

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Notas de clase

Procesamiento y Análisis de Imágenes Digitales

CE-5201

Dr. José Pablo Alvarado Moya

Borrador de 5 de junio de 2010

Prefacio

Este corresponde al primer esbozo de notas de clase para el curso CE-5201 Procesamiento y Análisis de Imágenes Digitales, que es parte de la oferta de cursos electivos dentro de los planes de estudio de Ingeniería Electrónica e Ingeniería en Computadores del Instituto Tecnológico de Costa Rica. El curso se imparte por primera vez en el segundo semestre de 2009, y a partir de allí este documento ha evolucionado para incorporar las necesidades detectadas en el desarrollo de las actividades académicas.

Se asumen conocimientos matemáticos sobre señales y sistemas, particularmente sobre series y transformadas de Fourier en tiempo continuo, como base para el material presentado.

Los ejemplos y ejercicios hacen uso de las herramientas de código libre Octave [20] y la biblioteca en C++ LTI-Lib [1], que representan dos paradigmas diferentes de trabajo: la primera muy útil para el prototipado rápido, la segunda más apta para el desarrollo de prototipos de aplicaciones más complejas.

Dr. José Pablo Alvarado Moya

Cartago, 5 de junio de 2010

Reservados todos los derechos. Ni la totalidad ni parte de este documento pueden reproducirse, registrarse o transmitirse en ninguna forma ni por ningún medio sin permiso previo del autor.

© 2005-2008 P. Alvarado Escuela de Electrónica Instituto Tecnológico de Costa Rica

Este documento ha sido elaborado con software libre. Principalmente L^AT_EX, BibT_EX, GNUPlot, Octave, XFig, GNU-Make, LTI-Lib y Subversion en Linux

Índice general

Índice de tablas	v
Índice de ejemplos	vii
Revisar	ix
Lista de símbolos y abreviaciones	xi
1 Introducción	1
1.1 Áreas relacionadas con imágenes	1
1.2 Ejemplos de aplicaciones	5
1.3 Estructura del documento	5
2 Proceso de formación de imágenes	7
2.1 Fuentes de Energía	8
2.1.1 Energía acústica	8
2.1.2 Haces de partículas	14
2.1.3 Ondas electromagnéticas	14
2.2 Objetos y sus interacciones con la energía radiada	17
2.2.1 Interacciones de la energía irradiada con la materia	18
2.2.2 Propiedades relacionadas con interfaces y superficies	19
2.3 Configuración de la escena	22
2.4 Iluminación	22
2.5 Cámara	23
2.6 Estenoscopio y Proyección Central	24
2.7 Proyección con el modelo de lente gruesa	26
2.7.1 Profundidad de foco y de campo	27
2.8 Sistema de captura	28
2.9 El sistema visual humano	29
2.9.1 El ojo	29
2.10 Imágenes digitales y sus elementos	29
2.10.1 Representaciones de las imágenes	32
2.10.2 Resolución y tamaño de imágenes	34
2.10.3 Imágenes rasterizadas y vectoriales	34
2.11 Capturando imágenes con la LTI-Lib	36

2.11.1	Archivos	36
2.11.2	FireWire	36
2.11.3	USB	36
2.12	Problemas	37
3	Procesamiento en el dominio espacial	39
3.1	Operaciones y relaciones básicas	39
3.1.1	Vecindades	39
3.1.2	Operaciones aritméticas	40
3.1.3	Operaciones de conjuntos	41
3.1.4	Operaciones espaciales	41
3.1.5	Transformaciones de dominio	43
3.2	Transformaciones similares, afines y proyectivas	44
3.2.1	Transformación euclidiana	45
3.2.2	Transformación de similitud	46
3.2.3	Transformación afín	46
3.2.4	Transformación Proyectiva	46
3.3	Transformaciones de niveles de gris	47
3.3.1	Histogramas	48
3.3.2	Histogramas locales	49
3.4	Teoría de sistemas y filtrado espacial	49
3.4.1	Linealidad e Invarianza a la traslación	50
3.4.2	Convolución	50
3.4.3	Filtros lineales	53
3.4.4	Filtros de rango	55
3.4.5	Filtros IIR en imágenes	55
4	Procesamiento en el dominio de la frecuencia	57
4.1	Frecuencia en señales continuas y discretas	57
4.1.1	Señal sinusoidal continua	57
4.1.2	Señal sinusoidal discreta	59
4.1.3	Exponenciales complejas relacionadas armónicamente	63
4.1.4	Muestreo de señales analógicas	63
4.2	Análisis discreto de Fourier	66
4.2.1	Serie generalizada de Fourier	66
4.2.2	Serie discreta de Fourier	67
4.2.3	Transformada de Fourier de señales discretas	68
4.2.4	El teorema del muestreo	69
4.2.5	Propiedades de la transformada de Fourier de señales discretas	73
4.3	Transformada Discreta de Fourier	75
4.3.1	Muestreo en el dominio de la frecuencia	75
4.3.2	La transformada discreta de Fourier	79
4.3.3	Relación de la DFT con otras transformadas	80

4.3.4	Propiedades de la DFT	82
4.3.5	Filtrado lineal basado en la DFT	86
4.4	Análisis de Fourier en múltiples dimensiones	89
4.5	Análisis de Fourier en dos dimensiones	89
4.5.1	Aliasing en 2D	89
4.5.2	Patrones Moirè	91
4.5.3	Propiedades de la DFT en dos dimensiones	91
4.5.4	Filtrado en el dominio de la frecuencia	92
4.6	Tomografía	92
5	Procesamiento de imágenes en color	93
5.1	Características de color	95
5.2	Otros espacios de color	97
5.3	Algoritmos de procesamiento de color	97
5.3.1	Procesamiento de pseudo color	97
5.3.2	Procesamiento completo de color	97
6	Morfología	101
6.1	Fundamentos	101
6.2	Dilatación y Erosión	102
6.3	Apertura y Clausura	103
6.4	Transformación de añadir o eliminar	104
6.5	Algoritmos	104
6.5.1	Extracción Morfológica de Bordes	104
6.5.2	Extracción de componentes conectados	104
6.5.3	Adelgazamiento y ensanchado	105
6.5.4	Esqueleto	105
6.5.5	Transformación de distancia	105
6.6	Morfología con escalas de grises	105
6.6.1	Elementos planos	105
6.6.2	Elementos no planos	106
7	Algoritmos de análisis de imágenes	107
7.1	Detección de discontinuidades	107
7.1.1	Detección de Bordes	107
7.1.2	Detección de esquinas	110
7.1.3	Detección de líneas	110
7.2	Transformada de Hough para líneas	110
7.3	Segmentación	112
7.3.1	Métodos basados en el espacio de características	113
7.3.2	Métodos basados en el dominio de la imagen	120
7.3.3	Métodos híbridos	126
7.3.4	Anotaciones sobre la tarea de segmentación	129
7.4	Evaluation of Segmentation Algorithms	131

7.4.1	Evaluation Using the Pareto Front	134
7.4.2	Fitness Functions	135
7.5	Análisis de movimiento	139
7.5.1	Métodos de análisis de movimiento diferencial	139
7.5.2	Flujo Óptico	140
7.5.3	Rastreo por desplazamiento de medias	142
Bibliografía		145
A Respuestas a Problemas		153
Índice alfabético		155

Índice de tablas

2.1	Rapidez de propagación del sonido en función de la temperatura T en Celsius	9
2.2	Objetos y superficies idealizadas	20
2.3	Características de las señales	31
3.1	Transformaciones de niveles de gris.	47
4.1	Propiedades de la DFT	86
4.2	Análisis de Fourier en una dimensión	88
4.3	Análisis de Fourier en dos dimensiones	90
7.1	Comparación de técnicas de segmentación a nivel de imágenes	132

Índice de ejemplos

4.1	Frecuencias positivas y negativas.	58
-----	--	----

Revisar

Por hacer:.....	5
Por hacer:.....	16
Por hacer:.....	17
Por hacer:.....	21
Por hacer:.....	21
Por hacer:.....	23
Por hacer:.....	25
Por hacer:.....	28
Por hacer:.....	47
Por hacer:.....	48
Por hacer:.....	48
Por hacer:.....	49
Por hacer:.....	49
Por hacer:.....	49
Por hacer:.....	52
Por hacer:.....	53
Por hacer:.....	55
Por hacer:.....	89
Por hacer:.....	92
Por hacer:.....	100
Por hacer:.....	101
Por hacer:.....	102
Por hacer:.....	102
Por hacer:.....	102
Por hacer:.....	104
Por hacer:.....	114

Lista de símbolos y abreviaciones

Notación general

$\mathbb{N}^+, \mathbb{N}^*$	Conjunto de los números naturales sin cero $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$.
\mathbb{N}, \mathbb{N}_0	Conjunto de los números naturales $\mathbb{N} = \{0, 1, 2, \dots\}$.
\mathbb{Z}	Conjunto de los números enteros $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$.
\mathbb{Q}	Conjunto de los números racionales $\mathbb{Q} = \{q \mid q = \frac{n}{d}; n, d \in \mathbb{Z}\}$.
\mathbb{R}	Conjunto de los números reales.
\mathbb{C}	Conjunto de los números complejos.
$A \subset B$	A es subconjunto de B .
$A \cup B$	A unión B .
$A \cap B$	A intersección B .
$A \setminus B$	A menos B .
F_m	Frecuencia espacial de muestreo de una señal analógica. $F_m = 1/S$
F	Frecuencia en ciclos por metro para señales de variable continua.
f	Frecuencia en ciclos por muestra para señales de variable discreta.
j	$j^2 = -1$
$\circ \text{---} \bullet$	Mapeo de un dominio temporal al dominio frecuencial o z
$\bullet \text{---} \circ$	Mapeo de un dominio frecuencial (o z) al dominio temporal
(a, b)	Par ordenado con primer componente a y segundo componente b .
$\langle a, b \rangle$	Producto interno entre a y b
z^*	Complejo conjugado de z
$\angle z$ ó $\arg z$	Ángulo o argumento del número complejo z
$\text{Im}(z)$ ó z_{Im}	Parte imaginaria del número complejo z
$\text{Re}(z)$ ó z_{Re}	Parte real del número complejo z
S_p	Periodo fundamental $S_p = 1/F$ para señales de variable continua.
$\mathcal{T}[\cdot]$	Transformación realizada por un sistema
$\mathcal{F}\{\cdot\}$	Transformada de Fourier
$\mathcal{F}^{-1}\{\cdot\}$	Transformada inversa de Fourier
$\mathcal{L}\{\cdot\}$	Transformada de Laplace
$\mathcal{L}^{-1}\{\cdot\}$	Transformada inversa de Laplace
$\mathcal{Z}\{\cdot\}$	Transformada z
$\mathcal{Z}^{-1}\{\cdot\}$	Transformada inversa z
$x_a(s)$	Señal analógica sobre la dimensión espacial s .
y	Escalar.

A	Matriz.
\mathbf{A}	$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$
$x(n)$	Señal de variable discreta.
$\underline{\mathbf{x}}$	Vector.
	$\underline{\mathbf{x}} = [x_1 \ x_2 \ \dots \ x_n]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$
Ω	Frecuencia angular en radianes por metro para señales de variable continua.
ω	Frecuencia angular en radianes por muestra para señales de variable discreta.

Evaluation

$A_{\mathbf{u}}$	Algorithm A parameterized with \mathbf{u} .
$\widehat{\mathcal{P}}$	Pareto front.
$F(A_{\mathbf{u}}, \mathcal{G})$	Aggregate fitness function of algorithm A considering the reference data \mathcal{G} .
$f_{ce}(A_{\mathbf{u}}, \mathcal{G})$	Pixel-wise certainty.
$f_{ra}(A_{\mathbf{u}}, \mathcal{G})$	Region integrity.
$f_{pa}(A_{\mathbf{u}}, \mathcal{G})$	Pixel-wise potential accuracy.
$f_r(A_{\mathbf{u}}, \mathcal{G}_I)$	Mean normalized region size.
$f_{ra}(A_{\mathbf{u}}, \mathcal{G})$	Region-wise potential accuracy.
$f_{ri}(A_{\mathbf{u}}, \mathcal{G})$	Region-wise information content.
$f_t(A_{\mathbf{u}}, \mathcal{G}_I)$	Throughput of the algorithm $A_{\mathbf{u}}$ for the reference image set \mathcal{G}_I .
\mathcal{G}	Set of reference data $\{\langle \underline{\mathcal{I}}_i, \underline{\mathcal{S}}_i \rangle \mid i = 1 \dots n\}$.
$\underline{\mathcal{I}}_i$	Reference image used in the evaluation.
$\underline{\mathcal{S}}_i$	Ideal segmentation of the reference image $\underline{\mathcal{I}}_i$.

Abreviaciones

ACA	Adaptive Clustering Algorithm
DIP	<i>Digital Image Processing</i> .
DSP	<i>Digital Signal Processing (o Processor)</i> .
MAP	Maximum a-posteriori
MDD	Maximal Directional Derivative
MRF	Markov Random Field
PDS	Procesamiento Digital de Señales.
PID	Procesamiento de Imágenes Digitales.
ROC	Región de convergencia (<i>Region of Convergence</i>).

Capítulo 1

Introducción

Los sistemas perceptivos biológicos se han adaptado en el proceso evolutivo a los ambientes particulares de cada especie. Así, el sistema sonar de los delfines les permite construir *imágenes* de su entorno espacial bajo el agua, de modo similar al sistema de navegación en la oscuridad desarrollado por los quirópteros. Los primates capturan la información espacial principalmente a través de su sistema visual, que interpreta proyecciones ópticas sobre la retina de la luz reflejada por el entorno.

El vocablo *imagen* proviene del sustantivo latino *imāgo*, y éste a su vez del verbo *imitari* que da origen al verbo castellano *imitar*. De este modo, el término *imagen* está relacionado con la imitación de la realidad. El Diccionario de la Real Academia brinda como tercera acepción:

3. f. *Ópt.* Reproducción de la figura de un objeto por la combinación de los rayos de luz que proceden de él.

De este modo, las *imágenes* son representaciones de la realidad, y en el caso que compete al presente texto, son representaciones que pueden ser percibidas visualmente por personas.

1.1 Áreas relacionadas con imágenes

Históricamente la especie humana utilizó desde los albores de su civilización imágenes con fines ilustrativos, artísticos e incluso místicos, para pasar con el tiempo a brindarles otras funciones documentales (figura 1.1). En ciencia e ingeniería desde hace siglos las imágenes han formado parte de procesos de análisis *cualitativos*. El gran número de variables involucradas en el proceso de generación de las imágenes dificultó, hasta hace poco, su uso en procesos de análisis cuantitativo. El acelerado incremento en las capacidades de almacenamiento y cómputo ocurrido en las últimas dos décadas del Siglo XX hizo posible extender la aplicación de las imágenes en más áreas del quehacer humano, al hacerse posibles nuevas formas para su análisis.

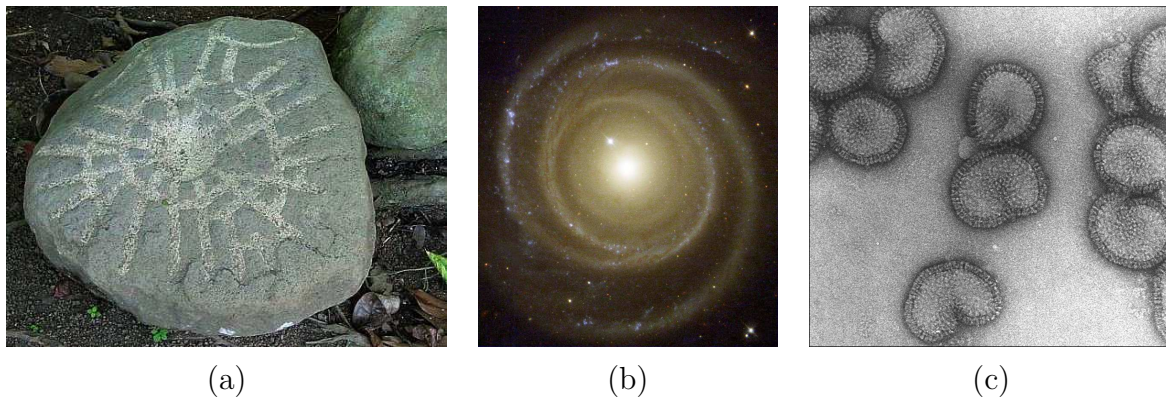


Figura 1.1: (a) Petroglifo encontrado en Guayabo, Costa Rica, representando un mapa estelar. (b) Imagen de una galaxia capturada por el telescopio Hubble de la NASA. (c) Imagen por microscopía electrónica de virus de influenza (Rob Ruigrok/ UVHCI).

El uso del computador para tratar imágenes se ha especializado en los últimos años, y se distinguen las siguientes áreas:

- Adquisición de imágenes (*Image Acquisition*)
- Procesamiento de imágenes (*Image Processing*)
- Análisis de imágenes (*Image Analysis*)
- Comprensión de imágenes (*Image Understanding*)
- Visión por computador (*Computer Vision*)
- Visión industrial (*Machine Vision*)

No existe consenso en la literatura técnica sobre dónde inicia y dónde termina cada una de estas áreas; sin embargo, sí es común colocar en la gama de especialidades al procesamiento de imágenes en un extremo y a la visión por computador en el otro. En la figura 1.2 se esbozan las áreas, enfatizando la diferencia entre ellas de acuerdo al tipo de información éstas producen.

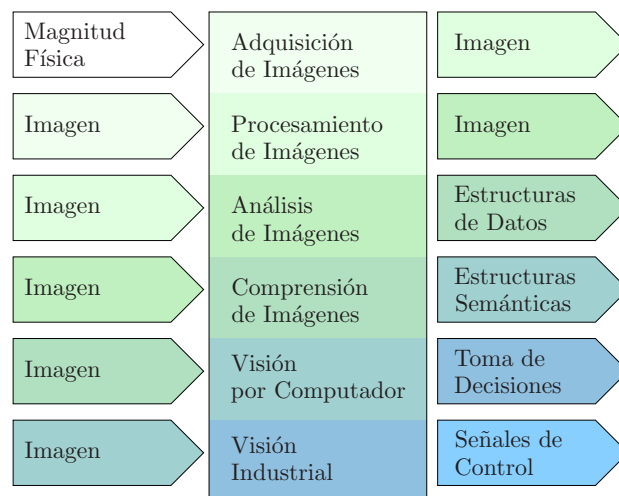


Figura 1.2: Gama de especialidades relacionadas con el tratamiento de imágenes.

La *adquisición de imágenes* busca cómo generar representaciones adecuadas del espacio físico a partir de la medición y análisis de alguna magnitud física concreta. Los avances de ésta área en los últimos años han sido notables, pues se han logrado depurar técnicas, particularmente en las áreas médicas, para generar imágenes a partir de fuentes de energía adicionales a la luz visible. La figura 1.3a muestra un ejemplo de reconstrucción

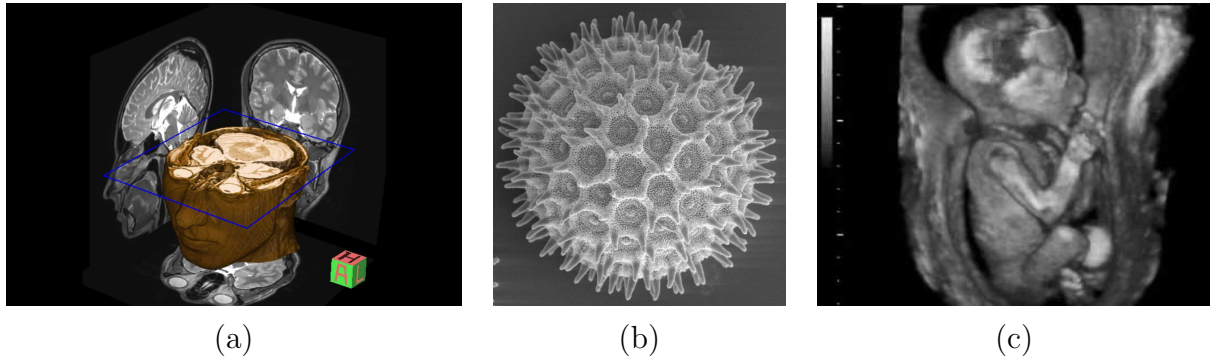


Figura 1.3: Imágenes generadas con diferentes técnicas. (a) Resonancia Magnética (Trio 3T Scanner, Universidad de Iowa). (b) Imagen de polen tomada con un microscopio electrónico (Dartmouth College, Rippel Electron Microscope Facility) (c) Ultrasonido de bebé de 18 semanas (D.M. Angel, Your Ultrasound)

del interior de la cabeza de un paciente a partir de imágenes captadas por resonancia magnética; la imagen mostrada es resultado del tratamiento de los datos medidos, los cuales no son directamente representables de forma visual. La figura 1.3b fue capturada con un microscopio electrónico, donde en la formación de la imagen los rayos de fotones son reemplazados por haces de electrones. Finalmente, la figura 1.3c presenta una imagen generada por medios ultrasónicos de un feto de 18 semanas. Los avances tecnológicos han permeado también la captura de imágenes cotidianas, a tal punto que en los teléfonos celulares se integran tecnologías para captura de imágenes de cada vez mejor calidad.

El *procesamiento o tratamiento digital de imágenes* consiste en procesos algorítmicos que transforman una imagen en otra en donde se resalta cierta información de interés, y/o se atenúa o elimina información irrelevante para la aplicación. Así, las tareas del procesamiento de imágenes comprenden la supresión de ruido, mejoramientos de contraste, eliminación de efectos no deseados en la captura como difuminaciones o distorsiones por efectos ópticos o de movimiento, mapeos geométricos, transformaciones de color, etc.

El *análisis de imágenes* pretende extraer información cuantitativa de las imágenes. La información se empaqueta en estructuras de datos concretas que varían desde simples números escalares (como en el caso del cálculo de magnitudes estadísticas de una imagen, incluyendo el valor medio o la desviación estandar de los niveles de gris), magnitudes vectoriales, matriciales o tensoriales (como el caso de histogramas), listas de puntos (denotando por ejemplo bordes, esquinas o líneas encontradas en la imagen), grafos denotando estructuras complejas, etc. Una subárea del análisis se concentra en cambios de representación de imágenes, lo que incluye los algoritmos de compresión, con y sin pérdida de información, así como los modernos algoritmos de cifrado que ocultan información de otra naturaleza

en las imágenes. Herramientas de software para *edición* de imágenes, tales como Gimp [74] o Adobe® Photoshop® [36], incorporan algoritmos para el procesamiento y análisis de imágenes.

La *comprensión de imágenes* pretende asignar significado al contenido de las imágenes, y una de las tareas clásicas para ésta área consiste en la segmentación completa de la imagen, que consiste en buscar una partición de la imagen en regiones correspondientes a objetos de la escena que la imagen captura. Algunos autores consideran a ésta tarea parte del análisis de imágenes, otros parte de la visión por computador.

La *visión por computador* es una rama de la inteligencia artificial que persigue emular al sistema visual humano, incorporando a los sistemas capacidades de aprendizaje, reconocimiento de patrones, inferencia y toma de decisiones basándose en el contenido de imágenes.

Finalmente, la *visión industrial* puede interpretarse como la aplicación de todas las otras áreas al contexto industrial, donde se utiliza en control de calidad, operaciones de conteo, captura y mejora de imágenes para revisión por operarios humanos, etc. Con frecuencia la salida de los procesos de visión industrial es alguna señal que activa mecanismos de clasificación de objetos, pero no se limita a ello. Los textos especializados en visión industrial prestan atención especial a los aspectos de formación de las imágenes, particulares para el contexto industrial.

Como se advirtió inicialmente, esta división de áreas no es compartida por todos los autores, pero brinda una idea de las capacidades de sistemas actuales y las tareas realizadas por medio del manejo de imágenes con medios electrónicos.

En la sección anterior se manejó el término *imagen* exclusivamente para indicar representaciones del dominio espacial. Si a éstas representaciones se les agrega la variable temporal para así poder representar además los cambios del espacio en el transcurso del tiempo, se obtienen señales de *vídeo*. Algunos autores incluyen a las señales de vídeo dentro de las especialidades anteriores, pero otros prefieren especificar por aparte al procesamiento y análisis de señales de vídeo.

Nótese que dentro del conjunto anterior no se incluyó al área de *gráficos por computador*, más relacionada con la adquisición de imágenes que con el resto. En principio, las áreas anteriores, excepto la primera, reciben como entradas imágenes (o vídeo) y procuran extraer información de ellas. El área de gráficos por computador va en la dirección contraria, y procura a partir de modelos matemáticos existentes que describen una escena, generar las imágenes correspondientes, simulando de cierto modo la captura de las imágenes. Parte de la adquisición de imágenes hace uso de las técnicas de su generación computarizada, como por ejemplo las imágenes de ultrasonido (figura 1.3c), o las imágenes por resonancia magnética (figura 1.3a).

Todas las áreas anteriores, incluyendo la generación gráficos por computador, junto a las tecnologías de visualización, constituyen el área conocida como visualística computacional. Ésta es a su vez es parte de la *visualística*, que incluye aspectos desde la psicología

cognitiva de la visión humana, la neurología de los sistemas visuales biológicos, hasta aspectos sociales y psicológicos relacionados con la comunicación visual, representaciones artísticas, etc.

1.2 Ejemplos de aplicaciones

Por hacer:

Elaborar

1. Análisis de imágenes satelitales
2. Análisis de imágenes médicas
3. Mediciones visuales industriales
4. Análisis de imágenes de microscopía

1.3 Estructura del documento

Este texto se concentra en el procesamiento y análisis de imágenes, particularmente para aquellas bidimensionales generadas con luz visible.

El capítulo 2 presenta los conceptos involucrados en la formación y la captura de las imágenes. El capítulo 3 trata conceptos del procesamiento de imágenes en el dominio espacial, incluyendo filtrados lineales y no lineales. El dominio de la frecuencia es utilizado en el capítulo 4 para introducir técnicas de procesamiento más eficientes de aplicar en un dominio alterno al espacial. La importancia del color para el sistema visual humano se refleja en las temáticas cubiertas en el capítulo 5. La morfología matemática es ampliamente utilizada en las áreas en cuestión, y es presentada en el capítulo 6. El documento concluye con una selección de algoritmos de análisis presentados en el capítulo 7, que pretenden brindar un muestreo no exhaustivo sobre las posibilidades del área.

Capítulo 2

Proceso de formación de imágenes

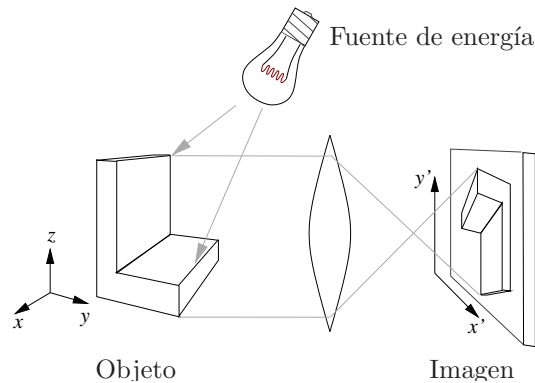


Figura 2.1: Formación de una imagen

La figura 2.1 muestra un esquema para representar la formación de una imagen. En el proceso intervienen cuatro elementos:

1. Una o varias fuentes de energía, que pueden ser de luz visible, rayos X, u otros tipos de ondas electromagnéticas, haces de partículas atómicas o subatómicas, ultrasonido, etc.
2. El objeto (u objetos) a ser capturado en la imagen. Estos objetos interactúan con la energía emitida, ya sea por reflexión, transmisión, refracción, absorción, difracción, etc.
3. La *cámara* es un sistema óptico o de otra índole que colecta la energía recibida y la proyecta al sistema de captura de la imagen.
4. El sistema de captura de la imagen transforma la señal proyectada a una representación apropiada al contexto; por ejemplo, la retina en el ojo humano transforma la proyección a impulsos neuronales, un chip CCD transforma la imagen a señales eléctricas aptas para su procesamiento en computador, o una película fotosensible captura la imagen de modo analógico.

Los cuatro componentes son el ámbito de estudio de la adquisición de imágenes mencionada anteriormente, y reciben particular atención en el área de visión industrial. Si

bien es cierto este documento se concentra en algoritmos que reciben como entrada la imagen capturada por el cuarto elemento, la llamada *configuración de escena* (es decir, las interacciones entre los cuatro elementos y las características de cada uno de ellos) es esencial en la determinación de la calidad de la imagen y la información rescatable por los algoritmos. La configuración de escena es determinante de la complejidad requerida en los sistemas de procesamiento y análisis posteriores.

2.1 Fuentes de Energía

Las fuentes de energía empleadas en la formación de imágenes debe permitir crear ondas de propagación que puedan interactuar con la materia que conforma la escena a observar. La interacción debe ser tal, que las ondas que finalmente alcancen al plano proyectivo de la imagen contengan *información* sobre el objeto irradiado. Tres fuentes de energía son utilizados con frecuencia en este contexto:

1. Energía acústica
2. Energía electromagnética
3. Energía cinética en haces de partículas

Las tres formas de energía comparten propiedades en el contexto de formación de imágenes, como por ejemplo la conformación de ondas con longitud de onda λ que determina el nivel de detalle distinguible en las imágenes. Como regla empírica, solo aquellas estructuras con tamaños mayores a la longitud de onda podrán ser capturados.

2.1.1 Energía acústica

Las ondas acústicas se propagan en un medio mecánico por medio de deformaciones elásticas. Así, para la construcción de imágenes este tipo de energía se utiliza en aplicaciones donde debe observarse el interior de objetos, lo que incluye al cuerpo humano, y materiales sólidos desde la madera hasta las estructuras de metal.

Por medio de presión isotrópica es posible generar ondas acústicas longitudinales que producen una compresión uniforme y por tanto deformaciones a lo largo de la dirección de propagación de la onda (figura 2.2). La densidad local ρ , la presión local p y la rapidez del sonido u se relacionan por medio de la ecuación de onda:

$$\frac{\partial^2 \rho}{\partial t^2} = u^2 \Delta \rho, \quad \frac{\partial^2 p}{\partial t^2} = u^2 \Delta p \quad \text{con} \quad u = \frac{1}{\sqrt{\rho_0 \beta_{ad}}} \quad (2.1)$$

Aquí, ρ_0 denota la densidad estática y β_{ab} es la compresibilidad adiabática, que corresponde al cambio relativo de volumen ante un cambio de presión homogéneo bajo la condición de que no ocurra intercambio térmico:

$$\beta_{ad} = -\frac{1}{V} \frac{dV}{dP}$$

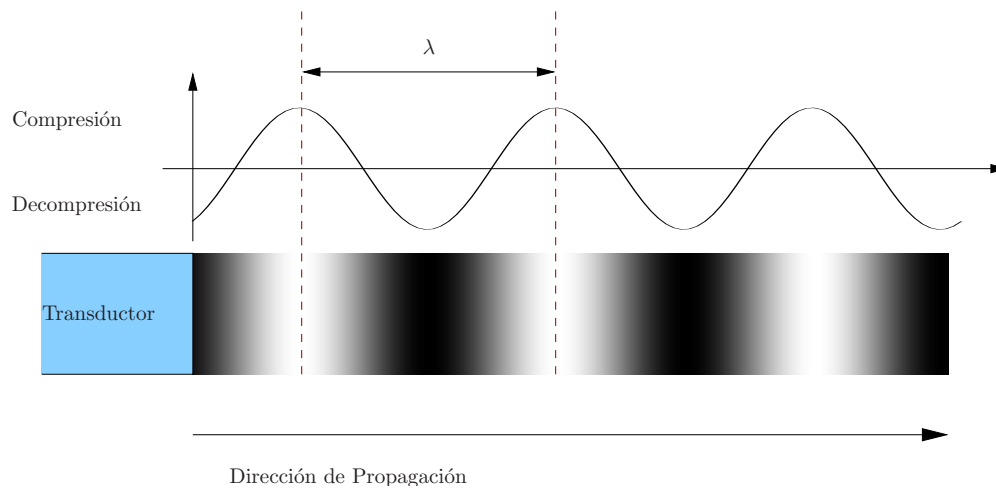


Figura 2.2: Propagación de una onda acústica.

Mientras menor sea la densidad y compresibilidad, mayor es la rapidez del sonido. La tabla 2.1 presenta algunos ejemplos de rapidez del sonido, donde para el caso particular de aire y agua se ha incluido la dependencia de la temperatura.

Tabla 2.1: Rapidez de propagación del sonido en función de la temperatura T en Celsius

Medio	Rapidez [m/s]
Aire ($0^{\circ}\text{C} < T < 20^{\circ}\text{C}$)	$331,3\sqrt{1 + \frac{T}{273,15^{\circ}\text{C}}}$
Agua ($0^{\circ}\text{C} < T < 100^{\circ}\text{C}$)	$1402,39 + 5,04T + 0,058T^2$
Sangre	1570
Grasa	1465
Músculos	1529-1580
Huesos	2650-4040

Considerando que la longitud de onda λ se puede calcular a partir de la frecuencia f y la rapidez de propagación de la onda u como

$$\lambda = \frac{u}{f}$$

se puede calcular que una onda acústica de 3 kHz en el aire tiene una longitud de onda de aproximadamente 10 cm.

En aplicaciones médicas se utilizan frecuencias entre 3 MHz y 10 MHz, tal y como se ilustra en la figura 2.3. Considerando la rapidez del sonido en el tejido humano, las longitudes de onda oscilan entre $500 \mu\text{m}$ y $150 \mu\text{m}$, respectivamente. La longitud de onda está relacionada con la *resolución* alcanzable, es decir, con el tamaño del mínimo detalle perceptible al sistema de captura, de modo que a mayor frecuencia, mayor detalle es posible capturar. Esto se debe a que para que ocurra reflexión el objeto en el que se refleje la onda debe tener un tamaño mayor que λ .



Figura 2.3: Rangos de sonido, según la frecuencia de la onda. El infrasonido tiene frecuencias subherzianas, el rango audible comprende desde 2 Hz hasta 20 kHz aproximadamente, y desde allí hasta el rango de los GHz se cubre el rango del ultrasonido, dentro del cual está el rango utilizado en aplicaciones médicas. El hipersonido supera las frecuencias de GHz.

Profundidad máxima observable en un sistema de ultrasonido

La figura 2.4 muestra el principio de funcionamiento de un sistema de visualización ultrasónico [66]. El transductor con apertura a emite pulsos de duración limitada Δt , los cuales, asumiendo una rapidez de propagación c , ocupan un ancho espacial $\Delta r = c\Delta t$. Los pulsos son emitidos regularmente por el transductor cada T segundos, lo que limita la ventana de tiempo en que pueden recibirse reflexiones de los pulsos cuando estos encuentran un cambio en la impedancia acústica del medio de propagación, y a su vez limita la profundidad r_{\max} que pueden penetrar los impulsos, dada por

$$r_{\max} = uT/2$$

Debe considerarse que, además del periodo T , la disipación energética que sufren los pulsos a lo largo de su trayectoria limitan la profundidad máxima alcanzable. Esta disipación energética, asociada con el fenómeno de absorción, es mayor mientras mayor sea la frecuencia del pulso, lo que implica que a mayor frecuencia, menor será la profundidad observable.

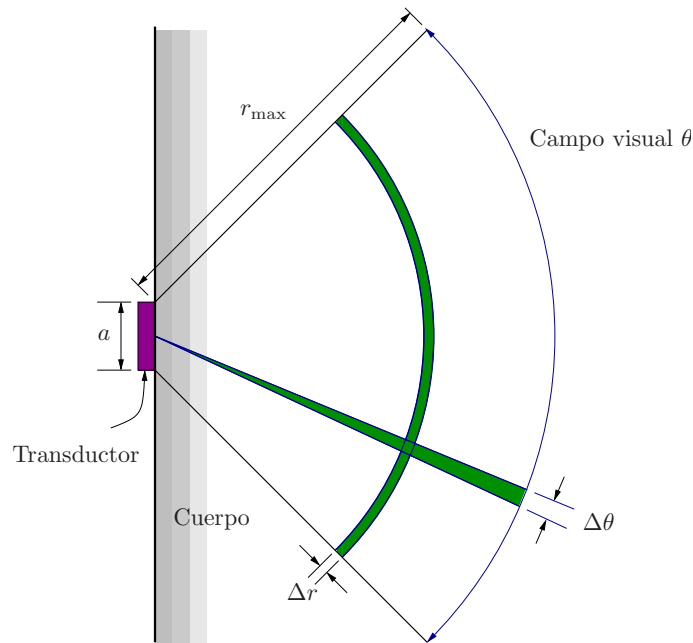


Figura 2.4: Principio de funcionamiento de un sistema de visualización ultrasónico.

Resolucion axial

Sea $r(t)$ un pulso rectangular dado por

$$r(t) = \begin{cases} 1 & \text{si } -1 \leq t \leq 1 \\ 0 & \text{en el resto} \end{cases}$$

con transformada de Fourier $\mathcal{F}\{r(t)\} = \text{sa } \omega$. El impulso acústico de duración Δt modulado con frecuencia f_c (figura 2.5) está dado por

$$p(t) = r\left(\frac{2}{\Delta t}t\right) \cos(2\pi f_c t) \quad (2.2)$$

y utilizando las propiedades de escalamiento y modulación, su Transformada de Fourier está dada con $\omega_c = 2\pi f_c$ por

$$P(j\omega) = \frac{\Delta t}{4} \left[\text{sa} \left(\frac{\Delta t}{2}(\omega + \omega_c) \right) + \text{sa} \left(\frac{\Delta t}{2}(\omega - \omega_c) \right) \right]$$

$$P(j2\pi f) = \frac{\Delta t}{4} [\text{sa}(\pi\Delta t(f + f_c)) + \text{sa}(\pi\Delta t(f - f_c))]$$

que se muestra en la figura 2.6. Nótese que la mayor cantidad de energía se concentra en el lóbulo principal de las componentes $\text{sa}(\cdot)$, y cada uno tiene un ancho de banda $B = 2/\Delta t$. De este modo, si se desea aumentar la resolución axial, debe aumentarse el ancho de banda para soportar pulsos de duración Δt suficientemente corta. Por otro lado, el ancho del pulso no puede ser inferior a $1/f_c$, puesto que de otro modo el pulso incluiría menos de un periodo de la señal portadora, por lo tanto

$$f_c \geq \frac{B}{2}$$

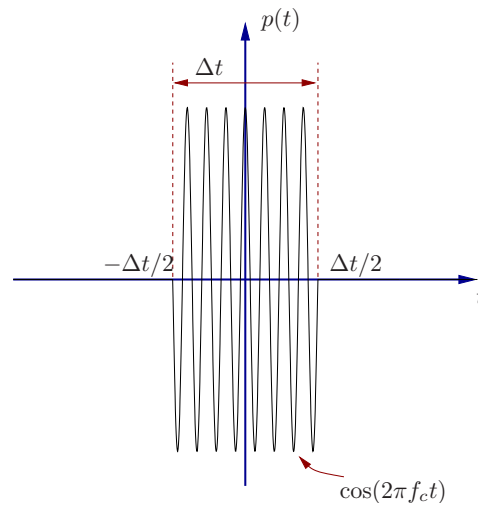


Figura 2.5: Pulso de duración Δt soportado por una frecuencia portadora f_c .

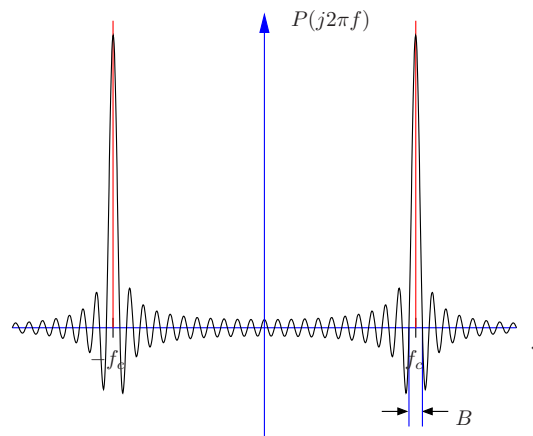


Figura 2.6: Espectro del pulso de duración Δt modulado a una frecuencia portadora f_c .

Resolución angular

Para determinar la resolución angular se parte de un modelo simplificado, en el que cada punto del transductor emite una onda de propagación esférica, y todos los puntos transmiten el mismo pulso modulado $p(t)$ especificado en (2.2), que se propaga sin atenuación con una rapidez constante c . Para la derivación, obsérvese la figura 2.7.

Para determinar la señal total $S(t)$ sobre el punto P se debe considerar el aporte de todos los puntos que constituyen el transductor, por medio de la integral

$$S(t) = \int_{-a/2}^{a/2} \cos \left[2\pi f_c \left(t - \frac{r(y)}{c} \right) \right] dy$$

donde el término $r(y)/c$ denota el tiempo que tarda el pulso emitido en el punto y en alcanzar al punto P . Considerando que el $\cos(\alpha) = \text{Re}\{e^{j\alpha}\}$, la linealidad del operador de integración, y que $r(y)$ se puede descomponer en una constante d correspondiente a la

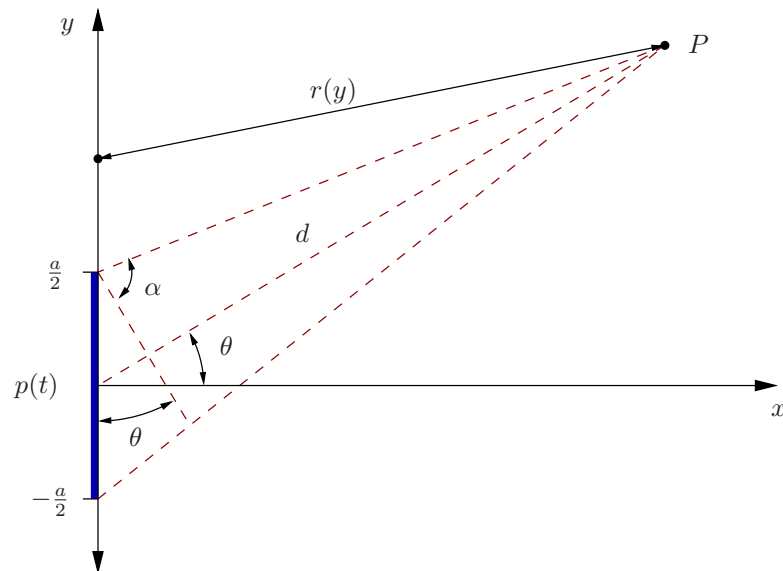


Figura 2.7: Diagrama para el cálculo del perfil de resolución angular.

distancia de P al origen más una función $\Delta r(y)$

$$\begin{aligned}
 S(t) &= \operatorname{Re} \left\{ \int_{-a/2}^{a/2} e^{-j2\pi f_c(t-r(y)/c)} dy \right\} \\
 &= \operatorname{Re} \left\{ e^{-j2\pi f_c t} \int_{-a/2}^{a/2} e^{j2\pi f_c r(y)/c} dy \right\} \\
 &= \operatorname{Re} \left\{ e^{-j2\pi f_c t} \int_{-a/2}^{a/2} e^{j2\pi f_c (d+\Delta r(y))/c} dy \right\} \\
 &= \operatorname{Re} \left\{ e^{-j2\pi f_c t} e^{j2\pi f_c d/c} \int_{-a/2}^{a/2} e^{j2\pi f_c (d+\Delta r(y))/c} dy \right\} \\
 &= \operatorname{Re} \left\{ e^{-j2\pi f_c (t+d/c)} \int_{-a/2}^{a/2} e^{j2\pi f_c (\Delta r(y))/c} dy \right\}
 \end{aligned}$$

El factor que multiplica a la integral solo depende del tiempo y de la distancia d del punto P al origen. La forma del patrón de señal sobre el arco con distancia d al origen depende entonces únicamente de la integral en la ecuación anterior. Si se asume que $d \gg a$, entonces $\alpha \approx 90^\circ$ y $\Delta r(y) \approx -y \sin \theta$, con lo que se tiene

$$\int_{-a/2}^{a/2} e^{j2\pi f_c (\Delta r(y))/c} dy \approx \int_{-a/2}^{a/2} e^{-j2\pi f_c y \sin \theta/c} dy = a \operatorname{sinc}[\pi f_c a \sin \theta/c]$$

que posee un lóbulo principal de ancho

$$\Delta\theta = 2 \operatorname{arcsen} \left(\frac{c}{af_c} \right)$$

de donde se deriva que la resolución angular mejora si se aumenta la apertura a o si se aumenta la frecuencia central f_c . En la práctica los lóbulos laterales introducen ruido en

la imagen, lo que se soluciona modificando el perfil de intensidad de las ondas acústicas sobre el transductor, de modo que en los extremos es menor que en el medio.

Para controlar la dirección donde se encuentra el máximo de la señal, se utilizan desfaces del pulso emitido sobre la superficie del transductor.

2.1.2 Haces de partículas

Los haces de partículas son flujos de partículas cargadas o neutrones que se desplazan a velocidades inferiores a la de la luz debido a que su masa en reposo no es cero. Estos haces pueden ser dirigidos por magnetos y enfocados con lentes electrostáticos. Para la conformación de imágenes el caso más relevante lo constituyen los haces de electrones, que conforman la radiación beta cuando son producidos por elementos radioactivos. En otras aplicaciones se encuentran haces de núcleos de átomos de hidrógeno o de helio.

Los haces de partículas se comportan como ondas, donde la longitud de onda λ y la frecuencia f están determinadas por la energía E y el momento m de la partícula, a través de la condición de frecuencia de Bohr:

$$f = E/h$$

y la relación de longitud de onda de de-Broglie:

$$\lambda = h/m$$

donde h es la constante de Planck ($h = 6,62606896 \times 10^{-34} \text{ J s}$).

Los haces de partículas se utilizan para la conformación de imágenes puesto que sus longitudes de onda son hasta en tres órdenes de magnitud menores que las longitudes de onda de la luz visible, lo que permite, en el caso de microscopía, alcanzar factores de amplificación en el rango de 2×10^6 en contraste al factor 2000 alcanzable con microscopía de luz.

2.1.3 Ondas electromagnéticas

Las ondas electromagnéticas están conformadas por campos eléctricos y magnéticos que oscilan de forma perpendicular, entre sí y con respecto a la dirección de propagación. Estas ondas pueden propagarse tanto en la materia como en el vacío, donde se propagan con la rapidez de la luz $c \approx 3 \times 10^8 \text{ m/s}$. Entre la longitud de onda λ y la frecuencia f existe la relación

$$\lambda f = c$$

El rango de frecuencias de ondas electromagnéticas abarca unas 24 décadas, y prácticamente todo el rango tiene aplicaciones en visualística. La figura 2.8 ilustra los nombres brindados a las distintas regiones espectrales.

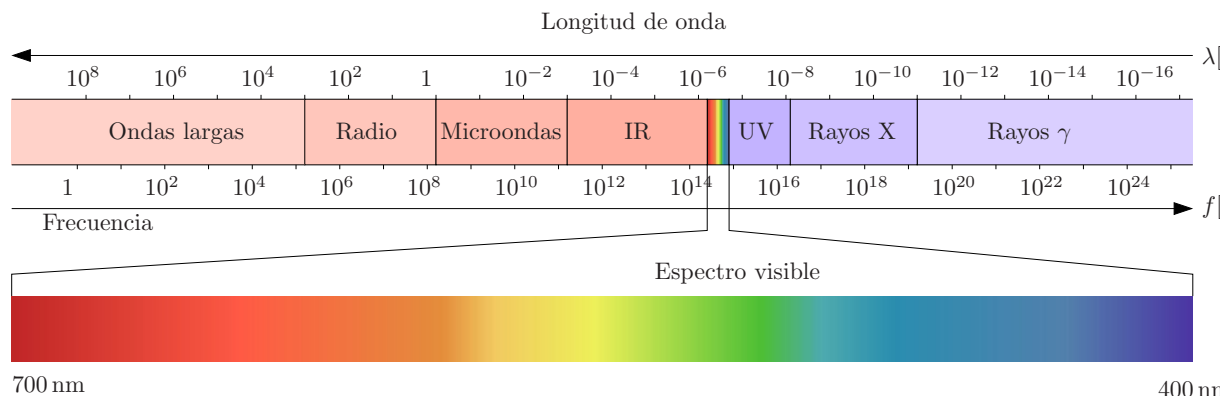


Figura 2.8: Espectro electromagnético.

Una fuente de ondas electromagnéticas se caracteriza por su distribución espectral de energía $C(\lambda)$, que especifica la tasa de energía que la fuente emite por intervalo unitario de longitud de onda λ . La potencia total en watts (W) emitida por una fuente es denominada *flujo irradiado* y se calcula como

$$P = \int_0^{\infty} C(\lambda) d\lambda$$

Cualquier cuerpo emite energía electromagnética en función de su temperatura. En física se utiliza el *cuerpo negro* para idealizar este efecto, y su distribución de energía espectral está dada por la Ley de Planck:

$$C(\lambda) = \frac{C_1}{\lambda^5 \left(\exp\left(\frac{C_2}{\lambda T}\right) - 1 \right)}$$

donde T es la temperatura del cuerpo en Kelvin, y C_1 y C_2 son constantes. Cerca del espectro visible la Ley de Planck se puede simplificar en la Ley de Radiación de Wien:

$$C(\lambda) = \frac{C_1}{\lambda^5 \exp\left(\frac{C_2}{\lambda T}\right)}$$

ilustrada en la figura 2.9. Obsérvese que alrededor de la temperatura del sol ($T = 6000$ K), la radiación cubre principalmente el espectro de luz visible, lo que podría explicarse con el hecho de que el sistema visual humano se ha adaptado para percibir la región del espectro electromagnético con mayor cantidad de energía en el medio. El modelo de cuerpo negro es utilizado para los irradiadores incandescentes con temperaturas en el rango entre $T = 1500$ K y 3500 K.

Otros principios de emisión son utilizados en los procesos de formación de imágenes, incluyendo la fluorescencia y los cada vez más utilizados LED (*Light Emitting Diode*).

En el proceso de formación de imágenes otras características de las fuentes de luz determinan intervienen en las cualidades reproducidas en las imágenes, incluyendo la polarización y la coherencia de la luz.

La polarización se define a través de la orientación del vector de campo eléctrico \underline{E} . Si el vector se confina en un plano, se habla de polarización lineal. Si se superponen dos

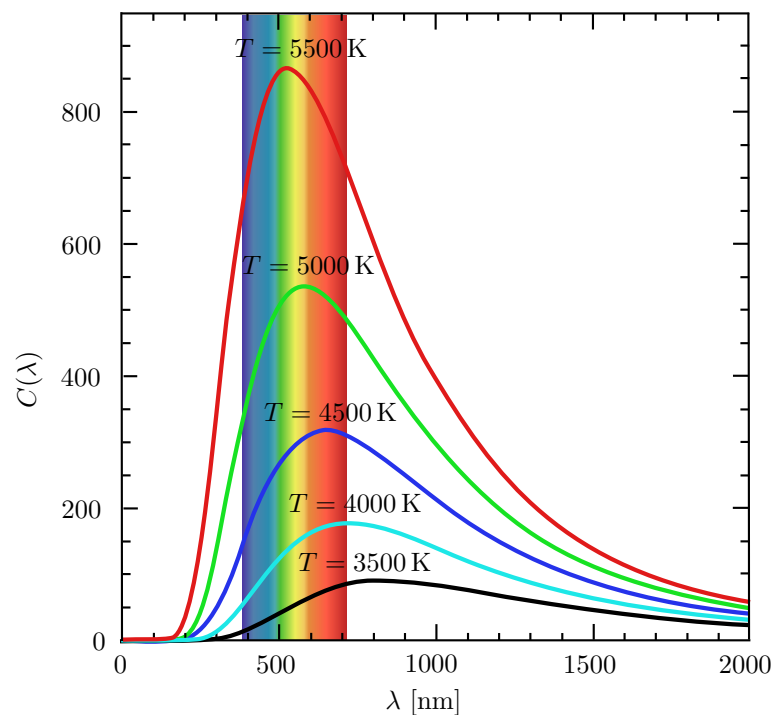


Figura 2.9: Ley de Wien para la emisión de un cuerpo negro.

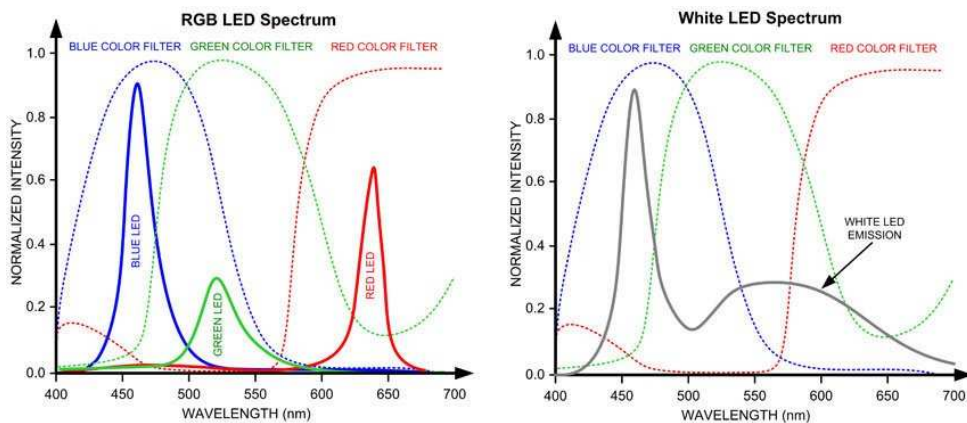


Figura 2.10: Composición espectral de radiación electromagnética emitida por LED.

frentes de onda que viajan en la misma dirección, cada uno de ellos confinado a un plano diferente, y los campos oscilan con fases diferentes, se obtiene polarización elíptica o circular, dependiendo si la fase es diferente o igual a 90° , respectivamente.

Por hacer:

Figura de polarización

En cuanto a la coherencia de la luz, esta se obtiene cuando existe una relación fija directa entre la fase de los vectores del campo electromagnético de todas las componentes de un frente de radiación, tal y como ocurre con la emisión estimulada de radiación utilizada en los laser. La coherencia permite construir interferencias constructivas y destructivas si se

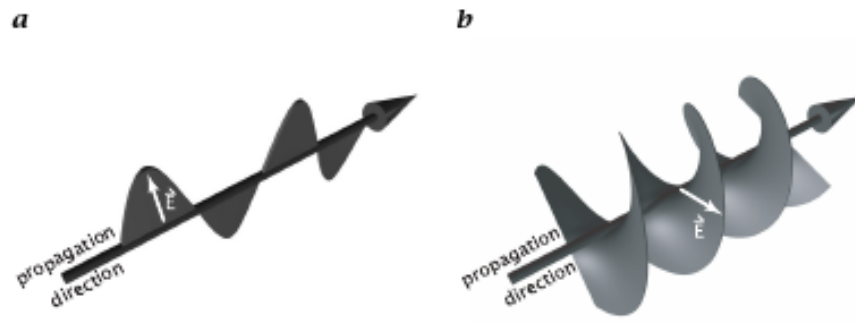


Figure 2.2: Illustration of **a** linear and **b** circular polarization of electromagnetic radiation. (By C. Garbe, University of Heidelberg.)

Figura 2.11: Polarización lineal y circular (tomado de Jähne 1999)

superponen las ondas.

Por hacer:

Figura de coherencia

Para el caso que compete, como fuentes de radiación lumínica se utilizan en ocasiones sistemas compuestos por fuentes de energía y materiales que interactúan con esta energía. Así, también las fuentes se pueden caracterizar por:

1. Orden de la iluminación. Un iluminador de primer orden produce las ondas electromagnéticas directamente (por ejemplo, un LED). Un iluminador de segundo orden redirecciona o modifica la energía irradiada por un iluminador de primer orden (por ejemplo, un iluminador basado en espejos, o un difuminador).
2. Fuente puntual o difusa. En el primer caso, el modelo de emisión se modela como un punto, de donde surgen todos rayos de luz que iluminan la escena; esto conduce a conformación de sombras fuertes. Las fuentes difusas producen luz desde un área, que se compone de infinitas fuentes de rayos emitidos en diferentes direcciones lo que produce una iluminación homogénea que difumina las sombras.

2.2 Objetos y sus interacciones con la energía radiada

La información utilizada para la conformación de las imágenes es aportada precisamente por la interacción que tiene la energía irradiada con los objetos a ser capturados. El conocer en detalle los principios físicos involucrados permite elegir con fundamento aquellas configuraciones de escena que permiten resaltar los detalles de interés en los objetos.

A todo el proceso desde la emisión de la luz, hasta su arribo a la cámara se le conoce como la cadena radiométrica de la formación de imágenes, que se ilustra en la figura 2.12.

Propiedades como la profundidad de penetración o la reflectividad superficial, que dependen de las longitudes de onda de la energía irradiada, determinan las modificaciones que

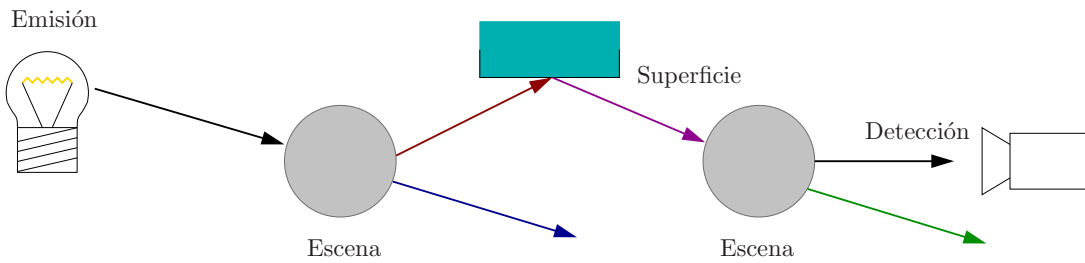


Figura 2.12: Cadena radiométrica en la formación de imágenes.

sufre dicha energía en su ruta hasta el detector.

2.2.1 Interacciones de la energía irradiada con la materia

La radiación que incide o que atraviesa los objetos en una escena sufre modificaciones en su interacción con la materia. Estas modificaciones incluyen cambios en su dirección de propagación, atenuación, amplificación, cambios en la composición espectral, o incluso polarización.

Algunas definiciones son pertinentes:

- El *flujo radiante* Φ se define como la potencia total emitida por una fuente o recibida por un detector, medida en watts [W].
- La *excitancia radiante* M es la potencia emitida por unidad de área, y se mide en $[\text{W}/\text{m}^2]$.
- La *irradiancia* es la potencia recibida por unidad de área y se mide en $[\text{W}/\text{m}^2]$.

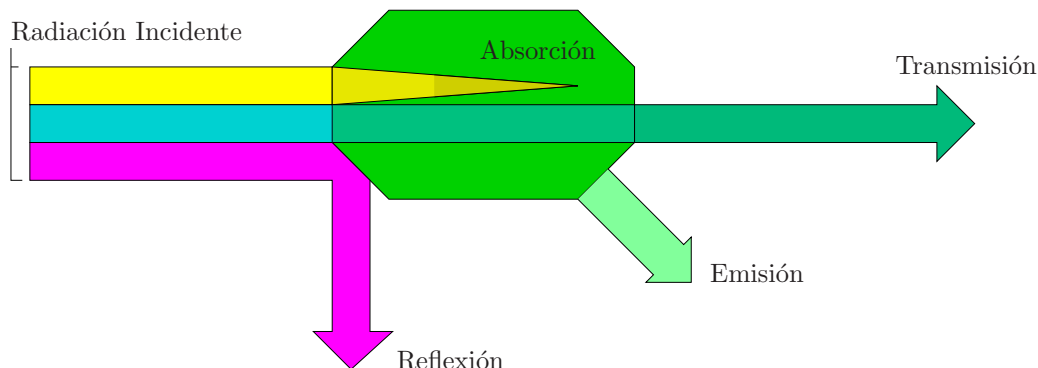


Figura 2.13: Flujo radiante Φ_i incidente en un objeto es parcialmente reflejado, absorbido, transferido o incluso emitido, para mantener el equilibrio termodinámico.

Para caracterizar las propiedades ópticas de superficies y objetos se definen entonces las siguientes magnitudes adimensionales:

Reflectividad $\tilde{\rho}$ (o reflectancia) se define como la tasa de flujo radiante reflejado Φ_r contra el flujo radiante incidente Φ_i :

$$\tilde{\rho} = \frac{\Phi_r}{\Phi_i}$$

Absortividad $\tilde{\alpha}$ (o absortancia) es la razón entre el flujo radiante absorbido Φ_a y el flujo radiante incidente Φ_i .

$$\tilde{\alpha} = \frac{\Phi_a}{\Phi_i}$$

Transmisividad $\tilde{\tau}$ (o transmitancia) describe cuanto del flujo radiante incidente Φ_i se mantiene en como flujo radiante transmitido Φ_t .

$$\tilde{\tau} = \frac{\Phi_t}{\Phi_i}$$

Emisividad $\tilde{\epsilon}$ (o emitancia) cuantifica el desempeño de un objeto activamente radiante comparado con un cuerpo negro, y se define como la razón entre las exitancias

$$\tilde{\epsilon} = \frac{M_e(T)}{M_{cn}(T)}$$

donde M_e denota la exitancia de la fuente emisora y M_{cn} la exitancia del cuerpo negro a una temperatura T . Esta magnitud $\tilde{\epsilon}$ es siempre menor que uno puesto que el modelo de cuerpo negro describe la exitancia máxima de cualquier objeto puede exhibir a una temperatura T .

Nótese que la reflectividad $\tilde{\rho}$, absortividad $\tilde{\alpha}$ y transmisividad $\tilde{\tau}$ definen las modificaciones que puede realizar un receptor *pasivo* al flujo radiante incidente Φ_i , mientras que la emisividad $\tilde{\epsilon}$ es una característica de objetos activamente radiantes. Todas estas magnitudes dependen de la dirección, longitud de onda y polarización de la energía incidente.

La emisión, transmisión, reflexión y absorción de radiación se pueden referir tanto a interacciones con superficies (o interfaces entre objetos), como al efecto neto sobre objetos de espesor finito. Varios autores coinciden en utilizar los términos con sufijo *-ividad* para las propiedades superficiales (o intrínsecas) y el sufijo *-ancia* para las propiedades para las propiedades volumétricas (o extrínsecas) de los objetos.

Si un objeto se encuentra en equilibrio termodinámico con su entorno, por la ley de conservación de la energía se debe cumplir:

$$\Phi_i = \Phi_a + \Phi_r + \Phi_t$$

y por tanto

$$\tilde{\alpha} + \tilde{\rho} + \tilde{\tau} = 1$$

lo que permite definir objetos idealizados (tabla 2.2).

2.2.2 Propiedades relacionadas con interfaces y superficies

Se define interfaz como una discontinuidad en la propiedades ópticas en una distancia mucho menor que la longitud de onda de la radiación.

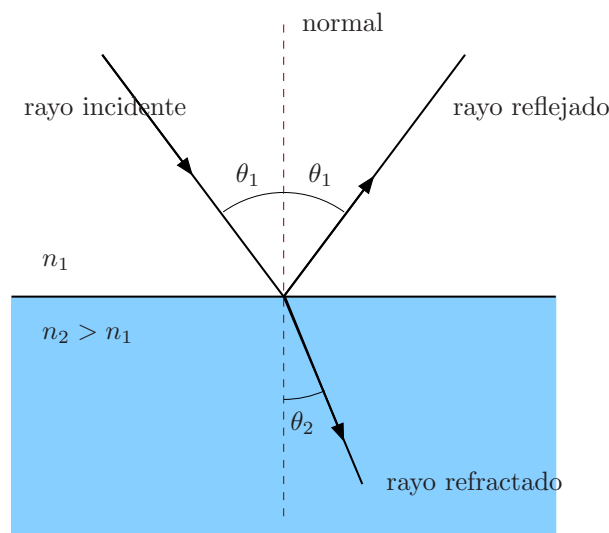
Tabla 2.2: Objetos y superficies idealizadas

Objeto	Propiedad	Descripción
Cuerpo opaco	$\tilde{\epsilon}(\lambda) + \tilde{\rho}(\lambda) = 1$ $\tilde{\tau}(\lambda) = 0$	No puede ser penetrado por radiación. Toda radiación es emitida o reflejada.
Ventana ideal	$\tilde{\epsilon}(\lambda) = \tilde{\rho}(\lambda) = 0$ $\tilde{\tau}(\lambda) = 1$	Toda la energía es transmitida sin atenuación y no hay emisión.
Espejo	$\tilde{\epsilon}(\lambda) = \tilde{\tau}(\lambda) = 0$ $\tilde{\rho}(\lambda) = 1$	Toda la energía es reflejada y no hay emisión.
Cuerpo negro	$\tilde{\rho}(\lambda) = \tilde{\tau}(\lambda) = 0$ $\tilde{\epsilon}(\lambda) = 1$	Toda la energía es absorbida y tiene la exitancia máxima posible.

Refracción

El índice de refracción en óptica geométrica es la razón entre la velocidad de la luz en el vacío contra la velocidad de la luz en el medio bajo consideración, y determina el cambio en la dirección de propagación de la radiación atravesando la interfaz entre dos materiales con propiedades dieléctricas diferentes. De acuerdo a la Ley de Snell, los ángulos de incidencia θ_1 y refracción θ_2 se relacionan con (figura 2.14)

$$\frac{\sin \theta_1}{\sin \theta_2} = \frac{n_2}{n_1} \quad (2.3)$$

**Figura 2.14:** Ley de Snell para la refracción y reflexión en interfaces.

Reflección especular

Los rayos incidentes, reflejados y la normal a la superficie se encuentran sobre un plano perpendicular a la superficie. En superficies suaves entre dos materiales con propiedades dieléctricas propias, ocurre reflexión especular. Aquí, los ángulos de incidencia y reflexión son iguales (figura 2.14).

Reflección difusa

Cuando asperezas en las superficies tienen tamaños en el mismo orden de magnitud que las longitudes de onda de la radiación incidente, ocurren fenómenos de difracción en las microestructuras. Si las microestructuras son relativamente grandes, los rayos son reflejados en distintas direcciones. Si la distribución de las reflexiones es isotrópica, entonces se habla de superficies de Lambert, pero las microestructuras pueden orientar la luz parcialmente, alrededor de la dirección de la reflexión especular.

Una combinación de reflexión especular y difusa puede ser causada por fenómenos de dispersión subsuperficial (figura 2.15).

Por hacer:

Agregar figura con las dos reflexiones, tanto con rayos y superficies, así como el ejemplo de los tipos de reflexión en distintos materiales



Figure 3.6: *a* Specular, *b* diffuse, *c* and subsurface reflection at interfaces.

Figura 2.15: Reflexión especular, difusa y subsuperficial (tomado de Jähne 1999)

Propiedades intramateriales

Propiedades intramateriales: atenuación (absorción y dispersión), emisión volumétrica

Atenuación es determinante para la determinación de la profundidad visible, por lo que ultrasonido, si bien es cierto más frecuencia implica más resolución, la profundidad visible es menor (depth)

Por hacer:

Elaborar ¿por qué el cielo es azul?



Figure 3.9: Spheres shaded using the Phong illumination model: a ambient reflection, b diffuse reflection, and c specular reflection. (By C. Garbe, University of Heidelberg.)

Figura 2.16: Reflexión ambiental, difusa y subsuperficial (tomado de Jähne 1999)

2.3 Configuración de la escena

2.4 Iluminación

Existen varias configuraciones para colocar la iluminación con respecto a los objetos a ser observados. En aplicaciones de visión industrial se acostumbran las configuraciones mostradas en la figura 2.17.

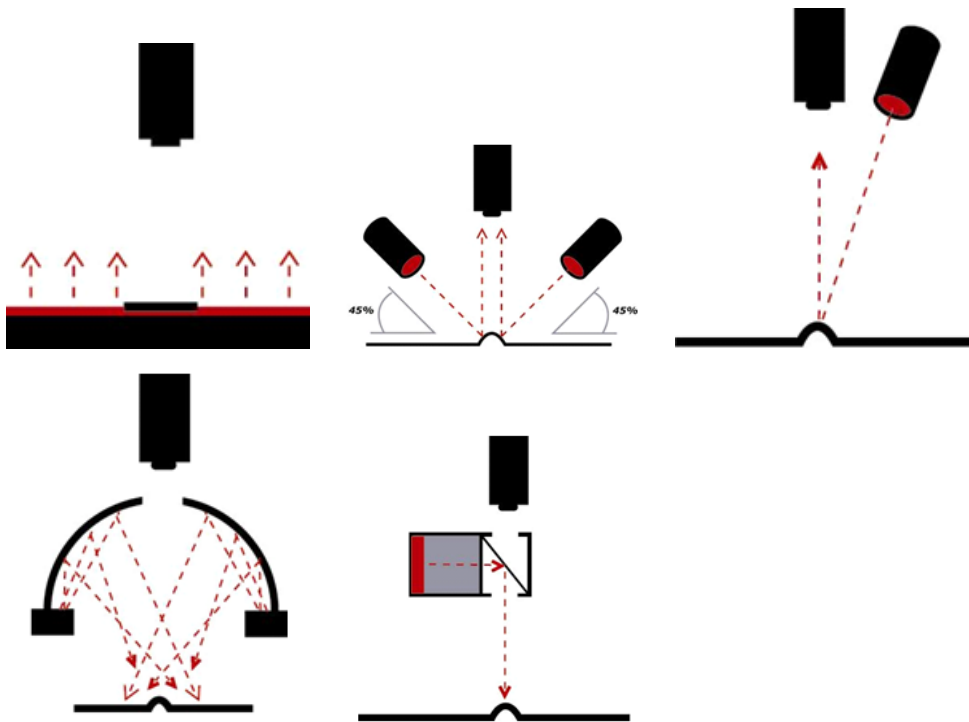


Figura 2.17: Configuraciones de iluminación (tomado de <http://zone.ni.com/devzone/cda/tut/p/id/6903>)

Por hacer:

Agregar información de Jähne, HCV 1, 6.4, pp. 157

2.5 Cámara

La posición de objetos en el espacio tridimensional pueden ser descritas desde dos perspectivas. Las coordenadas universales son un sistema referencial ubicado directamente en la escena observada. Un punto en este sistema de referencias se denotará aquí como

$$\underline{\mathbf{x}}' = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix}$$

Las coordenadas x'_1 y x'_2 engendran un plano horizontal y x'_3 denota una posición vertical. El segundo sistema de referencia es conocido como coordenadas de cámara

$$\underline{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

que están fijos al plano de proyección de la cámara que captura la escena. El eje x_3 se alinea con el eje óptico de la cámara, x_1 se alinea con el eje horizontal de la imagen y x_2 con el eje vertical. La conversión de un sistema coordenado a otro se realiza trasladando orígenes por el vector $\underline{\mathbf{t}}$, y rotando los sistemas coordenados, lo que se puede expresar por medio de la matriz ortonormal \mathbf{R} utilizando operaciones lineales de la forma:

$$\underline{\mathbf{x}} = \mathbf{R}(\underline{\mathbf{x}}' - \underline{\mathbf{t}}) \quad (2.4)$$

$$\underline{\mathbf{x}}' = \mathbf{R}^T \underline{\mathbf{x}} + \underline{\mathbf{t}} \quad (2.5)$$

$$(2.6)$$

donde se ha hecho uso de las propiedades derivadas de la ortonormalidad de \mathbf{R} , principalmente que $\mathbf{R}^{-1} = \mathbf{R}^T$.

La matrix \mathbf{R} usualmente se obtiene por medio de la cascada de rotaciones sobre los ejes coordenados, como por ejemplo

$$\mathbf{R}_\phi = \begin{bmatrix} \cos \phi & \sen \phi & 0 \\ -\sen \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sen \phi \\ 0 & -\sen \phi & \cos \phi \end{bmatrix}$$

$$\mathbf{R}_\psi = \begin{bmatrix} \cos \phi & 0 & \sen \phi \\ 0 & 1 & 0 \\ -\sen \phi & 0 & \cos \phi \end{bmatrix}$$

Las ecuaciones (2.4) y (2.5) se pueden simplificar introduciendo las llamadas coordenadas homogéneas, que se utilizan en geometría proyectiva. En éste sistema, los puntos espaciales se expanden en una dimensión, tal que

$$\underline{\mathbf{x}}'_h = \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{bmatrix} \quad \underline{\mathbf{x}}_h = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

Un punto en coordenadas homogéneas se transforma a un punto con coordenadas euclidianas de la siguiente forma:

$$\underline{\mathbf{x}}_h = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \alpha \end{bmatrix} \rightarrow \underline{\mathbf{x}} = \begin{bmatrix} x_1/\alpha \\ x_2/\alpha \\ x_3/\alpha \end{bmatrix}$$

de modo que para un punto con coordenadas euclidianas $[x_1, x_2, x_3]^T$ existen un infinito número de puntos equivalentes en el espacio proyectivo $[x_1\alpha, x_2\alpha, x_3\alpha, \alpha]^T$, para todo $\alpha \neq 0$.

Así, las ecuaciones (2.4) y (2.5) se reducen a

$$\underline{\mathbf{x}}_h = \mathbf{M}_h \underline{\mathbf{x}}'_h \quad (2.7)$$

$$\underline{\mathbf{x}}'_h = \mathbf{M}_h^{-1} \underline{\mathbf{x}}_h \quad (2.8)$$

con

$$\mathbf{M}_h = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

donde los nueve términos r_{ij} componen la matriz \mathbf{R} de 3×3 utilizada en (2.4) y (2.5).

2.6 Estenoscopio y Proyección Central

El modelo más simple de proyección es el realizado por el estenoscopio, que consiste en colocar sobre el plano focal una placa opaca de extensión infinita con un único agujero infinitesimalmente pequeño en su centro. El eje óptico atraviesa al plano focal por el agujero, de modo que un único rayo de luz proveniente de un punto en particular de la escena puede alcanzar el plano de imagen, puesto que debe atravesar el agujero (figura 2.18). La distancia entre los planos focal y de imagen se conoce como distancia focal f .

Para establecer el modelo de proyección realizado por el estenoscopio se asume que la distancia entre los planos de imagen y de objeto es x'_3 y se considera la semejanza de los

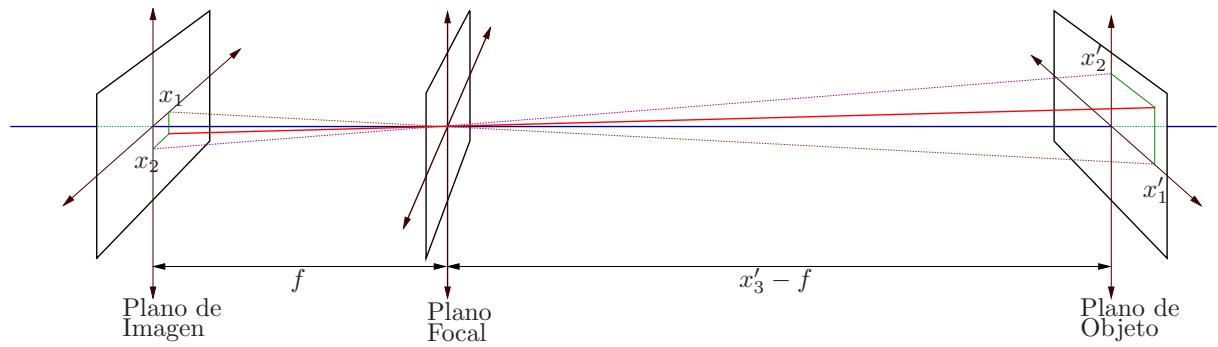


Figura 2.18: Modelo del estenoscopio.

triángulos formados a ambos lados del plano focal, con lo que se cumple

$$x_1 = -x'_1 \frac{f}{x'_3 - f} = x'_1 \frac{f}{f - x'_3}$$

$$x_2 = -x'_2 \frac{f}{x'_3 - f} = x'_2 \frac{f}{f - x'_3}$$

es decir, las coordenadas del punto en el plano de imagen se obtienen escalando las coordenadas del punto en el plano del objeto por un término que depende de la distancia focal y de la distancia del objeto al plano focal ($x'_3 - f$). El signo negativo pone en evidencia la inversión ocurrida en la imagen proyectada. Utilizando coordenadas homogéneas, ésta relación se puede plantear a través de una matriz de proyección \mathbf{P} como

$$\begin{bmatrix} x_{1h} \\ x_{2h} \\ \alpha \end{bmatrix} = \mathbf{P} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{bmatrix}$$

donde utilizando las propiedades de coordenadas homogéneas se obtiene finalmente para las coordenadas sobre la imagen

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_{1h}/\alpha \\ x_{2h}/\alpha \end{bmatrix} = \frac{f}{f - x'_3} \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix}$$

Si f es menor que cero, entonces el modelo anterior se transforma en un modelo de proyección central. En éste modelo se asume que reemplazando el agujero se convierte en la única fuente de radiación puntual, y se elige que el plano de objeto sea intermedio al plano focal (o de emisión) y al plano de imagen. Éste modelo es la primera simplificación del proceso proyectivo que ocurre por ejemplo con Rayos X, donde la radiación atraviesa al objeto, y lo que se captura es la radiación que logra atravesar al objeto (figura 2.19).

Por hacer:

Ejemplo con la LTI-Lib-2: Proyectando una imagen en el espacio 3D.

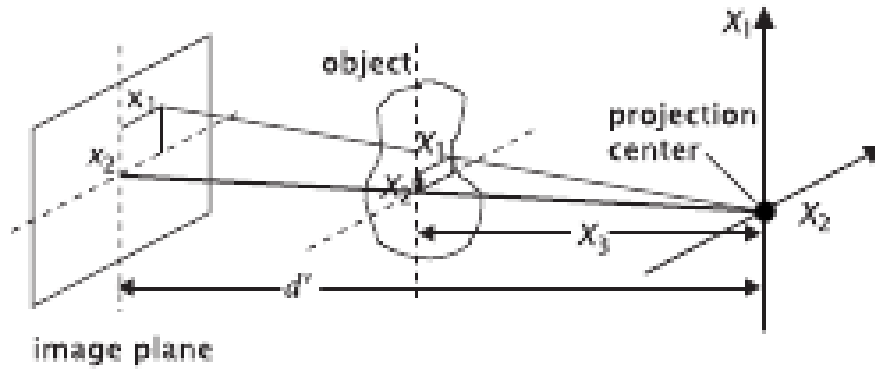


Figure 7.5: Perspective projection with x-rays.

Figura 2.19: Modelo de proyección central. (Jahne 2005)

2.7 Proyección con el modelo de lente gruesa

El modelo del estenoscopio sobresimplifica los procesos ocurridos en cualquier sistema de proyección óptica utilizado en la actualidad. Dicho modelo proyecta imágenes perfectamente enfocadas sin importar la distancia del objeto al plano de proyección; sin embargo, los sistemas reales solo permiten producir imágenes enfocadas en un rango finito de distancias. El modelo de lente gruesa simétrica modifica al estenoscopio levemente y permite describir dichos efectos.

En el modelo de lente gruesa simétrico el plano focal del estenoscopio es reemplazado por dos *planos principales*. El eje óptico interseca dichos planos en los llamados *puntos principales*. Cualquier rayo que ingrese al primer punto principal sale con el mismo ángulo del segundo punto principal, y viceversa. La distancia entre los dos planos principales modela, así, la extensión axial del sistema óptico.

Los rayos paralelos al eje óptico que ingresen al sistema, continúan paralelos a dicho eje dentro del sistema y a la salida se refractan de modo tal que intersecan al eje óptico en el punto focal. La distancia entre el punto principal y su correspondiente punto focal se denomina distancia focal efectiva f .

La relación entre la distancia d del objeto a su punto focal se relaciona con la distancia d' del plano de imagen a su punto focal a través de la ecuación de imagen de Newton:

$$dd' = f^2$$

que también se puede expresar en la forma de Gauss como

$$\frac{1}{d' + f} + \frac{1}{d + f} = \frac{1}{f}$$

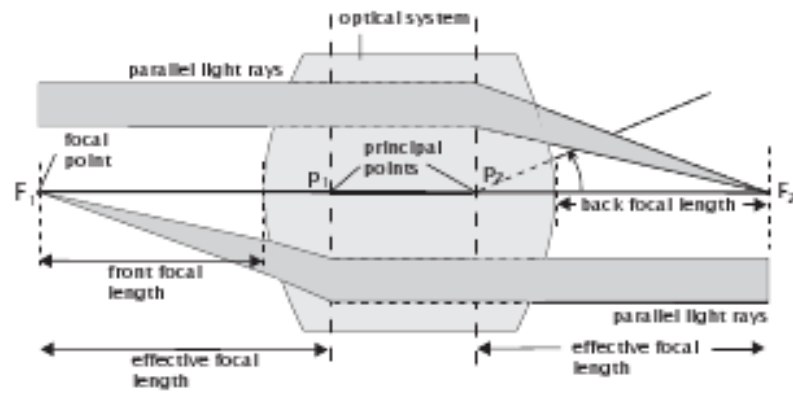


Figure 7.6: Black box model of an optical system.

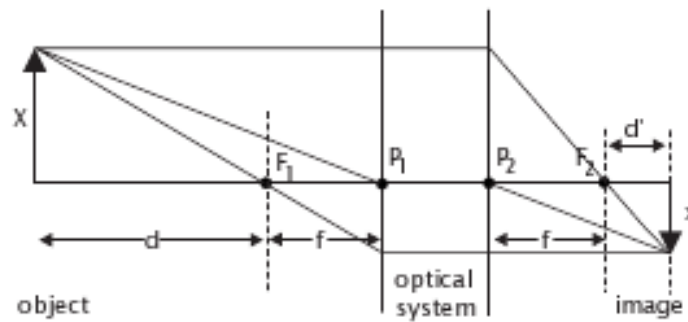


Figure 7.7: Optical imaging with an optical system modeled by its principal points P_1 and P_2 and focal points F_1 and F_2 . The system forms an image that is a distance d' behind F_2 from an object that is the distance d in front of F_1 .

Figura 2.20: Modelo de lente gruesa. (Jahne 2005)

2.7.1 Profundidad de foco y de campo

Si el plano de imagen se desplaza hacia adelante o hacia atrás, o si el objeto se mueve hacia adelante o hacia atrás, la imagen proyectada se difumina proporcionalmente al desplazamiento de los planos en foco.

Puesto que los elementos utilizados en la captura de imágenes digitales tienen tamaño finito, es permisible aceptar un desenfoque tal que un punto se proyecte en un disco de radio ϵ que quepa dentro del elemento fotosensible.

Sea n_r la razón entre la distancia focal f y el diámetro de apertura $2r$. El diámetro del disco de desenfoque será entonces

$$\epsilon = \frac{1}{2n_r} \frac{f}{f + d'} \Delta x_3$$

donde Δx_3 es la distancia al plano enfocado de la imagen. El rango $[d' - \Delta x_3, d' + \Delta x_3]$ para el cual el radio del disco de desenfoque es menor que ϵ se conoce como la *profundidad de foco*.

La *profundidad de campo* (*depth of field*) es el rango de distancias del objeto que producen

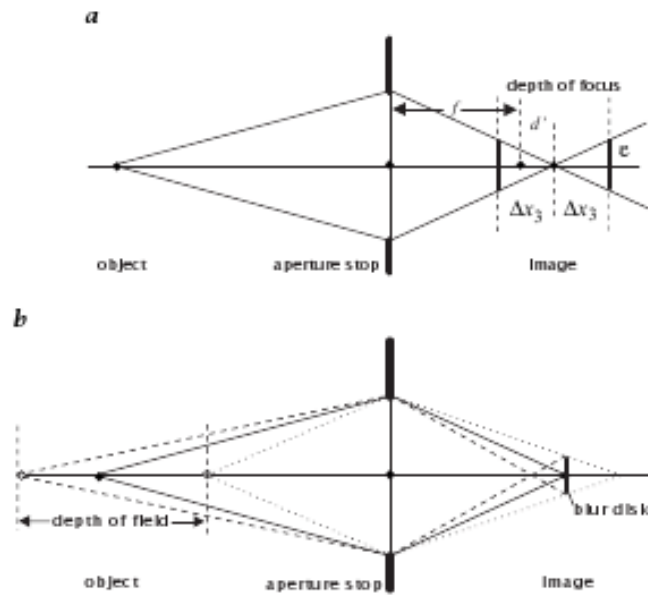


Figure 7.8: Illustration of the **a** depth of focus and **b** depth of field with an on-axis point object.

Figura 2.21: Profundidad de foco y profundidad de campo. (Jähne 2005)

un radio del disco de desenfoco sobre un plano de imagen fijo menores a ϵ .

Por hacer:

Agregar condiciones de que en escalas de m_l (magnificación lateral) baja la profundidad de campo es grande, pero para m_l muy grandes, como en microscopios, el DOF es muy pequeño.

Transformaciones entre el espacio de la escena y el espacio de la imagen. (Cap.7 del Jähne). Espacio euclidiano, y espacio proyectivo.

Óptica básica: El modelo del estenoscopio (pinhole camera) y la proyección central. (Pag. 18-19 de Meyer-Ebrecht)

Profundidad de foco y profundidad de campo.

Objetivos

Distorsiones [34]

Muestreo

2.8 Sistema de captura

Discretización de la señal

Sensores CCD y CMOS

Captura del color RGB y multiespectral para imágenes satelitales.

Interfaces para la captura (de hardware USB, FireWire, CameraLink, etc. y de software V4L, V4W, IIDC, etc.)

2.9 El sistema visual humano

2.9.1 El ojo

El entorno del ojo humano: composición de la luz sobre la superficie terrestre.

Sistema de proyección

El cristalino

La pupila y el iris

Sistema de conversión

Retina: Conos y bastoncillos, respuesta frecuencial, densidad de fotosensores, la fovea y la visión periférica.

Sensibilidad: conos requieren más fotones que bastoncillos para funcionar, por lo que la mayor densidad en la fovea es solo capaz de funcionar con suficiente luz

Sistema de procesamiento

Primera fase: campos receptivos en la misma retina

Cortex visual

2.10 Imágenes digitales y sus elementos

Las secciones anteriores describieron el proceso de formación de una imagen, abarcando desde las fuentes de radiación, su interacción con los objetos de la escena, el proceso de proyección de la radiación sobre el sensor, y la conversión de dicha proyección a una señal eléctrica digital, lista para ser procesada.

Corresponde ahora a describir varias de las representaciones matemáticas utilizadas usualmente para describir las imágenes capturadas.

Como primer punto, nótese que una imagen es una señal, y como tal se puede caracterizar a través de las cinco características ilustradas en la tabla 2.3. Las imágenes son señales

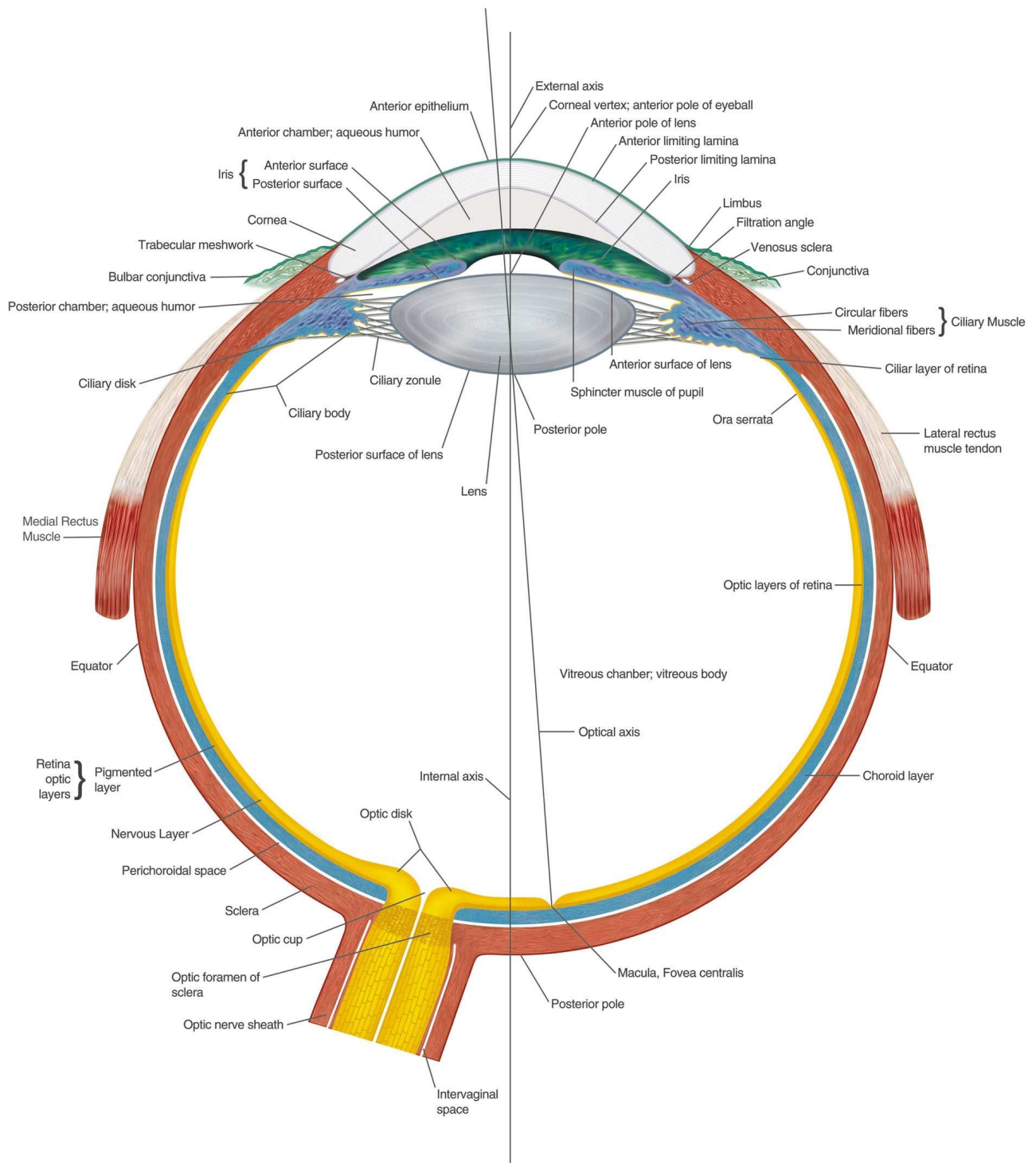


Figura 2.22: Anatomía del ojo

usualmente de una, dos o tres dimensiones espaciales. Por ejemplo, los scanners utilizan cámaras lineales, que producen solo una línea de información espacial. Las cámaras fotográficas comunes en el mercado actual utilizan chips CCD que capturan directamente información bidimensional. En el área médica se dispone de tecnologías capaces de capturar información en el espacio tridimensional, como la tomografía axial computarizada o

Tabla 2.3: Características de las señales

Característica	Valores	
Número de variables	una variable	múltiples variables
Dimensionalidad	escalar	vectorial (multicanal)
Variables independientes	discretas	continuas
Valores de la señal	discretos	continuos
Naturaleza estadística	deterministas	aleatorias

la resonancia magnética. Cuando la señal, además de las dimensiones espaciales, se define para la variable independiente de tiempo, se habla entonces de una señal de *vídeo*.

En cuanto a la dimensionalidad de la señal, imágenes monocromáticas tienen un solo *canal* de información correspondiente a la intensidad energética en algún rango de frecuencias de la energía capturada. Las imágenes a color convencionales usualmente utilizan tres canales de información monocromática: rojo, verde y azul. Imágenes satelitales separan el espectro en intervalos más estrechos que aquellos en las imágenes convencionales, conteniendo de tres a siete o más canales que pueden abarcan rangos infrarrojos y ultravioleta.

Las variables espaciales usualmente son discretas. Aun en películas fotosensibles existe una granularidad que cuantifica la captura de información espacial lumínica de una forma irregular. Los sensores CCD o CMOS tienen elementos fotosensibles distribuidos regularmente en una rejilla rectangular, que automáticamente discretiza la información espacial capturada. Nótese que la imagen proyectada sobre el sensor es de naturaleza continua, pero las técnicas de captura conocidas, incluso el mismo ojo biológico, la discretizan.

En cuanto a los valores de la señal, éstos pueden considerarse tomados de un intervalo continuo en películas fotosensibles, y en algunos medios analógicos de transmisión de imágenes (como el vídeo compuesto). La sensibilidad del ojo humano es tal que permite cuantificar los niveles de intensidad de canales de imágenes monocromáticas con un nivel relativamente bajo de bits por muestra, así que con frecuencia se utilizan en imágenes de alta calidad tan solo ocho bits por canal, o incluso es común emplear cuatro o cinco bits por canal.

Al igual que con señales temporales, el contenido de una imagen se puede considerar desde una perspectiva probabilística o determinista.

Como *imágenes digitales* se consideran aquellas que tienen tanto las variables independientes discretas así como los valores que cada canal de la imagen puede adquirir. Las imágenes en espacio discreto difieren de las anteriores en que los valores se asumen continuos.

2.10.1 Representaciones de las imágenes

Imágenes como Funciones

Como señales, la primera representación utilizada para imágenes corresponde a funciones matemáticas. Las imágenes en espacio discreto se definen en un dominio de coordenadas espaciales discretas y finitas, en forma de una rejilla rectangular. Dicho dominio es mapeado hacia un conjunto de valores vectoriales en \mathbb{R}^n , donde n equivale a la dimensión de la señal; por ejemplo, para imágenes monocromáticas $n = 1$, para imágenes convencionales a color $n = 3$. Así, la función f definida como

$$f : \mathbb{X} \rightarrow \mathbb{R}^n$$

representará una imagen en espacio discreto, donde $\mathbb{X} = \{0, 1, \dots, D - 1\}^d \subset \mathbb{N}^d$ es el conjunto de posiciones válidas de los píxeles en un espacio de d dimensiones. Así $f(\underline{\mathbf{x}})$ es una función que retorna el vector que representan la composición espectral de una imagen sobre la posición $\underline{\mathbf{x}}$.

Si el codominio \mathbb{R}^n es remplazado por un conjunto discreto de valores \mathbb{V}^n entonces se tiene una *imagen digital*.

Obsérvese que en general las funciones que representan una imagen tienen carácter vectorial. Como *canal k* se denota una única componente $f_k(\underline{\mathbf{x}})$, $k = 1, \dots, n$ de la imagen $f(\underline{\mathbf{x}})$.

En el caso de imágenes bidimensionales es común simplificar la notación del vector $\underline{\mathbf{x}} = [x, y]^T$ como par ordenado (x, y) .

Imágenes como matrices

Para el caso particular del espacio bidimensional ($d = 2$), la forma de rejilla que asume el dominio de definición de las funciones utilizadas para representar imágenes y canales permite derivar una segunda forma de representación basada en matrices, particularmente apta para el manejo computacional de estas señales.

La matriz \mathbf{F} representa a la misma señal dada por la función f si se cumple

$$f_{ji} = f \left(\begin{bmatrix} i \\ j \end{bmatrix} \right)$$

donde además

$$\mathbf{F} = \begin{bmatrix} f_{0,0} & f_{0,1} & \dots & f_{0,C-1} \\ f_{1,0} & f_{1,1} & \dots & f_{1,C-1} \\ \vdots & \vdots & \ddots & \vdots \\ f_{R-1,0} & f_{R-1,1} & \dots & f_{R-1,C-1} \end{bmatrix}$$

con R igual al número de filas y C igual al número de columnas.

De nuevo, si $f_{ji} \in \mathbb{R}^n$ se tiene una imagen en espacio discreto, y si $f_{ji} \in \mathbb{I}^n$ se dice que \mathbf{F} es una imagen digital. Lo anterior se extiende de forma directa para canales.

Imágenes como conjuntos

La diversidad de técnicas existentes para el procesamiento y análisis de imágenes han forzado a crear otros tipos de representación adicionales a los dos enuncados anteriormente. Una representación frecuentemente utilizada es considerar las imágenes como conjuntos de píxeles (del inglés *picture element*). De forma general un píxel \mathbf{p} es una tupla $\langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle$ compuesta por un vector de posición $\underline{\mathbf{x}}$ y un vector $\underline{\mathbf{c}}$ que representa la composición espectral de la energía incidente en la posición del píxel. Es necesario contar con dos operadores que permitan extraer la posición o el valor de un píxel. Estos operadores se denominarán operador de posición “pos” y operador de valor “val”, de modo que

$$\begin{aligned}\text{pos}(\mathbf{p}) &= \text{pos}(\langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle) = \underline{\mathbf{x}} \\ \text{val}(\mathbf{p}) &= \text{val}(\langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle) = \underline{\mathbf{c}}\end{aligned}$$

La imagen \mathcal{I} está definida como

$$\mathcal{I} = \{ \mathbf{p} \mid \mathbf{p} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle, \quad \underline{\mathbf{x}} \in \mathbb{X}, \quad \underline{\mathbf{c}} = f(\underline{\mathbf{x}}) \}$$

Nótese que con esta definición, no todos los valores posibles de posiciones $\underline{\mathbf{x}}$ deben estar necesariamente incluidos en la imagen. De este modo, aparecen operaciones que agregan o eliminan píxeles a la imagen, lo que no es directamente representable con funciones o matrices.

En el caso particular de la morfología (capítulo 6), se utilizan imágenes binarias, donde el codominio de definición de f es un conjunto booleano. Esto permite a los píxeles adquirir valores blanco o negro, verdadero o falso, pertenece o no pertenece a la imagen, etc., lo que hace que la representación como conjunto se simplifique e incluya únicamente las posiciones de los píxeles que asuman uno de los dos valores posibles (por ejemplo, verdadero), sin necesidad de especificar entonces el valor de $\underline{\mathbf{c}}$.

Dependiendo del contexto, como píxel se denotará únicamente a la posición del píxel, sobreentendiéndose que dicha posición tiene un valor espectral correspondiente.

Imágenes como grafos

Las tareas del análisis de imágenes y de visión por computador usualmente tienen como fin detectar estructuras u *objetos* visuales con determinadas características en las imágenes. Para los procesos de detección de dichos objetos es usual encontrar estructuras de datos basadas en grafos, que se componen de nodos y aristas. Como nodos se utilizan los píxeles, definidos como tuplas de posición y valor espectral $\langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle$. Las aristas y sus pesos definen relaciones adicionales entre los píxeles, que pueden basarse tanto en comparaciones de sus posiciones $\underline{\mathbf{x}}$ como de sus valores $\underline{\mathbf{c}}$.

2.10.2 Resolución y tamaño de imágenes

El tamaño T de una imagen se refiere siempre al número de píxeles que tiene una imagen. Las dimensiones físicas D que adquiere la representación de una imagen depende del medio de representación. Otra característica relacionada con lo anterior, es la *resolución espacial* ρ , definida como el número de píxeles por unidad de área, y relaciona entonces las dimensiones físicas de la representación con el tamaño de la imagen:

$$D = T \times \rho$$

También se utiliza el término *resolución de intensidad* para especificar el número de bits necesarios para codificar al número de niveles que puede tomar el valor de los píxeles de una imagen digital.

2.10.3 Imágenes rasterizadas y vectoriales

Debe enfatizarse que no se debe confundir una imagen, que es una señal (representada como se expresó anteriormente a través de funciones, matrices, conjuntos o grafos), con los métodos y técnicas que se emplean para almacenarlas.

La creación sintética de imágenes digitales puede permitir simular resoluciones infinitas, en el sentido de que puede aumentarse el número de píxeles por unidad de área arbitrariamente, manteniendo el tamaño de la imagen constante. Esto se utiliza por ejemplo en despliegue de información, tanto sobre pantallas como en material impreso.

Para esto se utilizan representaciones *vectoriales* del contenido de las imágenes. Esto quiere decir, que en vez de crear la rejilla para una imagen y asignar a los píxeles correspondientes el contenido deseado, se almacenan primitivas geométricas que pueden reexpresarse en forma de una nueva imagen *rasterizada* cada vez que se quiere cambiar la resolución. Los gráficos vectoriales son preferidos para almacenar textos y figuras regulares esquemas con alta calidad visual y poco espacio de almacenamiento. Las imágenes rasterizadas son resultado de procesos de captura de escenas reales, y serán utilizadas cuando se requiera extraer información de dichas escenas. Si bien es cierto, la visualización de los gráficos vectoriales implica su conversión a una imagen rasterizada, la conversión inversa de una imagen rasterizada a su contenido con representación vectorial es un proceso complejo que requiere avanzadas técnicas de análisis y de visión por computador.

Un ejemplo de estandar para el almacenamiento de imágenes vectoriales lo constituye el SVG (Scalable Vector Graphics [78]), propuesto para el almacenamiento de gráficos en el web con alta calidad. Para uso convencional existen formatos más sencillos incluyendo el WMF (Windows Meta File), EMF (Enhanced Metafiles), ambos de Microsoft, o XFig y OpenDocument Graphics, que son formatos abiertos.

El almacenamiento de imágenes rasterizadas utiliza formatos libres de pérdida como BMP (BitMap) de Microsoft, PNG (Portable Network Graphics), TIFF, GIF, y muchos más,

pero también formatos con pérdida de información cuando se debe reducir espacio de almacenamiento y no es crítica dicha pérdida, lo que incluye a los formatos JPEG y JPEG2000, así como al PNG con pérdida.

2.11 Capturando imágenes con la LTI-Lib

2.11.1 Archivos

2.11.2 FireWire

2.11.3 USB

2.12 Problemas

Capítulo 3

Procesamiento en el dominio espacial

3.1 Operaciones y relaciones básicas

Las diferentes representaciones de imágenes permiten definir operaciones y relaciones básicas entre imágenes o entre píxeles. En esta sección se realiza un bosquejo general de operadores y en secciones siguientes se analizarán casos particulares.

3.1.1 Vecindades

Las vecindades de un píxel se definen en términos de distancias entre las posiciones a otros píxeles, donde la distancia se define como un operador binario entre las componentes espaciales:

$$d : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$$

Dicho operador puede basarse, entre otras, en las métricas de Minkowsky L_p , definidas como

$$L_p(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}$$

de tal modo que L_2 corresponde a la distancia euclídeana y L_1 a la distancia de *cuadras de ciudad*. Puede demostrarse que el límite cuando p tiende a infinito, entonces L_∞ produce la máxima distancia entre coordenadas $\max |x_i - y_i|$. La figura 3.1 presenta en un plano \mathbb{R}^2 todos los puntos con distancias $L_i(\underline{\mathbf{x}}, \underline{\mathbf{0}}) = 1$ al origen $\underline{\mathbf{0}}$. Así mismo, el punto denotado con $\underline{\mathbf{x}}$ en dicha figura se encuentra a una distancia hacia el origen de $L_1(\underline{\mathbf{x}}, \underline{\mathbf{0}}) = 2$, $L_2(\underline{\mathbf{x}}, \underline{\mathbf{0}}) = \sqrt{2}$ y $L_\infty(\underline{\mathbf{x}}, \underline{\mathbf{0}}) = 1$.

La transformación de distancia evaluada en el capítulo 6 se basa en estas definiciones.

Como *vecindad de 4* de un píxel $\mathbf{p} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle$, denotada como $N_4(\mathbf{p})$ se denota al conjunto de todos los píxeles $\mathbf{q}_i = \langle \underline{\mathbf{y}}, \cdot \rangle$ que cumplen $L_1(\underline{\mathbf{x}}, \underline{\mathbf{y}}) = 1$. En el caso de imágenes bidimensionales, si $\underline{\mathbf{x}} = [x, y]^T$, entonces $N_4(\mathbf{p})$ incluye a píxeles en las posiciones

$$[x + 1, y]^T, [x - 1, y]^T, [x, y + 1]^T, [x, y - 1]^T$$

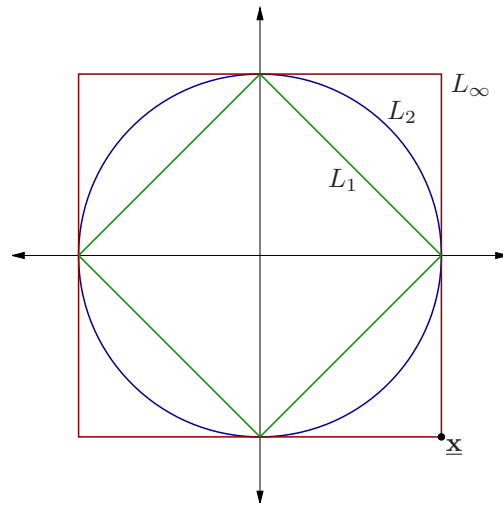


Figura 3.1: Distancias de Minkowsky iguales hacia el origen del plano \mathbb{R}^2 iguales a 1.

Como *vecindad de 8* de un píxel $\mathbf{p} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle$, denotada como $N_8(\mathbf{p})$ se denota al conjunto de todos los píxeles $\mathbf{q}_i = \langle \underline{\mathbf{y}}, \cdot \rangle$ que cumplen $1 \leq L_2(\underline{\mathbf{x}}, \underline{\mathbf{y}}) \leq \sqrt{2}$. En el caso de imágenes bidimensionales, si $\underline{\mathbf{x}} = [x, y]^T$, entonces $N_8(\mathbf{p})$ incluye a los píxeles de $N_4(\mathbf{p})$ más píxeles en las diagonales

$$[x + 1, y + 1]^T, [x - 1, y + 1]^T, [x + 1, y - 1]^T, [x - 1, y - 1]^T$$

Ambas vecindades se ilustran para el caso bidimensional en la figura 3.2

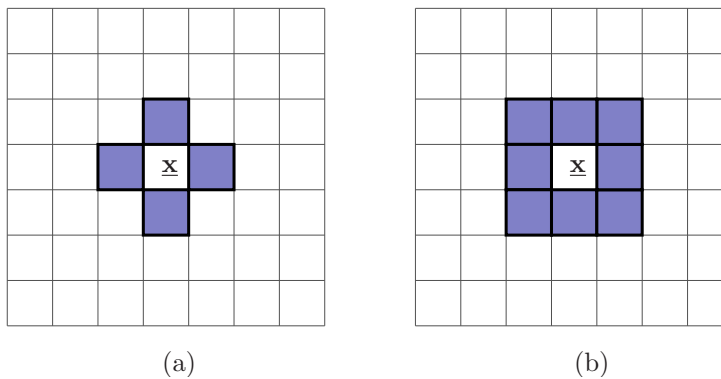


Figura 3.2: Vecindad (a) N_4 y (b) N_8 .

3.1.2 Operaciones aritméticas

La representación funcional de imágenes permite definir operaciones aritméticas básicas entre píxeles correspondientes:

$$s(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}) + g(\underline{\mathbf{x}})$$

$$d(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}) - g(\underline{\mathbf{x}})$$

$$p(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}}) \times g(\underline{\mathbf{x}})$$

$$v(\underline{\mathbf{x}}) = f(\underline{\mathbf{x}})/g(\underline{\mathbf{x}})$$

Estas operaciones son componentes elementales del procesamiento de las imágenes. Como ejemplos, la suma se usa en promediación de imágenes que es una técnica básica para la reducción de ruido; la substracción se utiliza como detector de cambios en secuencias de imágenes; y el producto es el concepto fundamental detrás del enmascaramiento.

3.1.3 Operaciones de conjuntos

La representación de imágenes como conjuntos permite definir operaciones de imágenes basadas en la unión, intersección, complemento y diferencia de otras imágenes. En éste documento se revisará posteriormente la morfología (capítulo 6) como caso particular de aplicación de éstas operaciones.

En general, se parte de la rejilla completa $\mathbb{X} = \{0, 1, \dots, D - 1\}^d \subset \mathbb{N}^d$, de tal modo que imágenes son subconjuntos de píxeles sobre dichas rejillas. La figura 3.3 ilustra las operaciones básicas de dos imágenes binarias bidimensionales.

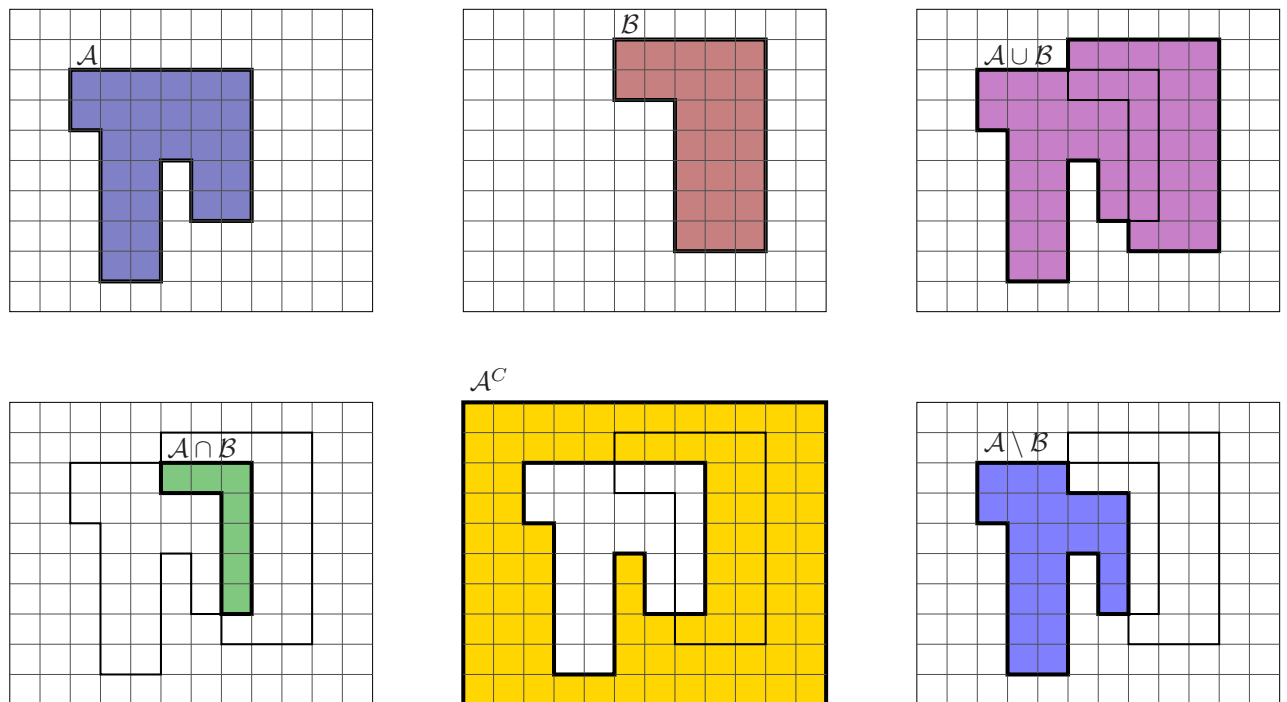


Figura 3.3: Operaciones de conjunto entre dos imágenes \mathcal{A} y \mathcal{B} definidas sobre la rejilla \mathbb{X} .

3.1.4 Operaciones espaciales

Las operaciones espaciales se aplican directamente sobre los píxeles de una imagen. Utilizando la representación de conjuntos, se pueden plantear las siguientes transformaciones

espaciales:

1. Operaciones sobre de valores de píxeles
2. Operaciones sobre los valores de vecindades
3. Operaciones geométricas

Una operación sobre los valores de los píxeles en una imagen \mathcal{A} está definida como

$$\mathcal{B} = \{\mathbf{b} \mid \mathbf{b} = \langle \underline{\mathbf{y}}, \underline{\mathbf{d}} \rangle, \quad \underline{\mathbf{d}} = T(\underline{\mathbf{c}}), \quad \mathbf{a} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle \in \mathcal{A}\}$$

lo que se intepreta como una modificación del valor de cada píxel, en función únicamente del valor del píxel correspondiente en la imagen destino. La figura 3.4 muestra el efecto de modificar los valores de gris de cada píxel en una imagen siguiendo una transformación simple de contraste.



Figura 3.4: Modificación del valor de cada píxel.

En una operación de vecindad, el valor de un píxel $\mathbf{q} = \langle \underline{\mathbf{x}}, \underline{\mathbf{d}} \rangle$ en la imagen destino \mathcal{B} depende de los valores de los píxeles en una vecindad del píxel \mathbf{p} en \mathcal{A} con igual posición $\underline{\mathbf{x}}$ que \mathbf{q} . Si $N(\underline{\mathbf{p}})$ denota una vecindad del píxel $\underline{\mathbf{p}}$, es decir, un conjunto de píxeles que conservan una relación geométrica predefinida con respecto a $\underline{\mathbf{x}}$, entonces la transformación de vecindad se puede expresar matemáticamente como

$$\mathcal{B} = \{\mathbf{b} \mid \mathbf{b} = \langle \underline{\mathbf{x}}, \underline{\mathbf{d}} \rangle, \quad \underline{\mathbf{d}} = T(\text{val}(N(\mathbf{a}))), \quad \mathbf{a} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle \in \mathcal{A}\}$$

La figura 3.5 ilustra una operación de vecindad.

Una operación geométrica de la imagen \mathcal{A} hacia la imagen \mathcal{B} está definida como

$$\mathcal{B} = \{\mathbf{b} \mid \mathbf{b} = \langle \underline{\mathbf{y}}, \underline{\mathbf{c}} \rangle, \quad \underline{\mathbf{y}} = T(\underline{\mathbf{x}}), \quad \mathbf{a} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle \in \mathcal{A}\}$$

donde los píxeles de la imagen original \mathcal{A} mantienen su vector espectral $\underline{\mathbf{c}}$, pero cambian su posición $\underline{\mathbf{x}}$ en la imagen destino \mathcal{B} . En la práctica, puesto que la imagen destino debe



Figura 3.5: Modificación del valor de cada píxel.

ser ubicada en una rejilla regular, es costumbre utilizar una transformación inversa junto con operaciones de interpolación:

$$\mathcal{A} = \{ \mathbf{a} \mid \mathbf{a} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle, \quad \underline{\mathbf{x}} = T^{-1}(\underline{\mathbf{y}}), \quad \mathbf{b} = \langle \underline{\mathbf{y}}, \underline{\mathbf{c}} \rangle \in \mathcal{B} \}$$

La figura 3.6 muestra el efecto de aplicar un mapeo bilineal a las coordenadas de cada píxel. recorriendo todas las posiciones de los píxeles de \mathcal{B} y determinando entonces cuál



Figura 3.6: Mapeo bilineal aplicado a una imagen monocromática.

valor de $\underline{\mathbf{c}}$ corresponde a cada píxel.

3.1.5 Transformaciones de dominio

Una última posibilidad de operar con imágenes la representan las transformaciones de dominio, en donde las imágenes y sus píxeles son transformados a representaciones equivalentes en otros dominios (por ejemplo, el dominio de frecuencia). Las relaciones espaciales

pueden quedar ocultas en la nueva representación, pero la transformación puede presentar ventajas para la detección de cierta información relevante para alguna aplicación particular. La mayoría de transformaciones utilizadas son reversibles, en el sentido de que la representación en el dominio de transformación puede ser reconvertido a la imagen que le dio origen.

Un caso particular lo representan las transformaciones lineales de la forma

$$F(u, v) = \sum_{y=0}^{R-1} \sum_{x=0}^{C-1} f(x, y) r(x, y, u, v)$$

donde a $r(x, y, u, v)$ se le conoce como *kernel* de transformación. Ésta transformación tiene inversa si existe un kernel $s(u, v, x, y)$ que cumple

$$f(x, y) = \sum_{y=0}^{R-1} \sum_{x=0}^{C-1} F(u, v) s(u, v, x, y)$$

Al kernel r se le denomina separable si cumple

$$r(x, y, u, v) = r_1(x, u) r_2(y, v)$$

y es simétrico si además $r_1(x, y) = r_2(x, y)$ con lo que

$$r(x, y, u, v) = r_1(x, u) r_1(y, v)$$

3.2 Transformaciones similares, afines y proyectivas

Un primer caso de transformación espacial consiste en la familia de transformaciones geométricas básicas resultantes de rotaciones, escalamientos y traslaciones de imágenes bidimensionales. Para las siguientes secciones se utilizará un sistema coordinado izquierdo, correspondiente con las representaciones usuales de imágenes digitales, en donde el origen se posiciona en la esquina superior izquierda, el eje horizontal x se utiliza para numerar los píxeles de izquierda a derecha, y el eje vertical y se utiliza para numerar los píxeles de arriba hacia abajo. Se mantendrá en las siguientes ecuaciones la convención de ángulos positivos como ángulos medidos en sentido antihorario.

Dada la posición \underline{x} de un píxel, extendida a coordenadas homogéneas en \underline{x}_h , es decir:

$$\underline{x} = \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \underline{x}_h = \begin{bmatrix} \alpha x \\ \alpha y \\ \alpha \end{bmatrix}$$

donde α debe ser diferente de cero y se elige usualmente como $\alpha = 1$. Entonces se distinguen los siguientes tipos de transformaciones elementales:

- Rotación en un ángulo θ en el plano de la imagen sobre el origen:

$$\underline{\mathbf{u}} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \theta & \text{sen } \theta \\ -\text{sen } \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{R}_\theta \underline{\mathbf{x}}$$

que en coordenadas homogéneas se puede expresar como

$$\underline{\mathbf{u}}_h = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & \text{sen } \theta & 0 \\ -\text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\theta & \underline{\mathbf{0}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{R}_{\theta h} \underline{\mathbf{x}}_h$$

- Escalamiento isotrópico por un factor s :

$$\underline{\mathbf{u}} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{Z}_s \underline{\mathbf{x}}$$

En coordenadas homogéneas el escalamiento isotrópico se puede obtener como

$$\underline{\mathbf{u}}_h = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{Z}_{sh} \underline{\mathbf{x}}_h$$

- Finalmente la traslación se obtiene en coordenadas homogéneas con

$$\underline{\mathbf{u}}_h = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \underline{\mathbf{t}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{T}_{(t_x, t_y)_h} \underline{\mathbf{x}}_h$$

En coordenadas euclidianas es posible representar la traslación como una suma vectorial, pero no es práctico para la composición de transformaciones más complejas.

3.2.1 Transformación euclidiana

Una transformación euclidiana solo rota y traslada. La matriz que la representa se puede obtener como el producto de $\mathbf{R}_{\theta h}$ con $\mathbf{T}_{(t_x, t_y)_h}$. Nótese que el orden del producto sí altera el resultado: no es lo mismo trasladar y luego rotar con respecto al origen, que rotar con respecto al origen y luego trasladar. En general, la matriz \mathbf{M}_h de una transformación euclidiana está dada por:

$$\mathbf{E}_h = \begin{bmatrix} \cos \theta & \text{sen } \theta & t_x \\ -\text{sen } \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\theta & \underline{\mathbf{t}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix}$$

de donde se observa que la matriz depende únicamente de tres parámetros θ , t_x y t_y , por lo que se dice que tiene tres grados de libertad.

Ésta transformación tiene la propiedad de mantener longitudes y áreas en las imágenes inalteradas.

3.2.2 Transformación de similitud

La transformación de similitud es una transformación euclidiana a la que se le aplica un escalado isotrópico. La matriz homogénea que representa ésta transformación está dada por

$$\mathbf{S}_h = \begin{bmatrix} s \cos \theta & s \sin \theta & t_x \\ -s \sin \theta & s \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s\mathbf{R}_\theta & \underline{\mathbf{t}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix}$$

que tiene cuatro grados de libertad por depender únicamente de los parámetros θ , t_x , t_y y s . Esta transformación mantiene constantes razones entre longitudes y ángulos.

3.2.3 Transformación afín

La transformación afín está descrita por la matriz homogénea

$$\mathbf{A}_h = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \underline{\mathbf{t}} \\ \underline{\mathbf{0}}^T & 1 \end{bmatrix}$$

donde la submatriz

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

no debe ser singular, y se puede descomponer como

$$\mathbf{A} = \mathbf{R}_\theta \mathbf{R}_{-\phi} \mathbf{D}_{(\lambda_1, \lambda_2)} \mathbf{R}_\phi$$

donde la matriz de escalamiento anisotrópico $\mathbf{D}_{(\lambda_1, \lambda_2)}$ está dada por

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

En palabras, la matriz \mathbf{A} representa una rotación inicial en un ángulo ϕ , seguida por un escalamiento anisotrópico que distorsiona la figura original al escalar en x por un factor y en y por otro, retorna la imagen a su ángulo original y vuelve a rotar por un nuevo ángulo θ .

La transformación total \mathbf{A}_h tiene entonces seis grados de libertad. Ésta transformación mantiene el paralelismo de las líneas, y mantiene la razón de longitudes de segmentos de líneas paralelos entre sí. Además, la razón entre dos áreas también es constante.

3.2.4 Transformación Proyectiva

La matriz de transformación proyectiva tiene la forma general

$$\mathbf{H}_h = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ v_1 & v_2 & v \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \underline{\mathbf{t}} \\ \underline{\mathbf{v}}^T & v \end{bmatrix}$$

Puesto que en la conversión de coordenadas homogéneas a coordenadas euclidianas ocurre una normalización, puede demostrarse que los grados de libertad de ésta matriz son ocho, y no nueve como el número de variables involucradas. En cuanto a invariantes, la razón de razones de longitudes entre cuatro puntos colineales es invariante ante esta transformación.

Por hacer:

Ejemplo de uso de `lti::matrixTransform`

3.3 Transformaciones de niveles de gris

Las transformaciones sobre niveles de gris son el caso básico de operaciones sobre de valores de píxeles. Con la notación funcional de imágenes, una imagen $f(x, y)$ es transformada en una imagen $g(x, y)$ con

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

donde T es una función escalar para el caso en cuestión.

La tabla 3.1 ilustra varios casos de transformaciones funcionales utilizadas sobre funciones escalares, asumiendo que los valores de la imagen de entrada y salida varían entre 0 y $L - 1$. La normalización se utiliza cuando se pasa de representaciones numéricas de punto

Tabla 3.1: Transformaciones de niveles de gris.

$s = T(r)$	Tipo de Transformación
$s = \frac{r}{L - 1}$	Normalización
$s = L - 1 - r$	Negación
$s = c \log(1 + r)$	Logarítmica
$s = cr^\gamma$	Ley potencial (o transformación gama)
$s = \begin{cases} 0 & \text{si } r < \tau \\ 1 & \text{si } r \geq \tau \end{cases}$	Binarización por umbral τ

fijo a representaciones de punto flotante, donde el valor 1 representa el máximo valor de intensidad representable. La negación produce el “*negativo*” de la imagen de entrada, que es empleado cuando se quiere invertir el contraste de la imagen para facilitar la percepción de determinados detalles, considerando que el ojo humano puede percibir mejor los colores claros que los oscuros. La transformación logarítmica se utiliza para resaltar imágenes en donde la mayor cantidad de información se concentra en niveles bajos de intensidad, como por ejemplo las magnitudes de transformadas de Fourier. La transformación gama se emplea en dispositivos de captura, despliegue e impresión (cámaras, monitores e impresoras), para compensar no linealidades en la respuesta de los píxeles físicos con respecto a las representaciones digitales de las imágenes. La compensación es un proceso necesario para asegurar que la intensidad percibida en los distintos medios es equivalente a la real.

Por hacer:

Imágenes ejemplo de las transformaciones gamma. Fig. 3.3, pag. 108 González Woods

Además de estas transformaciones, puede concebirse cualquier mapeo cerrado sobre el intervalo $[0, L - 1]$, como segmentos de recta, supresiones de segmentos, etc. La mejor forma de implementar éste tipo de transformaciones para imágenes digitales con un número discreto de valores a la entrada, es utilizando tablas de consulta (*look-up tables*, LUT) que se precaculan para luego utilizarlas en la búsqueda del valor concreto de cada píxel.

Por hacer:

Ejemplo de uso de LUT para transformar canales de 8 bits

3.3.1 Histogramas

El mejoramiento del contraste en imágenes, entendido como el proceso de optimización del uso del rango dinámico disponible, puede hacer uso de la información estadística contenida en histogramas de la imagen completa o de secciones en ella.

Sea R el intervalo de valores escalares que puede tomar un píxel en una imagen o canal

$$R = [0, L - 1]$$

Asúmase una partición de dicho intervalo, regular o irregular, en K subintervalos tales que

$$R = [0, \dots, L - 1] = [r_0, r_1[\cup [r_1, r_2[, \cup \dots [r_k, r_{k+1}[\dots \cup [r_{K-1}, r_K]$$

El histograma se define como la función $h(k) = n_k$ que asocia al k -ésimo subintervalo $[r_k, r_{k+1}[$ el número de píxeles cuyo valor se encuentra contenido en él:

$$h(k) = |\{\mathbf{p} \mid \mathbf{p} = \langle \mathbf{x}, r \rangle, r \in [r_k, r_{k+1}[\}|$$

El histograma está ligado con la distribución de probabilidad para los valores de gris. Sea N el total de píxeles en la imagen:

$$N = \sum_{k=0}^{K-1} h(k)$$

La probabilidad de que un píxel tenga un valor dentro del k -ésimo intervalo está dada por

$$p(k) = \frac{h(k)}{N}$$

Obsérvese que

$$\sum_{k=0}^{K-1} p(k) = 1$$

puesto que un píxel debe adquirir cualquiera de los valores posibles.

La función de distribución acumulativa

$$q(k) = \sum_{j=0}^k p(j)$$

expresa cuál es la probabilidad de que un píxel adquiriera el valor dentro del intervalo $[0, r_{k+1}]$

Por hacer:

Ejemplos de imágenes con sus respectivos histogramas.

Ecualización

La concentración de valores en el histograma en una región de los subintervalos de valores de gris indica un mal contraste, puesto que el rango dinámico de valores se estaría subutilizando. Por otro lado, imágenes con un histograma homogéneamente distribuido se consideran de buen contraste por utilizar todo el rango. Esto último es equivalente a afirmar que lo ideal es que la probabilidad de los píxeles de adquirir un determinado valor es igual para todos los intervalos de valores.

Una estrategia de mejorar el contraste en imágenes es encontrar una transformación de niveles de gris tal que *ecualice* el histograma, expandiéndolo para utilizar todo el rango disponible, y buscando que las probabilidades de todos los valores sean iguales (de ahí el nombre de la operación).

Por hacer:

Demostrar de donde viene el siguiente resultado

Se puede demostrar que

$$s_k = T(k) = (L - 1)q(k) = (L - 1) \sum_{j=0}^k p(j)$$

es una transformación de niveles de gris que ecualiza el histograma.

3.3.2 Histogramas locales

3.4 Teoría de sistemas y filtrado espacial

Las operaciones de filtrado en el dominio del procesamiento de imágenes tiene sus fundamentos en la teoría de sistemas.

Por hacer:

Diagrama de bloques con entrada a , función de sistema h y salida b

Hay tres tareas básicas de la teoría de sistemas

1. Simulación de sistemas h y a conocidos, determinar b
2. Reconstrucción de señales h y b conocidos, determinar a
3. Identificación de sistemas a y b conocidos, determinar h

En el procesamiento de señales existen estos tres problemas y se utilizan las técnicas ya revisadas en otros cursos de la carrera, con las diferencias de que

1. la señales se definen en el espacio y no en el tiempo
2. dos variables independientes espaciales en vez de una temporal.

3.4.1 Linealidad e Invarianza a la traslación

Sea un sistema T que transforma una imagen de entrada a en una imagen de salida b :

$$b(x, y) = T[a(x, y)]$$

donde los paréntesis cuadrados deben hacer explícito en este caso que trata de cualquier tipo de transformación.

El sistema se denomina *lineal* si

$$T[k_1 a_1(x, y) + k_2 a_2(x, y)] = k_1 T[a_1(x, y)] + k_2 T[a_2(x, y)]$$

y se considera invariante a la traslación si

$$T[a(x - x_0, y - y_0)] = b(x - x_0, y - y_0)$$

El sistema es lineal e invariante a la traslación (LSI de *linear and shift invariant*) si cumple ambas propiedades.

3.4.2 Convolución

El primer paso para continuar, es definir un equivalente al impulso unitario utilizado en sistemas de tiempo discreto unidimensionales, pero aplicable en imágenes bidimensionales. El impulso espacial unitario $\delta(x, y)$ se define en términos del impulso unidimensional como:

$$\delta(x, y) = \delta(x)\delta(y) = \begin{cases} 1 & \text{si } x = y = 0 \\ 0 & \text{en el resto} \end{cases}$$

Sea $f(x, y)$ la función que describe una imagen sobre una rejilla discreta. Obsérvese que es posible extraer el valor de un píxel en las coordenadas (x, y) por medio de:

$$f(x, y) = f(u, v)\delta(x - u, y - v)$$

para (u, v) variables, puesto que ese producto es cero, excepto cuando $u = x$ y $v = y$. Esto permite reconstruir una imagen como suma de sus píxeles:

$$f(x, y) = \sum_{(u,v)} f(u, v)\delta(x - u, y - v)$$

Si T es un sistema LSI y $h(x, y) = T[\delta(x, y)]$ entonces

$$\begin{aligned} g(x, y) = T[f(x, y)] &= T \left[\sum_{(u,v)} f(u, v)\delta(x - u, y - v) \right] \\ &= \sum_{(u,v)} T [f(u, v)\delta(x - u, y - v)] \\ &= \sum_{(u,v)} f(u, v)h(x - u, y - v) = f(x, y) * h(x, y) \end{aligned}$$

operación conocida como la convolución de la imagen f con la máscara o *kernel* h .

De forma equivalente al caso unidimensional, la ecuación de convolución

$$f(x, y) * h(x, y) = \sum_{(u,v)} f(u, v)h(x - u, y - v) \quad (3.1)$$

se puede interpretar geoméricamente siguiendo cuatro pasos

1. Invertir h tanto en x como en y , lo que equivale a una rotación del plano en 180° .
2. Trasladar h invertida de tal modo que su origen quede sobre el píxel de interés en (x, y)
3. Multiplicar punto a punto h invertida y trasladada con x
4. Sumar todos los productos anteriores y asignarlos al resultado $g(x, y)$.

Del mismo modo que con el caso unidimensional, se puede demostrar que la operación de convolución es conmutativa y asociativa.

Separabilidad

En el procesamiento de imágenes por medio de convolución debe considerarse el costo asociado a ésta operación. Asíumase que la imagen de entrada $f(x, y)$ tiene R filas y C columnas, y asíumase que el *kernel*, *núcleo* o *máscara* de convolución $h(x, y)$ es simétrica con N filas y M columnas con valores $x \in \{x_i, \dots, x_f\}$ y $y \in \{y_i, \dots, y_f\}$, tales que $N = y_f - y_i + 1$ y $M = x_f - x_i + 1$, de modo que la suma de convolución se puede reescribir como

$$f(x, y) * h(x, y) = \sum_{v=y-y_f}^{y-y_i} \sum_{u=x-x_f}^{x-x_i} f(u, v)h(x - u, y - v) \quad (3.2)$$

donde en los índices de las sumas aseguran que la máscara tenga únicamente índices válidos. Sea $K = NM$ el tamaño de la máscara en píxeles, y $S = RC$ el tamaño de la

imagen en píxeles. De la ecuación (3.2) se deduce que para cada píxel se requieren K productos más $K-1$ sumas, por lo que el total de productos requeridos para convolucionar toda la imagen es SK y el total de sumas $S(K-1)$.

En el caso de que la máscara sea separable, es decir, que se puede expresar

$$h(x, y) = h_h(x)h_v(y)$$

entonces (3.2) se puede reescribir como

$$\begin{aligned} f(x, y) * h(x, y) &= \sum_{v=y-y_f}^{y-y_i} \sum_{u=x-x_f}^{x-x_i} f(u, v)h_h(x-u)h(y-v) \\ &= \sum_{v=y-y_f}^{y-y_i} h(y-v) \sum_{u=x-x_f}^{x-x_i} f(u, v)h_h(x-u) \end{aligned}$$

Nótese que la suma

$$g_x(x, v) = \sum_{u=x-x_f}^{x-x_i} f(u, v)h_h(x-u)$$

corresponde a un filtrado unidimensional de cada fila v de la función $f(x, y)$. El filtrado de cada píxel en una fila requiere M productos y $M-1$ sumas, por lo que una fila requiere MC productos y $MC-1$ sumas. Filtrar todas las filas de esta manera implicará RCM productos y $R(MC-1)$ sumas. Falta realizar un último paso:

$$f(x, y) * h(x, y) = \sum_{v=y-y_f}^{y-y_i} h(y-v)g_x(x, v)$$

también equivalente a un filtrado unidimensional, ahora de cada columna del resultado parcial anterior, por separado. El filtrado de cada píxel de la columna requiere N productos y $N-1$ sumas, por lo que una columna requiere NR productos y $NR-1$ sumas. El total de columnas implicará entonces NRC productos y $C(NR-1)$ sumas. Agrupando ambos términos se obtiene un total de productos de la convolución de un núcleo separable de $S(N+M)$ y el total de sumas es $S(N+M) - (R+C)$.

Asumiendo una máscara cuadrada con $N=M$ y una imagen igualmente cuadrada con $R=C$, entonces el número de productos de la convolución bidimensional es R^2N^2 y $R^2(N^2-1)$ sumas, contra R^22N productos y R^22N-2R sumas. Se deduce entonces un crecimiento del número de productos cuadrático en N para la implementación directa y lineal en N para la implementación separable. Por ejemplo, para una imagen de 512×512 con una máscara de 7×7 se presenta una diferencia entre 12,84 millones de productos de la convolución directa contra 3,67 millones de la convolución separable: tan solo un 28,6% de las operaciones. Es por ello que siempre se intentan utilizar filtros separables en los procesos de convolución.

Por hacer:

Relación con la correlación

3.4.3 Filtros lineales

Existen gran variedad de filtros FIR en el procesamiento digital de imágenes. En esta sección se describen algunas típicas

Máscaras de suavizamiento

Filtros de media móvil Los filtros de media móvil, o promediadores, tienen una máscara cuadrada con valores todos iguales a $1/S$, donde S es el número de píxeles de la máscara.

Ésta máscara es trivialmente separable, pero existen algoritmos de implementación particularmente eficientes para su realización.

Aproximación del gradiente La información en imágenes se encuentra en zonas de cambio entre regiones homogéneas. La forma más elemental de detección de cambio en una función multidimensional radica en el cálculo de su gradiente. La magnitud del gradiente indica qué tan fuerte es el cambio. La dirección dada por el vector normalizado en magnitud, indica la dirección en el dominio de la función sobre la que ocurre el mayor aumento de la función.

El gradiente es una operación básica en la detección de características como bordes, esquinas, líneas, curvas, etc.

$$\nabla f = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}$$

El problema fundamental del cálculo del gradiente es que la función con que se cuenta está muestreada, y por tanto no es una función derivable. Lo único con que se cuenta entonces es mecanismos de aproximación, que pueden finalmente expresarse en términos de convoluciones.

Un problema asociado al cálculo del gradiente es que como detector de cambios, es muy sensible al ruido, y tiende a amplificarlo.

Por hacer:

Ampliación del ruido por gradientes: matemáticamente... correlaciones

Aproximación básica

$$\frac{\partial}{\partial x} f(x, y) \approx f(x, y) - f(x - 1, y)$$

$$\frac{\partial}{\partial y} f(x, y) \approx f(x, y) - f(x, y - 1)$$

que tiene el problema de que el resultado en realidad es la aproximación de las derivadas entre dos píxeles y no sobre ninguno de ellos.

Roberts Como el anterior pero diagonal. 2×2

Sobel Kernel muy empleado en la literatura $\begin{bmatrix} 1 & 2 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

Prewitt La aproximación del gradiente puede considerar otras estrategias. Por ejemplo, se puede aproximar la función en cada píxel por una función cuadrática, y luego derivar dicha función.

Esto produce el kernel de Prewitt 3×3 , $\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \end{bmatrix}$

Robinson 3×3 non-separable kernel (see `lti::robinsonKernelX`)

Kirsch 3×3 non-separable kernel

Derivaciones orientadas de la función gaussiana Puesto que la derivación acentúa el ruido, es recomendable utilizar un filtro suavizador antes de la derivación:

$$g(x, y) = \nabla(h(x, y) * f(x, y))$$

Puesto que la convolución es conmutativa y asociativa, y la derivación es un operador lineal, lo anterior se puede reescribir como:

$$g(x, y) = f(x, y) \nabla(h(x, y))$$

Para la función gaussiana se pueden calcular $\nabla h(x, y)$ con el muestreo de la derivación de dicha gaussiana en x y en y . Cada componente es conocida como OGD (oriented gaussian derivative) en la dirección x y y respectivamente.

Kernels de Ando Kernels optimizados: se pueden optimizar para invarianza a rotación, para magnitud, para una función interpoladora específica, etc.

Laplaciano

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (3.3)$$

Aumento de nitidez

$$g(x, y) = f(x, y) + c\nabla^2 f(x, y)$$

3.4.4 Filtros de rango

Filtros no lineales, donde no se convolucionan, sino que se extraen los píxeles sobre la máscara, y se elige alguno de acuerdo a su posición en el arreglo ordenado de dichos píxeles.

Ventaja: mantienen los valores de los píxeles existentes en las imágenes.

3.4.5 Filtros IIR en imágenes

Por hacer:

Tema poco tratado, pero útil

Capítulo 4

Procesamiento en el dominio de la frecuencia

4.1 Frecuencia en señales continuas y discretas

El estudio clásico de señales se basa en su descomposición en señales elementales cuyo comportamiento es conocido o predecible en los sistemas con que se trabaja. En cursos anteriores se han introducido conceptos del análisis de Fourier, transformada de Laplace, donde las señales elementales, sean estas exponenciales complejas o sinusoides, se caracterizan por su frecuencia temporal y fase respecto a alguna referencia común. El concepto de frecuencia espacial será introducido en las siguientes secciones, donde se apreciarán además diferencias entre los dominios discreto y continuo. El punto de partida para el análisis serán las funciones sinusoidales, como representantes espectrales “puras”.

4.1.1 Señal sinusoidal continua

Una oscilación armónica simple se describe matemáticamente a través de la señal continua [58]:

$$x_a(s) = A \cos(\Omega s + \theta), \quad -\infty < s < \infty \quad (4.1)$$

donde el subíndice a indica que es una señal analógica y s denota desplazamiento espacial, A representa la *amplitud* de la señal, Ω es la *frecuencia angular espacial* en radianes por unidad de longitud y θ es la *fase* en radianes. Es también común utilizar la frecuencia F en ciclos por unidad de longitud con

$$\Omega = 2\pi F$$

con lo que (4.1) se puede expresar como

$$x_a(s) = A \cos(2\pi F s + \theta), \quad -\infty < s < \infty.$$

En el caso particular de procesamiento de imágenes, suele tomarse s en milímetros y F en líneas por milímetro, que guarda una relación de $1/1000$ con las unidades establecidas por el SI de ciclos por metro.

Si F es constante, entonces x_a es periódica

$$x_a(s + S_p) = x_a(s)$$

con período fundamental $S_p = 1/F$ (figura 4.1).

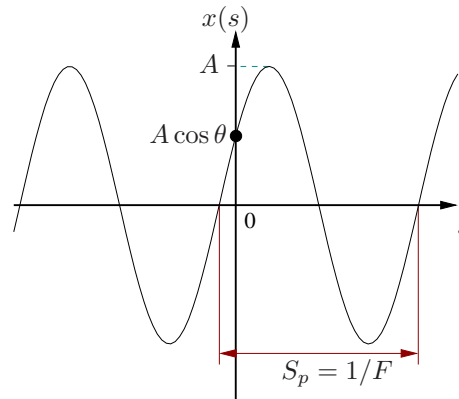


Figura 4.1: Ejemplo de una señal analógica sinusoidal en el espacio.

Este senoide está muy relacionado con la función exponencial compleja

$$x_a(s) = Ae^{j(\Omega s + \theta)}$$

a través de la identidad de Euler

$$e^{j\phi} = \cos \phi + j \operatorname{sen} \phi.$$

La frecuencia espacial en contextos físicos representa el número de ciclos u oscilaciones por unidad de longitud, motivo por el cual ésta solo puede tomar valores positivos; sin embargo, el utilizar frecuencias negativas trae consigo ciertas ventajas de manipulación matemática, como lo muestra el ejemplo 4.1.

Ejemplo 4.1 Demuestre que la ecuación (4.1) puede ser expresada como la suma de dos señales exponenciales conjugadas.

Solución:

Utilizando la identidad de Euler se tiene que

$$\begin{aligned} e^{j\phi} + e^{-j\phi} &= \cos \phi + j \operatorname{sen} \phi + \cos(-\phi) + j \operatorname{sen}(-\phi) \\ &= \cos \phi + j \operatorname{sen} \phi + \cos \phi - j \operatorname{sen} \phi \\ &= 2 \cos \phi \end{aligned}$$

reemplazando $\phi = \Omega s + \theta$ en el resultado anterior y multiplicando ambos lados por la amplitud A se obtiene finalmente

$$A \cos(\Omega s + \theta) = \frac{A}{2} e^{j(\Omega s + \theta)} + \frac{A}{2} e^{-j(\Omega s + \theta)}$$

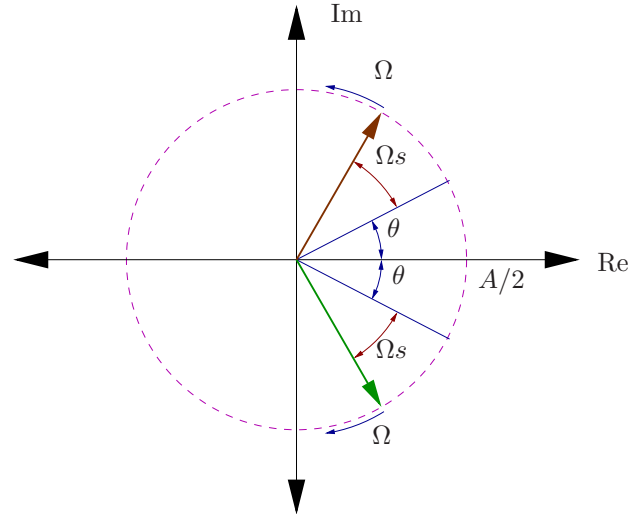


Figura 4.2: Representación de la función cosenoidal con desfase como la suma de dos funciones exponenciales complejas conjugadas.

El primer término del lado derecho de la ecuación se puede interpretar como un fasor de magnitud $A/2$ que gira en sentido contrario a las manecillas del reloj (frecuencia positiva), mientras que el lado izquierdo es otro fasor que gira en el sentido de las manecillas del reloj (frecuencia negativa), ambos a una frecuencia de F ciclos por unidad de longitud (figura 4.2). 4.1

4.1.2 Señal sinusoidal discreta

La señal sinusoidal en tiempo discreto se expresa como

$$x(n) = A \cos(\omega n + \theta), \quad -\infty < n < \infty \quad (4.2)$$

con la variable entera $n \in \mathbb{Z}$ (también denominada *número de muestra*), A la magnitud, ω es la frecuencia en radianes por muestra y θ la fase en radianes. De forma similar al caso continuo, puede utilizarse $\omega = 2\pi f$ para reescribir (4.2) como

$$x(n) = A \cos(2\pi f n + \theta), \quad -\infty < n < \infty. \quad (4.3)$$

En este caso las dimensiones de la frecuencia son ciclos por muestra, y tal como estas unidades lo indican, usualmente tiene valores menores que la unidad. La figura 4.3 muestra un ejemplo con $f = 1/12$ y $\theta = \pi/3$.

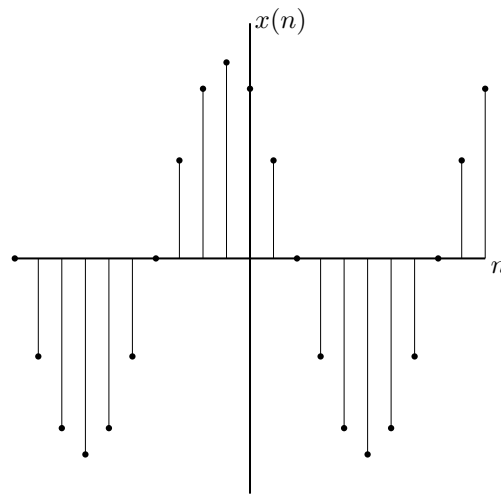


Figura 4.3: Ejemplo de una señal sinusoidal discreta ($\omega = \pi/6$ y $\theta = \pi/3$) [58].

Periodicidad de un senoide discreto

Por definición una señal de variable discreta $x(n)$ es periódica con periodo N ($N > 0$) si y solo si

$$x(n + N) = x(n) \quad \text{para todo } n \quad (4.4)$$

El menor valor de N para el que se cumple (4.4) se denomina *periodo fundamental*. Una señal sinusoidal de frecuencia f_0 es periódica si

$$\cos(2\pi f_0(N + n) + \theta) = \cos(2\pi f_0 n + \theta)$$

lo que se cumple solo si existe un entero k tal que

$$2\pi f_0 N = 2k\pi$$

o, en otros términos

$$f_0 = \frac{k}{N} \quad (4.5)$$

que es siempre un número racional. Esto quiere decir, que una señal sinusoidal discreta con un valor irracional de frecuencia ¡no es periódica!

Para determinar el periodo fundamental de una señal sinusoidal se expresa su frecuencia como en (4.5) y se simplifican los factores comunes de tal forma que k y N formen números primos relativos. Por ejemplo si $f_1 = 5/32 = 0,15625$ el periodo fundamental es $N = 32$, pero si la frecuencia es $f_2 = 4/32 = 0,125$ el periodo fundamental es $N = 8$ (figura 4.4). Esto quiere decir que aún cuando los valores de frecuencia se encuentren relativamente cerca, sus periodos pueden cambiar radicalmente.

Equivalencia de frecuencias en sinusoides discretos

Considérese de nuevo la señal sinusoidal $\cos(\omega_0 n + \theta)$. Es fácil de obtener que:

$$\cos((\omega_0 + 2\pi)n + \theta) = \cos(\omega_0 n + 2\pi n + \theta) = \cos(\omega_0 n + \theta)$$

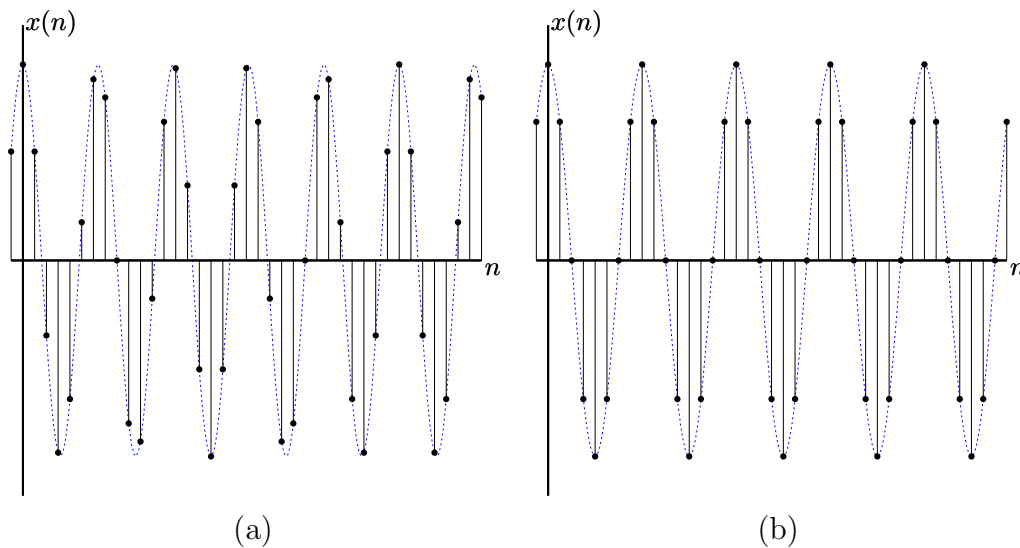


Figura 4.4: Comparación de dos frecuencias cercanas en señales de variable discreta. (a) Frecuencia $f = 5/32$, periodo $N = 32$. (b) Frecuencia $f = 4/32$, periodo $N = 8$. La línea punteada denota una señal continua con frecuencia equivalente para simplificar la comparación.

por lo que todas las secuencias sinusoidales

$$x_k(n) = A \cos(\omega_k n + \theta), \quad k = 0, 1, 2, \dots$$

con

$$\omega_k = \omega_0 + 2k\pi$$

son idénticas. Por otro lado, las secuencias de cualesquiera dos señales sinusoidales discretas con frecuencias en el rango $-\pi \leq \omega \leq \pi$ o $-\frac{1}{2} \leq f \leq \frac{1}{2}$ son diferentes. Combinando los resultados anteriores se obtiene que cualquier secuencia sinusoidal de frecuencia $|\omega| > \pi$ o $|f| > \frac{1}{2}$ tiene una señal equivalente con $|\omega| \leq \pi$ o $|f| \leq \frac{1}{2}$. A las frecuencias $|\omega| \leq \pi$ o $|f| \leq \frac{1}{2}$ se les considera *frecuencias fundamentales* y a las frecuencias $|\omega| > \pi$ o $|f| > \frac{1}{2}$ se les denomina *alias*.

Tasa máxima de oscilación

Considérese ahora la secuencia sinusoidal $x_0(n) = \cos(\omega_0 n)$ para el rango de frecuencias $\omega_0 \in [0, \pi]$. La figura 4.5 muestra algunos casos particulares como ejemplo. Se puede apreciar que la tasa de oscilación aumenta conforme la frecuencia aumenta.

Ahora, para analizar el caso de frecuencias angulares mayores que π considérese el caso especial de $\omega_1 = 2\pi - \omega_0$. Si $\omega_0 \in [0, \pi]$ entonces $\omega_1 \in [\pi, 2\pi]$ de tal forma que si ω_0 aumenta ω_1 disminuye.

Debido a que

$$x_1(n) = A \cos(\omega_1 n) = A \cos((2\pi - \omega_0)n) = A \cos(-\omega_0 n) = x_0(n)$$

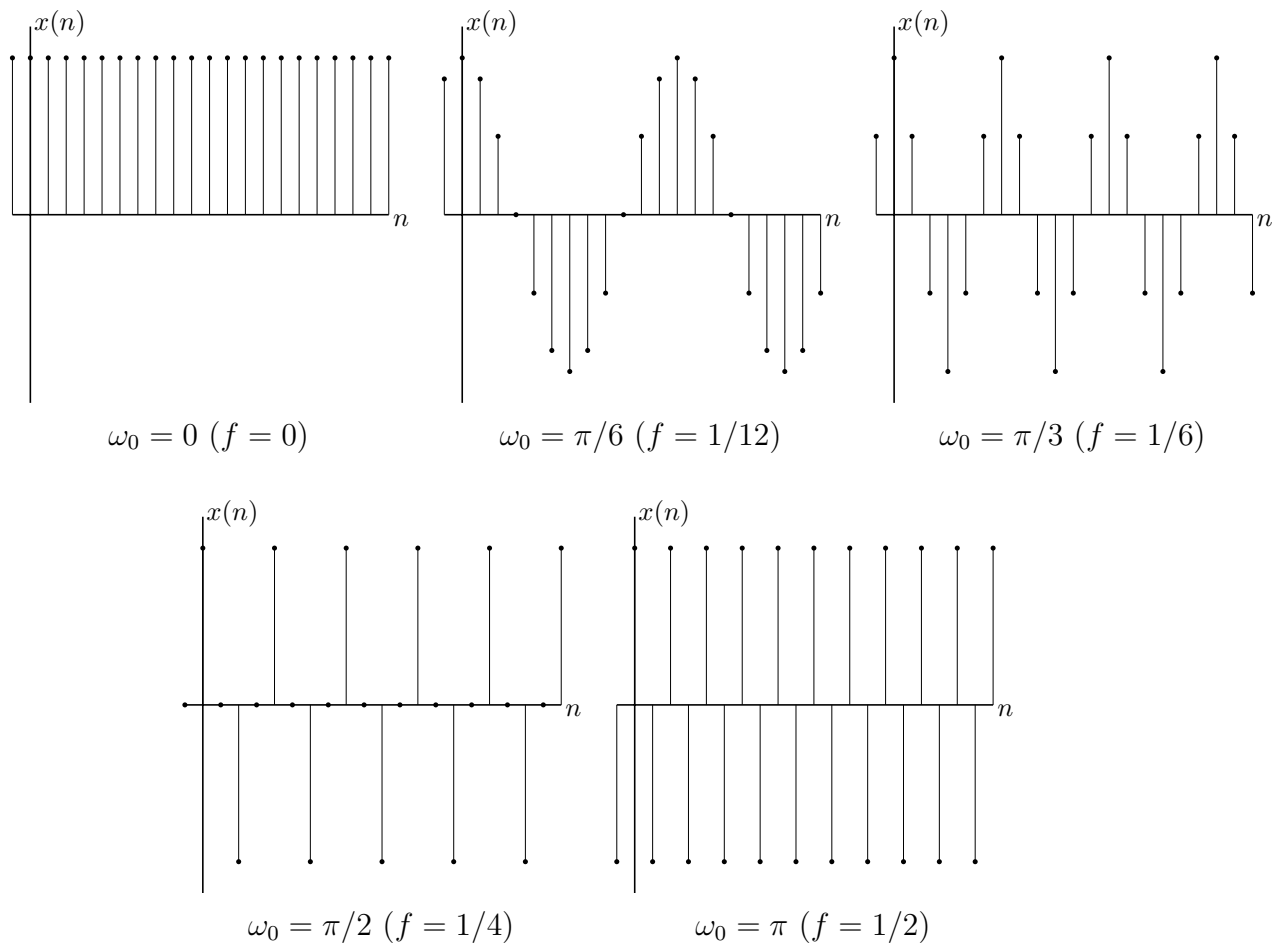


Figura 4.5: Secuencia sinusoidal con diferentes frecuencias.

la frecuencia angular ω_1 es un alias de ω_0 . De aquí se concluye que si la frecuencia angular aumenta de π hacia 2π la tasa de oscilación se reduce, al representar estos alias frecuencias que bajan de π a 0.

De forma similar al caso de senoides de variable continua puede utilizarse la identidad de Euler en el caso discreto para introducir el concepto de frecuencia negativa:

$$x(n) = A \cos(\omega n + \theta) = \frac{A}{2} e^{j(\omega n + \theta)} + \frac{A}{2} e^{-j(\omega n + \theta)}$$

Debido a que las señales sinusoidales de variable discreta con frecuencias separadas por un entero múltiplo de 2π son idénticas, entonces todas las frecuencias en un intervalo $[\omega_1, \omega_1 + 2\pi]$ representan todas las frecuencias existentes para estas señales. En otras palabras, el rango de frecuencias distinguibles para sinusoides de variable discreta es finito con un ancho de banda de 2π . Usualmente se utilizan los rangos de frecuencias angulares $\omega \in [-\pi, \pi]$ ($f \in [-\frac{1}{2}, \frac{1}{2}]$) o $\omega \in [0, 2\pi]$ ($f \in [0, 1]$) y reciben el nombre de *rango fundamental*.

4.1.3 Exponenciales complejas relacionadas armónicamente

Las señales de variable continua

$$\epsilon_k(s) = e^{jk\Omega_0 s} = e^{j2\pi k F_0 s} \quad k = 0, \pm 1, \pm 2, \dots$$

se denominan señales exponenciales relacionadas armónicamente. El periodo fundamental de la señal $\epsilon_k(s)$ es $1/(kF_0) = S_p/k$, lo que equivale a una frecuencia fundamental de kF_0 . Puesto que una señal periódica con periodo S_p/k es también periódica con periodo $k(S_p/k) = S_p$ con $k \in \mathbb{Z}$ entonces todas las señales $\epsilon_k(s)$ tienen como periodo común S_p . El nombre proviene de la relación de estas funciones con las oscilaciones de una cuerda, que tienen relaciones “armónicas” (mantienen los nodos externos fijos) cuando las frecuencias son múltiplos enteros de una frecuencia fundamental.

En este caso de variable continua, para $k_1 \neq k_2$ se cumple siempre que $\epsilon_{k_1}(s) \neq \epsilon_{k_2}(s)$, o en otros términos, existe un número infinito de señales complejas relacionadas armónicamente con $e^{j2\pi F_0 s}$.

Para el caso de señales exponenciales complejas de variable discreta, puesto que son señales periódicas solo si su frecuencia es racional, se escoge $f_0 = 1/N$ y se define el conjunto de señales relacionadas armónicamente como

$$\epsilon_k(n) = e^{jk\omega_0 n} = e^{j2\pi k f_0 n} \quad k = 0, \pm 1, \pm 2, \dots \quad (4.6)$$

A diferencia del caso continuo se tiene para $k_1 = k + N$

$$\epsilon_{k+N}(n) = e^{j2\pi \frac{(k+N)}{N} n} = e^{j2\pi \frac{kn}{N}} e^{j2\pi n} = e^{j2\pi \frac{kn}{N}} = \epsilon_k(n)$$

Esto significa que en el caso discreto solo existen N señales complejas relacionadas armónicamente, donde todos los miembros del conjunto descrito en (4.6) tienen como periodo común N muestras.

4.1.4 Muestreo de señales analógicas

El muestreo espacial realizado por chips CCD o CMOS es *muestreo periódico* o *uniforme*, con la relación entre la señal analógica y la señal de variable discreta dada por

$$x(n) = x_a(nS) \quad -\infty < n < \infty$$

donde la secuencia $x(n)$ contiene entonces muestras de la señal analógica $x_a(s)$ separadas por un intervalo S (figura 4.6).

Las variables s y n de las señales de variable continua y discreta respectivamente están relacionadas a través del intervalo de muestreo S

$$s = nS = n/F_m$$

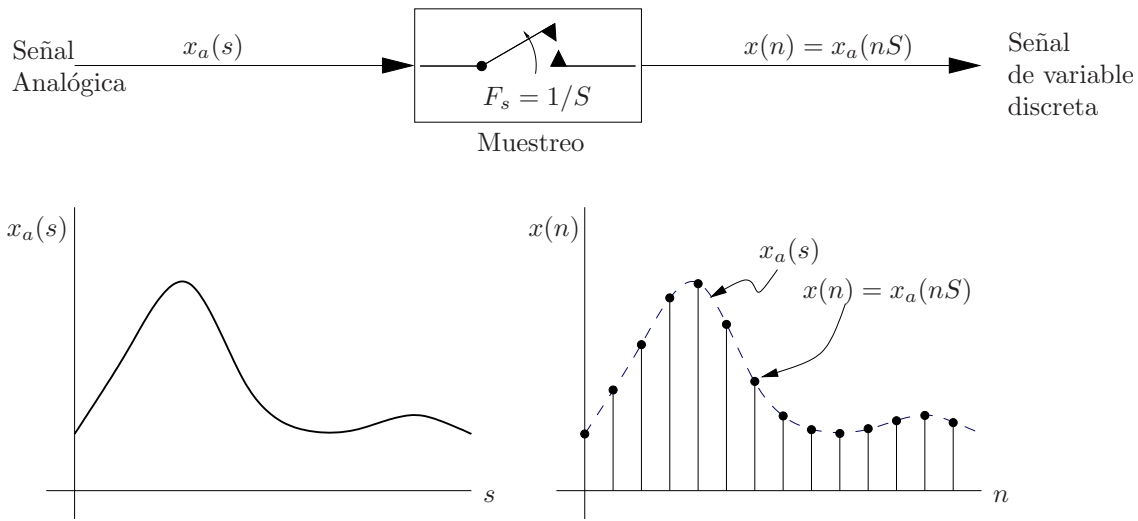


Figura 4.6: Muestreo periódico de una señal analógica.

donde a F_m se le denomina tasa de muestreo espacial. Para encontrar la relación entre las frecuencias de las señales en ambos dominios considérese la señal analógica

$$x_a(s) = A \cos(2\pi F s + \theta)$$

la cual, muestreada periódicamente con una tasa $F_m = 1/S$ resulta

$$\begin{aligned} x_a(nS) \equiv x(n) &= A \cos(2\pi F S n + \theta) \\ &= A \cos\left(\frac{2\pi F n}{F_m} + \theta\right) \end{aligned}$$

y comparando con (4.3) se obtiene

$$\omega = \Omega S$$

o

$$f = \frac{F}{F_m} \quad (4.7)$$

lo que justifica que a f se le denomine también *frecuencia espacial normalizada*. Con esto queda claro que la frecuencia f puede corresponder con las unidades de frecuencia de F (por ejemplo, en líneas por milímetro) si y solo si se conoce la frecuencia de muestreo F_m .

En las secciones anteriores se obtuvo que el rango para la frecuencia F es prácticamente infinito, mientras que el rango de la frecuencia f es finito $([-\frac{1}{2}, \frac{1}{2}])$. Utilizando (4.7) se obtiene

$$\begin{aligned} -\frac{1}{2} &< f \leq \frac{1}{2} \\ -\frac{1}{2} &< \frac{F}{F_m} \leq \frac{1}{2} \\ -\frac{F_m}{2} &< F \leq \frac{F_m}{2} \end{aligned}$$

El muestreo periódico de la señal de variable continua representa entonces un mapeo del rango infinito de frecuencias F a un rango finito de frecuencias f . Puesto que la mayor

frecuencia de f es $f = \frac{1}{2}$ puede concluirse que con la tasa de muestreo F_m como máximo se podrá representar

$$F_{max} = \frac{F_m}{2} = \frac{1}{2S}$$

$$\Omega_{max} = \pi F_m = \frac{\pi}{S}$$

En los párrafos anteriores se demostró que las frecuencias f son únicas dentro de un rango finito: el rango fundamental, y que todas las frecuencias fuera de ese rango son alias de alguna frecuencia fundamental. Se demostró además que para una señal sinusoidal la frecuencia angular ω y $\omega + 2k\pi$ son equivalentes para todo $k \in \mathbb{Z}$, lo que implica que $f + k$ es un alias de f . Con (4.7) se tiene que

$$f + k = \frac{F}{F_m} \Rightarrow fF_m + kF_m = F$$

o en otras palabras $fF_m + kF_m$ es un alias de $F = fF_m$. Con esto, la relación entre las frecuencias de las señales de variable continua y discreta puede graficarse como lo muestra la figura 4.7.

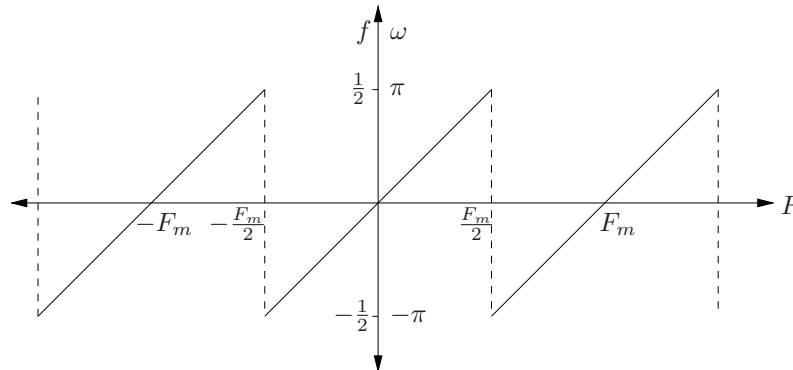


Figura 4.7: Relación entre las frecuencias de señales de variable discreta y continua para el caso de muestreo uniforme.

La figura 4.8 muestra un ejemplo con las frecuencias $F_1 = 2/7$ y su alias $F_2 = -5/7$ para una tasa de muestreo de $F_m = 1$ ciclo/unidad. Aquí se aprecia con claridad que las muestras tomadas para las dos señales coinciden, por lo que estas muestras por si solas no pueden determinar de cuál de las dos señales fue muestreada.

La frecuencia $F_m/2$ corresponde entonces a la frecuencia máxima representable en forma discreta con una tasa de muestreo F_m , que corresponde entonces a $f = \frac{1}{2}$. La frecuencia fundamental correspondiente a un alias de frecuencia mayor a $F_m/2$ se puede terminar utilizando $F_m/2$ como pivote para reflejar el alias dentro del rango fundamental. A $F_m/2$ (o $\omega = \pi$) se le denomina entonces *frecuencia de plegado*, o en inglés *folding frequency*.

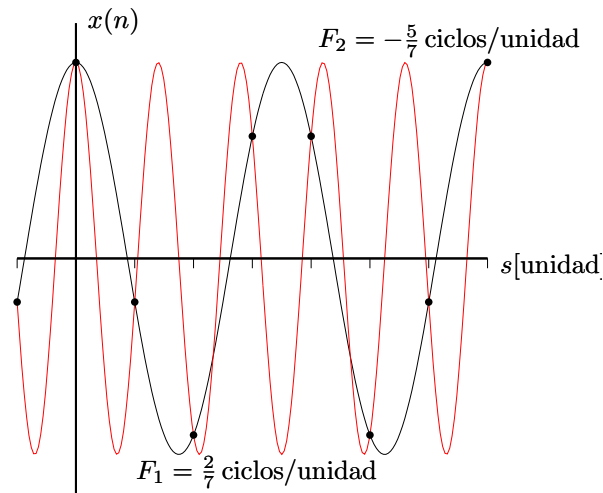


Figura 4.8: Muestreo de una señal sinusoidal y de un alias.

4.2 Análisis discreto de Fourier

4.2.1 Serie generalizada de Fourier

Un conjunto (posiblemente) infinito de funciones ortogonales puede servir de base generadora de un espacio funcional. Sea \mathcal{U} un conjunto de funciones ortogonales $\mathcal{U} = \{u_{n_1}(s), \dots, u_0(s), u_1(s), \dots, u_{n_2}(s)\}$. Este conjunto puede utilizarse como conjunto generador de un espacio funcional para toda función

$$x_m(s) = \sum_{i=n_1}^{n_2} c_i u_i(s) \quad (4.8)$$

donde c_i representa los coeficientes escalares de la combinación lineal de las funciones generadoras $u_i(s)$.

Si se desea aproximar una función $x(s)$ en un intervalo (s_1, s_2) con la combinación lineal $x_m(s)$ de $m = n_2 - n_1 + 1$ funciones generadoras definidas en ese mismo intervalo, puede hacerse uso de la ortogonalidad de las funciones para obtener los coeficientes c_i . A partir de la representación de la función $x(s) \approx x_m(s)$ como serie (ver ecuación (4.8)), se evalúa el producto interno por una función generadora particular $u_k(s)$ para obtener

$$\begin{aligned} \langle u_k(s), x(s) \rangle &\approx \left\langle u_k(s), \sum_{i=n_1}^{n_2} c_i u_i(s) \right\rangle \\ &= \sum_{i=n_1}^{n_2} \langle u_k(s), c_i u_i(s) \rangle \\ &= \sum_{i=n_1}^{n_2} c_i \langle u_k(s), u_i(s) \rangle \\ &= c_k \langle u_k(s), u_k(s) \rangle \end{aligned}$$

con lo que se deriva

$$c_k = \frac{\langle u_k(s), x(s) \rangle}{\|u_k(s)\|^2}. \quad (4.9)$$

Si la base funcional $\{u_k(s)\}$, $k \in \mathbb{Z}$ es completa, es decir, si la aproximación de la función $x(s)$ con la serie infinita converge a la función:

$$x(s) = \sum_{k=-\infty}^{\infty} c_k u_k(s) \quad (4.10)$$

con las funciones generadoras $u_k(s)$ ortogonales y los coeficientes c_k calculados con (4.9), entonces a la expansión en serie se le denomina *serie generalizada de Fourier*. A (4.9) se le conoce como ecuación de *análisis* y a (4.10) como ecuación de *síntesis*.

4.2.2 Serie discreta de Fourier

En el caso de las funciones en espacio continuo, las series de Fourier emplean como base generadora al conjunto de las funciones exponenciales armónicamente relacionadas $e^{jk\omega_0 s}$, donde k puede crecer indefinidamente al existir infinitas frecuencias correspondientes a los infinitos múltiplos enteros de periodos $2\pi/\omega_0$.

Para señales de naturaleza discreta solo existe unicidad para frecuencias normalizadas en el intervalo $] -1/2; 1/2]$, ó $[0; 1[$. Una señal discreta periódica con periodo fundamental N puede tener componentes frecuenciales separadas por $\omega = 2\pi/N$ radianes o $f = 1/N$ ciclos, que se representan por los exponenciales complejos armónicamente relacionados:

$$\epsilon_k(n) = e^{j2\pi kn/N}, \quad k = 0 \dots N-1$$

Utilizando series generalizadas de Fourier con estas funciones base se obtiene para la síntesis de una señal:

$$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \quad (4.11)$$

y considerando que

$$\sum_{n=0}^{N-1} a^n = \begin{cases} N & a = 1 \\ \frac{1-a^N}{1-a} & a \neq 1 \end{cases}$$

se puede afirmar para las exponenciales complejas que el producto interno entre dos de ellas está dado por

$$\langle \epsilon_l(n), \epsilon_k(n) \rangle = \sum_{n=0}^{N-1} e^{-j2\pi ln/N} e^{j2\pi kn/N} = \begin{cases} N & k = l \\ 0 & k \neq l \end{cases}$$

tomando $a = e^{j2\pi(k-l)/N}$ por lo que $a^N = 1$.

Calculando el producto punto entre $\langle \epsilon_l(n), x(n) \rangle$ se obtiene:

$$\begin{aligned} \sum_{n=0}^{N-1} x(n) e^{-j2\pi ln/N} &= \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} e^{-j2\pi ln/N} \\ &= \sum_{k=0}^{N-1} c_k \sum_{n=0}^{N-1} e^{j2\pi(k-l)n/N} \\ &= N c_l \end{aligned}$$

y finalmente

$$c_l = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi ln/N}, \quad l = 0, 1, \dots, N-1$$

Nótese que en esta serie discreta de Fourier el coeficiente c_k representa la amplitud y fase de la componente frecuencial $\epsilon_k(n) = e^{j2\pi kn/N} = e^{j\omega_k n}$ con frecuencia angular normalizada $\omega_k = 2\pi k/N$ (o lo que es equivalente $f_k = k/N$). Puesto que $\epsilon_k(n)$ es periódica al ser f_k un número racional ($\epsilon_k(n) = \epsilon_k(n+N)$), entonces los coeficientes c_k también son periódicos con periodo fundamental N :

$$c_{k+N} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi(k+N)n/N} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} e^{-j2\pi n} = c_k$$

Por conveniencia se utiliza para c_k normalmente el intervalo $k = 0 \dots N-1$.

La relación de Parseval es en este caso

$$\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |c_k|^2$$

es decir, la potencia media de la señal es igual a la suma de las potencias medias de cada componente en el dominio de la frecuencia.

4.2.3 Transformada de Fourier de señales discretas

La transformada de Fourier de una señal de energía finita en tiempo discreto $x(n)$ se define como:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$$

Nótese que $X(\omega) = X(\omega + 2\pi k)$, lo que implica que el rango de frecuencias únicas se limita a $[0, 2\pi[$, o de forma equivalente a $]-\pi, \pi]$, que contrasta con el rango $[-\infty, \infty]$ de las señales continuas.

Como la señal $x(n)$ es discreta, una sumatoria reemplaza la integral del caso continuo. Puede demostrarse además que

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega.$$

La transformada de Fourier converge si

$$|X(\omega)| = \left| \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \right| \leq \sum_{n=-\infty}^{\infty} |x(n)| < \infty$$

es decir, si la señal es absolutamente sumable.

La relación de Parseval en este caso es:

$$E_x = \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 dt$$

Para señales reales, el espectro $X(\omega)$ tiene magnitud par y fase impar, es decir, se cumple que $X(\omega) = X^*(-\omega)$, por lo que usualmente basta determinar el espectro para $\omega \in [0, \pi]$, pues la contraparte puede generarse por simetría.

Debe notarse la correspondencia entre periodicidad en un dominio y la naturaleza discreta de la representación en el dominio complementario. Es decir, si una señal es periódica, su espectro será discreto y si el espectro es periódico, es porque la señal es discreta. Si la distancia entre las muestras o la líneas espectrales es α entonces la periodicidad en el otro dominio será $1/\alpha$.

$$\begin{aligned} \text{periódico} &\longleftrightarrow \text{discreto} \\ \text{aperiódico} &\longleftrightarrow \text{continuo} \end{aligned}$$

Por ejemplo, una señal continua periódica tendrá un espectro aperiódico discreto; una señal discreta aperiódica tendrá un espectro periódico continuo.

4.2.4 El teorema del muestreo

La máxima frecuencia normalizada representable en una señal muestreada es $f_{max} = \frac{F_{max}}{F_m} = \frac{1}{2}$. Puesto que la representación de Fourier de la señal discreta es

$$x(n) \circ \bullet X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}, \quad \omega \in]-\pi, \pi]$$

y $X(\omega)$ es periódica con periodo fundamental 2π , se obtiene que la frecuencia mínima de muestreo F_m debe ser elegida de tal modo que sea al menos el doble de la frecuencia máxima de la señal (figura 4.9).

De otro modo existirá entre las repeticiones periódicas del espectro de la señal analógica un traslape, denominado *aliasing*. Sea $x_a(s)$ la señal analógica muestreada periódicamente cada S unidades de longitud para producir la señal de variable discreta $x(n)$:

$$x(n) = x_a(nS), \quad -\infty < n < \infty.$$

Si $x_a(s)$ es aperiódica de energía finita, entonces se cumple para su espectro:

$$X_a(F) = \int_{-\infty}^{\infty} x_a(s)e^{-j2\pi Fs} dt \bullet \circ x_a(s) = \int_{-\infty}^{\infty} X_a(F)e^{j2\pi Fs} dF$$

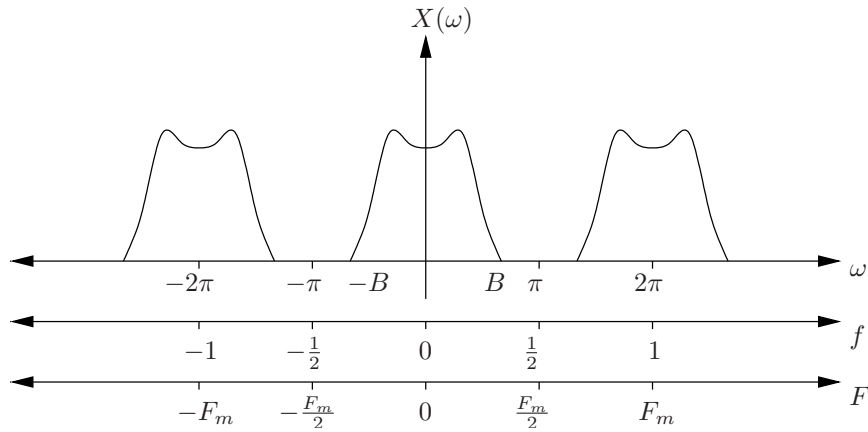


Figura 4.9: Espectro de señal discreta con banda limitada sin aliasing.

En el dominio discreto se tiene:

$$X(f) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi f n} \bullet \circ x(n) = \int_{-\frac{1}{2}}^{\frac{1}{2}} X(f) e^{j2\pi f n} df$$

Puesto que para la señal muestreada se tiene

$$s = nS = \frac{n}{F_m}$$

entonces se cumple que

$$x(n) = x_a(nS) = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi n F / F_m} dF \stackrel{!}{=} \int_{-1/2}^{1/2} X(f) e^{j2\pi f n} df$$

Considerando que $f = F/F_m$ y $df = dF/F_m$ se obtiene entonces

$$\frac{1}{F_m} \int_{-F_m/2}^{F_m/2} X(F/F_m) e^{j2\pi n F / F_m} dF = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi n F / F_m} dF$$

El lado derecho se puede segmentar en bloques de ancho F_m de la siguiente manera:

$$\int_{-\infty}^{\infty} X_a(F) e^{j2\pi n F / F_m} dF = \sum_{k=-\infty}^{\infty} \int_{(k-1/2)F_m}^{(k+1/2)F_m} X_a(F) e^{j2\pi n F / F_m} dF$$

Con un cambio de variable en la integral $F' = F - kF_m$, $dF' = dF$, se obtiene un desplazamiento del k -ésimo bloque al intervalo $[-F_m/2, F_m/2]$:

$$\begin{aligned} \int_{(k-1/2)F_m}^{(k+1/2)F_m} X_a(F) e^{j2\pi n F / F_m} dF &= \int_{-F_m/2}^{F_m/2} X_a(F' + kF_m) e^{j2\pi n \frac{F' + kF_m}{F_m}} dF' \\ &= \int_{-F_m/2}^{F_m/2} X_a(F + kF_m) e^{j2\pi n F / F_m} dF \end{aligned}$$

por lo que

$$\begin{aligned} \frac{1}{F_m} \int_{-F_m/2}^{F_m/2} X\left(\frac{F}{F_m}\right) e^{j2\pi n F/F_m} dF &= \sum_{k=-\infty}^{\infty} \int_{-F_m/2}^{F_m/2} X_a(F + kF_m) e^{j2\pi n F/F_m} dF \\ &= \int_{-F_m/2}^{F_m/2} \left[\sum_{k=-\infty}^{\infty} X_a(F + kF_m) \right] e^{j2\pi n F/F_m} dF \end{aligned}$$

es decir

$$X\left(\frac{F}{F_m}\right) = X(f) = F_m \sum_{k=-\infty}^{\infty} X_a((f - k)F_m)$$

Nótese que el espectro de la señal discreta $X(f)$ es igual a la repetición periódica con periodo F_m del espectro escalado $F_m X_a(F)$.

Si el espectro de la señal analógica es de banda limitada, es decir, $X_a(F) = 0$ si $|F| \geq B$, entonces si se elige F_m mayor a $2B$ se cumple para el rango de frecuencias únicas $|F| \leq F_m/2$:

$$X\left(\frac{F}{F_m}\right) = F_m X_a(F), \quad |F| \leq F_m/2.$$

En este caso no existe aliasing y los espectros de las señales continua y discreta son idénticos en el intervalo de frecuencias $|F| \leq F_m/2$, exceptuando el factor de escala F_m (figura 4.10).

Si $F_m < 2B$ entonces el solapamiento espectral impide que la señal original pueda ser recuperada a partir de las muestras. Si no hay solapamiento, entonces

$$X_a(F) = \begin{cases} \frac{1}{F_m} X\left(\frac{F}{F_m}\right) & |F| \leq F_m/2 \\ 0 & |F| > F_m/2 \end{cases}$$

y puesto que

$$X\left(\frac{F}{F_m}\right) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi F n/F_m}$$

y además

$$x_a(s) = \int_{-F_m/2}^{F_m/2} X_a(F) e^{j2\pi F s} dF$$

se tiene que

$$\begin{aligned} x_a(s) &= \frac{1}{F_m} \int_{-F_m/2}^{F_m/2} \left[\sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi F n/F_m} \right] e^{j2\pi F s} dF \\ &= \frac{1}{F_m} \sum_{n=-\infty}^{\infty} x(n) \int_{-F_m/2}^{F_m/2} e^{j2\pi F (s - \frac{n}{F_m})} dF \end{aligned}$$

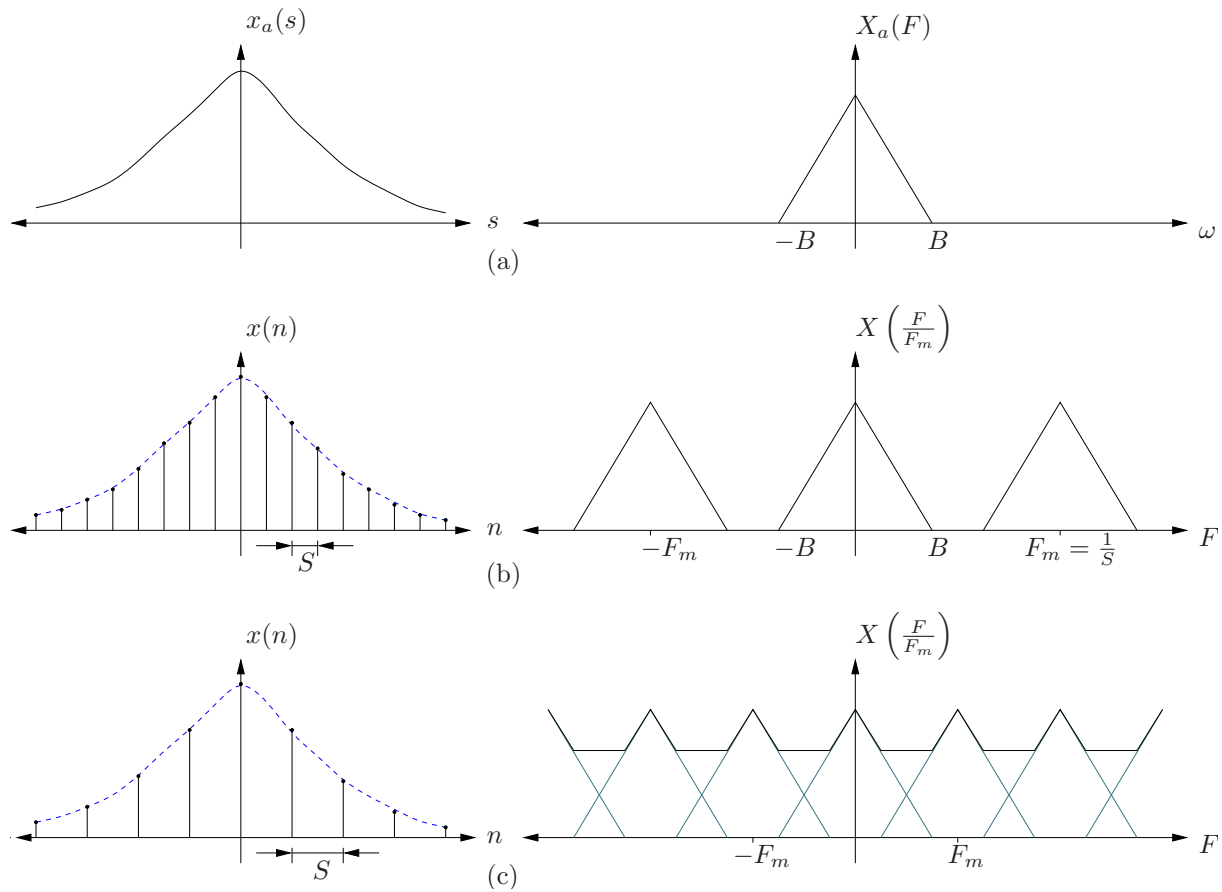


Figura 4.10: Señal analógica de espectro finito y los efectos de diferentes frecuencias de muestreo. (a) Señal analógica $x_a(s)$ y su espectro. (b) Muestreo con $F_m > 2B$. (c) Muestreo con $F_m < 2B$.

y puesto que

$$\begin{aligned}
 \int_{-F_m/2}^{F_m/2} e^{j2\pi F(s - \frac{n}{F_m})} dF &= \frac{e^{j2\pi F(s - \frac{n}{F_m})}}{j2\pi(s - \frac{n}{F_m})} \Big|_{-F_m/2}^{F_m/2} = \frac{e^{j\pi F_m(s - \frac{n}{F_m})} - e^{-j\pi F_m(s - \frac{n}{F_m})}}{j2\pi(s - \frac{n}{F_m})} \\
 &= \frac{\text{sen}\left(\pi F_m\left(s - \frac{n}{F_m}\right)\right)}{\pi\left(s - \frac{n}{F_m}\right)} = \frac{F_m \text{sen}\left(\pi F_m\left(s - \frac{n}{F_m}\right)\right)}{\pi F_m\left(s - \frac{n}{F_m}\right)} \\
 &= F_m \text{sa}\left(\pi F_m\left[s - \frac{n}{F_m}\right]\right)
 \end{aligned}$$

entonces

$$\begin{aligned}
 x_a(s) &= \sum_{n=-\infty}^{\infty} x(n) \text{sa}\left(\pi F_m\left[s - \frac{n}{F_m}\right]\right) \\
 &= \sum_{n=-\infty}^{\infty} x_a(nS) \text{sa}\left(\pi F_m[s - nS]\right)
 \end{aligned}$$

que es la interpolación de las muestras utilizando el interpolador ideal

$$g(s) = \frac{\text{sen}\left(\frac{\pi s}{S}\right)}{\pi \frac{s}{S}} = \text{sa}\left(\pi \frac{s}{S}\right). \quad (4.12)$$

Nótese que $g(s)$ es cero para $s = kS$, $k \in \mathbb{Z} \setminus 0$. En otras palabras, si $x_a(s)$ es de banda limitada B y $x(n) = x_a(nS)$ es muestreada con $F_m \geq 2B$ entonces $x_a(s)$ puede ser reconstruida completamente y de forma única utilizando el interpolador $g(s)$ dado en (4.12). Debe advertirse, sin embargo, que $g(s)$ tiene extensión infinita y no es causal, por lo que en aplicaciones prácticas suele aproximarse con interpoladores finitos. La figura 4.11 muestra un ejemplo de interpolación de una secuencia finita de muestras $x(n) = \{0, 3, 2, -1, 1, 0\}$. Se aprecia que la función interpolada atraviesa todas las muestras, mientras que para valores de t no enteros todas las muestras contribuyen al valor interpolado.

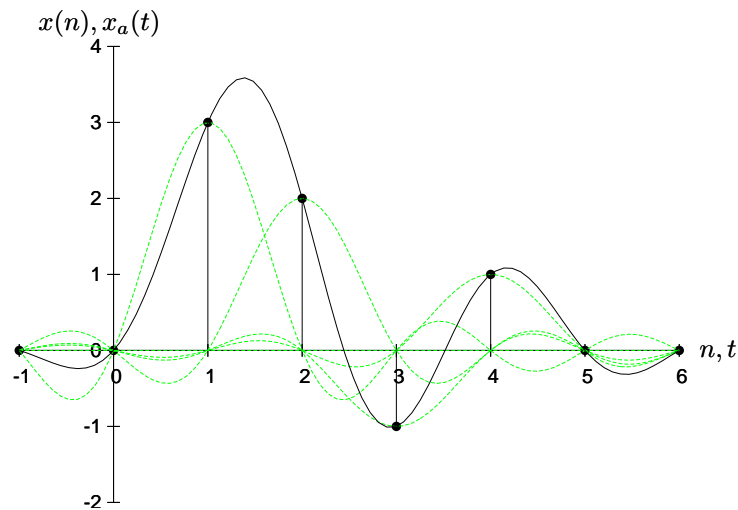


Figura 4.11: Ejemplo de interpolación ideal para una secuencia de muestras finita.

4.2.5 Propiedades de la transformada de Fourier de señales discretas

Simetría: Anteriormente se discutió el hecho de que cualquier señal puede representarse como la suma de una componente par y otra impar. Esto se puede aplicar tanto a la parte real como a la imaginaria del espectro y de su señal. Se puede demostrar que la relación entre estas componentes es la siguiente:

$$\begin{array}{ccccccccc} x(n) & = & [x_R^e(n) & + & jx_I^e(n)] & + & [x_R^o(n) & + & jx_I^o(n)] \\ \circ & & \circ & & \circ & & \circ & & \circ \\ \bullet & & \bullet & & \bullet & & \bullet & & \bullet \\ X(\omega) & = & [X_R^e(\omega) & + & jX_I^e(\omega)] & + & [jX_I^o(\omega) & + & X_R^o(\omega)] \end{array}$$

Si $x(n)$ es real, entonces $X(-\omega) = X^*(\omega)$

Linealidad: Si $x_1(n) \circ \bullet X_1(\omega)$ y $x_2(n) \circ \bullet X_2(\omega)$, entonces:

$$a_1x_1(n) + a_2x_2(n) \circ \bullet a_1X_1(\omega) + a_2X_2(\omega)$$

Desplazamiento espacial:

$$x(n) \circ \bullet X(\omega) \Rightarrow x(n - k) \circ \bullet e^{-j\omega k} X(\omega)$$

Reflexión espacial:

$$x(n) \circ \bullet X(\omega) \Rightarrow x(-n) \circ \bullet X(-\omega)$$

Teorema de la Convolución:

$$\begin{aligned} x_1(n) \circ \bullet X_1(\omega), x_2(n) \circ \bullet X_2(\omega) \\ x_1(n) * x_2(n) \circ \bullet X_1(\omega)X_2(\omega) \end{aligned}$$

Teorema de la Correlación:

$$\begin{aligned} x_1(n) \circ \bullet X_1(\omega), x_2(n) \circ \bullet X_2(\omega) \\ r_{x_1x_2}(n) \circ \bullet S_{x_1x_2}(\omega) = X_1(\omega)X_2(-\omega) \end{aligned}$$

Si la señal es real, puesto que $X(-\omega) = X^*(\omega)$ entonces:

$$r_{xx}(n) \circ \bullet X(\omega)X(-\omega) = X(\omega)X^*(\omega) = |X(\omega)|^2 = S_{xx}(\omega)$$

que se conoce como el teorema de Wiener-Khinchin.

Desplazamiento frecuencial:

$$x(n) \circ \bullet X(\omega) \Rightarrow e^{j\omega_0 n} x(n) \circ \bullet X(\omega - \omega_0)$$

Teorema de modulación:

$$x(n) \circ \bullet X(\omega) \Rightarrow x(n) \cos(\omega_0 n) \circ \bullet \frac{1}{2} \{X(\omega + \omega_0) + X(\omega - \omega_0)\}$$

Teorema de Parseval:

$$\begin{aligned} x_1(n) \circ \bullet X_1(\omega), x_2(n) \circ \bullet X_2(\omega) \\ \Rightarrow \sum_{n=-\infty}^{\infty} x_1(n)x_2^*(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X_1(\omega)X_2^*(\omega)d\omega \end{aligned}$$

Demostración:

$$\begin{aligned}
\frac{1}{2\pi} \int_{-\pi}^{\pi} X_1(\omega) X_2^*(\omega) d\omega &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(\sum_{n=-\infty}^{\infty} x_1(n) e^{-j\omega n} \right) X_2^*(\omega) d\omega \\
&= \sum_{n=-\infty}^{\infty} x_1(n) \frac{1}{2\pi} \int_{-\pi}^{\pi} X_2^*(\omega) e^{-j\omega n} d\omega \\
&= \sum_{n=-\infty}^{\infty} x_1(n) x_2^*(n)
\end{aligned}$$

Para el caso $x_1(n) = x_2(n) = x(n) \circ \bullet X(\omega)$:

$$E_x = \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(\omega)|^2 d\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_{xx}(\omega) d\omega$$

Teorema del emventanado (multiplicación de secuencias):

$$\begin{aligned}
&x_1(n) \circ \bullet X_1(\omega), x_2(n) \circ \bullet X_2(\omega) \\
\Rightarrow x_3(n) = x_1(n)x_2(n) \circ \bullet X_3(\omega) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} X_1(\lambda) X_2(\omega - \lambda) d\lambda = X_1(\omega) * X_2(\omega)
\end{aligned}$$

donde el símbolo $*$ a la derecha, denota la convolución periódica en el dominio de la frecuencia.

Diferenciación en el dominio de la frecuencia:

$$x(n) \circ \bullet X(\omega) \Rightarrow nx(n) \circ \bullet j \frac{dX(\omega)}{d\omega}$$

4.3 Transformada Discreta de Fourier

En las secciones anteriores se observó que señales discretas aperiódicas tienen un espectro periódico pero continuo, que no tiene una representación directa para ser manejado por medios digitales. En esta sección se estudia la transformada discreta de Fourier (DFT, *Discrete Fourier Transform*) que representa un mecanismo para el estudio frecuencial por medios digitales para las mencionadas señales.

4.3.1 Muestreo en el dominio de la frecuencia

Toda señal aperiódica $x(n)$ de energía finita tiene un espectro continuo

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}.$$

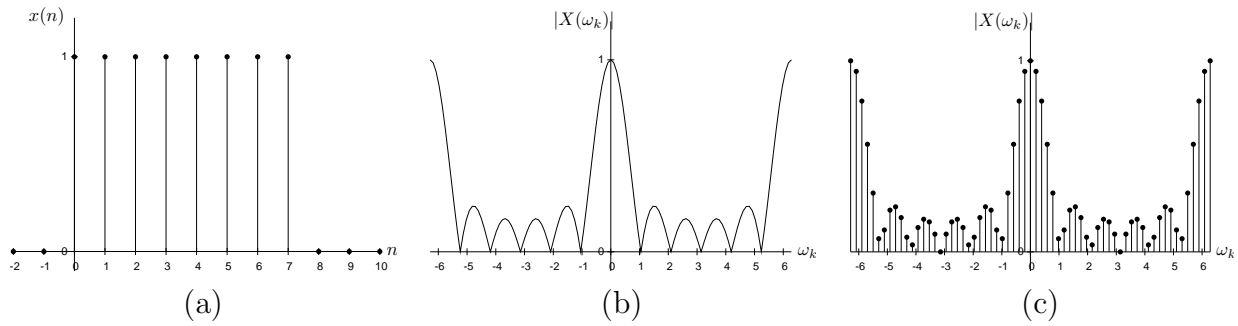


Figura 4.12: Señal discreta aperiódica, su espectro continuo, y su espectro muestreado.

Si $X(\omega)$ se muestrea periódicamente (figura 4.12) con un espaciado frecuencial $\delta\omega = \frac{2\pi}{N}$ solo se necesitarán N muestras puesto que el espectro, por corresponder a una señal discreta, es a su vez periódico con periodo 2π . Las muestras en $\omega = k\delta\omega = \frac{2\pi}{N}k$ son entonces

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi kn/N} \quad k = 0, 1, \dots, N-1$$

donde descomponiendo la sumatoria en subintervalos de N elementos y haciendo un cambio de variable para trasladar cada subintervalo al origen se modifica en

$$X\left(\frac{2\pi}{N}k\right) = \sum_{l=-\infty}^{\infty} \sum_{n=lN}^{lN+N-1} x(n)e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} \underbrace{\left[\sum_{l=-\infty}^{\infty} x(n-lN) \right]}_{x_p(n)} e^{-j2\pi kn/N}$$

con $k = 0, 1, \dots, N-1$.

La señal

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n-lN)$$

se obtiene repitiendo periódicamente $x(n)$ cada N muestras, y por ser periódica puede calcularse su serie de Fourier como

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}, \quad n = 0, 1, \dots, N-1$$

con los coeficientes

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j2\pi kn/N} = \frac{1}{N} X\left(\frac{2\pi}{N}k\right)$$

Con esta serie $x_p(n)$ puede ser reconstruido del espectro muestreado con

$$x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N}, \quad n = 0, 1, \dots, N-1$$

La señal $x(n)$ puede recuperarse de $x_p(n)$ si y solo si no hay *aliasing* en el dominio del espacio, es decir, si la longitud L de $x(n)$ es menor que el periodo N de $x_p(n)$ (figura 4.13):

$$x(n) = \begin{cases} x_p(n), & 0 \leq n \leq N - 1 \\ 0, & \text{en el resto} \end{cases}$$

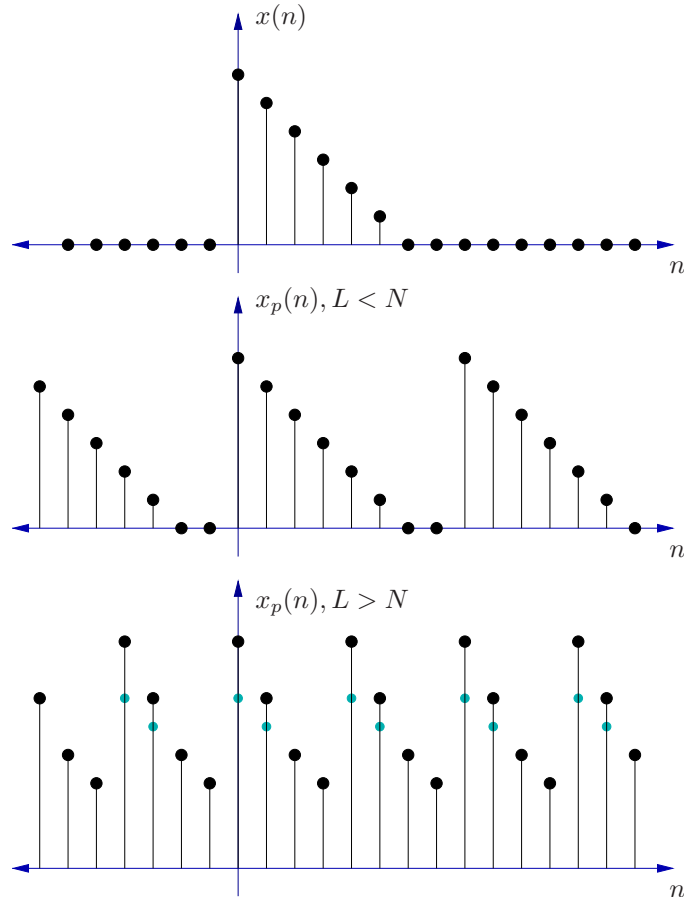


Figura 4.13: Extensión periódica por muestreo del espectro. a) Señal original, b) Extensión con $L < N$, c) Extensión con $L > N$.

Si no hay aliasing en el espacio ($L < N$) entonces

$$x(n) = x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N}, \quad 0 \leq n \leq N - 1$$

con espectro

$$\begin{aligned} X(\omega) &= \sum_{n=0}^{N-1} x(n) e^{-j\omega n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N} \right] e^{-j\omega n} \\ &= \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) \left[\frac{1}{N} \sum_{n=0}^{N-1} e^{-j\left(\omega - \frac{2\pi k}{N}\right)n} \right] \\ &= \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) P\left(\omega - \frac{2\pi k}{N}\right) \end{aligned}$$

donde

$$\begin{aligned} P(\omega) &= \frac{1}{N} \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1}{N} \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} \\ &= \frac{1}{N} \frac{\text{sen}(\omega N/2)}{\text{sen}(\omega/2)} e^{-j\omega(N-1)/2} \end{aligned} \quad (4.13)$$

es la función de interpolación que sustituye en este caso a $\text{sen}(\theta)/\theta$ con una versión de propiedades similares pero periódica (figura 4.14):

$$P\left(\frac{2\pi}{N}k\right) = \begin{cases} 1 & k = 0 \\ 0 & k = 1, 2, \dots, N-1 \end{cases}$$

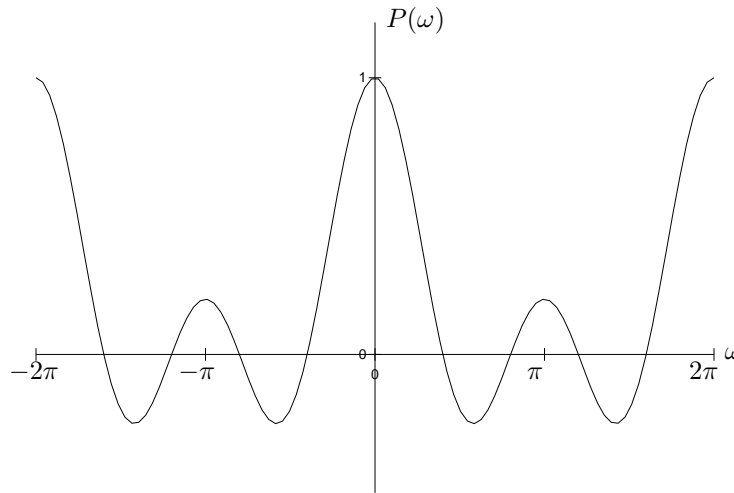


Figura 4.14: Interpolador ideal $P(\omega)$ para espectro muestreado con $N = 5$.

Ejemplo 4.2 Determine el aliasing de la secuencia $x(n) = a^n u(n)$, $|a| < 1$, si el espectro se muestrea a las frecuencias $\omega_k = 2\pi k/N$.

El espectro es

$$X(\omega) = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \frac{1}{1 - ae^{-j\omega}}$$

Las muestras están dadas por $X(\omega_k) = X\left(\frac{2\pi}{N}k\right) = \frac{1}{1 - ae^{-j2\pi k/N}}$

La secuencia periódica $x_p(n)$ representada por este espectro discreto es

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) = \sum_{l=-\infty}^0 a^{n-lN} = a^n \sum_{l=0}^{\infty} a^{lN} = a^n \frac{1}{1 - a^N}, \quad 0 \leq n \leq N-1$$

donde la constante $\frac{1}{1-a^N}$ representa el aliasing y tiende a uno si $N \rightarrow \infty$, como es de esperar. 4.2

4.3.2 La transformada discreta de Fourier

En la sección anterior se demostró que el espectro muestreado de una señal $x(n)$ de longitud finita L no representa a $x(n)$ directamente sino a su extensión periódica $x_p(n)$ obtenida de

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN)$$

Si no hay aliasing espacial, entonces $x_p(n)$ es simplemente la concatenación periódica de $x(n)$ de longitud L , que se expande con ceros para alcanzar la longitud $N > L$:

$$x_p(n) = \begin{cases} x(n) & 0 \leq n \leq L - 1 \\ 0 & L \leq n \leq N - 1 \end{cases}$$

Nótese que el efecto de rellenar con ceros $x(n)$ produce un muestreo más detallado de su espectro $X(\omega)$ que tendrá N y no L muestras. Para todo $N > L$ las muestras no proveen mayor información que el caso $N = L$, solo una “mejor” representación (figura 4.15).

La transformación de $x(n)$ hacia su espectro muestreado

$$X(k) = X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{L-1} x(n)e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N - 1$$

se conoce como transformada discreta de Fourier DFT y a la reconstrucción de $x(n)$ de las muestras espectrales

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}, \quad n = 0, 1, \dots, N - 1$$

se le conoce como transformada discreta de Fourier inversa (IDFT).

Estas transformaciones se expresan a menudo como

$$\begin{aligned} \text{DFT: } X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{kn} & k = 0, 1, \dots, N - 1 \\ \text{IDFT: } x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-kn} & n = 0, 1, \dots, N - 1 \end{aligned}$$

donde $W_N = e^{-j2\pi/N}$ es una raíz N -ésima de la unidad. Nótese que para cada k , $X(k)$ se puede expresar como el producto punto de la entrada $x(n)$ y un vector que consiste en $[1, W_N^k, W_N^{2k}, \dots, W_N^{(N-1)k}]$, o, si se interpreta $X(k)$ también como vector, entonces

$$X(k) \equiv \underline{\mathbf{X}}_N = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}$$

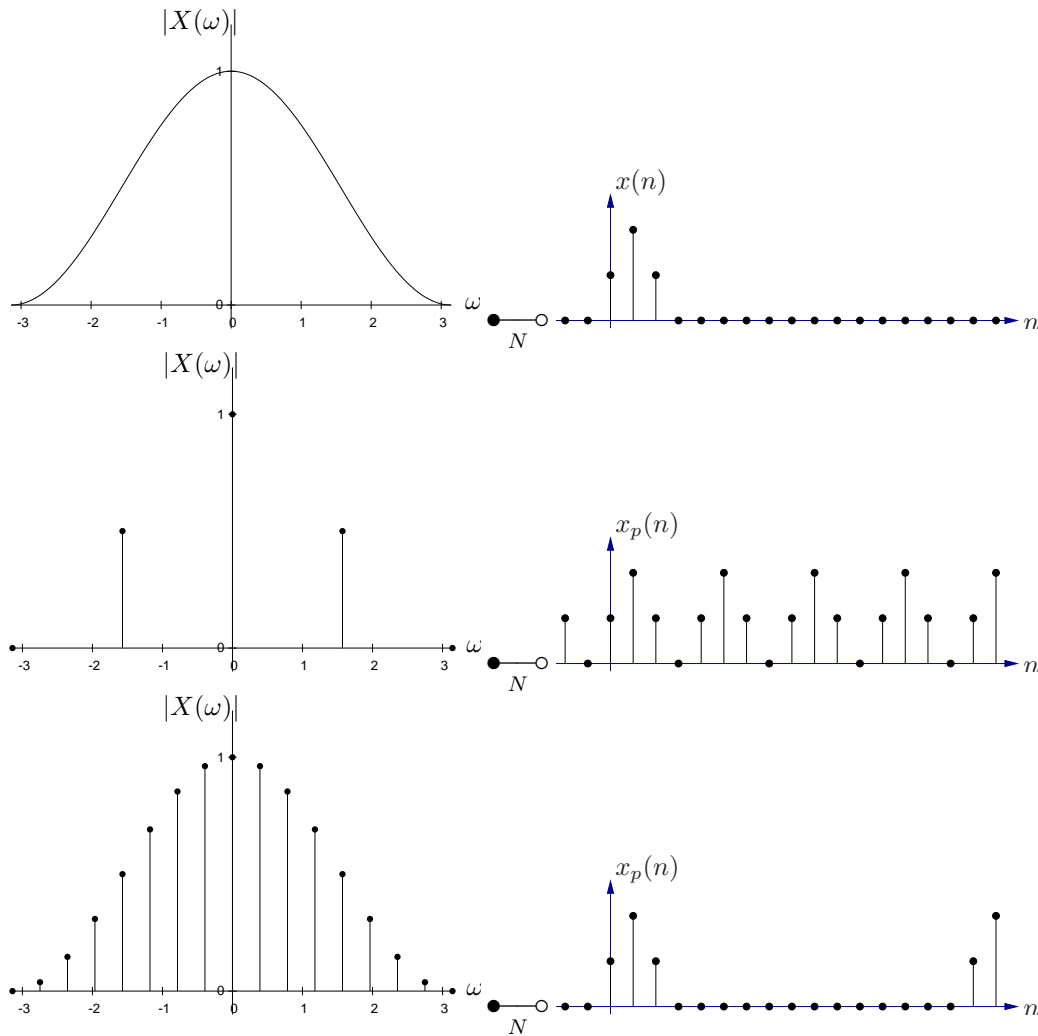


Figura 4.15: Efecto de extensión de señal con ceros y la DFT resultante.

o en forma compacta

$$\underline{\mathbf{X}}_N = \mathbf{W}_N \underline{\mathbf{x}}_N$$

de donde se deduce

$$\underline{\mathbf{x}}_N = \mathbf{W}_N^{-1} \underline{\mathbf{X}}_N = \frac{1}{N} \mathbf{W}_N^* \underline{\mathbf{X}}_N$$

4.3.3 Relación de la DFT con otras transformadas

Series de Fourier

La secuencia $x_p(n)$ por ser periódica con periodo N tiene una representación en series de Fourier

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi nk/N}, \quad -\infty < n < \infty$$

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j2\pi nk/N} \quad k = 0, 1, \dots, N-1$$

Esto es $X(k) = Nc_k$.

Transformada de Fourier de secuencias aperiódicas

Ya se verificó la relación entre la DFT $X(k)$ como muestras del espectro continuo $X(\omega)$

$$X(k) = X(\omega)|_{\omega=\frac{2\pi}{N}k}$$

que a su vez corresponden con los coeficientes de la DFT de $x_p(n)$, la extensión periódica de $x(n)$. Ambas secuencias son idénticas para $n = 0, 1, \dots, L-1$ si no hay aliasing espacial, es decir, si $x(n)$ es finita de longitud L y $N > L$.

Transformada z

Si la transformada z de $x(n)$ incluye la circunferencia unitaria, entonces

$$X(k) = X(z)|_{z=e^{j2\pi nk/N}} = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N}, \quad k = 0, 1, \dots, N-1$$

que representa al muestreo de la transformada z sobre el círculo unitario con ángulos distribuidos homogéneamente. Sin aliasing espacial se cumple entonces, con $x(n)$ finita de longitud N

$$\begin{aligned} X(z) &= \sum_{n=0}^{N-1} x(n) z^{-n} = \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \right] z^{-n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} (e^{j2\pi k/N} z^{-1})^n \\ X(z) &= \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{j2\pi k/N} z^{-1}} \end{aligned}$$

que es la reconstrucción de $X(z)$ a partir de las muestras $X(k)$ de la DFT. Evaluando en la circunferencia unitaria se obtiene para la transformada de Fourier

$$X(\omega) = \frac{1 - e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{-j(\omega - \frac{2\pi k}{N})}}$$

que es otra forma de interpolación de las muestras $X(k)$ equivalente a la expresada anteriormente con $P(\omega)$ (ver ecuación 4.13 en la página 78).

4.3.4 Propiedades de la DFT

Sea $x(n)$ una secuencia de longitud $L < N$ y $X(k)$ su correspondiente DFT de longitud N . Se cumple que

$$\begin{aligned} \text{DFT: } X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{kn} & k = 0, 1, \dots, N-1 \\ \text{IDFT: } x(n) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k)W_N^{-kn} & n = 0, 1, \dots, N-1 \end{aligned}$$

con $W_N = e^{-j2\pi/N}$.

La relación entre $x(n)$ y su DFT $X(k)$ se denota con

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k) \quad \text{ó} \quad x(n) \overset{\circ}{\underset{N}{\longleftrightarrow}} X(k)$$

Linealidad

Si $x_1(n) \overset{\circ}{\underset{N}{\longleftrightarrow}} X_1(k)$ y $x_2(n) \overset{\circ}{\underset{N}{\longleftrightarrow}} X_2(k)$ entonces

$$a_1x_1(n) + a_2x_2(n) \overset{\circ}{\underset{N}{\longleftrightarrow}} a_1X_1(k) + a_2X_2(k)$$

Simetría circular

Puesto que la DFT de N puntos de una secuencia finita $x(n)$ de longitud $L \leq N$ es equivalente a la DFT de N puntos de una secuencia periódica $x_p(n)$ de periodo N obtenida como (figura 4.16)

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN)$$

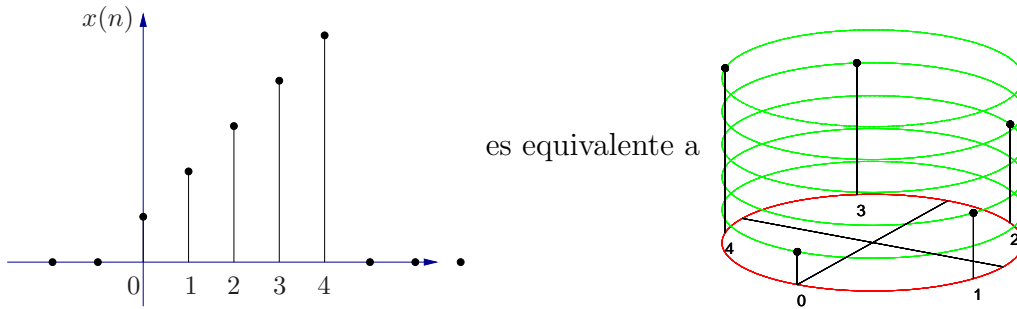


Figura 4.16: Secuencia circular equivalente a $x(n)$.

entonces un desplazamiento de $x_p(n)$ k unidades hacia la derecha equivale a un desplazamiento circular (figura 4.17).

$$x_p(n - k) \overset{\circ}{\underset{N}{\longleftrightarrow}} \cdot \cdot \cdot \overset{\circ}{\underset{N}{\longleftrightarrow}} x(n - k \bmod N) \equiv x((n - k))_N$$

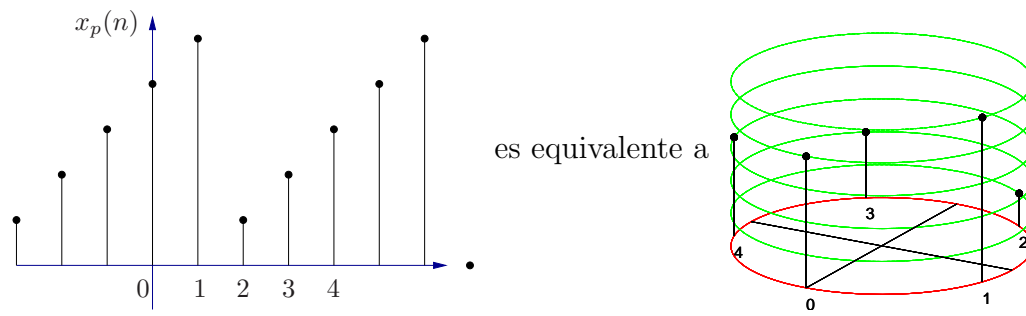


Figura 4.17: Secuencia circular equivalente a $x_p(n - 2)$.

Una secuencia es *circularmente par* si es simétrica con respecto al punto cero de la circunferencia (figura 4.18):

$$x(N - n) = x(n), \quad 1 \leq n \leq N - 1$$

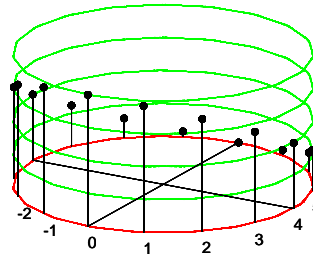


Figura 4.18: Simetría circular par

La secuencia es *circularmente impar* si es antisimétrica con respecto al punto cero (figura 4.19):

$$x(N - n) = -x(n), \quad 1 \leq n \leq N - 1$$

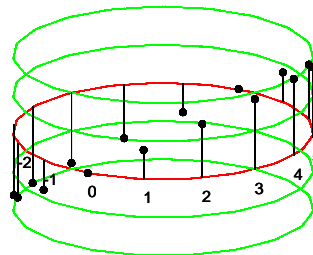


Figura 4.19: Simetría circular impar

La reflexión espacial circular se obtiene con

$$x((-n))_N = x(N - n), \quad 0 \leq n \leq N - 1$$

Utilizando $x_p(n)$ en vez de $x(n)$ se tiene

$$\begin{aligned}
\text{par: } & x_p(n) = x_p(-n) = x_p(N-n) \\
\text{impar: } & x_p(n) = -x_p(-n) = -x_p(N-n) \\
\text{conjugada par: } & x_p(n) = x_p^*(N-n) \\
\text{conjugada impar: } & x_p(n) = -x_p^*(N-n)
\end{aligned}$$

La secuencia se puede descomponer en

$$\begin{aligned}
x_p(n) &= x_{pe}(n) + x_{po}(n) \\
x_{pe}(n) &= \frac{1}{2} [x_p(n) + x_p^*(N-n)] \\
x_{po}(n) &= \frac{1}{2} [x_p(n) - x_p^*(N-n)]
\end{aligned}$$

Si $x(n) = x_R(n) + jx_I(n)$, $0 \leq n \leq N-1$, y $X(k) = X_R(k) + jX_I(k)$ entonces se obtiene con la definición de la DFT.

$$\begin{aligned}
X_R(k) &= \sum_{n=0}^{N-1} \left[x_R(n) \cos\left(\frac{2\pi kn}{N}\right) + x_I(n) \sen\left(\frac{2\pi kn}{N}\right) \right] \\
X_I(k) &= -\sum_{n=0}^{N-1} \left[x_R(n) \sen\left(\frac{2\pi kn}{N}\right) - x_I(n) \cos\left(\frac{2\pi kn}{N}\right) \right] \\
x_R(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \left[X_R(k) \cos\left(\frac{2\pi kn}{N}\right) - X_I(k) \sen\left(\frac{2\pi kn}{N}\right) \right] \\
x_I(n) &= \frac{1}{N} \sum_{k=0}^{N-1} \left[X_R(k) \sen\left(\frac{2\pi kn}{N}\right) + X_I(k) \cos\left(\frac{2\pi kn}{N}\right) \right]
\end{aligned}$$

Si $x(n)$ es real, entonces

$$X(N-k) = X^*(k) = X(-k) \Rightarrow |X(N-k)| = |X(k)|, \angle X(N-k) = -\angle X(k)$$

Si $x(n)$ es real y par entonces $x(n) = x(N-n)$, para $0 \leq n \leq N-1$, y la DFT se torna

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos\left(\frac{2\pi kn}{N}\right), \quad 0 \leq k \leq N-1$$

que también es real y par. Puesto que $X_I(k) = 0$ entonces

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cos\left(\frac{2\pi kn}{N}\right), \quad 0 \leq n \leq N-1$$

Si $x(n)$ es real e impar entonces

$$\begin{aligned}
x(n) &= -x(N-n), \quad 0 \leq n \leq N-1 \\
X(k) &= -j \sum_{n=0}^{N-1} x(n) \sen\left(\frac{2\pi kn}{N}\right), \quad 0 \leq k \leq N-1
\end{aligned}$$

y puesto que $X_R(k) = 0$ entonces

$$x(n) = \frac{j}{N} \sum_{k=0}^{N-1} X(k) \operatorname{sen} \left(\frac{2\pi kn}{N} \right), \quad 0 \leq n \leq N-1$$

Convolución circular

Sean $x_1(n)$ y $x_2(n)$ dos secuencias de duración finita, ambas de longitud N , y sus respectivas DFT de N puntos

$$X_1(k) = \sum_{n=0}^{N-1} x_1(n) e^{-j2\pi nk/N}, \quad k = 0, 1, \dots, N-1$$

$$X_2(k) = \sum_{n=0}^{N-1} x_2(n) e^{-j2\pi nk/N}, \quad k = 0, 1, \dots, N-1$$

Si $X_3 = X_1(k)X_2(k)$, $k = 0, 1, \dots, N-1$ entonces

$$\begin{aligned} x_3(m) &= \frac{1}{N} \sum_{k=0}^{N-1} X_3(k) e^{j2\pi km/N} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} X_1(k) X_2(k) e^{j2\pi km/N} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum_{n=0}^{N-1} x_1(n) e^{-j2\pi kn/N} \right] \left[\sum_{l=0}^{N-1} x_2(l) e^{-j2\pi kl/N} \right] e^{j2\pi km/N} \\ &= \frac{1}{N} \sum_{n=0}^{N-1} x_1(n) \sum_{l=0}^{N-1} x_2(l) \left[\sum_{k=0}^{N-1} e^{j2\pi k(m-n-l)/N} \right] \end{aligned}$$

y con $a = e^{j2\pi(m-n-l)/N}$ en

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & \text{si } a = 1 \Leftrightarrow m - n - l = pN \Rightarrow l = m - n - pN = ((m - n))_N \\ \frac{1-a^N}{1-a} = 0 & \text{si } a \neq 1, \text{ puesto que } a^N = 1 \end{cases}$$

por lo que

$$x_3(m) = \sum_{n=0}^{N-1} x_1(n) x_2((m-n))_N, \quad m = 0, 1, \dots, N-1 = x_1(m) \circledast x_2(m)$$

operación denominada convolución circular. Nótese que esta función es para los elementos $x_3(m)$ con $m = 0, 1, \dots, N-1$ equivalente a la convolución de $x_1(n)$ con la extensión periódica con periodo N de $x_2(n)$.

Otras propiedades se resumen en la tabla 4.1.

Tabla 4.1: Propiedades de la DFT

Propiedad	Dominio espacial	Dominio frecuencial
Notación	$x(n), y(n)$	$X(k), Y(k)$
Periodicidad	$x(n) = x(n + N)$	$X(k) = X(k + N)$
Linealidad	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(k) + a_2X_2(k)$
Reflexión espacial	$x(N - n)$	$X(N - k)$
Desplazamiento espacial circular	$x((n - l))_N$	$X(k) e^{-j2\pi kl/N}$
Desplazamiento frecuencial circular	$x(n)e^{j2\pi ln/N}$	$X((k - l))_N$
Conjugación compleja	$x^*(n)$	$X^*(N - k)$
Convolución circular	$x_1(n) \circledast x_2(n)$	$X_1(k)X_2(k)$
Correlación circular	$x(n) \circledast y^*(-n)$	$X(k)Y^*(k)$
Multiplicación de dos secuencias	$x_1(n)x_2(n)$	$\frac{1}{N}X_1(k) \circledast X_2(k)$
Teorema de Parseval	$\sum_{n=0}^{N-1} x(n)y^*(n)$	$\frac{1}{N} \sum_{k=0}^{N-1} X(k)Y^*(k)$

4.3.5 Filtrado lineal basado en la DFT

La existencia de algoritmos eficientes para extraer la DFT (llamados FFT del inglés *Fast Fourier Transform*) hace posible que sea en ocasiones más eficiente calcular el efecto de un filtro en el dominio de la frecuencia que directamente en el dominio del espacio.

Sea la entrada $x(n)$ de longitud L y un filtro FIR con respuesta impulsional $h(n)$ de longitud M . Asíumase además que ambas secuencias son causales, es decir

$$\begin{aligned} x(n) &= 0, & n < 0, n \geq L \\ h(n) &= 0, & n < 0, n \geq M \end{aligned}$$

La salida del filtro puede calcularse en el dominio del espacio con la convolución

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n - k) = h(n) * x(n)$$

que por la naturaleza finita de $x(n)$ y $h(n)$ también es finita. Puede demostrarse que el resultado de esta convolución tendrá longitud $M + L - 1$.

En el dominio de la frecuencia la salida es

$$Y(\omega) = X(\omega)H(\omega)$$

Si se desea representar $y(n)$ a través de muestras de $Y(\omega)$ se necesitan entonces al menos $N \geq M + L - 1$ muestras para evitar así el aliasing espacial y se obtiene con la DFT:

$$Y(k) = Y(\omega)|_{\omega=2\pi k/N} = X(\omega)H(\omega)|_{\omega=2\pi k/N} = X(k)H(k)$$

donde $X(k)$ y $H(k)$ representan entonces la DFT de las secuencias $x(n)$ y $h(n)$ que han sido extendidas con ceros para alcanzar la longitud N .

Nótese que la compensación de longitud de $x(n)$ y $y(n)$ logra que la convolución circular sea equivalente a la convolución lineal.

Ejemplo 4.3 Calcule la respuesta del filtro con respuesta impulsional $h(n) = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$ ante la entrada $x(n) = \{4, 2, 2, 4\}$.

La longitud de $x(n)$ es $L = 4$ y la de $h(n)$ es $M = 3$, por lo que la salida necesita al menos $L + M - 1 = 6$ muestras. Se utilizará $N = 8$ por simplicidad y porque los algoritmos de FFT usualmente requieren $N = 2^k$ muestras, con $k \in \mathbb{N}$. Así

$$\begin{aligned} X(k) &= \sum_{n=0}^7 x(n)e^{j2\pi kn/8} \\ &= x(0) + x(1)e^{-j\frac{\pi}{4}k} + x(2)e^{-j\frac{\pi}{2}k} + x(3)e^{-j\frac{3\pi}{4}k} \\ &= 4(1 + e^{-j\frac{3\pi}{4}k}) + 2(e^{-j\frac{\pi}{4}k} + e^{-j\frac{\pi}{2}k}) \end{aligned}$$

con lo que se obtiene

$$\begin{aligned} X(0) &= 12 \\ X(1) &= 4 - \sqrt{2} - j(2 + 3\sqrt{2}) \\ X(2) &= 2 + j2 \\ X(3) &= 4 + \sqrt{2} + j(2 - 3\sqrt{2}) \\ X(4) &= 0 \\ X(5) &= 4 + \sqrt{2} - j(2 - 3\sqrt{2}) = X^*(3) \\ X(6) &= 2 - j2 = X^*(2) \\ X(7) &= 4 - \sqrt{2} + j(2 + 3\sqrt{2}) = X^*(1) \end{aligned}$$

De forma similar

$$\begin{aligned} H(k) &= \sum_{n=0}^7 h(n)e^{-j2\pi kn/8} \\ &= h(0) + h(1)e^{-j\frac{\pi}{4}k} + h(2)e^{-j\frac{\pi}{2}k} \\ &= \frac{1}{4}(1 + e^{-j\frac{\pi}{2}k}) + \frac{1}{2}e^{-j\frac{\pi}{4}k} \end{aligned}$$

con lo que se obtiene

$$\begin{aligned} H(0) &= 1 \\ H(1) &= \frac{1 + \sqrt{2}}{4} - j\frac{1 + \sqrt{2}}{4} = H^*(7) \\ H(2) &= -\frac{j}{2} = H^*(6) \\ H(3) &= \frac{1 - \sqrt{2}}{4} + j\frac{1 - \sqrt{2}}{4} = H^*(5) \\ H(4) &= 0 \end{aligned}$$

El producto de ambas secuencias es

$$\begin{aligned}
 Y(0) &= 12 \\
 Y(1) &= -\frac{3 + \sqrt{2}}{2} - j\frac{5 + 2\sqrt{2}}{2} = Y^*(7) \\
 Y(2) &= 1 - j = Y^*(6) \\
 Y(3) &= \frac{\sqrt{2} - 3}{2} + j\frac{5 - 2\sqrt{2}}{2} = Y^*(5) \\
 Y(4) &= 0
 \end{aligned}$$

Y con la IDFT se obtiene finalmente

$$\begin{aligned}
 y(n) &= \sum_{k=0}^7 Y(k) e^{j2\pi kn/8}, \quad n = 0, 1, \dots, 7 \\
 &= \left\{ 1, \frac{5}{2}, \frac{5}{2}, \frac{5}{2}, \frac{5}{2}, 1, 0, 0 \right\}
 \end{aligned}$$

4.3

La tabla 4.2 resume las herramientas de análisis de Fourier en una dimensión.

Tabla 4.2: Análisis de Fourier en una dimensión

Serie de Fourier continua	$x(s) = \sum_{k=-\infty}^{\infty} c_k e^{jk\Omega_0 s}$ $c_k = \frac{1}{S_p} \int_{s_0}^{s_0+S_p} e^{-j\Omega_0 ks} x(s) ds$
Serie de Fourier discreta	$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}$ $c_l = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi ln/N} \quad l = 0, 1, \dots, N-1$
Transformada de Fourier	$X(\Omega) = \int_{-\infty}^{\infty} x(s) e^{-j\Omega s} ds$ $x(s) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega) e^{j\Omega s} d\Omega$
Transformada de Fourier de señales discretas	$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n}$ $x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} d\omega$
Transformada discreta de Fourier (DFT)	$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1$ $x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N}, \quad n = 0, 1, \dots, N-1$

4.4 Análisis de Fourier en múltiples dimensiones

En el caso de espacios de w dimensiones, la transformada de Fourier se deriva por medio del producto interno definido por

$$\langle f(\underline{\mathbf{x}}), g(\underline{\mathbf{x}}) \rangle = \int_{-\infty}^{\infty} f^*(\underline{\mathbf{x}})g(\underline{\mathbf{x}}) d^w x$$

para aquellas funciones que cumplen

$$\int_{-\infty}^{\infty} |f(\underline{\mathbf{x}})|^2 d^w x = \langle f(\underline{\mathbf{x}}), f(\underline{\mathbf{x}}) \rangle = \|f(\underline{\mathbf{x}})\|_2^2 < \infty$$

como

Por hacer:

Ver Jähne pp.48

4.5 Análisis de Fourier en dos dimensiones

Para extender el análisis anterior al caso de imágenes bidimensionales se repiten los pasos seguidos anteriormente utilizando para la base funciones exponenciales complejas armónicamente relacionadas en dos dimensiones, definidas como

$$\epsilon_{k,l}(x, y) = e^{j(k\Omega_{x0}x + l\Omega_{y0}y)} = e^{j2\pi(kF_{x0}x + lF_{y0}y)} \quad k, l = 0, \pm 1, \pm 2, \dots$$

Obsérvese que éstas funciones son separables, en el sentido de que

$$\epsilon_{k,l}(x, y) = \epsilon_k(x)\epsilon_l(y) = e^{jk\Omega_{x0}x} e^{jl\Omega_{y0}y}$$

Es posible demostrar que éstas funciones son ortogonales entre sí, definiendo el producto interno para funciones de dos dimensiones como

$$\langle u(x, y), v(x, y) \rangle = \int_{y_1}^{y_2} \int_{x_1}^{x_2} u^*(x, y)v(x, y) dx dy$$

donde $[x_1, x_2] \times [y_1, y_2]$ es la región del plano espacial bajo análisis. Empleando las funciones exponenciales bidimensionales armónicamente relacionadas y la anterior definición de producto interno se obtienen las herramientas de análisis de Fourier resumidas en la tabla 4.3

4.5.1 Aliasing en 2D

Con imágenes en N dimensiones, si el muestreo es homogéneo deben entonces considerarse las posibilidades de traslape en las N dimensiones.

Tabla 4.3: Análisis de Fourier en dos dimensiones

Serie de Fourier continua	$u(x, y) = \sum_{l=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_{k,l} e^{j(k\Omega_{x0}x + l\Omega_{y0}y)}$ $c_{k,l} = \frac{1}{S_x S_y} \int_{y_0}^{y_0+S_y} \int_{x_0}^{x_0+S_x} e^{-j(k\Omega_{x0}x + l\Omega_{y0}y)} u(x, y) dx dy$
Serie de Fourier discreta	$u(n, m) = \sum_{l=0}^{M-1} \sum_{k=0}^{N-1} c_{k,l} e^{j2\pi(kn/N + lm/M)}$ $c_{k,l} = \frac{1}{NM} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} u(n, m) e^{-j2\pi(kn/N + lm/M)}$ $0 \leq k < N, 0 \leq l < M$
Transformada de Fourier	$X(\Omega_x, \Omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(x, y) e^{-j(\Omega_x x + \Omega_y y)} dx dy$ $u(x, y) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(\Omega_x, \Omega_y) e^{j(\Omega_x x + \Omega_y y)} d\Omega_x d\Omega_y$
Transformada de Fourier de señales discretas	$X(\omega_x, \omega_y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} u(n, m) e^{-j(\omega_x n + \omega_y m)}$ $u(n, m) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} X(\omega_x, \omega_y) e^{j(\omega_x n + \omega_y m)} d\omega_x d\omega_y$
Transformada discreta de Fourier (DFT)	$X(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} u(n, m) e^{-j2\pi(kn/N + lm/M)}$ $0 \leq k < N, 0 \leq l < M$ $u(n, m) = \frac{1}{NM} \sum_{l=0}^{M-1} \sum_{k=0}^{N-1} X(k, l) e^{j2\pi(kn/N + lm/M)}$ $0 \leq n < N, 0 \leq m < M$

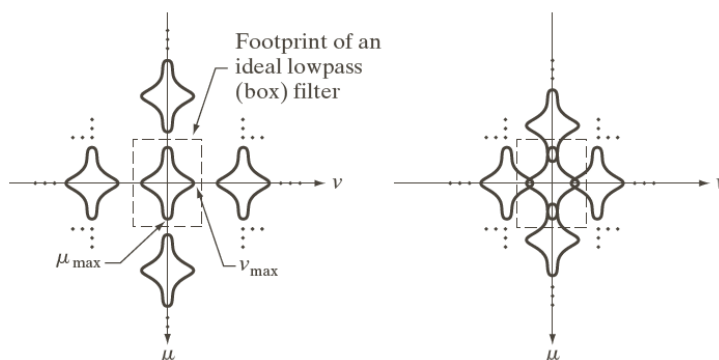


Figura 4.20: Traslape espectral en el caso del aliasing en 2D (tomado de Gonzalez y Woods [29]).

4.5.2 Patrones Moirè

Los patrones de moirè son patrones emergentes que surgen ya sea al muestrear señales bidimensionales regulares o al sobreponer patrones regulares con diferentes orientaciones o frecuencias.

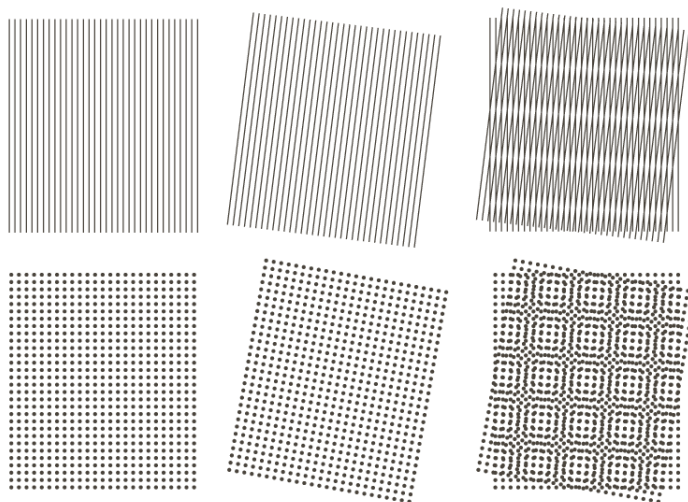


Figura 4.21: Traslape espectral en el caso del aliasing en 2D (tomado de Gonzalez y Woods [29]).

4.5.3 Propiedades de la DFT en dos dimensiones

Periodicidad y Simetría

Los conceptos de periodicidad representados por medio de continuaciones cilíndricas se transforman en el caso bidimensional a dos dimensiones a simetrías toroidales.

Por hacer:

Revisar Jähne, pp. 41

Simetrías: (ver Jähne) par, impar, hermítica (hermitian) y anti-hermítica

Periodicidad

Similitud

Sea a un número real diferente de cero, \mathbf{A} una matriz real invertible, \mathbf{z} \mathbf{R} una matriz ortogonal representando una rotación de sistemas coordenados ($\mathbf{R}^{-1} = \mathbf{R}^T$, $\det \mathbf{R} = 1$).

Se cumple en el dominio continuo:

$$\begin{aligned}
 u(a\mathbf{x}) &\circ\text{---}\bullet \frac{1}{|a|^w} U(\mathbf{k}/a) \\
 u(\mathbf{A}\mathbf{x}) &\circ\text{---}\bullet \frac{1}{\det \mathbf{A}} U((\mathbf{A}^T)^{-1}\mathbf{k}) \\
 u(\mathbf{R}\mathbf{x}) &\circ\text{---}\bullet U(\mathbf{R}\mathbf{k})
 \end{aligned}$$

Principio de Incertidumbre y filtros de Gabor

Traslación y Rotación. Caso de $(-1)^{x+y}$

Convolución (Efectos de la periodicidad)

4.5.4 Filtrado en el dominio de la frecuencia

Filtros paso-bajo

1. Ideal
2. Gaussiano
3. Butterworth

Filtros paso-alto

1. Ideal
2. Gaussiano
3. Butterworth

Laplaciano

Corrección para efecto moirè.

4.6 Tomografía

Capítulo 5

Procesamiento de imágenes en color

El *color* es un fenómeno fisiopsicológico no comprendido aun en su totalidad. Es una cualidad perceptiva asociada a la composición espectral de la luz visible que incide en la retina. En la sección 2.9 se revisaron características fisiológicas del sistema visual humano, que permiten simultáneamente percibir millones de colores, contra aproximadamente 20 tonos de gris. Una componente de la percepción cromática que no ha sido aun posible simular satisfactoriamente por medios computacionales, es la *constancia* de color, que consiste en la capacidad de percibir los mismos colores a pesar de que la iluminación varíe, y por tanto la composición espectral de la luz incidente.

La luz se denomina *acromática* si su único atributo visible es la *intensidad*, atribuible a los niveles de gris. En principio contiene a todo el ancho de banda del espectro visible. Tres cantidades se utilizan para describir la intensidad:

- radiancia: potencia de la fuente luminosa, medida en watts (W)
- luminancia: cantidad de energía percibida, medida en lumens (lm)
- brillo: descriptor subjetivo, relacionado con la noción acromática de la intensidad.

Por otro lado, la luz *cromática* tiene una composición espectral que la caracteriza. En el ojo humano, la percepción de color es posible por la presencia de tres tipos de conos, para longitudes de onda largas (rojo, 65%), medias (verde, 33%) y cortas (azul, 2%).

La constitución tricromática de la estructura fotosensible de la retina justifica que se puedan percibir los colores como combinación de los colores primarios rojo, verde y azul. La mezcla de tres componentes lumínicas con los tres colores primarios aditivos resulta en luz blanca.

Los colores secundarios cian, magenta y amarillo se producen como combinación de dos colores primarios aditivos (figura 5.1(a)). Así con colores de luz, cian es la mezcla de azul y verde, magenta la mezcla de rojo y azul y amarillo la mezcla de verde y rojo.

Las tres componentes rojo (R), verde (G) y azul (B) engendran un espacio de color conocido como el espacio RGB. La 'Comisión Internacional de Iluminación' CIE (Commission Internationale de l'Eclairage) asignó en 1931 la norma:

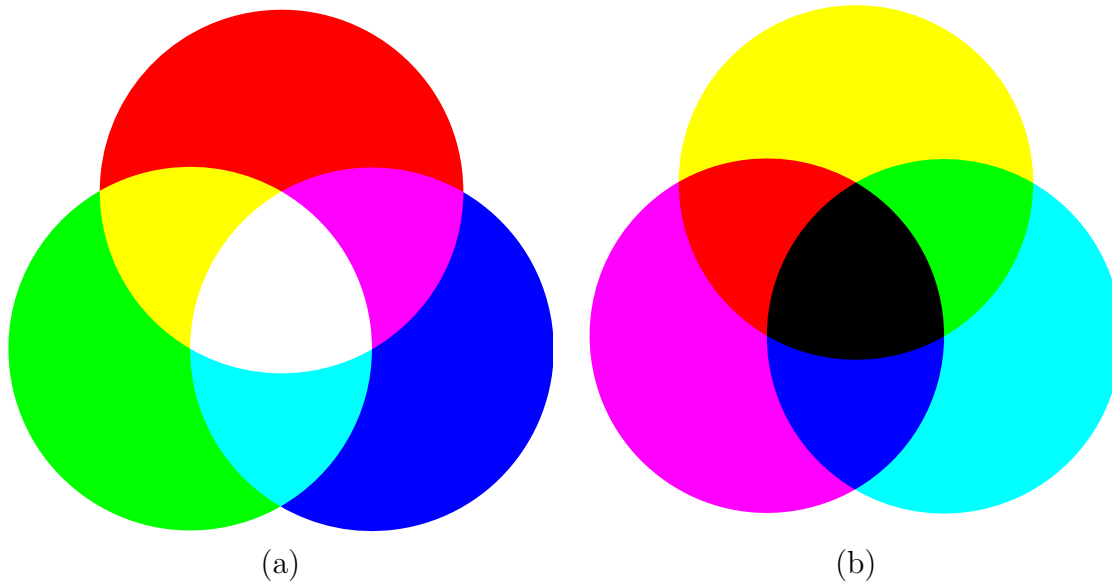


Figura 5.1: Colores primarios (a) Aditivos, (b) Sustractivos

- rojo = 700 nm
- verde = 546,1 nm
- azul = 435,8 nm

antes de conocer en 1965 las respuestas espectrales de los conos. Estas definiciones se utilizan en sistemas de despliegue como tubos de rayos catódicos (CRT), monitores de cristal líquido (LCD), TFT y plasma. Sin embargo, como se deriva de las curvas espectrales, no existe un único rojo, o verde o azul. Es evidente que la suma de los tres colores primarios no permite producir todos los colores, en el sentido de la combinación lineal de tres impulsos no puede generar todas las distribución espectrales posibles.

Si en vez de luz se utilizan pigmentos, a éstos se les atribuyen colores primarios si *absorben* un único color primario aditivo. De este modo, los colores primarios sustractivos son cian (el pigmento absorbe rojo), magenta (se absorbe el verde) y amarillo (se absorbe el azul), y sus mezclas producen como colores secundarios sustractivos al rojo, verde y amarillo, como lo ilustra la figura 5.1(b). La mezcla de los tres colores primarios sustractivos produce negro. Las tres componentes cian (C), magenta (M) y amarillo (Y) engendran al espacio de color CMY. Éste espacio en la práctica se complementa con el canal negro (K) para reducir el uso de tintes de color para producir el negro, y para compensar el hecho de que la producción de los colores primarios perfectos no es alcanzable y por tanto su mezcla no produce el negro ideal. CMYK es el nombre con el que se designa a este espacio de color.

5.1 Características de color

Anteriormente ya se introdujeron los espacios de color RGB, CMY, y CMYK, que son sistemas coordenados que se emplean para describir colores. Los anteriores no son los únicos. Particularmente para caracterizar características perceptivas del color por parte del sistema visual humano, se les han asignado tres características:

1. Brillo: revisado anteriormente y correspondiente a la noción acromática de la intensidad de luz.
2. Matiz (en inglés *hue*): relacionado con la frecuencia dominante del espectro de luz incidente.
3. Croma o saturación: relacionado con la pureza de la frecuencia dominante

Como crominancia o cromaticidad se denota al par matiz-croma.

El CIE ha estandarizado las curvas de ponderación de la distribución espectral de potencia (DEP) para calcular los tres valores XYZ (triestímulo) con los que se describe un color [57], como lo ilustra la figura 5.2. Estas curvas de cierta forma representan a un observador humano promedio.

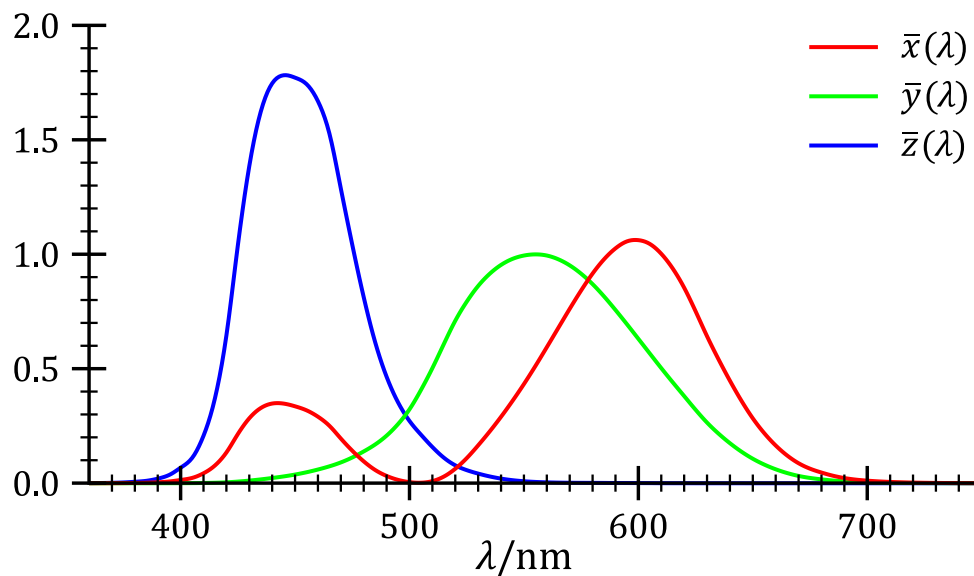


Figura 5.2: Curvas de distribución espectral de los tres valores XYZ.

Con las DEP estandares y asumiendo que $I(\lambda)$ es la distribución espectral de potencia de la luz incidente, se calculan los valores triestímulo XYZ con

$$\begin{aligned}
 X &= \int_0^{\infty} I(\lambda)\bar{x}(\lambda) d\lambda \\
 Y &= \int_0^{\infty} I(\lambda)\bar{y}(\lambda) d\lambda \\
 Z &= \int_0^{\infty} I(\lambda)\bar{z}(\lambda) d\lambda
 \end{aligned}$$

donde Y corresponde al brillo o luminancia del color.

El color como tal es caracterizado por los coeficientes

$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad z = \frac{Z}{X+Y+Z}$$

de donde se deriva

$$x + y + z = 1$$

Esto quiere decir que para especificar el color solo son necesarios los términos x e y puesto que $z = 1 - x - y$. Con x e y se produce el llamado Diagrama de Cromaticidad, ilustrado en la figura 5.3. Los colores monocromáticos, es decir, con una sola componente espectral,

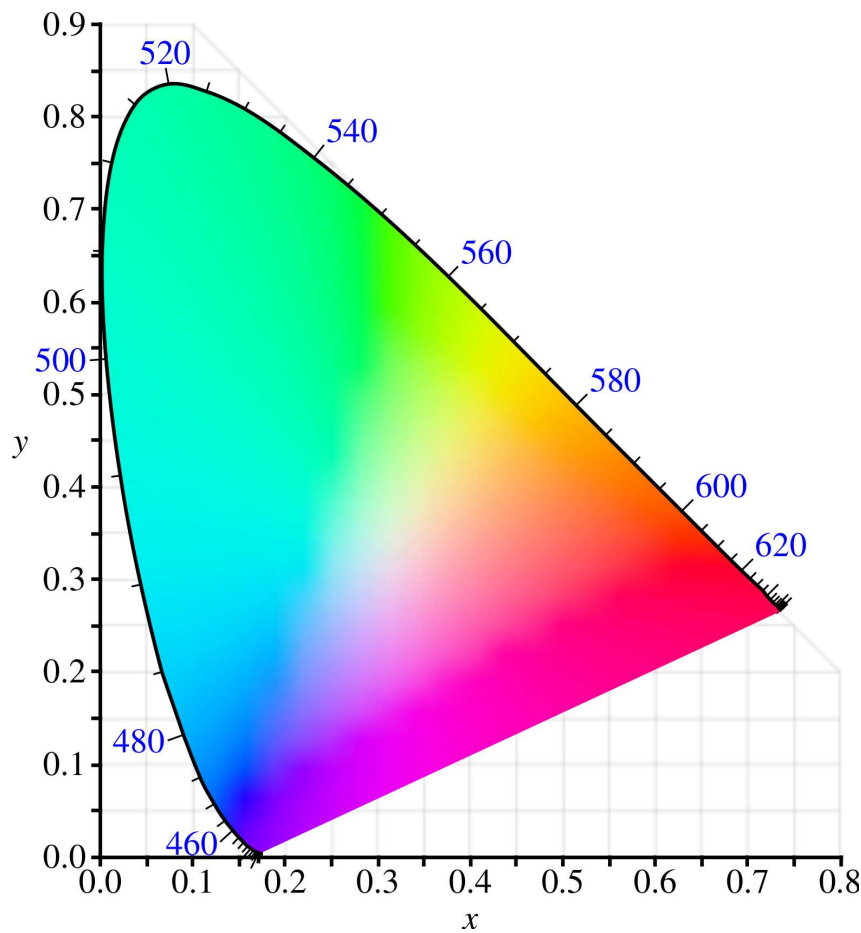


Figura 5.3: Diagrama de Cromaticidad CIE

se encuentran en el borde. El punto en $x = y = 1/3$ se denomina punto de equienergía (*equal energy*), donde la saturación es cero.

Todos los colores sobre un segmento de recta se pueden generar con combinaciones lineales de los colores extremos. Extendiendo la combinación lineal a tres puntos, éstos demarcan un triángulo sobre el diagrama, lo que demuestra que no todos los colores pueden ser expresados con la combinación lineal de tres colores primarios.

El espacio de color xyY preserva líneas en XYZ, pero no distancias.

5.2 Otros espacios de color

Otros modelos o sistemas de color se han diseñado como estándares para facilitar la especificación de colores en aplicaciones específicas (impresión, gráficos por computadora, percepción, etc.)

Ejemplos de espacios: RGB, CMY, CMYK, HSI, HSV.

Los dispositivos como cámaras, y monitores utilizan RGB como salida directa de los convertidores analógicos y digitales

5.3 Algoritmos de procesamiento de color

5.3.1 Procesamiento de pseudo color

Usualmente el objetivo es mejorar representaciones para el análisis humano.

5.3.2 Procesamiento completo de color

1. Descomposición en canales, procesamiento monocromático, mezcla
2. Procesamiento directo

Cálculo de gradiente de color

Varias técnicas de detección de discontinuidades se basan en el uso del gradiente de canales de grises, pues éstos han mostrado robustez como representación de la *bordicidad*. Its approximation for discrete digital images has been analyzed in detail in the past. Most methods involve the use of well known convolution kernels, like the operators by Roberts, Robinson, Prewitt, Kirsch, or Sobel [61, 73]. Ando [3] proposes a design strategy for consistent gradient operators, which minimize the inconsistency between continuous and discrete gradient implementations.

On the other hand, not much attention has been paid to the computation of gradient equivalents in the multi-spectral case. It has been often suggested to decompose the problem into several mono-spectral instances that are somehow combined. For example, Sonka et al. [73] suggest the use of the maximum among all gradient magnitudes of the red, green, and blue components. Other possibilities involve the computation of the mean or the median of all spectral channels. This sort of combinations ignores the interrelationships among all spectral bands (Figure 5.4).

In this work, color contrast measures based on the maximal directional derivative (MDD) [17, 30] have been chosen as representations of the pixel edgeness. The directional derivative of the functional image representation $\underline{\mathbf{f}}_{\mathcal{I}}$ in the direction $\underline{\mathbf{u}} \in \mathbb{R}^2$ at the image position $\underline{\mathbf{p}}$ is given by

$$\frac{d}{dt}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}} + t\underline{\mathbf{u}})|_{t=0} \quad (5.1)$$

With the Jacobian of $\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}}) = [f_1(\underline{\mathbf{p}}), \dots, f_d(\underline{\mathbf{p}})]^T$ defined as

$$\mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}}) = \begin{bmatrix} \frac{\partial f_1}{\partial p_1} & \frac{\partial f_1}{\partial p_2} \\ \vdots & \vdots \\ \frac{\partial f_d}{\partial p_1} & \frac{\partial f_d}{\partial p_2} \end{bmatrix} \quad (5.2)$$

it can be proven that the directional derivative equals the scalar product between the Jacobian and the unit directional vector [30]:

$$\frac{d}{dt}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}} + t\underline{\mathbf{u}})|_{t=0} = \mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}}) \cdot \underline{\mathbf{u}}. \quad (5.3)$$

The *maximal directional derivative* (MDD) is the set of two-dimensional vectors $\{\underline{\boldsymbol{\zeta}}_j\}$ lying in the direction of a unit vector $\underline{\mathbf{u}}$ that maximizes $\|\mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}}) \cdot \underline{\mathbf{u}}\|_m$, with the L_m norm $\|\underline{\mathbf{x}}\|_m = \sqrt[m]{|x_1|^m + |x_2|^m}$:

$$\{\underline{\boldsymbol{\zeta}}_j\} = \left\{ \underline{\mathbf{u}}_o \mid \|\mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}}) \cdot \underline{\mathbf{u}}_o\|_m = \max_{\|\underline{\mathbf{u}}\|=1} \|\mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}}) \cdot \underline{\mathbf{u}}\|_m \right\}. \quad (5.4)$$

For $m = 2$ (Euclidean metric) the vectors $\underline{\boldsymbol{\zeta}}_i$ can be found via the eigensolution of the square of the Jacobian:

$$[\mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}})^T \mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}})] \underline{\mathbf{e}}_j = \lambda_j \underline{\mathbf{e}}_j. \quad (5.5)$$

The vector $\underline{\mathbf{u}}_o$ is $\underline{\mathbf{e}}_i / \|\underline{\mathbf{e}}_i\|$, for $i = \arg \max_j \lambda_j$. Since the square of the Jacobian is a 2×2 matrix, the eigensolution can be efficiently computed for each pixel.

The *color contrast* is defined as the vector in the direction of the maximal directional derivative with a magnitude equal to the difference between the largest and smallest eigenvalues of the squared Jacobian $\mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}})^T \mathbf{D}\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\mathbf{p}})$.

Both the MDD and the color contrast can be used as color edgeness representations. Figure 5.4 shows an example, where the maximum among the channel gradients fails, while the color contrast still detects all available edges. The MDD result is in this case identical to the color contrast, since in this case there is only one eigenvector in the horizontal position.

Estimación de mapas de probabilidad de color

Varias aplicaciones usuales requieren encontrar en una imagen los lugares donde se encuentran objetos de algún color específico. Consierando el proceso de formación de imagen

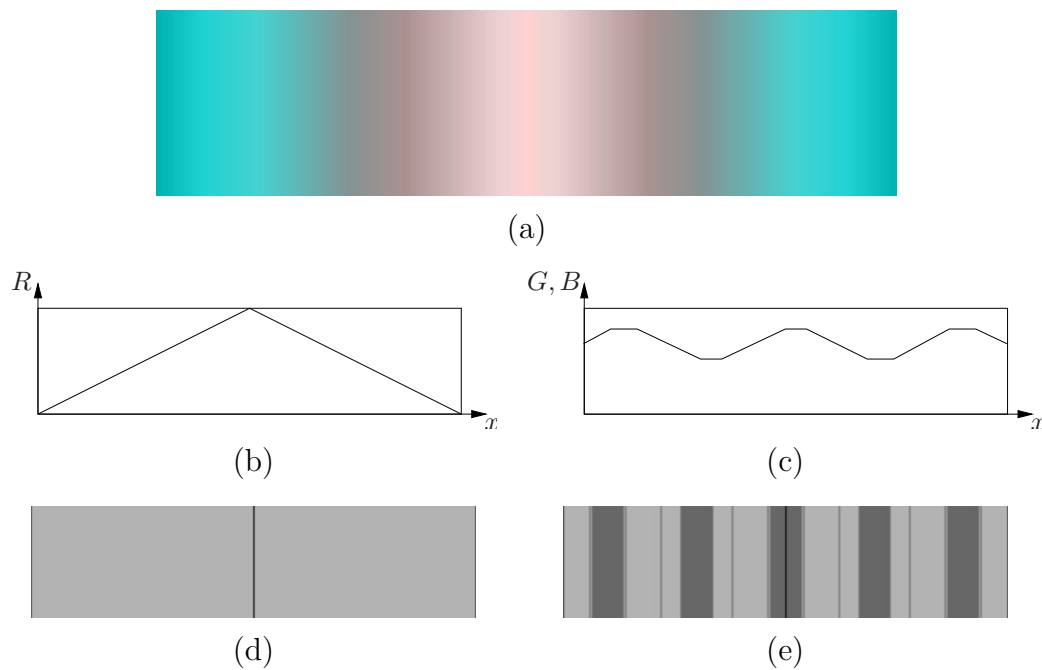


Figura 5.4: Color Contrast vs. Gradient Maximum. The image in (a) has the monochromatic profiles depicted in (b) for the red channel and in (c) for the green and blue channels. The maximum among all gradients is dominated by the red component, which suppresses the activities in all other channels (d). The color contrast can still detect such variations (e).

se sabe que el color recibido depende de las fuentes de luz, el material del objeto e incluso la respuesta espectral del sensor, por lo que la variabilidad en el color capturado es por métodos analíticos difícil de determinar.

Un método estadístico puede emplearse para entrenar un sistema a que distinga ciertos objetos de acuerdo a sus colores.

Para ello defínanse dos clases: la clase *objeto* y todo lo que no es objeto, la clase \neg *objeto*. Se desea encontrar para una imagen la probabilidad de que cada uno de sus píxeles pertenezca a la clase *objeto* o a la clase \neg *objeto*, de acuerdo únicamente a su color.

Para ello, se aproxima primero la probabilidad de que un determinado color c forma parte del objeto. Ésto se realiza por medio de un histograma generado con datos de entrenamiento. En él se acumulan los colores encontrados en píxeles que un usuario ha indicado previamente como parte del objeto. Estas probabilidades se denotan como $p(c | \text{objeto})$ y $p(c | \neg \text{objeto})$.

Obsérvese que con la notación de probabilidades condicionales lo que se busca es $p(\text{objeto} | c)$, es decir, la probabilidad de que dado un color c , el píxel pertenezca a la clase dada. Para ello se utiliza el teorema de Bayes, que establece

$$p(a | b) = \frac{p(b | a)p(a)}{p(b)} \quad (5.6)$$

donde al término $p(a | b)$ se le conoce como la probabilidad *a posteriori*, a $p(b | a)$ se le denomina valor de verosimilitud (o *likelihood* en inglés), $p(a)$ es la probabilidad *a priori* y $p(b)$ es una constante de normalización. Para el caso que compete, se tiene

$$p(\text{objeto} | c) = \frac{p(c | \text{objeto})p(\text{objeto})}{p(c)}$$

y utilizando las reglas de probabilidad se conoce que

$$p(c) = p(c | \text{objeto})p(\text{objeto}) + p(c | \neg\text{objeto})p(\neg\text{objeto})$$

donde además $p(\neg\text{objeto}) = 1 - p(\text{objeto})$.

Por hacer:

Generar imágenes de ejemplo

Capítulo 6

Morfología

6.1 Fundamentos

El procesamiento morfológico de imágenes utiliza como fuente de entrada la representación de imágenes como conjuntos de píxeles utilizada anteriormente:

$$\mathcal{I} = \{\mathbf{p} \mid \mathbf{p} = \langle \underline{\mathbf{x}}, \underline{\mathbf{c}} \rangle, \quad \underline{\mathbf{x}} \in \mathbb{X}, \quad \underline{\mathbf{c}} = f(\underline{\mathbf{x}})\}$$

En los siguientes casos se utilizan inicialmente imágenes binarias, donde los valores $\underline{\mathbf{c}}$ de los píxeles son escalares binarios, es decir cero o uno. Esto permite especificar la imagen como el conjunto de píxeles cuyo valor es uno, con lo que la imagen está enteramente especificada a través de las posiciones de dichos píxeles únicamente. En términos matemáticos, la imagen binaria \mathcal{B} está especificada por

$$\mathcal{B} = \{\underline{\mathbf{x}} \mid f(\underline{\mathbf{x}}) = 1\}$$

Puesto que la salida de un operador morfológico sigue siendo del mismo tipo que su entrada: un conjunto de píxeles, éstos pueden considerarse como operadores de procesamiento.

La morfología de imágenes tiene sus raíces en la teoría de conjuntos. Para poder definir los operadores más utilizados del área, es necesario establecer algunas operaciones básicas.

La *reflexión* $\hat{\mathcal{B}}$ de la imagen \mathcal{B} se define como

$$\hat{\mathcal{B}} = \{\underline{\mathbf{w}} \mid \underline{\mathbf{w}} = -\underline{\mathbf{x}}, \text{ para } \underline{\mathbf{x}} \in \mathcal{B}\}$$

Por hacer:

Figura de reflexión

La *traslación* en $\underline{\mathbf{z}}$ de un conjunto \mathcal{B} , denotada como $(\mathcal{B})_{\underline{\mathbf{z}}}$, se define como

$$(\mathcal{B})_{\underline{\mathbf{z}}} = \{\underline{\mathbf{c}} \mid \underline{\mathbf{c}} = \underline{\mathbf{x}} + \underline{\mathbf{z}}, \text{ para } \underline{\mathbf{x}} \in \mathcal{B}\}$$

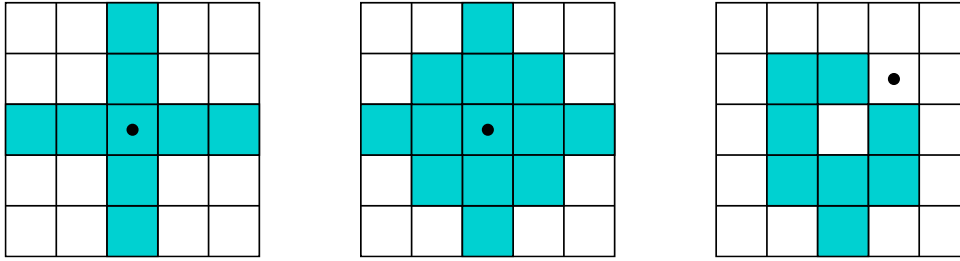
Por hacer:

Figura de traslación

La reflexión y traslación son operaciones básicas sobre los denominados *elementos estructurales* (EE), que indican de qué manera una vecindad de un píxel debe ser considerada en el cálculo de los nuevos píxeles. Usualmente éstos EE se representan por medio de máscaras rectangulares, donde se demarcan las posiciones que pertenecen a él.

Por hacer:

Figura de elementos estructurales

**Figura 6.1:** Elementos estructurales comunes

La figura 6.1 ilustra tres casos de EE, donde la posición (0,0) se muestra con un círculo negro. Obsérvese que el EE no necesariamente está centrado alrededor de (0,0).

Otra operación básica es el *complemento* de \mathcal{B} , denotado con \mathcal{B}^C , y constituido por

$$\mathcal{B}^C = \{\underline{\mathbf{w}} \mid \underline{\mathbf{w}} \notin \mathcal{B}\}$$

6.2 Dilatación y Erosión

La erosión de \mathcal{A} por el elemento estructural \mathcal{B} se denota con $\mathcal{A} \ominus \mathcal{B}$ y se define como

$$\mathcal{A} \ominus \mathcal{B} = \{\underline{\mathbf{x}} \mid (\mathcal{B})_{\underline{\mathbf{x}}} \subseteq \mathcal{A}\} \quad (6.1)$$

$$= \{\underline{\mathbf{x}} \mid (\mathcal{B})_{\underline{\mathbf{x}}} \cap \mathcal{A}^C = \emptyset\} \quad (6.2)$$

En otras palabras, el conjunto resultante de la erosión es aquel de todos los puntos $\underline{\mathbf{x}}$ tales que si el EE \mathcal{B} se desplaza en $\underline{\mathbf{x}}$, todos sus elementos forman parte de \mathcal{A} .

Por hacer:

Ejemplo de erosión

La dilatación de \mathcal{A} con el elemento estructural \mathcal{B} , denotada como $\mathcal{A} \oplus \mathcal{B}$ se define como

$$\begin{aligned} \mathcal{A} \oplus \mathcal{B} &= \{\underline{\mathbf{z}} \mid (\hat{\mathcal{B}})_{\underline{\mathbf{z}}} \cap \mathcal{A} \neq \emptyset\} \\ &= \{\underline{\mathbf{z}} \mid [(\hat{\mathcal{B}})_{\underline{\mathbf{z}}} \cap \mathcal{A}] \subseteq \mathcal{A}\} \end{aligned}$$

Obsérvese que en el proceso de dilatación el elemento estructural es reflejado con respecto a su origen antes de ser trasladado. La dilatación de \mathcal{A} por \mathcal{B} es el conjunto de todos los desplazamientos \underline{x} tales que $\hat{\mathcal{B}}$ y \mathcal{A} se traslapan en al menos un elemento.

La dilatación y la erosión son operaciones duales entre sí, con respecto al complemento y reflexión de conjuntos. Esto es

$$(\mathcal{A} \ominus \mathcal{B})^C = \mathcal{A}^C \oplus \hat{\mathcal{B}}$$

y

$$(\mathcal{A} \oplus \mathcal{B})^C = \mathcal{A}^C \ominus \hat{\mathcal{B}}$$

Obsérvese que para EE simétricos que cumplen $\mathcal{B} = \hat{\mathcal{B}}$, la erosión de una imagen con \mathcal{B} se obtiene dilatando el fondo de la imagen \mathcal{A}^C .

6.3 Apertura y Clausura

La apertura y la clausura son operadores compuestos de erosiones y dilataciones. La apertura usualmente suaviza el contorno de objetos, eliminando *istmos* delgados, y pequeñas *penínsulas*. La clausura fusiona pequeños “agujeros” en los bordes o *golfos* delgados, así como cierra agujeros y rendijas.

La apertura del conjunto \mathcal{A} por el EE \mathcal{B} se denota como $\mathcal{A} \circ \mathcal{B}$ y se calcula con

$$\mathcal{A} \circ \mathcal{B} = (\mathcal{A} \ominus \mathcal{B}) \oplus \mathcal{B}$$

es decir, la apertura se obtiene erosionando \mathcal{A} con \mathcal{B} y luego dilatando el resultado con \mathcal{B} . La apertura tiene las siguientes propiedades:

- $\mathcal{A} \circ \mathcal{B}$ es un subconjunto de \mathcal{A}
- Si \mathcal{C} es un subconjunto de \mathcal{D} , entonces $\mathcal{C} \circ \mathcal{B}$ es un subconjunto de $\mathcal{D} \circ \mathcal{B}$
- $(\mathcal{A} \circ \mathcal{B}) \circ \mathcal{B} = \mathcal{A} \circ \mathcal{B}$

La clausura del conjunto \mathcal{A} con el EE \mathcal{B} se denota con $\mathcal{A} \bullet \mathcal{B}$ y está definida como

$$\mathcal{A} \bullet \mathcal{B} = (\mathcal{A} \oplus \mathcal{B}) \ominus \mathcal{B}$$

lo que indica que clausurar con \mathcal{B} se equivalente a dilatar primero con \mathcal{B} , y al resultado erosionarlo por el mismo elemento estructural. La clausura tiene las siguientes propiedades:

- \mathcal{A} es un subconjunto de $\mathcal{A} \bullet \mathcal{B}$
- Si \mathcal{C} es un subconjunto de \mathcal{D} , entonces $\mathcal{C} \bullet \mathcal{B}$ es un subconjunto de $\mathcal{D} \bullet \mathcal{B}$
- $(\mathcal{A} \bullet \mathcal{B}) \bullet \mathcal{B} = \mathcal{A} \bullet \mathcal{B}$

La clausura y la apertura son duales entre sí, con respecto al complemento y reflexión. Esto es

$$\begin{aligned}(\mathcal{A} \bullet \mathcal{B})^C &= (\mathcal{A}^C \circ \hat{\mathcal{B}}) \\ (\mathcal{A} \circ \mathcal{B})^C &= (\mathcal{A}^C \bullet \hat{\mathcal{B}})\end{aligned}$$

6.4 Transformación de añadir o eliminar

La transformación de añadir-o-eliminar (*hit or miss*) permite detectar regiones con una forma específica. Para ello, se utiliza un elemento estructural compuesto $(\mathcal{B}_1, \mathcal{B}_2)$, tal que $\mathcal{B}_1 \cap \mathcal{B}_2 = \emptyset$. \mathcal{B}_1 describe la forma u objeto que se desea encontrar y \mathcal{B}_2 se asocia al borde de \mathcal{B}_1 . Esta transformación se define entonces como

$$\begin{aligned}\mathcal{A} \circledast (\mathcal{B}_1, \mathcal{B}_2) &= (\mathcal{A} \ominus \mathcal{B}_1) \cap (\mathcal{A}^C \ominus \mathcal{B}_2) \\ &= (\mathcal{A} \ominus \mathcal{B}_1) \cap (\mathcal{A} \oplus \mathcal{B}_2)^C \\ &= (\mathcal{A} \ominus \mathcal{B}_1) - (\mathcal{A} \oplus \hat{\mathcal{B}}_2)\end{aligned}$$

Obsérvese que es posible definir la dilatación y la erosión en términos de ésta transformación.

6.5 Algoritmos

6.5.1 Extracción Morfológica de Bordos

Un primer algoritmo es la extracción de bordes, obtenido como

$$\beta(\mathcal{A}) = \mathcal{A} - (\mathcal{A} \ominus \mathcal{B})$$

6.5.2 Extracción de componentes conectados

Con \mathcal{B} un EE cuadrado de dimensiones 3×3 , es posible encontrar todos los elementos conectados a un conjunto inicial de puntos \mathcal{X}_0 con el algoritmo iterativo

$$\mathcal{X}_k = (\mathcal{X}_{k-1} \oplus \mathcal{B}) \cap \mathcal{A} \quad k = 1, 2, 3, \dots$$

Por hacer:

lti::fastRelabeling

6.5.3 Adelgazamiento y ensanchado

El adelgazamiento se realiza con un elemento estructural compuesto $(\mathcal{B}_1, \mathcal{B}_2)$ y se define como

$$\begin{aligned}\mathcal{A} \odot (\mathcal{B}_1, \mathcal{B}_2) &= \mathcal{A} - \mathcal{A} * (\mathcal{B}_1, \mathcal{B}_2) \\ &= \mathcal{A} \cap (\mathcal{A} * (\mathcal{B}_1, \mathcal{B}_2))^C\end{aligned}$$

El ensanchamiento se define también para un elemento estructural compuesto como

$$\mathcal{A} \oslash (\mathcal{B}_1, \mathcal{B}_2) = \mathcal{A} \cup (\mathcal{A} \odot (\mathcal{B}_1, \mathcal{B}_2))$$

6.5.4 Esqueleto

El esqueleto $S(\mathcal{A})$ está definido como el conjunto de todos los puntos \underline{z} tales que si $(\mathcal{D})_{\underline{z}}$ es el mayor disco centrado en \underline{z} y contenido en \mathcal{A} (el disco máximo), entonces $\mathcal{D})_{\underline{z}}$ toca el borde de \mathcal{A} en al menos dos puntos diferentes.

6.5.5 Transformación de distancia

La transformación de distancia es un operador morfológico que asigna a cada píxel de \mathcal{A} la distancia más cercana a un punto de \mathcal{A}^C .

6.6 Morfología con escalas de grises

Las operaciones morfológicas se extienden a imágenes con valores de gris. Se identifican dos tipos

- Elemento estructural plano
- Elemento estructural no plano

6.6.1 Elementos planos

Erosión:

$$[f \ominus b](x, y) = \min_{(s,t) \in b} \{f(x + s, y + t)\}$$

Dilatación:

$$[f \oplus b](x, y) = \max_{(s,t) \in b} \{f(x - s, y - t)\}$$

6.6.2 Elementos no planos

Erosión:

$$[f \ominus g](x, y) = \min_{(s,t) \in g} \{f(x + s, y + t) - g(s, t)\}$$

Dilatación:

$$[f \oplus g](x, y) = \max_{(s,t) \in g} \{f(x - s, y - t) + g(s, t)\}$$

Capítulo 7

Algoritmos de análisis de imágenes

7.1 Detección de discontinuidades

En imágenes la información está principalmente codificada en discontinuidades. Experimentos con reconstrucción de imágenes considerando combinaciones de fase y magnitud de fuentes diversas muestran, además de que la fase es la portadora de esta información, que con únicamente con las discontinuidades es posible reconocer el contenido de la imagen. Esto justifica las abundantes propuestas de algoritmos de detección de discontinuidades, tanto ya establecidas, como recientes.

Se distinguen tres tipos de discontinuidades:

1. bordes
2. líneas
3. esquinas

Un *borde* ideal existe entre dos regiones homogéneas, tal que a lo largo del borde, y perpendicular a éste, la función que representa a la imagen describe un escalón en su valor, y sobre los píxeles del borde la función es aproximadamente constante.

Una *línea* es una discontinuidad entre dos funciones homogéneas, donde en la perpendicular a la línea se presentan discontinuidades a ambos lados del píxel.

Una *esquina* es un píxel que pertenece al borde de dos o más direcciones.

7.1.1 Detección de Bordes

Un borde es un conjunto de píxeles de borde adyacentes, y un píxel de borde es aquel que presenta un cambio abrupto de una característica en *una* dirección particular.

Algoritmo de Marr-Hildred

Este detector ya se encuentra prácticamente en desuso, pero es interesante desde el punto de vista de su origen. David Marr y Ellen Hildred propusieron utilizar el laplaciano de gaussianas (LoG) para la detección preliminar visual de bordes. Como aproximación de dicho operador los autores mostraron que la diferencia de gaussianas (filtros de sombrero mexicano) puede aproximar la respuesta de campos receptivos de las células ganglionares en la retina de un gato. Luego de aplicar el LoG se utiliza un detector de cruces por cero par obtener los bordes.

Éste operador tiene dos limitaciones: genera respuestas que no corresponden con bordes (falsos bordes), el error de localización puede ser considerable en bordes curvos. Sin embargo, tiene además la ventaja de que genera siempre contornos cerrados.

Algoritmo de Canny

Algoritmo de Canny [9] es un algoritmo compuesto de varias fases, y diseñado para ser óptimo con respecto a tres criterios:

1. El criterio de *baja tasa de error* indica que todo borde fuerte debe ser detectado y no deben haber respuestas espurias.
2. El criterio de *localización* establece que la distancia entre la posición real del borde y la posición detectada debe ser mínima.
3. El criterio de *monorespuesta* minimiza múltiples respuestas ante un único borde.

Nótese que el tercer criterio está relacionado con el primero, puesto que si un borde en la escena capturada produce varias respuestas, todas excepto una serán falsas.

En el desarrollo del algoritmo se parte de una señal unidimensional, para la que se encuentra una solución cerrada en el proceso de considerar los criterios uno y dos. Cuando se agrega el tercer criterio, se encuentra otra solución por métodos numéricos, que puede ser aproximada con un error menor al 20% utilizando la primera derivada de la curva gaussiana

$$\frac{d}{dx} e^{-\frac{1}{2} \frac{x^2}{\sigma^2}} = -\frac{x}{\sigma^2} e^{-\frac{1}{2} \frac{x^2}{\sigma^2}}$$

Los conceptos anteriores se extienden al caso bidimensional, aplicando los desarrollos unidimensionales sobre la dirección perpendicular al borde.

A partir de la curva gaussiana

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{1}{2} \frac{x^2+y^2}{\sigma^2}}$$

es posible encontrar su derivada en la dirección \mathbf{n} con

$$G_{\mathbf{n}} = \frac{\partial}{\partial \mathbf{n}} = \mathbf{n} \cdot \nabla G$$

La dirección $\underline{\mathbf{n}}$ debe encontrarse perpendicular al borde, cuya dirección es inicialmente desconocida, pero puede estimarse para la imagen f como

$$\underline{\mathbf{n}} = \frac{\nabla(G * f)}{|\nabla(G * f)|} \quad (7.1)$$

Los bordes se encuentran sobre los máximos locales de $G_{\underline{\mathbf{n}}} * f$ en la dirección $\underline{\mathbf{n}}$, donde se cumple

$$\frac{\partial}{\partial \underline{\mathbf{n}}} G_{\underline{\mathbf{n}}} * f = \frac{\partial^2}{\partial \underline{\mathbf{n}}^2} G * f = 0 \quad (7.2)$$

Por la asociatividad entre la convolución y la derivación es posible convolucionar primero la imagen con la función gaussiana G y posteriormente calcular la segunda derivada en la dirección $\underline{\mathbf{n}}$ calculada con (7.1). Al proceso de eliminar todo aquello que no consiste en borde, es decir, que no cumple con (7.2). A cada píxel del borde se le asigna un valor de peso igual a la magnitud del gradiente

$$|G_{\underline{\mathbf{n}}} * f| = |\nabla(G * f)|$$

El ruido en la imagen causa respuestas espurias y puede cortar bordes fuertes. Para decidir cuáles píxeles del paso anterior son definitivamente bordes, se utiliza un proceso de umbralización con histéresis: si el valor de respuesta de un píxel supera un umbral superior, entonces dicho píxel se asume como borde. Si el valor de respuesta del píxel es menor que el umbral superior, pero mayor que otro umbral inferior y además dicho píxel está conectado en su vecindad con otro píxel de borde, entonces se asume que también pertenece al borde. Dichos umbrales se pueden determinar teóricamente si se conoce la razón señal a ruido de la imagen.

El algoritmo de Canny se resume en la figura 7.1.

1. Convolucionar la imagen f con la máscara gaussiana G con desviación estándar σ .
2. Calcular la magnitud y fase del gradiente para la salida del paso anterior.
3. Encontrar píxeles candidatos para los bordes utilizando la supresión de no-máximos (7.2).
4. Utilizar la umbralización con histéresis para decidir cuáles candidatos o son bordes.

Figura 7.1: Algoritmo de Canny para detección de bordes.

7.1.2 Detección de esquinas

Haris

Surf

7.1.3 Detección de líneas

7.2 Transformada de Hough para líneas

La detección de bordes utilizada en los algoritmos de Marr-Hildred o Canny tienen sentido cuando no hay conocimiento previo de la forma geométrica de los bordes que se busca. En otras aplicaciones la forma es previamente conocida, y otras familias de métodos se han presentado en la literatura.

La transformada de Hough permite en su versión más simple encontrar líneas o círculos en imágenes.

Para la detección de líneas rectas, debe explicarse primero el espacio paramétrico polar, como se describe en la figura 7.2. La línea recta que pasa por los puntos (x_i, y_i) y (x_j, y_j) se puede describir por la menor distancia entre esa línea y el origen, así como el ángulo formado entre el rayo uniendo que une al origen con el punto más cercano sobre la línea.

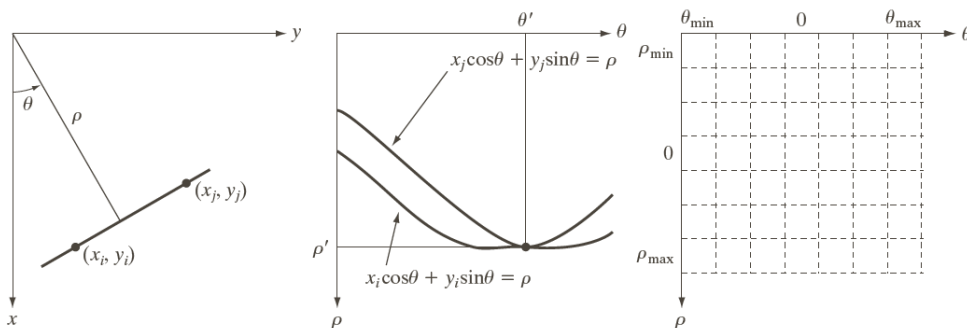


Figura 7.2: Espacio paramétrico utilizado en la transformada de Hough (tomado de [29])

La transformada de Hough es un concepto basado en votación, donde cada píxel de la imagen vota por todas las formas a evaluar que lo podrían componer; así, en el caso de las líneas, cada punto vota por todas las líneas que pueden pasar por él. La votación se realiza sobre el espacio paramétrico, donde puede demostrarse que el punto (x_k, y_k) está incluido en todos los pares (ρ, θ) que satisfacen:

$$\rho = x_k \cos \theta + y_k \sin \theta$$

La transformada de Hough se sintetiza con los siguientes pasos:

1. Obténgase una imagen de bordes binaria utilizando cualquiera de las técnicas anteriores

2. Segmentar el espacio $\rho\sigma$ en una rejilla regular
3. Acumular en la rejilla los votos para cada píxel de borde
4. Determinar las celdas de la rejilla con mayor número de votos.
5. Elegir las estructuras en la imagen original de acuerdo a los votos en la rejilla

Opcionalmente, pueden establecerse mecanismos para determinar segmentos de recta.

Una forma de acelerar los cálculos es, en el proceso de acumulación, considerar el ángulo del gradiente calculado en la determinación del borde. Esto restringe el número de celdas donde se acumularán los votos, a aquellas cercanas al ángulo del borde.

El caso de la transformada de Hough para detectar círculos funciona de manera similar. Para círculos de radio constante, el espacio paramétrico también tiene dos dimensiones, con las coordenadas x e y del centro del círculo. Si además el radio debe ser determinado, el espacio paramétrico es tridimensional.

7.3 Segmentación

El problema de *segmentación* ha recibido bastante atención en los últimos años. Cientos de publicaciones aparecen cada año, intentando encontrar soluciones óptimas para aplicaciones específicas. La mayoría de autores coinciden en los siguientes hechos acerca de esta tarea:

- el objetivo es partir la imagen en varios segmentos o regiones
- es una de las primeras tareas de los sistemas de visión por computador, considerado parte de las etapas de preproceso.
- es una etapa crítica en los sistemas de análisis, debido a que usualmente se siguen arquitecturas de procesamiento secuenciales, que tienen a la segmentación como una de las etapas iniciales, y sus resultados afectan toda etapa posterior.

Exactamente qué significa un *segmento* no está claramente definido. Algunos autores como Sonka et al. [73] definen el objetivo de la segmentación como dividir la imagen en partes que tienen alta correlación con objetos o áreas en el mundo real capturadas en la imagen. Se distinguen dos tipos de segmentación: parcial y completa, donde la primera encuentra regiones en la imagen que corresponden a superficies, sin agruparlas en objetos, mientras que la segunda detecta los objetos reales.

Puesto que el significado de *objeto real* depende de cada aplicación, otros autores (por ejemplo, Jain [37], González and Woods [28]) prefieren referirse a la segmentación como la tarea de descomponer la imagen en sus *componentes*, donde el término componente tiene que entenderse como una representación compacta de una parte de la imagen, apropiada para el contexto en particular [24]. Por la dependencia de la aplicación, otros autores, como Forsyth [23] indican que la segmentación y el reconocimiento de objetos son en esencia la misma tarea.

Otros autores, incluyendo [53, 70, 42], definen segmentación en términos de un criterio de homogeneidad sobre características de bajo nivel. Este criterio deben ser satisfecho dentro de cada región, pero debe ser violado en tanto se fusionen dos regiones adyacentes. Esta definición es la utilizada en este documento, y se denomina segmentación *a nivel de imagen*, en contraste con las segmentaciones a niveles de superficies u objetos, que requieren otras estrategias para incorporar información del contexto, no incluida directamente en la imagen. La figura 7.3 muestra resultados de segmentación en distintos niveles. En el nivel de imagen, características como posición de píxeles y su valor de gris se utilizan. El objeto “fondo” está compuesto por varios parches que no comparten dichas características. Adicionalmente, la sombra proyectada en la base del cono tiene un tono similar al cono, aunque ambas regiones pertenezcan a objetos diferentes.

Para definir formalmente la tarea de segmentación se requiere retomar la notación de imágenes como conjuntos detallada en la sección 2.10.1, así como las siguientes definiciones.

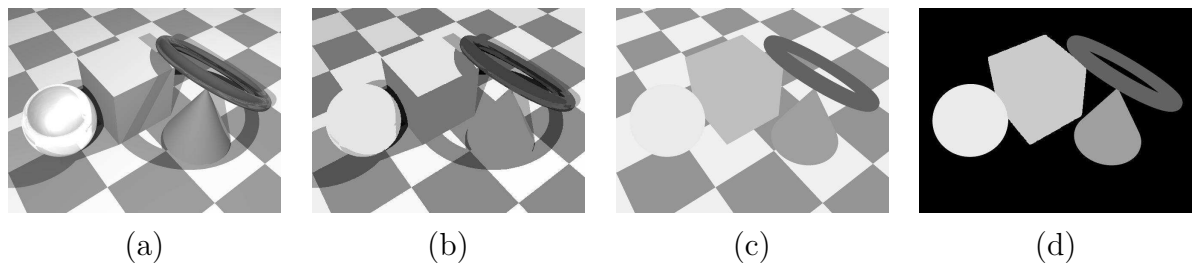


Figura 7.3: Resultados de segmentación a distintos niveles. (a) Imagen original (b) nivel de imagen, (c) nivel de superficies, y (d) nivel de objetos.

Una *region* \mathcal{R} en una imagen es un subconjunto no vacío de la imagen \mathcal{I} , esto es $\mathcal{R} \subseteq \mathcal{I}$, $\mathcal{R} \neq \emptyset$. Una región no requiere ser topológicamente conexa, y por tanto dos partes visiblemente separadas pertenecientes a un objeto parcialmente oculto detrás de otro pueden pertenecer a la misma región.

La partición \mathcal{P} de una imagen \mathcal{I} es un conjunto de n regiones $\{\mathcal{R}_i; i = 1 \dots n\}$ tales que $\bigcup_{i=1}^n \mathcal{R}_i = \mathcal{I}$ y $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ para $i \neq j$. Una partición se dice ser más *fina* que otra si parte a la imagen en regiones más pequeñas.

A nivel de imágenes no existe concepto para el término *objeto*. En vez de ello, se determinan componentes o partes utilizando un *predicado* de uniformidad u homogeneidad H_I para alguna característica específica, y otro predicado que evalúa la adyacencia de regiones A .

La segmentación basada en imagen \mathcal{S}_I de una imagen \mathcal{I} es una partición de \mathcal{I} que satisface para cada región $\mathcal{R}_i \in \mathcal{S}_I$, $H_I(\mathcal{R}_i)$, y $\neg H_I(\mathcal{R}_i \cup \mathcal{R}_j)$ para $A(\mathcal{R}_i, \mathcal{R}_j)$.

La condición $H(\mathcal{R}_i \cup \mathcal{R}_j)$ previene partir la imagen en demasiadas regiones (sobre-segmentación): si el resultado de fusionar cualquier par de regiones adyacentes es inhomogéneo, entonces eso implica que la segmentación ya contiene el mayor número de regiones que satisfacen el predicado de homogeneidad. Todas las estrategias de segmentación a nivel de imagen se puede considerar como una implementación particular del predicado H . En ellos se agregan restricciones adicionales a la definición región, permitiendo especificar aún más las propiedades que debe cumplir una región válida. Por ejemplo, una condición usualmente impuesta a H es la conectividad de las regiones.

La figura 7.4 muestra la categorización de métodos de segmentación a nivel de imagen. Tres clases se definen: basadas en el espacio de características, basadas en el dominio de la imagen, y métodos híbridos.

7.3.1 Métodos basados en el espacio de características

Feature-space approaches generally neglect spatial relationships between image elements and analyze exclusively the configuration of their feature vectors \mathbf{c} . Algorithms in this category delimit sections in the feature space and assign the same region label to all image

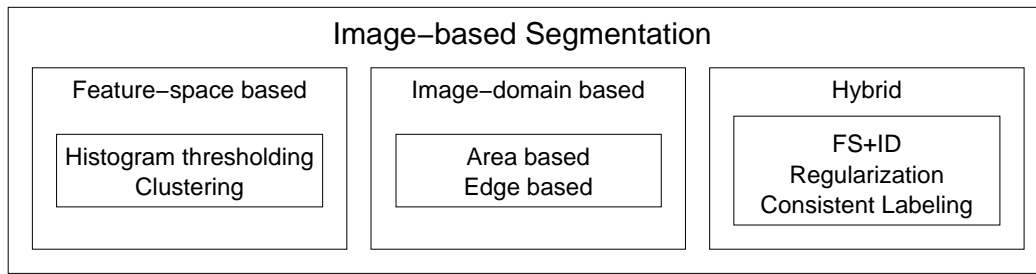


Figura 7.4: Categorización de segmentación algoritmos at the image-based processing level (FS: Feature-space based; ID: Image-domain based).

elements falling into the same section. Two principles are common. The first one finds sections detecting peaks in unidimensional or multidimensional feature histograms. The second one uses traditional clustering algorithms (Figure 7.5).

Detección de umbral

En la sección 3.3 se indicó la binarización de una imagen con una transformación de niveles de gris de la forma:

$$s = T(r) = \begin{cases} 0 & \text{si } r < \tau \\ 1 & \text{si } r \geq \tau \end{cases}$$

Esto se puede considerar como técnica de segmentación, en donde las regiones detectadas tienen etiquetas cero o uno. Las estrategia de segmentación por *detección* de umbral lo que persiguen es automatizar la selección de τ basándose en el contenido de la imagen.

Si se puede asumir que la distribución de los valores de niveles de gris de cada uno de los componentes en la imagen siguen una distribución gaussiana

$$p_i(g) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{g-\mu_i}{\sigma_i}\right)^2}$$

entonces la distribución de probabilidades de niveles de gris para la imagen completa es

$$p(g) = \alpha_1 p_1(g) + \alpha_2 p_2(g) \quad \alpha_1 + \alpha_2 = 1$$

Por hacer:

Demostrar lo que sigue con teoría de decisión

De teoría de la decisión se deduce que la selección óptima del umbral τ se da cuando se cumple

$$\alpha_1 p_1(\tau) = \alpha_2 p_2(\tau)$$

En el caso particular de que $\sigma_1 = \sigma_2 = \sigma$ el umbral se calcula como

$$\tau = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln \frac{\alpha_2}{\alpha_1}$$

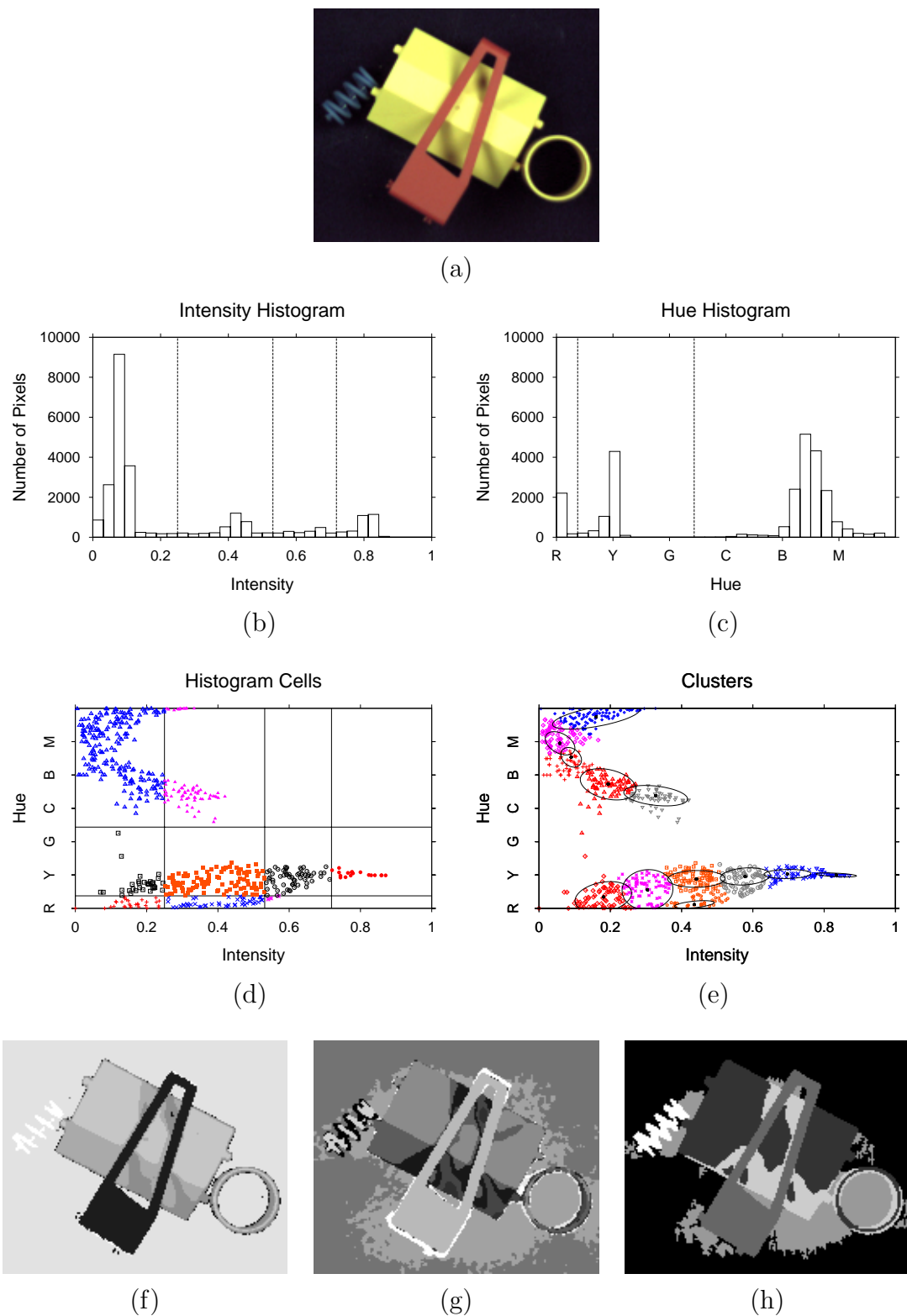


Figura 7.5: Segmentación en el espacio de características. (a) Imagen original. Métodos basados en histogramas calculan en ocasiones histogramas unidimensionales (b,c) y detectan picos y sus regiones, realizando particiones regulares del espacio de características (d). Algoritmos de aglomeración detectan conglomerados en el espacio de características (e). Los resultados basados en histogramas (f) y en aglomeración (g) difieren. Los detalles se mantienen mejor con la aglomeración, pero los bordes detectados en la imagen suelen ser ruidosos y requieren procesamiento adicional. El resultado utilizando desplazamiento de medias se ilustra en (h).

y se observa que utilizar la media de las medias es aplicable en teoría solo cuando ambas regiones son constantes en nivel de gris ($\sigma = 0$) o cuando las amplitudes de las dos funciones gaussianas es igual ($\alpha_1 = \alpha_2$).

Un algoritmo simple para estimar τ sigue los siguientes pasos:

1. Seleccione un estimado inicial del umbral τ
2. Segmente la imagen utilizando τ
3. Estime la media de los valores de gris para las dos regiones resultantes
4. Calcule el nuevo umbral como

$$\tau = \frac{1}{2}(\mu_1 + \mu_2)$$

5. Repita los pasos desde 2 hasta 4, hasta que la diferencia en el valor del umbral sea menor que un parámetro predeterminado $\Delta\tau$.

El algoritmo de Otsu selecciona el umbral de modo que se maximice la varianza entre clases. El algoritmo tiene lo siguientes pasos:

1. Calcule el histograma normalizado de la imagen, como aproximación de la densidad de probabilidad de valores de gris $p(g)$
2. Calcule las probabilidades de pertenecer a la clase oscura o clase fondo C_1 que tiene valores menores al umbral τ para todos los valores discretos de gris $P_1(\tau)$, para $\tau = 0, 1, 2, \dots, L - 1$:

$$P_1(\tau) = \sum_{i=0}^{\tau} p(i)$$

3. Calcule las medias acumulativas $\mu(k)$ para $\tau = 0, 1, 2, \dots, L - 1$

$$\mu(\tau) = \sum_{i=0}^{\tau} ip(i)$$

4. Calcule la media de intensidad global

$$\mu_G = \sum_{i=0}^{L-1} ip(i)$$

5. Calcule la varianza entre clases $\sigma_B^2(\tau)$ para $\tau = 0, 1, 2, \dots, L - 1$ con

$$\sigma_B^2(\tau) = \frac{[\mu_G P_1(\tau) - \mu(\tau)]^2}{P_1(\tau)[1 - P_1(\tau)]}$$

6. Determine el umbral de Otsu τ_o como el valor de τ para el que $\sigma_B^2(\tau)$ es máximo. Si el máximo no es único, se selecciona τ_o como el promedio de todos los máximos.

Métodos basados en histogramas

Histogram peak detection algorithms have a long history beginning with gray-valued histogram thresholding (e. g. [51, 52, 33, 75, 10, 67, 56, 41]). Early methods for color

segmentation work with several one-dimensional histograms, which implies that the correlation between different dimensions is ignored. More recent algorithms work in two or three dimensional color spaces and are characterized by different techniques to robustly detect peaks and their corresponding boundaries in the feature space. Peak detection is in principle a difficult task easily affected by noise. An additional problem of this approach is the usually required sub-sampling of the feature space in order to keep the size of data structures tractable. Many algorithms search for peaks by approximating the histograms with a mixture of Gaussians, and fail if this assumption does not hold (a fact that, in real images, is almost always the case).

Métodos basados en aglomeración

Clustering approaches can be interpreted as unsupervised classification methods. Several concepts are based on the k -means and fuzzy c -means clustering algorithms [44, 19] applied on different color and texture spaces [48, 60, 79, 43, 84, 82]. One of the major drawbacks of the original clustering methods is that the number of clusters (k or c) must be known a-priori. Several heuristics have been suggested to compute k automatically based on some image statistics (e. g. [59, 82]). To achieve illumination invariance, a modification to the k -means algorithm is suggested in [80] to use an angular distance in a zero-mean color space instead of the classical euclidean distance.

D. Comaniciu and P. Meer [13] propose a segmentation algorithm based on mean-shift clustering, a concept that relies on the *mean-shift procedure* to detect the modes of highest density in a multidimensional feature space and their corresponding *basins of attraction*.

Given n feature points $\underline{\mathbf{c}}_1 \dots \underline{\mathbf{c}}_n$ in a d -dimensional space \mathbb{R}^d , the kernel density estimation at the point $\underline{\mathbf{c}}$, with a kernel $K(\underline{\mathbf{c}})$ and a bandwidth parameter h , is given by

$$\hat{f}(\underline{\mathbf{c}}) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\underline{\mathbf{c}} - \underline{\mathbf{c}}_i}{h}\right). \quad (7.3)$$

Clusters in the feature space can be detected by searching for maxima in the density function $\hat{f}(\underline{\mathbf{c}})$. In other words, the modes of the density can be found at those feature points $\underline{\mathbf{c}}$ for which the gradient $\nabla \hat{f}(\underline{\mathbf{c}})$ becomes zero. The so-called mean-shift procedure allows to find these points without requiring the explicit computation of \hat{f} or its gradient. Only the derivatives of the kernel $K(\underline{\mathbf{c}})$ are necessary. The class of kernels with the form $K(\underline{\mathbf{c}}) = \alpha k(\|\underline{\mathbf{c}}\|^2)$, with α a normalization constant and $k(x)$ a *profile* function are especially adequate. The Epanechnikov profile

$$k_E(x) = \begin{cases} 1 - x & 0 \leq x \\ 0 & x > 1 \end{cases} \quad (7.4)$$

minimizes the mean square error between the real density and its estimate asymptotically, i. e. if the number of data points $n \rightarrow \infty$. The profile

$$k_N(x) = \exp\left(-\frac{1}{2}x\right) \quad (7.5)$$

yields the normal kernel

$$K_N(\underline{\mathbf{c}}) = (2\pi)^{-d/2} \exp\left(-\frac{1}{2}\|\underline{\mathbf{c}}\|^2\right). \quad (7.6)$$

which is often preferred as it is differentiable for all x .

The computation of the density relies on a function $g(x) = -k'(x)$, with which a new kernel $G(\underline{\mathbf{c}})$ can be defined:

$$G(\underline{\mathbf{c}}) = \beta g(\|\underline{\mathbf{c}}\|^2) \quad (7.7)$$

where β is also a normalization constant. From the Epanechnikov profile the d -dimensional unit sphere is obtained, while $G_N(\underline{\mathbf{c}}) = -\beta k'_N(\|\underline{\mathbf{c}}\|^2)$ keeps the form of $K_N(\underline{\mathbf{c}})$. The mean value of the feature points within a window delimited by the kernel $G(\underline{\mathbf{c}})$ and centered at the point $\underline{\mathbf{c}}$ is

$$\underline{\boldsymbol{\mu}}_{h,G}(\underline{\mathbf{c}}) = \frac{\sum_{i=1}^n \underline{\mathbf{c}}_i g\left(\left\|\frac{\underline{\mathbf{c}}-\underline{\mathbf{c}}_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{\underline{\mathbf{c}}-\underline{\mathbf{c}}_i}{h}\right\|^2\right)}. \quad (7.8)$$

The mean-shift vector is then defined as

$$\underline{\mathbf{m}}_{h,G}(\underline{\mathbf{c}}) = \underline{\boldsymbol{\mu}}_{h,G}(\underline{\mathbf{c}}) - \underline{\mathbf{c}}. \quad (7.9)$$

This vector points always toward the direction of maximum increase in the density.

The localization of the modes is finally achieved with an iterative process: first, the kernel is placed at an arbitrary point $\underline{\mathbf{c}}$ of the feature space; then, the mean-shift vector $\underline{\mathbf{m}}_{h,G}(\underline{\mathbf{c}})$ is computed and used to translate the kernel window $G(\underline{\mathbf{c}})$. At the new position the mean-shift vector is re-computed followed by the corresponding kernel shift, and so on. Hence, the path traced from $\underline{\mathbf{c}}$ to the density mode can be expressed as a sequence of points $\underline{\mathbf{s}}_t$:

$$\begin{aligned} \underline{\mathbf{s}}_0 &= \underline{\mathbf{c}} \\ \underline{\mathbf{s}}_{t+1} &= \underline{\boldsymbol{\mu}}_{h,G}(\underline{\mathbf{s}}_t) \end{aligned} \quad (7.10)$$

For convex and monotonically decreasing profiles $k(x)$ this iterative *mean-shift procedure* converges to one mode of the estimated density function $\hat{f}(\underline{\mathbf{c}})$ [13]. Figure 7.6 shows a simple example in a two-dimensional feature space.

A cluster in the feature space is detected by grouping all those locations with paths converging to the same mode (Figure 7.6c). Since an image generally consists of several thousand pixels, finding the paths for each single one is a computationally expensive task. Therefore, the feature space is usually quantized to reduce the data volume the algorithm needs to deal with. Additionally, techniques for multidimensional range searching are necessary in order to efficiently find the points falling in the window specified by the kernel.

Comaniciu and Meer choose as feature space a joint domain combining spatial and color information. Since both subspaces are different in nature, a kernel has been selected that

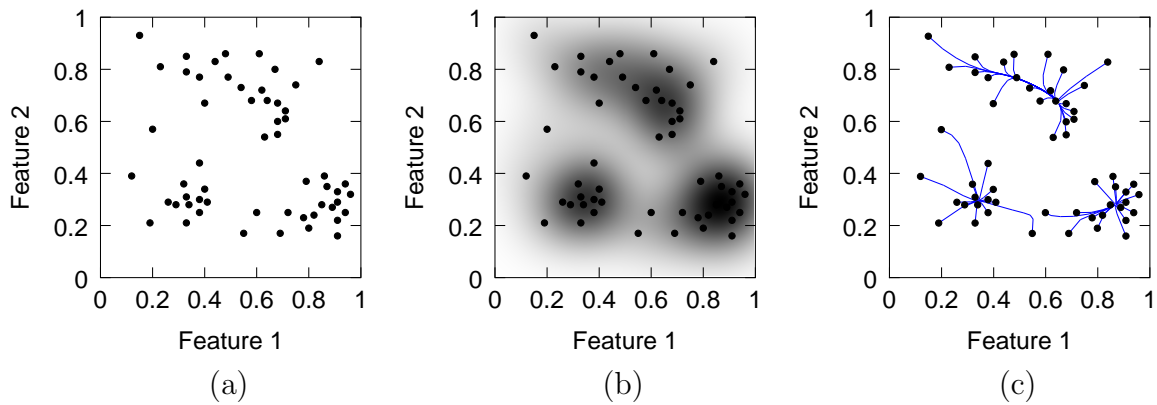


Figura 7.6: La aglomeración por desplazamiento de medias en un espacio de características bidimensional. (a) 56 puntos característicos. (b) Densidad estimada con (7.3), un ancho de banda $h = 0.1$ y un kernel normal. La densidad es proporcional a la tinción negra. (c) Tres conglomerados detectados y sus rutas respectivas de los pasos de desplazamiento de medias para cada punto. Aproximadamente seis iteraciones por punto fueron requeridas.

allows a separate consideration of both components:

$$K_{h_p, h_c}(\underline{\mathbf{p}}, \underline{\mathbf{c}}) = \frac{C}{h_p^2 h_c^d} k\left(\left\|\frac{\underline{\mathbf{p}}}{h_p}\right\|^2\right) k\left(\left\|\frac{\underline{\mathbf{c}}}{h_c}\right\|^2\right) \quad (7.11)$$

where $\underline{\mathbf{p}}$ represents the pixel position and $\underline{\mathbf{c}}$ its d -dimensional feature value (a scalar gray value or a three-dimensional vector in the CIE $L^*u^*v^*$ color space).

The first stage of the mean-shift segmentation is a discontinuity preserving smoothing. It reduces noise and suppresses fine textures that are irrelevant for the segmentation task. Let $\hat{e}_i = \langle \hat{\underline{\mathbf{p}}}_i, \hat{\underline{\mathbf{c}}}_i \rangle$ denote the convergence point of a mean-shift procedure started at $e_i = \langle \underline{\mathbf{p}}_i, \underline{\mathbf{c}}_i \rangle$ in the joint domain. The mean-shift filtering replaces on each pixel the feature component $\underline{\mathbf{c}}_i$ with $\hat{\underline{\mathbf{c}}}_i$.

The second stage seeks clusters among all convergent points $\hat{e}_i = \langle \hat{\underline{\mathbf{p}}}_i, \hat{\underline{\mathbf{c}}}_i \rangle$ that are closer than the bandwidth h_p in the spatial domain and h_c in the color space. Each pixel e_i is assigned to the cluster detected for its corresponding convergent point \hat{e}_i . The image partitioning is finished grouping all connected pixels with the same cluster label into one region.

The most important parameters of this algorithm are the spatial and color bandwidths, that have to be adapted to each application. Large bandwidths force an under-segmentation and small bandwidths an over-segmentation. Comaniciu and Meer suggest that the ideal values should be a function of the location in the feature space and their computation should be controlled by high-level knowledge.

The problem of the mean-shift concept is its computational cost: applying the mean-shift procedure to each single pixel is a relatively expensive task when it is compared to the approach described in the next section. For this reason, Meer et al. propose to

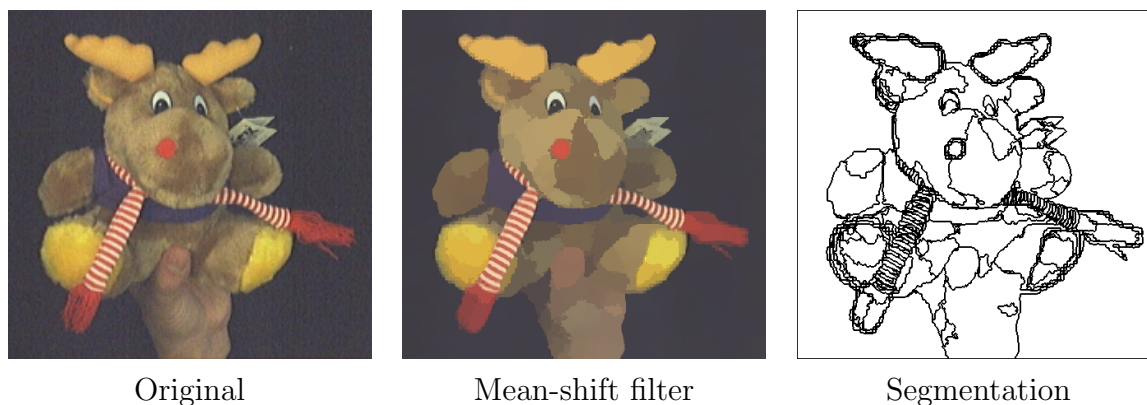


Figura 7.7: Ejemplo de segmentación por desplazamiento de medias. Con anchos de banda $h_p = 15$ píxeles y $h_c = 5$ unidades de color el algoritmo toma 240 s en un computador con un procesador Intel Pentium 4 (2.8 GHz).

replace parts of the exact algorithm with approximations that increase the speed at cost of a fair decrease in the segmentation quality. For example, the kernel is approximated with a hyper-box instead of a more adequate hyper-sphere. Further details on these optimizations can be found in [39]. Figure 7.7 shows a segmentation example for an image of size 442×424 , computed with the exact algorithm.

7.3.2 Métodos basados en el dominio de la imagen

Another way to cope with the image-based segmentation problem is to compare the feature values of each pixel in the image-domain, i. e. pixels are compared within pre-defined spatial neighborhoods. Two major groups of algorithms can be identified: the first one defines regions through the feature similarity between their elements (area-based approaches). The second one identifies feature discontinuities as boundaries between homogeneous regions (edge-based approaches). Many modern segmentation strategies try to satisfy both concepts simultaneously [50].

Métodos basados en área

Traditional area-based techniques utilize one of two principles: *region growing* or *split-and-merge*. Region growing methods assume the existence of some *seed*-points, to which adjacent pixels will be added if they fulfill a homogeneity criterion [2, 37, 28, 6, 76, 73]. The main advantage of these methods is the creation of spatially connected and compact regions, which contrast with the usually noisy image partition obtained with pure feature-based segmentation approaches. They are frequently applied to separate one single homogeneous object (e. g. background, Figure 7.8) from the rest of the image, but using several seeds positioned at different objects it is also possible to perform more sophisticated segmentations [6]. The required seed selection is a subtask of this approach, which can be solved by taking advantage of clustering methods or morphological operations,

among others [42, 22].

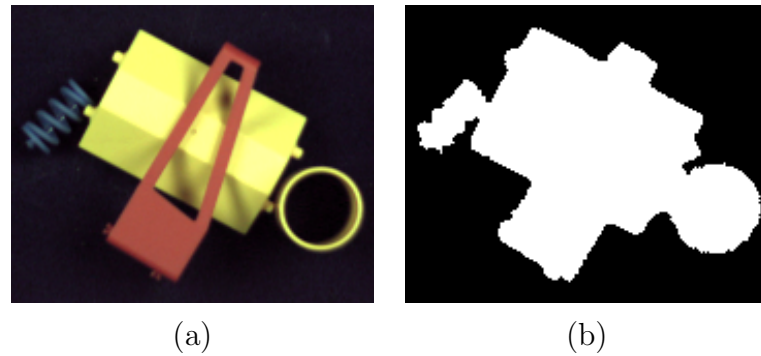


Figura 7.8: Ejemplo de segmentación por crecimiento regional. Cuatro semillas en las esquinas de la imagen original (a) se utilizaron como puntos iniciales del crecimiento, que permite detectar el fondo (b).

Split-and-merge algorithms first over-partition an image into small regions, followed by a merging stage that joins adjacent regions still fulfilling the homogeneity predicate (Figure 7.9). Quad-trees, Delauney triangulation or Voronoi diagrams belong to the tessellation techniques habitually employed to split the image [54, 27, 73, 42]. The comparison between adjacent regions can use simple statistics or can be based on more elaborated mathematical models, like Markov Random Fields (MRF), which also permit merging regions of similar texture [54].

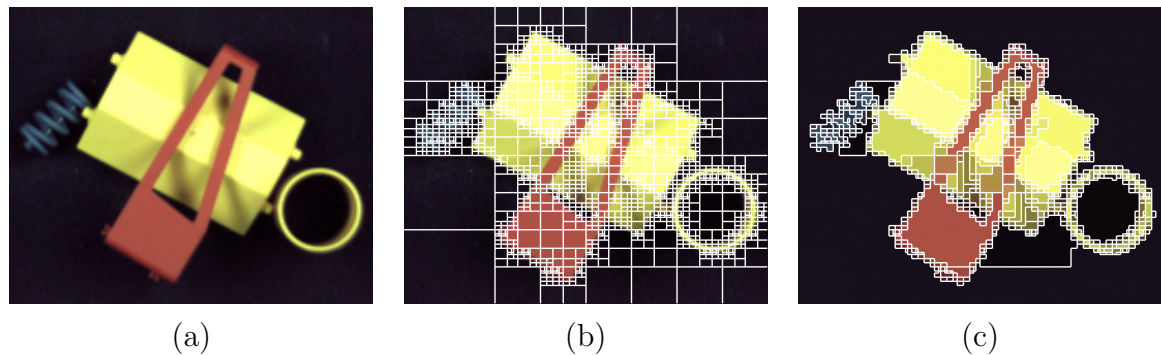


Figura 7.9: Ejemplo de segmentación por división y fusión. La imagen original (a) se parte utilizando un QuadTree (b) seguido por una etapa de fusión de regiones similares (c).

Other area-based methods employ a graph-representation of the image, where pixel similarities are coded as weights of the edges between the graph nodes (image pixels). For example Shi and Malik [68] introduce a technique called *normalized cuts* to optimally split the image into two sub-graphs that maximize the intra-subgraph similarity and simultaneously maximize the inter-subgraph dissimilarity. Yu [83] enhanced the method to optimally split an image into k regions. Barbu and Zhu [4] suggest the use of the Swendsen-Wang cuts instead. The opposite approach is followed by Haris et al. [32], who utilize a graph structure to decide which regions need to be merged, instead of how the image should be split.

Métodos basados en bordes

Edges are discontinuities in the feature characteristics (e. g. intensity) of adjacent pixels. Their detection is just the first stage of any edge-based segmentation approach. Further processing is necessary in order to provide a valid segmentation as stated as defined above. Since standard detectors like Marr and Hildreth's [46], Canny's [9] or SUSAN [72] usually leave some gaps between object boundaries, some mechanisms are required to fill them appropriately (Figure 7.10). This task can be solved combining the edge detection with

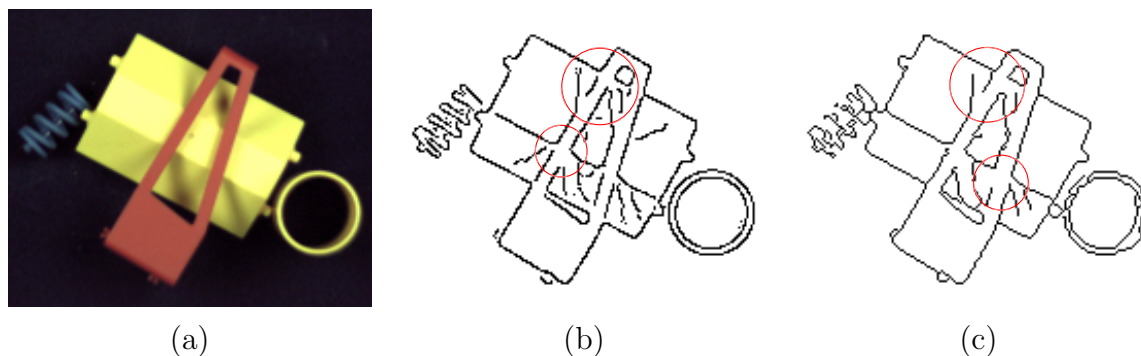


Figura 7.10: Ejemplos de detección de bordes. La imagen original (a) se analiza con (b) el algoritmo de Canny (c) el algoritmo SUSAN. Algunos espacios permanecen en la detección de bordes y se han indicado con círculos.

an area-based or a feature-space-based approach [87]. Recently a new generation of edge detectors based on the Earth Mover's Distance have been proposed [45, 64]. They show a better performance due to their capability to detect junctions and corners. However, since the computational load required to analyze a single pixel is relatively high, their throughput is very low compared with traditional techniques.

Morphological watershed segmentation [77, 71, 62] can also be categorized as an edge-based approach (Figure 7.11). They work on a topographical edgeness map, where the probability of a pixel to be an edge is modeled by its altitude. A "flooding" step begins which fills the valleys with water. The watershed lines are detected when the water of two different valleys encounters. Relatively efficient algorithms exist to compute the watershed lines and catchment basins, but they are sensitive to noise and tend to over-partition the images. The available techniques work on gray-valued images obtained usually as the gradient of the intensity.

The term *watershed* denotes in topography the region of land that drains into a particular body of water (e. g. river, lake or ocean). Hence, rain that falls anywhere in the watershed will end at its corresponding body of water. Another meaning, which is the one of interest in image processing, is the topographical dividing line between these drainage basins. Watersheds, therefore, usually run along mountain ridges.

Two efficient algorithms for the computation of watersheds deserve special attention. The *immersion simulation* of Vincent and Soille [77] is optimized for images with integer-valued pixels. It detects the catchment basins by piercing the minima of the image and

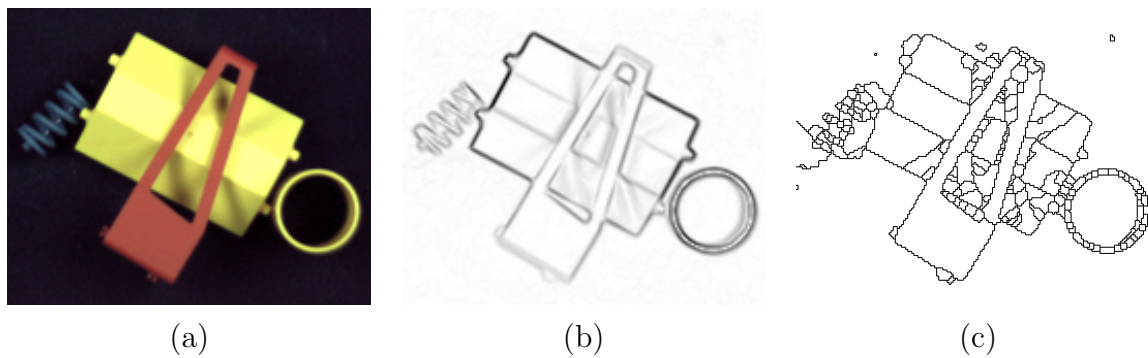


Figura 7.11: Ejemplo de segmentación por cuencas. El gradiente de la imagen original (a) se muestra en (b). Es usado como mapa topológico para encontrar cuencas y líneas divisorias (c).

then slowly immersing it into water. The watersheds are detected in those locations where water coming from different basins (or minima) meets. (Figure 7.12).

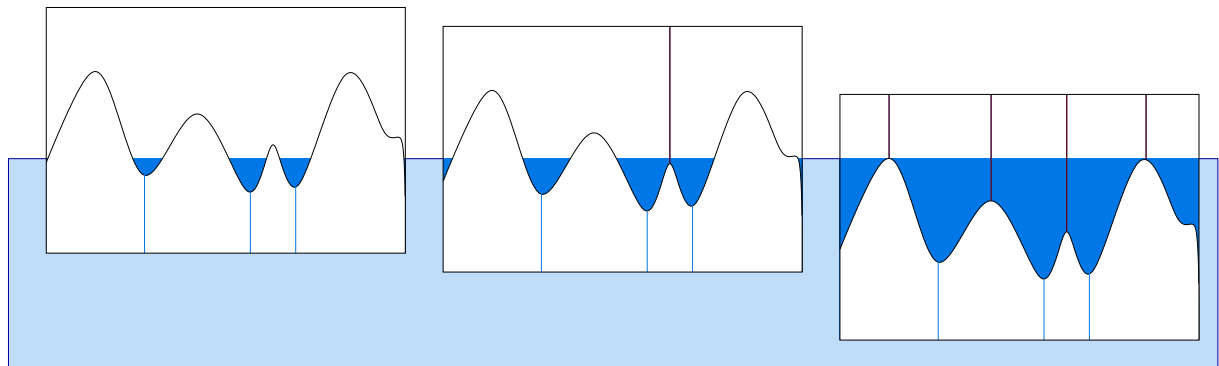


Figura 7.12: Watershed detection with immersion simulation. The image is pierced at its minima and then immersed into water. Watersheds are found at the meeting locations of water coming from different catchment basins.

The De Smet and Piret [71] *rain-falling* algorithm is more appropriate for images with a floating-point representation of their gray values. It groups pixels that lie in the path of a rain drop towards a local minimum and assigns all convergent paths to the same catchment basin (Figure 7.13). The rain-falling algorithm is faster in the detection of catchment basins than the immersion simulation, which has been optimized to identify the watershed lines. Hence, the rain-falling concept is more adequate for the current level of segmentation, as the catchment basins are direct representations of homogeneous regions.

Reducing Over-Segmentation

The major problem of watersheds is the resulting over-segmentation. Several measures have to be taken in order to find an image partition with fewer regions. The concept

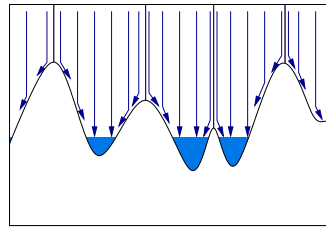


Figura 7.13: Watershed detection with a rain-falling algorithm. All pixels in the path of a rain-drop toward a basin’s minimum are assigned to that catchment basin

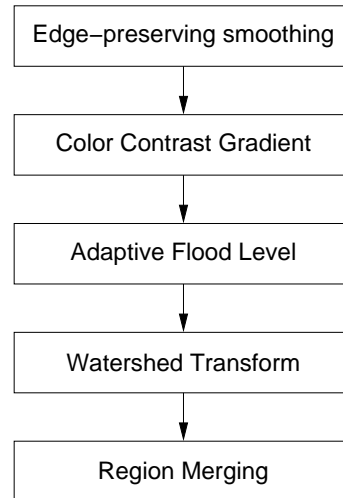


Figura 7.14: Watershed-based segmentation concept.

followed here is depicted in Figure 7.14. It is an extension of the segmentation approach for gray valued images introduced by Haris et al. [32]: the color edgeness computation replaces the scalar gradient, and an adaptive computation of the initial flood level has been added.

The first stage for edge-preserving smoothing suppresses the effects of noise and small irrelevant image structures. It relies on operators like the median filter, the mean-shift filter discussed in the previous section, or any other form of image diffusion. Here, the former has been chosen, since it is considerably faster than the other methods and still generates acceptable results.

The second method to control the degree of over-segmentation makes use of a pre-defined flood level l , that drowns catchment basins formed by small edgeness hills (Figure 7.15). However, finding appropriate values for l is a difficult task, since the optimal value range strongly depends on the analyzed image. Figure 7.16 shows the sensitiveness of the region number against small variations of the initial flood level. Setting l to zero and without pre-smoothing, more than 22000 regions are detected in each image (an average of just 9 pixels per region), making the results of a “pure” watershed transform impractical. A median filter of size 5×5 helps to reduce this number of regions by one order of magnitude. The flood level used to generate the middle column seems to be more appropriate to segment

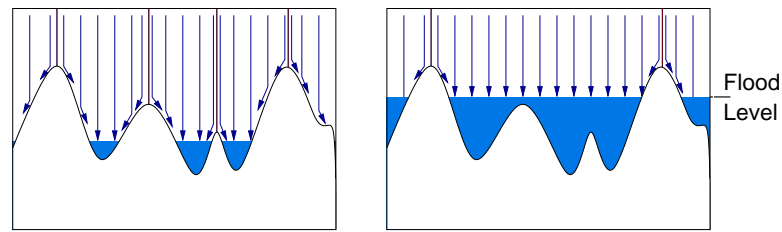


Figura 7.15: Elevating the initial flood level l reduces the number of detected regions. With a low level (left) five regions are detected. Less conspicuous edges are removed increasing l (right). In the latter case only three regions are detected.

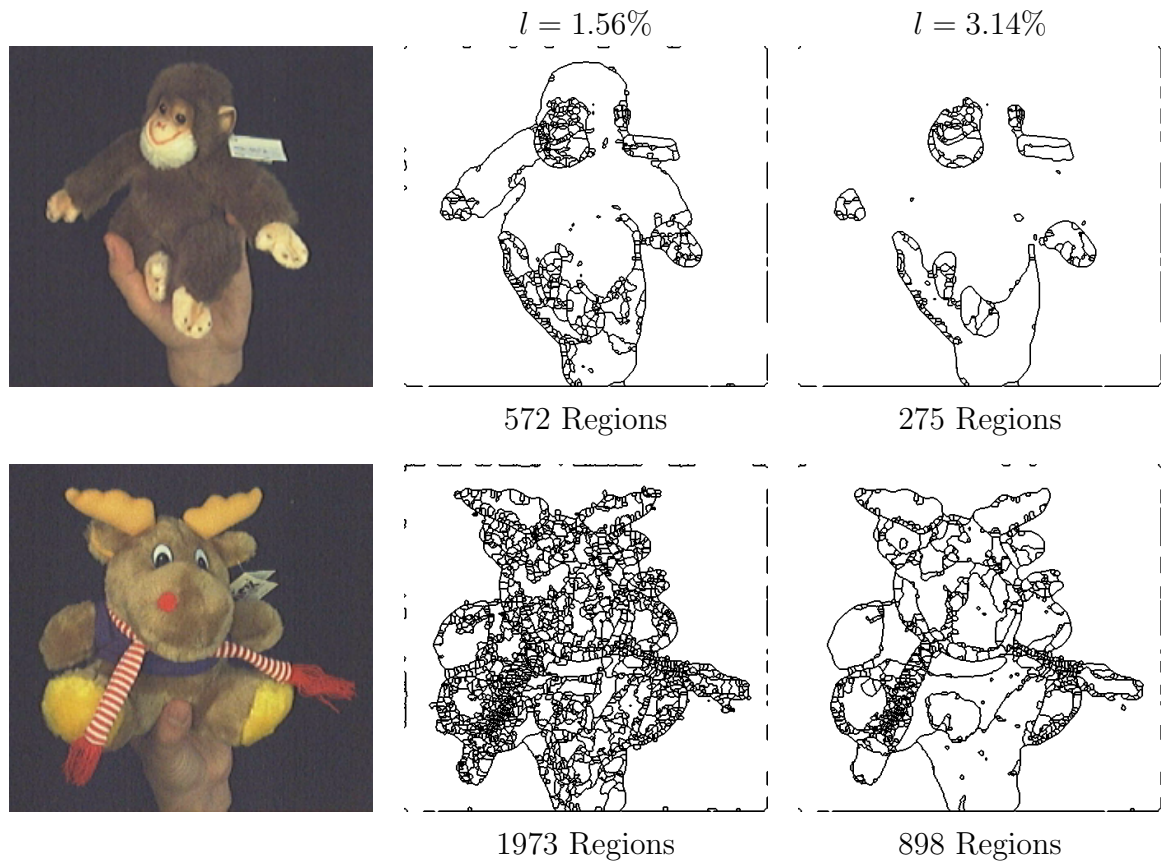


Figura 7.16: Reducing the over-segmentation of watersheds. A median-filter of size 5×5 smoothes each color channel. The initial flood levels are set to 1.6% and 3.1% of the maximum value of the gradient magnitude.

the darker object, but for the brighter one this setting doubles the cardinality of a better partition obtained with $l = 3.14\%$. On the other hand, selecting the higher flood level forces an under-segmentation of the first object.

Hence, an adaptive selection of the flood level l can be performed, based on the observation that only a small percentage of all pixels in an image can be part of the edges between semantically meaningful regions. Let $\|\nabla \mathbf{f}_I(\mathbf{p})\|$ denote the edgeness at the image position \mathbf{p} , and let $(1 - \delta)$ be the fraction of the image expected to contain edge candidates. The

level l is selected such that

$$\left| \left\{ \underline{\mathbf{p}}_j \mid \|\nabla \mathbf{f}_{\mathcal{I}}(\underline{\mathbf{p}}_j)\| \geq l \right\} \right| = (1 - \delta)|\mathcal{I}|. \quad (7.12)$$

This choice of δ maintains the semantic of the flood level l : a value of zero means to leave the result of the watersheds untouched, while $\delta = 1$ floods the complete topological map, generating a segmentation with only one region. The parameter $\delta = 67.5\%$ reproduces the best partitions in Figure 7.16 automatically.

Active contours

Another family of edge-based algorithms are the active contours (also known as snakes) [38, 63, 15, 31, 81, 12, 35]. They are frequently used in medical applications, where, due to the imaging techniques employed, noise is not negligible. A parametrical curve model for the boundary of an object is assumed to be known, leaving the algorithms with the task of determining its parameters. This is accomplished by minimizing a functional, which depends on both an external image energy that tries to attract the active contour and an internal contour energy that hinders contour deformation. The convergence of the curves to one specific object depends on the initialization and makes them suitable for interactive segmentation applications. Active contours are not suitable for object retrieval applications, as they have been conceived to detect single objects projected into one connected region. Thus, the detection of several (maybe overlapping) objects in complex scenes is, with them, a difficult task.

7.3.3 Métodos híbridos

All previous methods have intrinsic disadvantages that can be partially compensated by combining different techniques. For instance, clustering methods detect homogeneous regions in the feature space. However, since spatial relationships are ignored, the region boundaries in the image-domain are highly irregular. The incorporation of image-domain concepts can provide a solution to this problem [26]. Further examples for this family of hybrid strategies include the combination of active contours, region growing and a Bayesian probabilistic framework [86], the use of both edge detection and region growing stages [65, 69, 87], a combination of watersheds and region growing [32], and the interaction of an adaptive clustering algorithm with a Markov Random Field which provides some spatial constraints [55, 11]. A detailed review of techniques to combine area-based and edge-based approaches can be found in [50].

Besides these rather classical edge and area-based techniques, another group of algorithms exists based on a more theoretical statement. They try to find edges and homogeneous regions simultaneously. Two groups can be distinguished: regularization methods and consistent labeling methods [49].

Regularization means to restore the well-posedness of an ill-posed problem by constraining

the number of possible solutions. Let the image \mathcal{I} be the result of a two-dimensional projection operation F of a scene S , i. e. $\mathcal{I} = F(S)$. The general vision task is the inverse function F^{-1} that infers which scene S can be the causal stimulus for a given image \mathcal{I} : $S = F^{-1}(\mathcal{I})$. The regularization theory restricts the space of acceptable solutions by choosing the scene S that minimizes an appropriate functional $G(S)$. The constraints are given through the functional $G(S)$ and a norm $\|\cdot\|$. The solution can be found using one of following three methods [49]:

- find S such that the constraint $\|G(S)\| < C$ is satisfied and S best approximates the data, i. e. $\min \|F(S) - \mathcal{I}\|$, or
- find S such that it is close enough to the data and is the most “regular”, i. e. $\|F(S) - \mathcal{I}\| \leq C$, $\min \|G(S)\|$, or
- find S that minimizes a weighted sum of the closeness to the data and the degree of regularization, i. e. $\min(\|F(S) - \mathcal{I}\|^2 + \lambda\|G(S)\|^2)$.

A stochastic formulation of the regularization problem uses the Bayesian rule, and states the problem as the selection of the most likely model of the scene S given the image \mathcal{I} :

$$P(S|\mathcal{I}) = \frac{P(\mathcal{I}|S)P(S)}{P(\mathcal{I})}$$

The likelihood $P(\mathcal{I}|S)$ corresponds to the term $\|F(S) - \mathcal{I}\|$ and the a-priori probability $P(S)$ corresponds to the regularization term $\|G(S)\|$.

The same framework is used for segmentation replacing the scene S by a segmentation model \mathcal{S} , i. e. a regularization-based segmentation method tries to find the most probable segmentation for a given image. Searching for the optimal model requires relatively expensive algorithms, like simulated annealing [25] or expectation maximization [8]. Even if these algorithms find an optimal or almost optimal solution of the functional minimization task, this fact does not ensure that the result is better suited for an application than a partition obtained with more simple and faster approaches. Regularization methods can incorporate some contextual knowledge in the specification of the functional $G(S)$ to improve their results. However, the possible degrees of freedom are relatively limited in order to keep the algorithmic complexity tractable.

Consistent labeling algorithms try to assign a discrete or continuous set of labels to a set of regions under consideration of a compatibility criterion. Markov Random Fields are frequently used for a stochastic labeling approach [25, 54, 18] together with a maximum a-posteriori (MAP) optimization approach, similar to the previously mentioned stochastic regularization formulation. Other optimization methods besides the maximum a-posteriori estimation are possible. For example, maximization of the posterior marginals [47] or sequential maximum a-posteriori [8] are suggested to cope with different unfavorable issues of the MAP estimation. This kind of algorithms are employed in the segmentation of textured images [54], due to their ability to stochastically model the content of non-homogeneous regions. For the current application context these methods are rather unsuitable as a consequence of their high computational costs.

Adaptive Clustering Algorithm

Pappas [55] proposed for the segmentation task the Adaptive Clustering Algorithm (ACA), which also uses the k -means clustering as a first step in a framework that combines feature-space with image-domain information. The consideration of the latter is made, however, within a consistent labeling framework. His algorithm, developed originally for gray-valued images but easily extendable to color images (e. g. [11]), formulates segmentation as a maximum a posteriori (MAP) optimization problem. The image is assumed to be formed by a set of uniform or slowly varying regions, such that sharp transitions can only occur at the boundaries between them.

The probability that the segmentation \mathcal{S} is valid for a given image \mathcal{I} is

$$p(\mathcal{S} | \mathcal{I}) \propto p(\mathcal{I} | \mathcal{S})p(\mathcal{S}) . \quad (7.13)$$

where $p(\mathcal{I} | \mathcal{S})$ denotes the conditional density of the observed image given the distribution of regions, and $p(\mathcal{S})$ is the a-priori density of the region process. The former is modeled for each pixel as a white Gaussian process

$$p(\mathcal{I} | \mathcal{S}) \propto \exp \left(- \sum_{\langle \mathbf{p}_k, \mathbf{c}_k \rangle \in \mathcal{I}} \frac{(\mathbf{c}_k - \underline{\boldsymbol{\mu}}_{\mathcal{R}_i}(\mathbf{p}_k))^T (\mathbf{c}_k - \underline{\boldsymbol{\mu}}_{\mathcal{R}_i}(\mathbf{p}_k))}{2\sigma^2} \right) \quad (7.14)$$

where $\underline{\boldsymbol{\mu}}_{\mathcal{R}_i}(\mathbf{p}_k)$ denotes the expected value for the pixel at the position \mathbf{p}_k within the image region \mathcal{R}_i . This term constrains the expected colors in a region to the data.

The a-priori density is modeled with a Markov Random Field (MRF), i. e. the conditional probability of an image element $e_k = \langle \mathbf{p}_k, \mathbf{c}_k \rangle$ only depends on its direct neighbors: $p(e_k | \{e_j | k \neq j\}) = p(e_k | \{e_j | N_z(e_k, e_j)\})$, with the neighborhood predicates defined previously. The Hammersley-Clifford theorem [5] states that if the density $p(\mathcal{S})$ is a MRF then it is described by a Gibbs density with the clique potentials $V_C(\mathcal{S})$:

$$p(\mathcal{S}) = \frac{1}{Z} \exp \left(- \sum_C V_C(\mathcal{S}) \right) \quad (7.15)$$

with a normalization constant Z , and the sum over all cliques C . A clique is a set of neighbor points, and in this special case, any pair of two adjacent pixels (second order cliques). The clique potentials $V_C(\mathcal{S})$ are chosen such that there is a higher probability for two neighbor pixels to belong to the same region than to different regions. Thus, the region density $p(\mathcal{S})$ imposes spatial continuity:

$$V_C(\mathcal{S}) = \begin{cases} -\beta & \text{if } C = \langle e_k, e_j \rangle, e_k, e_j \in \mathcal{R}_i \in \mathcal{S}, N_z(e_k, e_j) \\ +\beta & \text{if } C = \langle e_k, e_j \rangle, e_k \in \mathcal{R}_a \in \mathcal{S}, e_j \in \mathcal{R}_b \in \mathcal{S}, a \neq b, N_z(e_k, e_j) \end{cases} \quad (7.16)$$

The combined probability for the segmentation \mathcal{S} is given by

$$p(\mathcal{S} | \mathcal{I}) \propto \exp \left(- \sum_{\langle \mathbf{p}_k, \mathbf{c}_k \rangle \in \mathcal{I}} \frac{(\mathbf{c}_k - \underline{\boldsymbol{\mu}}_{\mathcal{R}_i}(\mathbf{p}_k))^T (\mathbf{c}_k - \underline{\boldsymbol{\mu}}_{\mathcal{R}_i}(\mathbf{p}_k))}{2\sigma^2} - \sum_C V_C(\mathcal{S}) \right) \quad (7.17)$$

If the mean values for the regions $\underline{\mu}_{\mathcal{R}_i}(\underline{\mathbf{p}}_k)$ are assumed constant and $\beta = 0$ then (7.17) is equivalent to the k -means algorithm. Hence, this probabilistic framework can be regarded as a generalization of k -means clustering.

ACA uses the centroids detected by the k -means algorithm as initialization for $\underline{\mu}_{\mathcal{R}_i}(\underline{\mathbf{p}}_k)$. The next step is to update these mean values using a window and the current labeled mask (Figure 7.17): for each pixel position $\underline{\mathbf{p}}_k$ and a new hypothesized label i , the new

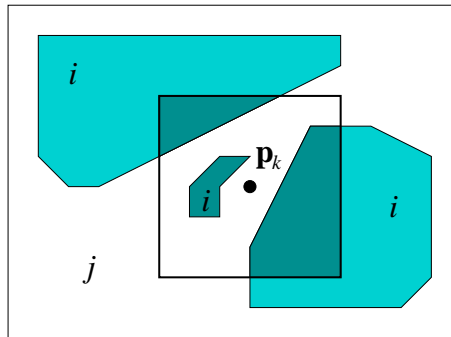


Figura 7.17: ACA mean value update.

mean value for the pixel is estimated from all image elements with label i that lie within the window centered at $\underline{\mathbf{p}}_k$. The probability (7.17) can be computed for all labels, and the one with the highest probability is assigned to the pixel. Several iterations for mean estimation and relabeling are repeated, until the number of label changes lie below a given threshold. The window size is then reduced and the whole process is repeated until a minimum window size is reached. At the end of the process, the mean value within a region changes slowly, allowing one label to represent very different colors at different locations in the image. If the smallest window size is kept large enough, the resulting means-image can also be considered as an edge-preserving smoothing filter. An example for the computed mean values and the obtained segmentation is depicted in Figure 7.18. The computation took 38 seconds on an Intel Pentium 4 (2.8 GHz) PC and it found 764 regions. Increasing β from 0.75 to 5.75 reduced the number of regions to 436, but the algorithm required almost six minutes to converge.

The unpredictable time requirements of this approach make its evaluation in a Pareto front optimization very expensive. Furthermore, five minutes is definitively too much time for a user to wait for a recognition result. For these reasons, the ACA segmentation will be no further considered for the image-based segmentation level.

7.3.4 Anotaciones sobre la tarea de segmentación

Skarbek and Koschan [70] conclude their survey with several important remarks. The most relevant for the current context follow:

1. Color images allow more reliable image segmentation than gray scale images.

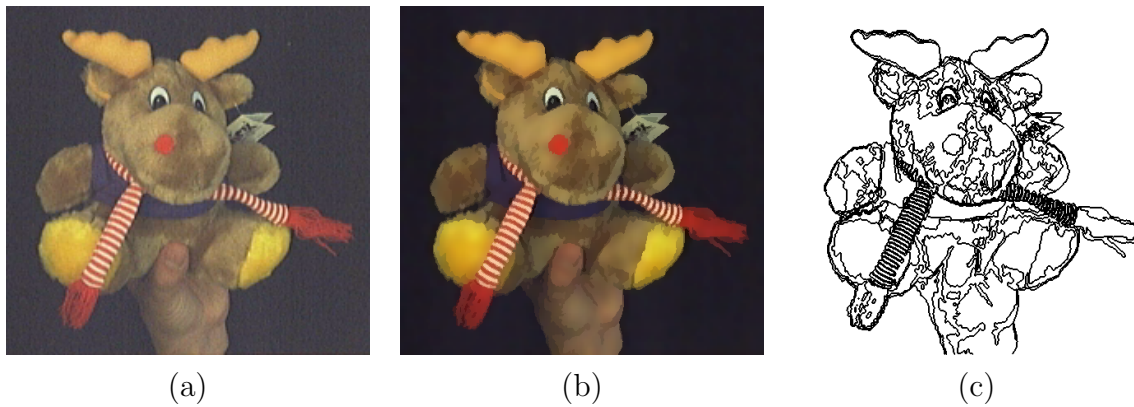


Figura 7.18: Example for ACA Segmentation. (a) Original, (b) computed means, (c) boundaries between regions. 764 regions were detected

2. General purpose algorithms are not robust and usually not algorithmically efficient.
3. As a rule, authors ignore comparing their novel ideas with existing ones.
4. As a rule, authors do not estimate the algorithmic complexity of their methods.
5. Most clearly specified, algorithmically efficient, and robust are methods designed for the particular small applications assuming well specified knowledge about the scene.
6. It seems that separating processes for region segmentation and for object recognition is the reason of failure of general purpose segmentation algorithms. This is in agreement with the previously mentioned thesis of D. Forsyth [23].
7. All techniques are dependent on parameters, constants and thresholds which are usually fixed on the basis of few experiments. Tuning and adapting of parameters is rarely performed.

Many of these conclusions are interrelated. General purpose algorithms completely ignore contextual information, and strongly depend on the choice of a proper parameter set. Small modifications in these parameters can substantially change the results. Algorithms developed for specific applications have a better behavior, since the additional knowledge can be used to force the desired results focusing on relevant information only. The limited number of scene configurations can also be exploited to design faster algorithms. It is pertinent to remark that the contextual knowledge is often only used in the selection of an algorithm or its parameters [11]. This *implicit* embedment of knowledge in the algorithms can not replace the *explicit* use of information gained from the scene itself during the recognition process.

Most authors only compare their work with strongly related algorithms, ignoring the capabilities of other segmentation classes. It is frequently unclear, if the results presented for an algorithm *A* could also be provided with another parameter choice of algorithm *B*. The lack of direct comparison possibilities, like the availability of source code or standard testbeds, makes the problem even worse. It seems there is no short term solution to these problems.

Not only the theoretical complexity of the algorithms is neglected, but also their efficiency: how large are the constants in the complexity analysis really? For the current retrieval application this issue is of great importance, since the segmentation task usually determines how long a user has to wait for a result and how long it takes to train the retrieval system. Some authors mention as advantage of their algorithms the efficient implementation for parallel computers, however, since common computer architectures still use just a few processors, such concepts would be too slow for real applications. Others stress the advantages of non-iterative algorithms neglecting the fact that iterative ones are not necessarily slower. Table 7.1 summarizes advantages and disadvantages of the discussed image-based segmentation categories.

7.4 Evaluation of Segmentation Algorithms

No segmentation algorithm is generally applicable to all images and different algorithms are not equally suitable for a particular application [53]. It is therefore necessary to have objective mechanisms to evaluate a promising set of techniques in the context of interest. Zhang [85] proposed the categorization for evaluation methods depicted in Figure 7.19.

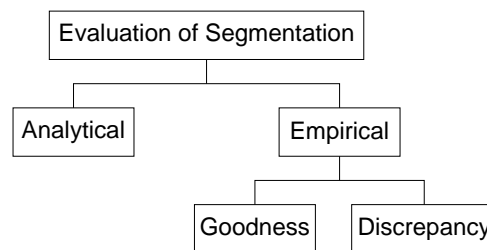


Figura 7.19: Evaluation approaches for segmentation algorithms according to Zhang [85].

Analytical methods deal with the algorithms themselves considering underlying principles, assumptions, constraints, complexity, etc. These techniques do not require the implementation of the algorithms and are exempted from bias effects caused by experimental settings. However, they cannot describe many important performance issues.

Empirical methods work with the segmentation output, requiring explicitly the implemented algorithms. Goodness methods define quantitative measures that judge the concordance of the segmentation results with an idealized concept, without the necessity of ground-truth data. They focus on the evaluation of homogeneity predicates and provide measures like intra-region uniformity or inter-region dissimilarity. The last category, discrepancy evaluation, compares the output with ground-truth information (also called *golden data*), and the differences are coded in some metric.

A major problem in the evaluation of segmentation techniques is the diversity of results of one algorithm depending on the chosen parameterization. This aspect is traditionally neglected in the literature of image segmentation: when a new approach is proposed, its advantages over other techniques are usually shown under consideration of only one

Tabla 7.1: Comparación de técnicas de segmentación a nivel de imágenes

Espacio de Características	<ul style="list-style-type: none"> + Detección de homogeneidad en un contexto global. - Relación espacial entre píxeles se ignora. 		
	Aglomeración	<ul style="list-style-type: none"> + Consideración simultánea de todas las dimensiones del espacio de características. + Aglomeradores basados en estimación de densidad encuentran conglomerados de forma arbitraria. + Existen algoritmos relativamente eficientes. - Tamaño o número de los conglomerados debe conocerse a-priori. - Estimadores de densidad requieren un ancho de banda o un algoritmo para estimarlo de los datos. 	
	Histogramas	<ul style="list-style-type: none"> + Métodos basados en histogramas unidimensionales son computacionalmente eficientes. - Sensibles al ruido. - Métodos 1D ignoran correlaciones entre diferentes dimensiones del espacio de características. - Modelos usados en detección de picos (e. g. Gaussianos) usualmente no logran ajustarse correctamente a las distribuciones reales. 	
Dominio de Imagen	Basados en Área	Reg. Crecim.	<ul style="list-style-type: none"> + Creación de regiones compactas conexas. + Algoritmos rápidos disponibles. + Aplicables en segmentación figura/fondo de objetos convexos compactos. - Selección de semillas puede ser un problema complejo. - Inapropiada para la segmentación de múltiples componentes.
		Div.&Fus.	<ul style="list-style-type: none"> + Algoritmos rápidos disponibles. + Apropriados para segmentación de texturas. - Estrategias de partición producen efectos severos de cuantificación.
		Grafos	<ul style="list-style-type: none"> + Proveen mecanismos para maximizar la similitud intra-grafo y la disparidad inter-grafo. - Caros computacionalmente, en particular si se inicia con un nodo por píxel.
	Basados en Bordos	<ul style="list-style-type: none"> + Bordos entre regiones son usualmente más suaves y precisos que aquellos detectados con los métodos en el espacio de características - Detectores de bordes usualmente dejan espacios en los bordes de regiones, dificultando la detección de regiones cerradas. 	
		Cuencas	<ul style="list-style-type: none"> + Detección de contornos cerrados. + Algoritmos eficientes disponibles. - Sensibles al ruido y a parámetros utilizados. - Imagen se sobre-segmenta fácilmente.
		Contor. Act.	<ul style="list-style-type: none"> + Robustos al ruido (especialmente apropiados en aplicaciones médicas). - Aplicable a solo un objeto en un fondo homogéneo. - Inicialización automática del contorno difícil. - No apropiada para segmentación no supervisada..
Híbridos	<ul style="list-style-type: none"> + Combinación de varios métodos puede adaptarse a las necesidades de cada aplicación. + Modelos complejos permiten la segmentación de texturas. - Costo computacional elevado para los métodos de modelado estocástico. 		

parameter set which is rarely optimized. Therefore, the evaluation has to support the choice for both the algorithms and their parameterization.

Everingham et al. [21] pointed out the risk of attempting to capture the goodness of an algorithm in a single metric, since it is impossible to reflect all dependencies between parameters and properties of a segmentation result in a single scalar value. A trade-off between several fitness or cost measures (for example, over- vs. under-segmentation, execution time vs. precision, etc.) can be made only if enough information about the effects of different parameterizations is available.

For example, consider the segmentation evaluation function of Liu and Yang [40, 7]

$$F(\mathcal{S}) = \frac{1}{1000a_1a_2} \sqrt{n} \sum_{i=1}^n \frac{\sigma_{\mathcal{R}_i}^2}{\sqrt{|\mathcal{R}_i|}}$$

where \mathcal{S} is the segmentation of an $a_1 \times a_2$ image with n regions \mathcal{R}_i and a feature variance of $\sigma_{\mathcal{R}_i}^2$ per region. The first term is a normalization factor, the second term penalizes over-segmentation and the last term penalizes inhomogeneities. There is no means to tell what value of this measure is appropriate for an application, since in some cases an over-partition is preferable to ensure some higher degree of homogeneity, but at other times the opposite could be the better choice. A minimization of this function finds only a more or less arbitrary balance between over-segmentation and homogeneity. Furthermore, exactly the same metric value can be produced with different ratios of both properties. In other words, it is possible to check that $F(\mathcal{S}_1) < F(\mathcal{S}_2)$, but this does not tell anything about the suitability of a specific parameterization of an algorithm for a given application.

Evaluation through empirical discrepancy is associated with the generation of a reference data set, including images and their *ideal* segmentation. The generation of such a golden set is, depending on the application, a difficult and expensive task, since the evaluation is significant only if the reference set is representative enough for the context. For this reason, many authors opt to simply show one or two (more or less arbitrarily chosen) examples, using only one set of parameters, and let the reader assess the quality of the algorithm. Another related problem is the common expectation for semantically meaningful results using an image-based algorithm. This is in part a consequence of the unclear definition of the term segmentation: a task defined at a low-level is expected to have the ability of guessing what objects look like. As already stated, this is a hopeless venture, unless additional scene configuration restrictions and object appearance knowledge are exploited.

In the context of object retrieval, the choice of a segmentation technique and its parameters will influence the final recognition results. For this reason, the recognition rate seems to be a suitable empirical measure of the appropriateness of a segmentation technique in this context. However, it can only evaluate the whole segmentation concept, making it difficult to assess the quality of the image-based and surface-based stages separately.

Everingham et al. [21] proposed an evaluation strategy based on multi-objective performance optimization. The result of such an evaluation mechanism is a *front*, describing the

“best” configurations in a multi-dimensional fitness space. For a given trade-off between the single fitness measures, this approach permits to objectively select not only the best algorithm, but also the parameterization that produces the desired operating point.

7.4.1 Evaluation Using the Pareto Front

The aggregate fitness function F for an algorithm A with the parameterization \mathbf{u} , evaluated using as reference the ground-truth data \mathcal{G} is defined as

$$F(A_{\mathbf{u}}, \mathcal{G}) = \Phi(f_1(A_{\mathbf{u}}, \mathcal{G}), \dots, f_n(A_{\mathbf{u}}, \mathcal{G})) \quad (7.18)$$

with the individual fitness functions $f_i(A_{\mathbf{u}}, \mathcal{G})$ defined to increase monotonically with the fitness of some particular aspect of the algorithm’s behavior. The functions f_i span a multidimensional fitness space, where each point represents the performance of an algorithm parameterized with one point \mathbf{u} in a parameter space. The general form of Φ is assumed unknown, but it has to increase monotonically with increasing values of *all* fitness functions f_i . This condition ensures that a point in the fitness space can be considered *fitter* than all other points with smaller values in all dimensions. In Figure 7.20, for example, the point q_1 is fitter than the point q_4 and all other elements within the gray rectangle. In this context, the point q_1 is said to *dominate* q_4 . All non-dominated points in a set define the Pareto front of that set. In the example of Figure 7.20 this front is defined by the points q_1 , q_2 and q_3 . Choosing a parameterization that is not in the Pareto front is always a bad choice, since there is another point on the front with a better aggregate fitness.

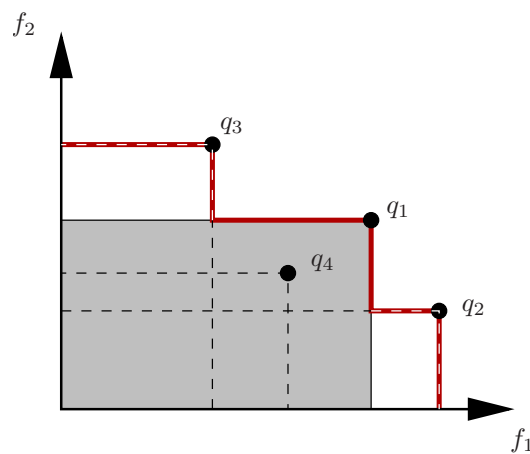


Figura 7.20: Pareto Front. The point q_1 dominates the region highlighted with a gray rectangle. Dashed lines delimit the dominated regions of the points q_2 , q_3 and q_4 . The thick solid line represents the Pareto front for the four points.

The previous concepts can be expressed mathematically using the following equation:

$$\widehat{\mathcal{P}} = \{ \langle \mathbf{u} \in \mathbb{P}_A, \mathbf{f}(A_{\mathbf{u}}, \mathcal{G}) \rangle \mid \neg \exists \mathbf{v} \in \mathbb{P}_A : \mathbf{f}(A_{\mathbf{v}}, \mathcal{G}) \succ \mathbf{f}(A_{\mathbf{u}}, \mathcal{G}) \} \quad (7.19)$$

where $\widehat{\mathcal{P}}$ is the Pareto front, $\underline{\mathbf{f}}$ is the vector of fitness functions $[f_1, \dots, f_n]^T$ and \mathbb{P}_A is the parameter space of algorithm A . The partial ordering relation “ \succ ” on $\underline{\mathbf{f}}$ describes the domination property and is defined as:

$$\underline{\mathbf{f}}(A_{\underline{\mathbf{v}}}, \mathcal{G}) \succ \underline{\mathbf{f}}(A_{\underline{\mathbf{u}}}, \mathcal{G}) \Leftrightarrow \forall i : f_i(A_{\underline{\mathbf{v}}}, \mathcal{G}) \geq f_i(A_{\underline{\mathbf{u}}}, \mathcal{G}) \wedge \exists i : f_i(A_{\underline{\mathbf{v}}}, \mathcal{G}) > f_i(A_{\underline{\mathbf{u}}}, \mathcal{G}) \quad (7.20)$$

Any algorithm that finds the Pareto front for a set of fitness points implements equations (7.19) and (7.20). Since the parameter space \mathbb{P}_A usually contains an infinite number of parameterizations, the next problem consists in choosing a representative set of samples from \mathbb{P}_A , such that their Pareto front can be assumed to be a reliable approximation of the exact front extracted for the complete space.

A naive approach would be to regularly sample the values of each parameter, since the number of necessary evaluations would increase exponentially with the number of parameters. For example, an algorithm with seven parameters, each sampled five times, would require $5^7 = 78125$ evaluations. Since a single evaluation comprises computations for a complete data set, the time requirements for this naive approach are enormous, even for such a coarse sampling of the parameter space.

Here, the proposal of Everingham et al. [21] is followed and the multi-objective evolutionary algorithm PESA (Pareto Envelope-based Selection Algorithm [16]) is used. This genetic approach suppresses the computation of useless parameterizations and concentrates the analysis on those regions of the parameter space that provide promising results. Even if this algorithm also samples the parameter space, the resolution used for each parameter is much higher (e. g. 256 or 1024 samples per parameter). The number of evaluations required is then proportional to the number of bits used to represent a parameterization.

All multi-objective optimization algorithms (including PESA) try to find the front containing parameterizations best optimized for the reference data set \mathcal{G} . Hence, it is important in the evaluation to use *representative* data taken from the application context.

7.4.2 Fitness Functions

For multi-objective evaluation algorithms several fitness functions have to be chosen. It is possible to use both goodness and discrepancy measures. Independently of the final selection, a reference data set \mathcal{G} has to be chosen. In the case of image segmentation, this set is defined as

$$\mathcal{G} = \{ \langle \underline{\mathcal{I}}_i, \underline{\mathcal{S}}_i \rangle \mid i = 1 \dots n \} \quad (7.21)$$

where $\underline{\mathcal{S}}_i$ is the ideal segmentation result for the reference image $\underline{\mathcal{I}}_i$. The set $\mathcal{G}_I = \{ \underline{\mathcal{I}}_i \mid i = 1 \dots n \}$ contains all reference images. $\mathcal{G}_S = \{ \underline{\mathcal{S}}_i \mid i = 1 \dots n \}$ is the set of all reference segmentations.

Let \mathcal{S}_i be the segmentation result of an algorithm A parameterized with $\underline{\mathbf{u}}$ for the reference image $\underline{\mathcal{I}}_i$:

$$\mathcal{S}_i = A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i). \quad (7.22)$$

The goal is to find functions that evaluate the difference between \mathcal{S}_i and $\underline{\mathcal{S}}_i$.

Seven fitness measures are used here to evaluate different aspects of the algorithms: throughput, mean normalized region size, pixel-wise potential accuracy, region-wise potential accuracy, region-wise information content, region integrity, and pixel-wise certainty. The first five have been originally proposed in [21]. The last two have been introduced in this work to cope with special properties of the different segmentation stages. For the next definitions it is always assumed $\mathcal{R}_j \in \mathcal{S}_i$ and $\underline{\mathcal{R}}_k \in \underline{\mathcal{S}}_i$. In accordance to (7.22), the expression $|A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i)|$ is used to represent the number of regions in the partition found by the algorithm $A_{\underline{\mathbf{u}}}$.

Throughput Throughput for the segmentation is defined as the number of images per second processed by an algorithm. Let $t(A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i))$ be the time required to compute the segmentation for the image $\underline{\mathcal{I}}_i$. Thus, the throughput is a goodness measure defined as

$$f_t(A_{\underline{\mathbf{u}}}, \mathcal{G}_I) = \frac{|\mathcal{G}_I|}{\sum_{\underline{\mathcal{I}}_i \in \mathcal{G}_I} t(A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i))} \quad (7.23)$$

Mean Normalized Region Size The more regions in a partition the more processing is necessary for its further analysis. Therefore, the number of regions found by a segmentation algorithm is a simple cost function related with the degree of over-segmentation. The reciprocal is used here as a fitness function. This goodness measure can also be interpreted as the mean region size as a fraction of the entire image:

$$f_r(A_{\underline{\mathbf{u}}}, \mathcal{G}_I) = \frac{1}{|\mathcal{G}_I|} \sum_{\underline{\mathcal{I}}_i \in \mathcal{G}_I} \frac{1}{|A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i)|} \quad (7.24)$$

Pixel-wise Potential Accuracy This fitness function measures to what extent it is possible to assign each region of the evaluated segmentation \mathcal{S}_i to a region of the reference segmentation $\underline{\mathcal{S}}_i$ under the assumption of a perfect classifier. Therefore, the pixel-wise potential accuracy is a good indicator for the appropriateness of an algorithm for the image-based segmentation stage: if its value is too low, the next stages of the segmentation framework, which play the role of real classifiers, will not be able to produce correct segmentations.

The pixel-wise potential accuracy f_{pa} is defined as

$$f_{pa}(A_{\underline{\mathbf{u}}}, \mathcal{G}) = \sum_{(\underline{\mathcal{I}}_i, \underline{\mathcal{S}}_i) \in \mathcal{G}} \frac{\sum_{\underline{\mathcal{R}}_j \in \underline{\mathcal{S}}_i} |\mathcal{P}(\underline{\mathcal{R}}_j, A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i))|}{\sum_{\underline{\mathcal{R}}_j \in \underline{\mathcal{S}}_i} |\underline{\mathcal{R}}_j|} \quad (7.25)$$

where $\mathcal{P}(\underline{\mathcal{R}}_j, \mathcal{S}_i)$ denotes the set of pixels in $\underline{\mathcal{R}}_j$ that also belong to the regions of \mathcal{S}_i that most overlap with $\underline{\mathcal{R}}_j$:

$$\mathcal{P}(\underline{\mathcal{R}}_j, \mathcal{S}_i) = \left\{ \underline{\mathbf{p}} \mid \langle \underline{\mathbf{p}}, \cdot \rangle \in \underline{\mathcal{R}}_j \wedge \langle \underline{\mathbf{p}}, \cdot \rangle \in \mathcal{R}_k, \mathcal{R}_k \in \mathcal{S}_i, j = \arg \max_l |\mathcal{R}_k \cap_p \underline{\mathcal{R}}_l| \right\} \quad (7.26)$$

The p -intersection of two regions $\mathcal{R}_k \cap_p \underline{\mathcal{R}}_l$ is defined as:

$$\mathcal{R}_k \cap_p \underline{\mathcal{R}}_l = \{ \underline{\mathbf{p}} \mid \langle \underline{\mathbf{p}}, \cdot \rangle \in \mathcal{R}_k \} \cap \{ \underline{\mathbf{q}} \mid \langle \underline{\mathbf{q}}, \cdot \rangle \in \underline{\mathcal{R}}_l \} \quad (7.27)$$

Region-wise Potential Accuracy If the distribution of region sizes in the reference data has a large variance, the pixel-wise potential accuracy will neglect the relevance of small regions belonging probably to small objects. The region-wise potential accuracy considers each reference region as equally important:

$$f_{ra}(A_{\underline{\mathbf{u}}}, \mathcal{G}) = \frac{\sum_{\langle \underline{\mathcal{I}}_i, \mathcal{S}_i \rangle \in \mathcal{G}} \sum_{\underline{\mathcal{R}}_j \in \mathcal{S}_i} \frac{w_{ra}(\underline{\mathcal{R}}_j)}{|\underline{\mathcal{R}}_j|} |\mathcal{P}(\underline{\mathcal{R}}_j, A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i))|}{\sum_{\langle \underline{\mathcal{I}}_i, \mathcal{S}_i \rangle \in \mathcal{G}} \sum_{\underline{\mathcal{R}}_j \in \mathcal{S}_i} w_{ra}(\underline{\mathcal{R}}_j)} \quad (7.28)$$

with the set \mathcal{P} as defined above and the weighting function

$$w_{ra}(\underline{\mathcal{R}}_j) = \begin{cases} \frac{\sum_{\mathcal{R}_k \in \mathcal{S}_i} |\mathcal{R}_k|}{|\underline{\mathcal{R}}_j|} & \text{if } |\underline{\mathcal{R}}_j| \geq r \\ 0 & \text{otherwise} \end{cases} \quad (7.29)$$

The constant r removes very small regions from the analysis and is set here to 0.05% of the image size. This discrepancy measure can be interpreted as the proportion of each reference region that can be correctly identified given the segmentation result $\mathcal{S}_i = A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i)$. Everingham et al. called an equivalent function “object-wise potential accuracy”. Their weighting function w_{ra} involves the reciprocal of a-priori probabilities for the reference regions. Here, the reciprocal of the image percentage is used instead. Since the reference segmentation not necessarily has to partition the image in regions belonging to objects, the name “region-wise potential accuracy” is here preferred.

Region-wise Information Content Assuming that the information about a reference region is uniformly distributed among its pixels, this discrepancy function measures the proportion of information of the reference region available within a single detected region. It is defined as:

$$f_{ri}(A_{\underline{\mathbf{u}}}, \mathcal{G}) = \frac{\sum_{\langle \underline{\mathcal{I}}_i, \mathcal{S}_i \rangle \in \mathcal{G}} \sum_{\underline{\mathcal{R}}_j \in A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i)} \frac{w_{ri}(\underline{\mathcal{R}}_j)}{|\mathcal{R}_{k=\arg \max_l |\mathcal{R}_j \cap_p \underline{\mathcal{R}}_l|}|} |\mathcal{Q}(\underline{\mathcal{R}}_j, \underline{\mathcal{S}}_i)|}{\sum_{\langle \underline{\mathcal{I}}_i, \mathcal{S}_i \rangle \in \mathcal{G}} \sum_{\underline{\mathcal{R}}_j \in A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i)} w_{ri}(\underline{\mathcal{R}}_j)} \quad (7.30)$$

with the set \mathcal{Q} representing the pixels of region \mathcal{R}_j that also belong to the reference region \mathcal{R}_k most overlapped by \mathcal{R}_j , i. e.

$$\mathcal{Q}(\mathcal{R}_j, \underline{\mathcal{S}}_i) = \left\{ \underline{\mathbf{p}} \mid \langle \underline{\mathbf{p}}, \cdot \rangle \in \mathcal{R}_j \wedge \langle \underline{\mathbf{p}}, \cdot \rangle \in \underline{\mathcal{R}}_k, \underline{\mathcal{R}}_k \in \underline{\mathcal{S}}_i, k = \arg \max_l |\mathcal{R}_j \cap \underline{\mathcal{R}}_l| \right\} \quad (7.31)$$

and the weighting function

$$w_{ri}(\mathcal{R}_j) = \begin{cases} 1 & \text{if } |\underline{\mathcal{R}}_k| \geq r, k = \arg \max_l |\mathcal{R}_j \cap \underline{\mathcal{R}}_l| \\ 0 & \text{otherwise} \end{cases} \quad (7.32)$$

As in the previous fitness function, r suppresses the consideration of very small reference regions.

The more regions are detected for a reference region, the smaller the fitness. This function punishes over-segmentation, but it accepts under-segmentation. Therefore, f_{ra} should not be used to evaluate partitions with a reduced number of regions: it even assigns the best fitness to segmentations with a single region.

Region integrity The region integrity function is used here to measure the degree of over- or under-segmentation of the reference regions:

$$f_i(A_{\underline{\mathbf{u}}}, \mathcal{G}) = \frac{1}{|\mathcal{G}|} \sum_{\langle \underline{\mathcal{I}}_i, \underline{\mathcal{S}}_i \rangle \in \mathcal{G}} h \left(\frac{1}{|\underline{\mathcal{S}}_i|} \sum_{\underline{\mathcal{R}}_j \in \underline{\mathcal{S}}_i} \left| \left\{ \mathcal{R}_k \mid \mathcal{R}_k \in A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i), j = \arg \max_l |\mathcal{R}_k \cap \underline{\mathcal{R}}_l| \right\} \right| \right) \quad (7.33)$$

with the weighting function $h(x) = xe^{-x}/e^{-1}$ chosen to be maximal at $x = 1$. The argument of h in (7.33) is the mean value of the number of detected regions per reference region. Under-segmentations lie below and over-segmentation above the optimal value of one. The fitness function f_i ignores, however, the exact position of the region boundaries.

Pixel-wise certainty The surface-based and object-based stages of the segmentation always produce partitions of low cardinality. Many of the previous measures have a discrete nature that cannot properly distinguish between small differences of the parameterizations. However, these later stages can also produce additional information that describe the certainty with which each pixel has been assigned to its label or class. If this information is available, it can also be employed as a goodness measure in the parameter optimization. Let $p(\underline{\mathbf{p}}_k | A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i))$ denote the certainty with which the segmentation algorithm $A_{\underline{\mathbf{u}}}$ assigns the pixel at $\underline{\mathbf{p}}_k$ to its label. The pixel certainty is then defined as

$$f_{ce}(A_{\underline{\mathbf{u}}}, \mathcal{G}_I) = \frac{\sum_{\underline{\mathcal{I}}_i \in \mathcal{G}_I} \sum_{\mathcal{R}_i \in A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i)} \sum_{\langle \underline{\mathbf{p}}_k, \cdot \rangle \in \mathcal{R}_i} p(\underline{\mathbf{p}}_k | A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i))}{\sum_{\underline{\mathcal{I}}_i \in \mathcal{G}_I} |A_{\underline{\mathbf{u}}}(\underline{\mathcal{I}}_i)|}, \quad (7.34)$$

which is the mean value of the certainty for all pixels in the reference set.

These seven fitness functions are used in the next chapters to evaluate the proposed concepts in the context of object retrieval.

7.5 Análisis de movimiento

El análisis de movimiento tiene como entrada una secuencia temporal de imágenes (en éste contexto usualmente denominadas *cuadros*), lo que conlleva una mayor cantidad de procesamiento. Tiene varias aplicaciones, desde la detección de zonas en movimiento para mejorar algoritmos de compresión de imágenes, hasta navegación o visión en robótica.

Interesan aquí dos grupos de problemas relacionados con movimiento:

1. Detección de movimiento es el problema más simple. Consiste en registrar cualquier movimiento detectado. Usualmente se utiliza una cámara estática y se usa en aplicaciones de seguridad.
2. Detección y ubicación de objetos en movimiento representa otra familia de problemas, en donde la cámara puede estar fija mientras que los objetos de la escena se mueven, o la escena puede ser fija y la cámara se mueve en ella. Dentro de estos problemas se encuentra el detectar o predecir trayectorias de objetos, así como su rastreo en la secuencia.

Las secuencias de imágenes se denotan ampliando a un argumento temporal cualquiera de las notaciones revisadas anteriormente. Por ejemplo, la notación funcional de una secuencia de imágenes bidimensionales será $f(x, y, n)$ con n el número de imagen en la secuencia.

7.5.1 Métodos de análisis de movimiento diferencial

Una forma sencilla de detectar movimiento es a través de imágenes binarias de diferencia, definidas como

$$d(x, y) = \begin{cases} 0 & \text{si } |f(x, y, n) - f(x, y, n - 1)| \leq \epsilon \\ 1 & \text{en el resto} \end{cases}$$

Los valores de uno en una imagen de diferencia se pueden deber a un píxel que pasó de objeto a fondo, de fondo a objeto, cambió de objeto, o se modificó por ruido o cambios de iluminación. Para corregir lo último se utilizan normalizaciones de intensidad y el pre-tratamiento de las imágenes de entrada para suprimir el ruido.

Otro tipo de imágenes diferenciales acumulan de forma ponderada la historia del movi-

miento:

$$d_a(x, y) = \sum_{k=1}^n a_k |f(x, y, n) - f(x, y, n - k)|$$

donde a través de los coeficientes a_k se puede dar mayor peso a cambios recientes que a cambios anteriores.

Pueden también emplearse estructuras recursivas para cada píxel sobre las imágenes de diferencia

$$d_h(x, y, n) = \begin{cases} \tau & \text{si } d(x, y) = 1 \\ \max(0, d_h(x, y, n - 1) - 1) & \text{en otro caso} \end{cases}$$

7.5.2 Flujo Óptico

La representación bidimensional de movimiento tridimensional se denomina *campo de movimiento*, en donde a cada punto se le asigna un vector de velocidad correspondiente a la dirección de movimiento, velocidad y ubicación en un lugar específico de la imagen.

El flujo óptico refleja los cambios en la imagen debidos al movimiento ocurrido en un diferencial de tiempo Δt , que en el caso de secuencias discretas usualmente corresponde a dos imágenes adyacentes en la secuencia.

En principio, se aspira a tener detectores de flujo óptico insensible a cambios de iluminación o a objetos irrelevantes (como sombras). Sin embargo, las técnicas usuales detectan movimiento, por ejemplo, si una fuente de luz se mueve sobre una esfera con superficie especular, y no detectan movimiento si la fuente de luz es constante y la esfera rota sobre su centro.

El cálculo del flujo óptico parte de dos suposiciones:

- El brillo percibido de cualquier punto de un objeto permanece aproximadamente constante en el tiempo.
- Puntos cercanos en el plano de imagen se mueven de forma similar (restricción de suavidad en la velocidad)

Para la imagen en espacio y tiempo continuos $f(x, y, t)$, se utilizan series de Taylor para aproximar el cambio de dicha función en el espacio-tiempo:

$$\begin{aligned} f(x + dx, y + dy, t + dt) &= f(x, y, t) + \frac{\partial}{\partial x} f(x, y, t) dx + \frac{\partial}{\partial y} f(x, y, t) dy + \frac{\partial}{\partial t} f(x, y, t) dt + O(\partial^2) \\ &= f(x, y, t) + f_x(x, y, t) dx + f_y(x, y, t) dy + f_t(x, y, t) dt + O(\partial^2) \end{aligned} \quad (7.35)$$

donde $O(\partial^2)$ denota los términos de orden superior y f_x , f_y y f_t simplifican la notación de las derivadas parciales. Si el diferencial de tiempo dt es suficientemente pequeño se puede asumir que el vecindario de (x, y) es trasladado una distancia pequeña (dx, dy) por lo que se puede asumir que

$$f(x + dx, y + dy, t + dt) = f(x, y, t)$$

Si los cambios dx, dy, dt son suficientemente pequeños, entonces los términos de orden superior $O(\partial^2)$ desaparecen y por tanto de (7.35) se obtiene

$$-f_t = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} \quad (7.36)$$

El objetivo es calcular el vector de velocidad

$$\underline{\mathbf{c}} = \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

con el cual (7.36) se reescribe como

$$-f_t = f_x u + f_y v = \nabla f \cdot \underline{\mathbf{c}} \quad (7.37)$$

donde ∇f representa al gradiente espacial de f .

La velocidad no se puede obtener de (7.37) directamente, por consistir en un sistema subdeterminado, por lo que se introducen restricciones de suavidad, que fuerzan a que el campo de velocidad cambie con suavidad en un vecindario determinado. En el método de Horn y Schunk para lograr esto se minimiza el error cuadrático definido como

$$E^2(x, y) = (f_x u + f_y v + f_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2) \quad (7.38)$$

donde u_x, u_y, v_x y v_y denotan las derivadas parciales de los componentes de velocidad, y λ es un multiplicador de Lagrange. La minimización se reduce a resolver el sistema de ecuaciones diferenciales:

$$\begin{aligned} (\lambda^2 + f_x^2)u + f_x f_y v &= \lambda^2 \bar{u} - f_x f_t \\ f_x f_y u + (\lambda^2 + f_y^2)v &= \lambda^2 \bar{v} - f_y f_t \end{aligned}$$

donde \bar{u} y \bar{v} son los valores medios de la velocidad en las direcciones x y y en algún vecindario de (x, y) . Se puede demostrar que una solución para el anterior sistema de ecuaciones diferenciales es

$$\begin{aligned} (\lambda^2 + f_x^2 + f_y^2)(u - \bar{u}) &= -f_x(f_x \bar{u} + f_y \bar{v} + f_t) \\ (\lambda^2 + f_x^2 + f_y^2)(v - \bar{v}) &= -f_y(f_x \bar{u} + f_y \bar{v} + f_t) \end{aligned}$$

que se resuelven utilizando un método iterativo con dos imágenes consecutivas.

El algoritmo de cálculo se resume como:

1. Inicialice los vectores de velocidad como $\underline{\mathbf{c}}(x, y) = \underline{\mathbf{0}}$ para todo x e y .
2. Sea k el número de iteración. Calcule los valores $u^{(k)}$ y $v^{(k)}$ para todos los píxeles (x, y) con

$$\begin{aligned} u^{(k)} &= \bar{u}^{(k-1)} - f_x \frac{f_x \bar{u}^{(k-1)} + f_y \bar{v}^{(k-1)} + f_t}{\lambda^2 + f_x^2 + f_y^2} \\ v^{(k)} &= \bar{v}^{(k-1)} - f_y \frac{f_x \bar{u}^{(k-1)} + f_y \bar{v}^{(k-1)} + f_t}{\lambda^2 + f_x^2 + f_y^2} \end{aligned}$$

donde f_x, f_y y f_t se calculan del par de imágenes consecutivas.

3. Detenga el proceso si

$$\sum_x \sum_y E^2(x, y) < \epsilon$$

donde ϵ es el mayor error permitido. Si el error es aun alto, repita a partir del paso 2.

7.5.3 Rastreo por desplazamiento de medias

El flujo óptico va más allá de la detección de movimiento, pero requiere procesamiento posterior para poder asignarlo a objetos. Una técnica de rastreo que sí puede utilizarse para rastrear objetos en imágenes es el rastreo por desplazamiento de medias (*mean-shift tracking*) [14]. El principio de funcionamiento es similar al utilizado en la segmentación por desplazamiento de medias, revisado en la sección 7.3.1.

La entrada para este algoritmo será una secuencia de imágenes, donde se recibe una imagen a la vez. Al inicio, se indica la región en la imagen que debe ser rastreada en la secuencia, y con cada nueva imagen se calcula la posición más probable de dicha región.

El método funciona mejor con imágenes a color. Para funcionar, el algoritmo requiere una distribución de probabilidad de los colores a rastrear, lo cual se puede generar a partir de una región de interés especificada para la primera imagen de la secuencia, o se puede obtener de algún entrenamiento anterior. Este modelo estadístico se expresa a través de

$$\hat{q}_u = C \sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2) \delta(b(\mathbf{x}_i^*) - u)$$

que es la probabilidad del color con índice u en el histograma. Aquí $\{\mathbf{x}_i^*\}_{i=1\dots n}$ es el conjunto de posiciones de los píxeles del modelo de referencia, que se asume centrado en $\mathbf{0}$; $b: \mathbb{R}^2 \rightarrow \{1 \dots m\}$ es una función que asocia a cada píxel con posición \mathbf{x}_i^* con el índice de la celda del histograma que corresponde al color de dicho píxel; k es el perfil del kernel, y C es una constante de normalización que asegura que $\sum_{u=1}^m \hat{q}_u = 1$, por lo que

$$C = \frac{1}{\sum_{i=1}^n k(\|\mathbf{x}_i^*\|^2)}$$

Sea $\hat{\mathbf{y}}_0$ la posición estimada del modelo en el cuadro anterior.

El primer paso consiste en inicializar la posición del objetivo en el nuevo cuadro con $\hat{\mathbf{y}}_0$, y se calcula la distribución de probabilidad del color u en el objetivo candidato con

$$\hat{p}_u(\mathbf{y}) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right) \delta[b(\mathbf{x}_i) - u]$$

donde C_h es la constante de normalización

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathbf{y} - \mathbf{x}_i}{h}\right\|^2\right)}$$

y $\{\underline{\mathbf{x}}_i\}_{i=1\dots n_h}$ son las posiciones de píxeles de dicho candidato, centrado en $\underline{\mathbf{y}}$. Obsérvese el uso del mismo kernel k pero con un radio h que especifica la escala del objetivo candidato.

Con la distribución del modelo a rastrear y la distribución del modelo candidato, se puede calcular el coeficiente de Bhattacharyya $\rho[\hat{\underline{\mathbf{p}}}(\hat{\underline{\mathbf{y}}}_0), \hat{\underline{\mathbf{q}}}]$ como indicador de la diferencia entre ambas distribuciones:

$$\rho[\hat{\underline{\mathbf{p}}}(\hat{\underline{\mathbf{y}}}_0), \hat{\underline{\mathbf{q}}}] = \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{\underline{\mathbf{y}}}_0) \hat{q}_u}$$

De una aproximación de Taylor se obtiene que

$$\rho[\hat{\underline{\mathbf{p}}}(\hat{\underline{\mathbf{y}}}), \hat{\underline{\mathbf{q}}}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(\hat{\underline{\mathbf{y}}}_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left(\left\| \frac{\underline{\mathbf{y}} - \underline{\mathbf{x}}_i}{h} \right\|^2 \right)$$

con los coeficientes

$$w_i = \sum_{u=1}^m \delta[b(\underline{\mathbf{x}}_i) - u] \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\hat{\underline{\mathbf{y}}}_0)}}$$

Para el cálculo de la nueva posición es necesario calcular todos los coeficientes $\{w_i\}_{i=1\dots n_h}$, pues

$$\hat{\underline{\mathbf{y}}}_1 = \frac{\sum_{i=1}^{n_h} \underline{\mathbf{x}}_i w_i g \left(\left\| \frac{\hat{\underline{\mathbf{y}}}_0 - \underline{\mathbf{x}}_i}{h} \right\|^2 \right)}{\sum_{i=1}^{n_h} w_i g \left(\left\| \frac{\hat{\underline{\mathbf{y}}}_0 - \underline{\mathbf{x}}_i}{h} \right\|^2 \right)}$$

El siguiente paso requiere recalculer las probabilidades para el candidato $\{\hat{p}_u(\hat{\underline{\mathbf{y}}}_1)\}_{u=1\dots m}$ y evaluar

$$\rho[\hat{\underline{\mathbf{p}}}(\hat{\underline{\mathbf{y}}}_1), \hat{\underline{\mathbf{q}}}] = \sqrt{\hat{p}_u(\hat{\underline{\mathbf{y}}}_1) \hat{q}_u}$$

Puesto que ésta operación no necesariamente implica que el coeficiente de Bhattacharyya se incrementará, es necesario validar la nueva posición con un ciclo:

Mientras $\rho[\hat{\underline{\mathbf{p}}}(\hat{\underline{\mathbf{y}}}_1), \hat{\underline{\mathbf{q}}}] < \rho[\hat{\underline{\mathbf{p}}}(\hat{\underline{\mathbf{y}}}_0), \hat{\underline{\mathbf{q}}}]$

$$\hat{\underline{\mathbf{y}}}_1 \rightarrow \frac{1}{2}(\hat{\underline{\mathbf{y}}}_0 + \hat{\underline{\mathbf{y}}}_1)$$

Si $\|\hat{\underline{\mathbf{y}}}_1 - \hat{\underline{\mathbf{y}}}_0\| < \epsilon$ el algoritmo termina. De otro modo asigna $\hat{\underline{\mathbf{y}}}_0 \rightarrow \hat{\underline{\mathbf{y}}}_1$ y repite desde el inicio.

Bibliografía

- [1] P. Alvarado and P. Doerfler. LTI-Lib [online]. 1998 [visitado el 5 de junio de 2010]. URL <http://ltilib.sourceforge.org>. c
- [2] M. Amadasun and R. A. King. Low-level segmentation of multispectral images via agglomerative clustering of uniform neighbourhoods. *Pattern Recognition*, 21(3):261–268, 1988. 120
- [3] S. Ando. Consistent gradient operators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):252–265, March 2000. 97
- [4] A. Barbu and S. C. Zhu. Graph partition by Swendsen-Wang Cuts. In *Proc. International Conference on Computer Vision*, 2003. 121
- [5] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B*, 26(2):192–236, 1974. 128
- [6] T. Bomberg and S. Posch. Regionensegmentierung von Farbbildfolgen. In E. Paulus and F. M. Wahl, editors, *Mustererkennung 97, 19. DAGM-Symposium*, Informatik-Fachberichte, Informatik Aktuell, pages 63–70, Braunschweig, 1997. Springer. 120
- [7] M. Borsotti, P. Campadelli, and R. Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19:741–747, 1998. 133
- [8] C. Bouman and M. Shapiro. A Multiscale Random Field Model for Bayesian image segmentation. *IEEE Transactions on Image Processing*, 3(2):162–177, March 1994. URL citeseer.nj.nec.com/bouman96multiscale.html. 127
- [9] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, August 1986. 108, 122
- [10] M. Celenk and M. Uijt de Haag. Optimal thresholding for color images. In *Proc. of the SPIE – The Int’l Soc. for Optical Eng., Nonlinear Image Processing IX*, pages 250–259, San Jose, CA, January 1998. 116
- [11] J. Chen. *Perceptually-based texture and color features for image segmentation and retrieval*. PhD thesis, Northwestern University, Illinois, December 2003. 126, 128, 130

- [12] C. Chesnaud, P. Réfrégier, and Vlady Boulet. Statistical region snake-based segmentation adapted to different physical noise models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1145–1157, November 1999. 126
- [13] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002. 117, 118
- [14] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition CVPR 2000 Cat NoPR00662*, number 8 in 2, pages 142–149. IEEE Comput. Soc, 2000. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=854761>. 142
- [15] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models — their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995. 126
- [16] D. W. Corne, J. D. Knowles, and M. J. Oates. The Pareto envelope-based selection algorithm for multiobjective optimization. In *Proceedings of International Conference on Parallel Problem Solving from Nature*, pages 839–848, 2000. 135
- [17] A. Cumani. Edge detection in multispectral images. *CVGIP: Graphical Models and Image Processing*, 53:40–51, 1991. 98
- [18] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, August 2001. 127
- [19] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001. 117
- [20] J. W. Eaton. Octave [online]. 1998 [visitado el 5 de junio de 2010]. URL <http://www.octave.org.c>
- [21] M. R. Everingham, H. Muller, and B. T. Thomas. Evaluating image segmentation algorithms using the Pareto Front. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision*, volume IV of *LNCS 2353*, pages 34–48. Springer, June 2002. 133, 135, 136
- [22] J. Fan, D. K. Y. Yau, A. Elmagarmid, and W. Aref. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Transactions on Image Processing*, 10(10):1454–1466, October 2001. 121
- [23] D. Forsyth. An empirical-statistical agenda for recognition. In D. A. Forsyth, J. L. Mundy, V. Gesú, and R. Cipolla, editors, *Shape, Contour and Grouping in Computer Vision*, volume 1681 of *Lecture Comments in Computer Science*, pages 9–21. Springer-Verlag, 1999. 112, 130

- [24] D. A. Forsyth and J. Ponce. *Computer Vision. A Modern Approach*. Prentice Hall, 2003. 112
- [25] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984. 127
- [26] T. Gevers and F. C. A. Groen. Segmentation of color images. In *Proc. of 7th Scandinavian Conference on Image Analysis*, 1991. 126
- [27] T. Gevers and A. W. M. Smeulders. Combining region splitting and edge detection through guided Delaunay image subdivision. In *Conference on Computer Vision and Pattern Recognition*, pages 1021–1028, Puerto Rico, June 1997. 121
- [28] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, Inc., 1992. 112, 120
- [29] R.C. González and R.E. Woods. *Digital Image Processing*. Prentice-Hall, 3rd edición, 2008. 91, 110
- [30] I. R. Greenshields. Coherent computation of the multispectral maximal directional derivative. *Image and Vision Computing*, 18:1–7, 1999. 98
- [31] R. P. Grzeszczuk and D.Ñ. Levin. “Brownian Strings”: Segmenting images with stochastically deformable contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1100–1114, October 1997. 126
- [32] K. Haris, S. Efstratiadis, N. Maglaveras, and A. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, 7(12):1684–1699, December 1998. 121, 124, 126
- [33] K. Holla. Opponent colors as a 2-dimensional feature within a model of the first stages of the human visual system. In *Proc. 6th Int. Conference on Pattern Recognition*, pages 561–563, Munich, Germany, October 1982. 116
- [34] A. Hornberg, editor. *Handbook of Machine Vision*. Wiley-VCH, 2008. 28
- [35] G. Iannizzotto and L. Vita. Fast and accurate edge-based segmentation without contour smoothing in 2-d real images. *IEEE Transactions on Image Processing*, 9(7):1232–1237, July 2000. 126
- [36] Adobe Systems Inc. Adobe photoshop family [online]. 2009 [visitado el 5 de junio de 2010]. URL <http://www.adobe.com/products/photoshop/family/>. 4
- [37] A. K. Jain. *Fundamentals of Digital Image Processing*. Information and System Sciences Series. Prentice Hall, 1989. 112, 120
- [38] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988. 126

- [39] Robust Image Understanding Laboratory. Edge detection and image segmentation (edison) system [online]. 2002 [visitado el 10 March 2004]. URL <http://www.caip.rutgers.edu/riul/research/code.html>. 120
- [40] J. Liu and Y.-H. Yang. Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):689–700, July 1994. 133
- [41] L. Lucchese and S. K. Mitra. Unsupervised low-frequency driven segmentation of color images. In *Proc. of 1999 Int'l Conf. on Image Processing (ICIP'99)*, Kobe, Japan, October 1999. 116
- [42] L. Lucchese and S. K. Mitra. Color image segmentation: A state-of-the-art survey. In *Proc. of the Indian National Science Academy (INSA-A)*, volume 67-2, pages 207–221, New Delhi, India, March 2001. 112, 121
- [43] J. Luo, R. Gray, and H. C. Lee. Towards physics-based segmentation of photographic color images. In *International Conference on Image Processing*, volume 3, pages 58–61, Washington DC, October 1997. 117
- [44] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. LeCam and J. Neyman, editors, *Proc. Fifth Berkeley Symposium on Math. Stat. and Prob.*, volume I, pages 281–297, California, 1967. University of California Press. 117
- [45] D. H. Marimont and Y. Rubner. A probabilistic framework for edge detection and scale selection. In *Proceedings of the 1998 IEEE International Conference on Computer Vision*, pages 207–214, Bombay, India, January 1998. 122
- [46] D. Marr and E. Hildreth. Theory of edge detection. In *Proceedings of the Royal Society*, volume B 207, pages 187–217, 1980. 122
- [47] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82:76–89, March 1987. 127
- [48] J. L. Marroquin and F. Girosi. Some extensions of the k-means algorithm for image segmentation and pattern classification. Technical Report AIM-1390, Massachusetts Institute Of Technology, Artificial Intelligence Laboratory A.I., 1993. 117
- [49] G. Medioni, M. S. Lee, and C. K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000. 126, 127
- [50] X. Muñoz, J. Freixenet, X. Cufí, and J. Martí. Strategies for image segmentation combining region and boundary information. *Pattern Recognition Letters*, 24:375–392, 2003. 120, 126

- [51] R. Ohlander, K. Price, and D. R. Reddy. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing*, 8:313–333, 1978. 116
- [52] Y. I. Ohta, T. Kanade, and T. Sakai. Color information for region segmentation. *Computer Graphics and Image Processing*, 13:1277–1294, 1980. 116
- [53] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, 1993. 112, 131
- [54] D. K. Panjwani and G. Healey. Markov Random Field models for unsupervised segmentation of textured color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):939–954, October 1995. 121, 127
- [55] T.Ñ. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Transactions on Signal Processing*, 40(4):901–914, April 1992. 126, 128
- [56] S. H. Park, I. D. Yun, and S. U. Lee. Color image segmentation based on 3-D clustering: Morphological approach. *Pattern Recognition*, 31(8):1061–1076, August 1998. 116
- [57] Charles Poynton. A guided tour of color space. In *Proceedings of the SMPTE Advanced Television and Electronic Imaging Conference*, pages 167–180, San Francisco, February 1995. 95
- [58] J. G. Proakis and D. G. Manolakis. *Tratamiento Digital de Señales*. Prentice Hall, 1998. 57, 60
- [59] S. Ray and R. H. Turi. Determination of number of clusters in k-means clustering and application in colour image segmentation. In N. R. Pal, A. K. De, and J. Das, editors, *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, pages 137–143, Calcutta, India, December 1999. 117
- [60] S. Ray, R. H. Turi, and P. E. Tischer. Clustering-based colour image segmentation: an evaluation study. In *Proc. of Digital Image Computing: Techniques and Applications*, pages 86–92, Australia, December 1995. 117
- [61] L. G. Roberts. Machine perception of three-dimensional solids. In J. T. Trippett, editor, *Optical and Electro-Optical Information Processing*, pages 159–197. MIT Press, Cambridge, MA, 1965. 97
- [62] J. B. T. M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000. 122
- [63] R. Ronfard. Region-based strategies for active contour models. *International Journal of Computer Vision*, 13(2):229–251, 1994. 126

- [64] M. Ruzon and C. Tomasi. Edge, junction, and corner detection using color distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1281–1295, November 2001. 122
- [65] E. Saber, A. M. Tekalp, and G. Bozdagi. Fusion of color and edge information for improved segmentation and edge linking. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, Georgia, May 1996. 126
- [66] Andrew Seager and David Liley. Basic principles of ultrasound imaging system design. Lecture Biomedical Imaging HET408, Swinburne University of Technology, Australia, Marzo 2002. URL <http://marr.bsee.swin.edu.au/lectures/het408/het408.html>. 10
- [67] L. Shafarenko, M. Petrou, and J. Kittler. Histogram-based segmentation in a perceptually uniform color space. *IEEE Transactions on Image Processing*, 7(9):1354–1358, September 1998. 116
- [68] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000. 121
- [69] D. Sinclair. Image parsing for image retrieval from large image data bases: from coloured image to coloured regions. Technical Report 97.4, AT&T Laboratories, Cambridge, 1997. URL <http://www-lce.eng.cam.ac.uk/publications/files/tr.97.4.pdf>. 126
- [70] W. Skarbek and A. Koschan. Color image segmentation. a survey. Technical Report 94-32, Technische Universität Berlin, 1994. 112, 129
- [71] P. De Smet and R. Pries. Implementation and analysis of an optimized rainfalling watershed algorithm. In *IS&T SPIE's 12th Annual Symposium Electronic Imaging 2000: Science and Technology Conference: Image and Video Communications and Processing*, pages 759–766, San Jose, California, USA, January 2000. 122, 123
- [72] S. M. Smith and J. M. Brady. SUSAN – A new approach to low level image processing. Technical Report TR95SMS1c, Oxford Centre for Functional Magnetic Resonance Image of the Brain, Oxford University, Chertsey, Surrey, UK, 1995. URL <http://citeseer.nj.nec.com/smith95susan.html>. 122
- [73] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis and machine vision*. PWS Publishing, 2nd edición, 1999. 97, 112, 120, 121
- [74] The GIMP Team. GNU Image Manipulation Program [online]. 2009 [visitado el 5 de junio de 2010]. URL <http://www.gimp.org/>. 4
- [75] S. Tominaga. A color classification method for color images using a uniform color space. In *Proc. 10th Int. Conf. on Pattern Recognition*, volume 1, pages 803–807, June 1990. 116

- [76] A. Tremeau and N. Borel. A region growing and merging algorithm to color segmentation. *Pattern Recognition*, 30(7):1191–1204, July 1997. 120
- [77] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991. 122
- [78] W3C. Scalable Vector Graphics. XML Graphics for the Web [online]. 2009 [visitado el 5 de junio de 2010]. URL <http://www.w3.org/Graphics/SVG/>. 34
- [79] A. R. Weeks and G. E. Hague. Color segmentation in the HSI color space using the k-means algorithm. In *Proc. of the SPIE – Nonlinear Image Processing VIII*, pages 143–154, San Jose, CA, February 1997. 117
- [80] S. Wesolkowski, S. Tominaga, and R. D. Dony. Shading and highlight invariant color image segmentation using the MPC algorithm. In *SPIE Color Imaging: Device-Independent Color, Color Hardcopy and Graphic Arts VI*, pages 229–240, San Jose, January 2001. 117
- [81] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, March 1998. 126
- [82] K. J. Yoon and I. S. Kweon. Color image segmentation considering of human sensitivity for color pattern variations. In *SPIE2001*, pages 269–278, Newton, USA, October 2001. 117
- [83] S. Yu. *Computational Models of Perceptual Organization*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2003. 121
- [84] C. Zhang and P. Wang. A new method of color image segmentation based on intensity and hue clustering. In *Proc. International Conference on Pattern Recognition*, volume 3, September 2000. 117
- [85] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996. 131
- [86] S. C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):884–900, September 1996. 126
- [87] D. Zugaj and V. Lattuati. A new approach of color images segmentation based on fusing region and edge segmentations outputs. *Pattern Recognition*, 31(2):105–113, 1998. 122, 126

Apéndice A

Respuestas a Problemas

A continuación se presentan los esbozos de las soluciones a los problemas planteados en los diferentes capítulos. Estos problemas tienen como objetivo conducir al lector en la profundización de la materia. Los problemas deben solucionarse primero sin considerar estas respuestas. De otra forma no se podrá asegurar el aprendizaje de conceptos.

Problema 4.2. Nótese primero que el valor de M no afecta el periodo fundamental de la señal, puesto que M y N son números primos relativos.

Las señales armónicamente relacionadas están dadas por

$$\epsilon_k(n) = e^{j(2\pi(\frac{M}{N}k)n)}$$

Determínese la señal armónicamente relacionada $\epsilon_{k+\alpha}(n)$:

$$\begin{aligned}\epsilon_{k+\alpha}(n) &= e^{j(2\pi(\frac{M}{N}(k+\alpha))n)} \\ &= e^{j(2\pi(\frac{Mk}{N} + \frac{M\alpha}{N})n)} \\ &= e^{j(2\pi(\frac{Mk}{N})n)} e^{j(2\pi(\frac{M\alpha}{N})n)}\end{aligned}$$

El segundo término del producto es igual a 1 cuando $M\alpha/N \in \mathbb{Z}$, lo cual, bajo la restricción de que M y N son primos relativos, es posible si y solo si α es un múltiplo de N . En otras palabras la frecuencia $(k + iN)\omega_0$, con $i \in \mathbb{Z}$ es un alias de $k\omega_0$. Así que solo valores de $\alpha < N$ generan frecuencias unívocas, pues con $\alpha \geq N$ es posible encontrar siempre un $\alpha < N$ que da paso a una señal equivalente.

Índice alfabético

- ACA, 128
- active contour, 126
- adquisición, 3
- alias, 61, 65
- aliasing, 69
 - espacial, 77
- amplitud, 57
- análisis
 - ecuación de, 67
- armónica
 - relación exponencial, 63
- cámara, 7
- canal, 32
- clique, 128
- clustering, 117
 - adaptive, 128
- color, 93
- complemento, 102
- contrast, 98
- coordenadas
 - universales, 23
- cuerpo negro, 15
- derivative
 - directional, 97
- DFT, 75
- Epanechnikov, 117
- evaluation
 - analytical, 131
 - discrepancy, 131
 - empirical, 131
 - goodness, 131
- exitancia, 18
- exponencial
 - relacionadas armónicamente, 63
- fase, 57
- fitness
 - mean normalized region size, 136
 - pixel-wise certainty, 138
 - pixel-wise potential accuracy, 136
 - region integrity, 138
 - region-wise information content, 137
 - region-wise potential accuracy, 137
 - throughput, 136
- flujo, 18
- frecuencia, 57
 - angular, 57
 - de plegado, 65
 - fundamental, 61
 - negativa, 58
- frecuencia espacial
 - normalizada, 64
- front
 - Pareto, 134
- Gibbs, 128
- golden set, 133
- imagen, 1
 - adquisición, 3
 - análisis, 3
 - procesamiento, 3
- MDD, 98
- mean-shift, 117, 118
- mean-shift procedure, 118
- MRF, 121, 128
- muestra
 - número, 59

muestreo
 periódico, 63
 uniforme, 63

normalized cut, 121

over-segmentation, 138

periodo
 fundamental, 60

procesamiento, 3

profile, 117

rango fundamental, 62

reflexión, 101

region growing, 120

resolución, 34

segmentación, 112

segmentation
 area-based, 120
 edge-based, 122
 evaluation, 131
 feature-space-based, 113
 hybrid, 126
 image-domain-based, 120

serie
 generalizada de Fourier, 67

simetría circular
 impar, 83
 par, 83

síntesis
 ecuación de, 67

sobre-segmentación, 113

split-and-merge, 120

teorema
 Wiener-Khinchin, 74

under-segmentation, 138

vídeo, 4, 31

visualística, 4

watershed, 122

Wiener-Khinchin, 74