



# UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA

CUANTIFICACIÓN VECTORIAL PARA IMÁGENES CON BASE EN  
LOS ALGORITMOS LBG Y LAMDA

TESIS

PARA OBTENER EL TÍTULO DE  
INGENIERO EN ELECTRÓNICA

PRESENTA

**JUAN GABRIEL ZAMBRANO NILA**

DIRECTOR DE TESIS

**Dr. ENRIQUE GUZMÁN RAMÍREZ**

HUAJUAPAN DE LEÓN, OAXACA. FEBRERO DE 2012



Tesis presentada en febrero de 2012

ante los sinodales:

Dr. Antonio Orantes Molina

M.S.R.C. José Antonio Moreno Espinosa

Dr. Rosebet Miranda Luna

Director de tesis:

Dr. Enrique Guzmán Ramírez



# Dedicatoria

A mis padres Antonio y Leonarda, a pesar de que ya no están entre nosotros, siguen presentes en mi pensamiento.

A mi hermana Elizabeth porque siempre me ha brindado su cariño y apoyo.

A mi esposa Margarita por su amor, consejos, apoyo moral en momentos difíciles y por alentarme a continuar.

A mis hijas Viviana, María Fernanda, Diana y a la bebita que pronto estará con nosotros; por ser el motor y la inspiración que me motivan a seguir adelante.

Gabriel.



# Agradecimientos

A mi director de tesis, el Dr. Enrique Guzmán Ramírez por su amistad, conocimientos compartidos, sugerencias y sobre todo por su paciencia para poder llevar a buen término este trabajo.

Al Dr. Antonio Orantes Molina, al M.S.R.C. José Antonio Moreno Espinosa y al Dr. Rosebet Miranda Luna; sinodales de este trabajo de tesis, por el tiempo dedicado en la revisión, así como, por los comentarios y las correcciones hechas con el fin de mejorarlo.

A los maestros, compañeros de generación y amigos que de alguna forma me motivaron a culminar este trabajo.

A todas y cada una de las personas que me apoyaron durante mi formación profesional y las que estuvieron involucradas, directa o indirectamente, con la realización de este trabajo.

Gabriel.



# Resumen

El presente trabajo propone la unión del algoritmo LBG, propuesto por Y. Linde, A. Buzo y R. Gray, y el algoritmo de aprendizaje para el análisis de datos multi-variables (LAMDA, *Learning Algorithm for Multivariate Data Analysis*), para la creación de un esquema alternativo para la cuantificación vectorial de imágenes. El esquema expuesto consta de dos bloques, el primero está constituido por dos fases y el segundo por cuatro fases.

En el primer bloque se genera un libro de códigos mediante el algoritmo LBG, este debe ser normalizado para que el algoritmo LAMDA lo pueda utilizar, a la nueva representación se le denomina libro de códigos LAMDA.

La primera fase del segundo bloque es normalizar los vectores de entrada, la segunda fase calcula el grado de adecuación marginal (MAD, *Marginal Adequacy Degree*) que existe entre los elementos de cada vector de entrada con los elementos de cada uno de los vectores de códigos del libro de códigos LAMDA. Para calcular los MADs se pueden usar distribuciones de probabilidad o una medida de similitud.

En la tercera fase se calcula el grado de adecuación global (GAD, *Global Adequacy Degree*) de los MADs anteriormente calculados, para ello se utiliza una función lineal convexa, operadores de agregación (T-normas y S-conormas) y un nivel de exigencia específico.

En la última fase se usa la regla de máxima adecuación cuando son distribuciones de probabilidad y la regla de mínima adecuación si es la medida de similitud, para calcular un conjunto de índices de los vectores de códigos, a los cuales pertenece cada vector de entrada. El conjunto de índices representa el resultado del proceso de cuantificación vectorial propuesto, aplicado sobre una imagen determinada.



# Abstract

This thesis proposes the union of the LBG algorithm proposed by Y. Linde, A Buzo and R. Gray, and the Learning Algorithm for Multivariate Data Analysis (LAMDA), for the creation of an alternative scheme for vector quantization of images. The described scheme consists of two blocks. The first one consists of two phases, and the second of four phases.

The first block generates a codebook using the LBG algorithm. It must be normalized to be used by LAMDA algorithm, and the new representation is called LAMDA codebook.

The first phase of the second block is to normalize the input vectors. The second phase calculates the marginal adequacy degree (MAD) between the elements of each input vector with the elements of each code vector of the LAMDA codebook. To calculate MADs Probability distributions or a similarity measure can be used.

The third phase calculates the global adequacy degree (GAD) of the MADs previously calculated. For this purpose a linear convex function, aggregation operators (T-norms and S-conorms) and a specific exigency level were used.

In the last phase, maximum adequacy rule is used when selecting a probability distribution; and the minimum adequacy rule is used when selecting the similarity measure, to calculate a set of indices of the code vectors, to which every input vector belongs. The index set is the result of the proposed vector quantization process applied to a particular image.



# Índice general

Dedicatoria	v
Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice general	xiii
Índice de figuras	xvii
Índice de tablas	xix
Acrónimos	xxi
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	3
1.2. Justificación . . . . .	4
1.3. Objetivo general y metas . . . . .	4
1.3.1. Objetivo general . . . . .	4
1.4. Contribuciones . . . . .	5
1.5. Descripción del desarrollo del esquema VQ-LAMDA . . . . .	6
1.5.1. Descripción del problema . . . . .	6
1.5.1.1. Especificaciones . . . . .	7
1.5.1.1.1. Generación del libro de códigos LAMDA . . . . .	7
1.5.1.1.2. Esquema VQ-LAMDA . . . . .	7

1.5.2.	Modelado o representación matemática . . . . .	8
1.5.3.	Diseño del esquema . . . . .	8
1.5.4.	Verificación del esquema . . . . .	9
1.5.5.	Estimación de la complejidad computacional . . . . .	9
1.5.6.	Implementación del esquema propuesto . . . . .	9
1.5.7.	Comprobación experimental del esquema propuesto . . . . .	10
1.5.8.	Documentación . . . . .	10
1.6.	Organización de la tesis . . . . .	10
<b>2.</b>	<b>Marco teórico y estado del arte</b>	<b>11</b>
2.1.	Compresión de imágenes . . . . .	11
2.1.1.	Redundancia inter-pixel . . . . .	11
2.1.2.	Redundancia psico-visual . . . . .	12
2.1.3.	Redundancia de codificación . . . . .	12
2.2.	Esquema de un compresor de imágenes . . . . .	12
2.2.1.	Compresión sin pérdidas y con pérdidas . . . . .	13
2.3.	Cuantificación . . . . .	14
2.4.	Cuantificación escalar . . . . .	15
2.4.1.	Cuantificación escalar uniforme . . . . .	15
2.4.2.	Cuantificación escalar no uniforme . . . . .	16
2.5.	Cuantificación vectorial . . . . .	16
2.5.1.	Desempeño de un compresor de imágenes . . . . .	19
2.6.	Estado del arte . . . . .	20
<b>3.</b>	<b>Algoritmos LBG y LAMDA</b>	<b>27</b>
3.1.	Algoritmo LBG . . . . .	27
3.2.	Algoritmo de aprendizaje para el análisis de datos multi-variables . . . . .	29
3.2.1.	Funcionamiento de la metodología LAMDA . . . . .	30
3.2.1.1.	Grado de Adecuación Marginal . . . . .	31
3.2.1.1.1.	Funciones de pertenencia . . . . .	31
3.2.1.2.	Grado de Adecuación Global . . . . .	33
3.2.1.2.1.	Operadores de agregación . . . . .	33
3.2.1.2.2.	T-normas y S-conormas . . . . .	34

---

3.2.1.2.3. Función lineal convexa T-S . . . . .	35
<b>4. Cuantificación Vectorial con base en el Algoritmo LAMDA</b>	<b>37</b>
4.1. Definiciones preliminares . . . . .	37
4.2. Descripción del esquema de codificación VQ-LAMDA . . . . .	38
4.2.1. Generación del libro de códigos-LAMDA . . . . .	39
4.2.2. Esquema VQ-LAMDA . . . . .	40
4.3. Proceso de decodificación . . . . .	42
<b>5. Resultados experimentales</b>	<b>45</b>
5.1. Primer experimento . . . . .	46
5.2. Segundo experimento . . . . .	47
5.3. Tercer experimento . . . . .	51
5.3.1. Complejidad del esquema propuesto . . . . .	51
5.3.1.1. Complejidad en tiempo . . . . .	51
5.3.1.2. Complejidad en espacio . . . . .	54
5.3.2. Cálculo del número y tipo de operaciones del esquema VQ-LAMDA	56
<b>6. Conclusiones y Trabajo a Futuro</b>	<b>59</b>
6.1. Trabajo a futuro . . . . .	61
6.2. Publicaciones generadas . . . . .	62
<b>Bibliografía</b>	<b>63</b>
<b>A. Pseudo-códigos del esquema VQ-LAMDA</b>	<b>69</b>



# Índice de figuras

1.1. Diagrama a bloques del esquema de cuantificación vectorial VQ-LAMDA. . . . .	9
2.1. Diagrama a bloques de un esquema de compresión de imágenes (Trad. de [44]).	13
2.2. Cuantificador escalar uniforme (Trad. de [40]). . . . .	16
2.3. Cuantificador escalar no uniforme (Trad. de [40]). . . . .	17
2.4. Estructura básica de un cuantificador vectorial. . . . .	19
3.1. Esquema de cuantificación vectorial LBG. . . . .	29
3.2. Estructura básica de la metodología LAMDA. . . . .	32
4.1. Esquema de la generación del libro de códigos LAMDA. . . . .	39
4.2. Diagrama a bloques del proceso de codificación del esquema VQ-LAMDA.	42
5.1. Imágenes de prueba usadas para medir el desempeño del esquema VQ-LAMDA	45
5.2. Imagen Elaine procesada con un libro de códigos de tamaño $64 \times 16$ . . . . .	49
5.3. Imagen Man procesada con un libro de códigos de tamaño $128 \times 16$ . . . . .	49
5.4. Imagen Peppers procesada con un libro de códigos de tamaño $256 \times 16$ . . . . .	50
5.5. Imagen Lena procesada con un libro de códigos de tamaño $512 \times 16$ . . . . .	50



# Índice de tablas

1.1. Lista de aplicaciones para la compresión de imágenes, video y audio [9]. . .	2
3.1. T-Normas y S-Conormas utilizadas en la metodología LAMDA . . . . .	35
4.1. T-Normas y S-Conormas usadas por el esquema VQ-LAMDA . . . . .	41
5.1. Caracterización de las veinte ecuaciones obtenidas. . . . .	47
5.2. Tabla comparativa entre los algoritmos VQ-LAMDA, LBG y FSA-EAM. . .	48
5.3. Complejidad en tiempo y espacio de las mejores configuraciones. . . . .	55
5.4. Operaciones requeridas por los algoritmos LBG, FSA-EAM y VQ-LAMDA para cuantificar un pixel. . . . .	57



# Acrónimos

bpp	bits por píxel
CD-ROM	Disco compacto- memoria de solo lectura (Compact Disc - Read Only Memory)
DCT	Transformada discreta del coseno (Discrete Cosine Transform)
DFT	Transformada discreta de Fourier (Discrete Fourier Transform)
DPI	Puntos por pulgada (Dots per inch)
DWT	Transformada discreta wavelet (Discrete Wavelet Transform)
EAM	Memorias asociativas extendidas (Extend Associative Memories)
FAX	facsímil
FF	Luciérnaga (firefly)
fps	Cuadros por segundo (frames per second)
FSA-EAM	Búsqueda rápida para cuantificación vectorial basado en memorias asociativas extendidas (Fast Search Algorithm Extend Associative Memories)
GAD	Grado de adecuación global (global Adequacy Degree)
GLA	Algoritmo generalizado de Lloyd (Generalized Lloyd Algorithm)
HBMO	Optimización del apareamiento de la abeja de miel (HBMO, Honey Bee Mating Optimization)
HDTV	Televisión de alta definición (High-definition television)
ib	Bloque de la imagen (image block)
iv	Vector de la imagen (image vector)

---

KLT	Transformación Karhunen-Loeve (Karhunen-Loeve Transform)
LAMDA	Algoritmo de aprendizaje para el análisis de datos multi-variables (Learning Algorithm for Multivariate Data Analysis)
LOT	Transformación ortogonal traslapada (Lapped Orthogonal Transform)
mA	Regla de mínima adecuación (minimum Adequacy)
MA	Regla de máxima adecuación (Maximum Adequacy)
MAD	Grado de adecuación marginal (Marginal Adequacy Degree)
MAE	Error absoluto medio (Mean Absolute Error)
ML	Aprendizaje maquina (Machine Learning)
MSE	Error cuadrático medio (Mean Square Error)
NIC	Clase No-Informativa (Non-Informative Class)
PCM	Modulación por pulsos codificados (pulse code modulation)
PNN	Vecinos más cercanos por parejas (Pairwise Nearest Neighbor)
PSNR	Relación señal a ruido (Peak Signal to Noise Ratio)
PSO	Optimización del enjambre de partículas (Particle Swarm Optimization)
RVQ	Cuantificación vectorial residual (Residual Vector Quantization)
S-conorma	Conorma triangular S (S-conorm)
SOM	Mapas Auto-Organizados (Self-Organizing Map)
SQ	Cuantificación escalar (Scalar Quantization)
SWT	transformación wavelet de sub-bandas (Subband Wavelet Transform)
T-norma	Norma triangular (T-norm)
TSVQ	cuantificador vectorial estructurado en árbol (Tree Structured Vector Quantizer)
VQ	Cuantificación vectorial (Vector Quantization)
VQ-LAMDA	Algoritmo para cuantificación vectorial basado en el algoritmo LAMDA

# Capítulo 1

## Introducción

Todos los días, se procesa, manipula, almacena, recupera y se transmite por medios digitales una enorme cantidad de información. Debido a que la mayor parte dicha información son imágenes y video, se ha estimulado el interés en la investigación de algoritmos para la compresión de imágenes que usualmente explotan las redundancias de codificación, inter-píxel y psico-visual, inherentes a las imágenes para reducir la tasa de bits [16].

La compresión de imágenes desempeña un papel importante en diversas aplicaciones multimedia (imagen, video y audio), tanto para la transmisión como para el almacenamiento en bases de datos, tales como videoconferencias, detección automática remota (el uso de imágenes satelitales para pronostico meteorológico), documentos e imágenes médicas, librerías digitales, redes de computadoras, televisión, transmisión facsímile (FAX), imágenes multi-espectrales, huellas dactilares y dibujos [16], [34].

La principal ventaja de la compresión es que reduce los requerimientos de almacenamiento de datos y también reduce el costo de las comunicaciones al transmitir grandes volúmenes de datos, utilizando eficientemente el ancho de banda disponible en los sistemas de transmisión. Por ejemplo, si se desea almacenar video a 30 cuadros por segundo con una resolución de  $720 \times 480$ , se generan 20.736 Mbytes/s de datos, de esta manera sólo se podrían almacenar 31 segundos en un CD-ROM de 650 Mbytes de capacidad, comprimiendo los datos se puede incrementar la capacidad de almacenamiento a 74 minutos. En la Tabla 1.1, se enlistan algunas de las aplicaciones multimedia más representativas, así como sus correspondientes transferencias de datos antes y después de comprimir, esta información da una idea real de la importancia que tiene la compresión para la transmisión de los datos [1], [9].

Tabla 1.1: Lista de aplicaciones para la compresión de imágenes, video y audio [9].

Aplicación	Transferencia de datos	
	Sin comprimir	Comprimido
Voz 8 kmuestras/s, 8 bits/muestra	64 kbps	2 - 4 kbps
Video en camara lenta (10 fps) tamaño del cuadro $176 \times 120$ , 24 bits/pixel	5.07 Mbps	8 - 16 kbps
Conferencia de audio 8 kmuestras/s, 16 bits/muestra	128 kbps	6 - 64 kbps
Conferencia de video (15 fps) tamaño del cuadro $352 \times 240$ , 24 bits/pixel	30.41 Mbps	64 - 768 kbps
Audio digital (estereo) 44.1 kmuestras/s, 16 bits/muestra	1.5 Mbps	128 - 768 kbps
Transferencia de archivos de video (15 fps) tamaño del cuadro $352 \times 240$ , 24 bits/pixel	30.41 Mbps	384 kbps
Video digital en CD-ROM (30 fps) tamaño del cuadro $352 \times 240$ , 24 bits/pixel	60.83 Mbps	1.5 - 4 Mbps
Emisión de video (30 fps) tamaño del cuadro $720 \times 480$ , 24 bits/pixel	248.83 Mbps	3 - 8 Mbps
HDTV (59.94 fps) tamaño del cuadro $1280 \times 720$ , 24 bits/pixel	1.33 Gbps	20 Mbps

Las señales multimedia se adaptan a la compresión debido a los siguientes factores [9], [44]:

- Existe una considerable cantidad de información estadística en la señal.
  - Cuando en una imagen o un cuadro de video existe abundante correlación entre píxeles vecinos, se le conoce como correlación espacial.
  - Cuando en los datos adquiridos por multiples sensores (ej. imágenes satelitales) existe cuantiosa correlación entre muestras, se le denomina correlación espectral.
  - Cuando hay una cantidad significativa de correlación entre muestras en diferentes segmentos de tiempo (ej. video), a esta se le refiere como correlación temporal.

- Hay una cantidad considerable de información en la imagen que es irrelevante desde el punto de vista perceptivo. Esto se debe a que el sistema perceptivo del ser humano es sustancialmente menos sensible a cambios rápidos en la variación de color de la imagen que a los cambios de intensidad.
- Algunos de los datos tienden a tener características de alto nivel que son redundantes a través del espacio y el tiempo, es decir, los datos son de naturaleza fractal.

La importancia de la cuantificación en la compresión de imágenes radica en el hecho de que es en este proceso donde se lleva a cabo la compresión de los datos, aunque ésta se logra por que existe pérdida de información, ya que por definición la cuantificación es un proceso con pérdidas desde que es un mapeo de un conjunto grande de datos a un conjunto de datos más pequeño.

## 1.1. Planteamiento del problema

Como se describió anteriormente las aplicaciones multimedia en cualquiera de sus ámbitos requieren un amplio ancho de banda y gran espacio para su almacenamiento, esto se debe a que este tipo de aplicaciones maneja gran cantidad de información. Una de las posibles soluciones a este problema se encuentra en la compresión de la información, buscando que se reduzcan los requerimientos en su procesamiento, almacenamiento y transmisión.

Considérese la transmisión de imágenes facsímile. En la mayoría de maquinas facsímile, el documento es escaneado y digitalizado. Una página de  $8.5 \times 11$  pulgadas es comunmente escaneada a 200 dpi, por lo tanto, se obtiene un archivo de 3.74 Mbits. La transmisión de estos datos a través de un modem a 14.4 kbits/s requiere aproximadamente de 5.62 minutos. Comprimiendo los datos, el tiempo de transmisión puede reducirse a 17 segundos. Esto trae como resultado un ahorro sustancial en costos de transmisión [9].

Aunque los sistemas computacionales actuales poseen gran capacidad de procesamiento, es necesaria la optimización de los algoritmos encargados del tratamiento de imágenes y video para obtener un mejor desempeño por parte de estos.

## 1.2. Justificación

En el diseño de un sistema de compresión de imágenes se deben considerar cuatro parámetros esenciales: la relación de compresión obtenida, la distorsión que genera el proceso de compresión sobre la imagen recuperada, la velocidad de procesamiento y recursos requeridos por el sistema.

Los algoritmos de compresión de imágenes se clasifican en dos tipos básicos: compresión sin pérdida de información y compresión con pérdida de información. La cuantificación vectorial es parte esencial de un esquema de compresión con pérdidas, este método toma ventaja de la redundancia de codificación, redundancia inter-píxel y redundancia psico-visual, que en conjunto permiten alcanzar tasas de compresión elevadas a expensas de sufrir una pérdida de información de la imagen original.

Por otra parte, el algoritmo LAMDA ha demostrado ser un clasificador competente en el reconocimiento de patrones, tanto para datos cuantitativos como cualitativos. Además, es fácilmente adaptable a diversas aplicaciones.

Con el propósito de explorar nuevas alternativas para la cuantificación vectorial, el presente trabajo propone un esquema de **Cuantificación Vectorial para Imágenes con base en los Algoritmos LBG y LAMDA**, la implementación de dicho esquema permitirá determinar que tanto mejora la relación señal a ruido (PSNR, *Peak Signal-to-Noise Ratio*) obtenida a partir de la nueva representación de la información generada por el esquema propuesto y además permitirá evaluar su eficiencia en términos de complejidad computacional (en tiempo y en espacio) con respecto a los algoritmos LBG y FSA-EAM.

## 1.3. Objetivo general y metas

### 1.3.1. Objetivo general

Diseño e implementación de un método de cuantificación vectorial para imágenes con base en los algoritmos LBG y LAMDA. A continuación se exponen las metas programadas para cumplir con el objetivo planteado:

- Implementar el algoritmo LBG.
- Implementar el algoritmo LAMDA.

- Integración de ambos algoritmos haciendo las adecuaciones necesarias para la cuantificación vectorial.
- Comparar los resultados obtenidos por el esquema propuesto con los resultados generados por los algoritmos LBG y FSA-EAM.

## 1.4. Contribuciones

La principal contribución de este trabajo, es que se demostró en forma práctica que el algoritmo LAMDA puede usar medidas de similitud en la etapa donde se calculan grados de adecuación marginal, además se propuso el uso de un operador de agregación en la etapa donde se calculan los grados de adecuación global, por último se determino la regla de adecuación mínima. A continuación se detalla cada una de las contribuciones mencionadas.

1. Aunque en la especificación de LAMDA descrita en la sección 3.2, se menciona que no utiliza medidas de similitud para calcular los MAD, desde que sólo usa distribuciones de probabilidad difusa. Se ha propuesto usar la medida de similitud, distancia city-block.

La medida de similitud métrica de Minkowski es una generalización de la distancia Euclidiana.

$$M = \left( \sum_{i=1}^n |\xi_i - \eta_i|^\beta \right)^{\frac{1}{\beta}}, \beta \in \mathfrak{R} \quad (1.1)$$

La distancia denominada city-block se obtiene cuando  $\beta = 1$  [29].

$$M = \sum_{i=1}^n |\xi_i - \eta_i| \quad (1.2)$$

Adaptado al esquema VQ-LAMDA.

$$MAD(x_i^j / c_i^l) = |x_i^j - \rho_i^l| \quad (1.3)$$

Es importante mencionar que cuando se usa la medida de similitud city-block, no es necesaria la normalización del libro de códigos y de los vectores de entrada.

2. En este trabajo se propuso el uso del operador de agregación extendido suma, para calcular las GADs. En lugar de una T-norma y su dual la S-conorma. También se implemento el operador de agregación media aritmética (promedio), propuesto en [3]. La razón por la cual se seleccionaron estos operadores, es porque son operadores en los cuales se realizan menor número de operaciones, en comparación con los operadores de agregación Lukasiewicz, Yager y Hammacher, que son los que se sugiere usar en la especificación de LAMDA.

A continuación se define una función de agregación extendida.

**Definición 1 *Función de agregación extendida:*** es una función  $I = [a, b] \subseteq [-\infty, \infty]$ ,  $EA : \bigcup_{n \in \mathbb{N}} I^n \rightarrow I$  tal que para todo  $n > 1$ ,  $EA^{(n)} = EA|_{I^n}$  es una función de agregación  $n$ -aria y  $EA^{(1)}$  es la identidad en  $I$  [32].

Algunos ejemplos de funciones de agregación extendida en  $I$ , son la suma, el producto, la media aritmética, el mínimo y el máximo [32].

3. Cuando se utilizo la medida de similitud city-block, fue necesario determinar la regla de mínima adecuación ( $mA$ , *minimum Adequacy*), la cual se especifica como:

$$mA = \min(GAD_{c_1}(\mathbf{x}^j), GAD_{c_2}(\mathbf{x}^j), \dots, GAD_{c_N}(\mathbf{x}^j)) \quad (1.4)$$

## 1.5. Descripción del desarrollo del esquema VQ-LAMDA

Para diseñar el esquema VQ propuesto se usó el enfoque sistemático mencionado en [39], que está formado por ocho etapas. A continuación se describe cada una de ellas.

### 1.5.1. Descripción del problema

En la introducción de este capítulo se expuso la importancia que tiene la compresión de imágenes para que la información que manejan las aplicaciones multimedia pueda ser procesada, manipulada, almacenada, recuperada y transmitida. En la Tabla 1.1 se dieron ejemplos reales de la cantidad de recursos que se utilizan para la transmisión de dicha

información. Por lo que es importante explotar los tipos de redundancias implícitas en las imágenes, además de que en la sección 2.5 se explica en concreto el proceso de cuantificación vectorial que es el tema sobre el que trata el esquema propuesto.

#### 1.5.1.1. Especificaciones

A continuación se describen las especificaciones de la generación del libro de códigos y del esquema VQ-LAMDA.

##### 1.5.1.1.1. Generación del libro de códigos LAMDA

- **Entrada:** La imagen que sirve como conjunto de entrenamiento, es una imagen en escala de grises, resolución  $512 \times 512$ , 8 bits por píxel.
- **Generador de libro de códigos:** Algoritmo LBG.
- **Normalización:** El libro de códigos se debe normalizar para que pueda ser utilizado por las distribuciones de probabilidad difusa, en el caso de la medida de similitud no es necesario este proceso.
- **Salida:** Libro de códigos LAMDA

##### 1.5.1.1.2. Esquema VQ-LAMDA

- **Entradas:** El esquema propuesto recibe las siguientes entradas:
  1. El libro de códigos LAMDA
  2. La imagen que va a ser cuantificada, debe ser una imagen en escala de grises, resolución  $512 \times 512$ , 8 bits por píxel.
    - La imagen a ser procesada se debe normalizar para que pueda ser utilizada por las distribuciones de probabilidad difusa, en el caso de la medida de similitud no es necesario este proceso.
- **Clasificador:** Algoritmo LAMDA, especificando la configuración deseada.
  1. Distribución de probabilidad, nivel de exigencia, operadores de agregación (T-norma y S-conorma) y regla de máxima adecuación.

2. Medida de similitud, nivel de exigencia, operadores de agregación (T-norma y S-conorma) y regla de mínima adecuación.

- **Salida:** Conjunto de índices de la imagen cuantificada.

### 1.5.2. Modelado o representación matemática

En el Capítulo 3 se dan los fundamentos teóricos y matemáticos que se necesitan conocer para la comprensión del esquema propuesto, a continuación se muestran las expresiones matemáticas de las configuraciones que tuvieron un mejor desempeño y que son usadas en el esquema propuesto.

$$indice = max \left( \prod_{i=1}^n (\rho_i^l)^{x_i^j} (1 - \rho_i^l)^{(1-x_i^j)} \right) \quad (1.5)$$

$$indice = max \left( \prod_{i=1}^n \frac{(\rho_i^l)^{x_i^j} (1 - \rho_i^l)^{(1-x_i^j)}}{(x_i^j)^{x_i^j} (1 - x_i^j)^{(1-x_i^j)}} \right) \quad (1.6)$$

$$indice = max \left( \frac{1}{n} \sum_{i=1}^n \frac{(\rho_i^l)^{x_i^j} (1 - \rho_i^l)^{(1-x_i^j)}}{(x_i^j)^{x_i^j} (1 - x_i^j)^{(1-x_i^j)}} \right) \quad (1.7)$$

$$indice = max \left( \sum_{i=1}^n \frac{(\rho_i^l)^{x_i^j} (1 - \rho_i^l)^{(1-x_i^j)}}{(x_i^j)^{x_i^j} (1 - x_i^j)^{(1-x_i^j)}} \right) \quad (1.8)$$

$$indice = min \left( ((0.5)min |x_i^j - \rho_i^l|) + ((0.5)max |x_i^j - \rho_i^l|) \right) \quad (1.9)$$

donde  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, M$  y  $l = 1, 2, \dots, N$ .  $x_i^j$  representa los elementos de los vectores de la imagen y  $\rho_i^l$  representa los elementos de los vectores de códigos.

### 1.5.3. Diseño del esquema

En el Capítulo 4 se da una explicación sobre las fases que se deben seguir para llevar a cabo la cuantificación vectorial con el esquema VQ-LAMDA, a continuación se muestra un diagrama a bloques.

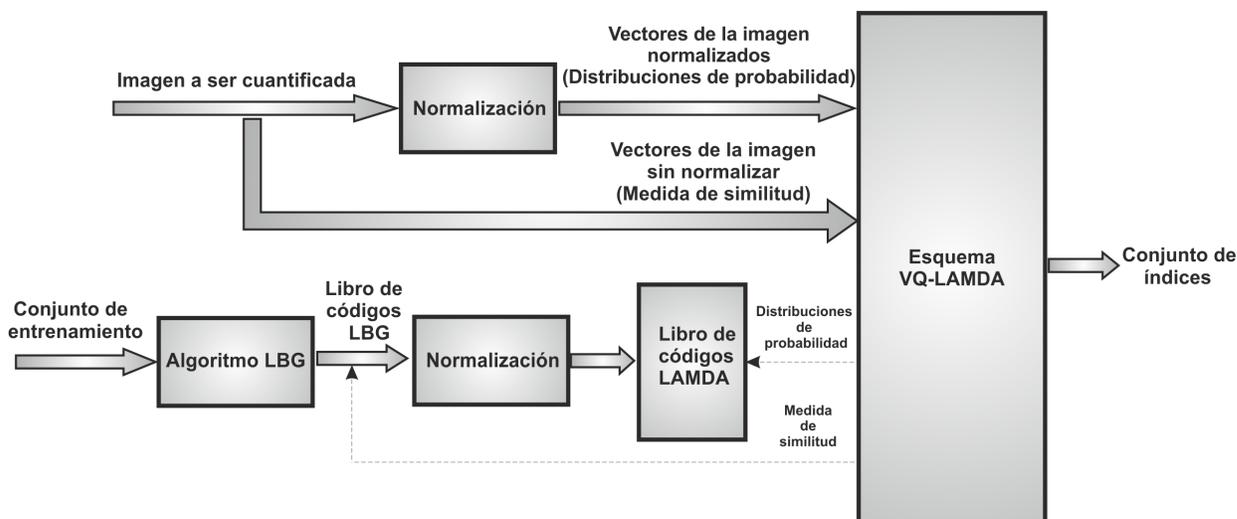


Figura 1.1: Diagrama a bloques del esquema de cuantificación vectorial VQ-LAMDA.

#### 1.5.4. Verificación del esquema

En el capítulo 5 en la sección 5.1 se muestran los resultados de la caracterización que se hizo de todas las posibles configuraciones que se pueden formar al combinar las distribuciones de probabilidad y la medida de similitud junto con los operadores de agregación (T-normas y S-conormas), con ello se obtienen cinco configuraciones que son las que tuvieron mejor desempeño para ser usadas en cuantificación vectorial.

#### 1.5.5. Estimación de la complejidad computacional

En la sección 5.3 del capítulo 5 se calculó la complejidad espacial y temporal, además se calculó el número y tipo de operaciones que utiliza el esquema propuesto en sus diferentes configuraciones.

#### 1.5.6. Implementación del esquema propuesto

Para la implementación del esquema propuesto se utilizó el lenguaje de programación MATLAB.

### 1.5.7. Comprobación experimental del esquema propuesto

En la sección 5.2 se compara el desempeño de VQ-LAMDA con respecto a los algoritmos LBG y FSA-EAM, a ambos algoritmos los supero en términos de la medida de distorsión PSNR, aunque en términos de complejidad espacial y temporal se vio superado.

### 1.5.8. Documentación

En el contenido de este documento de tesis, se encuentra toda la información recopilada para la realización de este esquema, además también se incluyen los resultados recabados de la implementación práctica del algoritmo propuesto.

## 1.6. Organización de la tesis

El presente trabajo de tesis esta estructurado de la siguiente forma:

En este capítulo han sido presentados: el planteamiento del problema, la justificación, los objetivos y las contribuciones de este trabajo, así como un resumen del esquema de cuantificación propuesto. El resto del trabajo está organizado como sigue:

El imprescindible marco teórico se presenta en el capítulo 2, donde se dan conceptos y definiciones relacionadas con la cuantificación vectorial y su importancia en esquemas de compresión de imágenes. Este capítulo también incluye el estado del arte, donde se exponen trabajos relacionados con el tema contemplado en este trabajo de tesis.

El capítulo tres abarca los conceptos, fundamentos teóricos y matemáticos sobre los algoritmos LBG y LAMDA.

En el capítulo 4 se describe el esquema de cuantificación vectorial propuesto, especificando las fases que se deben seguir para su implementación.

El capítulo 5 se exponen los resultados obtenidos, el análisis de complejidad en tiempo y espacio, así como, el número y tipo de operaciones que utiliza el esquema propuesto.

En el capítulo 6 se presentan las conclusiones obtenidas sobre el trabajo de tesis y las publicaciones generadas del mismo; además se mencionan los trabajos futuros que se plantean para continuar con esta investigación.

Por último se muestran las referencias bibliográficas consultadas para recopilar la información que da sustento a este trabajo de tesis.

# Capítulo 2

## Marco teórico y estado del arte

En este capítulo se presentan los fundamentos teóricos relacionados con la compresión de imágenes, es importante conocerlos para comprender la relevancia que tiene la cuantificación vectorial en el proceso de compresión. Además, se reseñan algunos trabajos relacionados con la cuantificación vectorial.

### 2.1. Compresión de imágenes

La compresión de imágenes se refiere a representar la información contenida en las imágenes de una forma eficiente en términos del menor número de bits posibles, manteniendo algunos requerimientos en cuanto a la representación de la imagen reconstruida.

En la compresión de imágenes digitales, existen tres tipos de redundancias las cuales deben ser identificadas y explotadas: redundancia inter-píxel, redundancia psico-visual y redundancia de codificación. Se logra compresión de datos cuando una o más redundancias son reducidas o eliminadas [16].

#### 2.1.1. Redundancia inter-píxel

Los píxeles dentro de la imagen tienen alta correlación. Por lo tanto, el valor de cualquier píxel dentro de una imagen puede ser razonablemente predicho a partir de los valores de los píxeles vecinos, se puede decir que la información transmitida por píxeles individuales es relativamente pequeña. A este tipo de redundancia también se le conoce como: redundancia espacial, redundancia geométrica o redundancia dentro de un cuadro (*interframe*) [16].

### 2.1.2. Redundancia psico-visual

La redundancia psico-visual está asociada con información visual real o cuantificable. Es importante mencionar que el sistema visual humano no responde con la misma sensibilidad a toda la información visual, es decir, cierta información tiene menor importancia relativa que otra. La eliminación de la información relacionada a la redundancia psico-visual es posible debido a que la información en sí misma no es esencial para el procesamiento visual normal. Desde que la eliminación de la redundancia psico-visual trae como resultado pérdida de información cuantitativa, es común que se le conozca también como cuantificación. El uso de esta palabra es congruente con su significado, ya que es un mapeo de un amplio intervalo de valores de entrada a un número limitado de valores de salida. Como es una operación irreversible (hay pérdida de información visual), la cuantificación resulta en una compresión de datos con pérdidas [16].

### 2.1.3. Redundancia de codificación

Si los niveles de gris de una imagen son codificados en una forma en la cual usan más símbolos de código que los absolutamente necesarios para representar cada nivel de gris, se dice que la imagen resultante contiene redundancia de codificación. En general, la redundancia de codificación está presente cuando los códigos asignados a un conjunto de eventos (ej. los valores en los niveles de gris) no han sido seleccionados para aprovechar al máximo las probabilidades de los eventos [16].

## 2.2. Esquema de un compresor de imágenes

Un sistema de compresión de imágenes está constituido por los siguientes bloques, un bloque de transformación seguido de un bloque de cuantificación y al final el codificador de entropía. En la Figura 2.1, se observa el diagrama a bloques de un compresor de imágenes.

El propósito de la transformación es remover o suprimir los datos redundantes (redundancia inter-píxel). Las transformaciones se basan en la descomposición de la frecuencia, se pueden usar las siguientes transformaciones, la transformación discreta de Fourier (DFT, *Discrete Fourier Transform*), la transformación discreta de coseno (DCT, *Discrete Cosine Transform*), la transformación wavelet de sub-bandas (SWT, *Subband Wavelet Transform*)

o la transformación Karhunen-Loeve (KLT, *Karhunen-Loeve Transform*) que es considerada la óptima, sólo por mencionar algunas. Las transformaciones pueden ser aplicadas a la imagen entera o a pequeños bloques de la imagen, usualmente los bloques no están traslapados, aunque en algunos métodos si se traslapan, ejemplo de ello es la transformación ortogonal traslapada (LOT, *Lapped Orthogonal Transform*) y SWT. Cabe mencionar que la transformación es un proceso reversible, es decir, se pueden recuperar los datos tal y como eran antes de la transformación.

En el bloque de cuantificación es donde se da la compresión por ello se le considera importante, un cuantificador es una transformación no lineal que selecciona valores representativos de los rangos de datos de entrada procedentes de la transformación, un dato a la vez si es cuantificación escalar o muchos datos (vector) a la vez si es cuantificación vectorial, en este bloque se explota la redundancia psico-visual. El procesamiento realizado por un cuantificador es irreversible, por lo cual presenta pérdida de información.

El codificador de entropía convierte los valores representativos (salida del cuantificador), a vectores de códigos eficientes de longitud variable, no es conveniente omitir este bloque, debido a que la salida del cuantificador son vectores de códigos de longitud fija, que usualmente son menos eficientes. Algunos ejemplos de códigos de longitud variable son los códigos Huffman y aritméticos. En este bloque se reduce la redundancia por codificación, cabe mencionar que la codificación es un proceso reversible [16], [47].

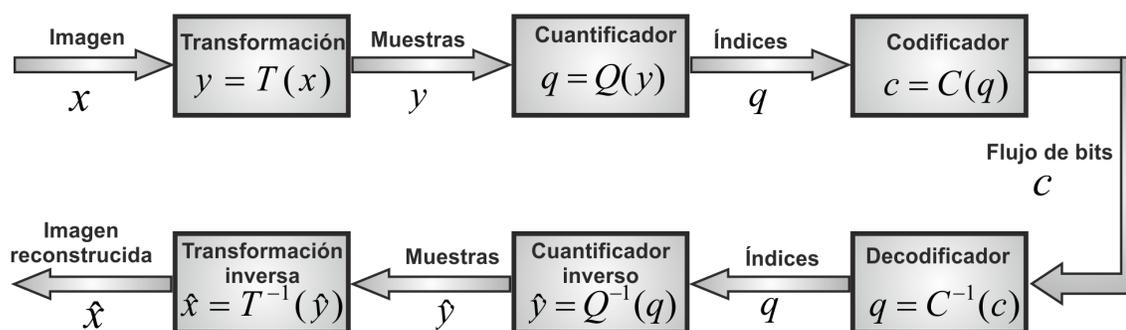


Figura 2.1: Diagrama a bloques de un esquema de compresión de imágenes (Trad. de [44]).

### 2.2.1. Compresión sin pérdidas y con pérdidas

Los esquemas de compresión en general tienen dos vertientes, la compresión sin pérdidas y la compresión con pérdidas.

Los esquemas de compresión sin pérdidas están limitados en la cantidad de compresión que pueden lograr explotando redundancias inter-píxel y de codificación, aunque en algunas circunstancias es aceptable. Es conveniente usar este esquema cuando los recursos disponibles para almacenamiento o transmisión son suficientes para cumplir con los requerimientos de este tipo de compresión o si las posibles consecuencias de perder información pueden ser mucho más caras que el costo de un recurso adicional para almacenamiento o transmisión. Por el contrario, si los recursos son limitados y no se requiere integridad absoluta en los datos recuperados, se puede mejorar la cantidad de compresión con un cierto grado de pérdida durante el proceso de compresión. Lo deseable es tener una cantidad mínima de distorsión mientras se comprime a la menor tasa posible [41].

## 2.3. Cuantificación

El proceso de cuantificación se encuentra dentro de los esquemas de compresión con pérdidas, debido a que es un proceso que representa un conjunto grande de valores (posiblemente infinito) con un conjunto más pequeño.

**Definición 2** La *cuantificación* describe una transformación no lineal de una señal  $x$  a una nueva señal  $y$  [45]:

$$y = Q(x) = r_k, \text{ si } d_k \leq x < d_{k+1} \quad (2.1)$$

donde  $r_k$  se refiere a un nivel de reconstrucción y  $d_k$  se refiere a un nivel de decisión.

La salida está limitada a un número finito de niveles de reconstrucción. Los valores de entrada pueden variar en forma continua o pueden ser valores que previamente han sido cuantificados por otro proceso. Es común digitalizar los datos de una imagen a una precisión de 12 a 16 bits (4096 a 65536 niveles en la escala de grises) y después se hace una nueva cuantificación de los datos a 8 bits para su visualización o almacenamiento [45]. En general se puede clasificar a la cuantificación en dos categorías, dependiendo del número de elementos que se desean cuantificar, si se desea cuantificar un dato a la vez (cuantificación escalar) o grupos de datos (cuantificación vectorial).

## 2.4. Cuantificación escalar

En cuantificación escalar (SQ, *Scalar Quantization*), la salida cuantificada es el resultado de dividir los datos de entrada por un parámetro de cuantificación, haciendo un redondeo al entero más cercano. Si  $X$  es una muestra de entrada y  $q$  es un parámetro de cuantificación, la salida cuantificada es [40]:

$$X_q = \text{round} \left( \frac{X}{q} \right) \quad (2.2)$$

El proceso inverso se puede lograr aplicando expresión (2.3), se debe tomar en cuenta de que como en el proceso de cuantificación se uso redondeo, al recuperar los datos no se van a recuperar los datos originales.

$$X = q(X_q) \quad (2.3)$$

A su vez, la cuantificación escalar se puede dividir en cuantificación escalar uniforme y no uniforme.

### 2.4.1. Cuantificación escalar uniforme

Considere  $X_{max}$  es el valor máximo de una fuente de entrada y los valores de entrada están distribuidos uniformemente en el rango  $[-X_{max}, +X_{max}]$ . Se puede diseñar un cuantificador escalar uniforme de  $N$  niveles dividiendo el intervalo  $[-X_{max}, +X_{max}]$  en  $N$  sub intervalos iguales. La longitud  $\Delta$  de cada sub intervalo es denominado tamaño del paso el cual está dado por

$$\Delta = \frac{2X_{max}}{N} \quad (2.4)$$

En la Figura 2.2, se puede observar un cuantificador escalar uniforme de 8 niveles. El eje horizontal representa la entrada y el eje vertical representa el valor correspondiente después de la cuantificación. La implementación de este tipo de cuantificador es muy sencilla y simple. El diseño se baso en el supuesto de que la fuente de entrada está distribuida uniformemente. Pero a menudo la distribución de probabilidad de la fuente no es uniforme y se obtiene como resultado una deficiente calidad en la reconstrucción. Una alternativa

para solucionar este problema es diseñar cuantificadores no uniformes para ese tipo de fuentes [40].

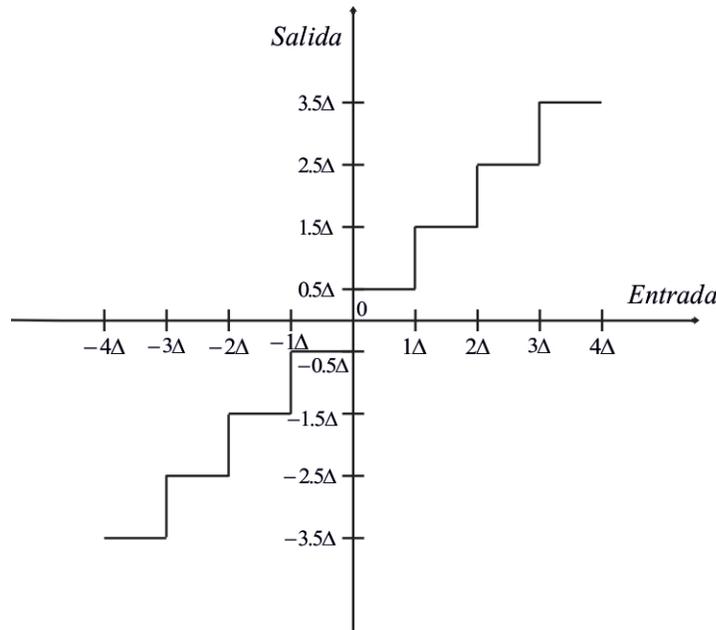


Figura 2.2: Cuantificador escalar uniforme (Trad. de [40]).

### 2.4.2. Cuantificación escalar no uniforme

Este tipo de cuantificador debe dividir el rango  $[-X_{max}, +X_{max}]$  de forma no uniforme para que la relación entre la entrada y la salida del cuantificador pueda igualar cualquier función lineal deseada. La Figura 2.3, muestra un cuantificador no uniforme, donde los pasos de cuantificación  $(\dots, X_1, X_2, \dots, X_{N-1}, X_N)$  y los niveles de salida  $(Y_1, X_2, \dots, Y_N)$  de un cuantificador de  $N$  niveles son fijos y pueden ser seleccionados para minimizar alguna función del error de cuantificación cuando la distribución de probabilidad de la entrada es conocida [40].

## 2.5. Cuantificación vectorial

Shannon fue el primero en sugerir que codificar una secuencia de muestras de una fuente podía proporcionar mejores resultados que codificar muestras individuales en términos de eficiencia de compresión [42]. La cuantificación vectorial (VQ, *Vector Quantization*) es una técnica de codificación en bloques que opera sobre vectores de muestras. Un cuantificador

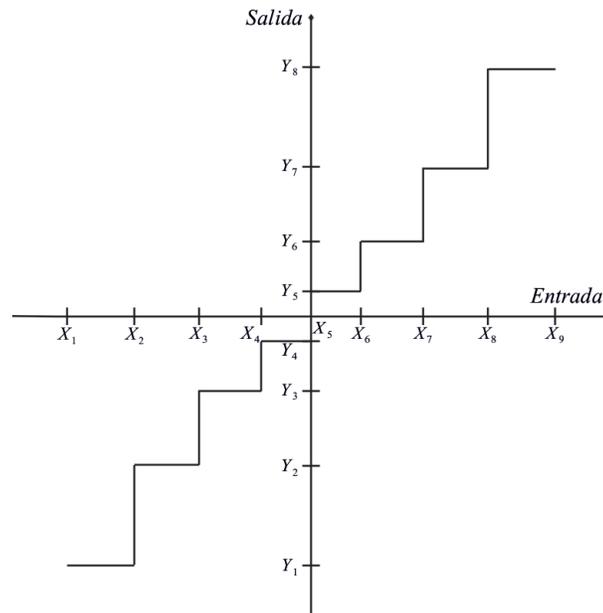


Figura 2.3: Cuantificador escalar no uniforme (Trad. de [40]).

vectorial puede procesar bloques de la imagen directamente en el dominio espacial y explota la correlación entre píxeles en el bloque. La popularidad de la cuantificación vectorial se debe al hecho de que los resultados obtenidos tienen menor distorsión que los resultados obtenidos por la cuantificación escalar. Por otro lado, la complejidad y requerimientos de memoria de un cuantificador vectorial incrementan exponencialmente con la dimensión del vector [17], [40].

El proceso de cuantificación vectorial consiste en dos pasos:

- Dividir el espacio de entrada en regiones de decisión.
- Asignar niveles de reconstrucción para cada una de las regiones.

Los niveles de reconstrucción se denominan libros de códigos. Dado que se asume que el espacio de entrada está cuantificado, solo hay un número finito de formas de dividir el espacio de entrada. El espacio de salida también está cuantificado, por lo que hay solamente un número finito de niveles de reconstrucción. Por supuesto que para cualquier número razonable de niveles de cuantificación, es evidente que es imposible probar todas las posibles combinaciones de las regiones de decisión y niveles de reconstrucción. Por lo tanto, se busca una forma de restringir el problema en términos de lo que puede ser matemáticamente manejable [45].

La cuantificación vectorial puede ser aplicada a cualquier tipo de datos digitales. Para una señal de una dimensión, se pueden formar vectores agrupando bloques de muestras contiguos de la secuencia. Para una señal de dos dimensiones (por ejemplo, imágenes digitales) los vectores pueden formarse agrupando bloques rectangulares, y con tales bloques se pueden formar vectores de una dimensión para que puedan ser procesados. Esta idea puede ser aplicada a datos que representen tres dimensiones (por ejemplo, video) [25].

**Definición 3** Un **cuantificador vectorial**  $Q$  de dimensión  $n$  y tamaño  $N$  es formalmente un mapeo del espacio Euclideo  $n$ -dimensional  $R^n$  en un conjunto finito  $C$ , que contiene  $N$  vectores de reproducción de  $R^n$ , por lo tanto [15]:

$$Q = R^n \rightarrow C \quad (2.5)$$

donde  $C = \{y_0, y_1, \dots, y_{N-1}\}$  e  $y_i \in R^n$  para cada  $i \in J = \{0, 1, \dots, N - 1\}$ .

El conjunto  $C$  es denominado libro de códigos (*codebook*) y contiene  $N$  número de vectores de códigos (*codevectors*) o vectores de reconstrucción (*codewords*)  $y_i$ , cada uno de estos vectores es distinto y pertenecen a  $R^n$ .

Un cuantificador vectorial puede ser dividido en dos operaciones, en un codificador y un decodificador.

**Definición 4** Un **codificador**  $E$  consiste en un mapeo de  $R^n$  a un conjunto de índices  $J$  [15]:

$$E = R^n \rightarrow J \quad (2.6)$$

El proceso de codificación tiene una interpretación útil e interesante del espacio de dimensión  $n$ . El conjunto de vectores de reproducción define una partición de  $R^n$  en  $N$  número de celdas  $R_i$ , donde  $i = 1, 2, \dots, N$ . La  $i$ -ésima celda esta definida por [25]:

$$R_i = \{\mathbf{x} \in R^n : Q(\mathbf{x}) = y_i\} \quad (2.7)$$

Se tiene que  $\bigcup_i R_i = R^n$  y  $R_i \cap R_j = \emptyset$  para todo  $i \neq j$ .

**Definición 5** Un **decodificador**  $D$  mapea un conjunto de índices  $J$  en el conjunto de reproducción  $C$  [15]:

$$D = J \rightarrow C \quad (2.8)$$

El funcionamiento general de un cuantificador vectorial puede ser considerado como la composición de las dos operaciones anteriormente definidas [15].

$$Q(\mathbf{x}) = D(E(\mathbf{x})) \quad (2.9)$$

En la Figura 2.4, se observa el diagrama a bloques de un codificador y un decodificador los cuales conforman al cuantificador vectorial. La salida del codificador es un índice que representa a la palabra de códigos que mejor aproxima al vector de entrada. El proceso de reconstrucción, el cual se denomina decodificación, involucra una búsqueda (por medio del índice) de la correspondiente palabra de códigos en el libro de códigos que es el mismo que se usa en el codificador. La resolución o tasa del cuantificador es  $r = (\log_2 N)/n$ , la cual mide el número promedio de bits por componente del vector, usados para representar al vector de entrada (y por lo tanto el grado de precisión), cabe mencionar que en cuantificación vectorial se pueden tener tasas de bit de tipo fraccionario tales como  $1/2, 3/4, 16/3$ , etcétera. Esto difiere de la cuantificación escalar donde la tasa es fija. Es importante reconocer que para una dimensión fija  $n$ , la resolución está determinada por el tamaño  $N$  del libro de códigos y no por el número de bits usados para describir los vectores de código almacenados en el libro de códigos [15], [25].

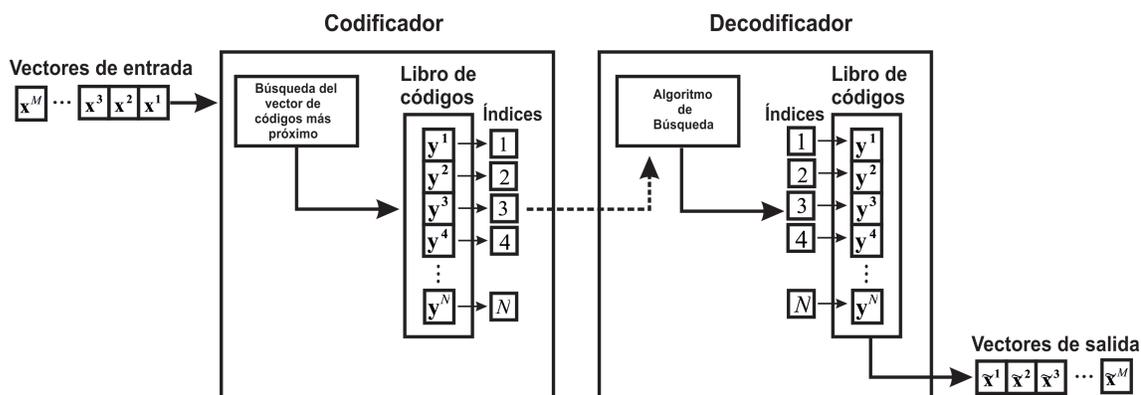


Figura 2.4: Estructura básica de un cuantificador vectorial.

### 2.5.1. Desempeño de un compresor de imágenes

El desempeño de un compresor de imágenes se mide por la calidad de la reproducción resultante ( $\tilde{p}_i$ ) en comparación con la original ( $p_i$ ). Una forma de lograr esto es mediante la definición de una medida de distorsión  $d(x, \hat{x})$  que cuantifica el costo o la distorsión

resultante. A una menor distorsión promedio, se tiene un mejor desempeño en términos de calidad visual. Los esquemas de cuantificación usan el error absoluto medio (MAE, *Mean Absolute Error*), el error cuadrático medio (MSE, *Mean Square Error*) y la relación señal a ruido (PSNR, *Peak Signal to Noise Ratio*) como medidas de distorsión [1], [25], representadas por las expresiones (2.10), (2.11) y (2.12) respectivamente.

$$MAE = \frac{1}{M} \sum_{i=1}^M |p_i - \tilde{p}_i| \quad (2.10)$$

$$MSE = \frac{1}{M} \sum_{i=1}^M (p_i - \tilde{p}_i)^2 \quad (2.11)$$

$$PSNR = 10 \log_{10} \left( \frac{(2^L - 1)^2}{\frac{1}{M} \sum_{i=1}^M (p_i - \tilde{p}_i)^2} \right) \quad (2.12)$$

donde  $L$  es el número de bits por píxel,  $M$  es el número de píxeles en la imagen,  $p_i$  es el  $i$ -ésimo píxel en la imagen original, y  $\tilde{p}_i$  es el  $i$ -ésimo píxel en la imagen reconstruida.

## 2.6. Estado del arte

La cuantificación vectorial ha sido utilizada en diferentes aplicaciones que involucran compresión de datos. En este trabajo se presenta un algoritmo alternativo para la cuantificación vectorial de imágenes basado en el algoritmo LBG y el algoritmo LAMDA, a continuación se hace una breve reseña de algunos trabajos que han sido parte importante en la evolución de la cuantificación vectorial.

En 1898 Sheppard hizo un análisis sobre el redondeo aplicado a la estimación de densidades por histogramas, el redondeo es el ejemplo más antiguo de cuantificación escalar. Cualquier número real  $x$  puede ser redondeado al número entero más cercano. Se tiene que  $q(x)$  representa el resultado de la cuantificación;  $q(x) = x + e$ , donde  $e$  es el error de cuantificación. De una forma más general, un cuantificador se define como un conjunto de intervalos o celdas  $S = \{S_i; i \in I\}$ , donde el conjunto de índices  $I$  es normalmente una colección de números enteros consecutivos, se tiene un conjunto de valores de reproducción o niveles  $C = \{y_i; i \in I\}$ , de tal modo un cuantificador  $q$  queda definido por  $q(x) = y_i$ , para  $x \in S_i$ , el cual puede ser expresado como:  $q(x) = \sum_i y_i 1_{S_i}(x)$ , donde la función  $1_S(x)$

es 1 si  $x \in S$  y 0 en cualquier otro caso. La función  $q(x)$  también es denominada regla de cuantificación.

En 1948, Oliver Pierce reconoció el papel fundamental que tenía la cuantificación en la modulación y en la conversión de analógico a digital en los sistemas de modulación por pulsos codificados (PCM, *pulse code modulation*); en ese mismo año Bennett publicó el primer análisis de alta resolución de la cuantificación y un análisis exacto del ruido de cuantificación para procesos Gaussianos, por su parte Shannon publicó los principios de la teoría sobre la tasa de distorsión, la cual ofrece una teoría para la cuantificación vista como una conversión analógica a digital y como una compresión de datos. Tiempo después la modulación PCM fue analizada detalladamente por Oliver, Pierce y Shannon, como dato adicional cabe mencionar que la modulación PCM fue la primera técnica digital para transmitir una señal de información analógica (voz vía telefónica) por un canal analógico (cable o la atmósfera) [18].

En el año 1957, Lloyd realizó un estudio sobre cuantificación e hizo tres contribuciones. Primero, encontró las condiciones necesarias y suficientes para que un cuantificador de tasa fija sea óptimo a nivel local. Si se satisfacen las condiciones implica que pequeñas perturbaciones en los niveles o umbrales aumentaría la distorsión. Las condiciones de optimización de Lloyd establecen que la partición del cuantificador debe ser óptima para el conjunto de niveles de reproducción y el conjunto de niveles de reproducción debe ser óptimo para la partición. Para el caso del error cuadrático medio, la primera condición implica una distancia mínima o la regla de cuantificación del vecino más cercano, para seleccionar el nivel de reproducción más cercano a la fuente que está siendo cuantificada, y la segunda condición implica que el nivel de reproducción correspondiente a una celda dada es la esperanza condicional o centroide del valor de la fuente que se encuentra en una celda especificada. Segundo, basándose en las condiciones de optimización, Lloyd desarrolló un algoritmo iterativo descendente para el diseño de cuantificadores dada una fuente de distribución. Empezando con una colección inicial de niveles de reproducción; optimiza la partición para esos niveles usando un mapeo de la distorsión mínima, la cual da una partición de la línea real en intervalos; es entonces cuando se optimiza el conjunto de niveles de la partición mediante la sustitución de los niveles anteriores por los centroides de las celdas de la partición, el algoritmo continúa hasta converger a un óptimo local. A este algoritmo se le denominó Método I, y es el que se utiliza para los cuantificadores vectoriales. Tercero,

Lloyd mostró que en algunas situaciones donde el óptimo global es solamente el óptimo local, los cuantificadores satisfacen las condiciones de optimización [18].

Los algoritmos de agrupamiento difuso (*fuzzy clustering*) consideran cada grupo como un conjunto difuso, mientras una función de membresía mide la posibilidad que cada vector de entrenamiento pertenezca a un grupo. En 1969, E.H. Ruspini desarrolló el primer algoritmo de agrupamiento difuso, basado en el concepto de conjuntos difusos introducida por Zadeh, después de este trabajo el uso de la lógica difusa, fue una nueva opción en la investigación. En 1973, J. C. Dunn considero una formulación alternativa del proceso de agrupamiento y propuso el algoritmo difuso k-means. En 1984, Bezdek *et al.* Extendieron la formulación que hizo Dunn y produjeron una familia de algoritmos difusos k-means, la cual incluye el algoritmo de Dunn como un caso especial [23].

Otra forma de abordar el problema fue usando una transformación, el propósito de usar una codificación por transformación es convertir los píxeles estadísticamente dependientes en coeficientes independientes, la transformación también tiene la propiedad de comprimir la mayor parte de la energía en un bloque con coeficientes de baja frecuencia. El libro de códigos diseñado en el dominio de la transformación se cree que está más cerca del óptimo que uno diseñado en el dominio espacial ya que los coeficientes transformados tienen una distribución mejor definida que los píxeles de la imagen, además algunos coeficientes de alta frecuencia son desechados y por lo tanto el costo computacional se reduce. Algunos trabajos sobresalientes relacionados con este tópico son los siguientes: En 1969, Andrews *et al.* usaron la transformación Hadamard, en 1972, Anderson y Huang aplicaron la transformación de Fourier, en 1974, Pratt *et al.* emplearon la transformación Slant, en 1986, Saito *et al.* utilizaron la transformación Discreta de Coseno, solo por mencionar algunos. Una de las principales desventajas de la cuantificación vectorial usando transformación es que la transformación debe ser aplicada tanto en el codificador como en el decodificador, esto trae como consecuencia que se incrementa la complejidad computacional, comparado con los sistemas tradicionales que en el decodificador sólo usan una tabla de búsqueda [34].

En el año 1980, Linde *et al.* propusieron el algoritmo LBG (Linde, Buzo y Gray) es una generalización del algoritmo diseñado por Lloyd para cuantificación escalar, por lo cual es también conocido como algoritmo generalizado de Lloyd (GLA, *Generalized Lloyd Algorithm*), en el contexto del reconocimiento de patrones se le conoce como algoritmo k-means [25], [31].

Un problema que se tiene en cuantificación vectorial es que se tiene que hacer la búsqueda en todo el libro de códigos, tratando de solucionar este problema Buzo *et al.* propusieron el cuantificador vectorial estructurado en árbol (TSVQ, *Tree Structured Vector Quantizer*). TSVQ consiste de un arreglo jerárquico de vectores de códigos, lo cual permite hacer búsquedas eficientes al libro de códigos. Tiene la propiedad de que el tiempo de búsqueda crece linealmente con la tasa en lugar de que crezca en forma exponencial. Los arboles binarios son a menudo usados para TSVQ porque son los más eficientes en términos de complejidad. TSVQ es un ejemplo de un cuantificador con restricciones que permite un aumento en la velocidad de búsqueda, pero con un incremento en memoria y una pequeña pérdida en el desempeño [25].

En 1989 Equitz W. propuso el algoritmo vecinos más cercanos por parejas (PNN, *Pairwise Nearest Neighbor*), se tiene un conjunto de entrenamiento y sistemáticamente se unen vectores hasta llegar al tamaño N del libro de códigos. La idea es identificar pares de vectores los cuales son más cercanos en términos de una medida de distorsión, y reemplazar esos dos vectores con un vector que contenga su promedio, con lo cual se reduce el tamaño del libro de códigos en cada etapa. Al unir las particiones correspondientes, puede dar como resultado un pequeño incremento en la distorsión, una desventaja que tiene este algoritmo es que la tarea de encontrar particiones para después unir las, tiene una demanda a nivel computacional. El libro de códigos diseñado por el algoritmo PNN puede ser utilizado directamente por un cuantificador vectorial o como un libro de códigos inicial para el algoritmo LBG, lo cual ha demostrado que conduce a soluciones que contienen un mejor óptimo a nivel local [25].

Una técnica de agrupamiento llamada mapas auto-organizativos (SOM, *Self-Organizing Map*) fue propuesto por Teuvo Kohonen como un algoritmo de aprendizaje para redes neuronales artificiales de aprendizaje competitivo. Los SOM también han sido usados con gran éxito en la creación de nuevos esquemas de cuantificación vectorial [27], [28], [29].

Seyed Bahram *et al.* Propusieron un método para diseñar cuantificadores vectoriales basados en aprendizaje competitivo para redes neuronales. Esta técnica fue usada para crear una similitud topológica entre el espacio de entrada y el espacio de índices, esta similitud ha sido usada para mejorar el desempeño de cuantificadores vectoriales en canales ruidosos [6].

En [7] se propuso un algoritmo que integra las ventajas presentadas por el algoritmo para cuantificación vectorial LBG y las redes neuronales artificiales de aprendizaje competitivo, con el fin de mejorar su desempeño en comparación con los algoritmos originales. Los resultados reportados muestran que la red neuronal propuesta proporciona mejor desempeño en términos de la relación de compresión y la calidad de la imagen reconstruida, también es competitiva en su velocidad de ejecución.

C. Amerijckx *et al.* propusieron un esquema de compresión con pérdidas para imágenes digitales utilizando el algoritmo de Kohonen. Ellos aplicaron los SOM tanto en las etapas de cuantificación y codificación del compresor de imágenes. En la etapa de cuantificación, el algoritmo SOM crea una correspondencia entre el espacio de entrada de estímulos y el espacio de salida constituido por los elementos del libro de códigos usando la distancia Euclidiana. Después del proceso de aprendizaje de la red, los elementos del libro de códigos aproximan a los vectores en el espacio de entrada en la mejor forma posible. En la etapa del codificador de entropía, un codificador entrópico diferencial utiliza la propiedad de preservación de la topología de los SOM que se obtuvo en el proceso de aprendizaje y la hipótesis de que los bloques consecutivos en la imagen son similares [4].

E. Guzmán *et al.* propusieron un nuevo algoritmo de búsqueda rápida para cuantificación vectorial basado en memorias asociativas extendidas (FSA-EAM, *Fast Search Algorithm-Extend Associative Memories*). Este algoritmo hace uso de las EAM para crear un libro de códigos-EAM aplicando las EAM en la etapa de entrenamiento al libro de códigos producido por el algoritmo LBG. El resultado de esta etapa es una red asociativa cuya meta es establecer una relación entre el conjunto de entrenamiento y el libro de códigos generado por el algoritmo LBG. Esta red asociativa es el libro de códigos-EAM el cual es usado por el FSA-EAM. El proceso de cuantificación vectorial del FSA-EAM es realizado usando la etapa de recordar (*recalling stage*) del EAM. Este proceso genera un conjunto de índices de clases a los cuales cada vector de entrada pertenece. La principal ventaja ofrecida por este algoritmo es la alta velocidad de procesamiento y baja demanda de recursos (memoria del sistema), mientras que la calidad de codificación es competitiva [19].

Hornig M-H., and Jiang T-W. proponen un nuevo método para generar libros de códigos basándose en el enfoque meta-heurístico, para ello utilizaron el algoritmo de la luciérnaga (FF, *firefly*), este algoritmo usa al algoritmo LBG por lo que los autores lo denominaron FF-LBG. Los métodos meta-heurísticos contra los que los autores comparan su propuesta,

se basan en algoritmos como abejas de miel (HB, *Honey Bee*) y optimización del enjambre de partículas (PSO, *Particle Swarm Optimization*), estos métodos se distinguen por que la solución que proporcionan es muy cercana a la solución global óptima. Los resultados prácticos proporcionaron un libro de códigos con menor distorsión lo que se ve reflejado en la calidad de la imagen reconstruida y además consume menos tiempo de ejecución, con respecto a los algoritmos LBG, PSO-LBG y HBMO-LBG [21].

En [43] Somasundaram K. y Shanthi Rani M. M. proponen un algoritmo que se basa en el algoritmo K-means y el algoritmo de cuantificación vectorial residual (RVQ, *Residual Vector Quantization*). El algoritmo involucra los siguientes procesos: particionamiento y agrupamiento, poda y construcción de un libro de códigos maestro y por último cuantificación vectorial residual. Los resultados experimentales demostraron que la solución que se obtiene es rápida y logra una mejor tasa de compresión que es comparable con los resultados de la implementación convencional del algoritmo k-means.



# Capítulo 3

## Algoritmos LBG y LAMDA

Los algoritmos LBG y LAMDA son expuestos en este capítulo, es importante su descripción ya que son la base esencial del esquema propuesto.

### 3.1. Algoritmo LBG

El algoritmo de agrupamiento (*clustering*) propuesto por Y. Linde, A. Buzo y R. Gray (LBG) es una generalización del algoritmo de Lloyd, y es usado para generar libros de códigos y como clasificador dentro de un esquema de cuantificación vectorial. Para el propósito de este trabajo, el interés se centra en el proceso de generación de libros de códigos del algoritmo LBG.

Este algoritmo se basa en la hipótesis de la optimización de los vectores de códigos en cada etapa, se asume como condición que los resultados de la etapa previa son óptimos, lo que prácticamente da como resultado el mejor libro de códigos, al menos a nivel local. El algoritmo de generación de libros de códigos consiste de la siguiente secuencia de pasos [12], [31]:

**Paso 1. Inicialización.** Se empieza con un libro de códigos inicial, formado por un conjunto finito de vectores de  $n$ -dimensión, este conjunto se denota como  $C = \{\mathbf{y}^i : i = 1, 2, \dots, N\}$ , y un conjunto de vectores de entrenamiento  $X = \{\mathbf{x}^j : j = 1, 2, \dots, M\}$ , un umbral de distorsión  $\epsilon$ , y se inicializa el índice  $k = 0$ .

**Paso 2. Partición.** Dado un conjunto definido por  $C^{(k)} = \{\mathbf{y}^i : i = 1, 2, \dots, N\}$ , encontrar todos los vectores de entrenamiento  $\mathbf{x}^j$  que están más cerca a  $\mathbf{y}^i$  que a cualquier otro  $\mathbf{y}^m$ , en otras palabras, se debe encontrar la partición que tenga mínima distorsión  $P(C^{(k)}) =$

$\{S^i : i = 1, 2, \dots, N\}$  del conjunto de vectores de entrenamiento:  $\mathbf{x}^j \in S^i$  si  $d(\mathbf{x}^j, \mathbf{y}^i) \leq d(\mathbf{x}^j, \mathbf{y}^m)$ , para todos los  $i \neq m$ . Calcular la distorsión promedio (distancia Euclidiana).

$$d^{(k)} = \sqrt{\sum_{l=1}^n (x_l - y_l)^2} \quad (3.1)$$

donde  $x_l$  es el  $l$ -ésimo componente del vector de entrada  $\mathbf{x}^j$  y  $y_l$  es el  $l$ -ésimo componente del vector de reconstrucción (*codeword*)  $\mathbf{y}^i$ .

**Paso 3.** *Actualización del libro de códigos.* En la iteración  $k = k + 1$ , en cada partición  $S^{i(k)}$ , se genera una nueva palabra de código mediante la aplicación de la condición del centroide:

$$\mathbf{y}^{i(k)} = \frac{1}{L} \sum_{i=1}^L x_l^i \quad (3.2)$$

donde  $L$  es el número de vectores de entrenamiento en  $S^{i(k)}$  and  $x_l^i$  es el  $l$ -ésimo componente de cada vector de entrenamiento.

**Paso 4.** *Comprobar la convergencia.* Detener cuando el proceso converge, si  $\frac{d^{k-1} - d^k}{d^k} \leq \epsilon$ . En otro caso ir al paso 2.

Dado que el algoritmo reduce el error de distorsión en cada paso y el error esta acotado por un número cercano a cero, esto garantiza la convergencia. Debido a que todos los valores están cuantificados al comienzo, el algoritmo convergerá en un número finito de pasos. El parámetro,  $\epsilon$ , tendrá un efecto sobre la tasa de convergencia, pero la experiencia ha demostrado que el algoritmo no es muy sensible a los cambios en  $\epsilon$ , valores alrededor de 0.0001 son comunes para imágenes con valores de 8 bits [45].

La Figura 3.1, muestra el esquema de cuantificación vectorial basado en el algoritmo LBG (LBG-VQ). En este esquema, una imagen de  $h \times w$  píxeles se divide en  $M$  bloques de dimensión  $n$  ( $n = d \times d$ ); estos bloques representan a los vectores de entrada  $X = \{\mathbf{x}^j : j = 1, 2, \dots, M\}$ . El algoritmo LBG-VQ consiste de un cuantificador  $Q$  de nivel  $N$  y de dimensión  $n$ , que hace un mapeo de un vector de entrada en un espacio Euclidiano  $\mathfrak{R}^n$ , en un sub-conjunto finito  $C$  de  $\mathfrak{R}^n$ , que contiene  $N$  vectores de reproducción, también se les conoce como vectores de códigos (*code vectors*) o palabras de códigos,  $Q : \mathfrak{R}^n \rightarrow C$ , donde  $C = \{\mathbf{y}^i : i = 1, 2, \dots, N\}$ ,  $\mathbf{y}^i \in \mathfrak{R}^n$  [15].

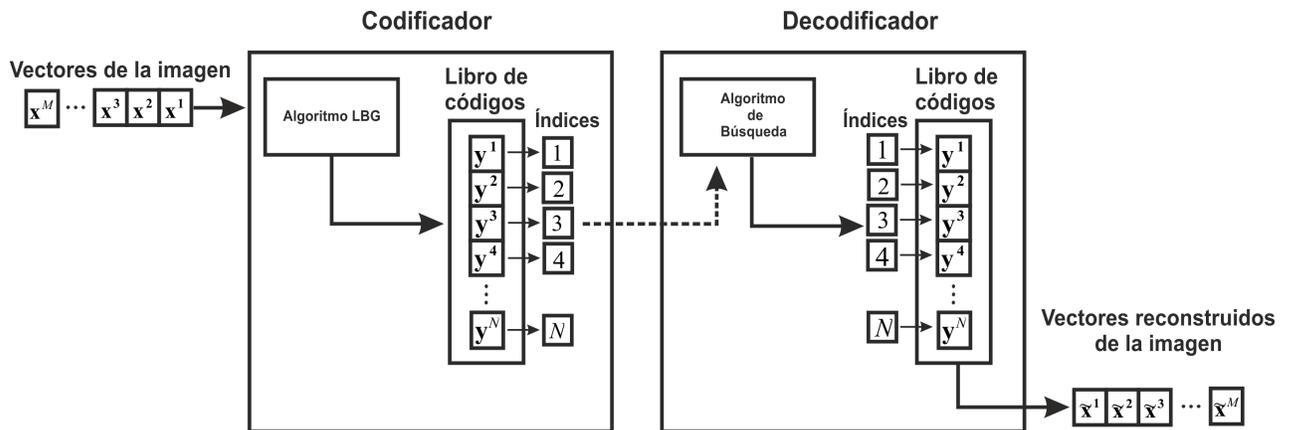


Figura 3.1: Esquema de cuantificación vectorial LBG.

### 3.2. Algoritmo de aprendizaje para el análisis de datos multi-variables

En áreas relacionadas con el análisis digital de imágenes, como el agrupamiento, análisis de datos y recientemente aprendizaje maquina (ML, *Machine Learning*), se investigan procedimientos que realicen un estudio integral donde se incluyan la mayor parte de las características o descriptores de la información a procesar, esto con la finalidad de conseguir algoritmos que generen la menor pérdida de información posible y eliminen las redundancias que impiden distinguir sus relaciones, esto se hace debido a que el conocimiento que la clasificación puede revelar depende del conjunto de características agrupadas [36].

El algoritmo de aprendizaje para el análisis de datos multi-variables (LAMDA, *Learning Algorithm for Multivariate Data Analysis*) es un método de agrupamiento conceptual incremental basado en lógica difusa, el cual puede ser aplicado en los procesos de formación y reconocimiento de conceptos (clases) [2], [36], [38].

LAMDA tiene las siguientes características:

- No es necesario el conocimiento previo de las clases (aprendizaje no supervisado).
- Los descriptores pueden ser cualitativos, cuantitativos o una combinación de ambos.

- LAMDA puede utilizar una etapa de aprendizaje supervisado seguida de una etapa no supervisada, por esta razón, es posible conseguir una clasificación evolutiva, esto le permite a la metodología LAMDA generar nuevas clases.
- La formación y reconocimiento de clases se basan en una regla de máxima adecuación (MA, *Maximum Adequacy*), es decir, se asigna un objeto a una clase, cuando el grado de adecuación es máximo.
- Con esta metodología se tiene la posibilidad de controlar la selectividad de la clasificación a través del parámetro  $\alpha$ .
- LAMDA modela el concepto de máxima entropía (homogeneidad). Este concepto está representado por una clase denominada clase No-Informativa (NIC, *Non-Informative Class*). El concepto NIC desempeña el papel de un umbral de decisión en el proceso de formación de clases.
- Los procesos de aprendizaje y reconocimiento son secuenciales.

Tradicionalmente el concepto de similitud entre los objetos ha sido considerado fundamental para determinar si los descriptores son miembros de una clase o no lo son. LAMDA no usa medidas de similitud entre los objetos con el fin de agruparlos, sino que calcula un grado de adecuación, este concepto está expresado como una función de pertenencia entre el descriptor y cualquiera de las clases establecidas previamente [3], [36].

### 3.2.1. Funcionamiento de la metodología LAMDA

Los objetos  $X$  (vectores de entrada) y las clases  $C$  están representadas por descriptores denotados por  $(d_1, \dots, d_n)$ , entonces cada  $d_i$  tiene su propio valor dentro del conjunto  $D_k$ , que está compuesto de todos los descriptores de la información a procesar, el producto  $n$ -ario de  $D_k$ , se escribe como  $D_1 \times, \dots, \times D_p$ , se le denomina Universo ( $U$ ) y esta representado como:  $\{(d_1, \dots, d_n) : d_i \in D_k \text{ for } 1 \leq i \leq n, 1 \leq k \leq p\}$ .

El conjunto de objetos pueden ser descritos por  $X = \{\mathbf{x}^j : j = 1, 2, \dots, M\}$  y cualquier objeto puede ser representado por un vector  $\mathbf{x}^j = (x_1, \dots, x_n)$ , donde  $x_i \in U$ , de modo que cada componente  $x_i$  corresponderá al valor dado por el descriptor  $d_i$  para el objeto  $\mathbf{x}^j$ . El conjunto de clases pueden ser descritas por  $C = \{\mathbf{c}^l : l = 1, 2, \dots, N\}$ , y cualquier

clase puede ser representada por un vector  $\mathbf{c}^l = (c_1, \dots, c_n)$ , donde  $c_i \in U$ , de modo que cada componente  $c_i$  corresponderá al valor dado por el descriptor  $d_i$  para la clase  $\mathbf{c}^l$  [38]. A continuación se describen los conceptos grado de adecuación margina y grado de adecuación global, que son la base del algoritmo LAMDA.

### 3.2.1.1. Grado de Adecuación Marginal

Dado un objeto  $\mathbf{x}^j$  y una clase  $\mathbf{c}^l$ , LAMDA calcula para cada descriptor el denominado grado de adecuación marginal (MAD, *Marginal Adequacy Degree*) por medio de una función de pertenencia,  $MAD(x_i^j/c_i^l) = \mathbf{x}^j \times \mathbf{c}^l \rightarrow [0, 1]^n$ , entre el valor del componente  $x_i$  de un objeto  $\mathbf{x}^j$  y el valor que el componente  $c_i$  toma en  $\mathbf{c}^l$ . Por lo tanto, un vector MAD puede ser asociado con un objeto  $\mathbf{x}^j$  como se observa en la Figura 3.2; además, para mantener consistencia con la lógica difusa los descriptores deben ser normalizados usando (3.3). Esta etapa genera  $N$  vectores de MADs, y este proceso se repite iterativamente para cada objeto con todas las clases [46].

$$x_i = \frac{\tilde{x}_i - x_{min}}{x_{max} - x_{min}} = \frac{\tilde{x}_i}{2^L - 1} \quad (3.3)$$

**3.2.1.1.1. Funciones de pertenencia** Las funciones de pertenencia (*membership*) son usadas para asociar un grado de pertenencia de cada uno de los elementos del dominio al correspondiente conjunto difuso. Este grado de pertenencia indica la certidumbre (o incertidumbre) de que el elemento pertenezca al conjunto. Las funciones de pertenencia para conjuntos difusos deberán satisfacer las siguientes restricciones [14].

- La función de pertenencia debe tener como límite inferior 0 y límite superior 1.
- El rango de una función de pertenencia por lo tanto debe ser  $[0, 1]$ .
- Para cada  $x \in U$ , la función de pertenencia  $\mu_{\mathbf{x}}(x) \rightarrow [0, 1]$ , debe ser única. Esto es, el mismo elemento no se puede mapear a diferentes grados de pertenencia para el mismo conjunto difuso.

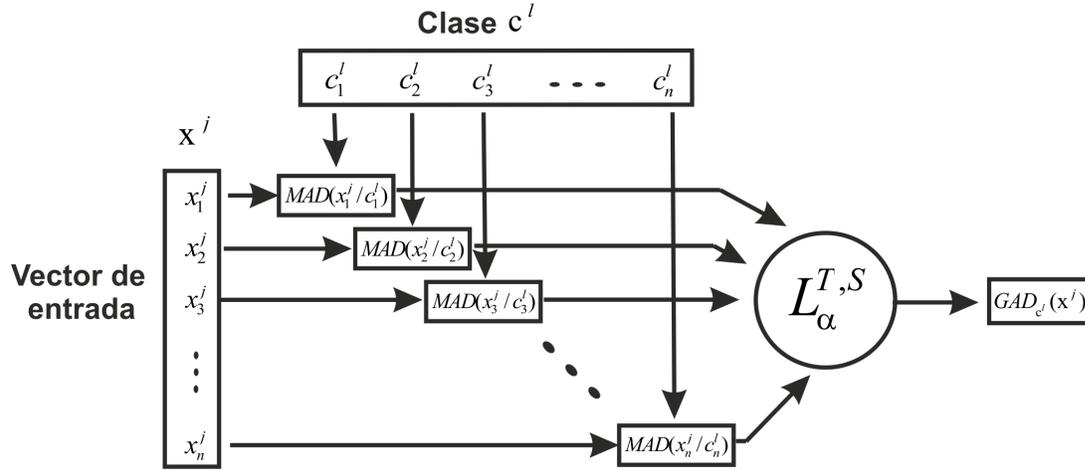


Figura 3.2: Estructura básica de la metodología LAMDA.

El MAD es una función de pertenencia derivada de una generalización difusa de una ley de probabilidad binomial. Tomando en cuenta que  $\mathbf{x}^j = (x_1, \dots, x_n)$  y que  $E$  es un subconjunto propio no vacío de  $X$ . Se tiene un experimento donde el resultado es considerado un “éxito” si el resultado  $x_i$  está en  $E$ , de lo contrario, el resultado es considerado un “fracaso”. Se dice que  $P(E) = \rho$  es la probabilidad de éxito de manera que  $P(E') = q = 1 - \rho$  es la probabilidad de fracaso, por tanto los valores intermedios tienen un grado de éxito o fracaso [10], [46]. La función de masa de probabilidad de  $X$  se define como:

$$f(x) = (\rho)^{(x)}(1 - \rho)^{(1-x)} \quad (3.4)$$

donde  $\rho \in [0, 1]$ .

Las siguientes distribuciones de probabilidad difusa son usadas comúnmente por la metodología LAMDA para calcular los MADs [3], [20].

- Distribución Binomial difusa.
- Distribución Binomial-Centro difusa.
- Distribución Binomial-Distancia difusa.
- Distribución Gaussiana.

### 3.2.1.2. Grado de Adecuación Global

El Grado de Adecuación Global (GAD, *Global Adequacy Degree*) se obtiene mediante la agregación de toda la información marginal calculada previamente, dada por los  $N$  vectores de *MADs* de un objeto  $\mathbf{x}^j$  relativo a la clase  $\mathbf{c}^l$ , a través de una función lineal convexa T-S  $L_\alpha^{T,S}$ , la cual usa operadores de agregación matemática (T-normas y S-conormas) [38], como se puede observar en la Figura 3.2. A continuación, se define el operador de agregación, las T-normas, S-conormas y la función lineal convexa T-S .

**3.2.1.2.1. Operadores de agregación** Los operadores de agregación son objetos matemáticos que tienen la función de reducir un conjunto de números en un único número representativo. Esto es simplemente una función la cual asigna un número real  $y$ , a cualquier vector de  $n$ -dimensión  $(x_1, x_2, \dots, x_n)$  de números reales,  $y = A(x_1, x_2, \dots, x_n)$  [13].

**Definición 6**  $A : \bigcup_{n \in \mathbb{N}} [0, 1]^n \rightarrow [0, 1]$  es un **operador de agregación** en el intervalo unitario si se cumplen las siguientes condiciones [13]:

$$\begin{aligned}
 A(x) &= x \text{ para todos los } x \in [0, 1] && \textit{Identidad cuando es unario.} \\
 A(0, \dots, 0) &= 0 \text{ y } A(1, \dots, 1) = 1 && \textit{Condiciones de frontera.} \\
 A(x_1, x_2, \dots, x_n) &\leq A(y_1, y_2, \dots, y_n) \text{ si } (\forall i = 1, 2, \dots, n) (x_i \leq y_i) && \textit{No decreciente.}
 \end{aligned}$$

Los operadores de agregación más populares son la media aritmética (3.5) (también conocida como promedio), mínimo (3.6), máximo (3.7), media geométrica (3.8), media armónica (3.9), media cuadrática (3.10), y la media- $\rho$  (3.11), para  $\rho \in ]0, \infty[$  [13], [26], a continuación se muestran las expresiones de estos operadores de agregación.

$$\mathbf{M}(x_1, \dots, x_n) = \frac{1}{n} \sum_{i=1}^n x_i \quad (3.5)$$

$$\mathbf{min} = \min(x_1, \dots, x_n) \quad (3.6)$$

$$\mathbf{max} = \max(x_1, \dots, x_n) \quad (3.7)$$

$$\mathbf{G}(x_1, \dots, x_n) = \left( \prod_{i=1}^n x_i \right)^{\frac{1}{n}} \quad (3.8)$$

$$\mathbf{H}(x_1, \dots, x_n) = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad (3.9)$$

$$\mathbf{Q}(x_1, \dots, x_n) = \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}} \quad (3.10)$$

$$M_\rho(x_1, \dots, x_n) = \left( \frac{1}{n} \sum_{i=1}^n x_i^\rho \right)^{\frac{1}{\rho}} \quad (3.11)$$

**3.2.1.2.2. T-normas y S-conormas** Las T-normas y las S-conormas son dos familias especializadas en la agregación en condiciones de incertidumbre. Pueden considerarse como una generalización de los conectivos lógicos Booleanos hacia la lógica multi-valuada. Las T-normas generalizan el operador de conjunción 'AND' (intersección) y las S-conormas generalizan el operador disyuntivo 'OR' (unión) [13].

**Definición 7 T-norma:** es una función de agregación multi-variable  $T : [0, 1]^n \rightarrow [0, 1]$ , tiene las siguientes propiedades [8]:

$$T(x_i) = T(x_j) \text{ donde } x \in [0, 1]^n, i = 0, \dots, n. \text{ y } j = n, \dots, 0. \quad (\mathbf{T1}) \text{ Conmutatividad.}$$

$$T_{n+1}(x_1, \dots, x_{n+1}) \leq T_n(x_1, \dots, x_n) \text{ para todos los } n > 1. \quad (\mathbf{T2}) \text{ Monotonicidad (incremento).}$$

$$T(x_1, \dots, x_n) = T(x_1, T(x_2, \dots, x_n)) = \dots = T(T(x_1, \dots, x_{n-1}), x_n) \quad (\mathbf{T3}) \text{ Asociatividad.}$$

$$T(x, 1) = T(1, x) = x \text{ para cada } x \in [0, 1] \quad (\mathbf{T4}) \text{ Uno como elemento neutral.}$$

Una propiedad bien conocida de las T-normas es la siguiente:

$$T(x) \leq \min(x) \text{ para cada } x \in [0, 1]^n$$

Esta propiedad es una consecuencia de los axiomas (T1, T2, T4).

**Definición 8 S-conorma:** es una función de agregación multi-variable  $S : [0, 1]^n \rightarrow [0, 1]$ , tiene las siguientes propiedades [8]:

$$S(x_i) = S(x_j) \text{ donde } x \in [0, 1]^n, i = 0, \dots, n. \text{ y } j = n, \dots, 0. \quad (\mathbf{S1}) \text{ Conmutatividad.}$$

$$S_{n+1}(x_1, \dots, x_{n+1}) \geq S_n(x_1, \dots, x_n) \text{ para todos los } n > 1. \quad (\mathbf{S2}) \text{ Monotonicidad (incremento).}$$

$$S(x_1, \dots, x_n) = S(x_1, S(x_2, \dots, x_n)) = \dots = S(S(x_1, \dots, x_{n-1}), x_n) \quad (\mathbf{S3}) \text{ Asociatividad.}$$

$$S(x, 0) = S(0, x) = x \text{ para cada } x \in [0, 1] \quad (\mathbf{S4}) \text{ Cero como elemento neutral.}$$

Una propiedad bien conocida de las S-conormas es la siguiente:

$$S(x) \geq \max(x) \text{ para cada } x \in [0, 1]^n$$

Esta propiedad es una consecuencia de los axiomas (S1, S2, S4).

Algunas T-normas y su dual S-conormas utilizadas en la metodología LAMDA son mostradas en la Tabla 3.1 [36], [38].

Tabla 3.1: T-Normas y S-Conormas utilizadas en la metodología LAMDA

Operador	T-Norma (Intersección)	S-Conorma (Unión)
Min-Max	$\min(x_1, \dots, x_n)$	$\max(x_1, \dots, x_n)$
Producto	$\prod_{i=1}^n x_i$	$1 - (\prod_{i=1}^n (1 - x_i))$
Lukasiewicz	$\max\{1 - n + \sum_{i=1}^n x_i, 0\}$	$\min\{\sum_{i=1}^n x_i, 1\}$
Yager	$1 - \min\left\{\left(\sum_{i=1}^n (1 - x_i)^{\frac{1}{\lambda}}\right)^{\lambda}, 1\right\}$	$\min\left\{\left(\sum_{i=1}^n (x_i)^{\frac{1}{\lambda}}\right)^{\lambda}, 1\right\}$
Hammacher	$\frac{1}{1 + \sum_{i=1}^n \left(\frac{1-x_i}{x_i}\right)}$ 0, si existe $x_i = 0$	$\frac{\sum_{i=1}^n \left(\frac{x_i}{1-x_i}\right)}{1 + \sum_{i=1}^n \left(\frac{x_i}{1-x_i}\right)}$ 1, si existe $x_i = 1$

**3.2.1.2.3. Función lineal convexa T-S** La función lineal convexa forma parte de las llamadas funciones compensatorias, y es utilizada para combinar una T-norma y una S-conorma con el fin de compensar sus efectos opuestos. Zimmermann H. y Zysno P., descubrieron que en el contexto de la toma de decisiones, los seres humanos no siguen exactamente el comportamiento de una T-norma (ni de una S-conorma) cuando agregan información. Con el fin de acercarse al proceso de agregación humana, propusieron un operador en el intervalo unitario basado en T-normas and S-conormas. La función lineal convexa T-S, es una clase de operador compensatorio no asociativo basado en T-normas y S-conormas [8], [13].

**Definición 9 Función lineal convexa T-S**  $L_{\gamma}^{T,S} : \bigcup_{n \in \mathbb{N}} [0, 1]^n \rightarrow [0, 1]$  está dada por [26]:

$$L_{\gamma}^{T,S}(x_1, \dots, x_n) = (1 - \gamma) \cdot T(x_1, \dots, x_n) + \gamma \cdot S(x_1, \dots, x_n) \quad (3.12)$$

donde  $\gamma \in [0, 1]$ ,  $T \leq L_{\gamma}^{T,S} \leq S$ ,  $T = L_0^{T,S}$  y  $S = L_1^{T,S}$ . El parámetro  $\gamma$  indica el grado de compensación o tolerancia.

En el caso de la metodología LAMDA se define  $\alpha = 1 - \gamma$ ; haciendo la sustitución la ecuación queda:

$$L_{\gamma}^{T,S}(x_1, \dots, x_n) = (1 - \gamma) \cdot T(x_1, \dots, x_n) + \gamma \cdot S(x_1, \dots, x_n)$$

$$L_{\alpha}^{T,S}(x_1, \dots, x_n) = (\alpha) \cdot T(x_1, \dots, x_n) + (1 - \alpha) \cdot S(x_1, \dots, x_n)$$

donde  $\alpha \in [0, 1]$ ,  $T \leq L_{\alpha}^{T,S} \leq S$ ,  $T = L_1^{T,S}$  (intersección) y  $S = L_0^{T,S}$  (unión). Al parámetro  $\alpha$  se le denomina nivel de exigencia [3], [36].

Finalmente, una vez calculado el valor de los GADs del objeto  $\mathbf{x}^j$  relacionado a todas las clases, y de acuerdo a la regla de máxima adecuación (MA, *Maximum Adequacy*), el objeto  $\mathbf{x}^j$  será asignado a la clase con mayor grado de adecuación [38].

La regla de máxima adecuación se define como:

$$MA = \max(GAD_{c^1}(\mathbf{x}^j), GAD_{c^2}(\mathbf{x}^j), \dots, GAD_{c^N}(\mathbf{x}^j)) \quad (3.13)$$

LAMDA ha sido aplicado a diferentes dominios: imágenes médicas [11], reconocimiento de patrones [37], detección y diagnóstico de fallas en procesos industriales [24], procesos biológicos [5], sistemas de distribución de energía eléctrica [33], procesos de producción de agua potable [20], monitoreo y diagnóstico de procesos industriales [22], selección de sensores [35].

# Capítulo 4

## Cuantificación Vectorial con base en el Algoritmo LAMDA

En este capítulo se describe un esquema de cuantificación vectorial con base en el algoritmo LBG y la metodología LAMDA, denominado VQ-LAMDA, el cual representa la propuesta del presente trabajo de tesis.

### 4.1. Definiciones preliminares

Sea una imagen representada por una matriz,  $\mathbf{A} = [a_{ij}]_{h \times w}$ , donde  $h$  (height) es la altura de la imagen y  $w$  (width) es el ancho de la imagen; y  $a$  representa el valor del  $ij$ -ésimo pixel:  $a \in \{0, 1, 2, \dots, 2^L - 1\}$ , donde  $L$  es el número de bits necesarios para representar el valor de un pixel.

**Definición 10** *bloque de la imagen (ib, image block)*: sea  $\mathbf{A} = [a_{ij}]$  una matriz de dimensiones  $h \times w$  que representa una imagen, y sea  $\mathbf{ib} = [ib_{ij}]$  una matriz  $d \times d$ . La matriz  $\mathbf{ib}$  se define como un bloque de la imagen  $\mathbf{A}$ , si la matriz  $\mathbf{ib}$  es un subconjunto de la matriz  $\mathbf{A}$  tal que:

$$ib_{ij} = a_{\delta_i \tau_j} \quad (4.1)$$

donde,  $i, j = 1, 2, \dots, d$ ;  $\delta = 1, d+1, 2d+1, \dots, h-d+1$ ;  $\tau = 1, d+1, 2d+1, \dots, w-d+1$ . y  $a_{\delta_i \tau_j}$  representa el valor de un pixel determinado por las coordenadas  $(\delta + i, \tau + j)$ , donde

$(\delta, \tau)$  y  $(\delta + d, \tau + d)$  son el inicio y fin del bloque de la imagen, respectivamente

**Definición 11** *vector de la imagen (iv, image vector)*: sea  $\mathbf{ib} = [ib_{ij}]$  un bloque de la imagen, y sea  $\mathbf{iv} = [iv_i]$  un vector de dimensión  $d$ . Se define como vector de una imagen a la  $k$ -ésima fila de la matriz  $\mathbf{ib}$  tal que:

$$iv_i = ib_{ki} \quad (4.2)$$

donde,  $i = 1, 2, \dots, d$ .

De cada bloque de la imagen se obtienen  $k$  vectores de la imagen.

$$\mathbf{iv}^k = [iv_i^k] = [ib_{ki}] \quad (4.3)$$

donde,  $k = 1, 2, \dots, d$ .

Por lo tanto.

$$\mathbf{x} = \{\mathbf{iv}^k : k = 1, 2, \dots, d\} \quad (4.4)$$

donde,  $x_p = iv_i^k | k, i = 1, 2, \dots, d, p = 1, 2, \dots, n, n = d \times d$ .

La nueva representación de la imagen está definida como

$$\mathbf{A} = \{\mathbf{x}^j : j = 1, 2, \dots, M\} \quad (4.5)$$

donde,  $M = (h/d)(w/d)$ .

## 4.2. Descripción del esquema de codificación VQ-LAMDA

El esquema propuesto consta de dos bloques, generación de libro de códigos LAMDA y esquema VQ-LAMDA; el primero está constituido por dos fases y el segundo por cuatro fases, en seguida se describe cada uno de los bloques mencionados.

### 4.2.1. Generación del libro de códigos-LAMDA

La obtención del libro de códigos LAMDA se lleva a cabo a través de dos fases (ver Figura 4.1)

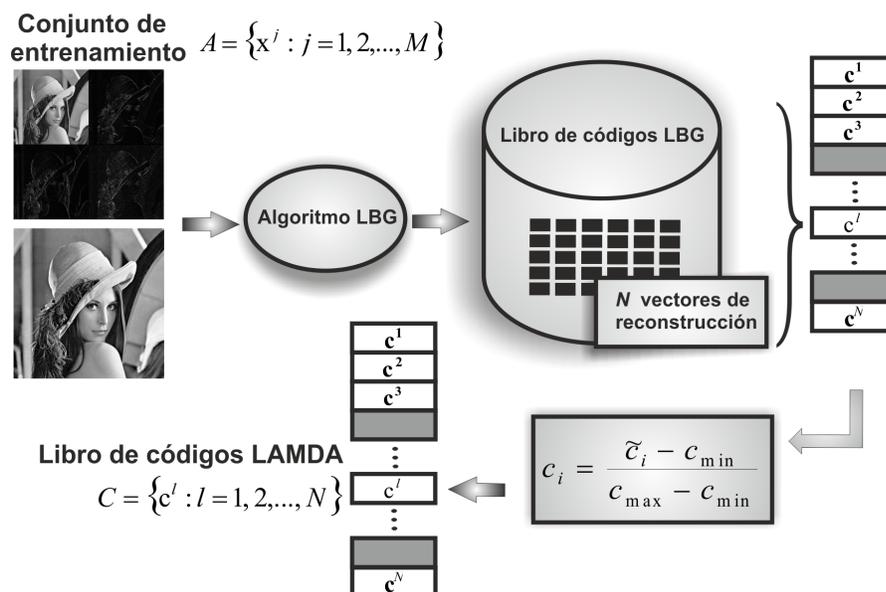


Figura 4.1: Esquema de la generación del libro de códigos LAMDA.

**Fase 1.** *Generación del libro de códigos LBG.* En esta fase se genera un libro de códigos basado en el algoritmo LBG. Este es un proceso supervisado; el conjunto de entrenamiento usado en la generación del libro de códigos puede ser formado por bloques de la imagen (codificación basada en VQ) o bloques de coeficientes obtenidos de una transformación (codificación basada en transformación, Transformada Discreta Coseno o Transformada Discreta Wavelet). Sea  $\mathbf{x} = [x_i]$  un vector de dimensión- $n$ , el cual representa un vector de la imagen; entonces, el conjunto de entrenamiento está definido por  $\mathbf{A} = \{\mathbf{x}^j : j = 1, 2, \dots, M\}$ . El resultado de esta etapa es un libro de códigos denotado como  $C = \{c^l : l = 1, 2, \dots, N\}$ , donde  $\mathbf{c} = [c_i]_n$ .

**Fase 2.** *Normalización del libro de códigos LBG.* Antes de usar el libro de códigos LBG, y para mantener coherencia con la lógica difusa, el libro de códigos debe ser normalizado usando (4.6).

$$c_i = \frac{\tilde{c}_i - c_{\min}}{c_{\max} - c_{\min}} = \frac{\tilde{c}_i}{2^L - 1} \quad (4.6)$$

donde,  $i = 1, 2, \dots, n$ ,  $\tilde{c}_i$  es el descriptor antes de ser normalizado,  $c_i$  es el descriptor normalizado,  $0 \leq c_i \leq 1$ ,  $c_{min} = 0$  y  $c_{max} = 2^L - 1$ ; en el contexto del procesamiento de imágenes,  $L$  es el número de bits necesario para representar el valor de un píxel. Los límites (mínimo y máximo) de los valores de los descriptores son los límites del conjunto de datos.

El conjunto de descriptores normalizados es denominado **libro de códigos-LAMDA** y es usado en el esquema **VQ-LAMDA**.

### 4.2.2. Esquema VQ-LAMDA

El esquema de cuantificación vectorial propuesto realiza la tarea de clasificación de acuerdo a un criterio pertenencia o similitud, calculada en cuatro fases.

**Fase 1.** *Normalización de los vectores renglón de la imagen.* Antes de usar los descriptores de los vectores renglón de la imagen en el esquema VQ-LAMDA, deben ser normalizados por medio de (4.7).

$$x_i = \frac{\tilde{x}_i - x_{min}}{x_{max} - x_{min}} = \frac{\tilde{x}_i}{2^L - 1} \quad (4.7)$$

donde  $i = 1, 2, \dots, n$ ,  $\tilde{x}_i$  es el descriptor antes de ser normalizado,  $x_i$  es el descriptor normalizado,  $0 \leq x_i \leq 1$ ,  $x_{min} = 0$  y  $x_{max} = 2^L - 1$ ; en el contexto del procesamiento de imágenes,  $L$  es el número de bits necesario para representar el valor de un píxel. Los límites (mínimo y máximo) de los valores de los descriptores son los límites del conjunto de datos.

**Fase 2.** *Grado de adecuación marginal (MAD).* Los MADs son calculados para cada descriptor  $x_i^j$  de cada vector de entrada  $\mathbf{x}^j$  con cada descriptor  $c_i^l$  de cada clase  $\mathbf{c}^l$ . Para este proposito, se utilizaron las siguientes distribuciones de probabilidad difusa y una medida de similitud (distancia city-block):

**Distribución binomial difusa:**

$$MAD(x_i^j/c_i^l) = (\rho_i^l)^{x_i^j} (1 - \rho_i^l)^{(1-x_i^j)} \quad (4.8)$$

donde  $i = 1, 2, \dots, n$ ;  $j = 1, 2, \dots, M$  y  $l = 1, 2, \dots, N$ . Para todas las distribuciones de probabilidad y la medida de similitud  $\rho_i^l = c_i^l$ .

**Distribución binomial-centro difusa:**

$$MAD(x_i^j/c_i^l) = \frac{(\rho_i^l)^{x_i^j} (1 - \rho_i^l)^{(1-x_i^j)}}{(x_i^j)^{x_i^j} (1 - x_i^j)^{(1-x_i^j)}} \quad (4.9)$$

**Distribución binomial-distancia difusa:**

$$MAD(x_i^j/c_i^l) = (a)^{(1-x_{dist})}(1-a)^{(x_{dist})} \quad (4.10)$$

donde  $a = \max |(\rho_i^l), (1 - \rho_i^l)|$  y  $x_{dist} = |x_i^j - \rho_i^l|$ .

**Función Gaussiana:**

$$MAD(x_i^j/c_i^l) = e^{-\frac{1}{2} \left( \frac{x_i^j - \rho_i^l}{\sigma^2} \right)^2} \quad (4.11)$$

donde  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i^j$  y  $\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i^j - \bar{x})^2$ , son la media aritmética y la varianza del vector  $\mathbf{x}^j$ , respectivamente.

**Distancia city-block:**

$$MAD(x_i^j/c_i^l) = |x_i^j - \rho_i^l| \quad (4.12)$$

**Fase 3.** *Grado de adecuación global (GAD).* Esta etapa determina el grado de pertenencia de cada vector de entrada  $\mathbf{x}^j$  a cada clase  $\mathbf{c}^l$ , por medio de la función lineal convexa (4.13) y el uso de operadores de agregación matemáticos (T-normas y S-conormas), los operadores utilizados se muestran en la Tabla 4.1.

Tabla 4.1: T-Normas y S-Conormas usadas por el esquema VQ-LAMDA

Operador	T-Norma (Intersección)	S-Conorma (Unión)
min-max	$\min (MAD(x_i^j/c_i^l))$	$\max (MAD(x_i^j/c_i^l))$
producto	$\prod_{i=1}^n MAD(x_i^j/c_i^l)$	$1 - (\prod_{i=1}^n MAD(x_i^j/c_i^l))$
media aritmética	$\frac{1}{n} \sum_{i=1}^n MAD(x_i^j/c_i^l)$	$1 - (\frac{1}{n} \sum_{i=1}^n MAD(x_i^j/c_i^l))$
suma	$\sum_{i=1}^n MAD(x_i^j/c_i^l)$	$1 - (\sum_{i=1}^n MAD(x_i^j/c_i^l))$

$$GAD_{c^l}(\mathbf{x}^j) = L_{\alpha}^{T,S} = (\alpha) \cdot T(MAD(x_i^j/c_i^l)) + (1 - \alpha) \cdot S(MAD(x_i^j/c_i^l)) \quad (4.13)$$

donde  $\alpha \in [0, 1]$ ,  $T \leq L_{\alpha}^{T,S} \leq S$ ,  $T = L_1^{T,S}$  (intersección) y  $S = L_0^{T,S}$  (unión). Al parámetro  $\alpha$  se le denomina nivel de exigencia.

**Fase 4.** *Obtención del índice.* Finalmente, en esta fase se obtiene el índice de la clase a la cual pertenece el vector de entrada. Según sea el caso, se usará la regla de máxima o de mínima adecuación.

Cuando se utiliza una distribución de probabilidad. El índice es determinado por el GAD que presenta el valor máximo (regla de máxima adecuación, MA).

$$indice = \max(GAD_{c^1}(\mathbf{x}^j), GAD_{c^2}(\mathbf{x}^j), \dots, GAD_{c^N}(\mathbf{x}^j)) \quad (4.14)$$

En el caso de la medida de similitud (distancia city-block). El índice es determinado por el GAD que presenta el valor mínimo (regla de mínima adecuación, mA).

$$indice = \min(GAD_{c^1}(\mathbf{x}^j), GAD_{c^2}(\mathbf{x}^j), \dots, GAD_{c^N}(\mathbf{x}^j)) \quad (4.15)$$

En la Figura 4.2, se muestra el esquema para cuantificación vectorial propuesto, como se menciona anteriormente se utilizó el algoritmo LAMDA como clasificador y el algoritmo LBG como generador del libro de códigos.

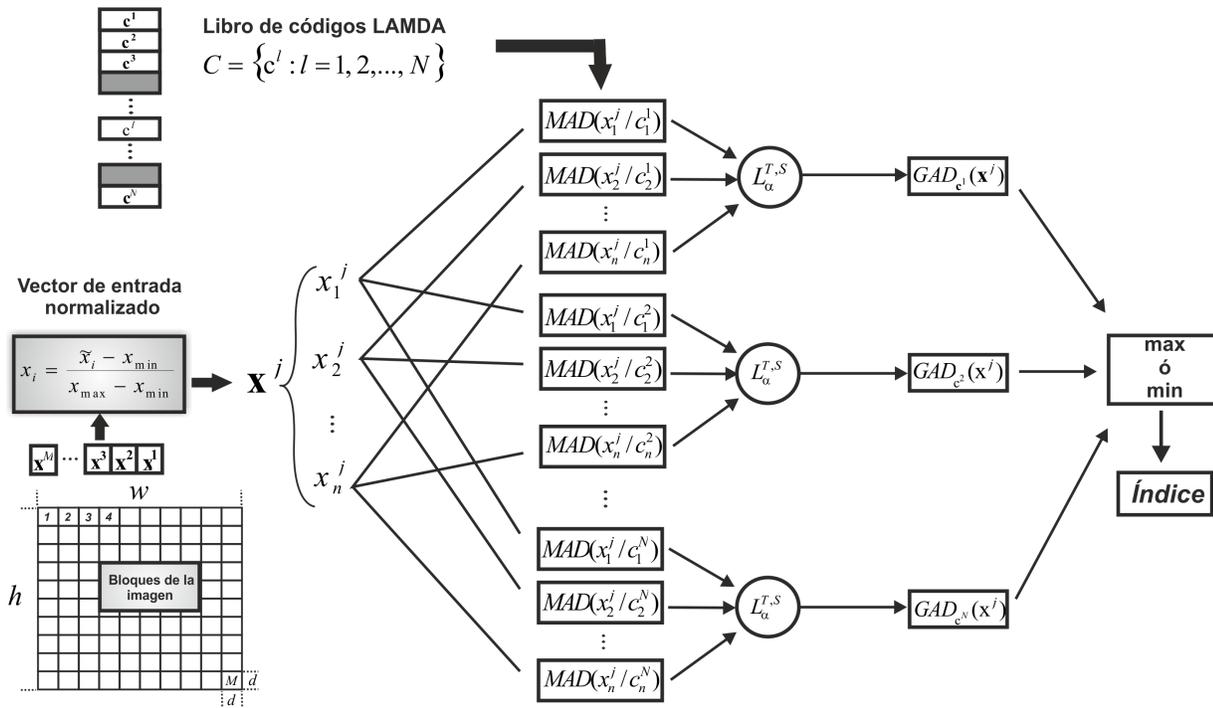


Figura 4.2: Diagrama a bloques del proceso de codificación del esquema VQ-LAMDA.

### 4.3. Proceso de decodificación

El proceso de reconstrucción consiste en un algoritmo de búsqueda, que recibe como entrada el conjunto de índices generado por el esquema de cuantificación vectorial VQ-

---

LAMDA. Cabe hacer la aclaración que tanto en el codificador como en el decodificador se debe tener el mismo libro de códigos LAMDA. Al terminar el proceso de asignación de cada índice a los vectores de códigos correspondientes, se obtiene la imagen reconstruida que no será igual a la imagen original debido a la pérdida de datos inherente a los esquemas de cuantificación vectorial.



# Capítulo 5

## Resultados experimentales

En este capítulo se presentan los resultados experimentales obtenidos al aplicar el esquema propuesto VQ-LAMDA a la cuantificación vectorial de imágenes, para ello se hicieron tres experimentos.

En primer lugar se hizo la caracterización de las ecuaciones del esquema propuesto. Posteriormente se comparó el desempeño del algoritmo VQ-LAMDA con respecto a los algoritmos de cuantificación vectorial LBG y FSA-EAM, en términos de la relación señal a ruido. Finalmente se analizó la complejidad en tiempo y espacio del esquema propuesto; además, se calculó el número y tipo de operaciones utilizadas por el esquema VQ-LAMDA, después se hizo una comparación con los algoritmos LBG y FSA-EAM.

En todos los experimentos se usó un conjunto de cuatro imágenes estándar (ver Figura 5.1 (a) Lena, (b) Peppers, (c) Elaine, (d) Man.), de dimensión  $512 \times 512$  píxeles, y de 8 bits por píxel, las cuales fueron divididas en bloques de tamaño  $4 \times 4$ , los bloques no están traslapados, estos bloques se convierten a vectores de dimensión 16 ( $n$ ). Los libros de códigos ( $N$ ) fueron generados con el algoritmo LBG, como imagen de entrenamiento se usó la imagen Lena.



Figura 5.1: Imágenes de prueba usadas para medir el desempeño del esquema VQ-LAMDA

Con el fin de evaluar el desempeño del algoritmo propuesto VQ-LAMDA, se utilizó el criterio de desempeño llamado relación señal a ruido (PSNR, *peak signal-to-noise ratio*), el cual se calcula mediante la siguiente expresión:

$$PSNR = 10 \log_{10} \left( \frac{(2^L - 1)^2}{\frac{1}{M} \sum_{i=1}^M (p_i - \tilde{p}_i)^2} \right) \quad (5.1)$$

donde  $L$  es el número de bits por píxel,  $M$  es el número de píxeles en la imagen,  $p_i$  es el  $i$ -ésimo píxel en la imagen original, y  $\tilde{p}_i$  es el  $i$ -ésimo píxel en la imagen reconstruida.

## 5.1. Primer experimento

En el primer experimento se hizo la caracterización de cada una de las ecuaciones obtenidas al combinar cada una de las distribuciones de probabilidad difusa y la distancia de similitud especificadas en la sección 4.2.2, fase 2 (cálculo de las GADS); con los operadores de agregación de la Tabla 4.1 y la función lineal convexa determinados en la fase 3 (cálculo de las GADS) de la misma sección.

Como se tienen cinco funciones y cuatro operadores de agregación, esto dio como resultado veinte ecuaciones posibles. Con el fin de determinar cuáles de esas veinte ecuaciones eran las mejores para ser usadas en cuantificación vectorial, se evaluaron cada una de las ecuaciones variándole el nivel de exigencia, en pasos de 0.1 dentro del rango de valores  $\alpha \in [0, 1]$ . Se usó (5.1) para obtener el desempeño. En la Tabla 5.1, se muestran los resultados obtenidos para el libro de códigos de tamaño 64, como se puede observar se dio una posición a cada ecuación de acuerdo al desempeño que tuvo cada una de ellas, con base a los resultados de esta caracterización se determino que las ecuaciones (1.5), (1.6), (1.7), (1.8) y (1.9) tienen un mejor desempeño en términos de la relación señal a ruido y pueden ser aplicadas en cuantificación vectorial, algunas de esas ecuaciones dan el mismo resultado dentro de un rango, por lo tanto, se puede usar cualquier valor para el nivel de exigencia  $\alpha$  dentro de ese rango; con el fin de reducir el número de operaciones realizadas y en consecuencia la complejidad, en los rangos donde  $\alpha$  va desde 0.51 a 1, se opto por usar  $\alpha = 1$ . En el caso de la ecuación city-block (min-max), el rango es 0.4-0.6, se seleccionó el nivel de exigencia igual a 0.5. El mismo procedimiento se hizo con los tres libros de códigos restantes (128, 256 y 512), en los cuales la tendencia fue la misma.

Tabla 5.1: Caracterización de las veinte ecuaciones obtenidas.

Posición	Ecuación	Nivel de exigencia ( $\alpha$ )	Operador de agregación	Libro de códigos	Relación señal a ruido (dB)			
					Elaine	Lena	Man	Peppers
4	Binomial	0.4-0.6			27.29	27.17	25.16	26.27
2	Binomial centro	0.1-0.2	Min-Max	64x16	28.43	28.18	26.06	27.15
5	Binomial distancia	1			20.36	20.04	20.21	20.95
3	Función Gausiana	1			27.94	28.11	25.78	26.53
1	Distancia city-block	0.4-0.6			28.45	28.20	26.06	27.17
1	Binomial	0.51-1			28.92	28.80	26.81	27.80
1	Binomial centro	0.51-1	Producto	64x16	28.92	28.80	26.81	27.80
4	Binomial distancia	0.51-1			19.27	19.39	19.15	20.16
2	Función Gausiana	1			28.46	28.83	26.28	27.81
3	Distancia city-block	0.51-1			16.62	16.49	16.14	16.52
2	Binomial	0.51-1			28.91	28.79	26.80	27.75
1	Binomial centro	0.51-1	Promedio	64x16	28.91	28.80	26.81	27.80
5	Binomial distancia	0.51-1			19.24	19.35	19.10	20.08
4	Función Gausiana	1			27.89	28.21	25.86	27.34
3	Distancia city-block	0.51-1			28.79	28.66	26.59	27.43
2	Binomial	0.51-1			28.91	28.79	26.80	27.75
1	Binomial centro	0.51-1	Suma	64x16	28.91	28.80	26.81	27.80
5	Binomial distancia	0.51-1			19.24	19.35	19.10	20.08
4	Función Gausiana	1			27.89	28.21	25.86	27.34
3	Distancia city-block	0.51-1			28.79	28.66	26.59	27.43

## 5.2. Segundo experimento

En este experimento también se obtuvo el desempeño, sólo que en este experimento se tiene el objetivo de comparar el desempeño del esquema VQ-LAMDA con respecto a los algoritmos LBG y FSA-EAM. En la Tabla 5.2 se puede observar que los resultados de las siguientes configuraciones en términos del PSNR son prácticamente los mismos, además son los valores más altos que se obtuvieron: Binomial-producto, Binomial centro-producto, Binomial centro-promedio y Binomial centro-suma. En el caso de City block-min max, sus valores se ubican en el medio ya que ni son tan altos con respecto a las configuraciones anteriores, ni son tan bajos como los obtenidos por los algoritmos LBG y FSA-EAM. La misma tendencia se mantuvo en el desempeño utilizando los diferentes tamaños de libros de

códigos. En las figuras 5.2, 5.3, 5.4 y 5.5 se pueden observar las imágenes reconstruidas con las siguientes configuraciones: inciso a), imágenes originales; inciso b), Binomial-producto; inciso c), Binomial centro-producto; inciso d), Binomial centro-promedio; inciso e), Binomial centro-suma y el inciso f), City block-min max. En la parte inferior de cada una de ellas se especifica el PSNR obtenido al ser procesadas con el esquema VQ-LAMDA.

Tabla 5.2: Tabla comparativa entre los algoritmos VQ-LAMDA, LBG y FSA-EAM.

Algoritmo	Nivel de exigencia ( $\alpha$ )	Operador de agregación	Libro de códigos	Relación señal a ruido (dB)			
				Elaine	Lena	Man	Peppers
VQ-LAMDA (Binomial)	1	Producto	64x16	28.92	28.80	26.81	27.80
VQ-LAMDA (Binomial centro)		Producto		28.92	28.80	26.81	27.80
VQ-LAMDA (Binomial centro)		Promedio		28.91	28.80	26.81	27.80
VQ-LAMDA (Binomial centro)		Suma		28.91	28.80	26.81	27.80
VQ-LAMDA (city-block)	0.5	Min-Max		28.45	28.20	26.06	27.17
LBG	-	-		28.91	27.14	24.20	26.38
FSA-EAM	-	-		28.31	26.31	23.28	25.12
VQ-LAMDA (Binomial)	1	Producto	128x16	29.69	29.73	27.53	28.60
VQ-LAMDA (Binomial centro)		Producto		29.69	29.73	27.53	28.60
VQ-LAMDA (Binomial centro)		Promedio		29.69	29.73	27.53	28.60
VQ-LAMDA (Binomial centro)		Suma		29.69	29.73	27.53	28.60
VQ-LAMDA (city-block)	0.5	Min-Max		29.16	29.17	26.76	27.90
LBG	-	-		29.68	28.21	24.93	27.18
FSA-EAM	-	-		29.07	27.47	23.96	26.06
VQ-LAMDA (Binomial)	1	Producto	256x16	30.31	30.44	28.25	29.13
VQ-LAMDA (Binomial centro)		Producto		30.31	30.44	28.25	29.13
VQ-LAMDA (Binomial centro)		Promedio		30.31	30.44	28.25	29.13
VQ-LAMDA (Binomial centro)		Suma		30.31	30.44	28.25	29.13
VQ-LAMDA (city-block)	0.5	Min-Max		29.76	29.87	27.48	28.42
LBG	-	-		30.30	29.08	25.49	27.64
FSA-EAM	-	-		29.68	28.39	24.55	26.54
VQ-LAMDA (Binomial)	1	Producto	512x16	30.76	31.05	28.73	29.74
VQ-LAMDA (Binomial centro)		Producto		30.76	31.05	28.73	29.74
VQ-LAMDA (Binomial centro)		Promedio		30.76	31.04	28.73	29.74
VQ-LAMDA (Binomial centro)		Suma		30.76	31.04	28.73	29.74
VQ-LAMDA (city-block)	0.5	Min-Max		30.18	30.44	27.96	29.05
LBG	-	-		30.74	29.98	26.02	28.21
FSA-EAM	-	-		30.11	29.25	25.09	27.04

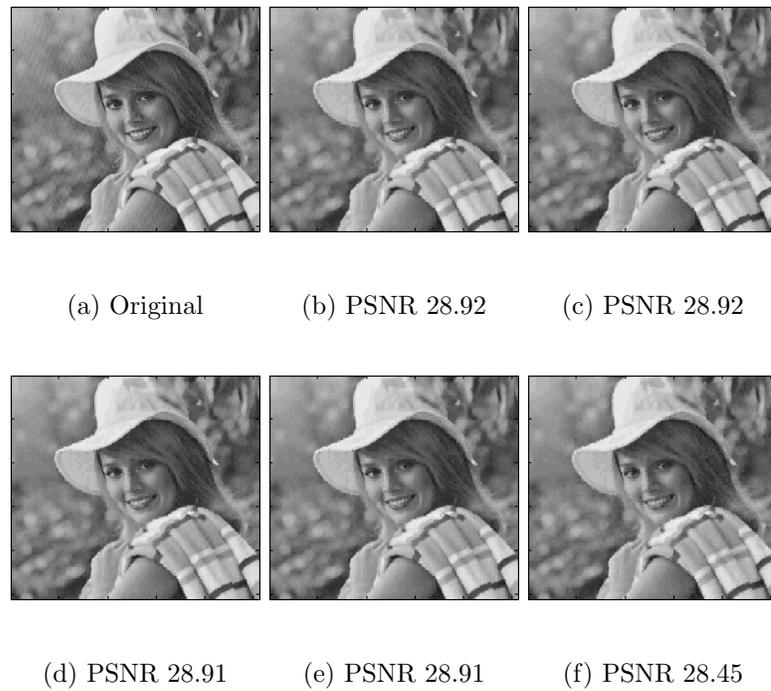


Figura 5.2: Imagen Elaine procesada con un libro de códigos de tamaño  $64 \times 16$

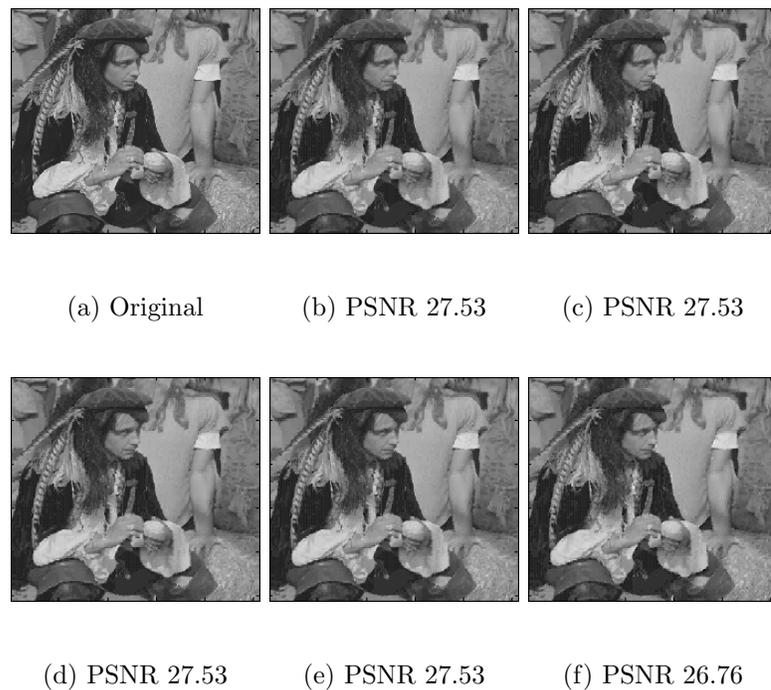


Figura 5.3: Imagen Man procesada con un libro de códigos de tamaño  $128 \times 16$

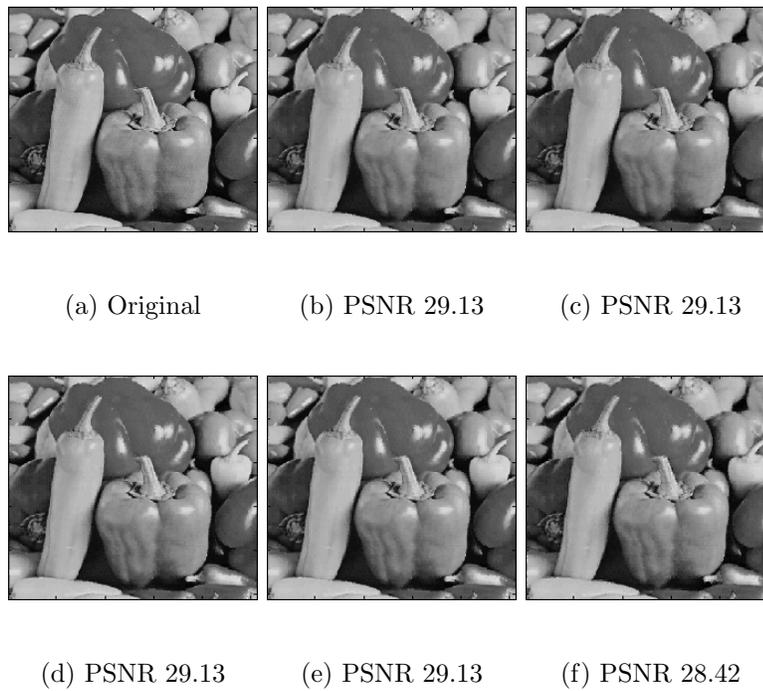


Figura 5.4: Imagen Peppers procesada con un libro de códigos de tamaño  $256 \times 16$



Figura 5.5: Imagen Lena procesada con un libro de códigos de tamaño  $512 \times 16$

## 5.3. Tercer experimento

El tercer experimento consta de dos tareas, la primera es determinar la complejidad en tiempo y espacio. La segunda es calcular el número y tipo de operaciones que se necesitan para implementar el esquema propuesto en sus diferentes configuraciones, también se muestra la cantidad requerida de espacio en memoria.

### 5.3.1. Complejidad del esquema propuesto

La complejidad de los algoritmos se mide mediante dos parámetros: la complejidad en tiempo y la complejidad en espacio. En esta sección, se calcula y analiza la complejidad en tiempo y en espacio del esquema VQ-LAMDA. Para ello se consideran los pseudo-códigos 5.3.1 y 5.3.2.

#### 5.3.1.1. Complejidad en tiempo

Para medir la complejidad en tiempo del algoritmo VQ-LAMDA. Primero se estima el tiempo de ejecución de la configuración que usa la distribución binomial difusa y el operador de agregación, producto; para ello se calculan el número de operaciones elementales (OE) que se necesitan para clasificar un patrón, como OE se consideran las siguientes operaciones: asignaciones, acceso a matriz o vector, comparación, suma, resta, división y exponente.

Considere el pseudo-código 5.3.1. El ciclo interno abarca las líneas 7 a 10, las líneas 8, 9 y 10 se ejecutan en todas las iteraciones de ese ciclo, en la línea 7 se ejecutan  $2n + 2$  OE, en la línea 8 se ejecutan 4 OE, en la línea 9 se ejecutan 10 OE y en la línea 10 se ejecutan 2 OE. Por lo tanto, en el ciclo interno se ejecutan  $2n + 2 + 4 + 10 + 2 = 2n + 18$  OE.

En el ciclo externo que incluye las líneas 5 a 13, las líneas 6 a 13 se ejecutan en todas las iteraciones. En la línea 5 se ejecutan  $2N + 2$  OE, en la línea 6 se ejecuta 1 OE, en la líneas 11 a la 13 se ejecutan 3 OE. El número total de operaciones ejecutadas por esas líneas son  $2N + 2 + 1 + 3 = 2N + 6$ , pero multiplicado por el número de operaciones elementales del ciclo interno da como resultado.

$$(2n + 18)(2N + 6) = 4Nn + 36N + 12n + 108 \quad (5.2)$$

En las líneas 2 a la 4 son ejecutadas 3 OE y en la línea 14 se ejecutan 2 OE, en total son 5 OE que son sumadas al resultado (5.2). El número total de operaciones elementales son:

$$4Nn + 36N + 12n + 108 + 5 = 4Nn + 36N + 12n + 113 \quad (5.3)$$

donde  $N$  es el tamaño del libro de códigos y  $n$  es la dimensión del patrón.

Considerando la expresión anterior, se puede concluir que el orden de crecimiento de la configuración que usa la distribución binomial difusa y el operador de agregación producto es  $O(Nn)$ . El mismo procedimiento se siguió para calcular el orden de crecimiento de las configuraciones, Binomial centro-producto, Binomial centro-promedio y Binomial centro-suma (en el Apéndice A se muestran los pseudo-códigos de dichas configuraciones), en la Tabla 5.3 se muestran los resultados obtenidos.

**Algorithm 5.3.1:** BINOMIAL-PRODUCTO( $VQ - LAMDA$ )

```

1.- comment: Calcula el índice usando una distribución de probabilidad
2.-  $value \leftarrow 0$ 
3.-  $index \leftarrow 1$ 
4.-  $j \leftarrow 1$ 
5.- for  $l \leftarrow 1$  to  $N$ 
   do {
     6.-  $GADS \leftarrow 1$ 
     7.- for  $i \leftarrow 1$  to  $n$ 
       do {
         8.-  $X[j][i] \leftarrow X[j][i]/255$ 
         9.-  $MADbin \leftarrow P[l][i] \wedge X[j][i] * (1 - P[l][i]) \wedge (1 - X[j][i])$ 
         10.-  $GADS \leftarrow GADS * MADbin$ 
       }
     11.- if  $GADS > value$ 
       then {
         12.-  $value \leftarrow GADS$ 
         13.-  $index \leftarrow l$ 
       }
   }
14.-  $indice[j] \leftarrow index$ 

```

Ahora se considera la configuración city block-min max. Se hace aparte el análisis debido a que es la única configuración que ocupa la distancia de similitud, para ello se calculan el número de operaciones elementales (OE) que se necesitan para clasificar un patrón, considerar el pseudo-código 5.3.2.

En el ciclo interno que abarca las líneas 9 a 17, las líneas 10 y 17 se ejecutan en todas las iteraciones, la ejecución de las líneas 12 y 13 dependen de la comparación de la línea 11

y la ejecución de las líneas 15 y 16 dependen de la comparación de la línea 14. Se toma en cuenta el mejor caso, es decir, que sólo uno de los ciclos condicionales (if) se ejecutan en todas las iteraciones. tomando en cuenta lo anterior el número de operaciones elementales que se ejecutan en el ciclo interno son las siguientes.

En la línea 9 se ejecutan  $2n + 2$  OE, en la línea 10 se ejecutan 5 OE, si se asume que se cumple la condición de la línea 11, se ejecutan 3 OE, tomando las operaciones elementales de las líneas 11, 12 y 13.

**Algorithm 5.3.2:** CITY BLOCK-MIN MAX( $VQ - LAMDA$ )

```

1.- comment: Calcula el índice usando la distancia de similitud
2.-  $value \leftarrow 256$ 
3.-  $index \leftarrow 1$ 
4.-  $j \leftarrow 1$ 
5.-  $alfa \leftarrow 0.5$ 
6.- for  $l \leftarrow 1$  to  $N$ 
   {
7.-  $aux0 \leftarrow 0$ 
8.-  $aux1 \leftarrow 256$ 
9.- for  $i \leftarrow 1$  to  $n$ 
   {
10.-  $MADcity \leftarrow |X[j][i] - P[l][i]|$ 
11.- if  $MADcity > aux0$ 
   {
12.-  $aux0 \leftarrow MADcity$ 
13.-  $G1 \leftarrow aux0$ 
14.- if  $MADcity < aux1$ 
   {
15.-  $aux1 \leftarrow MADcity$ 
16.-  $G2 \leftarrow aux1$ 
17.-  $GADS \leftarrow (alfa * G2) + ((1 - alfa) * G1)$ 
18.- if  $GADS < value$ 
   {
19.-  $value \leftarrow GADS$ 
20.-  $index \leftarrow l$ 
21.-  $indice[j] \leftarrow index$ 

```

En la línea 17 se ejecutan 5 OE, por lo tanto, el número total de operaciones del ciclo interno son:

$$2n + 2 + 5 + 3 + 5 = 2n + 15 \quad (5.4)$$

En el ciclo externo que incluye las líneas 6 a la 20, la línea 6 ejecuta  $2N + 2$  OE, en las líneas 7 y 8 se ejecutan 2 OE todas las iteraciones. Las líneas 19 y 20 dependen de la condición de la línea 18. Se hará la suposición que siempre va a entrar al ciclo condicional por lo que se ejecutarían 3 OE. Tomando en cuenta lo anterior el número total de operaciones elementales son  $2N + 2 + 2 + 3 = 2N + 7$ , pero multiplicado por el número de operaciones elementales del ciclo interno, da como resultado.

$$(2n + 15)(2N + 7) = 4Nn + 30N + 14n + 105 \quad (5.5)$$

En las líneas 2 a la 5 son ejecutadas 4 OE y en la línea 21 se ejecutan 2 OE, en total son 6 OE que son sumadas al resultado (5.5). El número total de operaciones elementales son:

$$4Nn + 30N + 14n + 105 + 6 = 4Nn + 30N + 14n + 111 \quad (5.6)$$

donde  $N$  es el tamaño del libro de códigos y  $n$  es la dimensión del patrón.

Considerando la expresión anterior, se puede concluir que el orden de crecimiento de la configuración que usa la medida de similitud city-block y el operador de agregación, min-max es  $O(Nn)$ . En la Tabla 5.3 se muestra este resultado.

### 5.3.1.2. Complejidad en espacio

La complejidad en espacio esta determinada por la cantidad de memoria requerida para la ejecución del algoritmo. Para cuantificar una imagen de dimensión  $h \times w$ , la cual contiene  $M = (h \times w)/n$ , número de patrones de dimensión  $n$ , el esquema VQ-LAMDA requiere una matriz de tamaño  $[M][n]$  que representa los vectores de la imagen a ser procesados en la misma matriz se guardan los vectores de la imagen normalizados, una matriz de tamaño  $[N][n]$  la cual contiene el libro de códigos LAMDA y finalmente un vector de tamaño  $[M]$ , para almacenar los valores de los índices. Por lo tanto, el número de unidades de memoria requeridos para procesar toda la imagen es

$$(M)(n) + (N)(n) + (M) = n(M + N) + (M) \quad (5.7)$$

Las diferentes configuraciones del esquema propuesto usan las siguientes operaciones, comparaciones, sumas, restas, productos, divisiones y potencias. Por lo tanto, el resultado siempre da un número de punto flotante. Para el procesamiento de una imagen en escala de grises, 8 bits/pixel, el esquema VQ-LAMDA requiere variables de tipo *float* de 32 bits. Por consiguiente, el número total de *bytes* requeridos son:

$$4(n(M + N) + (M)) \quad (5.8)$$

El número total de unidades de memoria dependerá del tamaño de la imagen ( $M$ ), tamaño del libro de códigos ( $N$ ) y el tamaño del vector de códigos ( $n$ ), seleccionados para el proceso de cuantificación vectorial.

Por último el proceso se repite para las configuraciones restantes, en la Tabla 5.3 se muestran los resultados obtenidos para todas las configuraciones.

Tabla 5.3: Complejidad en tiempo y espacio de las mejores configuraciones.

VQ-LAMDA	Complejidad	
	Espacio	Tiempo
Binomial (Producto)	$4(n(M + N) + (M))$	$O(Nn)$
Binomial centro (Producto)	$4(n(M + N) + (M))$	$O(Nn)$
Binomial centro (Promedio)	$4(n(M + N) + (M))$	$O(Nn)$
Binomial centro (Suma)	$4(n(M + N) + (M))$	$O(Nn)$
Distancia city-block (Min-Max)	$4(n(M + N) + (M))$	$O(Nn)$

Haciendo un análisis de los resultados expuestos en la Tabla 5.3, tomando en cuenta la complejidad en tiempo se podría pensar que todas las configuraciones se ejecutan en el mismo tiempo, pero en realidad no es así. Esto se debe a que para calcular la complejidad en tiempo se consideraron como operaciones elementales las asignaciones, acceso a matriz o vector, comparación, suma, resta, división y exponente. No se debe olvidar que el tiempo de ejecución de cada una de ellas no es el mismo, por ello es que el tiempo de ejecución del algoritmo que utiliza la medida de similitud es menor ya que sólo utiliza comparaciones,

sumas, restas y multiplicaciones; por otro lado las configuraciones basadas en distribuciones de probabilidad utilizan comparaciones, sumas, restas, multiplicaciones, divisiones y exponentes. Esto queda más claro en la siguiente subsección.

### 5.3.2. Cálculo del número y tipo de operaciones del esquema VQ-LAMDA

En esta subsección se cálculo el número de operaciones que se requieren para cuantificar un pixel, también se especifica el tipo de operación (comparación, suma o resta, raíz cuadrada, división o exponente). Para tal efecto, se examina el pseudo-código 5.3.1.

Lo primero que se hace es determinar el tipo de operación y su ubicación dentro del algoritmo. En el ciclo interno que comprende las líneas 7 a la 10, hay 1 división, 2 potencias, 2 multiplicaciones y 2 restas. Para calcular el número de operaciones requeridas para cuantificar un píxel se utiliza la siguiente expresión

$$oper\_pixel_1 = \frac{(oper \times n) \times N}{n} \quad (5.9)$$

donde  $N$  es el tamaño del libro de códigos,  $n$  es la dimensión del patrón,  $oper$  es el número de operaciones que están dentro del ciclo que se está evaluando.

En el ciclo externo que incluye las líneas 5 a la 13, sólo se encuentra una comparación, esa operación se realiza  $N$  veces, pero como se desea calcular el número de operaciones por píxel, se divide entre la dimensión del patrón.

$$oper\_pixel_2 = \frac{N}{n} \quad (5.10)$$

Mediante las ecuaciones (5.9) y (5.10), se obtuvieron los resultados para la configuración Binomial-producto, dichos resultados se encuentran expresados en la Tabla 5.4. Para calcular el numero de operaciones por pixel de las demás configuraciones que tienen distribuciones de probabilidad se hizo el mismo procedimiento.

A continuación se muestra como se obtuvo el número de operaciones por pixel de la configuración que usa la distancia de similitud. Para tal efecto, se inspecciona el pseudo-código 5.3.2.

Primero se determina el tipo de operación y su ubicación dentro del algoritmo. En el ciclo interno que comprende de la línea 9 a la línea 17, hay 3 elementos que pertenecen a una

Tabla 5.4: Operaciones requeridas por los algoritmos LBG, FSA-EAM y VQ-LAMDA para cuantificar un pixel.

Algoritmo	Operador de agregación	Libro de códigos	Número promedio de operaciones por pixel					Memoria usada (bytes)	
			Comparaciones	Sumas o restas	Multiplicaciones	Raíz cuadrada	Divisiones		Exponentes
VQ-LAMDA (Binomial)	Producto		8	256	256	-	128	256	1122304
VQ-LAMDA (Binomial centro)	Producto		8	512	384	-	256	512	1122304
VQ-LAMDA (Binomial centro)	Promedio		8	640	256	-	384	512	1122304
VQ-LAMDA (Binomial centro)	Suma	128x16	8	640	256	-	256	512	1122304
VQ-LAMDA (city-block)	Min - Max		264	384	256	-	-	-	1122304
LBG	-		8	387	129	8.062	-	-	41088
FSA-EAM	-		136	128	-	-	-	-	37080
VQ-LAMDA (Binomial)	Producto		16	512	512	-	256	512	1130496
VQ-LAMDA (Binomial centro)	Producto		16	1024	768	-	512	1024	1130496
VQ-LAMDA (Binomial centro)	Promedio		16	1280	512	-	768	1024	1130496
VQ-LAMDA (Binomial centro)	Suma	256x16	16	1280	512	-	512	1024	1130496
VQ-LAMDA (city-block)	Min - Max		528	768	512	-	-	-	1130496
LBG	-		16	771	257	16.062	-	-	49280
FSA-EAM	-		272	256	-	-	-	-	41432
VQ-LAMDA (Binomial)	Producto		32	1024	1024	-	512	1024	1146880
VQ-LAMDA (Binomial centro)	Producto		32	2048	1536	-	1024	2048	1146880
VQ-LAMDA (Binomial centro)	Promedio		32	2560	1024	-	1536	2048	1146880
VQ-LAMDA (Binomial centro)	Suma	512x16	32	2560	1024	-	1024	2048	1146880
VQ-LAMDA (city-block)	Min - Max		1056	1536	1024	-	-	-	1146880
LBG	-		32	1539	513	32.062	-	-	65664
FSA-EAM	-		544	512	-	-	-	-	50136

suma o resta, 2 multiplicaciones y 2 comparaciones. Para calcular el número de operaciones requeridas para cuantificar un píxel se utiliza la expresión (5.9), las comparaciones no se calculan con esa expresión.

En este ciclo, para evaluar las comparaciones se toma el peor caso, que es cuando se ejecutan las dos comparaciones en cada iteración y además se le suma la comparación de la línea 18, que se ejecutarán  $N$  veces, para calcular ahora sí todas las comparaciones se utiliza la expresión.

$$oper\_pixel_3 = \frac{((oper \times n) \times N) + N}{n} \quad (5.11)$$

En la Tabla 5.4, se encuentran los resultados obtenidos para la medida de similitud city-block.

# Capítulo 6

## Conclusiones y Trabajo a Futuro

Este capítulo contiene las conclusiones a las que se ha llegado después de implementar las diferentes configuraciones del esquema alternativo para la cuantificación vectorial de imágenes, denominado VQ-LAMDA. También, se exponen trabajos futuros que se deben seguir para continuar con esta investigación.

- Se comprobó experimentalmente que usando la medida de similitud city-block, para calcular los MADs, se puede omitir el proceso de normalización de los libros de códigos y de los vectores de entrada, esto trae como consecuencia que se ejecuten menos operaciones por pixel y por ende el tiempo de ejecución mejora, con respecto a las distribuciones de probabilidad que si deben ser normalizadas. Además esto demuestra que el uso de medidas de similitud es viable, sólo se debe cambiar la regla de máxima adecuación por una regla de mínima adecuación.
- Se utilizó el operador de agregación media aritmética propuesto en [3], para calcular los GADs, los resultados obtenidos demostraron que es factible.
- En este trabajo se propuso el uso de la sumatoria para calcular los GADs, la sumatoria es un operador de agregación extendido, este operador de agregación no aparece en la bibliografía consultada referente al algoritmo LAMDA.
- El hecho de que esta metodología tiene la habilidad de controlar la selectividad de la clasificación a través del parámetro  $\alpha$ , fue fundamental cuando se hizo la caracterización de las veinte configuraciones, ya que permitió determinar las cinco mejores configuraciones para ser usadas en cuantificación vectorial.

- La razón por la cual se seleccionaron los operadores media aritmética y sumatoria, es porque son operadores que realizan un menor número de operaciones, en comparación con Lukasiewicz, Yager y Hammacher. Esto tiene relevancia en la cuantificación vectorial debido a que la velocidad de procesamiento es una cuestión esencial.
- Las configuraciones Binomial-producto y Binomial centro-producto, son configuraciones del algoritmo LAMDA como originalmente fue diseñado. De las modificaciones hechas a la versión original del algoritmo LAMDA, se derivaron las siguientes configuraciones, Binomial Centro-promedio, Binomial Centro-suma y distancia City Block-min max, se puede observar que los tres cambios propuestos fueron útiles. La distribución de probabilidad que ofrece los mejores resultados, es la distribución Binomial centro desde que logro buenos resultados con tres operadores de agregación, pero la desventaja que tiene es que ejecuta más operaciones.
- Los resultados obtenidos en términos del PSNR por las ecuaciones que utilizan distribuciones de probabilidad difusa son muy similares, en la mayoría de los casos se obtiene el mismo resultado. Por lo tanto, cuando se elija alguna de ellas, se optara por la distribución binomial ya que es la que realiza menor número de operaciones.
- En el caso de la medida de similitud distancia city-block se obtuvo menor desempeño en términos del PSNR aunque la diferencia en el desempeño no es tan grande, con respecto a las distribuciones de probabilidad difusa; y comparada con los algoritmos LBG y FS-EAM, la calidad de la codificación permanece competitiva.
- La desventaja que tienen las cinco configuraciones del esquema VQ-LAMDA es que en términos del número de operaciones por pixel realizan más operaciones que los algoritmos LBG y FSA-EAM.
- La ventaja de haber hecho las modificaciones para calcular los MADs y los GADs en el algoritmo original, es que se obtuvo un algoritmo más robusto y eficiente.
- Finalmente, como se mostró en la Tabla 5.2, los resultados obtenidos del esquema propuesto proporciona mejor relación señal a ruido (PSNR) que los algoritmos LBG y FSA-EAM.

## 6.1. Trabajo a futuro

De la investigación y propuestas presentadas en este trabajo de tesis, surgieron nuevos temas de investigación que sirven para complementar la contribución hecha en este trabajo.

- En el caso del cálculo de los MADs se demostró que es viable usar medidas de similitud, por lo tanto, se podrían adecuar medidas como la distancia Euclidiana (6.1) y distancia coseno (6.2) [30].

$$d_E = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (6.1)$$

$$d_{\cos} = 1 - \frac{\sum_{i=1}^n x_i \times y_i}{\sqrt{\sum_{i=1}^n (x_i)^2} \times \sqrt{\sum_{i=1}^n (y_i)^2}} \quad (6.2)$$

- Para el cálculo de los GADs se pueden usar nuevos operadores de agregación, tales como: media geométrica (**G**), media armónica (**H**), media cuadrática (**Q**), y la media- $\rho$  ( $M_\rho$ ) [26].

$$\mathbf{G}(x_1, \dots, x_n) = \left( \prod_{i=1}^n x_i \right)^{\frac{1}{n}}$$

$$\mathbf{H}(x_1, \dots, x_n) = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

$$\mathbf{Q}(x_1, \dots, x_n) = \left( \frac{1}{n} \sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}$$

$$M_\rho(x_1, \dots, x_n) = \left( \frac{1}{n} \sum_{i=1}^n x_i^\rho \right)^{\frac{1}{\rho}}$$

- En lugar de usar una función lineal convexa T-S, se propone utilizar los siguientes conectivos mixtos de compensación, función exponencial convexa T-S (6.3) y el operador gamma (6.4) [8]; en [33] el autor propone utilizar la función geométrica (6.5) y la función armónica (6.6).

$$E_{\gamma,T,S}(x_1, \dots, x_n) = (T(x_1, \dots, x_n))^{1-\gamma} \cdot (S(x_1, \dots, x_n))^\gamma \quad (6.3)$$

$$f_\gamma(x_1, \dots, x_n) = \left( \prod_{i=1}^n x_i \right)^{1-\gamma} \cdot \left( 1 - \prod_{i=1}^n (1 - x_i) \right)^\gamma \quad (6.4)$$

$$G_\alpha(x_1, \dots, x_n) = T(x_1, \dots, x_n)^\alpha \cdot S(x_1, \dots, x_n)^{1-\alpha} \quad (6.5)$$

$$H_\alpha(x_1, \dots, x_n) = \frac{T(x_1, \dots, x_n) \cdot S(x_1, \dots, x_n)}{\alpha \cdot S(x_1, \dots, x_n) + (1 - \alpha) \cdot T(x_1, \dots, x_n)} \quad (6.6)$$

donde  $\gamma \in [0, 1]$  y  $\alpha \in [0, 1]$ .

- Por último se propone explorar la faceta del algoritmo LAMDA como generador de libro de códigos, para determinar que tan eficiente es con respecto a otros algoritmos generadores de libros de códigos.

## 6.2. Publicaciones generadas

1. Guzmán E., **Zambrano J. G.**, Orantes A. and Pogrebnyak O. (2009). “A Theoretical Exposition of Image Vector Quantization Based on LAMDA Methodology”. *IEEE Computer Society Press, Proceedings, 52nd IEEE International Midwest Symposium on Circuits and Systems*, pp. 743-746. ISBN: 978-1-4244-4479-3. ISSN: 1548-3746.
2. Guzmán E., **Zambrano J. G.**, García I. and Pogrebnyak O. (2011). “LAMDA Methodology Applied to Image Vector Quantization”. *Advances in Intelligent and Soft Computing* (ISSN: 1867-5662), Vol. 95. *Computer Recognition systems 4*, ISBN: 978-3-642-20319-0. Springer-Verlag Berlin Heidelberg Publisher, pp. 347-356.

# Bibliografía

- [1] Acharya, T. and Tsai, P.-S. “*JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures*”, ISBN: 0-471-484222-9, Wiley-Interscience, 2004.
- [2] Aguilar-Martin, J., Sarrate, R. and Waissman, J. “*Knowledge-based Signal Analysis and Case-based Condition Monitoring of a Machine Tool*”, Joint 9th IFSA World Congress and 20th NAFIPS International Conference Proceedings, Vol. 1, 2001.
- [3] Aguilar-Martin, J., Agell, N., Sánchez, M. and Prats, F. “*Analysis of Tensions in a Population Based on the Adequacy Concept*”, 5th Catalanian Conference on Artificial Intelligence (CCIA), Vol. 2504, 2002.
- [4] Amerijckx, C., Legat, J.-D. and Verleysen, M. “*Image Compression Using Self-Organizing Maps*”, Systems Analysis Modelling Simulation, Vol. 43, No. 11, 2003.
- [5] Atine, J.-C., Doncescu, A. and Aguilar-Martin, J. “*A Fuzzy Clustering Approach for Supervision of Biological Processes by Image Processing*”, European Society for Fuzzy Logic and Technology (EUSFLAT), 2005.
- [6] Bahram, S., Azami, Z. and Feng, G. “*Robust Vector Quantizer Design Using Competitive Learning Neural Networks*”, In Proceedings of European Workshop on Emerging Techniques for Communications Terminals, 1997.
- [7] Basil, G. and Jiang, J. “*An Improvement on Competitive Learning Neural Network By LBG Vector Quantization*”, In Proc. IEEE International Conference on Multimedia Computing and Systems (ICMCS), Vol. 1, 1999.
- [8] Beliakov, G., Pradera, A. and Calvo, T. “*Aggregation Functions: A Guide for Practitioners*”, Kacprzyk, J., (eds.), ISBN: 978-3-540-73720-9, Springer-Verlag Berlin Heidelberg, Vol. 221, 2007.

- 
- [9] Bhaskaran, V. and Konstantinides, K. “*Image and Video Compression Standards: Algorithms and Architectures*”, ISBN: 0-7923-9952-8, Kluwer Academic Publishers, 1997.
- [10] Buckley, J. J. “*Simulating Fuzzy Systems*”, Kacprzyk, J., (eds.), ISBN: 3-540-24116-7, Springer-Verlag Berlin Heidelberg, Vol. 171, 2005.
- [11] Chan, M., Aguilar-Martin, J., Piera, N., Celsis, P. and Vergnes, J. “*Classification techniques for feature extraction in low resolution tomographic evolutives images: Application to cerebral blood flow estimation*”, In 12th Conf. Groupe d’Etudes du Traitement du Signal et des Images (GRETI), 1989.
- [12] Chang-Qian, C., Soo-Ngee, K. and Sivaprakasapillai, P. “*A modified generalised Lloyd algorithm for VQ codebook design*”, IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), Vol. 1, 1996.
- [13] Detyniecki, M. “*Mathematical Aggregation Operators and their Application to Video Querying*”, Ph.D. dissertation, Université Pierre et Marie Curie, Francia, 2000.
- [14] Engelbrecht, A. P. “*Computational intelligence, An introduction*”, ISBN: 978-0-470-03561-0, John Wiley and Sons Ltd, 2007.
- [15] Gersho, A., Wang, S. and Zeger, K. “*Vector Quantization Techniques in Speech Coding*”, in Advances in Speech Signal Processing, Furui, S. and Sondhi, M. M., eds., ISBN: 0-8247-8540-1, Marcel Dekker, 1991.
- [16] Gonzalez, R. C. and Woods, R. E. “*Digital image Processing*”, ISBN: 978-0201180756, Prentice Hall, 2001.
- [17] Gray, R. M. “*Vector Quantization*”, IEEE ASSP Magazine. Vol. 1, 1984.
- [18] Gray, R. M. and Neuhoff, D. L. “*Quantization*”, IEEE Transactions on Information Theory, Vol. 44, No. 6, 1998.
- [19] Guzmán, E., Pogrebnyak, O., Sánchez, L. and Yañez, C. “*A Fast Search Algorithm for Vector Quantization based on Associative Memories*”, Lecture Notes in Computer Science, Springer Berlin/Heidelberg (LNCS 5197), 2008.

- 
- [20] Hernández, H. R. “*Supervision et diagnostic des procédés de production d’eau potable*”, Ph. D. thesis, l’Institut National des Sciences Appliquées de Toulouse. France, 2006.
- [21] Horng, M.-H. and Jiang, T.-W. “*The codebook design of image vector quantization based on the firefly algorithm*”, Proceedings of the Second international conference on Computational collective intelligence (ICCCI), Vol. 3, 2010.
- [22] Isaza, C. V. “*Diagnostic par Techniques d’apprentissage Floues : Conception d’une Methode De Validation Et d’optimisation des Partitions*”, Ph. D. thesis, l’Université de Toulouse. France, 2007.
- [23] Karayiannis, N. B. and Pai, P.-I. “*Fuzzy Vector Quantization Algorithms*”, The 3rd IEEE Conference on Fuzzy Systems, 1994
- [24] Kempowsky, T. “*Surveillance de Procédes a Base de Methodes de Classification: Conception d’un Outil d’aide Pour la Detection et le Diagnostic des Defaillances*”, Ph. D. thesis, l’Institut National des Sciences Appliquées de Toulouse. France, 2004.
- [25] Khan, M. A. and Smith, M. J. T. “*Fundamentals of Vector Quantization*”, in Handbook of Image and Video Processing, Bovik, A., ed., ISBN: 0-12-119790-5, Academic Press, 2000.
- [26] Klement, E. P., Mesiar, R. and Pap, E. “*Triangular Norms*”, ISBN: 0-7923-6416-3. Kluwer Academic Publishers, 2000.
- [27] Kohonen, T. “*Automatic formation of topological maps of patterns in a self-organizing system*”, In Proc. 2SCIA, Scand. Conf. on Image Analysis, 1981.
- [28] Kohonen, T. “*Self-organizing formation of topologically correct feature maps*”, Biological Cybernetics, Vol. 43, No. 1, 1982.
- [29] Kohonen, T. “*Self-Organizing Maps*”, ISBN: 3-540-67921-9, Springer-Verlag, 2001.
- [30] Kpalma K. and Ronsin J. “*An Overview of Advances of Pattern Recognition Systems in Computer Vision*”, Vision Systems: Segmentation and Pattern Recognition, I-Tech Education and Publishing, ISBN: 978-3-902613-05-9, 2007.

- 
- [31] Linde, Y., Buzo, A. and Gray, R. “*An Algorithm for Vector Quantizer Design*”, IEEE Transactions on Communications, Vol. 28, No. 1, 1980.
- [32] Mesiar, R., Kolesárová, A., Calvo, T. and Komorníková, M. “*A Review of Aggregation Functions*”, in Fuzzy Sets and Their Extensions: Representation, Aggregation and Models, Herrera, H. B. F. and Montero, J., eds., ISBN: 978-3-540-73722-3, Springer-Verlag, 2008.
- [33] Mora, J. J. “*Localización de fallas en sistemas de distribución de energía eléctrica usando métodos basados en el modelo y métodos basados en el conocimiento*”, Ph. D. thesis, Universidad de Girona. Cataluña-España, 2006.
- [34] Nasbaradi, N. M. and King, R. A. “*Image Coding Vector Quantization: A Review*”, IEEE Transactions on communications, Vol. 36, No. 8, 1988.
- [35] Orantes, A., Kempowsky, T., Lann, M.-V. L., Prat, L., Elgue, S., Gourdon, C. and Cabassud, M. “*Selection of sensors by a new methodology coupling a classification technique and entropy criteria*”, Chemical engineering research and design Journal, 2007.
- [36] Piera, N., Desroches, P. and Aguilar-Martin, J. “*LAMDA: An Incremental Conceptual Clustering Method*”, LAAS. Laboratoire d’Automatique et d’Analyse des Systems, Report No. 89420, 1989.
- [37] Piera, N., Desroches, P. and Aguilar-Martin, J. “*Variation points in pattern recognition*”, Pattern Recognition Letters, Vol. 11, 1990.
- [38] Piera, N. and Aguilar-Martin, J. “*Controlling Selectivity in Nonstandard Pattern Recognition Algorithms*”, IEEE Transactions on Systems, Man and Cybernetics, Vol. 21, No.1, 1991.
- [39] Pu, I. M. “*Fundamental Data Compression*”, ISBN: 978-0-7506-6310-6, Butterworth-Heinemann, 2006.
- [40] Ray, A. K. and Acharya, T. “*Information Technology: Principles and Applications*”, ISBN: 81-203-2184-7, Prentice-Hall of India, 2004.

- 
- [41] Sayood, K., *“Introduction to Data Compression”*, ISBN: 978-0126208627, Morgan Kaufmann, 2005.
- [42] Shannon, C. E. *“A Mathematical Theory of Communication”*, The Bell System Technical Journal, Vol. 27, 1948.
- [43] Somasundaram, K. and M. Mary Shanthi Rani. *“Mode Based K-Means Algorithm with Residual Vector Quantization for Compressing Images”*, Control, Computation and Information Systems. Vol. 140, 2011.
- [44] Taubman, D. S., Marcellin, M. W. *“JPEG2000 Image Compression Fundamentals, Standards and Practice”*, ISBN 0-7923-7519-X, Kluwer Academic Publishers, 2002.
- [45] Trussell, H. J. and Vrhel, M. J. *“Fundamentals of Digital Imaging”*, ISBN: 978-0-521-86853-2, Cambridge University Press, 2008.
- [46] Waissman, J., Ben-Youssef, C. and Vázquez, G. *“Fuzzy Automata Identification Based on Knowledge Discovery in Datasets for Supervision of a WWT Process”*, 3rd International Conference: Sciences of Electronic Technologies of Information and Telecommunications (SETIT), 2005.
- [47] Woods, J. W. *“Multidimensional Signal, Image, and Video Processing and Coding”*, ISBN: 978-0-12-088516-9, Elsevier Inc, 2006.



# Apéndice A

## Pseudo-códigos del esquema VQ-LAMDA

**Algorithm A.0.1:** BINOMIAL CENTRO-PRODUCTO( $VQ - LAMDA$ )

```
1.- comment: Calcula el índice usando una distribución de probabilidad
2.-  $value \leftarrow 0$ 
3.-  $index \leftarrow 1$ 
4.-  $j \leftarrow 1$ 
5.- for  $l \leftarrow 1$  to  $N$ 
   do {
     6.-  $GADS \leftarrow 1$ 
     7.- for  $i \leftarrow 1$  to  $n$ 
       do {
         8.-  $X[j][i] \leftarrow X[j][i]/255$ 
         9.-  $MADbincen \leftarrow P[l][i] \wedge X[j][i] * (1 - P[l][i]) \wedge (1 - X[j][i]) /$   

            $X[j][i] \wedge X[j][i] * (1 - X[j][i]) \wedge (1 - X[j][i])$ 
         10.-  $GADS \leftarrow GADS * MADbincen$ 
       }
     11.- if  $GADS > value$ 
       then {
         12.-  $value \leftarrow GADS$ 
         13.-  $index \leftarrow l$ 
       }
   }
14.-  $indice[j] \leftarrow index$ 
```

**Algorithm A.0.2:** BINOMIAL CENTRO-PROMEDIO(VQ – LAMDA)

```

1.- comment: Calcula el índice usando una distribución de probabilidad
2.-  $value \leftarrow 0$ 
3.-  $index \leftarrow 1$ 
4.-  $j \leftarrow 1$ 
5.- for  $l \leftarrow 1$  to  $N$ 
   do {
     6.-  $GADS \leftarrow 0$ 
     7.- for  $i \leftarrow 1$  to  $n$ 
       do {
         8.-  $X[j][i] \leftarrow X[j][i]/255$ 
         9.-  $MADbincen \leftarrow P[l][i] \wedge X[j][i] * (1 - P[l][i]) \wedge (1 - X[j][i]) /$ 
            $X[j][i] \wedge X[j][i] * (1 - X[j][i]) \wedge (1 - X[j][i])$ 
         10.-  $GADS \leftarrow GADS + MADbincen/n$ 
       }
     11.- if  $GADS > value$ 
       then {
         12.-  $value \leftarrow GADS$ 
         13.-  $index \leftarrow l$ 
       }
   }
14.-  $indice[j] \leftarrow index$ 

```

**Algorithm A.0.3:** BINOMIAL CENTRO-SUMA( $VQ - LAMDA$ )

```

1.- comment: Calcula el índice usando una distribución de probabilidad
2.-  $value \leftarrow 0$ 
3.-  $index \leftarrow 1$ 
4.-  $j \leftarrow 1$ 
5.- for  $l \leftarrow 1$  to  $N$ 
   {
6.-  $GADS \leftarrow 0$ 
7.- for  $i \leftarrow 1$  to  $n$ 
   {
8.-  $X[j][i] \leftarrow X[j][i]/255$ 
9.-  $MADbincen \leftarrow P[l][i] \wedge X[j][i] * (1 - P[l][i]) \wedge (1 - X[j][i]) /$ 
   {
10.-  $GADS \leftarrow GADS + MADbincen$ 
11.- if  $GADS > value$ 
   {
12.-  $value \leftarrow GADS$ 
13.-  $index \leftarrow l$ 
   }
   }
   }
14.-  $indice[j] \leftarrow index$ 

```

