

01170



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA  
DIVISION DE ESTUDIOS DE POSGRADO

CODIFICACION UNIDIMENSIONAL DE SECUENCIAS DE IMAGENES POR MEDIO DE LA TRANSFORMADA POLINOMIAL

**T E S I S**

QUE PARA OBTENER EL GRADO DE:  
MAESTRO EN INGENIERIA ELECTRICA  
(COMUNICACIONES)

P R E S E N T A  
CARLOS JOEL RIVERO MORENO



DIRECTOR DE TESIS: DR. BORIS ESCALANTE RAMIREZ

CIUDAD UNIVERSITARIA,

2000

280337



Universidad Nacional  
Autónoma de México



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

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

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

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

## **ÉSTA TESIS ESTA DEDICADA A:**

Mi Abuela Lilia Toscana  
quien permitió que yo emprendiera mi vuelo

Mis Padres, sus Parejas, y a mis Hermanos  
quienes me han proporcionado su compañía, apoyo, y amor

Mi Abuela Bertha Murillo  
quien está presente con su cariño

Dr. Alberto Siniego  
quien ha sido un gran guía y me abrió las puertas al lado humano

LAXKMMY  
quienes han sido impulso fundamental en mi vida

Dr. Boris Escalante  
quien me ha apoyado, brindado su amistad,  
y guiado en el camino del saber

Rocío Abascal  
quien con su apoyo y cariño ha sido  
parte de mi impulso para seguir adelante

A la Universidad Nacional Autónoma de México  
y especialmente a la División de Estudios de Posgrado de la Facultad de Ingeniería  
por brindarme su sabiduría y conocimiento para mi realización

Y a todos los que contribuyeron en la realización de ésta Tesis.

A todos ustedes muchas gracias por su apoyo constante e incondicional.

## **UN AGRADECIMIENTO ESPECIAL A:**

Dr. Boris Escalante Ramírez  
Dr. Francisco García Ugalde  
Dr. Miguel Moctezuma Flores  
Dr. Víctor García Garduño  
Dr. Jorge Lira Chávez

Por haber aceptado formar parte de mi jurado y por su gran apoyo

# CODIFICACIÓN UNIDIMENSIONAL DE SECUENCIAS DE IMÁGENES POR MEDIO DE LA TRANSFORMADA POLINOMIAL

## Índice

Resumen .....	3
Introducción.....	4
<b>Capítulo 1. Proyecciones Vectoriales en el Espacio de Señales .....</b>	<b>6</b>
1.1 Introducción .....	6
1.1.1. Isomorfismos .....	6
1.2. Espacios de Hilbert.....	8
1.2.1. Señales de Energía en $L_2(R)$ .....	9
1.2.2. Espacio Vectorial Lineal .....	10
1.2.3. Subespacio Vectorial.....	11
1.2.4. Conjuntos Generadores ( <i>span</i> ), Base y Dimensión.....	11
1.2.5. Producto Interno o Producto Vectorial.....	11
1.2.6. Norma de un vector y Energía de la señal.....	12
1.2.7. Distancia o Métrica entre vectores.....	13
1.2.8. Propiedades .....	14
1.3. Expansiones y Proyecciones Vectoriales .....	17
1.3.1. Coordenadas .....	18
1.3.2. Subespacios Ortogonales.....	18
1.3.3. Componentes Vectoriales.....	19
1.3.4. Bases Ortonormales .....	21
1.3.5. Proyecciones Ortogonales .....	22
1.3.6. Expansiones Sobrecompletas ( <i>Frames</i> ) .....	24
1.4. Expansión de Señales.....	25
<b>Capítulo 2. Modelos del Sistema de Percepción Visual Humana.....</b>	<b>27</b>
2.1. Introducción .....	27
2.2. Sistema Visual Primario.....	27
2.2.1. Estructura del Ojo Humano .....	27
2.2.2. Camino Visual Primario.....	30
2.3. Campos Receptivos Corticales .....	32
2.4. Modelo del Sistema Visual .....	32
2.5. Modelos de los Campos Receptivos .....	34
2.5.1. Modelo de Gabor.....	34
2.5.2. Modelo de Derivadas Gaussianas.....	35
2.6. Funciones Hermite.....	36
2.6.1. Principio de Incertidumbre de Heisenberg .....	36
2.6.2. Localización ( <i>resolución</i> ).....	36
2.6.3. Métrica de Incertidumbre Conjunta .....	37

2.6.4. Funciones y Polinomios de Hermite .....	38
2.7. Modelo basado en la Transformada Polinomial .....	41
2.7.1. Transformada Hermite .....	42
2.7.2. Transformada de Laguerre .....	42
<b>Capítulo 3. La Transformada Polinomial.....</b>	<b>43</b>
3.1. Introducción .....	43
3.2. Transformadas Polinomiales Unidimensionales.....	43
3.3. Transformada Hermite .....	46
3.3.1. Función Ventana.....	46
3.3.2. Función de peso .....	47
3.3.3. Funciones Filtro .....	47
3.3.4. Funciones Patrón.....	49
3.3.5. Transformada Hermite Finita .....	50
3.4. Transformadas Polinomiales Discretas.....	51
3.4.1. Transformada Hermite Discreta.....	52
3.5. Transformadas Polinomiales en Tres Dimensiones.....	53
3.5.1. Transformada Hermite en Tres Dimensiones.....	54
<b>Capítulo 4. Proyección Unidimensional de Secuencias de Imágenes .....</b>	<b>57</b>
4.1. Introducción .....	57
4.2. Codificación Unidimensional con la Transformada Hermite (UCHT).....	58
4.2.1. Coordenadas Esféricas .....	58
4.2.2. Rotación de Estructuras 1D.....	59
4.2.3. Proyección 1D con la Transformada Polinomial .....	60
4.2.4. Proyección 1D con la Transformada Hermite .....	62
4.3. Implementación y Consideraciones Prácticas .....	63
<b>Capítulo 5. Resultados y Evaluación.....</b>	<b>65</b>
<b>Conclusiones.....</b>	<b>78</b>
<b>Anexos .....</b>	<b>79</b>
Anexo 1. Listado de los programas para la UCHT .....	79
<b>Bibliografía .....</b>	<b>97</b>

## Resumen

El presente trabajo tiene por objetivo utilizar una representación de secuencias de imágenes a través de la transformada Polinomial, específicamente la transformada Hermite, con la innovación de codificar las expansiones tridimensionales (3D) en una dimensión (1D) a partir de las proyecciones de los coeficientes sobre las estructuras relevantes unidimensionales.

A su vez, se pretende que ésta representación codifique la información de la secuencia original con un número menor de información. Para el caso de la transformada Polinomial 3D se tiene una expansión inherente en la representación a través de los coeficientes polinomiales. Con la codificación 1D de la secuencia, se pretende optimizar la representación, de tal manera que al elegir estructuras unidimensionales de relevancia para el sistema visual humano, se tenga una compresión inherente a través de la proyección unidimensional. Para esto, se hace una interpretación de los coeficientes codificados unidimensionalmente.

Con esto, el enfoque que se le da a la codificación unidimensional de secuencias de imágenes es el de un sistema análisis/síntesis que relaciona las características del sistema visual humano y los conceptos de proyecciones vectoriales, proporcionando un esquema de codificación relevante a patrones visuales.

Finalmente, con esta forma de codificar las secuencias de imágenes, se pretende cimentar las bases a partir de las cuales se puedan realizar técnicas de procesamiento como son la estimación de movimiento y la compresión en secuencias de imágenes como consecuencia de aplicaciones de almacenamiento y teleconferencia.

## Introducción

Una secuencia de imágenes digitales es una serie ordenada de imágenes representada por una función de intensidad o luminancia  $I(x,y,z)$ , definida sobre una malla o rejilla discreta tridimensional  $S$ . Una secuencia real se forma mediante la proyección de escenas tridimensionales (3D) en un plano bidimensional (2D), conocido como plano imagen. Mientras que en la escena real, el movimiento 3D puede caracterizarse en términos de velocidad o desplazamiento 3D instantáneos de cada uno de los puntos de la escena, en la secuencia todo el movimiento ocurre en el plano imagen.

Dentro del estudio de las secuencias de imágenes, están en particular las áreas del *procesamiento* y del *análisis* de ellas. El primero se refiere al conjunto de operaciones de filtrado, interpolación y/o submuestreo espaciotemporal, y compresión, con los objetivos de mejorar la calidad visual de las imágenes, convertir las secuencias a distintos formatos de video, y reducir la cantidad de datos para su transmisión efectiva, respectivamente. El segundo trata sobre las operaciones realizadas sobre las secuencias con el propósito de obtener datos que permitan su *representación* o *interpretación*. Estas formas de representación para su interpretación, e incluso procesamiento, reciben el nombre de *codificación*.

La transformada Polinomial (Martens (1990a)) es una técnica que permite descomponer una imagen o una secuencia de ellas en componentes locales a través de una ventana de análisis. La secuencia de imágenes se aproxima mediante una suma ponderada de polinomios ortonormales a esta ventana de análisis, lo que permite llevar un procesamiento adaptivo dentro de la secuencia.

En el caso particular de emplear ventanas Gaussianas, el modelo simula algunas de las propiedades del sistema visual humano cuyo aspecto fundamental en sus primeras etapas es extraer información relevante para analizar la secuencia y representarla con menos información, dejando aquella que sea de relevante importancia y que proporcione buena percepción visual, similar a como trabaja el sistema visual humano. Hay evidencia de que esta información (Koenderink, 1984; Marr & Hildreth, 1980; Brinker & Roufs, 1992) consiste principalmente en orientación, localización, y contraste de las transiciones de luminancia. Esta información se extrae por procesamiento local de los campos receptivos. En el caso de secuencias de imágenes, un atributo importante de percepción visual es la velocidad, ya que ésta proporciona información de movimiento a lo largo del eje temporal (Brinker & Roufs, 1992).

Para el caso de la transformada Hermite, la cual se obtiene al emplear ventanas Gaussianas (Martens, 1990a), la separabilidad y simetría rotacional de los filtros de análisis permiten buscar de manera óptima las orientaciones de estructuras unidimensionales, tales como líneas, bordes, y velocidades.

El *objetivo principal* del presente trabajo, es representar una secuencia de imágenes de una manera eficiente y con menor información a la original. La manera de llevar a cabo esto es con la localización de estructuras unidimensionales en campos receptivos esféricos para posteriormente realizar la proyección sobre ellas de los coeficientes polinomiales tridimensionales (3D) y tener una representación local unidimensional (1D) con información relevante. El interés estriba en encontrar los ángulos óptimos de dichas estructuras que permitan hacer las proyecciones sobre estas y establecer a través de funciones de ángulo las relaciones que permitan reconstruir los coeficientes 3D a partir de los codificados en 1D.

A su vez, con ésta manera de codificar una secuencia de imágenes, se pretende dejar una base para continuar el procesamiento digital de secuencias de imágenes, como sería la caracterización de las mismas, estimación y análisis del movimiento, y compresión.

Dado que la propuesta de codificación unidimensional de secuencias de imágenes se basa en la Transformada Polinomial, el esquema se presenta como un sistema análisis/síntesis en que destacan patrones visuales, similar a la forma en que opera el sistema visual humano, y aplicando los conceptos de proyecciones vectoriales.

Debido a lo anterior, la codificación a desarrollar involucra los conceptos de proyecciones vectoriales, expansiones de señales polinomiales, y modelos del sistema visual humano. Esto es lo que da la estructura de presentar los capítulos que conforman este trabajo.

El capítulo 1 introduce los conceptos de espacios vectoriales y específicamente los espacios de Hilbert que son útiles para el tratamiento de señales. Se hace énfasis en las proyecciones vectoriales y su relación con la expansión y descomposición de señales para obtener aproximaciones o representaciones de ellas. Este capítulo proporciona las bases teóricas del capítulo 3.

El capítulo 2 plantea los modelos del sistema visual humano, por tratarse la etapa final de los sistemas de procesamiento de imágenes y vídeo. Se comienza con una revisión de su estructura anatómica y se plantea la importancia de los campos receptivos corticales en las etapas primarias de la percepción visual para la identificación de estructuras relevantes. Dado que los campos receptivos corticales son los codificadores primarios del sistema visual humano, tienen gran importancia para establecer modelos de codificación que permitan establecer una codificación similar a ellos. Por último, se introducen los modelos matemáticos asociados a ellos.

El capítulo 3 define la transformada Polinomial, la cual se puede ver como una fusión de las expansiones de señales vistas en el capítulo 1 con los modelos matemáticos del sistema visual humano vistos en el capítulo 2. Es aquí donde se define la transformada Hermite, caso particular de la transformada Polinomial, que introduce como ventana de análisis las ventanas Gaussianas, las cuales son la relación directa con los campos receptivos corticales. La transformada Polinomial es un esquema de análisis/síntesis que permite codificar una secuencia de imágenes.

El capítulo 4 plantea una mejora de la codificación de secuencias de imágenes con la transformada Hermite del capítulo 3. Es aquí donde se desarrollan los objetivos del presente trabajo, es decir, se crea una codificación a partir de patrones unidimensionales locales. La codificación unidimensional con la transformada Hermite (UCHT) involucra los conceptos de los capítulos anteriores y proporciona una base para el área de compresión.

El capítulo 5 es la aplicación de la codificación unidimensional presentada en el capítulo 4 a una secuencia de imágenes real. Es aquí donde se hace una interpretación de los resultados y se observa el desempeño de la UCHT como forma alternativa de representar una secuencia de imágenes.

Capítulo 1  
Proyecciones Vectoriales en el  
Espacio de Señales

# Capítulo 1. Proyecciones Vectoriales en el Espacio de Señales

## 1.1. Introducción

En lo que concierne al procesamiento de señales, es de gran utilidad explotar el *isomorfismo* existente entre los espacios vectoriales de dimensión  $N$ , cuya base matemática se establece dentro del álgebra lineal, y el conjunto de señales de la misma dimensión.

Una señal es una variable física cuyos valores dependen de otras variables, y donde el número de variables independientes definen la dimensión de esa señal. Matemáticamente esto corresponde a la definición de función, es decir, una señal es función de una o más variables.

Por ejemplo, la actividad eléctrica del corazón se puede medir como un voltaje que va cambiando con el tiempo, es decir, para determinado valor de tiempo existe un voltaje. El conjunto de todos los valores de voltaje define el comportamiento eléctrico del corazón, lo cual se conoce como electrocardiograma (ECG). Si representamos a  $V$  como el voltaje y a  $t$  como el tiempo, entonces la señal que caracteriza al ECG es  $V(t)$ , esto es,  $V$  es una función de  $t$ . Dado que  $V$  solamente depende de una variable, en este caso  $t$ , se tiene entonces que la dimensión de esta señal es uno, es decir, se trata de una señal unidimensional, lo cual se expresa como 1D.

Otro ejemplo sería el caso del vídeo digital o secuencias de imágenes, donde cada valor de intensidad luminosa  $I$  en la posición  $(x,y)$  de la imagen va cambiando con el tiempo  $t$ , es decir,  $I$  es una función de  $(x,y,t)$  representado como  $I(x,y,t)$ . Este sería el caso de una señal en tres dimensiones, o sea, 3D.

Se dice que una señal es real o compleja dependiendo del campo sobre cual se definen los valores de las variables, ya sea real  $R$ , o complejo  $C$ , donde  $R^n$  y  $C^n$  representan el conjunto de  $n$  valores ordenados  $(x_1, x_2, \dots, x_n)$  de números reales o complejos respectivamente. En este caso  $n$  define la dimensión de la señal o función  $f$ . Así por ejemplo, una función  $f(x,y,z)$  es una función 3D ya que depende de tres variables, y será una función real de variable real si:  $f: (R^3 = R \times R \times R) \rightarrow R$ .

### 1.1.1. Isomorfismos

El concepto de isomorfismo es de relevante importancia en las matemáticas; especialmente desde el punto de vista de sus aplicaciones. Dicho concepto se encuentra subyacente en el empleo mismo de modelos matemáticos, ya sea para la resolución de problemas físicos o para la resolución de problemas matemáticos más complicados.

El término “*isomorfo*”, que etimológicamente significa “de igual forma”, se emplea en el álgebra lineal para denotar la idea de que dos sistemas son tan parecidos que pueden considerarse, en esencia, como el mismo. Es este el caso de los espacios vectoriales que se definen dentro del álgebra lineal, y las señales estudiadas en ingeniería para su procesamiento.

En general, la sustitución de los elementos de un conjunto A por los elementos de un conjunto B puede hacerse mediante una función  $f: A \rightarrow B$ . Cuando dicha función es biyectiva los elementos de A y de B se encuentran “uno a uno”, y cada uno de ellos puede considerarse como el “reflejo” de su elemento correspondiente en el otro conjunto.

Si en el conjunto  $A$  está definida una operación  $\star$  y en el conjunto  $B$  una operación  $\Delta$ , es necesario que los resultados obtenidos en el sistema  $(A, \star)$  se conserven al efectuar las operaciones en el sistema  $(B, \Delta)$  con las imágenes respectivas; por lo que la función  $f$  debe ser tal que

$$f(a \star b) = f(a) \Delta f(b); \forall a, b \in A$$

Esta propiedad de la función  $f$  garantiza que se llega al mismo resultado empleando cualquiera de los dos procedimientos siguientes:

1. Efectuando la operación  $\star$  en el sistema  $(A, \star)$  y aplicando después la función  $f$  al resultado.
2. Aplicando la función  $f$  a cada uno de los elementos y efectuando después la operación  $\Delta$  con las imágenes en el sistema  $(B, \Delta)$ .

La diferencia entre ambos procedimientos puede apreciarse claramente en la figura 1

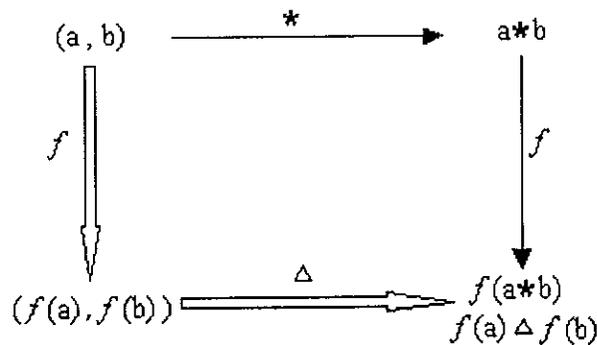


Figura 1. Isomorfismo entre  $(A, \star)$  y  $(B, \Delta)$

Una función que satisface dicha propiedad se dice que es un “*homomorfismo*” de  $(A, \star)$  en  $(B, \Delta)$ . Cuando la función es, además, biyectiva, se dice que es un “*isomorfismo*” entre  $(A, \star)$  y  $(B, \Delta)$ , y que dichos sistemas son “isomorfos”.

El caso particular del isomorfismo ofrece una ventaja adicional: por ser  $f$  una función biyectiva existe su inversa  $f^{-1}$  y podemos, mediante ésta última, emprender el “regreso” del sistema  $(B, \Delta)$  al sistema  $(A, \star)$  una vez que se ha efectuado la operación. Esto último fundamenta el empleo de modelos matemáticos para la resolución de problemas físicos ya que, debido al isomorfismo, es posible emplear un proceso como el que se indica en la figura 2.

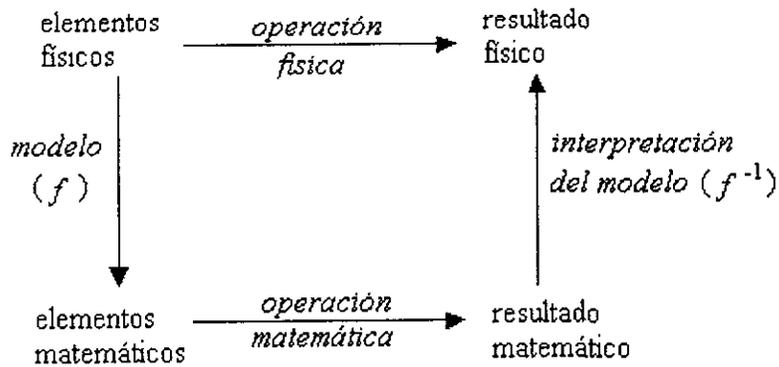


Figura 2. Isomorfismo entre modelos matemáticos y físicos

En el caso del conjunto de señales de dimensión  $N$  y la definición de un espacio vectorial de misma dimensión hay un isomorfismo entre ambos. El espacio vectorial viene a ser el conjunto de señales, mientras que cada señal del conjunto es un vector del mismo.

Por último, las definiciones que se desarrollan para espacios vectoriales de dimensión  $N$  se pueden obtener directamente a través de aquellas de dimensión uno, haciendo el mapeo del conjunto de valores ordenados, que representan las variables independientes, a una variable vectorial, es decir,  $f(x_1, x_2, \dots, x_n) = f(r)$ ;  $r = (x_1, x_2, \dots, x_n)$ . En el resto de este capítulo veremos las principales definiciones para una dimensión (1D), y en la mayoría de los casos denotaremos a  $f(r)$  simplemente por  $f$ .

## 1.2. Espacios de Hilbert

Las secciones 1.2 y 1.3 de éste capítulo están basadas principalmente en la información encontrada en Vetterli & Kovačević (1995), y Stremmer (1982).

El interés principal de hacer un tratamiento de las señales a través de su representación vectorial radica en la necesidad de aproximar una señal  $f$  por medio de una combinación lineal de un conjunto de señales (funciones) conocidas  $\varphi_k$ . Este conjunto será finito ( $N$  finito) o infinito ( $N \rightarrow \infty$ ) dependiendo de la dimensión  $N$  del espacio de señales. Lo anterior se representaría como:

$$f = \sum_{k=1}^N f_k \cdot \varphi_k \quad (1.1)$$

Los valores numéricos,  $f_k$ , que representan la combinación lineal de estas funciones conocidas,  $\varphi_k$ , forman un conjunto de valores que no dependen de las variables independientes de la señal, y es lo que se conoce como *coordenadas* o *proyecciones* de la señal  $f$  sobre las señales o vectores  $\varphi_k$  que caracterizan al espacio vectorial, es decir, las señales  $\varphi_k$  equivalen a los “ejes coordenados” que son ortogonales entre sí.

Otro interés radica en que la energía de la señal  $f$  se concentra más en algunas funciones  $\varphi_k$  que en otras, pudiendo así descartarse esos valores de menor energía (haciéndolos cero) y teniendo aún así una buena aproximación de la señal  $f$ .

Por último, derivando de lo anterior, se puede obtener representaciones de señales de dimensión  $N$  en espacios de dimensión menor, utilizando los conceptos de *proyección vectorial*.

### 1.2.1. Señales de Energía en $L_2(\mathcal{R})$

Dada una señal  $f(t)$  sobre el campo  $\mathcal{R}$ , se define la energía de una señal,  $E_f$ , como la integral de la magnitud al cuadrado de dicha señal, esto es

$$E_f = \int_{-\infty}^{\infty} |f(t)|^2 dt \quad (1.2)$$

Una *señal de energía* es aquella para la cual la ecuación (1.2) es finita, aún cuando el intervalo de integración sea infinito; esto es, cuando  $|f|^2$  es integrable.

$$\int_{-\infty}^{\infty} |f(t)|^2 dt < \infty \quad (1.3)$$

Si la ecuación (1.3) se satisface, se dice que la señal de energía  $f$  es cuadrado-integrable y que pertenece al conjunto de señales denotado por  $L_2(\mathcal{R})$ , donde el conjunto  $\mathcal{R}$  es el campo de los números reales.

Para el caso de señales discretas, tenemos una secuencia relacionada a la señal continua o función de variable continua, en donde frecuentemente la primera es una versión muestreada de la segunda, y dicha secuencia está indexada por un valor entero, esto es:

$$x[n] = f(nT), \quad n \in \mathcal{Z}, T \in \mathcal{R} \quad (1.4)$$

Como se puede observar en la ecuación (1.4), utilizamos paréntesis alrededor de una variable continua y corchetes alrededor de una discreta, esto es,  $f(t)$  y  $x[n]$ . Además, la integral de una variable continua se convierte en una sumatoria para el caso discreto. La expresión equivalente a la ecuación (1.3) para el caso discreto es:

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty, \quad n \in \mathcal{Z} \quad (1.5)$$

De igual manera al caso continuo, si la expresión (1.5) se satisface, entonces  $x[n] \in L_2(\mathcal{R})$ .

En particular,  $\delta(t)$  y  $\delta[n]$  denotan las funciones *delta de Dirac* para el caso continuo y *delta de Kronecker* para el caso discreto respectivamente, en donde

$$\int_{-\infty}^{\infty} \delta(t) dt = 1; \quad \delta(t) = 0, \quad \text{si } t \neq 0 \quad (1.6.1)$$

$$\delta[n] = \begin{cases} 1, & \text{si } n = 0 \\ 0, & \text{si } n \neq 0 \end{cases} \quad (1.6.2)$$

El álgebra lineal se dedica al estudio de los espacios vectoriales de dimensión finita, y cuyos operadores se pueden representar por matrices también de dimensión finita  $n$ . Estos espacios involucran a los vectores sobre  $R$  y  $C$ , denotados por  $R^n$  y  $C^n$  respectivamente. En ellos se definen conceptos importantes como son la norma o longitud de un vector.

La idea es generalizar los conceptos de un espacio vectorial finito al caso de un espacio vectorial de *dimensión infinita*, el cual corresponde al espacio de señales. Para esto es necesario restringir a los vectores o señales a una norma o longitud finita, a pesar de que sean de dimensión infinita. Esto conduce a lo que se llaman *espacios de Hilbert*.

Por ejemplo, el espacio de secuencias cuadrado-sumables, denotado por  $L_2(Z)$ , es el espacio vectorial  $C^{\infty}$  con una restricción de norma. Un ejemplo de un conjunto de vectores generadores de  $L_2(Z)$  es el conjunto  $\{\delta[n-k]\}$ ,  $k \in Z$ . Una extensión adicional con respecto al álgebra lineal es que los vectores como conjuntos ordenados de valores reales o complejos pueden ser generalizados para incluir funciones de una variable continua.

A continuación se definirán los espacios de Hilbert considerando las señales  $\forall f, g, h \in V$ , siendo  $V$  el conjunto o espacio de señales definido sobre el campo complejo  $(C, +, \cdot)$ , y considerando los escalares  $\forall a, b, c \in C$ .

## 1.2.2. Espacio Vectorial Lineal

Sea  $V$  un conjunto de vectores definido sobre el conjunto de números complejos  $C$  que forma un campo  $(C, +, \cdot)$  con respecto a la adición y multiplicación.  $V$  es un espacio vectorial lineal si para  $\forall f, g, h \in V$ , y  $\forall a, b, c \in C$  se cumple que

(1)  $(V, +)$  es un grupo conmutativo, es decir

- |                     |  |         |
|---------------------|--|---------|
| a) Cerradura:       | $f + g \in V$                          | (1.7.1) |
| b) Conmutatividad:  | $f + g = g + f$                        | (1.7.2) |
| c) Asociatividad:   | $(f + g) + h = f + (g + h)$            | (1.7.3) |
| d) Elemento Neutro: | $\exists 0 \in V \mid f + 0 = f$       | (1.7.4) |
| e) Complemento:     | $\exists (-f) \in V \mid f + (-f) = 0$ | (1.7.5) |

(2) Además se cumple con respecto a la multiplicación  $\cdot$  el producto escalar, el cual satisface

- |                               |   |         |
|-------------------------------|---|---------|
| f) Cerradura:                 | $a \cdot f \in V$                           | (1.8.1) |
| g) Distributividad Vectorial: | $a \cdot (f + g) = a \cdot f + a \cdot g$   | (1.8.2) |
| h) Distributividad Escalar:   | $(a + b) \cdot f = a \cdot f + b \cdot f$   | (1.8.3) |
| i) Asociatividad:             | $(a \cdot b) \cdot f = a \cdot (b \cdot f)$ | (1.8.4) |
| j) Elemento Idéntico:         | $1 \cdot f = f$                             | (1.8.5) |

Si el valor de las señales y los escalares son reales, entonces se tiene un espacio vectorial Real.

### 1.2.3. Subespacio Vectorial

Un subconjunto  $S$  de  $V$  es un subespacio de  $V$  si se cumplen las siguientes propiedades de *cerradura* en  $S$  respecto a la adición (+) y a la multiplicación ( $\cdot$ ) para  $\forall f, g \in S$  y  $\forall a \in C$ .

$$\text{a) Adición:} \quad f + g \in S \quad (1.9.1)$$

$$\text{b) Multiplicación:} \quad a \cdot f \in S \quad (1.9.2)$$

### 1.2.4. Conjuntos Generadores (*span*), Base y Dimensión

Dado  $S \subset V$ , el conjunto generador (*span*) de  $S$  es el subespacio de  $V$  el cual consiste de todas las combinaciones lineales de los vectores en  $S$ , esto es

$$\text{span}(S) = \left\{ \sum_{k=1}^N a_k f_k \mid a_k \in C, f_k \in S \right\} \quad (1.10)$$

Se dice que los vectores  $f_1, f_2, \dots, f_N$  son *linealmente independientes* si se cumple que  $\sum_{k=1}^N a_k f_k = 0$  si y sólo si  $a_k = 0, \forall k$ . En caso contrario, se dice que los vectores son *linealmente dependientes*.

Un conjunto  $\{f_1, f_2, \dots, f_N\}$  de un espacio vectorial  $V$  es una *base* de  $V$  si se cumple que el conjunto es un generador de  $V$ , es decir,  $V = \text{span}(f_1, f_2, \dots, f_N)$ , y  $f_1, f_2, \dots, f_N$  son linealmente independientes.

Si  $B = \{f_1, f_2, \dots, f_N\} \in V$  es una base del espacio vectorial  $V$  sobre  $C^N$ , se dice que  $V$  es de *dimensión  $N$* , denotado por  $\text{dim}(V) = N$ .

El espacio vectorial  $V$  es de *dimensión infinita* si contiene un conjunto infinito de vectores linealmente independientes.

Los vectores  $\{f_1, f_2, \dots, f_N\} \in V$  se llaman también *vectores base*. Para el caso de señales o de funciones, corresponden a las *señales o funciones base*.

### 1.2.5. Producto Interno o Producto Vectorial

Sea  $V$  un espacio vectorial sobre  $C$ . Un producto interno en  $V$  es una función de  $V \times V$  en  $C$ , denotada por  $\langle \cdot, \cdot \rangle$  que asigna a cada pareja ordenada  $(f, g)$  de vectores de  $V$  un escalar  $\langle f, g \rangle \in C$ , que satisface las siguientes propiedades:

$$\text{a) } \langle f + g, h \rangle = \langle f, h \rangle + \langle g, h \rangle \quad (1.11.1)$$

$$\text{b) } \langle f, a \cdot g \rangle = a \langle f, g \rangle \quad (1.11.2)$$

$$c) \langle f, g \rangle^* = \langle g, f \rangle \text{ (Simetría Hermitiana)} \quad (1.11.3)$$

$$d) \langle f, g \rangle \geq 0, \text{ y } \langle f, f \rangle = 0 \text{ si y sólo si } f \equiv 0 \quad (1.11.4)$$

donde  $a^*$  denota el complejo conjugado de  $a \in C$ .

De las propiedades (b) y (c) tenemos que

$$\langle a \cdot f, g \rangle = a^* \langle f, g \rangle \quad (1.12)$$

y en general

$$\langle f, a \cdot g + b \cdot h \rangle = a \langle f, g \rangle + b \langle f, h \rangle \quad (1.13.1)$$

$$\langle a \cdot f + b \cdot g, h \rangle = a^* \langle f, h \rangle + b^* \langle g, h \rangle \quad (1.13.2)$$

De (a) y (b) es claro que el producto interno es lineal.

Un espacio de producto interno real, es decir, definido sobre  $R$ , se llama también *Espacio Euclidiano*.

Un espacio de producto interno complejo, es decir, definido sobre  $C$ , se llama también *Espacio Pre-Hilbert*.

Un tipo de producto interno que satisface las condiciones anteriores es aquel que se define para el espacio de señales continuas sobre  $R$ , y sobre  $Z$  para el caso de señales discretas, ambas valuadas en forma compleja, y los cuales son

$$\langle f, g \rangle = \int_{-\infty}^{\infty} w^2(t) f^*(t) g(t) dt \quad (1.14.1)$$

$$\langle f, g \rangle = \sum_{n=-\infty}^{\infty} w^2[n] f^*[n] g[n] \quad (1.14.2)$$

siempre y cuando  $w(t)$  y  $w[n]$  sean funciones reales.  $w$  es una función real fija positiva llamada "*función de peso*" que permite la convergencia (existencia) del producto interno. El producto interno para cuando  $w=1$  corresponde al caso de espacios de señales que pertenecen a  $L_2(C)$ , es decir, al espacio de funciones cuadrado integrables sobre el conjunto o campo  $C$ ; mientras que si  $w \neq 1$  entonces el producto vectorial se extiende a señales que no pertenecen a  $L_2(C)$ .

### 1.2.6. Norma de un vector y Energía de la señal

La norma o longitud de un vector se define a partir del producto interno como

$$\|f\| = \sqrt{\langle f, f \rangle} \quad (1.15)$$

La norma asigna un número real,  $\|f\|$ , a cada vector  $f$ . Para que sea útil, se necesita que  $V$  sea un espacio normado, es decir, para  $\forall f, g \in V$ , y  $\forall a \in C$  que se cumpla

$$a) \|f\| \geq 0; \text{ si } \|f\| = 0 \Rightarrow f = 0 \quad (1.16.1)$$

$$b) \|af\| = |a|\|f\| \quad (1.16.2)$$

$$c) \|f + g\| \leq \|f\| + \|g\| \quad (\text{Desigualdad de Minkowsky}) \quad (1.16.3)$$

Un vector para el cual se cumple que  $\|f\| = 1$  se llama *normalizado*.

Aplicando la definición de norma para el espacio de señales en  $L_2(C)$  se observa que

$$\|f\|^2 = \langle f, f \rangle = \int_{-\infty}^{\infty} f^*(t)f(t)dt = \int_{-\infty}^{\infty} |f(t)|^2 dt \quad (1.17.1)$$

$$\|f\|^2 = \langle f, f \rangle = \sum_{n=-\infty}^{\infty} f^*[n]f[n] = \sum_{n=-\infty}^{\infty} |f[n]|^2 \quad (1.17.2)$$

Lo cual de acuerdo a la ecuación (1.2) se observa que la norma al cuadrado de una señal  $\|f\|^2$  es igual al valor de su *energía*  $E_f$ .

Nótese que otros tipos de norma pueden ser definidos siempre y cuando las ecuaciones (1.16) se satisfagan. En el caso de las señales de energía, la *norma cuadrada* o *Euclidiana* definida por la ecuación (1.15) y las ecuaciones (1.17), es la que se utiliza.

A su vez, la norma Euclidiana, o longitud del vector, tiene todas las propiedades necesarias para que cada espacio con producto interno sea también un espacio normado, es decir, aquel en que se puede tener una medida de distancia entre dos vectores.

### 1.2.7. Distancia o Métrica entre vectores

Para comparar que tanto se aproximan entre sí dos señales, se define la distancia entre ellas que. Esta medida de la distancia o métrica se puede tener en cualquier espacio vectorial.

La distancia entre dos señales o vectores,  $f$  y  $g$ , es simplemente la norma o magnitud de sus diferencias, denotada por

$$d(f, g) = \|f - g\| \quad (1.18)$$

que cumple las siguientes propiedades para  $\forall f, g, h \in V$ :

$$a) d(f, g) \geq 0; \text{ si } d(f, g) = 0 \Rightarrow f = g \quad (1.19.1)$$

$$b) d(f, g) = d(g, f) \quad (\text{Simetría}) \quad (1.19.2)$$

$$c) \quad d(f, h) \leq d(f, g) + d(g, h) \quad (\text{Desigualdad del Triángulo}) \quad (1.19.3)$$

Un *espacio normado* es aquel en que puede tener una medida de la distancia, y también se conoce como *espacio métrico*.

Un tipo especial de espacio normado es el *espacio de producto interno* o *espacio Pre-Hilbert*.

Cabe mencionar que la distancia entre dos señales indica que tan parecida es una señal a otra, a tal grado que si las señales son iguales entonces la distancia es cero, de acuerdo a la ecuación (1.19.1). Entre más cercano sea un vector a otro, la distancia entre ellos será menor.

Aplicando la definición de distancia para el espacio de señales en  $L_2(C)$  se observa que

$$d^2(f, g) = \|f - g\|^2 = \langle f - g, f - g \rangle = \int_{-\infty}^{\infty} |f(t) - g(t)|^2 dt \quad (1.20.1)$$

$$d^2(f, g) = \|f - g\|^2 = \langle f - g, f - g \rangle = \sum_{n=-\infty}^{\infty} |f[n] - g[n]|^2 \quad (1.20.2)$$

Por lo que  $d^2(f, g) = \|f - g\|^2$  corresponde al *error cuadrático medio (MSE)* entre las señales  $f$  y  $g$ , es decir, a la *varianza*  $\sigma_{fg}^2$  entre las dos señales. En otras palabras, es una medida de la *energía del error* o *diferencia* entre las dos señales, la cual disminuye si las señales se aproximan entre sí; y ésta vale cero,  $\sigma_{fg}^2 = \|f - g\|^2 = 0$ , si las dos señales son iguales,  $f = g$ .

### 1.2.8. Propiedades

Las siguientes propiedades se satisfacen para los *espacios de Hilbert* que son espacios vectoriales con producto interno:

#### Desigualdad de Cauchy-Schwarz

$$|\langle f, g \rangle| \leq \|f\| \cdot \|g\| \quad (1.21)$$

con igualdad si y sólo si  $f = a \cdot g$ .

#### Desigualdad del triángulo

$$\|f + g\| \leq \|f\| + \|g\| \quad (1.22)$$

con igualdad si y sólo si  $f = a \cdot g$ , donde  $a$  es una constante positiva real.

## Ley del Paralelogramo

$$\|f + g\|^2 + \|f - g\|^2 = 2 \cdot \|f\|^2 + 2 \cdot \|g\|^2 \quad (1.23)$$

Esta ley que se satisface para una *norma de producto interno* caracteriza a los *espacios de Hilbert*, es decir, espacios vectoriales de producto interno completo; y la cual indica que una señal se puede aproximar a otra

## Punto más cercano

Sea  $U$  un subconjunto de un espacio vectorial complejo  $V$ . Este subconjunto es convexo si  $[\lambda \cdot f_1 + (1 - \lambda) \cdot f_2] \in U$ ,  $0 < \lambda < 1$ ,  $\forall f_1, f_2 \in U$ .

La propiedad del punto más cercano establece que para cualquier vector  $g$  de  $V$  existe un *vector único*  $g_u$  en el conjunto convexo  $U$  que está mas cerca de  $g$  que cualquier otro vector en  $U$ , es decir

$$d(g, g_u) = \|g - g_u\| = \inf_{f \in U} \|g - f\| \quad (1.24)$$

La propiedad del punto más cercano nos permite hacer proyecciones de un conjunto de vectores a otro subconjunto del mismo.

En un *espacio de Banach* (véase más adelante), al contrario, existen *infinitos vectores* más cercanos.

## Identidad de Polarización

Para  $\forall f, g \in V$  se satisface la identidad de polarización dada por

$$\langle f, g \rangle = \frac{1}{4} \sum_{n=0}^3 (-j)^n \cdot \|f + j^n g\|^2, \quad j^2 = -1 \quad (1.25)$$

Debido a ésta identidad de polarización, la definición de *norma* y *producto vectorial* se pueden intercambiar en un espacio de *producto interno complejo*.

## Ángulo relativo, Ortogonalidad, y Ortonormalidad

El ángulo relativo entre dos señales se define por:

$$\cos \theta(f, g) = \left\langle \frac{f}{\|f\|}, \frac{g}{\|g\|} \right\rangle \quad (1.26)$$

Dos vectores  $f$  y  $g$  son *ortogonales*,  $f \perp g$ , si y sólo si  $\langle f, g \rangle = 0$ . (1.27)

Dos vectores  $f$  y  $g$  son *ortonormales* si y sólo si los vectores son ortogonales y a su vez son unitarios, es decir, normalizados, por lo que se cumple que  $\|f\| = \|g\| = 1$  y  $\langle f, g \rangle = 0$ .

Esto es, los vectores  $f_i$  y  $f_j$  son ortonormales si su producto interno es igual a la delta de Kronecker definida en (1.6 2) para  $i=j$ , esto es

$$\langle f_i, f_j \rangle = \delta_{ij} = \delta[i - j] \quad (1.28)$$

### Teorema de Pitágoras

El teorema de Pitágoras dentro de espacios Euclidianos y espacios de Hilbert se satisface si dos vectores son ortogonales, lo cual se denota por  $f \perp g$ , esto es

$$\|f + g\|^2 = \|f\|^2 + \|g\|^2 \quad (1.29)$$

debido a que  $\|f + g\|^2 = \langle f + g, f + g \rangle = \|f\|^2 + \langle f, g \rangle + \langle g, f \rangle + \|g\|^2$

### Secuencia de Cauchy

Al aproximar una señal  $f$ , se genera una secuencia de señales  $\{f_k\}$  para las cuales la distancia  $d(f_k, f)$  disminuye conforme  $k$  crece.

Esa secuencia  $\{f_k\}$  en un espacio normado  $V$ , es decir, en un espacio en que se satisfacen las ecuaciones (1.16), se llama *secuencia de Cauchy* si para cada  $\varepsilon > 0$  hay un entero  $k_c$  tal que  $k, l > k_c$  implica que  $d(f_k, f_l) < \varepsilon$ .

### Espacio de Banach

Un *espacio normado completo*, también llamado *espacio de Banach*, es un espacio para el cual toda *secuencia de Cauchy* converge a un límite bien definido.

Ejemplos de Espacios de Banach:

- 1) Espacio vectorial de dimensión finita.
- 2) Espacio vectorial de funciones cuadrado-integrables,  $L_2(C)$ , para la mayoría de las normas de interés, tal como la norma del producto interno.

Los *espacios de Hilbert* son una subclase de los espacios de Banach.

### Mapeos Lineales

Los *Mapeos Lineales* son, también llamados *Mediciones sobre Señales*, son mapeos entre los vectores o señales y números complejos.

Una "*funcional lineal L*" sobre un espacio vectorial  $V$  es un mapeo que asigna un número complejo  $L(f)$  a cada vector  $f$  de  $V$  de tal forma que

$$L(a \cdot f + b \cdot g) = a \cdot L(f) + b \cdot L(g) \quad (1.30)$$

para  $\forall a, b \in C$  y  $\forall f, g \in V$ .

Todas las *funcionales lineales continuas* en un espacio de Hilbert tienen la forma  $L(f) = \langle \theta_L, f \rangle$  para algún vector dado  $\theta_L \in V$

Por lo tanto, el producto interno formaliza como se llevan a cabo mediciones sobre las señales, es decir, dejando que la señal  $f$  interactúe con la *señal de medición*  $\theta_L$ .

Un ejemplo de medición sobre una señal  $f$  en el espacio  $L_2(C)$  es su *valor promedio*  $L(f) = M_f$ , en donde la señal de medición sería equivalente a la unidad, es decir,  $\theta_L = 1$ . Con esto y aplicando la norma de producto interno en que  $w = 1$ , se tiene que

$$M_f = L(f) = \langle \theta_L, f \rangle = \int_{-\infty}^{\infty} f(t) dt \quad (1.31)$$

La señal de medición  $\theta_L = 1$  indica un valor constante, que en el espacio de señales corresponde al valor de DC (corriente directa) de la señal, es decir, su valor promedio. El hecho de que se tenga la unidad como medición implica obtener el 100% de dicho valor de DC.

### 1.3. Expansiones y Proyecciones Vectoriales

En esta parte retomaremos el planteamiento de la ecuación (1.1) en la cual deseamos aproximar una señal a través de una combinación lineal, es decir, como un conjunto de números  $\{f_k\}$ , de un conjunto de señales conocidas que son linealmente independientes  $\{\phi_k\}$ , las cuales constituyen un conjunto generador o span, y a su vez una base, de acuerdo a las definiciones vistas en la sección 1.2.4

Algo interesante es cuando la señal se aproxima por un subespacio vectorial  $W$ , de tamaño menor al espacio vectorial  $V$ , donde  $W \in V$ , si  $N$  es la dimensión del espacio vectorial  $V$ , entonces  $dim(W) \leq N$ ; la igualdad se cumple si  $W = V$ .

Como caso particular, el espacio vectorial euclidiano sobre el campo  $R^3$  es un espacio de dimensión  $N=3$ , teniendo como *base* al conjunto de vectores ortonormales (ortogonales y unitarios) representados por  $\{\hat{i}, \hat{j}, \hat{k}\}$  que caracterizan a los ejes coordenados  $(x, y, z)$ , es decir, estos vectores forman un conjunto generador (*span*) y a su vez una base (vectores linealmente independientes), ya que cualquier vector en  $R^3$  se puede obtener como una combinación lineal de ellos.

Los valores numéricos  $\alpha_i, i = 1, 2, 3$ ; que permiten obtener un vector  $\bar{x} \in R^3$  a partir de una combinación lineal de la base  $\{\hat{i}, \hat{j}, \hat{k}\}$  reciben el nombre de coordenadas y corresponden a las magnitudes de los vectores de la base sobre los cuales el vector  $\bar{x}$  se *proyecta*

perpendicularmente, esto es,  $\bar{x} = \alpha_1 \hat{i} + \alpha_2 \hat{j} + \alpha_3 \hat{k}$ . En otras palabras, el vector  $\bar{x}$  se obtiene como una *expansión* vectorial sobre la base  $\{\hat{i}, \hat{j}, \hat{k}\}$ .

En esta sección retomaremos los conceptos vistos en la sección 1.2.4 de conjuntos generadores (*span*) y bases de un espacio vectorial. En esto se basa la *expansión de señales*, esto es, las *transformaciones de señales*.

### 1.3.1. Coordenadas

Sea  $B = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  una base de un espacio vectorial  $V$  sobre  $C$ , y sea  $f \in V$ . Si

$$f = c_1 \cdot \varphi_1 + c_2 \cdot \varphi_2 + \dots + c_n \cdot \varphi_n \quad (1.32)$$

es decir

$$f = \sum_{k=1}^n c_k \cdot \varphi_k \quad (1.33)$$

los escalares  $c_1, c_2, \dots, c_n$  se llaman *coordenadas* de  $f$  en la base  $B$ ; y el vector de  $C^n$

$$(f)_B = [c_1, c_2, \dots, c_n]^T = [c_k]^T, \quad k = 1, \dots, n \quad (1.34)$$

se llama *vector de coordenadas* de  $f$  en la base  $B$ .

Si observamos, las ecuaciones (1.1) y (1.34) son idénticas. Por otro lado, para espacios vectoriales infinitos,  $n \rightarrow \infty$ , por lo que el vector de coordenadas  $(f)_B = [c_k]^T$  contiene un número infinito de elementos, el cual se puede considerar como una *señal discreta*.

Cabe hacer notar que, mediante el concepto de vector de coordenadas, es posible representar a un vector de cualquier espacio vectorial mediante un conjunto ordenado de escalares; es decir, mediante un elemento de  $C^n$ . Más aún, la función  $T: V \rightarrow C^n$  definida por

$$T(f) = (f)_B \quad (1.35)$$

para una base  $B$  fija, es un isomorfismo, además, dicha representación es única, es decir, el vector  $(f)_B$  es único.

### 1.3.2. Subespacios Ortogonales

Ampliaremos el concepto de ortogonalidad entre dos vectores, de acuerdo a las ecuaciones (1.27) y (1.28), a la ortogonalidad de un vector respecto a un conjunto de vectores.

Un *vector*  $f$  se dice que es *ortogonal* respecto al conjunto de vectores  $S = \{g_k\}$  si  $\langle f, g_k \rangle = 0$  para toda  $k$ . Esto se representa por  $f \perp S$ . Esto es una generalización de ortogonalidad para todos los vectores del conjunto  $S$ , de acuerdo a la ecuación (1.27).

En forma más general, dos subespacios  $S_1$  y  $S_2$  son *ortogonales* entre sí, si todos los vectores en  $S_1$  son ortogonales a todos los vectores en  $S_2$ , lo cual se expresa como  $S_1 \perp S_2$ .

Un conjunto de vectores  $\{f_1, f_2, \dots\}$  se llama *ortogonal* si  $f_i \perp f_j$  cuando  $i \neq j$ . Si los vectores están normalizados, es decir, si satisfacen la ecuación (1.28) lo cual implica que son de norma unitaria, se dice que el conjunto o *sistema es ortonormal*.

Los vectores en un sistema ortonormal son linealmente independientes. A su vez, un sistema ortonormal en un espacio vectorial  $V$ , es una *base ortonormal*  $B$  de  $V$  si genera a dicho espacio, es decir, si  $V = \text{span}(B)$ .

Dado un espacio de Hilbert  $V$  y un subespacio  $S \in V$ , llamamos el *complemento ortogonal* de  $S$  en  $V$ , denotado por  $S^\perp$ , al conjunto  $S^\perp = \{h \in V \mid h \perp S\}$ . Si además asumimos que  $S$  es cerrado, esto es, que contiene todos los límites de secuencias de vectores en  $S$  (véase "secuencias de Cauchy" en la sección 1.2.8); entonces, dado un vector  $g$  en  $V$ , existe un vector único  $f$  en  $S$  y un vector único  $h$  en  $S^\perp$  tal que  $g = f + h$ , es decir,  $h = f^\perp$ . De aquí que

$$V = S \oplus S^\perp \tag{1.36}$$

En otras palabras,  $V$  es la suma directa del subespacio y su complemento ortogonal. Lo anterior se puede apreciar gráficamente en la figura 3 considerando el espacio vectorial  $\mathbb{R}^3$ :

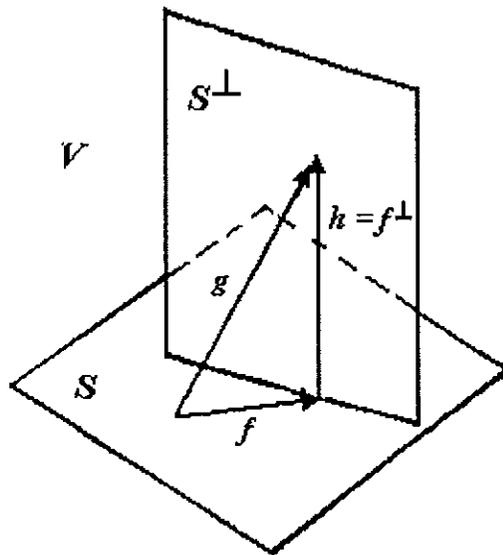


Figura 3. Espacio vectorial obtenido como la suma de un subespacio y su complemento ortogonal

### 1.3.3. Componentes Vectoriales

Un vector  $f$  está normalizado si su norma es unitaria, esto es, si  $\|f\| = 1$ . Si representamos al vector normalizado de  $f$  por  $\hat{f}$ , tal que  $\|\hat{f}\| = 1$ , éste se obtiene a través de

$$\hat{f} = \frac{f}{\|f\|} \quad (1.37)$$

La componente de un vector  $f$  “proyectado” perpendicularmente sobre un vector  $g$ , lo cual se expresa como  $\|r\| = \text{comp}_g f$ , corresponde a una “longitud” o magnitud que se denomina *componente de  $f$  sobre  $g$* , tal y como se aprecia en la figura 4.

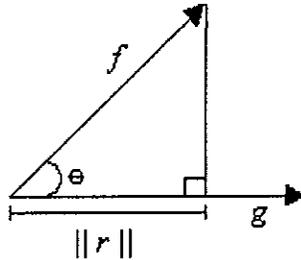


Figura 4. Proyección perpendicular de  $f$  sobre  $g$  (componente de  $f$  sobre  $g$ )

El vector  $r$  resultante de la proyección se obtiene como  $r = \|r\| \cdot \hat{g}$ . El ángulo  $\theta$  es el formado por los vectores  $f$  y  $g$ . De trigonometría se sabe que

$$\|r\| = \|f\| \cos \theta \quad (1.38)$$

Aplicando la ecuación (1.26) para el  $\cos \theta$  y la propiedad (1.12), considerando que  $\|f\|$  es real, en la ecuación (1.38) obtenemos que

$$\|r\| = \text{comp}_g f = \langle f, \hat{g} \rangle = \frac{\langle f, g \rangle}{\langle g, g \rangle} \quad (1.39)$$

y finalmente el vector  $r$  correspondiente a la *proyección perpendicular* del vector  $f$  sobre el vector  $g$  viene dado por  $r = \|r\| \cdot \hat{g}$ , cuya magnitud es precisamente  $\|r\|$ .

Para el caso de un espacio vectorial  $V$  con producto interno en el cual existe una *base ortogonal*  $B = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ , entonces  $\forall f \in V$ , se puede obtener como una combinación lineal de los vectores de dicha base de acuerdo a la ecuación (1.32), y cuyos valores representan las coordenadas del vector  $f$ , según ecuación (1.34). De acuerdo a la definición de componente de un vector sobre otro, las coordenadas del vector  $f$  representan sus componentes vectoriales sobre los vectores de la base  $B$ . Esto es

$$(f)_B = [c_k]^T, \text{ donde } c_k = \frac{\langle f, \varphi_k \rangle}{\langle \varphi_k, \varphi_k \rangle} = \text{comp}_{\varphi_k} f \quad (1.40)$$

En particular, si  $B$  es una *base ortonormal*, entonces las componentes  $c_k$  vienen dadas por

$$c_k = \langle f, \varphi_k \rangle \quad (1.41)$$

Para una *base ortonormal*  $B = \{\varphi_k\}$ , los coeficientes  $c_k$  de la expansión de  $f$ , de acuerdo a la ecuación (1.33) y los cuales vienen dados por (1.41), se llaman “*coeficientes de Fourier*” de  $f$  (con respecto a  $\{\varphi_k\}$ ).

### 1.3.4. Bases Ortonormales

Entre todas las bases posibles de un espacio de Hilbert, las *bases ortonormales* juegan un papel fundamental, por esto se requiere de un método para ortonormalizar una base cualesquiera dada. Esto es lo que se conoce como proceso de *ortogonalización de Gram-Schmidt*.

#### Ortogonalización Gram-Schmidt

Dado un conjunto de vectores linealmente independientes  $\{f_k\}$  en  $V$ , se puede construir un *conjunto ortonormal*  $\{\varphi_k\}$  a partir del mismo conjunto generador  $\{f_k\}$  como sigue

Comenzar con

$$\varphi_1 = \frac{f_1}{\|f_1\|} \quad (1.42.1)$$

y obtener recursivamente

$$\varphi_k = \frac{f_k - g_k}{\|f_k - g_k\|}, \quad k = 2, 3, \dots, n \quad (1.42.2)$$

donde

$$g_k = \sum_{i=1}^{k-1} \langle f_k, \varphi_i \rangle \varphi_i \quad (1.42.3)$$

Se puede apreciar de la ecuación (1.42.3) que el vector  $g_k$  es la suma vectorial de las proyecciones perpendiculares de  $f_k$  sobre  $\varphi_i$ ,  $i = 1, 2, \dots, k-1$ ; de acuerdo a lo visto en la sección 1.3.3. En otras palabras, el vector  $g_k$  es la *proyección ortogonal* de  $f_k$  sobre el subespacio generado por los vectores previamente ortogonalizados, y este es sustraído de  $f_k$ , seguido de la normalización.

La idea básica consiste en formar uno a uno los vectores  $\varphi_k$  de tal manera que cada nuevo vector sea ortogonal a todos los vectores obtenidos anteriormente. De ésta manera, el conjunto ortonormal  $\{\varphi_i\}$  forma una *base ortonormal* de  $V$ .

#### Desigualdad de Bessel

Si se tiene un sistema ortonormal de vectores  $\{\varphi_k\}$  en  $V$ , entonces para cada  $f \in V$  la siguiente desigualdad, conocida como *desigualdad de Bessel*, se satisface

$$\|f\|^2 \geq \sum_k |\langle f, \varphi_k \rangle|^2 \quad (1.43)$$

Si se tiene un sistema ortonormal que es completo en  $V$ , entonces se tiene una base ortonormal en  $V$ , y la relación de Bessel se convierte en igualdad, frecuentemente llamada *igualdad de Parseval*.

En dimensiones finitas, esto es, en  $R^n$  o  $C^n$ , el tener un conjunto ortonormal de tamaño  $n$  es suficiente para tener una base ortonormal. Como se espera, esto es más delicado en dimensiones infinitas, es decir, no es suficiente tener un conjunto ortonormal infinito.

Los siguientes teoremas proporcionan sentencias equivalentes que permiten verificar si un sistema ortonormal es también una base.

Dado un sistema ortonormal  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$  en  $V$ , las siguientes propiedades son equivalentes:

a) El conjunto de vectores  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$  es una base ortonormal de  $V$  (1.44.1)

b) Si  $\langle f, \varphi_k \rangle = 0$ , para  $k = 1, 2, \dots$ , entonces  $f = 0$  (1.44.2)

c)  $span(\{\varphi_k\})$  es denso en  $V$ , es decir, cada vector en  $V$  es un límite de una secuencia de vectores en  $span(\{\varphi_k\})$  (1.44.3)

d) Para cada  $f \in V$ ,  $\|f\|^2 = \sum_k |\langle f, \varphi_k \rangle|^2$ , llamada *igualdad de Parseval* (1.44.4)

e) Para cada  $f_1, f_2 \in V$ ,  $\langle f_1, f_2 \rangle = \sum_k \langle f_1, \varphi_k \rangle \langle f_2, \varphi_k \rangle^*$ , llamada *fórmula de Parseval* o *igualdad de Parseval generalizada* (1.44.5)

Combinando (1.41) y (1.44.4) se tiene que

$$\|f\|^2 = \sum_k |c_k|^2 \tag{1.45}$$

que para el caso de señales y secuencias de energía en  $L_2(C)$  corresponde al *teorema de Parseval*, o conservación de energía entre la señal original y los *coeficientes de Fourier*  $c_k$ , ya que de acuerdo a las ecuaciones (1.17), la ecuación (1.45) representa la energía de la señal.

### 1 3.5. Proyecciones Ortogonales

Uno de los resultados más importantes de la teoría de los espacios vectoriales con producto interno es el llamado *teorema de proyección* (o *teorema de la mejor aproximación*), debido a que constituye el fundamento de diversos métodos de optimización.

Lo anterior se basa en el hecho de que frecuentemente un vector  $g$  de un espacio de Hilbert  $V$  tiene que ser aproximado mediante un vector que forma parte de un subespacio vectorial  $U$ , cerrado y convexo, es decir, en el que existe la propiedad del punto más cercano. Asumiendo que  $V$  es separable, esto es, que  $U$  contiene una *base ortonormal*  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$ , en donde  $n \rightarrow \infty$ .

El vector  $Pg \in U$ ,  $U \subset V$  y  $dim(U) < dim(V)$ , que minimiza la distancia  $d(g, Pg) = \|g - Pg\|$  sobre todos los vectores de  $U$  es la *proyección perpendicular* de  $g$  en  $U$ .

Refiriéndonos a la figura 4, la *proyección ortogonal* o *perpendicular* de un vector sobre otro se define como:

$$r = Pf_g = \text{proy}_g f = \langle f, \hat{g} \rangle \cdot g = \frac{\langle f, g \rangle}{\langle g, g \rangle} g = \text{comp}_g f \cdot g \quad (1.46)$$

De aquí que una proyección ortogonal de  $g \in V$  sobre el subespacio  $U$ , denotada por  $Pg = [Pg]_U$ ,  $Pg \in U$ , y  $U = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ ; y llamada *proyección de  $g$  sobre  $U$* , se define como

$$Pg = [Pg]_U = \sum_{k=1}^n \langle g, \varphi_k \rangle \cdot \varphi_k \quad (1.47)$$

Considérese para ello el subespacio de  $R^3 = V$  representado por el plano  $U$  de la figura 5, y un vector arbitrario  $g \in R^3$ .

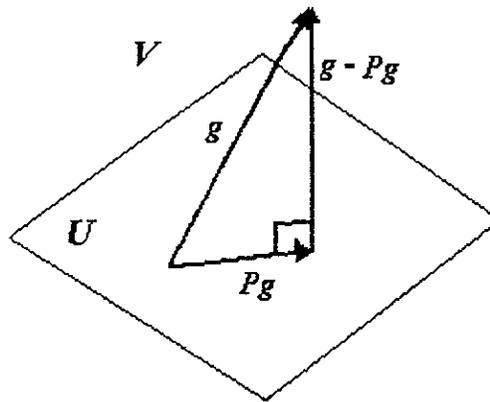


Figura 5. Proyección perpendicular  $Pg$  de un vector  $g$  sobre un subespacio  $U$

Se trata de encontrar un vector  $Pg$  del plano  $U$  que sea el “*más cercano*” a  $g$ , o el “*más aproximado*”, en el sentido de que la distancia entre  $g$  y  $Pg$  sea la menor distancia posible entre  $g$  y cualquier vector de  $U$ .

Para el problema geométrico planteado en  $R^3$ , la intuición sugiere que el vector  $Pg$  existe y es único, y que es precisamente la proyección ortogonal de  $g$  sobre el plano  $U$ , como se infiere en la figura 5.

### Teorema de Proyección

Sea  $V$  un espacio vectorial con producto interno y sea  $U$  un subespacio de  $V$ . Para cada vector  $g \in V$  existe uno y sólo un vector  $Pg \in U$  tal que

$$\|g - Pg\| < \|g - h\|, \quad \forall h \in U, h \neq Pg \quad (1.48)$$

dicho vector es la *proyección* de  $g$  sobre  $U$ .

De lo anterior, la diferencia  $d = g - Pg$  satisface que  $d \perp U$ , y en particular,  $d \perp Pg$ , así como el teorema de pitágoras  $\|g\|^2 = \|d\|^2 + \|Pg\|^2$ ,  $\|g\|^2 = \|g - Pg\|^2 + \|Pg\|^2$ .

Una propiedad importante de tal aproximación se debe a que es la mejor en el sentido de *minimos cuadrados*, esto es

$$\min \|g - h\| \tag{1.49}$$

con  $h \in U$  se tiene cuando  $h = \sum_k c_k \varphi_k$ , con  $c_k = \langle \varphi_k, g \rangle$ , esto es, los *coeficientes de Fourier*

### Producto Vectorial entre Vectores Proyectados

Del análisis anterior, éste se obtiene como:

$$\langle Pf, Pg \rangle = \langle Pf, g \rangle = \langle f, Pg \rangle \tag{1.50}$$

### 1.3.6. Expansiones Sobrecompletas (*Frames*)

Hasta aquí, hemos considerado expansión de señales sobre bases, es decir, los vectores utilizados en la expansión que son linealmente independientes. Sin embargo, se puede representar señales en términos de una combinación lineal de un *conjunto sobrecompleto o redundante* de vectores, donde los vectores ya no son independientes. Estos conjuntos de vectores sobrecompletos se denominan *frames*.

Una familia de funciones  $\{\varphi_k\}$  en un espacio de Hilbert  $V$  se llama “*frame*” si existen dos constantes  $A > 0$ ,  $B < \infty$ , tal que para cada  $f \in V$

$$A\|f\|^2 \leq \sum_{k=1}^n |\langle \varphi_k, f \rangle|^2 \leq B\|f\|^2 \tag{1.51}$$

$A, B$  se llaman *cotas del frame*, y cuando son iguales, se dice que el *frame* es “*angosto*” (*tight*). En un *tight frame* se tiene que

$$\sum_{k=1}^n |\langle \varphi_k, f \rangle|^2 = A\|f\|^2 \tag{1.52}$$

y la señal puede expandirse como

$$f = A^{-1} \sum_{k=1}^n \langle \varphi_k, f \rangle \varphi_k \tag{1.53}$$

Mientras que ésta última ecuación se parece a la fórmula de expansión en una base ortonormal, un *frame* no constituye una base ortonormal en general. En particular, los vectores pueden ser linealmente dependientes y de aquí que no formen una base. Si todos los vectores en un *tight frame* tienen norma unitaria, entonces la constante  $A$  proporciona la *relación de redundancia* (por ejemplo,  $A = 2$  significa hay dos veces el total de vectores necesarios para cubrir el espacio).

Notese que si  $A = B = 1$ , y  $\|\varphi_k\| = 1, \forall k$ , entonces  $\{\varphi_k\}$  constituye una base ortonormal.

Debido a la dependencia lineal que existe entre los vectores utilizados en la expansión, ésta ya no es única. Considérese el conjunto  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$  donde  $\sum_k \beta_k \varphi_k = 0$  (donde no todas las  $\beta_k$ 's son cero) debido a la dependencia lineal. Si  $f$  puede ser expresado como

$$f = \sum_k \alpha_k \varphi_k \quad (1.54)$$

entonces se puede añadir  $\beta_k$  a cada  $\alpha_k$  sin cambiar la validez de la expansión (1.54). La expansión (1.53) es única en el sentido de que minimiza la norma de la expansión entre todas las expansiones válidas.

## 1.4. Expansión de Señales

Hasta este punto se han caracterizado los espacios de Hilbert que son la base de los espacios de señales y sus expansiones. Veremos en forma resumida, retomando algunos conceptos de las secciones anteriores, lo relacionado a expansión de señales. Por simplificación, veremos el caso en una dimensión, lo cual se puede generalizar directamente a  $N$  dimensiones.

La razón de expandir señales, y específicamente señales discretas o secuencias, está ligada al análisis de señales, aproximación, codificación, y compresión, así como a sus algoritmos e implementaciones.

La representación de ésta expansión ésta dada por la ecuación (1.33) que combinada con la ecuación (1.41) obtenemos

$$f(t) = \sum_{k \in \mathbb{Z}} \langle \varphi_k, f \rangle \cdot \varphi_k(t) \quad (1.55.1)$$

$$f[n] = \sum_{k \in \mathbb{Z}} \langle \varphi_k, f \rangle \cdot \varphi_k[n] \quad (1.55.2)$$

donde  $f$  representa una señal continua  $f(t)$  o discreta  $f[n]$ ,  $t \in \mathbb{R}$ ,  $n \in \mathbb{Z}$ . De acuerdo a la expresión (1.55) se requiere construir un conjunto ortonormal de funciones base,  $\{\varphi_k\}$ , que sea completo en el espacio de las funciones cuadrado-integrables o secuencias cuadrado-sumables,  $L_2(\mathbb{R})$  o  $L_2(\mathbb{Z})$  respectivamente.

Debido a la necesidad de procesar una señal digitalmente, se requiere tener su representación discreta, de acuerdo a la expresión (1.4). Por esto el interés se centra en la expansión de señales discretas, es decir, en la representación de secuencias discretas. A su vez, las bases de tiempo discreto son más apropiadas para tareas de procesamiento de señales, las cuales deben tratar de

satisfacer dos requerimientos, propiamente dicho, obtener buena resolución frecuencia, y al mismo tiempo, conservar buena localización en el tiempo.

Adicionalmente, por razones prácticas y de cómputo, el conjunto de funciones base tiene que ser estructurado. Típicamente, el conjunto de funciones base  $\{\varphi_k\}$  se obtiene a través de un número finito de secuencias prototipo y sus versiones desplazadas en el tiempo.

Esto es lo que conduce a las implementaciones llamadas “*banco de filtros*” (*filter banks*) de tales expansiones estructuradas. A su vez, las expansiones deben ofrecer la propiedad de *aproximación sucesiva*. Esto significa que una “reconstrucción” basada en un subconjunto apropiado de funciones base conduce a una buena *aproximación* de la señal, la cual es una característica importante para aplicaciones tales como compresión de señales.

La *expansión ortonormal* de la señal  $f$  de la ecuación (1.55.2) puede describirse como

$$f[n] = \sum_{k \in \mathbb{Z}} \langle \varphi_k[l], f[l] \rangle \cdot \varphi_k[n] = \sum_{k \in \mathbb{Z}} F[k] \cdot \varphi_k[n] \quad (1.56)$$

donde

$$F[k] = \langle \varphi_k[l], f[l] \rangle = \sum_l \varphi_k^*[l] \cdot f[l] \quad (1.57)$$

La secuencia  $F[k]$  de la expresión (1.57) recibe los nombres de *transformada de la señal  $f[n]$*  o *análisis* de la señal.

La secuencia  $f[n]$  de la expresión (1.56) recibe los nombres de *transformada inversa de la señal  $F[k]$* , *expansión* de la señal, o *síntesis* de la señal.

Para señales continuas se tiene que

$$F[k] = \langle \varphi_k(t), f(t) \rangle = \int_t \varphi_k^*(t) f(t) dt \quad (1.58)$$

Las funciones base satisfacen la condición de ortonormalidad definida en la ecuación (1.28).

Una propiedad importante de las expansiones es la *conservación de energía*, dada por

$$\|f\|^2 = \|F\|^2 \quad (1.59)$$

que es una derivación del *teorema de Parseval*.

Capítulo 2  
Modelos del Sistema de  
Percepción Visual Humana

## Capítulo 2. Modelos del Sistema de Percepción Visual Humana

### 2.1. Introducción

La etapa final de casi cualquier sistema de procesamiento de imágenes y vídeo es el sistema visual humano, el cual tiene su interfase con el exterior a través del ojo. Por lo tanto es útil saber cómo y qué es lo que el ojo ve. Las respuestas a estas preguntas quedan fuera de nuestro contexto usual de ingeniería. Por esto, una comprensión básica de los procesos o mecanismos de percepción visual en el cerebro se presenta en forma breve con las interpretaciones en ingeniería para el procesamiento de imágenes y vídeo.

Lo que sigue es la presentación de ciertos aspectos del funcionamiento del sistema visual humano de acuerdo a lo que algunos investigadores han encontrado. Estos aspectos se ven de manera breve sin dar una descripción completa del sistema visual humano, para tener una idea general de los modelos asociados a él y su relación con el procesamiento digital de imágenes y vídeo.

Podemos dividir el análisis del *sistema visual* humano en:

- Estructura Anatómica
- Campos Receptivos Corticales
- Dos Modelos de los Campos Receptivos:
  - Funciones de Gabor
  - Funciones Hermite (Derivadas Gaussianas)

Cabe mencionar que los modelos mencionados son los más utilizados dentro de la codificación de imágenes y secuencias de ellas, pero esto no significa que sean los únicos modelos. Aún así, son los más acercados al procesamiento dentro de la corteza cerebral. Es interesante notar la similitud entre los campos receptivos y sus modelos, con las diversas máscaras utilizadas para la detección de bordes y líneas por operadores locales tales como los de Roberts, Sobel, Kirsch, Hueckel, y Mero-Vasy (Canny, 1986; Kunt *et al.*, 1994).

Este capítulo está basado principalmente en las publicaciones de *Jeffrey A Bloom* (1996, 1997). Específicamente las secciones 2.3 a 2.6.

### 2.2. Sistema Visual Primario

#### 2.2.1. Estructura del Ojo Humano

El ojo es el sensor de señales visuales. Se encarga de enfocarlas para formar imágenes en la retina. Ésta última analiza la imagen y envía el mensaje al cerebro a través del nervio óptico y caminos o *trazos ópticos* en la cabeza.

Las figura 6 (González & Woods, 1992) muestra una sección transversal del ojo humano.

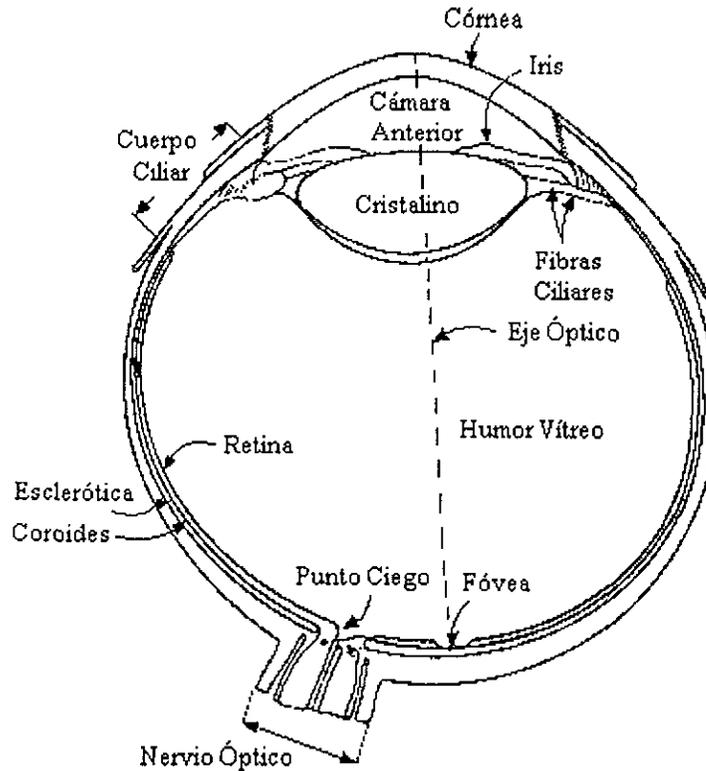


Figura 6. Sección transversal del ojo humano

El ojo es casi esférico, con un diámetro medio de 20mm- Está rodeado por tres membranas: la *córnea* y la *esclerótica*, que constituyen la cubierta exterior, la *coroides* y la *retina*. La *córnea* es un tejido resistente y transparente que cubre la superficie anterior del ojo. En prolongación de la *córnea*, la *esclerótica* es una membrana opaca que encierra el resto del globo ocular.

La *coroides* está inmediatamente debajo de la *esclerótica*. Esta membrana contiene una red de venas que constituyen la principal fuente de nutrición del ojo. La capa *coroides* está fuertemente pigmentada para ayudar a reducir la cantidad de luz exterior que entra en el ojo y la luz difundida en el interior del globo ocular. En su extremo anterior, la *coroides* está dividida en *cuerpo ciliar* y *diafragma* o *iris*. Este último se abre o se cierra para controlar la cantidad de luz que entra en el ojo. La abertura central del iris (la *pupila*) varía de diámetro desde unos 2 mm hasta unos 8 mm. La parte frontal del iris contiene el pigmento visible del ojo, mientras que la parte posterior contiene un pigmento negro.

El *cristalino* se encuentra formado por estratos concéntricos de células fibrosas y está suspendido por unas fibras que lo ligan al cuerpo ciliar. Contiene entre un 60% y un 70% de agua, un 6% de grasa y más proteínas que ningún otro tejido del ojo. El cristalino está colorado por una pigmentación amarillenta que va aumentando con la edad. Absorbe aproximadamente el 8% del espectro luminoso visible, con una absorción ligeramente superior en las longitudes de onda más cortas. Tanto la luz infrarroja como la luz ultravioleta son absorbidas de forma apreciable por las proteínas que forman la estructura del cristalino puesto que en cantidades excesivas pueden dañar el ojo.

La membrana más interna del ojo es la *retina*, que recubre la totalidad de la pared posterior. Cuando el foco está correctamente enfocado, la luz de un objeto exterior al ojo forma su imagen en la retina. La visión del objeto se debe a una distribución de receptores de luz repartidos por la superficie de la

retina. La retina transforma la luz proveniente del exterior, a través de sus *receptores*, en señales eléctricas que son transmitidas a la corteza visual a través del nervio óptico. Existen dos clases de *receptores*: los *conos* y los *bastones*.

En cada ojo existen entre 6 y 7 millones de *conos* localizados principalmente en la región central de la retina, denominada *fóvea*, y que son muy sensibles al color. Los seres humanos podemos apreciar detalles relativamente finos gracias a esos conos porque cada uno está conectado a su propia terminación nerviosa. Los músculos que controlan el ojo giran el globo ocular hasta que la imagen del objeto de interés queda en la *fóvea*. La visión mediante los conos se denomina *fotópica* o *visión de luz brillante*.

El número de *bastones* es mucho mayor: entre 75 y 150 millones están distribuidos sobre la superficie de la retina. Su mayor área de distribución, junto con el hecho de que grupos de varios bastones comparten una misma terminación nerviosa, reduce la cantidad de detalle discernible por estos receptores. Los bastones sirven para dar una visión general del campo de visión. No están implicados en la visión en color y son sensibles a niveles de iluminación bajos. Por ejemplo, los objetos que aparecen con colores brillantes bajo la luz del día, cuando se ven a la luz de la luna aparecen como formas sin color puesto que sólo se estimulan los bastones. Este fenómeno se conoce como *visión escotópica* o *visión tenue*.

La figura 7 (González & Woods, 1992) muestra la *densidad* o *distribución de bastones y conos* en la retina para una sección transversal del ojo derecho cruzando la región por donde emerge el nervio óptico desde el ojo. La ausencia de receptores en esta zona produce lo que se denomina el *punto ciego* (véase figura 6). Excepto en esta región, la distribución de receptores es radialmente simétrica alrededor de la *fóvea* (grados de separación del eje, medidos por el ángulo que forman el eje óptico y una línea que pase por el centro del cristalino. Hasta el punto de intersección con la retina. Obsérvese en la figura 7 que los conos son más densos en el centro de la retina (en el área de la *fóvea*). Obsérvese también que la intensidad de los bastones aumenta desde el centro hasta unos 20° fuera del eje y luego decrece conforme nos acercamos a la periferia de la retina.

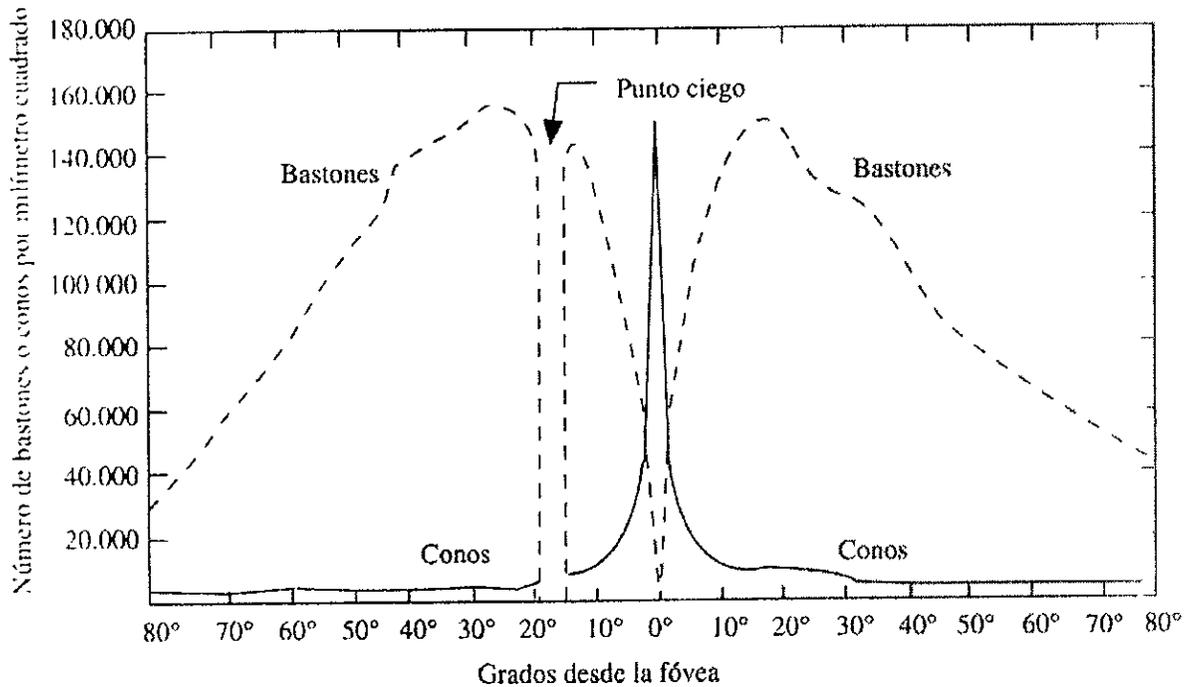


Figura 7. Distribución de bastones y conos en la retina

### 2.2.2. Camino Visual Primario

A continuación se muestra en la figura 8 (Kunt *et al.*, 1994) un esquema del *Camino Visual Primario* (PVP, por sus siglas en inglés, "*Primary Visual Path*") visto desde un corte inferior de la cabeza. Como se aprecia, se trata de un sistema bastante complejo que transforma la información visual, que es información espacio-temporal y de intensidad luminosa o variaciones electromagnéticas, en información codificada a manera de impulso electroquímicos para su procesamiento por las neuronas de la corteza visual. El camino por el que viaja ésta información tiene como entrada el ojo, descrito anteriormente, y a través de los nervios ópticos y trazo óptico tiene como destino las células corticales ubicadas en la parte posterior de la cabeza, en la corteza visual.

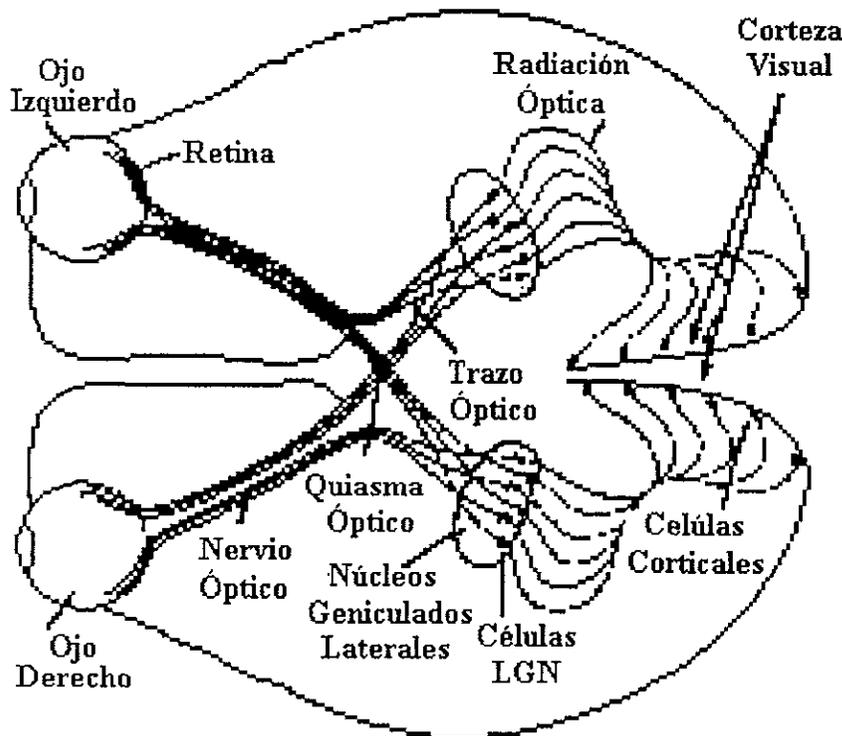


Figura 8. Camino visual primario (PVP) del sistema visual humano

Resumiendo la manera de transmisión de ésta información, tenemos que

- Primeramente, la luz entra en el ojo y es enfocada por una lente ubicada en la retina conocida como fovea. Allí, el tipo de células denominadas conos, convierten la luz en potenciales eléctricos. Como aproximación simple, estas señales pasan a través de un número de etapas en la retina: la salida de los conos es transferida a los bastones, que son células bipolares, y cuya salida se pasa a las células gangliares. La respuesta de las células gangliares toma la forma de un tren de impulsos de alta frecuencia y la información en la señal es codificada. Las células gangliares llevan la señal proveniente del nervio óptico hacia los *Núcleos Geniculados Laterales* (LGN, por sus siglas en inglés, "*Lateral Geniculate Nucleus*"), y finalmente pasan de aquí a un área del cerebro conocida como *Corteza Visual*, o más específicamente, *V1*, la cual se localiza en la parte posterior del cerebro.
- Aproximadamente el 50% de las células en *V1* son células simples. Estas células tienen una respuesta espacial lineal a la luz y reciben virtualmente toda la información de la retina.
- La respuesta de estas células es dependiente de la localización espacial de los estímulos de luz. Los investigadores han hecho un mapeo de las respuestas de estas células, a dichos estímulos de luz, a distintas posiciones del campo visual. El mapeo en 2D de estas respuestas para una sola célula cortical simple se llama "*Perfil del Campo Receptivo*" y representa la respuesta del PVP entero.

## 2.3. Campos Receptivos Corticales

Como se mencionó, la respuesta de las células V1 a la información visual reciben el nombre de *perfil del campo receptivo*, o bien, *campo receptivo cortical*. A continuación se muestran en la figura 9 tres *campos receptivos* representativos. Estos diagramas fueron tomados del trabajo de Daugman (1980, 1983, 1988) y Pollen (1981).



Figura 9. Perfiles de los campos receptivos corticales

Mientras que todos los campos receptivos de las células corticales simples de V1 difieren entre ellos, tienen algunas características comunes, por ejemplo, éstos son elípticos. Debido a la conexión en paralelo y a la retroalimentación entre las células corticales, un campo receptivo se divide en regiones. Algunos de ellos son de excitación, mientras que otros son de inhibición. Es decir, los perfiles de campo receptivo consisten en regiones de excitación e inhibición como bandas decaentes, orientadas y localizadas espacialmente.

Los campos receptivos se pueden caracterizar por:

- **Actividad Local:** Se muestran en estos diagramas solamente las partes de los perfiles donde está ocurriendo la actividad. Se observa que la actividad cae por debajo de un nivel nominal en los bordes de estos diagramas. Ese nivel nominal de la actividad se extiende hacia fuera del campo visual entero. Dos células pueden tener un perfil de campo receptivo similar y que difieren solamente en la región del campo visual donde ocurre la respuesta activa.
- **Bandas orientadas que decaen:** Los campos receptivos decaen desde sus centros. Experimentos con barras orientadas y rejillas senoidales sugieren que estas células actúan como filtros pasobajas con un ancho de banda aproximadamente de 1.5 octavas y orientación de aproximadamente 45 grados.

## 2.4. Modelo del Sistema Visual

Las características del PVP y del V1, así como de los campos receptivos, se pueden resumir en un diagrama de bloques, tal y como se aprecia en la figura 10 (Kunt *et al.*, 1994), donde se indican las partes relacionadas al cristalino, la retina, y la corteza.

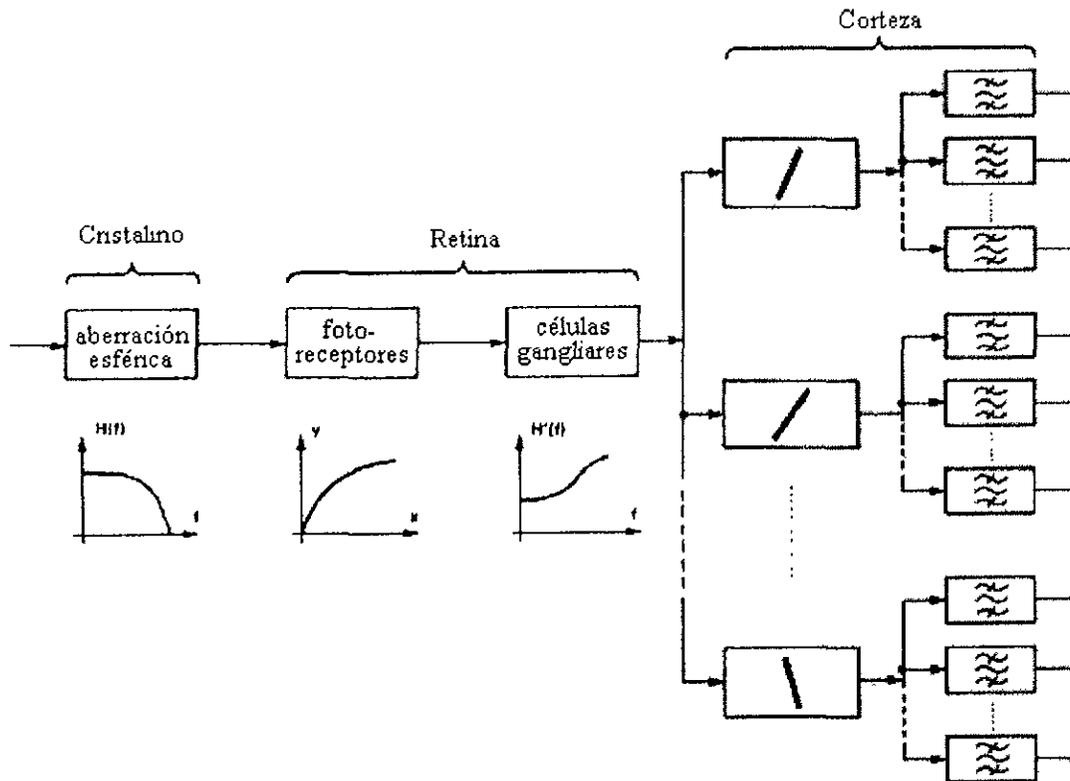


Figura 10. Diagrama de bloques del sistema visual humano

Las barras dentro de los rectángulos indican los *filtros direccionales* seguidos por otro banco de filtros para detectar la intensidad de los estímulos.

El primer bloque es un filtro espacial, paso-bajas, e isotrópico. Representa la aberración esférica del cristalino, el efecto de la pupila, y la limitación en frecuencia debido al número finito de fotoreceptores (conos y bastones). Este filtro es seguido de las características no lineales de los fotoreceptores. Aquí puede utilizarse una curva logarítmica por simplicidad o una ley del tipo  $L^{\gamma}$  para mayor precisión.

En la etapa de la retina, la transformación lineal anterior, y muy probablemente sin memoria, es seguida de un filtro isotrópico paso-altas correspondiente al fenómeno de *inhibición lateral* de las células gangliares. El proceso hecho por las células LGN del *núcleo geniculado lateral* puede estar incluido en este filtro paso-altas.

Finalmente, hay un banco de *filtros direccionales* que representan el procesamiento realizado por las *células corticales*. Debe notarse que el sistema entero es variante respecto al desplazamiento debido al decrecimiento en resolución lejos de la fovea.

## 2.5. Modelos de los Campos Receptivos

### 2.5.1. Modelo de Gabor

El modelo de campos receptivos corticales con la función de Gabor fue propuesto primeramente por Marčelja (1980), sin embargo, las funciones por sí mismas son acreditadas frecuentemente a Gabor (1946) que apoyó su uso en el área de comunicaciones.

- Las funciones de Gabor son funciones de variable compleja.
- En 1D pueden ser caracterizadas como el producto de una exponencial compleja y una Gaussiana.

$$g(x) = e^{j(ax+\theta)} e^{-\left(\frac{x^2}{2\sigma^2}\right)} \quad (2.1)$$

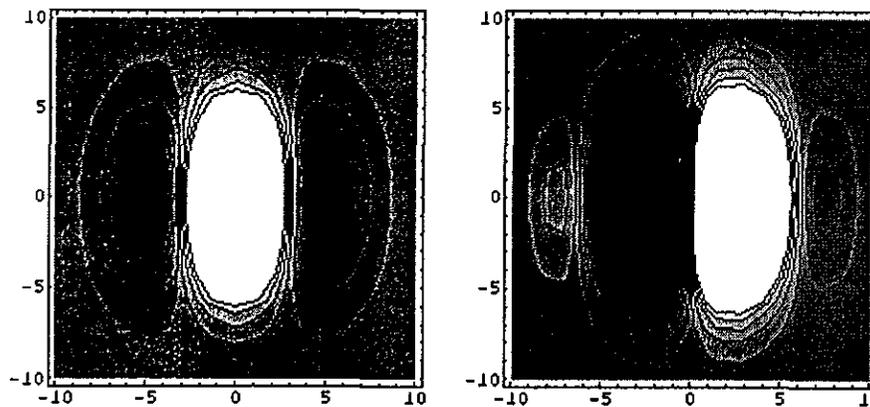


Figura 11. Funciones de Gabor 2D (parte real e imaginaria)

Los diagramas anteriores de la figura 11 (Bloom & Reed, 1996) corresponden a la parte real y a la parte imaginaria (izquierda y derecha respectivamente) de una función 2D compuesta de una *función de Gabor*, ecuación (2.1), a lo largo del eje  $x$  y de una Gaussiana a lo largo del eje  $y$ . Al igual que los campos receptivos, estas funciones tienen cierta actividad nominal (gris) la cual se incrementa (blanco) y se decreta (negro) en bandas locales orientadas.

### Funciones de Gabor

Estas son de la forma de la ecuación (2.1). Para el caso de  $N$  dimensiones, la variable independiente se transforma en un vector de  $N$  dimensiones, el cual genera una función separable de  $N$  variables.

- Las funciones de Gabor se pueden rotar en 2D, y su ancho de banda es selectivo.
- Las funciones de Gabor son usadas ampliamente como filtros y como funciones base de transformadas.
- Estas funciones optimizan la medida particular de localización:

$$U = (\Delta x)(\Delta \omega) \quad (2.2)$$

donde  $(\Delta x)^2$  y  $(\Delta \omega)^2$  son los segundos momentos normalizados de la densidad de energía

### 2.5.2. Modelo de Derivadas Gaussianas

El modelo de derivadas Gaussianas de campos receptivos corticales fue propuesto por primera vez por Young (1985, 1986, 1987).

- Las funciones de derivadas Gaussianas son funciones de variable real.
- En 1D pueden ser caracterizadas como el producto de un polinomio (polinomio generalizado de Hermite) y de una Gaussiana.

$$g(x) = \frac{d^n}{dx^n} e^{-\frac{x^2}{2\sigma^2}} = P_{n,\sigma}(x) e^{-\frac{x^2}{2\sigma^2}} \quad (2.3)$$

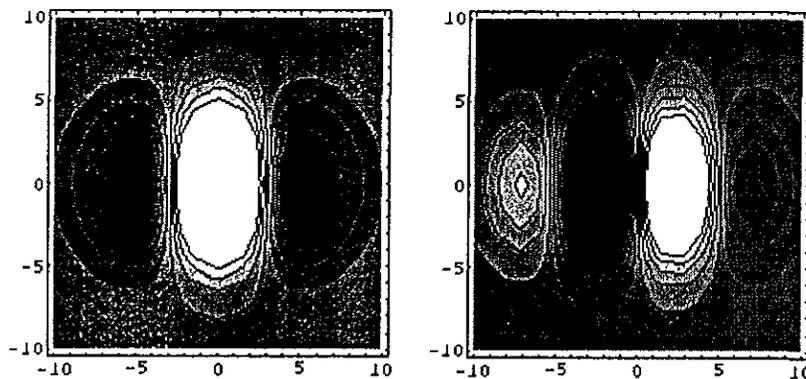


Figura 12. Funciones de derivadas Gaussianas 2D (segunda y tercera derivadas)

Los diagramas anteriores de la figura 12 (Bloom & Reed, 1996) corresponden a la segunda y tercera derivada de una Gaussiana (izquierda y derecha respectivamente) de una función 2D compuesta de las *funciones de derivadas Gaussianas*, ecuación (2.3), a lo largo del eje  $x$  y de una Gaussiana a lo largo del eje  $y$ . Al igual que los campos receptivos, estas funciones tienen cierta actividad nominal (gris) la cual se incrementa (blanco) y se decrementa (negro) en bandas locales orientadas.

### Funciones de Derivadas Gaussianas

Estas son de la forma de la ecuación (2.3). Para el caso de  $N$  dimensiones, la variable independiente se transforma en un vector de  $N$  dimensiones, el cual genera una función separable de  $N$  variables.

- Las derivadas de Gaussianas también se han utilizado para modelar las respuestas de neuronas sensibles a la disparidad.
- Las funciones de derivadas Gaussianas se pueden rotar en 2D, y su ancho de banda espectral es selectivo.
- La primera derivada es el bien conocido *detector de borde*.
- Los derivadas Gaussianas se han utilizado recientemente en un conjunto de aplicaciones como:

- Detección y Realce de bordes.
- Reducción de ruido.
- Detección de distorsiones radiadas en mamografías.
- Estimación de flujo óptico.
- Reconocimiento de rostros.

### Similitudes entre ambos modelos

Si comparamos las ecuaciones (2.1) y (2.2) se observa que ambos modelos son similares ya que contemplan una *función Gaussiana modulada (multiplicada)* por una función exponencial. Para el caso del modelo de Gabor esta función exponencial es de variable compleja, mientras que para el caso del modelo de derivadas Gaussianas la función exponencial es de variable real.

Ambos modelos son similares y corresponden a los modelos de los *campos receptivos corticales*. La analogía entre ambos modelos se aprecia en las figuras 11 y 12.

## 2.6. Funciones Hermite

### 2.6.1. Principio de Incertidumbre de Heisenberg

En esta parte plantearemos las funciones Hermite y su característica de ser un buen modelo de los campos receptivos corticales. A su vez, esta parte está basada en la interpretación siguiente del principio de incertidumbre de Heisenberg:

**“Una función no puede ser arbitrariamente local (*resolución*) en los dominios espacial y frecuencia–espacial simultáneamente.”**

Las funciones de interés son funciones base de algunas transformadas. El interés principal está en las transformadas locales, es decir, aquellas en que su localización y contenido espectral para un evento de la imagen se pueden obtener con un alto grado de certeza a partir de los coeficientes de la transformada. Esta localización (*resolución*) se logra utilizando funciones base, las cuales son locales en ambos dominios: espacial y frecuencia–espacial.

### 2.6.2. Localización (*resolución*)

En la figura 13 (Bloom & Reed, 1996) se aprecia a la izquierda la tercer derivada de una Gaussiana,  $g(x)$ , y su transformada de Fourier,  $G(\omega)$ , a la derecha.

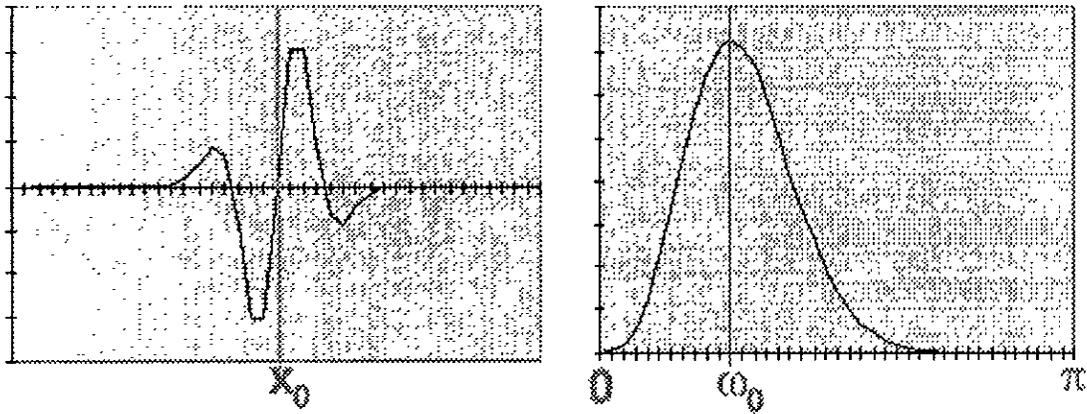


Figura 13. Tercera derivada de una Gaussiana y su transformada de Fourier

Esta función se considera local porque llega a ser arbitrariamente pequeña a cierta distancia lejos de la localización de su centro en ambos dominios (espacial y frecuencia-espacial). Lo anterior se puede expresar como:

$$|g(x - x_0)| < \varepsilon_x \quad |x - x_0| > \delta_x \quad (2.4.1)$$

$$|G(\omega - \omega_0)| < \varepsilon_\omega \quad |\omega - \omega_0| > \delta_\omega \quad (2.4.2)$$

La *localización* también se conoce como *resolución*. Una medida de esta característica es la definida por la ecuación (2.2) y la cual se deriva del principio de Heisenberg de la sección 2.5.1.

En otras palabras, la *localización* o *resolución* de una señal aumenta si en ambos dominios tiende a ser de soporte finito, es decir, si su extensión (duración) tiende a ser cero fuera del origen. Esto es lo que se conoce como buena resolución en el dominio de la señal y el dominio de su transformada de Fourier.

### 2.6.3. Métrica de Incertidumbre Conjunta

Una métrica de incertidumbre usada comúnmente, atribuida a menudo a Gabor, es el producto de dos intervalos efectivos:

$$U = (\Delta x)(\Delta \omega) \quad (2.5)$$

donde estos intervalos corresponden a la varianza normalizada de la densidad de energía en los dominios espacial  $(\Delta x)^2$  y frecuencia-espacial  $(\Delta \omega)^2$ ; en otras palabras, son los segundos momentos normalizados de la densidad de energía, dados por

$$(\Delta x)^2 = \frac{\int (x - x_0)^2 |g(x)|^2 dx}{\int |g(x)|^2 dx} \quad (2.6.1)$$

$$(\Delta\omega)^2 = \frac{\int (\omega - \omega_0)^2 |G(\omega)|^2 d\omega}{\int |G(\omega)|^2 d\omega} \quad (2.6.2)$$

Este producto tiene un límite inferior, según lo sugerido por Heisenberg:

$$U \geq 0.5 \quad (2.7)$$

Esta *métrica de incertidumbre* fundamenta el *principio de Heisenberg*, y es una métrica de *localización*, es decir, de *resolución*, entre dos dominios los cuales corresponden, en el caso de imágenes, a los dominios espacial y frecuencia-espacial.

Gabor demostró en 1946 que las funciones complejas o *funciones de Gabor*, definidas en la ecuación (2.1), alcanzan el límite inferior del producto de incertidumbre

#### 2.6.4. Funciones y Polinomios de Hermite

Gabor sugirió, pero no demostró, que las funciones de variable real como las de la ecuación (2.3), las funciones Hermite, tienen el producto de incertidumbre conjunto más bajo. Estas funciones se definen como

$$g_n(x) = H_n(x) e^{-\frac{1}{2}x^2} \quad (2.8.1)$$

$$H_n(x) \equiv (-1)^n e^{x^2} \frac{d^n}{dx^n} e^{-x^2} \quad (2.8.2)$$

donde  $g_n(x)$  son las *funciones Hermite* y  $H_n(x)$  son los llamados *polinomios de Hermite*.

Stork y Wilson hicieron una propuesta a través del *cálculo de variaciones* para encontrar las funciones de variable real que minimizan el producto de incertidumbre conjunto. Concluyeron que las soluciones son las funciones de derivadas Gaussianas, definidas en la ecuación (2.3).

Como se observa en la ecuación (2.8.1), las *funciones Hermite* se obtienen multiplicando los *polinomios de Hermite* de la ecuación (2.8.2) por una Gaussiana. Nótese que las funciones Hermite no son derivadas Gaussianas:

$$g_n(x) \neq \frac{d^n}{dx^n} e^{-x^2} \quad (2.9)$$

pero están *directamente relacionadas*, por lo que las *funciones Hermite* son un isomorfismo de las *derivadas Gaussianas*, es decir, es un modelo que se aproxima a ellas, y por lo tanto, este modelo se aproxima al de los campos receptivos corticales definido por Young.

## Minimización del Producto de Incertidumbre Conjunta

Retomando las ecuaciones (2.6.1) y (2.6.2), el producto de incertidumbre conjunta,  $U$ , se reduce al mínimo considerando:

- Sea  $x_0 = 0$ .
- El espectro de funciones reales es simétrico, entonces  $\omega = 0$ .
- Los denominadores es la energía de la función.
- En el numerador del producto,  $U^2 = (\Delta x)^2 (\Delta \omega)^2$ , tenemos el producto de dos integrales, de las cuales sus integrandos son cuadrados. Con esto, podemos utilizar la desigualdad de Schwartz para encontrar un límite superior.

Con esto obtenemos que la incertidumbre conjunta viene dada por

$$U^2 = \frac{\int x^2 g^2(x) dx \int \omega^2 |G(\omega)|^2 d\omega}{\left[ \int g^2(x) dx \right]^2} \quad (2.10)$$

## Propiedades de los Polinomios de Hermite

Una *función generalizada de Hermite* se define en el espacio y en la frecuencia espacial como

$$g(x) = (-1)^n H_{n,\sigma}(x) e^{-\frac{x^2}{4\sigma^2}} \quad (2.11.1)$$

$$G(\omega) = (-j)^n \sigma \sqrt{2} H_{n,\sigma}(2\sigma^2 \omega) e^{-\sigma^2 \omega^2} \quad (2.11.2)$$

Donde  $H_{n,\sigma}(x)$  son los *polinomios generalizados de Hermite*, definidos por la ecuación (2.8.2), para los cuales se cumple la relación siguiente

$$H_{n,\sigma}(x) = \left( \frac{1}{\sigma\sqrt{2}} \right)^n H_{n,\sigma} \left( \frac{x}{\sigma\sqrt{2}} \right) \quad (2.12)$$

Además de la relación anterior, los *polinomios generalizados de Hermite* de grado  $n$ , se pueden calcular recursivamente, ya que se satisfacen las relaciones

$$H_{n,\sigma}(x) = (-1)^n H_{n,\sigma}(-x) \quad (2.13.1)$$

$$H_{0,\sigma}(x) = 1 \quad (2.13.2)$$

$$H_{1,\sigma}(x) = 2x \quad (2.13.3)$$

$$H_{n+2,\sigma}(x) = 2xH_{n+1,\sigma}(x) - 2(n+1)H_{n,\sigma}(x) \quad , \quad n \geq 0 \quad (2.13.4)$$

La ecuación (2.13.1) nos indica que los polinomios de Hermite son *simétricos* respecto al origen. El tipo de simetría, par o impar, depende del orden del polinomio.

### Evaluación Directa de la Incertidumbre Conjunta

A falta de un resultado acertado que especifique las funciones reales que alcanzan el producto de incertidumbre conjunto más pequeño, conviene evaluar la incertidumbre directamente tanto para las *funciones Hermite* como para las *derivadas Gaussianas*.

Substituciones de variables y la aplicación de las propiedades de la transformada de Fourier permiten que las integrales en la ecuación (2.10), que expresa el producto de incertidumbre conjunto, puedan ser expresadas en otras formas conocidas. Combinando la ecuación (2.10) con las ecuaciones (2.11.1) y (2.11.2) obtenemos que

$$U^2_{HF} = \left( n + \frac{1}{2} \right)^2 \quad (2.14)$$

donde el prefijo *HF* se utiliza para indicar que se trata de las *funciones Hermite*.

Se observa que el producto de incertidumbre conjunto aumenta conforme *n* crece.

Escribiendo de nuevo la función de derivadas Gaussianas de la ecuación (2.3) y obteniendo su espectro o transformada de Fourier, tenemos que

$$g_n(x) = \frac{d^n}{dx^n} e^{-\frac{x^2}{2\sigma^2}} \quad (2.15.1)$$

$$G_n(\omega) = (-j\omega)^n \sigma e^{-\omega^2\sigma^2/2} \quad (2.15.2)$$

Obteniendo el producto de incertidumbre conjunto para las funciones de derivadas Gaussianas a través de la ecuación (2.10) tenemos que

$$U^2_{DG} = \frac{8n^2 + 2n - 1}{8n - 4} \quad (2.16)$$

donde el prefijo *DG* se utiliza para indicar que se trata de las *funciones de derivadas Gaussianas*.

Se observa que el producto de incertidumbre conjunto aumenta con la raíz cuadrada de *n*.

### Comparación de la Incertidumbre Conjunta

En la gráfica de la figura 14 se tiene la comparación entre la incertidumbre conjunta de las *funciones Hermite*,  $U_{HF}$  - ecuación (2.14) -, y de las *funciones de derivadas Gaussianas*,  $U_{DF}$  - ecuación (2.16) -. Los números en el eje *x* representan el orden del polinomio (HF) ó de la derivada

(DG) y el eje  $y$  es el producto de incertidumbre conjunta. Para  $n > 1$ , las funciones de derivadas Gaussianas tienen un producto de incertidumbre conjunta menor que el de las funciones de Hermite.

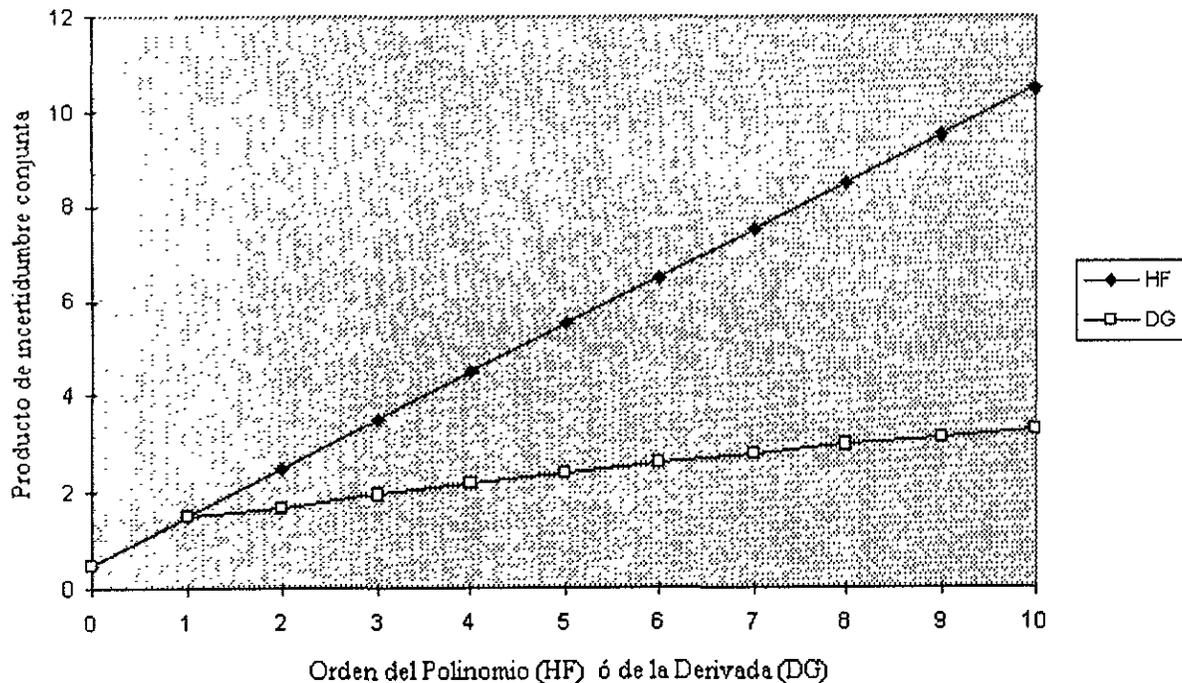


Figura 14. Comparación entre incertidumbres conjuntas (HF y DG)

En conclusión, las funciones de Hermite y las derivadas Gaussianas no son óptimas, pero las derivadas Gaussianas tienen mejor localización (resolución) que las funciones de Hermite. De cualquier manera, las funciones Hermite son una buena aproximación de las derivadas Gaussianas que modelan los campos receptivos corticales.

## 2.7. Modelo basado en la Transformada Polinomial

Una nueva propuesta para la descripción de imágenes y vídeo, llamada *transformada Polinomial*, fue desarrollada por Martens (1990). Esta descripción se basa en un análisis local de las señales (imágenes estáticas o secuencias de imágenes) dentro de ventanas traslapadas. Dentro de cada ventana la señal es expandida ortogonalmente en una suma de polinomios.

De aquí, contrario a la transformada de Gabor, que adopta un análisis en frecuencia de la señal de acuerdo a la ecuación (2.1), la transformada polinomial adopta un análisis espacial.

La consecuencia principal de esta propuesta alternativa es que las propiedades geométricas de las características de una imagen o secuencia de ellas, pueden ser determinadas fácilmente, tal es el caso de posición, orientación, curvatura, flujo, velocidad, etc.

La transformada Polinomial es también computacionalmente eficiente, y su versión discreta se puede implementar eficientemente en tiempo real a través de hardware dedicado. Además, la transformada Polinomial tiene diversas aplicaciones en codificación de imágenes, reducción de

ruido, restauración de difuminación, y otras aplicaciones, tal y como ha sido mostrado por Martens y otros investigadores.

Adicionalmente, estructuras piramidales también pueden ser usadas para construir representaciones multiresolución de imágenes o secuencias basadas en ésta transformada. A su vez, debido a su separabilidad en  $N$  dimensiones, se facilita su implementación y la posibilidad de proyectar una transformada de  $N$  dimensiones en una dimensión. La transformada polinomial será explicada en mayor detalle en el capítulo 3, ya que es la base de la codificación de secuencias de imágenes a través de proyecciones en una dimensión, presentada en el capítulo 4.

### 2.7.1. Transformada Hermite

Un caso especial interesante, desde el punto de vista de percepción visual, es la *transformada Hermite* (Martens, 1990a, b, c), la cual se obtiene si se aplica una transformada polinomial con una ventana Gaussiana. En el caso de la transformada Hermite, la imagen o secuencia necesita ser analizada por filtros que son *derivadas de Gaussianas*. Estos filtros, como se vio anteriormente, son parte del modelo del sistema de visión humana y específicamente de los campos corticales (Young, 1985, 1987). A su vez, son extensamente utilizados como detectores eficientes de características importantes de una imagen, como son bordes y líneas.

### 2.7.2. Transformada de Laguerre

Como se mencionó, la transformada Hermite tiene filtros de análisis que son derivadas de Gaussianas, por lo que concuerdan con el modelo de campos receptivos. A esto se debe su aplicación de codificación y procesamiento tanto para los patrones espaciales (imágenes) como para las imágenes dependientes del tiempo (secuencias de imágenes) (Martens, 1990a). Sin embargo, en el *dominio temporal* todo procesamiento debe adherirse a la *condición de causalidad*. En este aspecto, el procesamiento espacial y temporal difieren fundamentalmente. Por esto, la transformada Hermite no es compatible con la condición de causalidad debido a la naturaleza de sus filtros (Brinker & Roufs, 1992).

La *transformada de Laguerre* generalizada es una transformada polinomial con una ventana Gamma, la cual conduce a los polinomios ortogonales de Laguerre para la codificación de la señal. A su vez, desde el punto de vista neurofisiológico, es una descripción del procesamiento a estímulos temporales dentro del sistema visual humano en su etapa primaria (Brinker & Roufs, 1992).

Con la transformada de Laguerre aplicada a la dimensión temporal (para construir mecanismos sensibles a la velocidad) en combinación con la transformada Hermite aplicada a la dimensión espacial, es posible tener un sistema de codificación y procesamiento completo de secuencias de imágenes que representa un modelo de los campos receptivos corticales dentro del sistema visual humano (Brinker & Roufs, 1992).

# Capítulo 3

## La Transformada Polinomial

## Capítulo 3. La Transformada Polinomial

### 3.1. Introducción

En éste capítulo se muestra una técnica de descomposición de señales llamada *Transformada Polinomial* (Martens, 1990a, 1990b), en la cual las señales se aproximan por polinomios. Primero se introducen las ideas para señales analógicas de una dimensión, pero que pueden extenderse a señales de dos (imágenes), tres (vídeo) o más dimensiones, y al caso de señales discretas en diversas dimensiones. Después se introducen las definiciones para un caso de transformada polinomial de particular importancia: la *transformada Hermite*.

Como se verá más adelante, y retomando los conceptos de los capítulos anteriores, una transformada polinomial es una descomposición local de la señal en funciones ortogonales representadas por polinomios. De acuerdo a los conceptos de expansión de señales y ortogonalidad en los espacios de Hilbert, la *base* de expansión de la transformada polinomial está definida por el producto interno de polinomios de grado  $n$ , con  $n \rightarrow \infty$  para espacios de dimensión infinita, de acuerdo a las ecuaciones (1.14.1) y (1.14.2), que son ortogonales respecto a una *función de peso* o *ventana*,  $w^2(x)$ . Según sea el tipo de *ventana*, será el tipo de transformada. La ventana permite hacer el análisis local de la señal, es decir, aproximar la señal que está “dentro” de la ventana por los polinomios que son ortogonales a la misma.

Como se vio anteriormente en el capítulo 2, los campos receptivos corticales son parte del sistema visual primario, por lo que el análisis de imágenes y vídeo, que son las señales procesadas por el sistema visual humano, se realiza de manera eficiente si utilizamos los modelos de los campos receptivos, los cuales corresponden a derivadas de Gaussianas. De acuerdo a esto y a los modelos del capítulo 2, la transformada Hermite y la transformada de Laguerre son dos modelos de descomposición de señales que están estrechamente relacionados con el sistema visual primario. Para el caso de la transformada Hermite, la función de peso,  $w^2(x)$ , corresponde a una ventana Gaussiana, mientras que en el caso de la transformada de Laguerre corresponde a una ventana Gamma. En ambos caso, los polinomios ortogonales resultantes son los polinomios de Hermite y los polinomios de Laguerre, respectivamente.

### 3.2. Transformadas Polinomiales Unidimensionales

El análisis por transformada polinomial abarca dos etapas a seguir (Martens, 1990a). En la primera etapa, la señal se localiza multiplicándola por una ventana, y en la segunda etapa, se aproxima el segmento de la señal a través de una combinación lineal (suma ponderada) de la base formada por los polinomios ortogonales a la ventana por la que se multiplicó la señal. Es decir, se hace la proyección ortogonal de la señal sobre la base de polinomios. Estas etapas se describen detalladamente a continuación.

En la primera etapa, la señal original  $L(x)$  se localiza multiplicándola por una función ventana  $V(x)$ . Una descripción completa de la señal requiere que el proceso de localización se repita para un número suficiente de posiciones de la ventana. Se requiere un espaciamiento  $T$  constante entre las ventanas.

A partir de la localización de la función ventana  $V(x)$ , se puede construir una *función de peso*  $W(x)$  por medio de su repetición periódica.

$$W(x) = \sum_k V(x - kT) \quad (3.1)$$

Por lo tanto, la propia función de peso es periódica al definirse con el periodo de espaciamento  $T$ . Si se considera que la función de peso  $W(x)$  existe con valores diferentes de cero para toda  $x$ , es posible obtener

$$L(x) = \frac{1}{W(x)} \sum_k L(x) \cdot V(x - kT) \quad (3.2)$$

de tal manera que se garantice la localización de las señales  $L(x) \cdot V(x - kT)$  en todas las posiciones  $kT$  de la ventana y que en cada posición se contenga suficiente información acerca de la señal original.

La segunda etapa de la transformación polinomial consiste en aproximar el segmento de señal definido dentro de la ventana  $V(x - kT)$  por medio de una suma ponderada de polinomios. Como *funciones base* para la expansión polinomial se toman los polinomios  $G_n(x)$ , con grado del polinomio dado como  $[G_n(x)] = n$ , y que son ortonormales con respecto a  $V^2(x)$ , esto es,

$$\int_{-\infty}^{+\infty} V^2(x) G_m(x) G_n(x) dx = \delta_{mn} \quad (3.3)$$

Estos polinomios están determinados de manera única por  $V^2(x)$ . Si  $V^2(x)$  es función par, entonces se tienen los siguientes ejemplos de polinomios ortonormales (Boas & Buck, 1985) hasta de orden 3.

$$G_0(x) = \frac{1}{\sqrt{c_0}} \quad (3.4.1)$$

$$G_1(x) = \frac{x}{\sqrt{c_2}} \quad (3.4.2)$$

$$G_2(x) = \frac{(c_0 x^2 - c_2)}{\sqrt{c_0(c_0 c_4 - c_2^2)}} \quad (3.4.3)$$

$$G_3(x) = \frac{(c_2 x^3 - c_4 x)}{\sqrt{c_2(c_2 c_6 - c_4^2)}} \quad (3.4.4)$$

en donde

$$c_n = \int_{-\infty}^{+\infty} x^n V^2(x) dx \quad (3.5)$$

es el momento de orden  $n$  de  $V^2(x)$ .

En condiciones más generales, para la señal original  $L(x)$ , se obtiene (Boas & Buck, 1985)

$$V(x - kT) \left[ L(x) - \sum_{n=0}^{\infty} L_n(kT) \cdot G_n(x - kT) \right] = 0 \quad (3.6)$$

la cual es la expansión de la señal local a la ventana, es decir, la señal localizada  $L(x) \cdot V(x - kT)$ . Esta expresión concuerda con la de las ecuaciones (1.1) y (1.55.1).

donde

$$L_n(kT) = \int_{-\infty}^{+\infty} L(x) \cdot G_n(x - kT) V^2(x - kT) dx \quad (3.7)$$

son los coeficientes de la expansión (*coordenadas*), de acuerdo a las ecuaciones (1.55.1) y (1.58).

Si  $L(x)$  se puede descomponer en términos  $L_n(kT)$  como los de la ecuación (3.7), y es finita para toda  $x$ , se garantiza la convergencia de la expansión en series de la ecuación (3.6) para la mayoría de las funciones ventana. Por lo tanto, el error de aproximación entre la señal y un polinomio es menor conforme mayor sea el grado de la expansión polinomial, como ocurre en las expansiones de Taylor, es decir, una mejor aproximación se tiene cuando  $n \rightarrow \infty$ , lo cual es congruente con los espacios de dimensión infinita. Esto implica que la descripción de la señal localizada  $L(x) \cdot V(x - kT)$  contiene una aproximación de error arbitrariamente pequeño para un conjunto finito de coeficientes polinomiales  $L_n(kT)$ . La señal de energía dentro de la ventana se expresa en términos de los coeficientes (*coordenadas*) de la expansión, esto es,

$$E_{L \cdot V} = \int_{-\infty}^{+\infty} L^2(x) V^2(x - kT) dx = \sum_{n=0}^{\infty} L_n^2(kT) \quad (3.8)$$

Esta es la aplicación del *teorema de Parseval* para polinomios ortonormales sobre la señal local  $L \cdot V$ , de acuerdo a las ecuaciones (1.2), (1.17.2), (1.44.4), (1.45) y (1.59), las cuales establecen una relación de la energía en los dos dominios: el de la señal y el de los coeficientes de la expansión.

Combinando (3.2) y (3.6) se obtiene la siguiente expansión para la señal *completa* (no local):

$$L(x) = \sum_{n=0}^{\infty} \sum_k L_n(kT) \cdot P_n(x - kT) \quad (3.9)$$

en donde

$$P_n(x) = \frac{G_n(x) V(x)}{W(x)} \quad (3.10)$$

Las funciones dadas por la ecuación (3.10) reciben el nombre de *funciones patrón* o *filtros de síntesis*. Estas permiten obtener la reconstrucción de la señal a partir de su interpolación (suma de convoluciones) con los coeficientes polinomiales de la señal.

La ecuación (3.7) implica que los coeficientes  $L_n(kT)$  se derivan de la señal  $L(x)$  convolucionada con las *funciones filtro* o *filtros de análisis* dadas por

$$D_n(x) = G_n(-x)V^2(-x) \tag{3.11}$$

seguida por un muestreo en múltiplos de  $T$ .

Al mapeo de la señal original  $L(x)$  a los coeficientes polinomiales  $L_n(kT)$  a través de la ecuación (3.7), es decir, con la convolución entre la señal y las funciones *filtro*, se le denomina *transformada polinomial directa*. A la reconstrucción de la señal a partir de los coeficientes polinomiales de acuerdo a la ecuación (3.9), esto es, con la interpolación (suma de convoluciones para todo orden  $n$ ) entre los coeficientes y las funciones *patrón*, se le denomina *transformada polinomial inversa*.

Estas transformaciones permiten pasar del dominio de la señal al de los coeficientes polinomiales, y viceversa. En otras palabras, la transformada polinomial directa e inversa corresponde a las etapas de *análisis* y *síntesis*, respectivamente, de una señal. La transformada polinomial directa e inversa se ilustran en la figura 15 (Martens, 1990a), la cual coincide con la descomposición en subbandas de una señal a través de bancos de filtros.

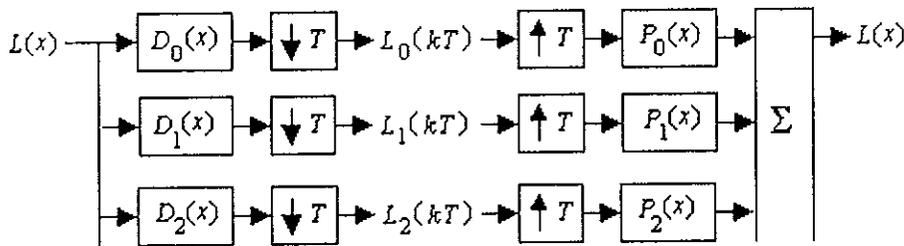


Figura 15. Transformada polinomial

Si los coeficientes polinomiales en la expansión de la señal de (3.9) se multiplican por constantes fijas  $t_n$ , entonces se presentan modificaciones que realzan o atenúan (según sea el caso) estructuras en la señal asociadas con las funciones de análisis, obteniéndose la señal modificada

$$\hat{L}(x) = \sum_{n=0}^{\infty} \sum_k t_n L_n(kT) \cdot P_n(x - kT) \tag{3.12}$$

### 3.3. Transformada Hermite

#### 3.3.1. Función Ventana

Para ilustrar las propiedades de las transformadas polinomiales, se considera el caso especial en que la función ventana es una función Gaussiana (Martens, 1990a), es decir,

$$V(x) = \frac{1}{\sqrt{\sqrt{\pi}\sigma}} e^{-\frac{x^2}{2\sigma^2}} \tag{3.13}$$

donde  $\sigma$  es la desviación estándar y el factor de normalización es tal que  $V^2(x)$  tiene energía unitaria. Los polinomios ortogonales que se asocian con  $V^2(x)$  son los conocidos *polinomios Hermite*, definidos por la ecuación (2.8.2) y los cuales satisfacen las relaciones dadas por las ecuaciones (2.12) y (2.13.1) a (2.13.4), y por lo tanto la técnica de descomposición local se denomina *transformada Hermite*.

Las razones por las cuales es común utilizar ventanas Gaussianas son múltiples. Primero, para este caso la teoría es matemáticamente tratable, consecuentemente las propiedades de la *transformada Hermite* pueden ser fácilmente derivadas y evaluadas. A su vez, la transformada Hermite discreta es una buena aproximación del caso analógico, y de aquí que se tengan varias propiedades relacionadas. Segundo, las ventanas Gaussianas que son separadas por el doble de su desviación estándar  $\sigma$  son un buen modelo de los campos receptivos traslapados que se han encontrados en experimentos fisiológicos (Sakitt & Barlow, 1982). Tercero, lo anterior conduce a que la transformada Hermite involucra funciones filtro que son derivadas de Gaussianas. Esto último coincide con lo visto en el capítulo 2, ya que estas funciones tienen un amplio uso en el modelado del sistema visual humano y visión por computadora (Young, 1986; Marr & Hildreth 1980). Finalmente, la ventana Gaussiana minimiza el producto de incertidumbres en los dominios espacial y frecuencia-espacial, lo cual es una propiedad interesante en análisis de imágenes (Wilson & Granlund, 1984), tal como se vio en el capítulo 2.

### 3.3.2. Función de peso

Como la función de peso  $W(x)$  es periódica con periodo  $T$  se puede expandir en series de Fourier (Boas & Buck, 1985) como

$$W(x) = \frac{\sqrt{2\sqrt{\pi}\sigma}}{T} w(x) \tag{3.14}$$

con

$$w(x) = 1 + 2 \sum_{k=1}^{\infty} e^{-\frac{1}{2} \left( k \frac{2\pi\sigma}{T} \right)^2} \cdot \cos \left( k \frac{2\pi\sigma}{T} x \right) \tag{3.15}$$

### 3.3.3. Funciones Filtro

Las funciones filtro determinan explícitamente la *información* en los coeficientes de la transformada Hermite. Por lo tanto, las principales propiedades de la transformada Hermite se definen en las funciones filtro. De la expresión general en (3.11) se obtiene que

$$D_n(x) = \frac{(-1)^n}{\sqrt{2^n n!}} \cdot \frac{1}{\sigma \sqrt{\pi}} H_n \left( \frac{x}{\sigma} \right) e^{-\frac{x^2}{2\sigma^2}} \tag{3.16}$$

Los polinomios Hermite  $H_n(x/\sigma)$  son ortogonales sobre la ventana Gaussiana  $V^2(x)$  (Abramowitz & Stegun, 1965).

Es demostrable (Abramowitz & Stegun, 1965) que la función filtro  $D_n(x)$  es igual a la derivada de orden  $n$  de una función Gaussiana, esto es

$$D_n(x) = \frac{(-1)^n}{\sqrt{2^n n!}} \cdot \frac{d^n}{d\left(\frac{x}{\sigma}\right)^n} \left[ \frac{1}{\sigma\sqrt{\pi}} e^{-\frac{x^2}{\sigma^2}} \right] \quad (3.17)$$

y su transformada de Fourier tiene una expresión simple:

$$d_n(\omega) = \frac{1}{\sqrt{2^n n!}} \cdot (j\omega\sigma)^n e^{-\frac{(\omega\sigma)^2}{4}} \quad (3.18)$$

En la figura 16 (Martens, 1990a) se muestran las funciones filtro (lado izquierdo) para  $n = 0, 1, \dots, 4$ , junto con su transformada de Fourier (lado derecho). En ellas se observa como están directamente relacionadas con las derivadas Gaussianas, las cuales son un modelo de los campos receptivos corticales, según lo visto en el capítulo 2 y en las secciones anteriores. De aquí su uso primordial en la detección de estructuras (bordes, líneas, y velocidad), tal y como lo hace el sistema visual humano, similares a las funciones filtro (Canny, 1986. Martens, 1990a).

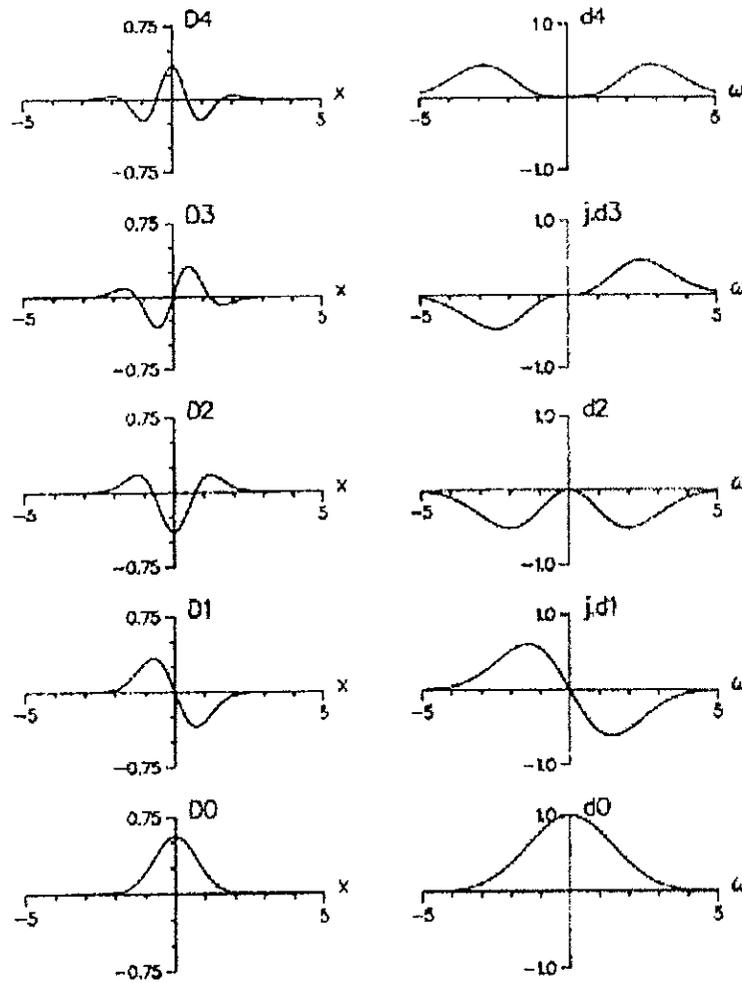


Figura 16. Funciones filtro en los dominios espacial y de la frecuencia para  $\sigma = 1$

### 3.3.4. Funciones Patrón

Las funciones patrón  $P_n(x)$ , también llamadas *funciones Hermite*, se requieren para sintetizar la señal original a partir de los coeficientes de la transformada Hermite. Estos vienen dados por la siguiente expresión analítica (Abramowitz & Stegun, 1965):

$$P_n(x) = \frac{T}{\sqrt{2^n n!}} \cdot \frac{1}{\sigma \sqrt{2\pi}} \cdot H_n\left(\frac{x}{\sigma}\right) \cdot \frac{1}{w(x)} e^{-\frac{x^2}{2\sigma^2}} \quad (3.19)$$

en donde  $w(x)$  es la función de peso dada en la ecuación (3.15). Además, la función Hermite también tiene la propiedad de ser *isomórfica* a su transformada de Fourier (Abramowitz & Stegun, 1965), esto es

$$p_n(\omega) = \frac{T}{\sqrt{2^n n!}} \cdot (-j)^n H_n(\omega\sigma) e^{-\frac{(\omega\sigma)^2}{2}} \quad (3.20)$$

De aquí que la figura 17 (Martens, 1990a), en la que muestra las funciones Hermite para  $\sigma = 1$ , puede ser considerada para representar tanto las funciones Hermite en el dominio espacial como su transformada de Fourier, es decir, las funciones Hermite en el dominio frecuencia-espacial.

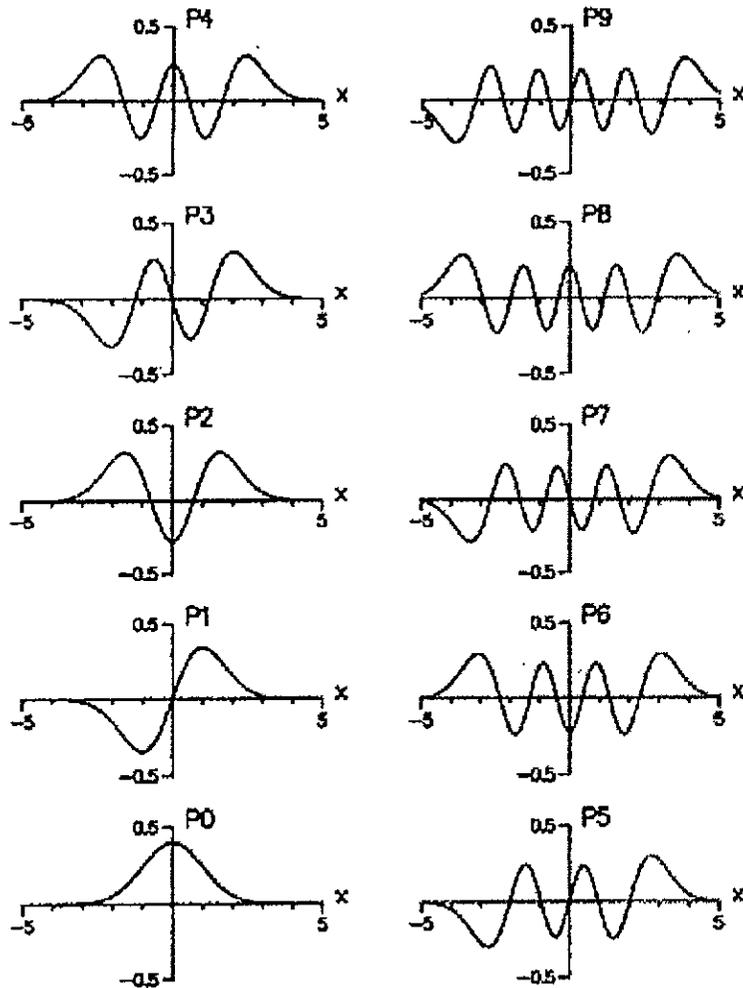


Figura 17. Funciones Hermite (funciones patrón) para  $\sigma = 1$

El parecido cercano entre estas funciones y, senos y cosenos truncados, indican que la transformada Hermite está relacionada con un análisis armónico.

### 3.3.5. Transformada Hermite Finita

En la práctica, la transformada Hermite se limita a los primeros términos de la expansión Este es el caso de la transformada Hermite finita, la cual está limitada por el orden del polinomio, es decir, cuando  $n = M$ , donde  $M < \infty$  (constante finita). Para que la transformada finita describa

adecuadamente a la señal,  $\sigma$  debe ser apropiadamente seleccionada. Es aquí cuando se tiene el problema de escala, porque la escala  $\sigma$  óptima depende del contenido local de la escena.

Se debe tener un compromiso entre dos consideraciones importantes. Por un lado, se desea que  $\sigma$  sea tan grande como sea posible, ya que sobre áreas grandes de integración mejora la relación señal-a-ruido de la salida y la eficiencia de la representación de la señal. Por otro lado,  $\sigma$  no debe ser demasiado grande, porque se carecería de una descripción precisa de la señal si se generan pocos términos en la expansión Hermite. Por último, conforme el orden de la expansión crece, el error de aproximación disminuye

### 3.4. Transformadas Polinomiales Discretas

Hasta ahora todas las señales y filtros se han considerado continuos. En aplicaciones prácticas de transformadas polinomiales se requiere una formulación para señales discretas. Esto se debe a que la mayoría de las señales que se procesan digitalmente tienen una representación matemática discreta, de hecho, las imágenes y las secuencias de ellas (vídeo digital) se almacenan y se procesan de manera digital en la mayoría de los casos debido a la naturaleza de los sistemas de almacenamiento y procesamiento. Hay dos alternativas para formular transformadas polinomiales discretas.

Una aproximación posible es ligar cada señal discreta con una continua (Martens 1990a), y con esto restringimos al caso de señales continuas o analógicas dadas por

$$L(x) = \sum_q L_q \cdot I(x - q\Delta) \quad (3.21)$$

que son completamente especificadas por un número contable de coeficientes  $L_q$  por medio de la interpolación con  $I(x)$ . Si se aplica una transformada polinomial directa a ésta señal, se obtienen los siguientes coeficientes

$$L_n(pT) = \sum_q L_q [D_n(x) * I(x)]_{x=pT-q\Delta} \quad (3.22)$$

Si  $T$  es múltiplo de la distancia de muestreo ( $T = T_\Delta \Delta$ ), entonces los coeficientes polinomiales de orden  $n$  se generan por una convolución discreta de la secuencia  $L_q$  con la secuencia del filtro

$$DI_n(q) = [D_n(x) * I(x)]_{x=q\Delta} \quad (3.23)$$

seguida de un submuestreo con factor de  $T_\Delta$ .

Sin embargo, también se pueden definir transformadas polinomiales directamente sobre señales discretas (Martens 1990a), esto es, sin la necesidad de ligar explícitamente las señales discretas con las continuas.

Las expresiones para las funciones de peso, filtro, y patrón permanecen válidas al reemplazar la variable continua  $x$  por una discreta. Sin embargo, todas las expresiones con integrales deben ser cambiadas por sumatorias. Por ejemplo, de acuerdo a la ecuación (3.5) el momento de orden  $n$  puede remplazarse por

$$c_n = \sum_x x^n V^2(x) \quad (3.24)$$

para  $n = 0, \dots, N$ .

Si la ventana discreta es finita, esto es,  $V(x) = 0$  para  $x < N_1$  y  $x > N_2$ , entonces la transformada polinomial tiene un orden finito  $N = N_2 - N_1$ . Los coeficientes polinomiales hasta orden  $N$  son entonces suficientes para obtener una reconstrucción perfecta para cualquier señal discreta. La razón es que la señal discreta dentro de la ventana  $V(x)$  tiene solamente  $N+1$  grados de libertad; es decir, corresponde a la longitud de la señal dentro de la ventana  $V(x)$ , la cual es finita, por lo que la expansión sobre polinomios ortonormales discretos puede considerarse sobre un espacio finito. De aquí que el orden máximo de la expansión sea el orden máximo de los polinomios, que a su vez coinciden con la dimensión del espacio vectorial finito.

### 3.4.1. Transformada Hermite Discreta

A continuación se verá el equivalente discreto de la transformada Hermite para el caso de una dimensión en que la señal  $L(x)$  es discreta, es decir,  $L(x)$  es una función real de variable entera  $x$ , o sea  $x \in \mathbb{Z}$ .

La contraparte discreta de una ventana Gaussiana es una *ventana binomial*, cuyo valor al cuadrado es

$$V^2(x) = \frac{1}{2^M} C_M^x \quad (3.25)$$

para  $x = 0, \dots, M$ ; y  $C_M^x$  es la combinatoria de  $M$  sobre  $x$ , esto es

$$C_M^x = \binom{M}{x} = \frac{M!}{(M-x)!x!} \quad (3.26)$$

Los polinomios ortonormales discretos que se asocian con esta ventana son los polinomios de Krawtchouck (Szegő, 1959)

$$G_n(x) = \frac{1}{\sqrt{C_M^n}} \sum_{k=0}^n (-1)^{n-k} C_{M-x}^{n-k} \cdot C_x^k \quad (3.27)$$

para  $x, n = 0, \dots, M$ .

Para valores grandes de  $M$ , la ventana binomial se reduce a una ventana Gaussiana. Esto es,

$$\lim_{M \rightarrow \infty} \frac{1}{2^M} C_M^{x+(M/2)} = \frac{1}{\sqrt{\pi} \sqrt{M/2}} \exp \left[ - \left( \frac{x}{\sqrt{M/2}} \right)^2 \right] \quad (3.28)$$

para  $x = -(M/2), \dots, M/2$ . Puede demostrarse (Szegő, 1959) que el mismo proceso de aplicar el límite a los polinomios de Krawtchouck da como resultado los polinomios Hermite, es decir,

$$\lim_{M \rightarrow \infty} G_n \left( x + \frac{M}{2} \right) = \frac{1}{\sqrt{2^n n!}} H_n \left( \frac{x}{\sqrt{M/2}} \right) \quad (3.29)$$

De aquí se tiene que la transformada Hermite discreta de longitud  $M$  aproxima a la transformada Hermite continua con desviación estándar  $\sigma = \sqrt{M/2}$ . Con esto, las propiedades de la transformada Hermite discreta pueden predecirse de una manera exacta mediante las propiedades correspondientes de la transformada Hermite continua. Esto nos permite tener un isomorfismo entre las transformadas Hermite discreta y continua, de tal manera que el modelado se puede hacer desde el dominio continuo y después extrapolarlo al dominio discreto con las consideraciones pertinentes.

Para el caso en que  $M$  es par, las funciones de análisis (funciones filtro) y síntesis (funciones patrón) se centran en el origen con un corrimiento de la ventana binomial de  $M/2$ . Esto conduce a la siguiente definición de los filtros de análisis o funciones filtro de las transformada Hermite discreta

$$D_n(x) = G_n \left( \frac{M}{2} - x \right) \cdot V^2 \left( \frac{M}{2} - x \right) \quad (3.30)$$

para  $x = -(M/2), \dots, M/2$ .

### 3.5. Transformadas Polinomiales en Tres Dimensiones

La transformada polinomial se puede generalizar a tres dimensiones (3D). Esto es útil para expandir señales que corresponden a espacios de 3D y hacer su análisis local. Las señales 3D de interés corresponden a las secuencias de imágenes o video digital. En ellas, se tiene un conjunto de imágenes estáticas a lo largo del eje  $z$  lo cual determina el cambio de las estructuras espaciales con el tiempo  $t$ , mientras que el eje  $x$  y  $y$  corresponden a las dimensiones espaciales en que se localizan dichas estructuras

Dada una función local ventana  $V(x, y, z)$ , los polinomios ortonormales  $G_{l, m-l, n-m}(x, y, z)$ , donde  $l$ ,  $m-l$ , y  $n-m$  son el grado del polinomio con respecto a  $x$ ,  $y$ , y  $z$ , respectivamente, se determinan de manera única por (de manera similar a la ecuación (3.3))

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V^2(x, y, z) G_{l, m-l, n-m}(x, y, z) G_{i, j-i, k-j}(x, y, z) dx dy dz = \delta_{li} \delta_{mj} \delta_{nk} \quad (3.31)$$

para  $n, k = 0, \dots, \infty$ ;  $m = 0, \dots, n$ ;  $l = 0, \dots, m$ ;  $j = 0, \dots, k$ ;  $i = 0, \dots, j$  (Martens 1990a).

La descomposición de señales en tres dimensiones (3D) en polinomios localizados se define como (siguiendo la ecuación (3.9))

$$L(x, y, z) = \sum_{n=0}^{\infty} \sum_{m=0}^n \sum_{l=0}^m \sum_{(p, q, r) \in S} L_{l, m-l, n-m}(p, q, r) \cdot P_{l, m-l, n-m}(x - p, y - q, z - r) \quad (3.32)$$

donde  $(p, q, r)$  corresponde a todas las coordenadas de la malla (*lattice*) de muestreo  $S$  en 3D. La única condición para la malla de muestreo es que la función de peso

$$W(x, y, z) = \sum_{(p,q,r) \in S} V(x-p, y-q, z-r) \quad (3.33)$$

sea diferente de cero para todas las coordenadas  $(x, y, z)$ .

Los coeficientes polinomiales  $L_{l,m-l,n-m}(p,q,r)$  se obtienen convolucionando la secuencia de imágenes (señal 3D) con las funciones filtro (de manera similar a la ecuación (3.11))

$$D_{l,m-l,n-m}(x, y, z) = G_{l,m-l,n-m}(-x, -y, -z) V^2(-x, -y, -z) \quad (3.34)$$

seguida por un muestreo de la salida en  $(p, q, r) \in S$ . Siguiendo la ecuación (3.10), las funciones patrón utilizadas para interpolar los coeficientes polinomiales en tres dimensiones vienen dadas por

$$P_{l,m-l,n-m}(x, y, z) = \frac{G_{l,m-l,n-m}(x, y, z) V(x, y, z)}{W(x, y, z)} \quad (3.35)$$

para  $n = 0, \dots, \infty$ ,  $m = 0, \dots, n$ ; y  $l = 0, \dots, m$ .

De acuerdo a la ecuación (3.8), el contenido de *energía local* de la señal viene dado por

$$E = \int_{-\infty}^{+\infty} L^2(x, y, z) V^2(x-p, y-q, z) dx dy dz = \sum_{n=0}^{\infty} \sum_{m=0}^n \sum_{l=0}^m L_{l,m-l,n-m}^2(p, q, r) \quad (3.36)$$

### 3.5 1. Transformada Hermite en Tres Dimensiones

Como se mencionó anteriormente, el caso tridimensional (3D) se refiere a señales espacio-temporales  $L(x, y, t)$ . Debido a que en la transformada Hermite se utilizan filtros Gaussianos con desviación estándar  $\sigma$  idéntica en todas las direcciones, se tiene que tener concordancia en algunas equivalencias entre las dimensiones espacial y temporal.

Por lo anterior se hará un mapeo de las señales espacio-temporales  $L(x, y, t)$  en señales tridimensionales  $L(x, y, z)$  seleccionando  $z = u \cdot t$  (Martens 1990a). La constante  $u$  tiene dimensiones de velocidad. En muchas aplicaciones, como el caso de señales muestreadas, el parámetro de equivalencia  $u$  es implícitamente seleccionado.

Hay indicaciones de que el sistema visual humano contiene dos subsistemas, uno sensible a frecuencias espaciales altas y temporales bajas, y el otro sensible a frecuencias espaciales bajas y temporales altas (Shulman & Mulvanny, 1983). Esto puede simularse por dos transformadas Hermite, una con  $u$  y  $\sigma$  pequeñas, y otra con  $u$  y  $\sigma$  grandes.

Un caso interesante especial de la transformada polinomial es cuando la función ventana es separable,  $V(x, y, z) = V(x)V(y)V(z)$  con malla de muestreo cuadrada. Entonces las funciones de análisis y síntesis también son separables, y fáciles de implementar. Por ejemplo, los coeficientes polinomiales se generan convolucionando la imagen con las funciones filtro  $D_l(x)D_{m-l}(y)D_{n-m}(z)$ , donde  $D_l(x)$  es la función filtro de una dimensión (1D) para la ventana  $V(x)$ , seguida por un muestreo de la salida en las direcciones horizontal, vertical, y temporal en múltiplos del espaciado de muestreo  $T$ .

La transformada Hermite se obtiene si la función ventana es Gaussiana. La ventaja importante de las ventanas Gaussianas en tres dimensiones es que tienen la propiedad única de ser tanto separables espacialmente como simétricas rotacionalmente (Martens 1990a). El análisis y síntesis de 3D se reducen al análisis y síntesis de 1D en cuanto a la implementación, es decir, cada dimensión  $(x,y,z)$  se procesa por separado, y la superposición de cada parte lleva al caso 3D.

Las propiedades correspondientes de las funciones filtro utilizando funciones ventana Gaussianas son su separabilidad tanto espacial como esférica, es decir, la separabilidad se tiene tanto en coordenadas cartesianas como en coordenadas esféricas.

La transformada de Fourier de  $D_l(x)D_{m-l}(y)D_{n-m}(z)$ , expresada en coordenadas esféricas, esto es, a través de las transformaciones  $\omega_x = \omega \cos \theta \cos \phi$ ,  $\omega_y = \omega \sin \theta \cos \phi$ , y  $\omega_z = \omega \sin \phi$ , es

$$d_l(\omega_x)d_{m-l}(\omega_y)d_{n-m}(\omega_z) = g_{l,m-l}(\theta) \cdot g_{m,n-m}(\phi) \cdot d_n(\omega) \quad (3.37)$$

donde  $d_n(\omega)$  es la transformada de Fourier de la función filtro Hermite  $D_n(r)$ , con  $r$  como coordenada radial, y

$$g_{m,n-m}(\theta) = \sqrt{\frac{n!}{(n-m)!m!}} \cos^m \theta \cdot \sin^{n-m} \theta \quad (3.38)$$

expresa la selectividad direccional del filtro. De esto, conforme el orden  $n$  del filtro se incrementa, el análisis será sucesivamente en frecuencias radiales más altas, es decir, resoluciones espaciales más altas, como el caso de 1D. Estos filtros pueden detectar la orientación de un patrón unidimensional (1D) independientemente de su estructura interna (Daugman, 1983). Con esto, la función  $g$  proporciona en los filtros de análisis para el caso de secuencias de imágenes (3D), selectividad direccional en el espacio y selectividad de velocidad en el tiempo (Martens 1990a).

La definición de la transformada Hermite en tres dimensiones (3D) se obtiene directamente empleando los conceptos anteriores de separabilidad y simetría rotacional, esto es

$$L(x, y, z) = \sum_{n=0}^{\infty} \sum_{m=0}^n \sum_{l=0}^m \sum_{(p,q,r) \in S} L_{l,m-l,n-m}(p, q, r) \cdot P_l(x-p) P_{m-l}(y-q) P_{n-m}(z-r) \quad (3.39)$$

donde los coeficientes Hermite se derivan de la señal original  $L(x,y,z)$  a través de la convolución con las funciones filtro  $D_l(x)D_{m-l}(y)D_{n-m}(z)$  descritas anteriormente.

Por simplicidad, se asumirá que  $(p, q, r) \in S$  es la malla de muestreo de los coeficientes  $L_{l,m-l,n-m}$ , por lo que  $(p,q,r)$  se omitirá de la notación, salvo cuando sea necesario indicarlo.

Para el caso de una transformada polinomial, la síntesis de la señal se puede aproximar considerando un número finito de términos, es decir, cuando el valor máximo de  $n$  es una constante fija  $N$ . Más aún, para el caso de una transformada polinomial discreta, se tiene una expansión finita definida por la longitud de la señal, y si además la ventana es de longitud finita, el orden se restringe a la longitud de la misma como máximo. El caso de reconstrucción exacta se tiene cuando  $N \rightarrow \infty$ . La señal aproximada para  $n = 0, \dots, N$ , se representa por  $\hat{L}_{l,m-l,n-m}$ .

De acuerdo con lo anterior, la medida de *energía local* de orden  $n$  se define como

$$E_n = \sum_{m=0}^n \sum_{l=0}^m L_{l,m-l,n-m}^2 \quad (3.40)$$

para  $n = 0, \dots, N$ . En el caso de secuencias de imágenes afectadas por ruido, la energía local  $E_n$  contiene contribuciones tanto de la señal como del ruido.

La medida de *energía local global* es la suma de todas las medidas de energía local hasta el máximo orden  $N$ , dada como

$$E = \sum_{n=1}^N E_n = \sum_{n=1}^N \sum_{m=0}^n \sum_{l=0}^m L_{l,m-l,n-m}^2 \quad (3.41)$$

El coeficiente de contraste que se utiliza como una medida local de las variaciones de la imagen (Martens, 1990b) viene dado por

$$C = \log[1 + E] = \log \left[ 1 + \sum_{n=1}^N \sum_{m=0}^n \sum_{l=0}^m L_{l,m-l,n-m}^2 \right] \quad (3.42)$$

Este coeficiente es cero si la secuencia puede ser aproximada por su valor promedio dentro de la ventana local binomial. Este indicador proporciona una buena indicación de las posiciones e intensidades de los bordes.

Capítulo 4  
Proyección Unidimensional de  
Secuencias de Imágenes

## Capítulo 4. Proyección Unidimensional de Secuencias de Imágenes

### 4.1. Introducción

Como se ha visto en el capítulo 3 (específicamente en la sección 3.5.), la *transformada polinomial* es una manera de representar una señal, ya sea imagen (señal en 2D) o una secuencia de ellas (señal en 3D). Más aún, cuando la ventana es Gaussiana, se pueden explotar muchas de las características vistas; entre ellas, la separabilidad y simetría rotacional de las funciones de análisis y síntesis, tanto en coordenadas cartesianas como esféricas, que permiten tener selectividad de patrones en 1D; por consiguiente, la fácil implementación de la transformada; y la posibilidad de representar la señal a través de un modelo que concuerda con el de los campos receptivos corticales del sistema de visión humano.

La interpretación que se puede dar a los algoritmos de codificación de secuencias de imágenes, es que representan la secuencia con la descomposición de la misma basándose en menor información. En la práctica, esto se hace a menudo determinando una aproximación óptima de la secuencia de imágenes de entrada por una suma ponderada de patrones elegidos a priori. Los pesos de los patrones de contribución constituyen la descripción de la secuencia de imágenes de entrada. Con estos coeficientes se sintetiza una aproximación de la secuencia de imágenes original.

Con el propósito de que la descripción de la secuencia de imágenes sea eficiente, es necesario que los patrones elegidos a priori estén bien adaptados a la secuencia de imágenes, esto es, que la secuencia de imágenes original se aproxime (localmente) con un número limitado de patrones. Sin embargo, como es grande la cantidad de características que ocurren en cada *frame* de la secuencia de imágenes, generalmente naturales (no sintéticas, como el caso de teleconferencia), es muy poco probable que un conjunto de patrones pueda ofrecer una descripción eficiente de todas esas características. Distintas estructuras espaciales, tales como regiones homogéneas, bordes, líneas, texturas, etc. Es probable que requieran descripciones distintas. La situación es más complicada por el hecho de que esas estructuras tengan tamaño diferente (escala espacial) y puedan variar en un rango de velocidades (escala temporal), como sucede a menudo en una secuencia de imágenes.

En este capítulo se propone la codificación de secuencias de imágenes basada en proyecciones unidimensionales de la transformada Hermite, esto es, representar la imagen espacio-temporal (otra forma de nombrar a la secuencia de imágenes) con la transformada Hermite en 3D a través de proyecciones sobre estructuras locales unidimensionales (1D), con el fin de tener un conjunto de coeficientes que describan a la secuencia de imágenes con menor información a la obtenida a través de la transformada Hermite en 3D. A su vez, se está explotando algunas ventajas de la transformada Hermite vistas en los capítulos precedentes.

La manera de encontrar los patrones 1D, que proporcionen información espacial (líneas y bordes) y temporal (velocidades y flujo óptico) de la imagen espacio-temporal, sobre los cuales se proyectarán los coeficientes 3D para describir a la señal, será con la utilización de los conceptos de expansión de señales y proyecciones vectoriales vistos en el capítulo 1. A su vez, se explotará la propiedad del teorema de Parseval para la conservación de la energía entre los dominios de análisis y de síntesis. Es con esta que se podría determinar la relevancia visual de ciertos coeficientes 1D, y de los cuales se podría tener una representación todavía menor a través del peso ponderado de ciertos coeficientes por debajo de un umbral de energía.

La *codificación unidimensional con la transformada Hermite (UCHT)* aplicada a secuencias de imágenes, pretende como uno de sus objetivos, además de proporcionar una representación eficiente de la secuencia, obtener una compresión de la secuencia, es decir, que haya una reducción de información, medida a través de las entropías, entre la secuencia y su representación UCHT.

Esta propuesta de codificación y compresión parte del hecho de que la UCHT se puede ver como un esquema de codificación tridimensional en subbandas de vídeo, es decir, la señal original se divide en subbandas de distintos anchos de banda en el dominio de la frecuencia espacio-temporal, de tal manera que la información se decorrelaciona permitiendo descartar los coeficientes que estén por debajo de un umbral de energía, y cuantizar los no eliminados (Podilchuk *et al.*, (1995)).

A diferencia de los sistemas de codificación y compresión de secuencias de imágenes basados en estimación y compensación de movimiento (como el caso de la DCT), y en los cuales se requiere a veces un modelo de la escena cuyo movimiento se estimará (Li *et al.*, (1994)), la codificación tridimensional en subbandas no requiere de un modelo a priori de la escena, y su en su representación está implícito el contenido de movimiento (Podilchuk *et al.*, (1995)).

## 4.2. Codificación Unidimensional con la Transformada Hermite (UCHT)

La presentación de ésta sección está completamente basada en el desarrollo por Martens (1990a, 1990b) de la teoría y aplicaciones de la transformada Hermite a aproximaciones de patrones en 1D.

### 4.2.1. Coordenadas Esféricas

Como se ha visto, una secuencia de imágenes es un arreglo tridimensional (3D) de pixels o intensidades de iluminación. Las direcciones espaciales corresponden a los ejes  $x$  y  $y$ , mientras que la dirección temporal corresponde al eje  $z$  que se relaciona con el eje  $t$  a través de la constante de velocidad  $u$ , es decir,  $z = u \cdot t$  (Martens, 1990a). De aquí el nombre de *secuencias de imágenes*, ya que se tiene una sucesión de imágenes, formadas en el plano  $x$  y  $y$ , a lo largo del eje  $z$ .

La ubicación de cualquier punto (pixel)  $P$  en el espacio de la secuencia se puede definir a través de sus coordenadas cartesianas  $(x,y,z)$ , o bien, a través de sus coordenadas esféricas  $(\rho, \theta, \phi)$ , tal y como se muestra en la figura 18

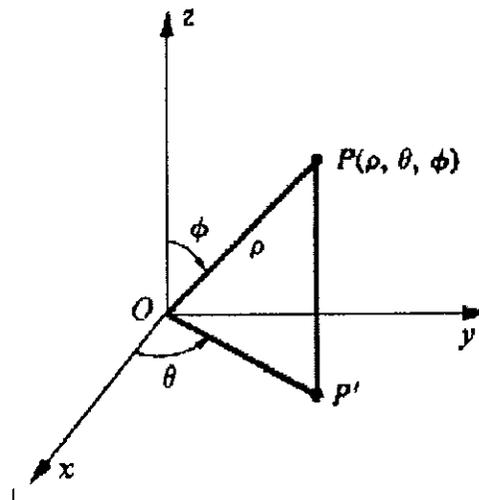


Figura 18. Representación de un punto en el espacio en coordenadas esféricas

Las relaciones de transformación de coordenadas cartesianas (rectangulares) a *esféricas* son

$$\rho^2 = x^2 + y^2 + z^2 \quad (4.1.1)$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (4.1.2)$$

$$\phi = \tan^{-1}\left(\frac{\sqrt{x^2 + y^2}}{z}\right) \quad (4.1.3)$$

Por el contrario, las relaciones de transformación de coordenadas esféricas a *cartesianas* son

$$x = \rho \sin \phi \cos \theta \quad (4.2.1)$$

$$y = \rho \sin \phi \sin \theta \quad (4.2.2)$$

$$z = \rho \cos \phi \quad (4.2.3)$$

#### 4.2.2. Rotación de Estructuras 1D

La figura 19 muestra el caso de una estructura unidimensional (1D) dentro de una imagen (2D) es decir, en el plano  $x,y$ ; que tiene un ángulo  $\theta$  con el ángulo  $x$ .

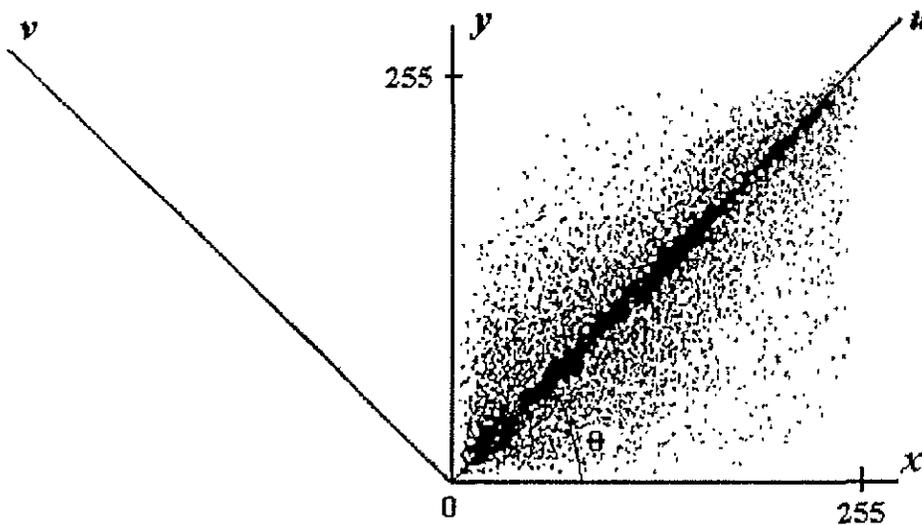


Figura 19. Rotación de un eje sobre una estructura unidimensional

La estructura está orientada sobre el eje  $u$ , por lo que se podría ubicar con el eje  $u$  y el ángulo  $\theta$ . Esto corresponde a rotar ángulo  $\theta$  el sistema de coordenada  $x,y$ ; y como resultado tener un nuevo sistema de coordenadas  $u,v$ .

Las relaciones que permiten hacer una rotación de ejes en 2D vienen dadas por

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \theta & \text{sen } \theta \\ -\text{sen } \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.3.1)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\text{sen } \theta \\ \text{sen } \theta & \cos \theta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \quad (4.3.2)$$

de aquí que la estructura se pueda representar en una dimensión (1D), el cual es el eje  $u$ , a partir de su ubicación respecto al sistema  $x,y$  determinada por el ángulo  $\theta$ . Con esto, la estructura 1D  $K(x,y)$  ubicada en 2D se puede mapear a su ubicación  $K_\theta(u)$ , donde

$$K_\theta(u) = K(x \cos \theta + y \text{sen } \theta) \quad (4.4)$$

Para el caso 3D, el eje  $u$  sobre el cual estaría la estructura 1D, se ubica a través del ángulo  $\theta$  que forma el eje  $x$  y la proyección de  $u$  sobre el plano  $x,y$ ; y el ángulo  $\phi$  que forma el eje  $u$  con el eje  $z$ . Con esto, la estructura 1D  $K(x,y,z)$  ubicada en 3D se puede mapear a su ubicación  $K_{\theta,\phi}(u)$ , donde

$$K_{\theta,\phi}(u) = K((x \cos \theta + y \text{sen } \theta) \cos \phi + z \text{sen } \phi) \quad (4.5)$$

### 4.2.3. Proyección 1D con la Transformada Polinomial

Para facilitar el desarrollo de la proyección a 1D de estructuras unidimensionales en 3D, comenzaremos el análisis para el caso 2D, y posteriormente los resultados se harán extensivos al caso 3D, con las consideraciones necesarias.

Como se ha mencionado, los patrones locales unidimensionales (1D) juegan un papel central en las etapas primarias de visión; tal es el caso de líneas y bordes para el espacio, y de velocidades (orientación del movimiento) para el tiempo.

La mejor aproximación local 1D dentro de una imagen se puede encontrar con la ayuda de la transformada polinomial, es decir, encontrar los ángulos  $\theta$  y  $\phi$  que permitan localizar las estructuras 1D, tal como se describió en la sección anterior.

Utilizando un criterio de error cuadrático de peso (energía del error cuadrático), minimizamos

$$E^2 = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} [K(x \cos \theta + y \text{sen } \theta) - L(x, y)]^2 V^2(x, y) dx dy \quad (4.6)$$

sobre todos los patrones  $K$  1D y ángulos  $\theta$ .

La función ventana 1D  $V_\theta^2(u)$  se define como la proyección de la función ventana 2D  $V^2(x, y)$  sobre un eje  $u$  que forma un ángulo  $\theta$  con el eje  $x$ . Aplicando la ecuación (4.3.2), esto es

$$V_\theta^2(u) = \int_{-\infty}^{+\infty} V^2(u \cos \theta - v \sin \theta, u \sin \theta + v \cos \theta) dv \quad (4.7)$$

Esta función ventana  $V_\theta^2(u)$  es independiente de la orientación si  $V^2(x, y)$  es rotacionalmente simétrica, como el caso de la ventana Gaussiana, ecuación (3.13), utilizada en la transformada Hermite.

Se puede expandir el patrón 1D  $K(u)$  en la base  $\{F_{n,\theta}(u); n = 0, 1, \dots\}$  de polinomios ortonormales a la ventana  $V_\theta^2(u)$ , esto es,

$$V_\theta^2(u) \left[ K(u) - \sum_{n=0}^{\infty} K_{n,\theta} \cdot F_{n,\theta}(u) \right] = 0 \quad (4.8)$$

Substituyendo las expansiones polinomiales 2D y 1D para  $L(x, y)$  y  $K(u)$ , respectivamente, en (4.6), y tomando la derivada parcial con respecto a  $K_{n,\theta}$ , se obtiene la siguiente solución óptima para los coeficientes patrón 1D, dados por

$$K_{n,\theta} = \sum_{k=0}^n \sum_{l=0}^k L_{l,k-l} \cdot h_{n,\theta}(l, k-l) \quad (4.9)$$

donde

$$h_{n,\theta}(l, k-l) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F_{n,\theta}(x \cos \theta + y \sin \theta) \cdot G_{l,k-l}(x, y) V^2(x, y) dx dy \quad (4.10)$$

es una *función de ángulo* que está completamente determinada por  $V^2(x, y)$ .

Los polinomios ortogonales  $F_{n,\theta}(u)$  y la función de ángulo pueden determinarse sin conocimiento explícito de  $V_\theta(u)$ . De hecho, solamente los momentos, de acuerdo a la ecuación (3.5), dados por,

$$c_{n,\theta} = \int_{-\infty}^{+\infty} u^n V_\theta^2(u) du \quad (4.11)$$

se requieren para especificar los polinomios ortogonales.

El cálculo de éstos momentos puede sin embargo basarse directamente en  $V^2(x, y)$ , dado que

$$c_{n,\theta} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x \cos \theta + y \sin \theta)^n V^2(x, y) dx dy \quad (4.12)$$

A partir de la ortogonalidad de los polinomios  $F_{n,\theta}(u)$ , se pueden derivar las siguientes propiedades para la *función de ángulo*:

$$h_{n,\theta}(l, k-l) = 0, \quad k > n \quad (4.13.1)$$

$$\sum_{k=0}^{\infty} \sum_{l=0}^k h_{m,\theta}(l, k-l) h_{n,\theta}(l, k-l) = \delta_{mn} \quad (4.13.2)$$

El error de aproximación 1D, dado por

$$E^2 = \sum_{k=0}^{\infty} \sum_{l=0}^k L_{l,k-l}^2 - \sum_{n=0}^{\infty} K_{n,\theta}^2 \quad (4.14)$$

puede minimizarse sobre el ángulo  $\theta$  maximizando la energía direccional

$$E_{\theta} = \sum_{n=0}^{\infty} K_{n,\theta}^2 \quad (4.15)$$

donde  $K_{n,\theta}$  está determinada por los coeficientes polinomiales 2D a través de la ecuación (4.9). El ángulo óptimo  $\theta$  es aquel para el cual la energía  $E_{\theta}$  es máxima, y por consiguiente el error de aproximación a 1D es mínimo

En la práctica, los primeros términos en esta medida de energía direccional son suficientes para hacer una buena estimación de la dirección óptima. Nótese que la energía direccional se encuentra a través de una simple combinación de los coeficientes polinomiales 2D.

Si la imagen original  $L(x,y)$  es localmente 1D, entonces el error de estimación debe ser cero para el ángulo óptimo  $\theta$ , de tal manera que los coeficientes polinomiales 2D satisfacen

$$L_{l,k-l} = \sum_{n=k}^{\infty} K_{n,\theta} \cdot h_{n,\theta}(l, k-l) \quad (4.16)$$

Si la imagen  $L(x,y)$  no es localmente 1D, entonces la expresión anterior todavía puede ser utilizada como una aproximación óptima 1D para los coeficientes polinomiales 2D.

Finalmente, la extensión al caso de proyección 1D a partir de los coeficientes 3D y su reconstrucción, es una extensión directa de lo visto en ésta sección. La única consideración que se debe contemplar es la rotación de ejes en 3D. Las ecuaciones aquí presentadas son válidas para el caso 3D.

#### 4.2.4. Proyección 1D con la Transformada Hermite

De acuerdo a lo visto en la sección 3.5.1., la transformada Hermite en 3D presenta la ventaja de ser una transformada polinomial con ventana Gaussiana que cumple la característica de separabilidad, esto es,  $V(x,y,z) = V(x)V(y)V(z)$ . A su vez, las funciones filtro son también separables, tanto espacial como esféricamente. Adicionalmente a estas características, también se tiene la separabilidad en el dominio de la frecuencia espacio-temporal, los filtros presentan una selectividad direccional, y son simétricamente rotacionales.

Todo esto conduce a la simplificación de las *funciones de ángulo*  $h_{n,\theta}$  vistas en la sección anterior y que permiten la proyección de 3D a 1D. De acuerdo a las ecuaciones (3.37) y (3.38), las funciones de ángulo para la transformada Hermite se relacionan con las funciones de selectividad direccional de las funciones filtro, ecuación (3.38), como

$$h_{n,\theta}(l, k-l) = \delta_{nk} \cdot g_{l,k-l}(\theta) \quad (4.17)$$

que resulta en una simplificación substancial sobre el caso general.

Adicionalmente a la selectividad direccional de los filtros de análisis, se tiene selectividad de velocidad. De hecho, la mejor aproximación de la señal original  $L(x,y,z)$  1D por un patrón unidimensional,  $K((x \cos \theta + y \sen \theta) \cos \phi + z \sen \phi)$ , de acuerdo a la ecuación (4.5), se encuentra maximizando la energía direccional

$$E_{\theta} = \sum_{n=0}^{\infty} K_{n,\theta,\phi}^2 = \sum_{n=0}^{\infty} \left[ \sum_{m=0}^n \sum_{l=0}^m g_{l,m-l}(\theta) \cdot g_{m,n-m}(\phi) \cdot L_{l,m-l,n-m} \right]^2 \quad (4.18)$$

sobre todo  $(\theta, \phi)$ , siempre y cuando el error de aproximación esté pesado por  $V^2(x, y, z)$ .

Si se tiene por ángulo óptimo  $\phi = \pi/2$  entonces la mejor aproximación 1D es un patrón puramente temporal. Esto incluye, por ejemplo, el caso de parpadeo uniforme. Si el ángulo óptimo  $\phi \neq \pi/2$ , la mejor aproximación 1D es un patrón haciendo un ángulo  $\theta$  con el eje  $x$  y moviéndose con velocidad constante.

El vector de velocidad es

$$v = (-u \cdot \tan \phi) \cdot (\cos \theta, \sen \theta) \quad (4.19)$$

La mejor aproximación 1D de los coeficientes Hermite 3D está dada por

Para  $n = 0, \dots, \infty$ ;  $m = 0, \dots, m$ ; y  $l = 0, \dots, m$ ; con los ángulos óptimos  $(\theta, \phi)$ , es decir, los ángulos que hacen que la energía direccional sea máxima y por consiguiente el error de aproximación sea mínimo.

### 4.3. Implementación y Consideraciones Prácticas

Para la implementación y aplicación de la UCHT se consideró una ventana par. La ventana binomial par más pequeña tiene escala  $M = 2$ . Además, consideramos el espaciado de muestreo entre ventanas como  $T = 2$ , por lo que se tiene reconstrucción perfecta para cualquier señal discreta con transformada de orden  $N = 2$ . Estos valores son para cada dimensión de la secuencia de imágenes (imagen espacio-temporal) dentro del espacio cartesiano.

Con esto, el procesamiento se hace a través de campos receptivos pequeños de tamaño  $3 \times 3 \times 3$ . Para la elección de éste tamaño se contempló el hecho de que en campos receptivos pequeños se pueden hacer búsquedas de estructuras de 1D de tal manera que la proyección sobre éstas sea una buena

aproximación. Esto equivale a considerar la conectividad 8 de los pixels de cada frame, y a su vez entre frames.

Para la implementación es importante determinar los ángulos existentes dentro de la malla de muestreo (lattice) de la ventana de análisis, ya que estos están determinados por el número de muestras de la ventana por tratarse de una representación digital. Para el caso de la ventana de  $3 \times 3 \times 3$ , los ángulos posibles son  $\theta = [-90, -45, 0, 45, 90]$  y  $\phi = [-54.7356, -45, 0, 45, 54.7356]$ . La combinación de ellos da la orientación de las estructuras unidimensionales en el espacio 3D.

Los programas asociados a la implementación de la UCHT fueron implementados en *MatLab*. Un listado de los mismos con su respectiva descripción se muestra en el *anexo 1*.

## Capítulo 5

### Resultados y Evaluación

## Capítulo 5. Resultados y Evaluación

A continuación se muestran los resultados obtenidos a través de los programas de la UCHT, y se analizan algunos aspectos de los mismos.

La secuencia de imágenes original sobre la cual se probó el algoritmo es un fragmento de la secuencia "interview", que para fines prácticos de procesamiento, como es el uso de memoria y tiempo de procesamiento, se tomaron subframes de la secuencia para la misma posición de cada frame.

Cabe mencionar que el algoritmo de UCHT se pretende aplicar principalmente en secuencia de imágenes naturales, específicamente de teleconferencia; pero esto no implica que se puedan aplicar en otro tipo de secuencias de imágenes con distinto contenido de movimiento, como el caso de secuencias sintéticas.

La figura 20 muestra seis subframes de la secuencia "interview" original compuesta por valores de pixels en tonos de grises.

Original



Figura 20. Secuencia original formada por seis subframes de la secuencia "interview"

Los resultados numéricos obtenidos a través del programa, como son las entropías, la razón de compresión, la razón señal-a-ruido (*SNR*) y la razón señal-a-ruido pico (*PSNR*) son:

```

1D coefficients and optimal angles values
-----
Entropy K0 :          6.4264 bits/pixel
Entropy K1 :          5.9035 bits/pixel
Entropy K2 :          5.2305 bits/pixel
Entropy THETA :       0.8763 bits/pixel
Entropy PHI :         1.3872 bits/pixel
Number of bits K0 :   54804.7576 bits
Number of bits K1 :   50344.9294 bits
Number of bits K2 :   44605.9004 bits
Number of bits THETA : 7473.1127 bits
Number of bits PHI :  11830.4348 bits

Total expansion values
-----
Entropy :              3.9648 bits/pixel
Number of bits :      169059.1349 bits

Original signal values
-----
Entropy :              8.0000 bits/pixel
Number of bits :      391680.0000 bits

Compression ratio
-----
Cr =    2.3168 : 1

Error signal measures
-----
SNR =    25.7280 dB
PSNR =   85.3355 dB

```

El cálculo del *PSNR* se determina por (González & Woods, (1992)):

$$PSNR = 10 \log_{10} \frac{(255)^2}{\sigma_e^2} \quad (5.1)$$

para el caso de secuencias de imágenes en tonos de gris, en donde  $\sigma_e^2$  es la varianza del ruido, es decir, de la señal error o error cuadrático medio MSE (*mean square error*), también llamada distorsión, la cual se obtiene con los valores de entrada originales  $x_i$  y los valores reconstruidos  $\hat{x}_i$ , como:

$$\sigma_e^2 = E \left\{ \sum_{i=0}^{N-1} |x_i - \hat{x}_i|^2 \right\} \quad (5.2)$$

Si el valor máximo no es 255, sino  $L$ , entonces se utiliza  $L^2$  en el cálculo del *PSNR*.

Para el cálculo de la varianza  $\sigma_e^2$  se puede utilizar un estimador insesgado de la misma, la cual corresponde a la varianza estadística  $s_e^2$  para el caso en que el tamaño de la muestra (longitud de la señal) es grande ( $N \geq 40$ ), lo cual se cumple en las secuencias de imágenes. Lo anterior se debe a que

$$s_e^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x_i - \hat{x}_i)^2 \quad (5.3)$$

y

$$\sigma_e^2 = E\left\{\frac{N}{N-1} s_e^2\right\} = E\{s_{e(N-1)}^2\} \quad (5.4)$$

tal que

$$s_{e(N-1)}^2 = \frac{N}{N-1} s_e^2 = \frac{1}{N-1} \sum_{i=0}^{N-1} (x_i - \hat{x}_i)^2 \quad (5.5)$$

donde  $s_{e(N-1)}^2$  es la varianza estadística *muestral*.

Con lo anterior, el cálculo del *PSNR* se obtiene computacionalmente como:

$$PSNR = 10 \log_{10} \frac{L^2}{s_{e(N-1)}^2} \quad (5.6)$$

con  $L = 255$  para secuencias de imágenes de 8 bits (tonos de grises). Con esto, el *PSNR* de la secuencia es

$$PSNR = 10 \log_{10} \left[ \frac{L^2}{\frac{1}{MNP} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{P-1} (x_{i,j,k} - \hat{x}_{i,j,k})^2} \right] \quad (5.7)$$

con dimensiones espaciales  $M \times N$  y dimensión temporal  $P$ .

El cálculo del *SNR* se determina, de manera similar, por (González & Woods, (1992)):

$$SNR = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2} \quad (5.8)$$

donde  $\sigma_x^2$  es la varianza de la señal original, es decir, de la secuencia de imágenes, determinada por

$$\sigma_x^2 = \frac{1}{MNP} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{P-1} (x_{i,j,k})^2 \quad (5.9)$$

La relación o tasa de compresión  $Cr$  viene dada por:

$$Cr = \frac{n_1}{n_2} \quad (5.10)$$

donde

$n_1$ : número de bits en la secuencia de imágenes original

$n_2$ : número de bits en la secuencia de imágenes comprimida

Si  $n_2 \geq n_1$  entonces hay una *expansión* y  $Cr \leq 1$ . Si  $n_2 < n_1$  entonces hay *compresión*, teniéndose  $Cr > 1$ . La  $Cr$  se expresa también como  $n_1:n_2$ . En general,  $Cr$  pertenece al intervalo abierto  $(0;\infty)$ .

La secuencia de imágenes *resultante* o *estimada* a partir de la UCHT se muestra en la figura 21

Estimated



Figura 21. Secuencia de imágenes resultante o estimada a partir de la UCHT

Como se observa, la reconstrucción es relativamente buena desde el punto de vista de apreciación. En la secuencia reconstruida se presenta cierta difuminación de algunos detalles, por lo que le da un

aspecto ligeramente borroso. Por supuesto, esta manera de medir la calidad del algoritmo, a partir de la apreciación de la reconstrucción, es subjetiva.

Una medida objetiva de la calidad de la reconstrucción está dada por las razones  $SNR$  y  $PSNR$ . Ésta última es muy utilizada en el área de codificación y compresión.

Si consideramos que un valor de  $PSNR = 40 \text{ dB}$  es aceptable, el algoritmo UCHT presenta un buen desempeño, al tenerse los valores de  $SNR = 25.7280 \text{ dB}$  y  $PSNR = 85.3355 \text{ dB}$ .

Los coeficientes 1D, obtenidos a partir de las proyecciones de los coeficientes Hermite 3D sobre estructuras unidimensionales, se muestran en la figura 22.

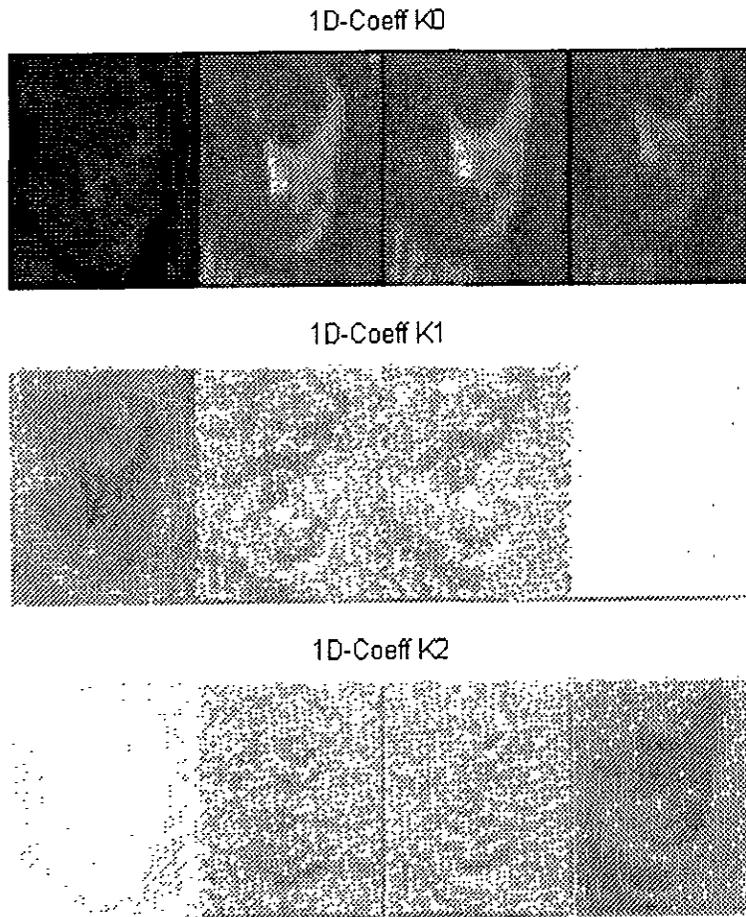


Figura 22. Coeficientes 1D de la UCHT hasta orden 2 ( $K_0$ ,  $K_1$ ,  $K_2$ )

Los *ángulos óptimos* theta ( $\theta$ ) y phi ( $\phi$ ) que resultan de la proyección unidimensional se muestran en la figura 23. Estos ángulos permiten reconstruir los coeficiente Hermite 3D para la posterior síntesis de la secuencia de imágenes a través de su interpolación con las funciones patrón.

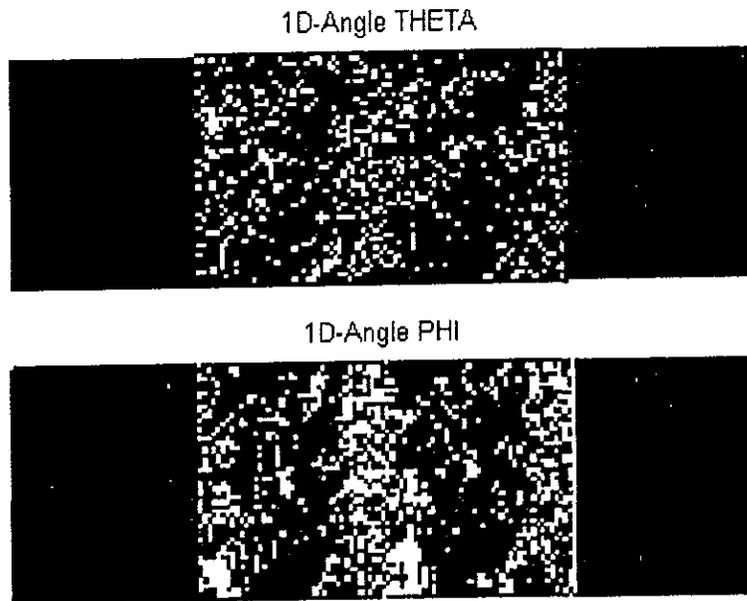


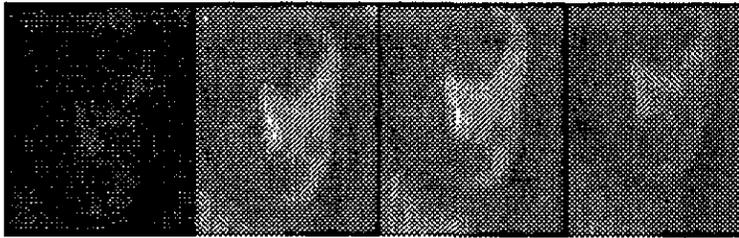
Figura 23. Ángulos óptimos de la UCHT ( $\theta$ ,  $\phi$ )

Nótese que la entropía de estos ángulos es baja, lo cual se debe al tamaño de los campos receptivos elegidos ( $3 \times 3 \times 3$ ), ya que el número de ángulos posibles es reducido y al mismo tiempo se contemplan todas las direcciones en conectividad 8 para la búsqueda de estructuras unidimensionales.

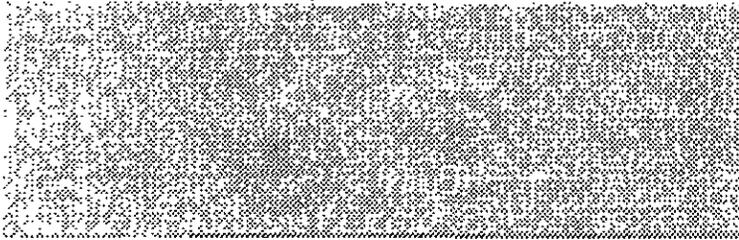
Aunque la transformada Hermite 3D no decorrelaciona totalmente la información, a través de la UCHT se incrementa la misma, reflejándose esto en la baja entropía de los ángulos.

Los coeficientes Hermite 3D se muestran en las figuras siguientes, 24 a 27, los cuales son hasta orden 2. El orden total corresponde a la suma del orden en cada dirección. La versión pasobajas es la asociada al coeficiente de orden cero,  $L_{0,0,0}$ .

Expansion 3D-Coeff 0 0 0



Expansion 3D-Coeff 1 0 0



Expansion 3D-Coeff 2 0 0

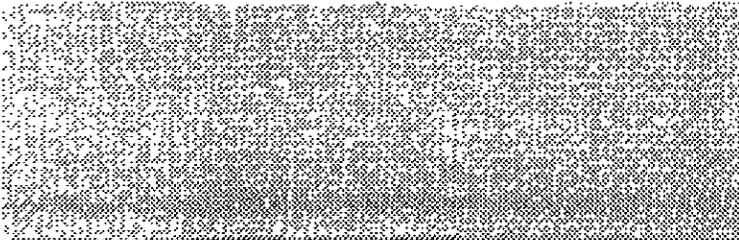


Figura 24. Coeficientes 3D de la HT ( $L_{000}$ ,  $L_{100}$ ,  $L_{200}$ )

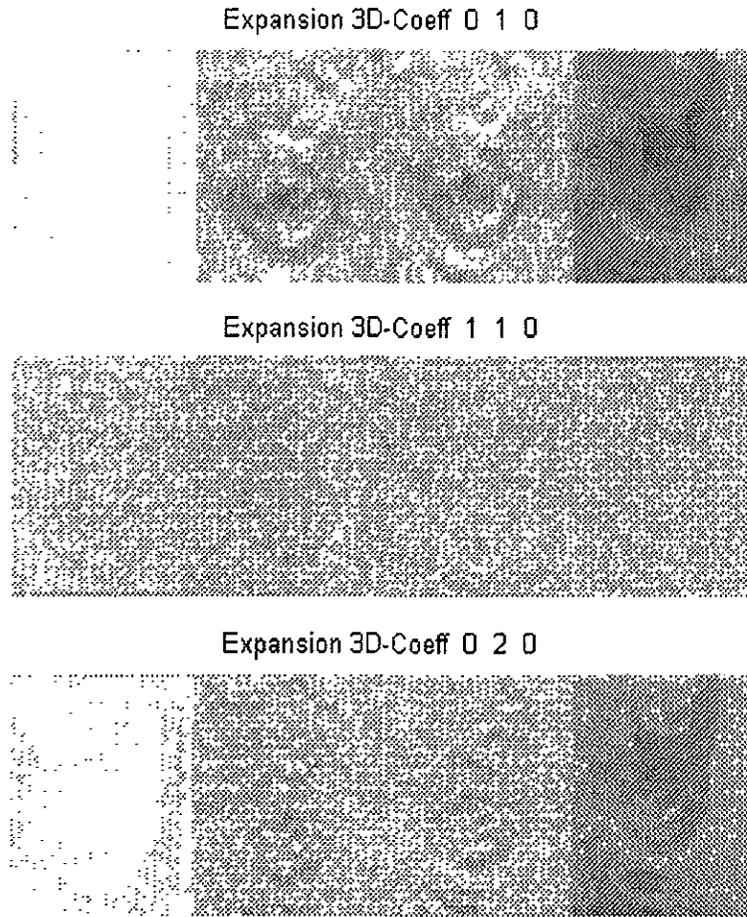
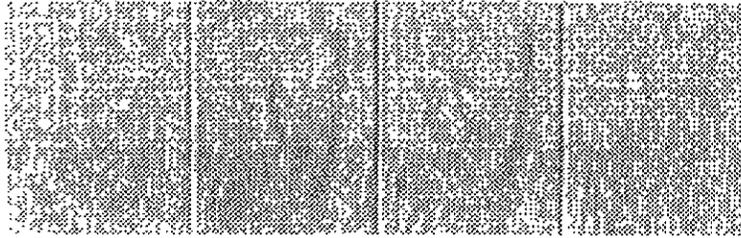
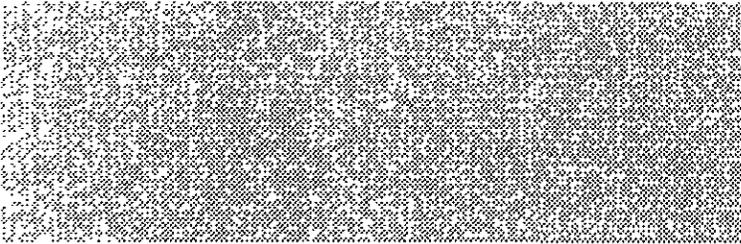


Figura 25. Coeficientes 3D de la HT ( $L_{010}$ ,  $L_{110}$ ,  $L_{020}$ )

Expansion 3D-Coeff 0 0 1



Expansion 3D-Coeff 1 0 1



Expansion 3D-Coeff 0 1 1

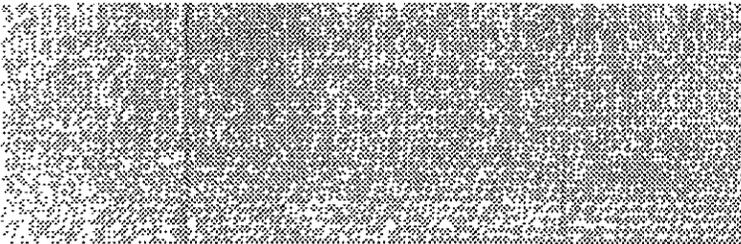


Figura 26. Coeficientes 3D de la HT ( $L_{001}$ ,  $L_{101}$ ,  $L_{011}$ )

Expansion 3D-Coeff 0 0 2

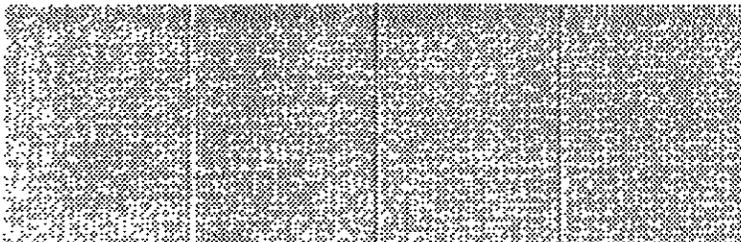


Figura 27. Coeficiente 3D de la HT ( $L_{002}$ )

Los coeficientes Hermite 3D, reconstruidos (estimados) a partir de los *angulos óptimos* y los coeficientes unidimensionales, se muestran en las figuras siguientes, 28 a 31.

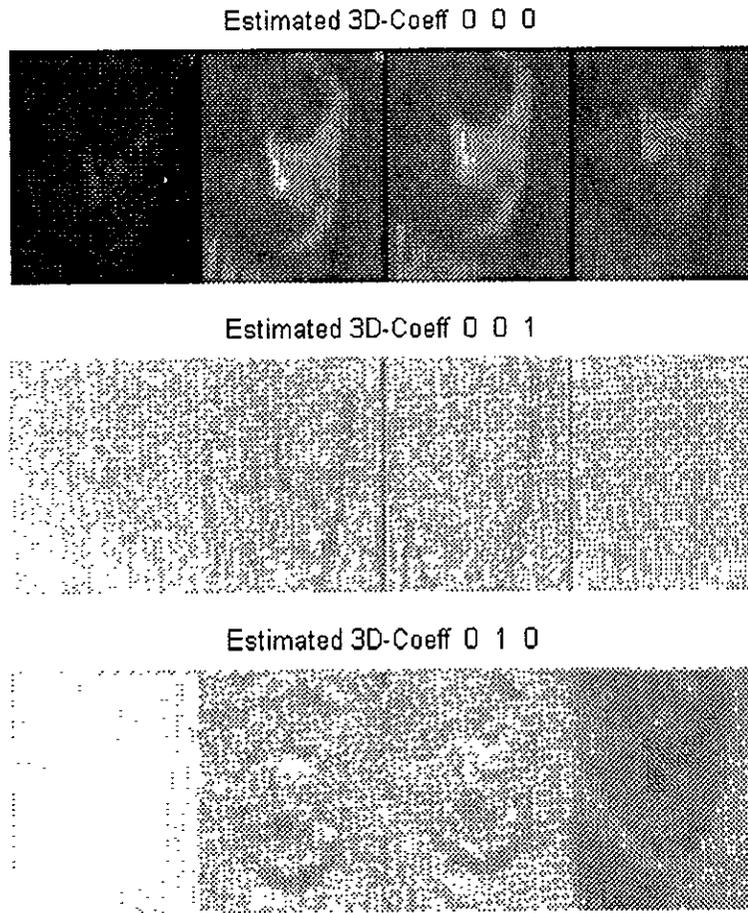


Figura 28. Coeficientes 3D estimados a partir de la UCHT ( $L_{000}$ ,  $L_{001}$ ,  $L_{010}$ )

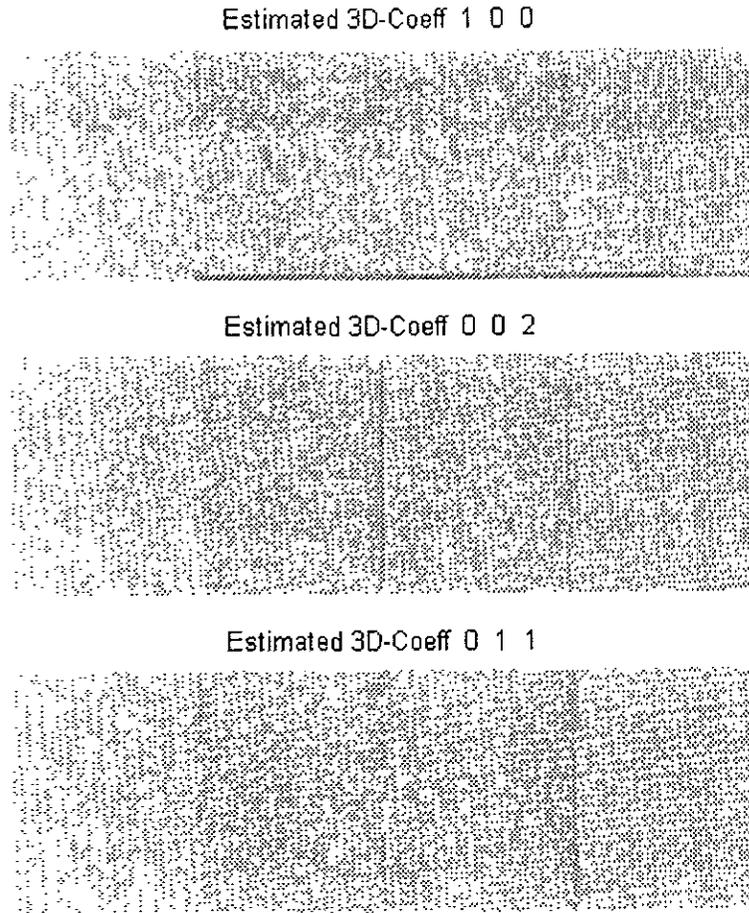


Figura 29. Coeficientes 3D estimados a partir de la UCHT ( $L_{100}$ ,  $L_{002}$ ,  $L_{011}$ )

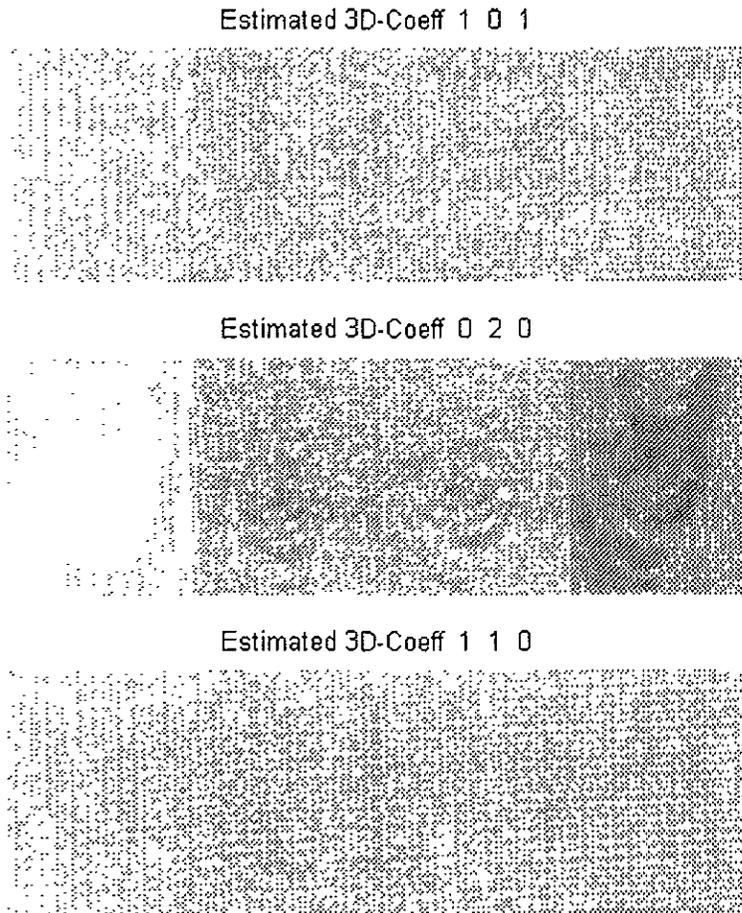


Figura 30. Coeficientes 3D estimados a partir de la UCHT ( $L_{101}$ ,  $L_{020}$ ,  $L_{110}$ )

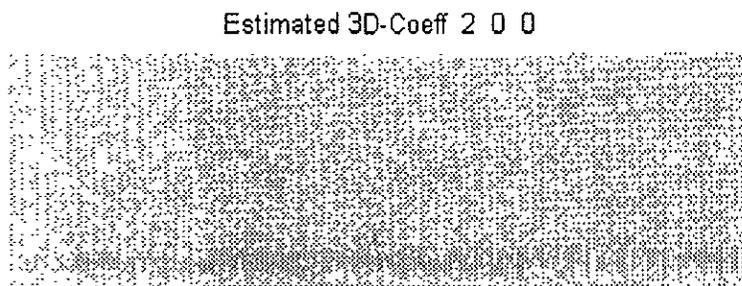


Figura 31. Coeficiente 3D estimados a partir de la UCHT ( $L_{200}$ )

Los coeficientes 3D de la expansión y los coeficientes estimados no se muestran en el mismo orden debido a las diferencias en la ordenación de éstos dentro de los programas de análisis y síntesis.

La reconstrucción de los coeficientes es muy aproximada a los originales de la expansión, como puede observarse en las gráficas anteriores.

Lo interesante es darle una interpretación a los coeficientes unidimensionales de la expansión.

Como se observa, los coeficientes 3D decorrelacionan información a través de la localización de las estructuras unidimensionales de mayor relevancia, las cuales corresponden a cada orden de la expansión (líneas, bordes y velocidades). La mayor parte de la energía se concentra en el coeficiente de orden cero,  $L_{0,0,0}$ , que es una versión pasobajas de la secuencia original.

La versión pasobajas temporal, orden cero en el eje  $z$  o temporal, proporciona las variaciones espaciales dentro de la secuencia. Los órdenes uno y dos de la expansión a lo largo del eje  $z$  o temporal, son la versión pasoaltas y proporcionan información temporal.

Recuérdese que si el ángulo con el eje  $z$  (o temporal) de la estructura unidimensional es  $\phi = \pi/2$ , entonces el patrón unidimensional es puramente temporal, como es el caso del parpadeo uniforme (Martens, 1990a).

Los coeficientes unidimensionales incluyen la información más relevante de los coeficientes 3D, ya que a partir de estos últimos se obtienen los primeros, haciendo una proyección que permite ubicar a la estructura 1D a través de los ángulos óptimos ( $\theta$ ,  $\phi$ ). El orden de los ángulos unidimensionales está directamente relacionado con el orden de los coeficientes 3D, ya que son correspondientes.

Con esto, se puede concluir que los coeficientes 1D proporcionan información espacial y temporal. La información espacial está concentrada prácticamente en el coeficiente de orden cero,  $K_0$ , mientras la información temporal se distribuye en los coeficientes de orden uno y dos,  $K_1$  y  $K_2$ . A través de éstos últimos es que se puede obtener información de movimiento, por lo cual se puede implementar un algoritmo de estimación del mismo. Esto queda abierto a desarrollos posteriores.

Dado que la transformada Hermite tiende a hacer una expansión, ya que el número de coeficientes es mayor al número de muestras de la señal original, se observa que a través del algoritmo de codificación unidimensional, el número de coeficientes es menor que en el simple caso de la transformada Hermite.

Debido al tamaño de la ventana,  $M = 2$ , al orden de la expansión,  $N = 2$ , y al espaciado entre ventanas,  $T = 2$  (el cual corresponde a un submuestreo en este valor de la señal filtrada), se tiene una compresión inherente de  $8/5 = 1.6$  a 1, esto es,  $Cr = 1.6:1$ . Para el caso de los coeficientes Hermite 3D, la relación de compresión es de  $8/10$ , esto es,  $Cr = 0.8:1$ , lo cual indica una expansión inherente.

Para el ejemplo de la secuencia aquí mostrada, figura 20, se tiene una relación de compresión mayor a la inherente, ya que  $Cr = 2.3168$ . Esto se debe a la reducción de la entropía en cada uno de los coeficientes de la expansión, y esencialmente, a la baja entropía presente en los valores de ángulo.

Con una buena cuantización sobre algún método predictivo de éstos coeficientes y con la aplicación de algún esquema de compresión como la codificación de longitud variable (*RCL: Run Length Coding*) o de *Huffman*, la razón de compresión puede ser elevada notablemente. Esto queda fuera del alcance de éste trabajo aunque queda abierto para desarrollos posteriores.

El presente trabajo propone solamente una nueva forma de codificación de las secuencias de imágenes, explotando los modelos asociados al sistema visual primario que identifican estructuras unidimensionales de importante relevancia.

## Conclusiones

- En el presente trabajo se plantea una forma de representar las secuencias de imágenes basada en proyecciones de los coeficientes Hermite tridimensionales (3D) sobre estructuras locales unidimensionales (1D), con el fin de tener un conjunto de coeficientes que describan a la secuencia de imágenes con menor información a la obtenida a través de la transformada Hermite en 3D.
- Esta codificación unidimensional con la transformada Hermite (UCHT) de secuencias de imágenes permite tener una compresión inherente con respecto a la representación 3D, además de ser una representación que involucra la información de patrones unidimensionales relevantes al sistema de visión humano.
- En la codificación UCHT se tiene un esquema de análisis/síntesis en el cual el problema de codificación unidimensional se resuelve al encontrar las estructuras en 1D que sean relevantes, es decir, que tengan la máxima energía. Estas estructuras 1D se localizan en el espacio 3D del campo receptivo de análisis con los *ángulos óptimos* ( $\theta$ ,  $\phi$ ). Los ángulos óptimos permiten reconstruir los coeficientes 3D a través de las *funciones de ángulo*, las cuales también se utilizan para la proyección de 3D en las estructuras 1D.
- Debido a que se requieren menos coeficientes para representar a la imagen espacio-temporal, y como los coeficientes de ángulo son de baja entropía al constituirse por un conjunto pequeño e ángulos que presentan cierta decorrelación local, existe entonces una compresión medida a través de la entropía global, adicional a la compresión inherente por la reducción en tamaño e la representación 3D con respecto a la secuencia original.
- Los coeficientes unidimensionales incluyen la información más relevante de los coeficientes 3D, ya que a partir de estos últimos se obtienen los primeros. La mayor parte de la energía se concentra en el coeficiente de orden cero, que es una versión pasobajas de la secuencia original. La versión pasobajas temporal, orden cero en el eje del tiempo, proporciona las variaciones espaciales dentro de la secuencia. Los órdenes superiores a lo largo del eje del tiempo, son la versión pasoaltas y proporcionan información temporal. El orden de los coeficientes 1D corresponde a una combinación de los coeficientes 3D del mismo orden.
- Algunas mejoras que se podrían tener en futuros trabajos con la UCHT son, la utilización de un umbral de energía por debajo del cual se descarten los valores de los coeficientes y obtener una representación con menor redundancia; interpretar de una manera minuciosa la importancia de cada uno de los coeficientes de la UCHT, multiplicando sus valores puntualmente por constantes que los atenúen y ver los efectos en la reconstrucción.
- Finalmente, esta forma de codificación, se deja una base para continuar el procesamiento digital de secuencias de imágenes, como sería la caracterización de las mismas, estimación y análisis del movimiento, y compresión.

## Anexos

### Anexo 1. Listado de los programas para la UCHT

Los siguientes programas son los desarrollados para la “Codificación Unidimensional con la Transformada Hermite (UCHT)” de secuencias de imágenes. Los programas fueron elaborados en MatLab® 5.2 de ©The MathWorks, Inc.

MatLab® es un sistema que contiene operaciones viables en un procesador digital de señales (DSP), por lo que los programas implementados en MatLab® son factibles de implementarse en un sistema basado en un DSP para que el algoritmo trabaje en tiempo real.

El contenido y descripción de las funciones que conforman UCHT están contenidos en el archivo CONTENTS.M.

```
*****
*** FileName: CONTENTS.M
*****

% Contents
%
% -----
% Unidimensional Coding by Hermite Transform (UCHT) of Image Sequences
% -----
%
% COMPRESS directory
%
% ENTROP          Entropy of a ND-signal
% HERMITE         Hermite Filter and Pattern functions
% GETANGFN       Extract the 2D angle function Gm,n-m of order Or=[m n-m]
% GETCOEFF       Extract the 3D coefficient Lmnp of order MNP=[m n p]
% GETWANGZERO    Gets array without angles equal to zero
% SHOWCOEFF      Shows the coefficient L of order mnp (3D) as an image deck
% SHOWHT1D       Shows the 1D projection from the 3D Hermite Transform
% SHOWHT3D       Shows the 3D sequence Hermite Transform
% SHT3DENER      Sequence Hermite Transform Energy
% SHTA3D         Sequence Hermite Transform (HT) Analysis for 3D
% SHTANG3D       Sequence Hermite Transform Angles for 3D-lattice
% SHTANGFN       Sequence Hermite Transform Angle Functions
% SHTEST3D       Sequence Hermite Transform Estimation of 3D coefficients from
1D
% SHTFIT1D       Sequence Hermite Transform (3D) fit to 1D projection
% SHTMULTA       Sequence Hermite Transform Multiresolution Analisis
% SHTMULTS       Sequence Hermite Transform Multiresolution Synthesis (zero
prediction)
% SHTPLOT        Sequence Hermite Transform Plots for 3D
% SHTS3D         Sequence Hermite Transform (HT) Synthesis for 3D
% SIGERROR       Signal Error for ND-signals
%
% Other functions
%
% COMBV          Combinatorial vectors
% COMBVZ        Combinatorial vectors with negatives
% ENTROPSEE      Display entropy in format as C-language '%15.4f'
```

ESTA TESIS NO DEBE SALIR DE LA BIBLIOTECA

```

* MULT1D2L          Multiresolution Analysis-Synthesis for 1D-coefficients with 2
Levels
* SEQSQUARED       Sequence of 10 frames of a squared object (hard-coded)
* SEQSQUARED2      Sequence of Nf frames of a squared object
* SHTFIT1D2        Same as SHTFIT1D but improved with coefficient energy
* SHTDATOS1        Generate some data for tests 1
* SHTDATOS2        Generate some data for tests 2
* SHTESCALA        Sequence Hermite Transform Scaling from display purposes
* SHTMULT          Sequence Hermite Transform Multiresolution Analysys-Synthesis
* SHTPRED          Sequence Hermite Transform Coefficients Prediction above order
zero.
* SHTPROC          Function calling for sequence hermite transform
* SHTROCIO1        Sequence Hermite Transform Processing 1
* SHTROCIO2        Sequence Hermite Transform Processing 2
* SHTROCIO3        Sequence Hermite Transform Processing 3
* SHTROCIO4        Sequence Hermite Transform Processing 4
* SHTROCIO5        Sequence Hermite Transform Processing 4

```

```

* C. Rivero, April 2000.
* Please send any error you find or any comment
* via e-mail to : carlosj@verona.fi-p.unam.mx

```

```

*****
*** FileName: GETCOEFF.M
*****

```

```

function [L]=getcoeff(Yt,MNP,Ord)
% [L]=GETCOEFF(Yt,MNP,Ord);
%
% Extract the 3D coefficient Lmnp of order MNP=[m n p] for the coefficient set Y
% $Y\0\0\0% For example, if MNP=[0 0 1] then the coefficient L001 is returned
within L.
% Ord is the order for the coefficient orders. This value is returned by SHTA3D.
* C. Rivero, April 2000.
* Please send any error you find or any comment
* via e-mail to : carlosj@verona.fi-p.unam.mx

```

```

[Ordx,Ordy]=size(Ord);

Ok=0;
for k=1:Ordx,
    if sum(find(Ord(k,:)==MNP))==6,
        Ok=1;
        break;
    end;
end;

L={};
if (Ok==1),
    L=Yt(:, :, :, k);
end;

```

```

*****
*** FileName: GETANGFN.M
*****

```

```

function [G]=getangfn(ANGSFN,OrdAng,Or,ValAng,TypeAng)
% [G]=GETANGFN(ANGSFN,OrdAng,Or,ValAng,TypeAng);
%
% Extract the 2D angle function Gm,n-m of order Or=[m n-m] from the angle

```

```

* functions array ANGSEFN (default for Or=[0 0]). ANGSEFN is returned by
SHTANGFN.
* The function G exists for angles THETA and PHI stores within array ANGSE
* which is returned by function SHTANG3D.
* For selecting the angle, argument TypeAng must be chosen between 0 and 1, i.e.
  Sy==== 0 yields THETA angle and 1 yields PHI angle (default=0).
*
* The ValAng argument corresponds to the row-index of the required value
angle. This value must not exceed the row-number of ANGSEFN (default=1).
* If ValAng is lower than 1 then all angle rows are returned within G for the
* specified type of angle theta or phi (TypeAng).

* OrdAng is the order for the angles orders. This value is returned by
SHTANGFN.
* For example, if Or=[0 1] then the angle function G01 is returned within G.

* C. Rivero, April 2000.
* Please send any error you find or any comment
* via e-mail to : carlosj@verona.fi-p.unam.mx

if (nargin < 5), TypeAng=0; end;
if (nargin < 4), ValAng=1; end;
if (nargin < 3), Or=[0 0]; end;
if (nargin < 2), error('At least two parameters are required'); end;

if prod(size(Or))~=2, error('Parameter Or must be a row vector of 1-by-2'); end;
DimAngsFn=size(ANGSEFN);

* G00 is always equal to 1 for all angles THETA and PHI
if sum(Or)==0,
    if (ValAng < 1), % all angles for TypeAng (theta or pi)
        G=ones(DimAngsFn(1),1);
    else % Specified angle referenced as index ValAng for TypeAng (theta or pi)
        G=1;
    end;
    return;
end;

%* Just for accepting ValAng as a scalar
% if ((ValAng < 1) | (ValAng > DimAngsFn(1))),
%     error('The subscript range for ValAng is not valid');
% end;

if ((TypeAng < 0) | (TypeAng > 1)), TypeAng=0; end;
TypeAng=TypeAng+1; % matrix index reference starts with 1

[Ordx,Ordz]=size(OrdAng);

Ok=0;
for k=1:Ordx,
    if sum(find(OrdAng(k,:)==Or))==3,
        Ok=1;
        break;
    end;
end;

G=[];
if (Ok==1),
    if (ValAng < 1), % all angles for TypeAng (theta or pi)
        G=ANGSEFN(:,TypeAng,k);
    else % Specified angle referenced as index ValAng for TypeAng (theta or pi)
        G=ANGSEFN(ValAng,TypeAng,k);
    end;
end;

```

```
end;
```

```
*****
*** FileName: ENTROP.M
*****
```

```
function [H,B]=entrop(X,q)
% [H,B]=ENTROP(X,q);
%
% Entropy of ND-signal X.
% For Hermite Transform coefficients, the partial entropies can be computed
% by passing the coefficients to this function. It is important to calculate
% ND-HT coefficient L000 = Yt(:, :, :, 1) entropy separated from the others.
%
% X ND-array signal to which entropy is calculated.
% q number of bits for quantization (PCM) levels of the source.
% (default: 8).
%
% H total entropy of the scalar-quantized signal X.
% B total number of bits used according to entropy H an size of X.
% This output parameter is optional.
%
% Note: If there are no output parameters, only entropy H is returned.
%
% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx
```

```
if (nargin < 2), q=8; end;
if (nargin < 1), error('At least one parameter is requiered'); end;
```

```
Q=2.^q; % number of quantization (PCM) levels.
Qmax=Q-1; % maximum level for initial level as zero.
```

```
P=fix(X(:).*Qmax);
Nbins=length(P);
N=hist(P,Q);
K=find(N);
Y=N(K)./Nbins;
```

```
H=-1.0.*sum(Y.*(log(Y)./log(2.0)));
B=length(X(:)).*H;
```

```
if (nargout < 2),
    clear B;
end;
```

```
*****
*** FileName: GETWANGZERO.M
*****
```

```
function [Y]=Getwangzero(X)
% [Y]=GETWANGZERO(X);
%
% Gets array without angles equal to zero.
%
% Returns the array X without vectors equal to zero
% For example, if X is a 4-by-2 matrix with one row [0 0] then Y yields a
% matrix 3-by-2 (without yhe vector [0 0]).
%
```

```

% This function is useful for calculating the angles lattice for the angle funct
% Sy====
% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

```

```

S=size(X);
D=length(S);
Smax=D*(D+1)/2;
XPows=S(1);
XCols=1;
for k=2:D,
    XCols=XCols.*S(k);
end;
X=reshape(X, [XRows, XCols]);
Z=zeros(1, XCols);

```

```

I=[];
Y=[];
for k=1:XRows,
    if sum(find(X(k, :)==Z))~=Smax,
        I=[I, k];
    end;
end;
Y=X(I, :);

```

```

Sy=size(Y);
D=length(S);
YRows=Sy(1);
YDims=[];
for k=2:D,
    YDims=[YDims, S(k)];
end;
Y=reshape(Y, [YRows, XCols]);

```

```

*****
*** FileName: MULT1D2L.M
*****

```

```

function [Xe]=mult1d2l(X)
% [Xe]=MULT1D2L(X);
%
% Multiresolution Analysis-Synthesis for 1D-coefficients with 2 Levels.
%
% X original 3D sequence.
% Xe synthesized 3D sequence.
%
% Note 1: X, Xe, and entropy values are saved to file 'mult1d2l.mat'.
% Note 2: The entropy of preserved coefficients is shown for coding purposes.
%
% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

```

```

% 1. Analysis
-----

```

```

% 3D-HT Analysis and 1D-HT projection at level 1
[Y1, Ord1]=shta3d(X, 'null');
[Kn1, Ktp1]=shtfit1d(Y1, Ord1);
% 3D-HT Analysis and 1D-HT projection at level 2
[Y2, Ord2]=shta3d(Kn1(:, :, :, 1), 'null');

```

```
[Kn2,Ktp2]=shtfit1d(Y2,Ord2);
```

```
* Coding results
```

```
-----
* Entropy for Km(I=1), m=1,2; Kn(I=2), n=0,1,2 (I: Ith-Level)
% and angles optimal angles Ktp_k(I=1), k=0,1,2; l=1,2
[Kx1, Ky1, Kz1, Kc1]=size(Kn1);
[Kx2, Ky2, Kz2, Kc2]=size(Kn2);
SizeKn1=Kx1*Ky1*Kz1;
SizeKn2=Kx2*Ky2*Kz2;
% Coefficient entropies
Hn=[entrop(Kn1(:,:,:,2)),...
    entrop(Kn1(:,:,:,3)),...
    entrop(Kn2(:,:,:,1)),...
    entrop(Kn2(:,:,:,2)),...
    entrop(Kn2(:,:,:,3))];
% Angle entropies
Htp=[entrop(Ktp1(:,:,:,1)),...
    entrop(Ktp1(:,:,:,2)),...
    entrop(Ktp2(:,:,:,1)),...
    entrop(Ktp2(:,:,:,2))];
% Coefficient and angle bits
Bn=[Hn(1)*SizeKn1, Hn(2)*SizeKn1, Hn(3)*SizeKn2, Hn(4)*SizeKn2, Hn(5)*SizeKn2];
Btp=[Htp(1)*SizeKn1, Htp(2)*SizeKn1, Htp(3)*SizeKn2, Htp(4)*SizeKn2];
BTot=sum(Bn)+sum(Btp); % Total
```

```
disp('Entropy for Km(I=1), m=1,2; Kn(I=2), n=0,1,2 (I: Ith-Level)');
disp('and optimal angles Ktp_k(I=1), k=0,1,2; I=1,2');
disp('Entropies for 1D-coefficients K1(1), K2(1), K0(2), K1(2), K2(2)');
```

```
%Hn
fprintf('Hn= \n');
fprintf(' %15.4f\n',Hn);
```

```
disp('Entropies for 1D-angles Ktp_1(1), Ktp_2(1), Ktp_1(2), Ktp_2(2)');
%Htp
fprintf('Htp= \n');
fprintf(' %15.4f\n',Htp);
```

```
disp('Total number of bits for optimum coding of 1D-coefficient projection');
%BTot
fprintf('BTot= %15.4f bits\n',BTot);
```

```
% Original sequence number of bits
Nbits=8; BOrg=prod(size(X)).*Nbits;
disp('Original sequence number of bits');
%BOrg
fprintf('BOrg= %15.4f bits\n',BOrg);
```

```
% Compression ratio
Cr=BOrg/BTot;
disp('Compression ratio (Cr = BOrg/BTot)');
%Cr
fprintf('Cr= %15.4f\n',Cr);
%fprintf('\n');
```

```
% 2. Synthesis
```

```
-----
% 3D-HT Reconstruction and synthesis from 1D-HT projection at level 2
[Y2,Ord2]=shtest3d(Kn2,Ktp2);
[Z]=shts3d(Y2,Ord2,size(Kn1(:,:,:,1)));
```

```

Knl(:, :, :, 1)=Z;
% 3D-HT Reconstruction and synthesis from 1D-HT projection at level 1
[Y1, Ord1]=shtest3d(Knl, Ktpl);
[Xe]=shts3d(Y1, Ord1, size(X));

```

• Quality measures

```

% SNR and PSNR in dB
[Esnr]=sigerror(X, Xe, 'SNR'); [Epsnr]=sigerror(X, Xe, 'PSNR');
disp('Signal-to-noise ratio (SNR) and Peak-SNR (PSNR)');
%[Esnr, Epsnr]
fprintf(' SNR= %9.4f dB\n', Esnr);
fprintf(' PSNR= %9.4f dB\n', Epsnr);
fprintf('\n');

```

% Save results

```

save mult1d21 X Xe Hn Htp BTot BOrg Cr Esnr Epsnr

```

```

*****
*** FileName: SEQSQUARED.M
*****

```

```

function [X]=seqsquared
% [X]=SEQSQUARED;
%
% Sequence of 10 frames of a squared object (hard-coded).
%
% X 3D sequence.
%
% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx
%
% Possible types of moving:
% 'S', 'H', 'V', 'HV' (Static, Horizontal, Vertical, 45° = Hor.-Ver.)
TypMov='H';

% Number of frames
Nf=10;

% Sequence (3D-signal) size and generation
DimX=[80 80 10];
X=zeros(DimX);
% Squared object (2D) size within each frame and generation
DimS=[20 20];
S=ones(DimS);

% Object initial position from the up-left corner of the first frame
PosS=[20 20];
% Object displacement increment for all directions
DPosS=4;

TypMov=uppercase(TypMov);
if strcmp(TypMov, 'S'), DPosS=0; end;

% Validation of frame size respect object size
if ~all((PosS+DPosS.*Nf+DimS) < [DimX(1) DimX(2)]),

```

```

error('Object dimensions and its displacing area are greater than frame dimen
      sy□□□end;

for k=1:Nf
    if strcmp(TypMov,'S'), % Static object
        X(PosS(1):PosS(1)+DimS(1)-1, ...
          PosS(2):PosS(2)+DimS(2)-1, k) = S;

    elseif strcmp(TypMov,'H'), % Horizontal moving
        X(PosS(1):PosS(1)+DimS(1)-1, ...
          PosS(2)+(k-1)*DPosS:PosS(2)+(k-1)*DPosS+DimS(2)-1, k) = S;

    elseif strcmp(TypMov,'V'), % Vertical moving
        X(PosS(1)+(k-1)*DPosS:PosS(1)+(k-1)*DPosS+DimS(1)-1, ...
          PosS(2):PosS(2)+DimS(2)-1, k) = S;

    elseif strcmp(TypMov,'HV'), % 45° moving
        X(PosS(1)+(k-1)*DPosS:PosS(1)+(k-1)*DPosS+DimS(1)-1, ...
          PosS(2)+(k-1)*DPosS:PosS(2)+(k-1)*DPosS+DimS(2)-1, k) = S;
    end;
end;

*****
*** FileName: SEQSQUARED2.M
*****

function [X]=seqsquared2(TypMov,Nf,DimF,DimS,PosS,DPosS)
% [X]=SEQSQUARED2(TypMov,Nf,DimF,DimS,PosS,DPosS);
%
% Sequence of Nf frames of a squared object.
%
% TypMov Possible types of moving (default = 'H'):
% 'S','H', 'V', 'HV' (Static, Horizontal, Vertical, 45° = Hor.-Ver.).
% Nf      Number of frames (default = 10).
% DimF    Frame size (2D) (default = [80 80]).
% DimS    Squared object size (2D) within each frame (default = [20 20]).
% PosS    Object initial position from the up-left corner of the first frame
%         (default = [20 20]).
% DPosS   Object displacement increment for all directions depending on
%         the type of moving (default = 4).
%         The new position of the squared object at frame f is
%         PosS+(f-1)*DPosS for the specified direction.
%
% X       3D sequence.
%
% example : X=seqsquared2('H',10,[80 80],[20 20],[20 20],4);

% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

if (nargin < 6), DPosS=4; end;
if (nargin < 5), PosS=[20 20]; end;
if (nargin < 4), DimS=[20 20]; end;
if (nargin < 3), DimF=[80 80]; end;
if (nargin < 2), Nf=10; end;
if (nargin < 1), TypMov='H'; end;

% Sequence (3D-signal) size and generation
DimF=DimF(:).'; % forces row-vector
DimX=[DimF, Nf];
X=zeros(DimX);

```

```

% Squared object (2D) size within each frame and generation
DimS=DimS(:).'; % forces row-vector
S=ones(DimS);

% Object initial position from the up-left corner of the first frame
PosS=PosS(:).'; % forces row-vector
% Object displacement increment for all directions
if (DPosS < 0), DPosS=0; end;

TypMov=upcase(TypMov);
if strcmp(TypMov,'S'), DPosS=0; end;

% Validation of frame size respect object size
if ~all((PosS+DPosS.*Nf+DimS) < [DimX(1) DimX(2)]),
    error('Object dimensions and its displacing area are greater than frame dimen
        $y□□□end;

for k=1:Nf
    if strcmp(TypMov,'S'), % Static object
        X(PosS(1):PosS(1)+DimS(1)-1, ...
            PosS(2):PosS(2)+DimS(2)-1, k) = S;

    elseif strcmp(TypMov,'H'), % Horizontal moving
        X(PosS(1):PosS(1)+DimS(1)-1, ...
            PosS(2)+(k-1)*DPosS:PosS(2)+(k-1)*DPosS+DimS(2)-1, k) = S;

    elseif strcmp(TypMov,'V'), % Vertical moving
        X(PosS(1)+(k-1)*DPosS:PosS(1)+(k-1)*DPosS+DimS(1)-1, ...
            PosS(2):PosS(2)+DimS(2)-1, k) = S;

    elseif strcmp(TypMov,'HV') | strcmp(TypMov,'VH') | strcmp(TypMov,'45'), % 45°
        $y□□□□ X(PosS(1)+(k-1)*DPosS:PosS(1)+(k-1)*DPosS+DimS(1)-1, ...
            PosS(2)+(k-1)*DPosS:PosS(2)+(k-1)*DPosS+DimS(2)-1, k) = S;
    end;
end;

*****
*** FileName: SHOWCOEFF.M
*****

function [Lr]=Showcoeff(L,RESC)
% [Lr]=SHOWCOEFF(L,RESC);
%
% Shows the coefficient L of order mnp (3D) as an image deck.
%
% RESC is an optional parameter used for rescaling the gray level.
% The image plot uses the default number of gray levels.
% If you use an empty matrix ([]) for [LOW HIGH], SHOWCOEFF uses
% {min(I(:)) max(I(:))}; the minimum value in I displays as black,
% and the maximum value displays as white.
%
% The optional output parameter Lr is the coefficient reshaped [M N P]
% within a M-by-(N*P) matrix.

% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

if (nargin < 1), error('At least one parameter is required'); end;

[Lx,Ly,Lz]=size(L);
Lr=reshape(L,[Lx,Ly.*Lz]);

```

```

if (nargin > 1),
    imshow(Lr,RESC);
else
    imshow(Lr);
end;

if (nargout < 1), clear Lr; end;

*****
*** FileName: SHOWHT1D.M
*****

function showht1d(X,Xe,Kn,Ktp)
% SHOWHT1D(X,Xe,Kn,Ktp);
%
% Shows the 1D projection from the 3D Hermite Transform in multiple windows.
%
% X    original 3D sequence.
% Xe   synthesized 3D sequence from 1D projection (optional).
% Kn   1D coefficients from the 3D-HT (optional).
% Ktp  1D optimal angles (optional).

* C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

% X
hfig=gcf; figure(hfig); showcoeff(X);
if (nargin > 1), % Xe
    hfig=hfig+1; figure(hfig); showcoeff(Xe);
end;
if (nargin > 2), % Kn
    hfig=hfig+1; figure(hfig); subplot(311); showcoeff(Kn(:,:,,1)); subplot(312)
        $y{}{}{}end;
if (nargin > 3), % Ktp
    hfig=hfig+1; figure(hfig); subplot(211); showcoeff(Ktp(:,:,,1)); subplot(212)
        $y{}{}{}end;

*****
*** FileName: SHOWHT3D.M
*****

function Showht3d(Yt,Ord)
% SHOWHT3D(Yt,Ord);
%
% Shows the 3D sequence Hermite Transform.
%
% Yt   is the 4D array that forms the 3D coefficient set.
% Ord  is the column-wise vector for the orders of Yt.

% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

[Ordx,Ordy]=size(Ord);
S=cell(sqrt(Ordx));

[Lx,Ly,Lz]=size(Yt(:,:,,1));

H=[];
V=[];

```

```

for k=1:OrdX,
    if rem(k,S)~=0,
        H=[H,reshape(Yt(:,:,k),[Lx,Ly.*Lz])];
    else
        V=[V;H];
        H=[];
    end;
end;

imshow(V,[]);

*****
*** FileName: SHT3DENER.M
*****

function [E]=sht3dener(Yt)
% [E]=SHT3DENER(Yt);
%
% Sequence Hermite Transform (3D) Energy.
%
% The energy is computed as sum[k=0,..,M](Yt(:,:,k+1)).
%
% Yt    are the 3D polynomial coefficients within a 4D array.
%
% E     is the 3D array energy of each element of Yt.
%
% For example, E=SHT3DENER(Yt(:,:,,1)); computes the puntual energy for the
% coefficient L000 as L000.^2. The cumulative energy is performed over the sum
% of the 3D-array E as sum(sum(sum(E)));

* C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

if (nargin < 1), error('At least one parameter is requiered'); end;

%if (rem(ndims(Yt),4)~=0),
%    error('The coefficients Yt must be stored as an 4D array of size [M N P C]')
%    $y[]%end;

% Size of each 3D-coefficient
[Yx,Yy,Yz,Yc]=size(Yt);

E=zeros(Yx,Yy,Yz);
for k=1:Yc,
    E=E+Yt(:,:,k).^2;
end;

*****
*** FileName: SHTA3D.M
*****

function [Yt,Ord]=shta3d(X,fname,MaxOrd,N_ht,M_ht,T_ht)
% [Yt,Ord]=SHTA3D(X,fname,MaxOrd,N_ht,M_ht,T_ht);
%
%
% Sequence Hermite Transform (HT) Analysis for 3D.
%
%
% X      is the 3D (array) input sequence of dimensions M-by-N-by-P.
% fname  is the MAT-file name to which the data are saved.
%        It must be specified without extension. If set to 'null', data
%        are not saved (default: 'ht3d').

```

```

% MaxOrd is the maximum expansion order. It means that 'sum' of coefficients
% through all dimensions does not exceed the value MaxOrd. For instance,
% if MaxOrd is equal to two then only next coefficients are computed:
% L000; L100, L010, L001; L200, L110, L020, L101, L011, L002.
% The sum m+n+p for the coefficient Lmnp doesn't exceed MaxOrd (base 0).
% (default: 2). If MaxOrd is set to -1 it means that all coefficients
% will be computed, i.e. if MaxOrd=-1 then will be 27 3D-coefficients.
% N_ht is the grade of the hermite polynomials expansion.
% (default: 2).
% M_ht is the window length in all directions: horizontal, vertical, and temp
% directions. This one must be even due to the window goes
from 0 to M.
% (default: 2).
% T_ht is the sampling time. (default: 2).
%
% Yt are the 3D polynomial coefficients within a 4D array.
% Ord is a matrix with the orders in column-wise for each 3D coefficient.
% This means each row of Ord corresponds to the index for the 4th dimensi
% of the array Yt which is equal to the coefficient of the
order specifie by Ord(index). For instance, if Ord(3,:)= [0 1
0], then the coefficient is equal to the element Yt(:, :, 3).
%
% Note 1: Parameters N_ht, M_ht, and T_ht must be power of two,
% and T_ht <= M_ht <= N_ht.
%
% Note 2: All output parameters are saved to file fname.
%
% Analysis
% X(3D) ----> [ D ]----> Y ----> ( T_ht Down)----> Yt(4D)
%
% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

if (nargin < 6), T_ht=2; end;
if (nargin < 5), M_ht=2; end;
if (nargin < 4), N_ht=2; end;
if (nargin < 3), MaxOrd=2; end;
if (nargin < 2), fname='ht3d'; end;
if (nargin < 1), error('At least one parameter is requiered'); end;

if (rem(ndims(X),3)~=0),
    error('The sequence X must be stored as an 3D array of size [M N P]');
end;

if (T_ht > M_ht), error('Parametr T_ht must be lower than or equal to M_ht'); en
    if (M_ht > N_ht), error('Parametr M_ht must be lower than or equal
to N_ht'); en
%> Checking for powers of two
Err_Str=['Parameters N_ht, M_ht, and T_ht must be power of two'];

N=T_ht;
NN=log(N)/log(2.0);
if (rem(NN,fix(NN))~=0), % N is not power of 2
    error(Err_Str);
else % N is power of 2
    N=2^nextpow2(N);
end;

N=M_ht;
NN=log(N)/log(2.0);
if (rem(NN,fix(NN))~=0), % N is not power of 2
    error(Err_Str);

```



```

function [ANGS]=shtang3d(M_ht)
% [ANGS]=SHTANG3D(M_ht);
%
% Sequence Hermite Transform Angles for 3D-lattice.
%
% Gets the angles for 3D- lattice as a matrix where the first column are the ang
% sy[]% THETA and their corresponding angles PHI are within the second
column.
%
% This angle lattice is useful for calculating the 1D-HT projection from the 3D-
% sy[]% The number of angles ANGS is computed through the length of the 3D
window
% (receptive field) which have the same length in both spatial and temporal axes
% sy[]%
% M_ht is the window length in all dirtections: horizontal, vertical, and tempo
% sy[]% directions. This one must be even due to the window goes
from 0 to M.
% (default: 2).
%
% ANGS is the ((M_th+1)^3-M_th-1)-by-2 matrix with column angles THETA and PHI
% respectively. THETA=Angles(:,1); PHI=Angles(:,2); and THETA is the
angle% with the X-axis of the 3D vector projection over the plane XY and
PHI
% is the angle with the Z-axis of that 3D vector. The angles are in radian
% sy[]%
% Note 1: Parameter M_ht must be power of two.
% N_ht is the grade of the hermite polynomials expansion
% (on the whole, default: 2).
% Parameter M_ht must be selected so that it is MIN(M_ht,N_ht).
%
% Note 2: These 3D-lattice angles are used by the function SHTANGFN to get the
% angle functions Gl,m-1(theta) and Gm,n-m(phi).

% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

if (nargin < 1), M_ht=2; end;

%%% Checking for powers of two
Err_Str={'Parameters M_ht must be power of two'};

N=M_ht;
NN=log(N)./log(2.0);
if (rem(NN,fix(NN))~=0), % N is not power of 2
    error(Err_Str);
else % N is power of 2
    % N=2^nextpow2(N);
end;

clear N NN;

%%% Begins algorithm

% Old version
if 0,
x=-M_ht/2:M_ht/2; % assumes a centered window
[X,Y]=meshgrid(x); % spatial orientation
T=atan2(Y(:),X(:)); % valid angles on plane XY
Tu=Getwangzero([T(:), sqrt(X(:).^2+Y(:).^2)]); % drop out the origin [0 0 0]
R=Tu(:,2); % projections above plane XY
[Z,Rv]=meshgrid(x,R); % temporal and XY-projection orientations
P=atan2(Rv,Z); % temporal orientation

```

```

[Z,Th]=meshgrid(x,Tu(:,1)); % gets equal number of angles theta as phi angles
phi=P.'; theta=Th.'; % PHI and THETA angles
theta=theta(:); phi=phi(:);
ANGS=[theta(:),phi(:)]; % 3D-angle lattice
end;

: New version
if 1,
nw=M_ht+1;
[X,Y] = meshgrid(0:(nw-1)/2);
theta = atan2(Y(:),X(:));
theta = unique(sort([-theta,theta]));
R = unique(sort(sqrt(X(:).^2+Y(:).^2)));
[Z,R] = meshgrid(1:(nw-1)/2,R);
phi = atan2(R(:),Z(:));
phi = unique(sort([-phi,phi]));
len_theta = length(theta);
len_phi = length(phi);
[TH,PH] = meshgrid(theta,phi);
angles = [TH(:),PH(:)];
ANGS=angles;
end;

if 0,
x=-M_ht/2:M_ht/2; % assumes a centered window
[X,Y]=meshgrid(x); % spatial orientation
T=atan2(Y(:),X(:)); % valid angles on plane XY
Tu=[T(:), sqrt(X(:).^2+Y(:).^2)]; % drop out the origin [0 0 0]
R=Tu(:,2); % projections above plane XY
[Z,Rv]=meshgrid(x,R); % temporal and XY-projection orientations
P=atan2(Rv,Z); % temporal orientation
[Z,Th]=meshgrid(x,Tu(:,1)); % gets equal number of angles theta as phi angles
phi=P.'; theta=Th.'; % PHI and THETA angles
theta=theta(:); phi=phi(:);
theta = unique(sort([-theta,theta]));
phi = unique(sort([-phi,phi]));
[TH,PH] = meshgrid(theta,phi);
angles = [TH(:),PH(:)];
ANGS=angles; % 3D-angle lattice
end;

*****
*** FileName: SHTANGFN.M
*****

function [ANGSFN,OrdAng]=shtangfn(ANGS,Ord,M_ht)
% [ANGSFN,OrdAng]=SHTANGFN(ANGS,Ord,M_ht)
%
% Sequence Hermite Transform Angle Functions.
%
% ANGS is the ((M_th+1)^3-M_th-1)-by-2 matrix with column angles THETA and
PHI% respectively. THETA=Angles(:,1); PHI=Angles(:,2); and THETA is the
angl % with the X-axis of the 3D vector projection above the plane
XY and PHI
% is the angle with the Z-axis of that 3D vector. The angles are in radia
% This matrix is return by the function SHTANG3D.
% Ord is a matrix with the orders in column-wise for each 3D coefficient.
% This means each row of Ord corresponds to the index for the 4th dimensi
% of the array Yt which is equal to the coefficient of the
order specifie % by Ord(idex). For instance, if Ord(3,:)= [0 1
0], then the coefficient L % is equal to the element Yt(:, :, 3).

```

```

* M_ht is the window length in all directions: horizontal, vertical, and temp
  Sy0000% directions. This one must be even due to the window goes
from 0 to M.
* (default: 2).
%
% ANGSEFN Angle Functions as a 3D array where the 1st and 2nd dimensions has the
  Sy0000% sizes that matrix ANGSEFN, and the size for the 3rd dimension
is equal to Sy0000% 2D combination of the maximum order of the 3D-HT
expansion within matrix Sy0000% i.e., the sum through each row of Ord.
For instance, if the maximum order Sy0000% the expansion is equal to 2 then
G00,G01,G10,G02,G11,G20 are computed a Sy0000% store within the 3rd
dimension of ANGSEFN, so its size is equal to 6 (nu Sy0000% of angle
functions Gm,n-m ).
% OrdAng is a matrix with the orders in column-wise for each angle function along
  Sy0000% the 3rd dimension of array ANGSEFN. For the case shown above
of Gm,n-m, % the associated matrix is [0 0; 0 1; 1 0; 0 2; 1 1; 2 0].
%
% Note 1: Parameter Ord can be a scalar representing the maximum 3D-expansion
% maximum order. For instance, if N_ht=2, then MaxOrd=max(sum(Ord.));
% so that Ord can take the scalar value equal to 2.
%
% Note 2: Parameter M_ht must be power of two.
% N_ht is the grade of the hermite polynomials expansion
% (on the whole, default: 2).
% Parameter M_ht must be selected so that it is MIN(M_ht,N_ht).
%
% Note 3: These angle functions are used by the function SHFIT1D to estimate
% the 1D projection as the best 1D fit.
%
% The angle functions Gm,n-m(x) of order (m,n-m) and angle x are calculated
% on the whole as
%  $G_{m,n-m}(x) = \sqrt{n! / [(n-m)! * m!]} * \cos(x)^m * \sin(x)^{(n-m)}$ 
% For discrete signal the correction amplification factor is given by
%  $\sqrt{1 - 1/M * ((\cos(x))^4 + (\sin(x))^4)}$  for  $\sum(m,n-m) > 1$ 
% and multiplied also by  $\sqrt{1 - 1/M}$  for  $[n \neq 2 * m]$ .
%
% The projection to 1D fit is achieved by equations shown in the paper showed by
  Sy0000% Jean-Bernard Martens, "The Hermite Transform-Theory,"
% IEEE Transactions on ASSP, Vol. 38, NO. 9, pp 1595-1606, September 1990.
%
% The generalized equations for 1D projection are given by:
%
% 
$$K_{n,t,p} = \sum_{m=0, \dots, n} \sum_{l=0, \dots, m} \{ G_{l,m-l}(t) * G_{m,n-m}(p) * L_{l,m-l,n-m} \}$$

%
% for angles t=theta, p=phi and order n.
% and Ll,m-l,n-m are the 3D-HT coefficients up to order n=N
%
% for optimum angles (t,p)=(T,P) the 1D coefficients up to order n=N
% are obtained as:
%
% 
$$K_{n,T,P} = \{ K_{n,t,p} \mid (t,p) \in \arg \max_{(t,p)} \{ \sum_{n=1, \dots, N} \{ (K_{n,t,p})^2 \} \}$$

%
% and the reconstruction of the 3d coefficients is done as:
%
% 
$$L_{l,m-l,n-m} = K_{n,T,P} * G_{l,m-l}(T) * G_{m,n-m}(P)$$

%
% n = 0, ..., N ; m = 0, ..., n ; l = 0, ..., m
%
%
% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

```

```

if (nargin < 3), M_ht=2; end;
if (nargin < 2), error('At least two parameters are requiered'); end;

% 3D-expansion maximum order
MaxOrd=max(sum(Ord. '));

ANGSFN=[];
OrdAng=[];
% Gm,n-m(alfa) ; n=0,...,N , m=0,...,n
for n=0:MaxOrd,
    for m=0:n,
        OrdAng=[OrdAng; [m n-m]]; % order of the angle function

        % the continuous case is used always
        % theta, phi (columns 1 and 2 respectively)
        G=sqrt(combvn(n,m)).*cos(ANGS).^m.*sin(ANGS).^(n-m);

        % NEW VERSION
        % This section is not computed (if with " & 0 ") for better results
        % -----
        % for order 2 the discrete case is introduced since
        % discrete and continuous cases are similar for orders 0,1.
        % G02, G11, G20
        if (n == 2) & 0,
            G=G.*sqrt( 1-(1/M_ht).*(cos(ANGS).^4+sin(ANGS).^4) );
            % G02, G20
            if (m ~= 1),
                G=G.*sqrt( 1-(1/M_ht) );
            end;
        end;
        % -----

        % constructs the 3D array of angle fucntions
        ANGSFN=cat(3,ANGSFN,G);

    end; % for m
end; % for n

*****
*** FileName: SHTDATOS1.M
*****

function shtdatos1
% SHTDATOS1;
%
% Generate some data for tests 1.
%
% T=[1111 1211; 2111 2211; 1121 1221; 2121 2221; 1112 1212; 2112 2212; 1122 1222
%     syooow=permute(reshape(T',[2 2 2 2]),[2 1 3 4]));

T=[1111 1211; 2111 2211; 1121 1221; 2121 2221; 1112 1212; 2112 2212; 1122 1222;
%     syooow=permute(reshape(T',[2 2 2 2]),[2 1 3 4]));

% [Yx,Yy,Yz,Yc]=size(Yt);
% [EL]=sht3dener(Yt(:,:,,2:Yc));

% [Kx,Ky,Kz,Kc]=size(Kn);
% [EK]=sht3dener(Kn(:,:,,2:Kc));

*****
*** FileName: SHTDATOS2.M

```

\*\*\*\*\*

```

function shtdatos2(X)
% SHTDATOS2(X);
%
% Generate some data for tests 2.
%
% X    3D sequence.

% C. Rivero, April 2000.
% Please send any error you find or any comment
% via e-mail to : carlosj@verona.fi-p.unam.mx

% [X]=seqsquared2('S',6); shtrocio2(X);

% Analysis-Synthesis through 3D-HT and 1D projection
[Y1,Ord1]=shta3d(X,'null'); [Kn,Ktp]=shtfit1d(Y1,Ord1); [Y2,Ord2]=shtest3d(Kn,Kt
    $Y0000Ey=[]; OrdEy=[]; for n=0:2, for m=0:n, for l=0:m,
[L1]=getcoeff(Y1,[l,m-1,n-m],0 $Y0000
% Notice that Ord1 <> Ord2 and Ord2 == OrdEy

% Undocument next line for getting LAST VERSION introducing error coefficients
% -----
%Y3=Y2; Y3(:,:,,8)=Y3(:,:,,8)+Ey(:,:,,8); [Xe]=shts3d(Y3,Ord2,size(X));

% FINAL VERSION of this file (SHTDATOS2) without error coefficients
% -----
[Xe]=shts3d(Y2,Ord2,size(X));
[Esnr]=sigerror(X,Xe,'SNR'); [Epsnr]=sigerror(X,Xe,'PSNR');
% -----

Hy=[]; for k=1:10, Hy=[Hy, entrop(Ey(:,:,,k))]; end;
By=mean(Hy).*prod(size(Ey));

% Show results (hard-coded)
fprintf('3D error coefficient order and entropy (bits/pixel)\n');
fprintf('  %g %g %g %15.4f \n',[OrdEy, Hy.].');
fprintf('3D error coefficient total number of bits\n');
fprintf('  %15.4f bits \n',By);
fprintf('3D error coefficient maximum values (for coefficient L020) \n');
fprintf(' Entropy : %15.4f bits/pixel \n',Hy(8));
fprintf('      Bits : %15.4f bits \n',Hy(8).*prod(size(Ey))./10);
fprintf('Error signal measures\n');
fprintf('      SNR = %9.

```

## Bibliografía

- Abramowitz, M. & Stegun, I. (1965). *Handbook of Mathematical Functions*. Dover, New York.
- Bloom, J.A. & Reed, T.R. (1996). A Gaussian derivative-based transform. *IEEE Transactions on Image Processing* **IP-5**.
- Bloom, J.A. & Reed, T.R. (1997). An uncertainty analysis of some real functions for image processing applications. *International Conference on Image Processing SPIE*.
- Boas, R. & Buck, R. (1985). *Polynomial expansions of analytical functions*. Springer, New York.
- Brinker, A.C. & Roufs, J.A.J. (1992). Evidence for a generalized Laguerre Transform of temporal events by the visual system. *Biological Cybernetics* **67**, 395-402.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**, 679-698.
- Daugman, J.G. (1980). Two dimensional spectral analysis of cortical receptive field profiles. *Vision Research* **20**, 847-856.
- Daugman, J.G. (1983). Six formal properties of two-dimensional anisotropic visual filters: Structural principles and frequency/orientation selectivity. *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-13**, 882-887.
- Daugman, J.G. (1988). Complete 2-D discrete Gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-36**, 1169-1179.
- Daugman, J.G. (1989). Entropy reduction and decorrelation in visual coding by oriented neural receptive fields. *IEEE Transactions on Biomedical Engineering* **BME-36**, 107-114.
- Escalante Ramírez, B (1992). *Perceptually-Assessed Digital Processing of Medical Images*. PhD Thesis. Technische Universiteit Eindhoven, Netherlands.
- Escalante Ramírez, B. & Martens, J.-B. (1992). Noise reduction in computerized tomography images by means of polynomial transforms. *Journal on Visual Communication and Image Representation* **3**, 272-285.
- Escalante Ramírez, B. & Martens, J.-B. (1993). Technique for image interpolation using polynomial transforms. *Visual Communication and Image Processing Proc. SPIE* **2094**, 522-528.
- Gabor, D. (1946). Theory of communication. *J. Inst. Electr. Eng.* **93**, 429-457.
- González, R.C. & Woods, R.E. (1992). *Digital Image Processing*. Addison-Wesley, Massachusetts.
- Koenderink, J (1984). The structure of images. *Biological Cybernetics* **50**, 363-370.

- Kunt, M., Ikonomopoulos, A. & Kocher, M. (1985). Second-generation image-coding techniques. *Proceedings of the IEEE* **73**, 549-574.
- Li, H., Lundmark, A. & Forchheimer, R. (1994). Image sequence coding at very low bitrates: A review. *IEEE Transactions on Image processing* **IP-3**, 589-609.
- Marčelja, S. (1980). Mathematical description of the responses of simple cortical cells. *Journal of the Optical Society of America* **70**, 1297-1300.
- Marr, D. & Hildreth, E. (1980). A theory of edge detection. *Proceedings of the Royal Society of London B* **207**, 187-217.
- Martens, J.-B. & Majoor, G.M.M. (1989). The perceptual relevance of scale space image coding. *Signal Processing* **17**, 353-364.
- Martens, J.-B. (1990a). The Hermite transform – Theory. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-38**, 1595-1606.
- Martens, J.-B. (1990b). The Hermite transform – Applications. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **ASSP-38**, 1607-1618.
- Martens, J.-B. (1990c). Deblurring digital images by means of polynomial transforms. *Computer Vision, Visual Processing, Graphics, and Image Processing* **50**, 157-176.
- Paraskevopoulos, P.N. (1985). Systema analysis and synthesis via orthogonal polynomial series and Fourier series. *Mathematics and Computers in Simulation* **27**, 453-469.
- Podilchuk, C.I., Jayant, N.S. & Farvardin, N. (1995). Three-diemnsional subband coding of video. *IEEE Transactions on Image processing* **IP-4**, 125-139.
- Pollen, D.A. & Ronner, S.F. (1981). Phase relationships between adjacent simple cells in the visual cortex. *Science* **212**.
- Sakitt, B. & Barlow, H. (1982). A model for the economical encoding of the visual image in cerebral cortex. *Biological Cybernetics* **43**, 97-108.
- Shulman, G. & Mulvanny, P. (1983). Discrimination of spatiotemporal patterns: The role of sustained and transient mechanisms. *Perception* **12**, 531-543.
- Stremler, F.G. (1982). *Introduction to Communication Systems, 2nd. ed.*. Addison-Wesley, Massachusetts.
- Szegö, G. (1959). *Orthogonal Polynomials*, Colloquium Publications, American Mathematical Society, Providence, RI.
- Vetterli, M. & Kovačević, J. (1995). *Wavelets and Subband Coding*. Prentice Hall, New Jersey.
- Wilson, R. & Granlund, G. (1984). The uncertainty principle in image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-6**, 758-767.

- Young, R. (1985). The Gaussian derivative theory of spatial vision: Analysis of cortical cell receptive field line-weighting profiles. *General Motors Research Technical Report GMR-4920*.
- Young, R. (1986). Simulation of human retinal function with the Gaussian derivative model. *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog., June 1986*.
- Young, R. (1987). The Gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spatial Vision* **2**, 273-293.