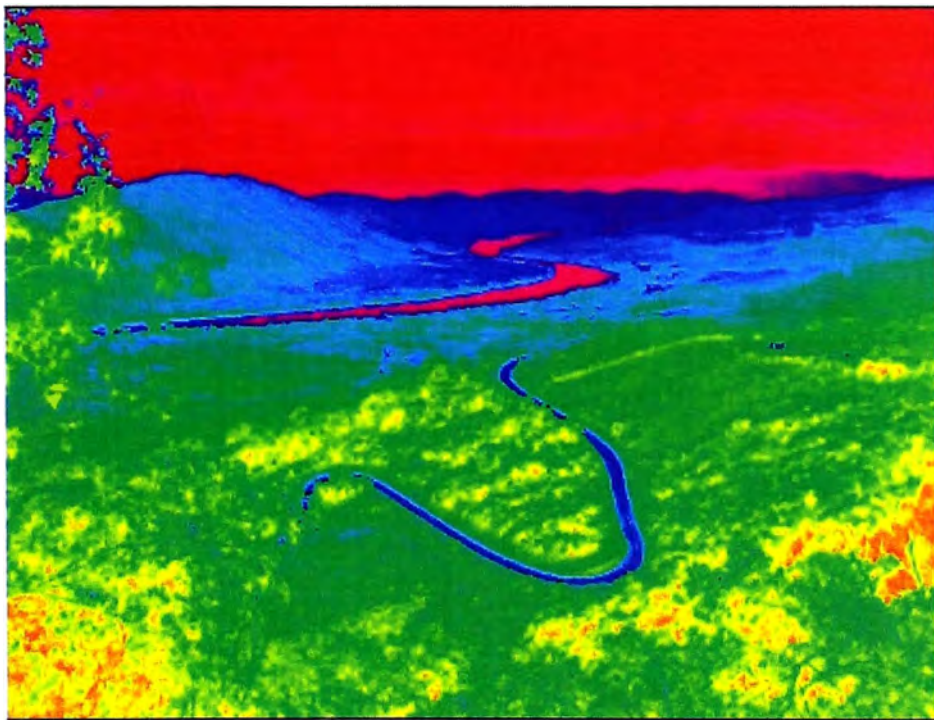
Valor mínimo = 0.00 - Valor Máximo = 0.60

**Figura N° 5.11: Resultados Pseudo-color Caso "A" – Patrón de color "hot"**



El Caso “B” corresponde a una imagen tomada en una zona de selva baja, la cual ha sido evaluada también con la aplicación de varios patrones de color y dos niveles de valores para los umbrales del mínimo y máximo para la asignación de un determinado color según el nivel de gris de cada uno de los píxeles en dicha imagen.

En la Figura N° 5.12, se presenta el resultado de aplicar un patrón de color (colormap) del tipo “hsv” con un valor mínimo = 0.00 y valor máximo = 0.60, donde se puede apreciar la existencia de una carretera, la que se presente en un trazo azul, característica que en el imagen original podría haberse confundido con la presencia de dos ríos o de dos caminos. La imagen en falso color o pseudo-color, permite la diferenciación evidente entre dicha información contenida en la imagen original.



**Figura N° 5.12 - Imagen hsv 0.00 – 0.60 – Caso “B”**

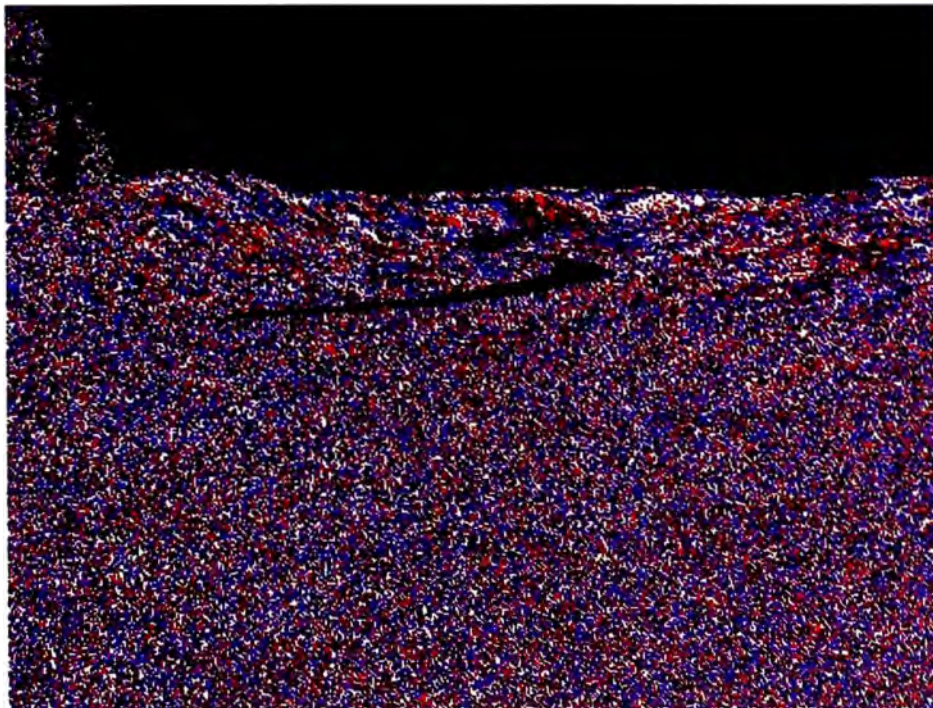
Por otro lado, en la Figura N° 5.13 se presenta el resultado obtenido mediante la utilización de un patrón de color (colormap) del tipo “hot” con valor mínimo = 0.0 y valor máximo = 0.80, se observa que el río y la carretera se presentan en colores diferentes, lo cual significa que se trata de elementos diferentes dentro de la imagen de origen.

Por último, en la Figura N° 5.14 se muestra el resultado de la aplicación de un patrón de color (colormap) tipo “flag” con un valor mínimo = 0.00 y valor máximo = 0.80, permite el reconocimiento de áreas en la imagen original con “ausencia de información” o que no corresponde a superficie de tierra. A partir de ello es viable la determinación de la

extensión de una imagen con determinado tipo de información, como por ejemplo, nieve, agua, área cultivada, etc.



**Figura N° 5.13: Imagen “hot” 0.00 – 0.80 – Caso “B”**



**Figura N° 5.14: Imagen “flag” 0.00 – 0.80 - Caso “B”**

En el ANEXO B, se presentan las imágenes resultantes de la aplicación de diferentes patrones de color (colormap), utilizando diferentes valores para el valor máximo (0.60 y 0.80).

## CONCLUSIONES

Luego de haber revisado los fundamentos teóricos del Procesamiento Digital de Imágenes, en cuanto a los conceptos de Muestreo, Discretización de Señales Continuas, Transformada de Fourier y en especial las técnicas y conceptos del Procesamiento Digital de Imágenes, así como la ejecución y análisis de los ejemplos prácticos descritos en el Capítulo V de este trabajo, se llega a las siguientes conclusiones:

1. El Procesamiento Digital de Imágenes basa su desarrollo en un amplio y completo conocimiento de los fundamentos matemáticos especialmente de la Transformada de Fourier y la potencialidad de la misma ha permitido el gran desarrollo alcanzado en los últimos años.
2. El Procesamiento Digital de Imágenes corresponde a un nuevo campo y especialidad en el conocimiento para la Ingeniería Electrónica, con múltiples e innumerables aplicaciones en todo tipo de actividades del desarrollo humano, incluyendo campos del conocimiento que hasta hoy tenían muy poca conexión, como es en el caso de señales o imágenes satelitales para fines de mejorar el desarrollo agrícola, de prevención de desastres naturales, o el reconocimiento de señales biológicas para determinar los niveles de salud de la persona humana. Es decir, no hay campo del conocimiento que no pueda aprovechar las ventajas y potencialidades del Procesamiento Digital de Imágenes.
3. En el Procesamiento Digital de Imágenes, se ha observado que existe una potencialidad enorme en el tratamiento de las mismas, donde en realidad se puede hacer cualquier cosa, existiendo hoy en día muchos algoritmos desarrollados para cada caso o aplicación en particular, y que gracias a la globalización de la información a través de INTERNET, ya no es necesario ni mucho menos conveniente redescubrir o rehacer códigos que ya están desarrollados. Basta con realizar la búsqueda pertinente y luego de acuerdo a los requerimientos específicos de cada problema o caso, realizar las

adaptaciones necesarias, con lo cual el tiempo de implementación de una solución realmente se ha reducido a niveles de horas o quizás minutos, en muchos casos.

4. Las herramientas existentes para poder simular o realizar las pruebas de implementación de alguna solución son realmente potentes, permitiendo la ejecución de una gama de pruebas y/o evaluaciones como forma de buscar la mejor solución para un problema determinado, sin tener que realizar grandes inversiones en equipos sofisticados, ni mucho menos complicados o implementar grandes desarrollos de circuitos; tal es el caso del MATLAB, SIMULINK; LABVIEW, etc., que son herramientas de tecnología de información con una amplia variedad de funcionalidades, tal como se demostró en las evaluaciones presentadas en el Capítulo V.
5. Cada tipo de imagen y su tratamiento digital, es todo un campo del conocimiento para que a través de la creatividad, análisis y conocimiento de detalle del tipo de información y las características de las misma, se puedan desarrollar técnicas y procedimientos para el tratamiento y procesamiento de dicho tipo de imagen, como pueden ser en el caso de la biomedicina, la información sísmica y/o geológica, la de mapas o imágenes satelitales, etc. Con seguridad en el corto plazo se abrirán nuevas "especialidades" para el tratamiento de determinado tipo de imagen.
6. Los resultados obtenidos como parte de los casos prácticos desarrollados, han permitido comprender de manera clara y concreta el significado del "Tratamiento o Procesamiento de Imágenes", al tener que haber realizado el desarrollo de algunos programas en MATLAB, haber ejecutado las pruebas y correcciones de dichos programas, así como la simulación de diferentes tratamientos a las imágenes de prueba.. Lo contrario hubiera significado solo haberse quedado en el campo de los conceptos y definiciones teóricos, de los fundamentos matemáticos de las técnicas y procedimientos para el Procesamiento de Digital de Imágenes, sin realizar la parte práctica correspondiente.
7. La representación de la imágenes a color en Tonos de Grises, permite que dichas imágenes puedan ser almacenadas y por consiguiente preservadas, utilizando un menor espacio en los dispositivos de almacenamiento masivo, teniendo una relación de 1 a 10 o hasta de 1 a 20, según sea el formato en el que la imagen a color este almacenada o generada. Este características, implica a su vez un menor costos de almacenamiento de

dicha información, o una mayor cantidad de información almacenada a un menor costo, o ambas características a la vez .

8. Con la utilización de Tonos de Grises para la representación de imágenes a color, estas pueden estar más fácilmente disponibles para ser compartidas por otros usuarios, dado que su transmisión ocupa un menor tiempo y ancho de banda, requiriendo de menores recursos tecnológicos para esta disposición.
9. La representación en Tonos de Grises, permite la utilización de recursos tecnológicos existentes, como son impresoras en blanco y negro, faxes, equipos monocromáticos, etc., para lo cual se convierte previamente a la imagen en color en una de Tonos de Grises.
10. La aplicación de técnicas de Pseudo-color o asignación de color, según un determinado patrón, permite el reconocimiento de información relevante para el observador, por la diferencia de colores a la cual el ojo humano es más sensible que a los diferentes tonalidades de grises.
11. Según cada imagen y el tipo de análisis a que dicha imagen esta sujeta, se pueden emplear diferentes patrones de color, no existiendo una única regla o patrón de color que de siempre la mejor información de la imagen procesada, según el interés del usuario.
12. Las técnicas de Pseudo-color son muy utilizadas para representaciones gráficas donde se debe enfatizar las diferencias entre los diferentes componentes de información de la imagen original.

### **Aplicaciones:**

La obtención de imágenes en tonos de grises a partir de imágenes a color, así como el procesamiento de imágenes en blanco y negro para obtener imágenes a color mediante procedimientos de pseudo-color, tienen una serie de aplicaciones prácticas, tales como las que a continuación se detallan;

1. Impresión de imágenes a color en dispositivos monocromáticos como son impresoras, faxes, etc., así como su visualización en equipos blanco y negro.
2. Disponer de gran volumen de información en línea, como consecuencia de procesos de escaneo de documentos originales a color, los cuales de mantenerse en su formato



original requerirían de hasta 20 veces más del espacio requerido por las imágenes en tonos de grises, en los dispositivos de almacenamiento masivo de información. Un ejemplo, permitirá una mejor comprensión de este caso:

Asumamos que tenemos que almacenar la información de 600,000 documentos, de un promedio de 50 páginas cada uno, donde la imagen de una página a color podría requerir de 10MB a 15MB, lo cual haría necesario disponer entre 300 GB ( $600,000 \times 50 \times 10 \text{ MB} = 300,000,000 \text{ MB}$ ) y 450 GB ( $600,000 \times 50 \times 15 \text{ MB}$ ). Sin embargo en el caso de tener dicha información en tonos de grises esto representaría entre 15GB y 45GB, haciendo mucho más fácil su disponibilidad, almacenamiento y capacidad de distribución, con fines de consulta y preservación, aplicable en Centros de Documentación, Bibliotecas, Archivos Centrales, etc.

3. Utilización de técnicas de Pseudo-color para el tratamiento de información que de manera natural esta en tonos de grises, como son la mayoría de información médica o biomédica, a la cual aplicando diferentes patrones de color se logra reconocer información correspondiente a los diversos componentes de la imagen, como son tejidos blandos, huesos, nervios, músculos, tejidos o cuerpos extraños o con malformaciones, etc.
4. En el procesamiento de información sísmica, la que originalmente también esta en tonos de grises, a la que por aplicación de determinados patrones de color, es posible reconocer formaciones geológicas que podrían corresponder a determinar yacimientos conteniendo determinado tipo de minerales y/o estructuras que podrían contener hidrocarburos como gas, petróleo o condensados.
5. Capacidad de procesar información antigua, para que aplicando un patrón de color determinado, se pueda tener una mejor presentación de la imagen original, en la que sea más fácil para la persona humana el reconocimiento de alguna información de interés.
6. En el reconocimiento y determinación de extensiones bajo nieve, con inundaciones, zonas desérticas, zonas de tierra, áreas sin información, son posibles a partir de ambas técnicas, para lo cual es necesario determinar el número de tonos de grises en un caso, o los niveles de umbral y patrón de color en el segundo caso. El resultado final de la imagen de un área determinada permitiría estimar y reconocer la extensión que se encuentra bajo alguna de las situaciones descritas anteriormente. El Caso "B"

presentado en el ANEXO B, y procesada con un patrón de color “flag” y valores umbrales de 0.00 y 0.80 muestra las zonas con información y las que no., mientras que en el Caso “A” mostrado en el mismo ANEXO B, al procesar la imagen con un patrón de color “pink” y valores umbrales de 0.00 y 0.80, permite reconocer claramente el área cultivada, en contraste con el área que no esta dedicada al cultivo.

## **ANEXO A**

## Programa Obtener\_Grises.m

El listado completo del programa se muestra continuación:

```
function Obtener_grises
clear all; clc; close all;
%
% Esta funcion convierte una imagen RGB a una imagen en Tonos de Grises.
% Se puede realizar con o sin la aplicación de filtros
% Pedro Arce Chirinos
% parce@perupetro.com.pe
%
% Presenta una ventana para poder seleccionar la imagen a procesar (leer)
%
buffer=pwd;
[archivo, path] = uigetfile('* .jpg','Carga Imagen a Color'); cd(path);
% Obtencioan de las partes del archivo; Directorio, nombre, extensión y versión
[path1,archi,ext1,vers1] = fileparts(archivo);

a_input=imread(archivo); % Lee archivo que corresponde o contiene una imagen a color
cd(buffer);
imagris=rgb2gray(a_input); % Convierte archivo leído a Tonos de Grises
figure (1)
% Para presentar la Imagen Original en la parte izquierda de la Pantalla
subplot(1,2,1)
imshow (a_input); title ('Imagen Original');
% Para presentar la Imagn en Tonos de Grises en la parte derecha de la Pantalla
subplot (1,2,2)
imshow (imagris); title ('Imagen Tonos de Grises');
%
% Creacion de diversos filtros a aplicar a la imagen bidimensional
%
haver=fspecial('average'); hdisk=fspecial('disk');
hgauss=fspecial('gaussian'); hlaplac=fspecial('laplacian');
hlog=fspecial('log'); hmotion=fspecial('motion');
hprewitt=fspecial('prewitt'); hsobel=fspecial('sobel');
hunsharp=fspecial('unsharp');
%
% Definición del nombre de los archivos para la grabación de los mismos en el
% Directorio de Trabajo. El Directorio de Trabajo es el mismo donde se encuentra
% el Modelo o Programa
%
arch0 = strcat(archi,'_gris',ext1); arch1 = strcat(archi,'_faver',ext1);
arch2 = strcat(archi,'_fdisk',ext1); arch3 = strcat(archi,'_fgauss',ext1);
```

```

arch4 = strcat(archi,'_flapl',ext1); arch5 = strcat(archi,'_flog',ext1);
arch6 = strcat(archi,'_fmot',ext1); arch7 = strcat(archi,'_fprew',ext1);
arch8 = strcat(archi,'_fsobel',ext1); arch9 = strcat(archi,'_unsh',ext1);
%
% Creacion de las imagenes con la aplicacion de los Filtros definidos
%
pphaver=imfilter(imagris,haver);    ppdisk=imfilter(imagris,hdisk);
ppgauss=imfilter(imagris,hgauss);   pplaplac=imfilter(imagris,hlaplac);
pplog=imfilter(imagris,hlog);       ppmotion=imfilter(imagris,hmotion);
ppprewitt=imfilter(imagris,hprewitt); ppsobel=imfilter(imagris,hsobel);
ppunsharp=imfilter(imagris,hunsharp);
%
% Creacion de una nueva pantalla (Nº 2) para presentar la Siguiete Información:
%
% Imagen Original
% Imagen en Tonos de Grises
% Imagen Resultante por la aplicación de cada Filtro definido en el Programa
%
figure(2)
subplot(3,4,1)
subimage(a_input); title ('Imagen Original a Color');

subplot(3,4,2);
imshow(imagris); title ('Tonos de Grises');

subplot(3,4,3);
imshow (pphaver); title('F. Average');

subplot (3,4,4);
imshow(ppdisk); title('F. Disk');

subplot(3,4,5);
imshow(ppgauss); title('F. Gausiano');

subplot(3,4,6)
imshow(pplaplac); title('F. Laplace');

subplot(3,4,7)
imshow(pplog); title('F. Log');

subplot(3,4,8)
imshow(ppmotion); title('F. Motion');

subplot(3,4,9)
imshow(ppprewitt); title('F. Prewitt');

```

```

subplot(3,4,10)
imshow(ppsobel); title('F. Sobel');

subplot(3,4,11)
imshow(ppunsharp); title('F. Unsharp');

figure(3)
subplot(2,1,1);
subimage(imagris); title ('Imagen Escala de Grises');

subplot(2,1,2);
imshow(ppunsharp); title ('Imagen Escala de Grises - Filtro Unsharp');
%
% Se solicita al Usuario si desea Guardar (archivar), la Imagen en Escala de Grises
%
user_j=input('Requiere Guardar la Imagen en Escala de Grises? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(imagris,arch0);
else
    disp(' '); disp('No se ha Guardado Archivo en Escala de Grises');
end

%
% Se solicita al Usuario si desea Guardar cada Imagen resultante de la Aplicación
% de cada Filtro.
%
user_j=input('Requiere Guardar la Imagen Filtrada - Average ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(pphaver,arch1);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Average');
end

user_j=input('Requiere Guardar la Imagen Filtrada - Disk ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppdisk,arch2);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Disk');
end

user_j=input('Requiere Guardar la Imagen Filtrada - Gauss ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppgauss,arch3);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Gauss');
end

```

```

user_j=input('Requiere Guardar la Imagen Filtrada - Laplacian ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(pplaplac,arch4);
else
disp(' '); disp('No se ha Guardado Archivo Filtrado - Laplacian');
end

```

```

user_j=input('Requiere Guardar la Imagen Filtrada - Log ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(pplog,arch5);
else
disp(' '); disp('No se ha Guardado Archivo Filtrado - Log');
end

```

```

user_j=input('Requiere Guardar la Imagen Filtrada - Motion ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppmotion,arch6);
else
disp(' '); disp('No se ha Guardado Archivo Filtrado - Motion');
end

```

```

user_j=input('Requiere Guardar la Imagen Filtrada - Prewitt ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppprewitt,arch7);
else
disp(' '); disp('No se ha Guardado Archivo Filtrado - Prewitt');
end

```

```

user_j=input('Requiere Guardar la Imagen Filtrada - Sobel ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppsobel,arch8);
else
disp(' '); disp('No se ha Guardado Archivo Filtrado - Sobel');
end

```

```

user_j=input('Requiere Guardar la Imagen Filtrada - Unsharp ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppunsharp,arch9);
else
disp(' '); disp('No se ha Guardado Archivo Filtrado - Unsharp');
end

```

```

disp(' ');
disp('PROCESO FINALIZADO');

```

```
function guarda_archivo(imagen,archi00)
%
% Permite grabar la información del archivo 'imagen' como 'archi00')
%
if(exist(archi00))
    warndlg('El Archivo ya Existe');
    disp ('debe borrar antes el archivo..');
    disp (archi00);
else
    imwrite(imagen,archi00);
    mensaje = strcat ('Archivo.. ',archi00,' Grabado Satisfactoriamente');
    disp (mensaje);
end

end
```



### Relación de Funciones y comandos:

La relación de las principales funciones y comandos utilizados en el programa `Obtener_grises.m`, así como su descripción específica y ejemplos de aplicación, son los que se presentan a continuación:

**Tabla A.1: Principales Funciones y Comandos Utilizados**

Nombre	Descripción	Ejemplo
<code>imread</code>	La función <code>imread</code> se puede usar para leer imágenes. Si la imagen es “escala de grises” entonces <code>imread</code> devuelve una matriz bidimensional. Si la imagen es RGB entonces <code>imread</code> devuelve un arreglo tridimensional.	<code>a_input=imread(archivo)</code> <code>a_input=imread('figura.jpg')</code> <code>a =imread(filename,fmt)</code>
<code>rgb2gray</code>	La función <code>rgb2gray</code> convierte imágenes RGB a escala de grises, conservando la información de luminiscencia y eliminando la de tonalidad y saturación.	<code>imagris=rgb2gray(a_input)</code>
<code>subplot</code>	<code>subplot</code> crea un conjunto de subventanas en una pantalla, a efectos de poder mostrar una información en cada subventana. Los parámetros de la función <code>subplot(m,n,s)</code> , significa la creación de un arreglo de “m” por “n” subventanas, activando la ventana “s”.	<code>subplot(3,4,1)</code> <code>subplot (1,2,2)</code>
<code>imshow</code>	<code>imshow</code> permite presentar en la ventana o subventana activa la imagen correspondiente. En el formato <code>imshow(I,n)</code> , “n” representa el número de niveles de grises a considerar en la presentación gráfica, en caso de omitir dicho parámetro, la función <code>imshow</code> utiliza 256 niveles de grises.	<code>imshow (imagris)</code> <code>imshow (imagen,100)</code>
<code>fspecial</code>	La función <code>fspecial</code> permite crear “máscaras” o filtros bi-dimensionales predefinidos, <code>fspecial(tipo)</code> puede ser “gaussian” para filtro gaussiano pasa bajo, “sobel” para un filtro Sobel horizontal para enfatizar los bordes, “prewitt” para un filtro Prewitt horizontal para enfatizar los bordes, “laplacian” para un filtro aproximador al operador Laplaciano, “log” para un filtro Laplaciano Gaussiano, “average” para un filtro de promedio, “unsharp” para un filtro que mejora el contraste unsharp, entre otros.	<code>haver=fspecial('average');</code> <code>hgauss=fspecial('gaussian');</code>
<code>strcat</code>	La función <code>strcat</code> , permite concatenar cadenas de caracteres	<code>arch9 = strcat(archi,'_unsh',ext1)</code>
<code>imfilter</code>	La función <code>imfilter</code> permite aplicar un filtro multidimensional, <code>imfilter(a,h)</code> realiza el filtrado de “a” aplicando la máscara o filtro “h”.	<code>ppprewitt=imfilter(imagris,hprewitt)</code> <code>ppsobel=imfilter(imagris,hsobel)</code>
<code>input</code>	La función <code>input</code> permite imprimir un mensaje en la línea de comandos de MATLAB y recuperar como valor de retorno un valor numérico o el resultado de una expresión escrita por el usuario	<code>input('Requiere Guardar la Imagen en Escala de Grises? Si [s] o No [n] ','s');</code>
<code>if</code>	La sentencia <code>if</code> permite evaluar / ejecutar otras sentencias si los resultados lógicos son verdaderos o falsos, según sea la lógica del proceso.	<code>if user_j=='s'</code> <code>    guarda_archivo(pplog,arch5);</code> <code>else</code> <code>    disp(' '); disp('No se ha Guardado Archivo Filtrado - Log');</code> <code>end</code>

disp	La función <b>disp</b> permite imprimir en pantalla un mensaje de texto, o el valor de una matriz	disp('Proceso Terminado correctamnete') disp(H)
imwrite	La función <b>imwrite</b> permite grabar o escribir una imagen en un archivo gráfico (digital), <b>imwrite(imagen,archivo,fmt)</b> , donde <b>imagen</b> es la información a escribir, <b>archivo</b> es el nombre donde se grabara la información y <b>fmt</b> es el formato de salida (jpg, tif, etc). La imagen puede ser una en escala de grises ( <b>m por n</b> ) o a color ( <b>m por n por 3</b> )	imwrite(imagen,archi00); imwrite(imag,'out_fin','jpg')
guarda_archivo	La función <b>guarda_archivo</b> , es uan función escrita específicamente para esta evaluación, que permite grabar la información de la <b>imagen</b> con el nombre de archivo definido en <b>archi00</b> , utilizando para ello el siguiente formato: <b>guarda_archivo(imagen, archi00)</b> , parámetros que le son asignados al momento de invocar la función <b>guarda_archivo</b> .	guarda_archivo(imagen,archi00)

### Programa Obtener\_Grises\_tif.m

El listado completo del programa se muestra continuación:

```
function Obtener_grises_tif
clear all; clc; close all;
%
% Esta funcion convierte una imagen RGB a una imagen en Tonos de Grises.
% Se puede realizar con o sin la función de aplicación de filtros
% Pedro Arce Chirinos
% parce@perupetro.com.pe
%
% Presenta una ventana para poder seleccionar la imagen a procesar (leer)
%
buffer=pwd;
[archivo, path] = uigetfile('*.tif','Carga Imagen a Color - ext TIF'); cd(path);
% Obtencioan de las partes del archivo; Directorio, nombre, extensión y versión
[path1,archi,ext1,vers1] = fileparts(archivo);

a_input=imread(archivo); % Lee archivo que corresponde o contiene una imagen a color
cd(buffer);
imagris=rgb2gray(a_input); % Convierte archivo leído a Tonos de Grises
figure (1)
% Para presentar la Imagen Original en la parte izquierda de la Pantalla
subplot(1,2,1)
imshow (a_input); title ('Imagen Original');
% Para presentar la Imagen en Tonos de Grises en la parte derecha de la Pantalla
subplot (1,2,2)
imshow (imagris); title ('Imagen Tonos de Grises');
%
% Creacion de diversos filtros a aplicar a la imagen bidimensional
%
haver=fspecial('average');   hdisk=fspecial('disk');
hgauss=fspecial('gaussian'); hlaplac=fspecial('laplacian');
hlog=fspecial('log');       hmotion=fspecial('motion');
hprewitt=fspecial('prewitt'); hsobel=fspecial('sobel');
hunsharp=fspecial('unsharp');
%
% Definición del nombre de los archivos para la grabación de los mismos en el
% Directorio de Trabajo. El Directorio de Trabajo es el mismo donde se encuentra
% el Modelo o Programa
%
arch0 = strcat(archi,'_gris',ext1);   arch1 = strcat(archi,'_faver',ext1);
arch2 = strcat(archi,'_fdisk',ext1);  arch3 = strcat(archi,'_fgauss',ext1);
```

```

arch4 = strcat(archi,'_flapl',ext1); arch5 = strcat(archi,'_flog',ext1);
arch6 = strcat(archi,'_fmot',ext1); arch7 = strcat(archi,'_fprew',ext1);
arch8 = strcat(archi,'_fsobel',ext1); arch9 = strcat(archi,'_unsh',ext1);
%
% Creación de las imágenes con la aplicación de los Filtros definidos
%
pphaver=imfilter(imagris,haver); ppdisk=imfilter(imagris,hdisk);
ppgauss=imfilter(imagris,hgauss); pplaplac=imfilter(imagris,hlaplac);
pplog=imfilter(imagris,hlog); ppmotion=imfilter(imagris,hmotion);
ppprewitt=imfilter(imagris,hprewitt); ppsobel=imfilter(imagris,hsobel);
ppunsharp=imfilter(imagris,hunsharp);
%
% Creación de una nueva pantalla (Nº 2) para presentar la Siguiete Información:
%
% Imagen Original
% Imagen en Tonos de Grises
% Imagen Resultante por la aplicación de cada Filtro definido en el Programa
%
figure(2)
subplot(3,4,1)
subimage(a_input); title ('Imagen Original a Color');

subplot(3,4,2);
imshow(imagris); title ('Tonos de Grises');

subplot(3,4,3);
imshow (pphaver); title('F. Average');

subplot (3,4,4);
imshow(ppdisk); title('F. Disk');

subplot(3,4,5);
imshow(ppgauss); title('F. Gausiano');

subplot(3,4,6)
imshow(pplaplac); title('F. Laplace');

subplot(3,4,7)
imshow(pplog); title('F. Log');

subplot(3,4,8)
imshow(ppmotion); title('F. Motion');

subplot(3,4,9)
imshow(ppprewitt); title('F. Prewitt');

```

```

subplot(3,4,10)
imshow(ppsobel); title('F. Sobel');

subplot(3,4,11)
imshow(ppunsharp); title('F. Unsharp');

figure(3)
subplot(2,1,1);
subimage(imagris); title ('Imagen Escala de Grises');

subplot(2,1,2);
imshow(ppunsharp); title ('Imagen Escala de Grises - Filtro Unsharp');
%
% Se solicita al Usuario si desea Guardar (archivar),
% la Imagen en Escala de Grises
%
user_j=input('Requiere Guardar la Imagen en Escala de Grises? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(imagris,arch0);
else
    disp(' '); disp('No se ha Guardado Archivo en Escala de Grises');
end

%
% Se solicita al Usuario si desea Guardar cada Imagen resultante
% de la Aplicación de cada Filtro.
%
user_j=input('Requiere Guardar la Imagen Filtrada - Average ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(pphaver,arch1);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Average');
end

user_j=input('Requiere Guardar la Imagen Filtrada - Disk ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppdisk,arch2);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Disk');
end

user_j=input('Requiere Guardar la Imagen Filtrada - Gauss ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppgauss,arch3);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Gauss');
end

```

```
end
```

```
user_j=input('Requiere Guardar la Imagen Filtrada - Laplacian ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(pplaplac,arch4);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Laplacian');
end
```

```
user_j=input('Requiere Guardar la Imagen Filtrada - Log ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(pplog,arch5);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Log');
end
```

```
user_j=input('Requiere Guardar la Imagen Filtrada - Motion ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppmotion,arch6);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Motion');
end
```

```
user_j=input('Requiere Guardar la Imagen Filtrada - Prewitt ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppprewitt,arch7);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Prewitt');
end
```

```
user_j=input('Requiere Guardar la Imagen Filtrada - Sobel ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppsobel,arch8);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Sobel');
end
```

```
user_j=input('Requiere Guardar la Imagen Filtrada - Unsharp ? Si [s] o No [n] ','s');
if user_j=='s'
    guarda_archivo(ppunsharp,arch9);
else
    disp(' '); disp('No se ha Guardado Archivo Filtrado - Unsharp');
end
```

```
disp(' ');
disp('PROCESO FINALIZADO');
```

```
function guarda_archivo(imagen,archi00)
%
% Permite grabar la información del archivo 'imagen' como 'archi00')
%
if(exist(archi00))
    warndlg('El Archivo ya Existe');
    disp ('debe borrar antes el archivo..');
    disp (archi00);
else
    imwrite(imagen,archi00);
    mensaje = strcat ('Archivo.. ',archi00,' Grabado Satisfactoriamente');
    disp (mensaje);
end

end
```



## **ANEXO B**

## Programa Principal.m

El listado completo del programa se muestra continuación:

### Programa carga\_imagen.m

```
function carga_imagen
%
% Este programa lee o carga la imagen en Tonos de Grises
% Por default lee archivos tipo *.jpg, sin embargo puede leer
% otros de tipos de archivos como *.tif
%
global gris_imagen; % Imagen en Tono de Grises o Blanco y Negro
global val_min;    % Umbral del valor mínimo para asignar color
global val_max;    % Umbral del valor máximo para asignar color
global tipox;      % Tipo de colormap
delete *.mat;

clc;
clear all;
val_max = 1; val_min = -1;
disp(val_max); disp(val_min);

buffer=pwd;
[archivo, path_name] = uigetfile('*.*jpg','Lee / Carga Imagen Tono de Grises');
cd(path_name);
gris_imagen=imread(archivo); % Lee Imagen en Tonos de Grises
cd(buffer);

figure(2)
imshow(gris_imagen); % Muestra la Imagen Original
save gris_imagen;    % Almacena la Imagen en Tonos de Grises
save val_max;       % Almacena el umbral Valor Máximo
save val_min;       % Almacena el umbral Valor Mínimo
```

### Programa selec\_patron.m

```
function selec_patron
%
% Selec_patron, permite seleccionar el tipo de colormap a utilizar
% para asignar el color según el nivel de tono de gris en cada pixel
%
global tipox; % Tipo de colormap seleccionado

d={'hot','hsv','gray','pink','cool','bone','flag'};
```

```
[tipox,v1] = listdlg('PromptString','Seleccione un patron de color:',...
    'SelectionMode','single',...
    'ListString',d);
```

```
save tipox; % Almacena el Tipo de Colormap seleccionado
```

### Programa deter\_valmin.m

```
function deter_valmin
%
% Deter_valmin, permite modificar el valor mínimo utilizado para
% asignar el color según el patrón seleccionado por Selec_patron.m
%
% Según el tipo de colormap asignado, presenta la Imagen en color
% resultante a partir de la Imagen Original (tonos de grises) y el
% patrón (colormap) seleccionado
%
%
% Verifica que se haya leído (cargado) previamente una imagen
if(~exist('gris_imagen.mat'))
    a=warndlg('Debe leer una imagen en Tonos de Grises ');
end

% Verifica que se hay seleccionado un patrón (colormap)
if(~exist('tipox.mat'))
    a=warndlg('Seleccione Colormap');
end
clear all
global gris_imagen; % Imagen en Tonos de Grises o Blanco y Negro
global val_min; % Umbral - Valor máximo para el colormap
global val_max; % Umbral - Valor mínimo para el colormap
global tipox; % Tipo de colormap seleccionado

load gris_imagen; % Carga a Memoria la Imagen a Procesar
load val_min; % Carga a Memoria el Valor máximo - colormap
load val_max; % Carga a Memoria el Valor mínimo - colormap
load tipox; % Carga a Memoria el tipo de colormap seleccionado

% Reconoce el calor a asignr como valor mínimo, según desplazamiento del
% cursor o marca en la regleta
v=get(gcbo,'Value');
Hndl=findobj(gcbf,'Tag','text1');
str=sprintf('%2f',v);
set(Hndl,'String',str);
val_min = v; % Asigna como nuevo valor mínimo el reconocido de la pantalla
save val_min; % Almacena el valor mínimo
save val_max; % Almacena el valor máximo
```

```

% Genera la matriz [o] con valores menores a 1
o=double(gris_imagen)/256;

% Segun tipo de colormap seleccionado previamente, asigna el valor de
% la variable ctipo para el procesamiento posterior
ctipo='default';
if(tipox==1)
    ctipo='hot';
elseif(tipox==2)
    ctipo='hsv';
elseif(tipox==3)
    ctipo='gray';
elseif(tipox==4)
    ctipo='pink'
elseif(tipox==5)
    ctipo='cool';
elseif(tipox==6)
    ctipo='bone';
elseif(tipox==7)
    ctipo='flag';
end

% Verifica que el valor mínimo sea menor al valor máximo
% En caso de haber un error, emite un mensaje indicando tal error
% para que el usuario modifique alguno de ellos.
%
% En caso de ser correcto, muestra la imagen en pseudo-color
% según el patrón (colormap) seleccionado y los valores umbrales
% definidos por el usuario
%
if val_max >= val_min
    figure(3);
    colormap(ctipo);
    pcolor(o);
    h=pcolor(o);
    set(h,'LineStyle','none');
    disp(tipox)
    disp(ctipo)
    caxis([val_min val_max]);
    axis ij;
else
    a=warndlg('Error: Val_min debe ser MENOR que Val_max');
end

```

### Programa deter\_valmax.m

```

function deter_valmax
%
```

```

% Deter_valmax, permite modificar el valor máximo utilizado para
% asignar el color según el patrón seleccionado por Selec_patron.m
%
% Según el tipo de colormap asignado, presenta la Imagen en color
% resultante a partir de la Imagen Original (tonos de grises) y el
% patrón (colormap) seleccionado
%
%
% Verifica que se haya leído (cargado) previamente una imagen
if(~exist('gris_imagen.mat'))
    a=warndlg('Debe leer una imagen en Tonos de Grises max');
end

% Verifica que se hay seleccionado un patron (colormap)
if(~exist('tipox.mat'))
    a=warndlg('Selecione Colormap');
end
clear all
global gris_imagen; % Imagen en Tonos de Grises o Blanco y Negro
global val_min; % Umbral - Valor máximo para el colormap
global val_max; % Umbral - Valor mínimo para el colormap
global tipox; % Tipo de colormap seleccionado

load gris_imagen; % Carga a Memoria la Imagen a Procesar
load val_min; % Carga a Memoria el Valor máximo - colormap
load val_max; % Carga a Memoria el Valor mínimo - colormap
load tipox; % Carga a Memoria el tipo de colormap seleccionado

% Reconoce el calor a asignar como valor mínimo, según desplazamiento del
% cursor o marca en la regleta
v=get(gcbo,'Value');
Hndl=findobj(gcbf,'Tag','text5');
str=sprintf('%2f',v);
set(Hndl,'String',str);
val_max = v; % Asigna como nuevo valor máximo el reconocido de la pantalla
save val_max; % Almacena el valor máximo
save val_min; % Almacena el valor mínimo

% Genera la matriz [o] con valores menores a 1
o=double(gris_imagen)/256;

% Según tipo de colormap seleccionado previamente, asigna el valor de
% la variable ctipo para el procesameint posterior
ctipo='default';
if(tipox==1)
    ctipo='hot';
elseif(tipox==2)
    ctipo='hsv';
elseif(tipox==3)

```

```

ctipo='gray';
elseif(tipox==4)
    ctipo='pink'
elseif(tipox==5)
    ctipo='cool';
elseif(tipox==6)
    ctipo='bone';
elseif(tipox==7)
    ctipo='flag';
end

% Verifica que el valor máximo sea mayor o igual al valor mínimo
% En caso de haber un error, emite un mensaje indicando tal error
% para que el usuario modifique alguno de ellos.
%
% En caso de ser correcto, muestra la imagen en pseudo-color
% según el patrón (colormap) seleccionado y los valores umbrales
% definidos por el usuario
%
if val_max >= val_min
    figure(3);
    colormap(ctipo);
    pcolor(o);
    h=pcolor(o);
    set(h,'LineStyle','none');
    disp(tipox)
    disp(ctipo)
    caxis([val_min val_max]);
    axis ij;
else
    a=warndlg('Error: Val_max debe ser MAYOR que Val_min');
end

```

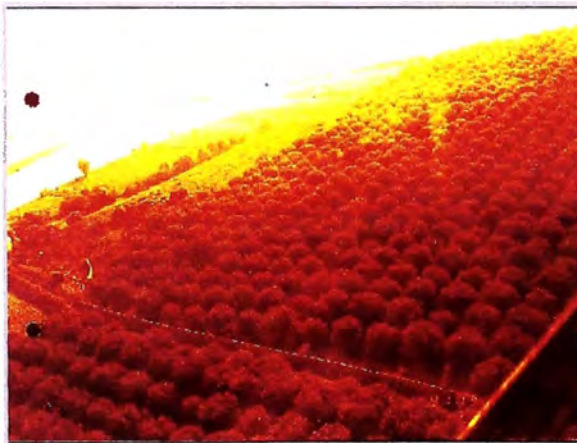
## Pseudo-color

A continuación se presentan los resultados de aplicar los diferentes patrones de color con un valor umbral máximo de 0.60 y 0.80, para los Casos “A” y “B”, respectivamente.:

### Caso “A”



**Figura N° B.1: Imagen Original – Caso “A”**

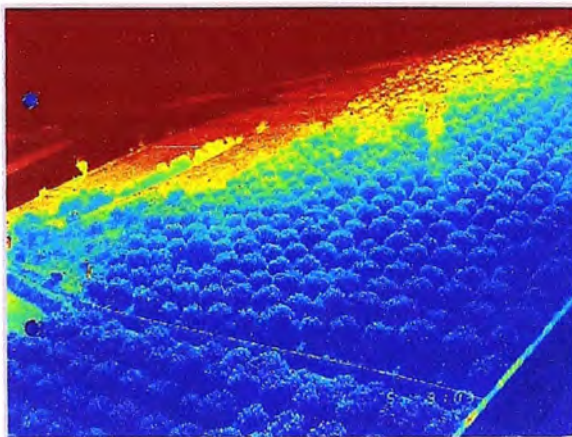


**Hot 0.00 – 0.60**

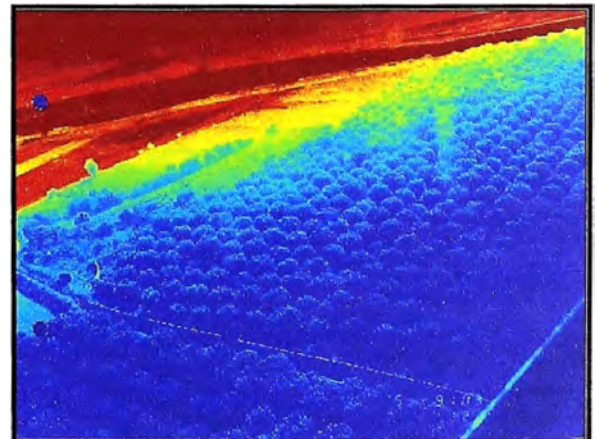


**Hot 0.00 – 0.80**

**Figura N° B.2: Imagen “hot” 0.00 – 0.80 – Caso “A”**



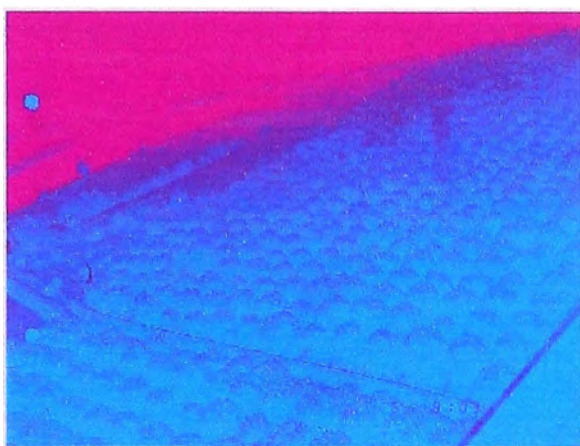
**Default 0.00 – 0.60**



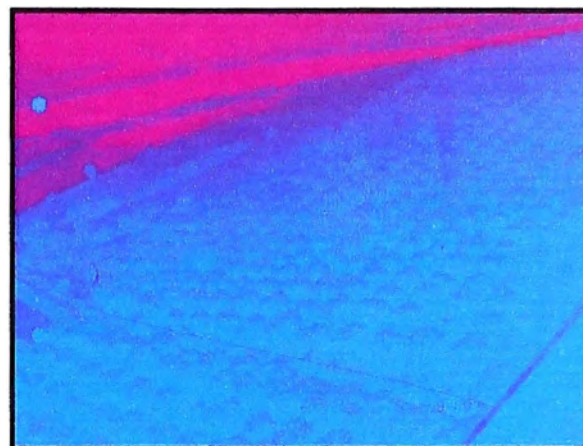
**Default 0.00 – 0.80**

**Figura N° B.3: Imagen “default” 0.00 – 0.80 – Caso “A”**





**Cool 0.00 – 0.60**



**Cool 0.00 – 0.80**

**Figura N° B.4: Imagen “cool” 0.00 – 0.80 – Caso “A”**

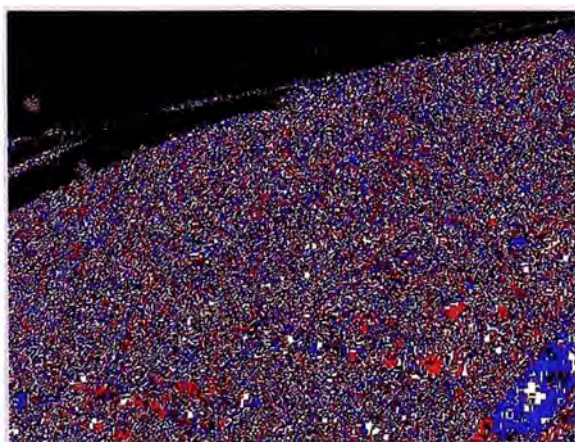


**Bone 0.00 – 0.60**

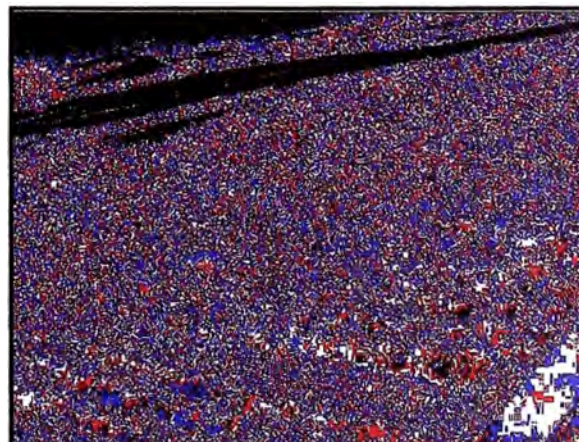


**Bone 0.00 – 0.80**

**Figura N° B.5: Imagen “bone” 0.00 – 0.80 – Caso “A”**



**Flag 0.00 – 0.60**



**Flag 0.00 – 0.80**

**Figura N° B.6: Imagen “flag” 0.00 – 0.80 – Caso “A”**



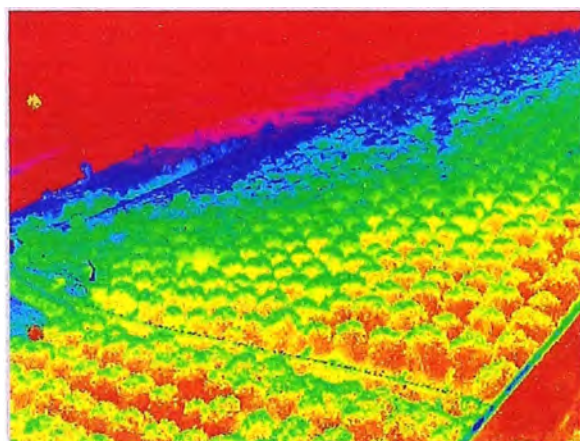


**Pink 0.00 – 0.60**

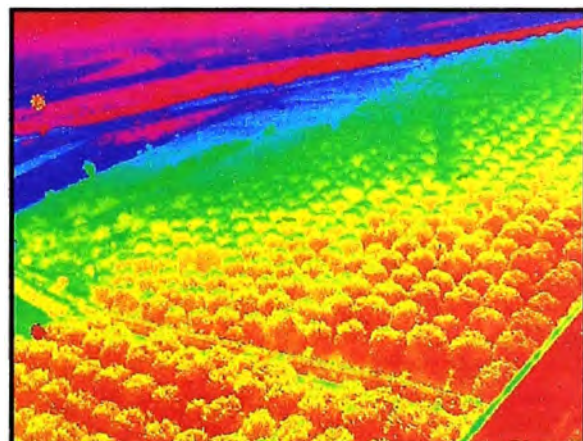


**Pink 0.00 – 0.80**

**Figura N° B.7: Imagen “pink” 0.00 – 0.80 – Caso “A”**



**Hsv 0.00 – 0.60**



**Hsv 0.00 – 0.80**

**Figura N° B.8: Imagen “hsv” 0.00 – 0.80 – Caso “A”**



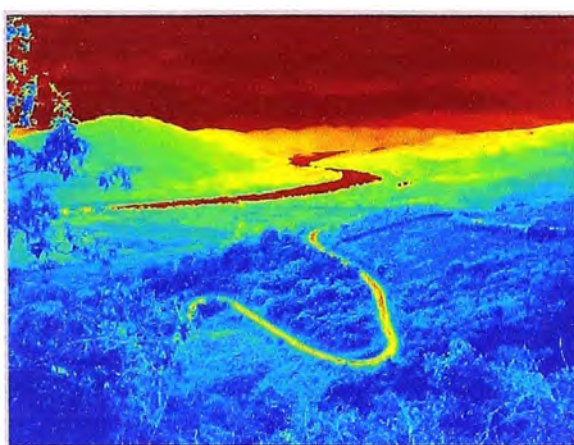
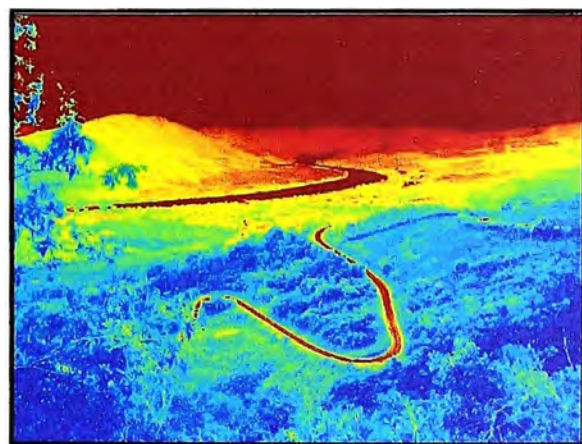
**Gris 0.00 – 0.60**



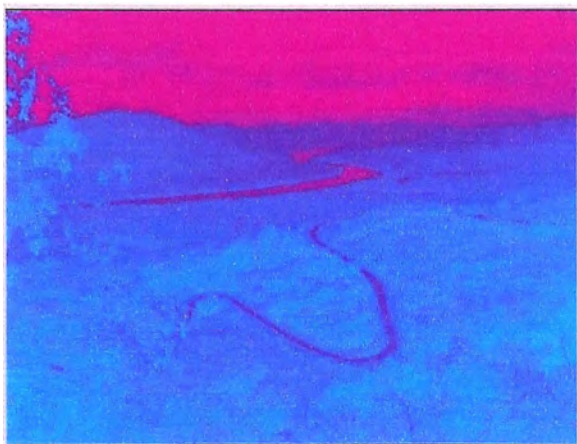
**Gris 0.00 – 0.80**

**Figura N° B.9: Imagen “gris” 0.00 – 0.80 – Caso “A”**

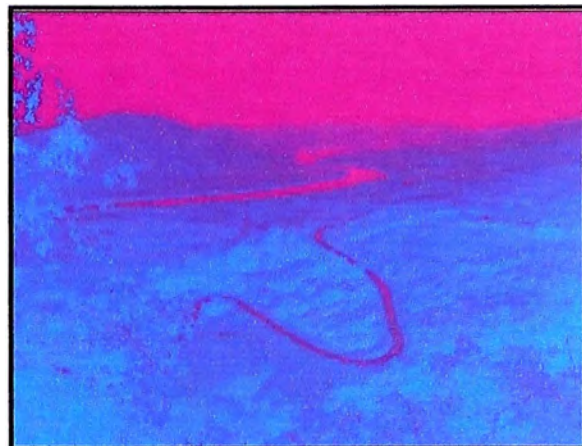


**Caso B:****Figura N° B.10: Imagen Original – Caso “B”****Hot 0.00 – 0.60****Hot 0.00 – 0.80****Figura N° B.11: Imagen “hot” 0.00 – 0.80 – Caso “B”****Default 0.00 – 0.60****Default 0.00 – 0.80****Figura N° B.12: Imagen “default” 0.00 – 0.80 – Caso “B”**





**Cool 0.00 – 0.60**



**Cool 0.00 – 0.80**

**Figura N° B.13: Imagen “cool” 0.00 – 0.80 – Caso “B”**

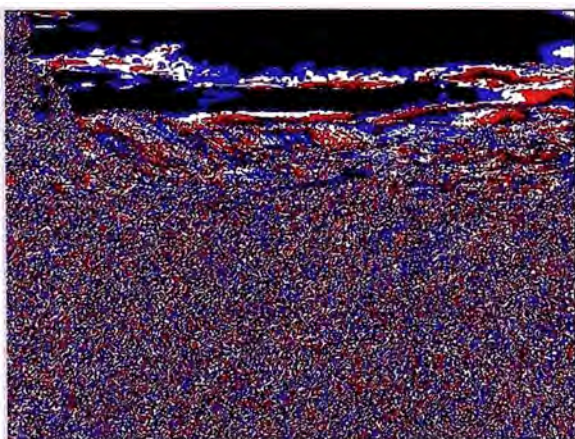


**Bone 0.00 – 0.60**

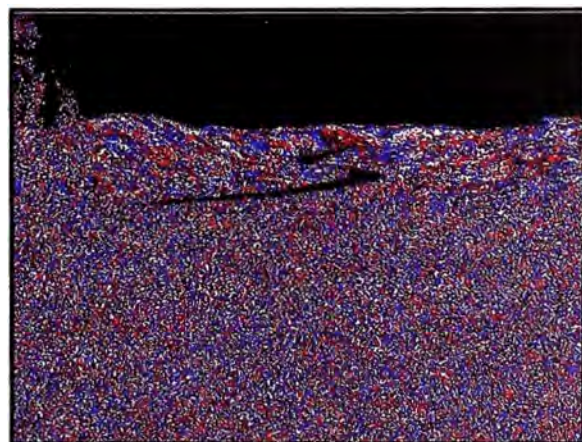


**Bone 0.00 – 0.80**

**Figura N° B.14: Imagen “bone” 0.00 – 0.80 – Caso “B”**



**Flag 0.00 – 0.60**



**Flag 0.00 – 0.80**

**Figura N° B.15: Imagen “flag” 0.00 – 0.80 – Caso “B”**



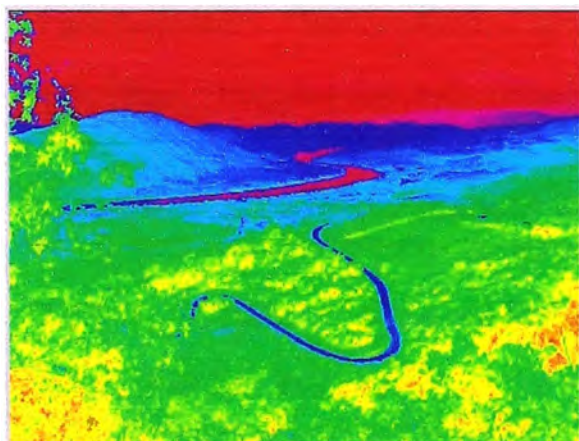


**Pink 0.00 – 0.60**

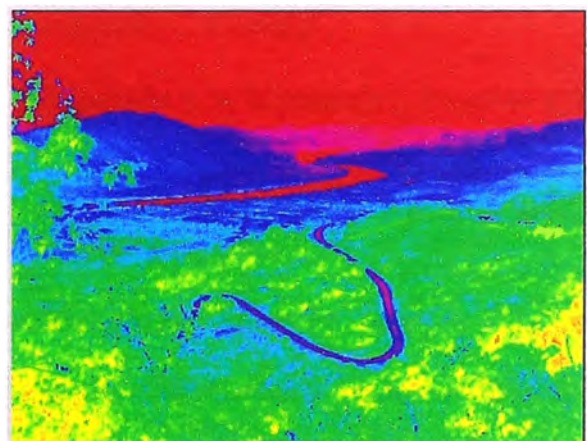


**Pink 0.00 – 0.80**

**Figura N° B.16: Imagen “pink” 0.00 – 0.80 - Caso “B”**



**Hsv 0.00 – 0.60**



**Hsv 0.00 – 0.80**

**Figura N° B.17: Imagen “hsv” 0.00 – 0.80 – Caso “B”**



**Gris 0.00 – 0.60**



**Gris 0.00 – 0.80**

**Figura N° B.18: Imagen “gris” 0.00 – 0.80 – Caso “B”**

## BIBLIOGRAFIA

1. Alan v. Oppenheim / Ronald W. Schafer , “Tratamiento de Señales en Tiempo Discreto”
2. Leon W. Couch II, “Sistemas de Comunicación Digitales y Analógicos”,
3. Arturo de la Escalera, “Visión por Computador – Fundamentos y Métodos”, Prentice Hall, 2001
4. Eligio Amthauer , “Procesamiento Digital de Imágenes (Tutorial)”
5. César Pérez, “Matlab y sus Aplicaciones en las Ciencia y la Ingeniería”, Prentice may, 2002.
6. Javier Garcia de Jalón / José Ignacio Rodríguez / Alfonso Brazales, “Aprenda Matlab 6.1”, Escuela Técnica Superior de Ingenieros Industriales, 2001.
7. H. Mehl – O. Peinado , “Fundamentos de Procesamiento Digital de Imágenes”,
8. Gonzalo Pajares / Jesús M. de la Cruz, “Visión por Computador – Imágenes Digitales y Aplicaciones”, Alfaomega Grupo Editor S. A. de C.V., 2002
9. Juan Alvaro Fernández Muñoz, “Estudio Comparativo de las Técnicas de Procesamiento Digital de Imágenes”, Facultad Informática – Universidad Politécnica de Madrid, 1999.
10. Gabriel Yuras, “Introducción a la Percepción Remota del Océano”, [gabriel@prof.c.udec.cl](mailto:gabriel@prof.c.udec.cl)
11. “Procesamiento Digital de Imágenes”, Facultad de Biología U.H – Curso TIC
12. Juan C. Ramírez, “Procesamiento de Imágenes (Parte II)”
13. José Luis Villareal Benítez / Rafael Montante López / Lizbeth Heras Lara, “Procesamiento Digital de Imágenes y Texto”, Dirección General de Servicios de Cómputo Académico, 2000