Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador (IPCV-LAB)

Algoritmo de segmentación de imágenes de intensidad basado en una umbralización por máxima verosimilitud usando distribuciones de probabilidad gamma de cuatro parámetros

Por:

Peter De Ford González

Ciudad Universitaria Rodrigo Facio

Febrero de 2014

Algoritmo de segmentación de imágenes de intensidad basado en una umbralización por máxima verosimilitud usando distribuciones de probabilidad gamma de cuatro parámetros

Por:

Peter De Ford González

Sometido a la Escuela de Ingeniería Eléctrica

de la Facultad de Ingeniería

de la Universidad de Costa Rica

como requisito parcial para optar por el grado de:

LICENCIADO EN INGENIERÍA ELÉCTRICA

Aprobado por el Tribunal:

Dr.-Ing. Geovanni Martínez Castillo
Director, Comité Asesor

M.Sc. Luis Diego Rojas Muñoz
Profesor Lector, Comité Asesor

M.Sc. Arnoldo Castro Castro
Profesor Lector, Comité Asesor

DEDICATORIA

A Dios, mi familia y amigos!

RECONOCIMIENTOS

Agradezco al Dr.-Ing. Geovanni Martínez por motivarme a realizar la tesis en el IPCV-LAB, por las largas conversaciones de donde surgieron muy buenas ideas, y por exigirme un alto grado de excelencia en todo. Además, agradezco a los profesores lectores M.Sc. Arnoldo Castro Castro y M.Sc. Luis Diego Rojas Muñoz por su interés en este proyecto. Finalmente, agradezco a las personas involucradas en la evaluación de resultados: Mami, Fico, Eliza, María del Mar, Jeilin y Danny.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS ..... **VII**

ÍNDICE DE TABLAS ..... **VIII**

ACRÓNIMOS ..... **IX**

NOMENCLATURA ..... **XI**

ABSTRACT ..... **XIV**

RESUMEN ..... **XV**

CAPÍTULO 1: INTRODUCCIÓN **1**

1.1 INTRODUCCIÓN AL TEMA 1
1.2 ESTADO DEL ARTE 4
1.3 PROPUESTA Y JUSTIFICACIÓN 6
1.4 METODOLOGÍA DEL ALGORITMO PROPUESTO 7
1.5 PROCEDIMIENTO DE EVALUACIÓN 7
1.6 OBJETIVOS 8
1.6.1 Objetivo general *8*
1.6.2 Objetivos específicos *8*

CAPÍTULO 2: DESARROLLO TEÓRICO **10**

2.1 IMÁGENES DIGITALES 10
2.1.1 Imágenes RGB y su conversión a imágenes de intensidad *10*
2.1.3 Histograma de una imagen de intensidad *11*
2.2 DISTRIBUCIONES DE PROBABILIDAD 12
2.2.1 Distribución Gaussiana de dos parámetros *12*
2.2.2 Distribución gamma de cuatro parámetros *15*

CAPÍTULO 3: ALGORITMOS DE UMBRALIZACIÓN POR MÁXIMA VEROSIMILITUD **17**

3.1 FUNCIÓN DE VEROSIMILITUD BASE 17
3.2 FUNCIONES DE VEROSIMILITUD DE LOS ALGORITMOS DE KITTLER, KURITA Y OTSU A PARTIR DE LA FUNCIÓN DE VEROSIMILITUD BASE 19
3.3 ESTIMACIÓN DE PARÁMETROS 21

CAPÍTULO 4: ALGORITMO PROPUESTO **23**

4.1 FUNCIÓN DE VEROSIMILITUD DEL ALGORITMO PROPUESTO A PARTIR DE LA FUNCIÓN DE VEROSIMILITUD BASE 23

4.2 ESTIMACIÓN DE PARÁMETROS 26

4.2.1 Método exhaustivo *26*

4.2.2 Método basado en el algoritmo Downhill Simplex *29*

4.3 FUNCIONES DE VEROSIMILITUD DE KITTLER, KURITA Y OTSU COMO CASOS ESPECIALES DE LA FUNCIÓN DE VEROSIMILITUD DEL ALGORITMO PROPUESTO 34

CAPÍTULO 5: RESULTADOS EXPERIMENTALES **36**

5.1 IMPLEMENTACIÓN 36

5.2 BIBLIOTECA DE IMÁGENES *IPCV-LAB IMAGE DATA BASE* 36

5.3 EJEMPLOS DE SEGMENTACIONES 37

5.4 EVALUACIÓN DE LOS RESULTADOS 46

5.4.1 Evaluación objetiva usando el Error Cuadrático Medio (MSE) *46*

5.4.2 Evaluación objetiva usando la métrica F1-score *48*

5.4.3 Evaluación subjetiva por parte de 8 humanos *51*

5.5 TIEMPOS DE PROCESAMIENTO 52

CAPÍTULO 6: CONCLUSIONES **53**

REFERENCIAS **55**

ARTÍCULOS Y LIBROS 55

PÁGINAS DE INTERNET 57

APÉNDICE **58**

A. ESTIMACIÓN POR MÁXIMA VEROSIMILITUD DE LOS PESOS DE LA FUNCIÓN DE VEROSIMILITUD BASE DADO EL UMBRAL 58

ÍNDICE DE FIGURAS

Figura 1: Generación de colores a partir de combinaciones de componentes rojo, verde y azul. 10
Figura 2: Imagen de intensidad con su respectivo histograma. 11
Figura 3: Distribuón Gaussiana, forma de campana 12
Figura 4: Forma de la distribución gamma para diferentes valores de $\boldsymbol{\alpha}$ y $\boldsymbol{\beta}$, con $\boldsymbol{\varphi = 0}$ y $\boldsymbol{\lambda = 1}$ 16
Figura 5: Ejemplo de un modelo basado en distribuciones gamma 24
Figura 6: Tipos de modificaciones hechos al simplex en cada iteración 30
Figura 7: Segmentación de imagen original a color de una moneda 38
Figura 8: Segmentación de imagen original a color de una fogata 39
Figura 9: Segmentación de imagen original a color de una boya 40
Figura 10: Segmentación de imagen original a color de un rio y un bosque. 41
Figura 11: Segmentación de imagen original a color de unas letras hechas con crayola 42
Figura 12: Segmentación de imagen original a color de un cuadrado cuyo histograma fue hecho sintéticamente 43
Figura 13: Segmentación de imagen original a color de un tatuaje 44
Figura 14: Segmentación de imagen original a color de un OVNI 45
Figura 15: Detección basada en la métrica F1-score aplicada a imágenes 49

ÍNDICE DE TABLAS

Tabla 1: Categorías de la *IPCV-LAB Image Data Base* 37
Tabla 2: Promedios de los MSE obtenidos para cada categoría de imágenes 47
Tabla 3: Promedios de los F1-score obtenidos para cada categoría de imágenes 50
Tabla 4: Promedios de las evaluaciones de los humanos para cada categoría de imágenes 51
Tabla 5: Tiempos de procesamiento promedio para cada categoría de imágenes 52

ACRÓNIMOS

IPCV-LAB Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador (por sus siglas en inglés: "Image Processing and Computer Vision Research Laboratory").

RGB Modelo de color utilizado en imágenes, en el que se representa cualquier color como una combinación lineal de componentes rojo, verde y azul (por sus siglas en inglés: "Red Green Blue").

JPG Formato de imágenes digitales (del inglés: "Joint Photographic Experts Group").

MLE Parámetro estimado por máxima verosimilitud (por sus siglas en inglés: "Maximum Likelihood Estimate"). Máxima verosimilitud es un método tomado de la literatura profesional que en este trabajo se usa para estimar los parámetros desconocidos de una distribución de probabilidad.

MSE Error Cuadrático Medio (por sus siglas en inglés: "Mean Squared Error"), es una métrica que cuantifica la diferencia entre los valores dados por un estimador y los valores reales de la cantidad estimada.

RAM Memoria de Acceso Aleatorio (por sus siglas en inglés: "Random Access Memory"), es una memoria física ubicada en el interior de un computador que se utiliza como memoria de trabajo para el sistema operativo y los programas.

GB Del inglés: “Giga Byte”, equivalente a 10^9 Bytes de información, donde 1 Byte es equivalente a 8 bits de información.

NOMENCLATURA

i	Intensidad del pixel.
I	Imagen de intensidad.
N	Número de píxeles en la imagen de intensidad.
th	Umbral (del inglés: “threshold”).
th_{op}	Umbral óptimo.
PDF^{I}	Función de densidad de probabilidad de los valores de intensidad de la imagen de intensidad (por sus siglas en inglés: “Probability Density Function of the intensity values of the intensity Image”).
PDF^{S}	Función de densidad de probabilidad de los valores de intensidad de la fuente de información (por sus siglas en inglés: “Probability Density Function of the information Source”).
P	Conjunto de parámetros que describen la PDF^{S}.
$\hat{P}_{th}$	Conjunto preliminar de parámetros estimados, dado el umbral th. Son un estimado preliminar de P.
$\hat{P}$	Conjunto de parámetros óptimos. Son el estimado de P.
$PDF^{S,op}$	PDF^{S} óptima, cuyos parámetros fueron obtenidos a partir de la maximización de la función de verosimilitud.
PDF_{0}^{S}	Función de densidad de probabilidad de la clase 0 de la fuente de información.

S_0	Conjunto de parámetros que describen la PDF_0^S.
$[a_0, b_0]$	Ventana cuadrada aplicada a la PDF_0^S, donde a_0 es el límite izquierdo y b_0 es el límite derecho.
w_0	Peso con el cual se sopesa la PDF_0^S.
PDF_1^S	Función de densidad de probabilidad de la clase 1 de la fuente de información.
S_1	Conjunto de parámetros que describen la PDF_1^S.
$[a_1, b_1]$	Ventana cuadrada aplicada a la PDF_1^S, donde a_1 es el límite izquierdo y b_1 es el límite derecho.
w_1	Peso con el cual se sopesa la PDF_1^S.
$u(*)$	Función escalón unitario.
$first0$	Primer valor de intensidad del histograma (de izquierda a derecha) mayor a cero.
$last0$	Último valor de intensidad del histograma (de izquierda a derecha) mayor a cero perteneciente a la clase 0 (valores de intensidad menores o iguales al umbral th).
$first1$	Primer valor de intensidad del histograma (de izquierda a derecha) mayor a cero perteneciente a la clase 1 (valores de intensidad mayores al umbral th).
$last1$	Último valor de intensidad del histograma (de izquierda a derecha) mayor a cero.
$c_{0,th}$	Área bajo la curva de la PDF^I que va de 0 a th: $\sum_{i=0}^{th} PDF^I(i)$.
$c_{1,th}$	Área bajo la curva de la PDF^I que va de $th + 1$ a 255: $\sum_{i=th+1}^{255} PDF^I(i)$.
μ_0	Media de los valores de intensidad de una PDF_0^S de tipo Gaussiana.
σ_0^2	Varianza de los valores de intensidad de una PDF_0^S de tipo Gaussiana.
μ_1	Media de los valores de intensidad de una PDF_1^S de tipo Gaussiana.

σ_1^2	Varianza de los valores de intensidad de una PDF_1^S de tipo Gaussiana.
α_0	Parámetro de forma de una PDF_0^S de tipo gamma.
β_0	Parámetro de escala de una PDF_0^S de tipo gamma.
φ_0	Parámetro de locación de una PDF_0^S de tipo gamma.
λ_0	Parámetro de signo de una PDF_0^S de tipo gamma.
α_1	Parámetro de forma de una PDF_1^S de tipo gamma.
β_1	Parámetro de escala de una PDF_1^S de tipo gamma.
φ_1	Parámetro de locación de una PDF_1^S de tipo gamma.
λ_1	Parámetro de signo de una PDF_1^S de tipo gamma.

ABSTRACT

This work was carried out in the Image Processing and Computer Vision Research Laboratory (IPCV-LAB) of the Electrical Engineering School of the University of Costa Rica as a requisite for the academic degree of licentiate in Electrical Engineering. We present a maximum likelihood thresholding algorithm whose likelihood function is derived from a new probability density function model, which consists in the sum of two weighted and windowed four-parameter gamma distributions, as an alternative to currently used models consisting of the sum of two weighted and windowed two-parameter Gaussian distributions. The ability of the new model for describing more complex shapes present in probability density functions of the intensity values of images along with the fact that the likelihood function derived from it is more general (since the Gaussian based likelihood functions are special cases of it) let the proposed algorithm to deliver better thresholds and therefore better segmentation results. This was confirmed by the performed experiments with 132 images, where the proposed algorithm resulted in average slightly better than the Gaussian based maximum likelihood thresholding algorithms found in the professional literature, and much better for segmenting low contrast images. The unique disadvantage of the new model is that the additional parameters and complexity of its likelihood function resulted in an increase of the processing time by a factor of three with respect to the Gaussian based maximum likelihood thresholding algorithms found in the professional literature. However, this disadvantage isn't very significant because the processing time of one image is in the range of milliseconds.

RESUMEN

Este trabajo fue desarrollado en el Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador (IPCV-LAB) de la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica como requisito para optar por el grado académico de licenciado en ingeniería eléctrica. En este se presenta un algoritmo de umbralización por máxima verosimilitud cuya función de verosimilitud es derivada de un nuevo modelo de distribución de probabilidad, el cual consiste en la suma de dos distribuciones gamma de cuatro parámetros sopesadas y ventaneadas como alternativa a los modelos usados actualmente, los cuales consisten en la suma de dos distribuciones Gaussianas de dos parámetros sopesadas y ventaneadas. La capacidad de este modelo de describir comportamientos estadísticos más complejos junto con el hecho de que la función de verosimilitud derivada de este es más general (dado que las funciones de verosimilitud derivadas de modelos Gaussianos son casos especiales de ella) hacen que el algoritmo propuesto provea mejores umbrales y por consiguiente mejores segmentaciones. Esto fue confirmado por los experimentos realizados en 132 imágenes, donde el algoritmo propuesto resultó ser en promedio ligeramente más preciso que los algoritmos de umbralización por máxima verosimilitud que hacen uso de modelos Gaussianos descritos en la literatura profesional, y resultó ser la mejor opción para segmentar imágenes de bajo contraste. La única desventaja del nuevo modelo propuesto es que los parámetros extra y la complejidad de su función de verosimilitud hacen que el tiempo de procesamiento sea en promedio tres veces mayor respecto a los algoritmos que hacen uso de modelos Gaussianos descritos en la literatura profesional. Sin embargo, esta desventaja no es significativa ya que el tiempo de procesamiento de una imagen está en el rango de los milisegundos.

CAPÍTULO 1: INTRODUCCIÓN

1.1 Introducción al tema

Una escena es un espacio tridimensional constituido por una fuente de iluminación y objetos cuyas superficies reflejan la luz proveniente de la fuente de iluminación. Un ejemplo de una escena podría ser una habitación en la que hay una lámpara iluminando un mueble de madera delante de una pared de mármol. Una cámara fotográfica digital puede ser usada para capturar una imagen de intensidad (fotografía) de una escena. Para ello cuenta con un pequeño orificio por el cual entran los rayos de luz reflejados por los diferentes puntos sobre la superficie de los objetos. El rayo de luz reflejado por un punto que logre entrar a través del orificio de la cámara incide sobre una cuadrícula de foto sensores. El foto sensor sobre el cual incidió convierte su intensidad en un valor analógico de voltaje, el cual mediante cuantización es convertido posteriormente en un valor discreto. Ese valor discreto es finalmente codificado mediante una palabra de código de 8 bits, donde cada código representa un valor dentro de una escala de intensidades en el rango de 0 (negro) a 255 (blanco). Esa escala de intensidades se conoce comúnmente como escala de grises. Una imagen de intensidad no es más que una cuadrícula de puntos, comúnmente denominados píxeles, con el mismo número de filas y columnas que la cuadrícula de foto sensores, donde un pixel arbitrario ubicado en cierta fila y columna (en cierta coordenada) de la imagen de intensidad contiene una copia de la palabra de código de 8 bits generada por el foto sensor en esa misma coordenada de la cuadrícula de foto sensores.

Una característica importante de una imagen de intensidad es que ésta siempre se puede describir como la unión de un conjunto de regiones de píxeles adyacentes, donde cada región está constituida por todos los pixeles cuyas coordenadas coinciden con las coordenadas de aquellos foto sensores sobre los cuales incidieron rayos de luz provenientes de un objeto en particular de la escena. Por ejemplo, en la escena de la habitación descrita anteriormente, cualquier imagen de intensidad de la escena se puede describir como la

unión de dos regiones: la región del mueble de madera (región del objeto) y la región de la pared de mármol (región del fondo).

En el área de procesamiento digital de imágenes comúnmente se necesita subdividir una imagen de intensidad (a escala de grises) que muestra un objeto sobre un fondo en las regiones de objeto y fondo. Los algoritmos que realizan este proceso son conocidos en la literatura profesional como algoritmos de segmentación de imágenes. Un algoritmo de segmentación clasifica cada pixel de la imagen de intensidad en una de dos posibles clases: la clase de los píxeles que conforman la región del fondo (clase 0) o la clase de los píxeles que conforman la región del objeto (clase 1), o viceversa. Entre los diferentes algoritmos de segmentación encontrados en la literatura profesional [10, 13, 14, 20], sobresalen los algoritmos de segmentación basados en umbralización, que realizan la clasificación comparando los valores de intensidad de los píxeles con un valor de intensidad de referencia llamado umbral de intensidad. Si el valor de intensidad del pixel es menor o igual al umbral, este es clasificado como perteneciente a la región del fondo (clase 0), de lo contrario es clasificado como perteneciente a la región del objeto (clase 1), o viceversa. El resultado del algoritmo es una imagen binaria, usualmente denominada imagen segmentada, con las mismas dimensiones que la imagen de intensidad original. Dicha imagen contiene el valor de 0 (negro) en los píxeles clasificados como pertenecientes a la región del fondo y el valor de 255 (blanco) en los píxeles clasificados como pertenecientes a la región del objeto, o viceversa.

Los algoritmos de segmentación basados en umbralización, también conocidos como algoritmos de umbralización, funcionan muy bien en imágenes de alto contraste y bien en imágenes de mediano contraste. En una imagen de alto contraste los valores de intensidad presentes en la región del fondo son diferentes a los valores de intensidad presentes en la región del objeto. Esta característica hace que el intervalo de valores de intensidad de la región del fondo no se traslape con el intervalo de valores de intensidad de la región del objeto. De aquí que la función de densidad de probabilidad de los valores de intensidad de la imagen de intensidad, denominada de aquí en adelante PDF^I, tenga la

forma de dos montañas totalmente separadas por un valle. Una montaña está conformada por las probabilidades de los valores de intensidad de los píxeles de la región del fondo y la otra por las probabilidades de los valores de intensidad de los píxeles de la región del objeto. Cualquier umbral de intensidad ubicado en el valle entre ambas montañas logra clasificar todos los píxeles correctamente, por lo que se consigue una segmentación perfecta de la imagen de intensidad. La PDF^I se puede calcular usando la definición estadística de probabilidad de von Mises [25], dividiendo para ello la frecuencia relativa de cada valor de intensidad (el número de ocurrencias de cada valor de intensidad en la imagen) entre el número total de píxeles en la imagen. La gráfica de la frecuencia relativa de los valores de intensidad de una imagen es conocida como histograma de la imagen. Por otro lado, en una imagen de mediano contraste, los valores de intensidad de algunos píxeles que conforman la región del objeto están también presentes en algunos píxeles que conforman la región del fondo. Esto hace que el intervalo de valores de intensidad de la región del fondo se traslape parcialmente con el intervalo de valores de intensidad de la región del objeto, por lo que la PDF^I de la imagen tiene la forma de dos montañas unidas por una meseta (traslapadas). Esto hace imposible que se pueda realizar una segmentación perfecta de la imagen usando un umbral de intensidad, ya que aquellos píxeles de la región del fondo y aquellos píxeles de la región del objeto que comparten un mismo valor de intensidad serán clasificados erróneamente en una misma clase sin importar a qué región pertenecen.

En el Laboratorio de Investigación en Procesamiento Digital de Imágenes y Visión por Computador (IPCV-LAB) de la Escuela de Ingeniería Eléctrica de la Universidad de Costa Rica son de interés los algoritmos de segmentación basados en una umbralización que determinan automáticamente el umbral con el cual se puede segmentar de la mejor manera posible una imagen arbitraria. Especialmente aquellos algoritmos de segmentación basados en una umbralización por maximización de una función de verosimilitud, pues han demostrado ser mucho más precisos que los demás propuestos en la literatura profesional [24]. Estos son conocidos como algoritmos de segmentación basados en una umbralización

por máxima verosimilitud. El umbral obtenido automáticamente recibe comúnmente el nombre de umbral óptimo. En este trabajo se desarrollará un nuevo algoritmo de umbralización por máxima verosimilitud más preciso que los algoritmos de este tipo propuestos hasta la fecha en la literatura profesional.

1.2 Estado del Arte

En la literatura profesional, los algoritmos de umbralización por máxima verosimilitud más reconocidos son los de Kittler [12], Kurita [14] y Otsu [20]. En estos algoritmos, el umbral óptimo th_{op} es aquel que maximiza una función de verosimilitud que consiste en el logaritmo natural de la probabilidad condicional $p(I|th)$ de la imagen I dado un umbral th, esto es $\ln\big(p(I|th_{op})\big) = max_{th}[\ln(p(I|th))]$. La probabilidad $p(I|th)$ se calcula haciendo uso de un modelo de formación de la imagen que asume que una fuente de información discreta sin memoria con un alfabeto de 256 símbolos $i : \{0,1,\dots,255\}$ generó cada valor de intensidad de la imagen I, en donde cada símbolo tiene una probabilidad de ocurrencia definida por la función de densidad de probabilidad de los valores de intensidad de la fuente, denominada $PDF^S(i;P)$, $i \in \{0,1,\dots,255\}$ o simplemente PDF^S (por sus siglas en inglés: "Probability Density Function of the information Source"). La PDF^S es una función de densidad de probabilidad compuesta por la suma de dos funciones de densidad de probabilidad sopesadas

$$PDF^S(i;P) = \{([u(a_0) - u(b_0)] * w_0 * PDF_0^S(i;S_0)\} + \{([u(a_1) - u(b_1)] * w_1 * PDF_1^S(i;S_1)\} \qquad (1.1)$$

La primera es la función de densidad de probabilidad $PDF_0^S(i;S_0)$, la cual está descrita por los parámetros S_0, sopesada por w_0, y ventaneada en el dominio $[a_0, b_0]$ por los escalones unitarios $u(a_0)$ y $u(b_0)$. La segunda es la función de densidad de probabilidad $PDF_1^S(i;S_1)$, la cual está descrita por los parámetros S_1, sopesada por w_1, y ventaneada en el dominio $[a_1, b_1]$ por los escalones unitarios $u(a_1)$ y $u(b_1)$. Usando las ventanas $[a_0 = 0, b_0 = th]$ y $[a_1 = th + 1, b_1 = 255]$ la fuente de información se restringe a

generar datos solo en el dominio [0,255]. Entonces, el conjunto de parámetros que describe la $PDF^S(i;P)$ es $P:\{th, S_0, S_1, w_0, w_1\}$, donde th se conoce como el umbral de la fuente de información. Los valores de los parámetros de este conjunto son desconocidos. Sin embargo, para cada posible umbral th, el resto de los parámetros se pueden estimar por máxima verosimilitud [14] haciendo uso de la información provista por la PDF^I derivada del histograma de la imagen. Usando los conjuntos de valores estimados de los parámetros de la PDF^S, denominados $\hat{P}_{th}$ (dependientes de cada umbral th), la probabilidad de la imagen dado cada umbral $p(I|th)$ se calcula como el producto de las probabilidades de ocurrencia de cada uno de los valores de intensidad de los N píxeles de la imagen, esto es $p(I|th) = \prod_{n=1}^{N} PDF^S(i_n, \hat{P}_{th})$, suponiendo que los valores de intensidad de los píxeles son estadísticamente independientes. Entonces, aquel umbral th que maximiza $\ln(p(I|th))$ es tomado como el umbral óptimo th_{op}. La PDF^S asociada a th_{op} se denomina $PDF^{S,op}$, y el conjunto de parámetros $\hat{P}_{th}$ que la describe es considerado el conjunto de parámetros óptimos $\hat{P}$. La $PDF^{S,op}$ es la PDF^S que con mayor probabilidad generó la imagen, la cual de acuerdo a la teoría de máxima verosimilitud, es la que mejor aproxima la PDF^I derivada del histograma de la imagen.

En la literatura profesional, Otsu [20] asume que la PDF_0^S y la PDF_1^S son funciones de densidad de probabilidad de tipo Gaussianas de dos parámetros (media y varianza), las cuales tienen pesos iguales ($w_0 = 0.5$, $w_1 = 0.5$), medias distintas y varianzas iguales. Dado que los pesos y las varianzas son iguales, el algoritmo de Otsu es capaz de hacer una segmentación precisa especialmente cuando los tamaños de ambas regiones son parecidos y cuando a su vez las varianzas de los valores de intensidad de cada región son parecidas. Kurita [14] hace que el algoritmo de Otsu sea funcional cuando las regiones tienen tamaños muy diferentes, para esto le hace una única modificación que consiste en asumir que los pesos son distintos en vez de asumirlos iguales. Kittler [12], al igual que Otsu y Kurita, asume que la PDF_0^S y la PDF_1^S son funciones de densidad de probabilidad Gaussianas de dos parámetros (media y varianza), pero a diferencia de ellos, asume que los pesos, las

medias y las varianzas son distintos. Esto hace que el algoritmo de Kittler sea capaz de hacer una segmentación precisa sin importar si las regiones tienen tamaños muy diferentes y varianzas de sus valores de intensidad muy diferentes.

En el último resumen exhaustivo de algoritmos de umbralización [24], el algoritmo de Kittler [12] fue catalogado como el número uno indiscutible (entre 40 algoritmos de umbralización de renombre que fueron probados) para dos áreas de aplicación importantes: imágenes de documentos de texto e imágenes biomédicas (imágenes de microscopías, radiografías, termografías, ultrasonidos, etc.). A pesar de la supremacía que el algoritmo de Kittler mostró sobre los demás, este no se desempeñó con excelencia en todas las imágenes en las que se probó, incluso dio malos resultados en algunas imágenes. Esto se debió a la incapacidad de la distribución Gaussiana de dos parámetros de modelar algunos comportamientos estadísticos presentes en las PDF^I de esas imágenes. De aquí que el algoritmo de Kittler puede ser mejorado si se utiliza una distribución de probabilidad de más parámetros cuya flexibilidad permita un mejor modelado de la PDF^I.

1.3 Propuesta y Justificación

Debido a que el algoritmo Kittler [12] no siempre da buenos resultados, en este trabajo se propone crear un mejor algoritmo de segmentación basado en una umbralización por máxima verosimilitud mediante el uso de una PDF^S más flexible que la usada por Kittler. La PDF^S del algoritmo propuesto está compuesta por la suma de dos funciones de densidad de probabilidad (PDF_0^S y PDF_1^S) sopesadas de tipo gamma de cuatro parámetros (forma, escala, locación y signo). Se escogió este tipo de función de densidad de probabilidad (o distribución de probabilidad) por dos razones: 1. a diferencia de la distribución Gaussiana, la distribución gamma de cuatro parámetros es asimétrica y tiene más parámetros, lo que hace que la PDF^S propuesta sea más flexible para modelar de buena manera cualquier PDF^I derivada del histograma de una imagen, y 2. con esta distribución se logra que los algoritmos de Kittler, Kurita y Otsu sean casos especiales del

algoritmo propuesto [3]. Dada la flexibilidad de la PDF^S compuesta por distribuciones gamma de cuatro parámetros, se espera que el algoritmo propuesto sea más preciso que el de Kittler.

1.4 Metodología del algoritmo propuesto

El algoritmo propuesto funcionará de la siguiente manera. Tomará como entrada la imagen de intensidad que se desea segmentar. Calculará el histograma de esta imagen. Obtendrá la PDF^I de la imagen dividiendo el histograma entre el número total de píxeles de la imagen. Para varios umbrales de la fuente seleccionados, estimará para cada uno los valores de los parámetros de las dos distribuciones gamma (PDF_0^S y PDF_1^S, cada una con parámetros: forma, escala, locación y signo) sopesadas haciendo uso de la información provista por la PDF^I. Usará los valores estimados de estos parámetros para calcular los logaritmos naturales de las probabilidades de la imagen dados cada uno de los umbrales de la fuente seleccionados. Tomará como umbral óptimo el umbral cuyo logaritmo natural de la probabilidad asociado sea mayor. Segmentará la imagen de intensidad usando el umbral óptimo. Finalmente, dará como salida la imagen segmentada.

1.5 Procedimiento de evaluación

Para la evaluación de la precisión del algoritmo propuesto, se elaborará una biblioteca con más de 100 imágenes de prueba distribuidas en siete categorías: sintéticas (histogramas generados usando distribuciones Gaussianas y gamma), generadas por computador (gráficos generados con programas de animación digital), de imagenología médica, letras, fotografías, objetos pequeños y bajo contraste. Estas categorías son muy variadas, y cubren una amplia gama de aplicaciones en ingeniería. La biblioteca contará con las segmentaciones manuales (hechas por un humano) de las imágenes, las cuales servirán como referencia al evaluar la precisión de las segmentaciones obtenidas con los

algoritmos ya que se considerará que el humano que las segmentó no cometió errores durante la segmentación. A cada una de las imágenes de prueba se le aplicará el algoritmo propuesto en esta tesis, así como los algoritmos de Kittler, Kurita y Otsu. La precisión de cada algoritmo se evaluará objetivamente de dos maneras: 1. calculando el Error Cuadrático Medio entre la PDF^I y la PDF^S obtenida por cada algoritmo; y 2. usando la métrica llamada F1-score [6], la cual hace uso de las segmentaciones manuales. Adicionalmente, ocho humanos evaluarán de manera subjetiva e independiente la calidad de las segmentaciones, esto con el fin de corroborar los resultados obtenidos con el F1-score y medir cómo los humanos perciben las segmentaciones hechas por los algoritmos. A partir de las evaluaciones de los resultados experimentales, se determinarán las ventajas y desventajas del algoritmo propuesto.

1.6 Objetivos

1.6.1 Objetivo general

Diseñar, programar y probar un algoritmo de umbralización por máxima verosimilitud para segmentar imágenes de intensidad de manera más precisa que los algoritmos de umbralización por máxima verosimilitud descritos en la literatura profesional.

1.6.2 Objetivos específicos

- Realizar una investigación introductoria al tema.
- Realizar un estudio de estado del arte de los algoritmos de umbralización por máxima verosimilitud.
- Proponer un nuevo algoritmo de umbralización por máxima verosimilitud.
- Programar el algoritmo propuesto en el lenguaje de programación C, así como los mejores algoritmos de umbralización descritos en la literatura profesional.

- Elaborar una biblioteca de imágenes que contenga más de 100 imágenes de diversos tipos con sus respectivas segmentaciones manuales.
- Usar los algoritmos de umbralización programados para segmentar todas las imágenes de la biblioteca.
- Evaluar la precisión y calidad de las segmentaciones hechas.
- Determinar ventajas, desventajas y posibles usos del algoritmo de umbralización propuesto.
- Hacer un informe escrito.

CAPÍTULO 2: DESARROLLO TEÓRICO

2.1 Imágenes digitales

2.1.1 Imágenes RGB y su conversión a imágenes de intensidad

Una imagen RGB está constituida por una cuadrícula de píxeles, donde cada píxel almacena la información del color mediante una combinación de componentes rojo, verde y azul. La componente de cada color se almacena en un byte (8 bits) de información, por lo que cada pixel almacena tres bytes (24 bits) de información. En la Figura 1 se ilustra cómo a partir de combinaciones de componentes rojo, verde y azul se pueden hacer todos los colores.

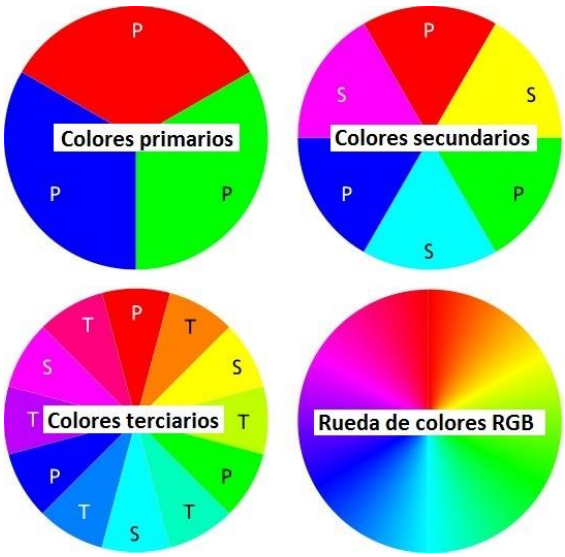

Figura 1: Generación de colores a partir de combinaciones de componentes rojo, verde y azul [29].

Una imagen RGB se puede convertir en una imagen de intensidad. Para esto, se convierte del formato RGB al formato YUV mediante la siguiente transformación matemática

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} +0.299 & +0.587 & +0.114 \\ -0.147 & -0.289 & +0.436 \\ +0.615 & -0.515 & -0.100 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.1}$$

en donde el formato YUV almacena la luminancia “Y” (intensidad luminosa) y las crominancias “U” y “V” (información de color) de cada pixel. Después, dado que la componente de luminancia “Y” es equivalente al valor de intensidad de gris, se toman las componentes de luminancia para a partir de ellas construir la imagen de intensidad.

2.1.3 Histograma de una imagen de intensidad

En el área de procesamiento de imágenes es de gran utilidad calcular la frecuencia relativa de los valores de intensidad de una imagen. La gráfica que muestra la frecuencia relativa de los valores de intensidad de 0 a 255 es conocida como histograma de la imagen, denominada $h(i)$. En la Figura 2, se muestra a la izquierda una imagen de intensidad y a la derecha su respectivo histograma. En el histograma, la montaña izquierda representa el bosque (parte oscura de la imagen), mientras que la montaña derecha del histograma representa el cielo (parte clara de la imagen).

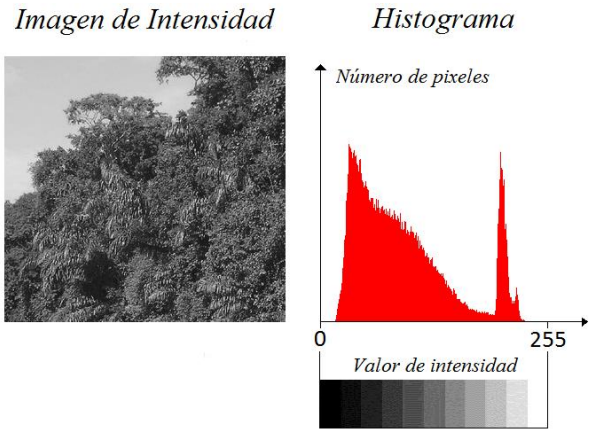

Figura 2: Imagen de intensidad con su respectivo histograma.

Usando la definición estadística de probabilidad dada por von Mises [25], se puede dividir el histograma entre el número total de píxeles de la imagen para obtener la función de densidad de probabilidad de los valores de intensidad de la imagen de intensidad, denominada PDF^I. Esta función de densidad de probabilidad es muy útil a la hora de estudiar las propiedades estadísticas de la imagen de intensidad.

2.2 Distribuciones de probabilidad

En esta sección se introducen dos distribuciones de probabilidad ampliamente usadas en este trabajo: la distribución Gaussiana de dos parámetros y la distribución gamma de cuatro parámetros. La distribución Gaussiana es usada por los algoritmos de umbralización que compiten contra el algoritmo propuesto en este trabajo, mientras que la distribución gamma es usada por el algoritmo propuesto en este trabajo.

2.2.1 Distribución Gaussiana de dos parámetros

La distribución continua de probabilidad más importante en todo el campo de la estadística es la distribución Gaussiana de dos parámetros, conocida también como distribución normal. Esta sirve para describir muchos fenómenos meteorológicos, biológicos, económicos, etc., además de errores en mediciones científicas. La distribución Gaussiana de dos parámetros se describe matemáticamente de la siguiente manera

$$PDF(i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-(i-\mu)^2}{2\sigma^2}\right)} \tag{2.2}$$

donde μ es la media y σ^2 la varianza de la distribución; y en donde i es la variable independiente y $PDF(i)$ la función de densidad de probabilidad que depende de i. En la Figura 3 se muestra la forma de esta distribución, la cual es similar a la forma de una campana.

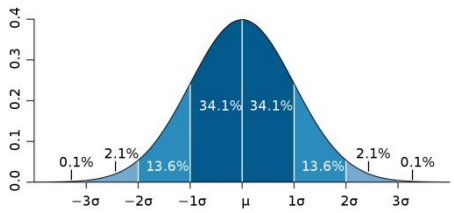

Figura 3: Distribución Gaussiana, forma de campana [30].

2.2.1.1 Distribución Gaussiana: estimación de parámetros por máxima verosimilitud

Se tienen N muestras estadísticamente independientes de una variable aleatoria distribuida Gaussianamente. Se va a demostrar que el primer momento respecto al origen (media) μ y el segundo momento respecto a la media (varianza) σ^2 de estas muestras son los parámetros que el método de máxima verosimilitud estima como los parámetros de la distribución Gaussiana que mejor corresponden a las muestras.

Si i_n es el valor de la muestra número n, para $n \epsilon [0, N-1]$, entonces su probabilidad de ocurrencia es

$$p(i_n) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-(i_n-\mu)^2}{2\sigma^2}\right)} \tag{2.3}$$

La probabilidad de ocurrencia de las N muestras es

$$p(i_0, i_1, \dots, i_{N-1}) = \prod_{n=0}^{N-1} p(i_n) = \prod_{n=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-(i_n-\mu)^2}{2\sigma^2}\right)} \tag{2.4}$$

Sacando el término constante e introduciendo la productoria en el exponente en forma de sumatoria

$$p(i_0, i_1, \dots, i_{N-1}) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} e^{\left(-\frac{\sum_{n=0}^{N-1}(i_n-\mu)^2}{2\sigma^2}\right)} \tag{2.5}$$

La media y varianza que maximizan esta probabilidad son los parámetros de máxima verosimilitud. Para simplificar esta maximización, se aplica el logaritmo natural a ambos lados de la ecuación

$$\ln\big(p(i_0, i_1, \dots, i_{N-1})\big) = \ln\left[\left(\frac{1}{2\pi\sigma^2}\right)^{\frac{N}{2}} e^{\left(-\frac{\sum_{n=0}^{N-1}(i_n-\mu)^2}{2\sigma^2}\right)}\right] \tag{2.6}$$

Aplicando el logaritmo natural a ambos términos del paréntesis

$$\ln\big(p(i_0, i_1, \dots, i_{N-1})\big) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{\sum_{n=0}^{N-1}(i_n-\mu)^2}{2\sigma^2} \tag{2.7}$$

Para encontrar el máximo, esta función se deriva parcialmente respecto a μ y a σ. Insertando la derivada parcial respecto a σ e igualándola a 0

$$0 = \frac{\partial}{\partial \sigma}\left[-\frac{N}{2}\ln(2\pi\sigma^2) - \frac{\sum_{n=0}^{N-1}(i_n - \mu)^2}{2\sigma^2}\right] \tag{2.8}$$

Aplicando la derivada a ambos términos

$$0 = -\frac{N}{\sigma} + \frac{\sum_{n=0}^{N-1}(i_n - \mu)^2}{\sigma^3} \tag{2.9}$$

Multiplicando la ecuación por σ^3 y simplificando

$$\sigma^2 = \frac{1}{N}\sum_{n=0}^{N-1}(i_n - \mu)^2 \tag{2.10}$$

Ahora se desea hacer la derivada de la Ecuación (2.7) respecto a μ, pero nótese que es una derivada difícil de hacer. Se introduce entonces el promedio de la muestra $\bar{m}$, obtenido de calcular el promedio aritmético de las N muestras

$$\bar{m} = \frac{1}{N}\sum_{n=0}^{N-1} i_n \tag{2.11}$$

Entonces, se puede expandir el último término de la Ecuación (2.7)

$$\sum_{n=0}^{N-1}(i_n - \mu)^2 = \sum_{n=0}^{N-1}\left({i_n}^2 - 2\mu i_n + \mu^2\right) = \sum_{n=0}^{N-1}\left({i_n}^2\right) - 2\mu\sum_{n=0}^{N-1}(i_n) + N\mu^2$$

$$= \sum_{n=0}^{N-1}\left({i_n}^2\right) - 2N\bar{m}\mu + N\mu^2 \tag{2.12}$$

Ahora se tiene una versión de la Ecuación (2.7) que es más fácil de derivar respecto a μ

$$\ln\big(p(i_0, i_1, \dots, i_{N-1})\big) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{\sum_{n=0}^{N-1}\left({i_n}^2\right) - 2N\bar{m}\mu + N\mu^2}{2\sigma^2} \tag{2.13}$$

Insertando la derivada parcial respecto a μ e igualando a 0

$$0 = \frac{\partial}{\partial \mu}\left[-\frac{N}{2}\ln(2\pi\sigma^2) - \frac{\sum_{n=0}^{N-1}\left({i_n}^2\right) - 2N\bar{m}\mu + N\mu^2}{2\sigma^2}\right] \tag{2.14}$$

Derivando solo el término no constante

$$0 = \frac{2N\bar{m} - 2N\mu}{2\sigma^2} \tag{2.15}$$

Simplificando

$$0 = \overline{m} - \mu \tag{2.16}$$

Reacomodando y sustituyendo $\overline{m}$

$$\mu = \overline{m} = \frac{1}{N}\sum_{n=0}^{N-1} i_n \tag{2.17}$$

2.2.2 Distribución gamma de cuatro parámetros

A pesar de que la distribución Gaussiana se utiliza para resolver gran parte de los problemas en ingeniería y ciencia que requieren distribuciones de probabilidad, también hay numerosas situaciones que requieren otros tipos de distribuciones, entre ellas la distribución gamma. Esta distribución juega un papel importante en teoría de colas, problemas de confiabilidad, modelado en imágenes de radar y ultrasonido [1, 5], entre otros. Hay varias versiones de la distribución gamma, en este trabajo se usa una de cuatro parámetros que se describe matemáticamente de la siguiente manera

$$PDF(i) = \frac{1}{\beta^{\alpha} * \Gamma(\alpha)}\left(\lambda(i-\varphi)\right)^{\alpha-1} e^{\frac{-\lambda(i-\varphi)}{\beta}} \tag{2.19}$$

donde el parámetro α (real positivo) describe la forma, β (real positivo) describe la escala, φ (real positivo o negativo) describe la locación, y λ (1 o -1) describe el signo, el cual indica si la distribución no está reflejada o sí está reflejada sobre el eje ***y***, respectivamente. El nombre gamma proviene de que la distribución contiene dentro de sí la función gamma

$$\Gamma(\alpha) = \int_0^{\infty} x^{\propto-1} e^{-x} dx\,, \;\; \alpha > 0 \tag{2.20}$$

En la Figura 4 se muestra la forma de esta distribución para diferentes valores de α y β, con $\varphi = 0$ y $\lambda = 1$.

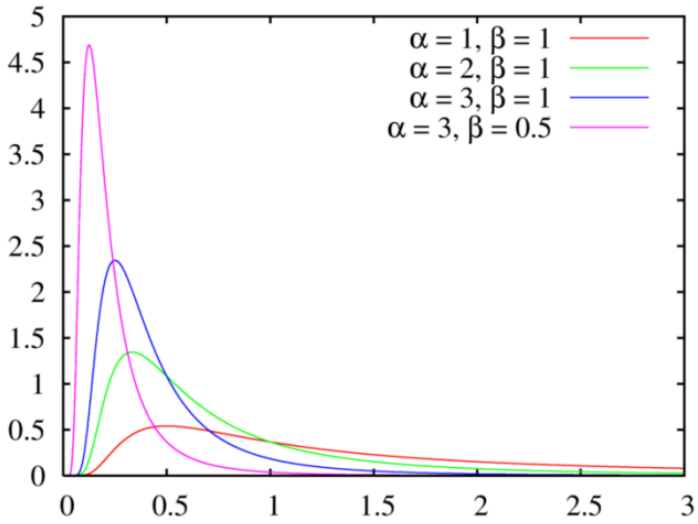

Figura 4: Forma de la distribución gamma para diferentes valores de $\boldsymbol{\alpha}$ y $\boldsymbol{\beta}$, con $\boldsymbol{\varphi} = \mathbf{0}$ y $\boldsymbol{\lambda} = \mathbf{1}$ [31].

CAPÍTULO 3: ALGORITMOS DE UMBRALIZACIÓN POR MÁXIMA VEROSIMILITUD

En este capítulo se describen los algoritmos de umbralización por máxima verosimilitud de Kittler, Kurita y Otsu. Para ello, se presenta la derivación matemática de las funciones de verosimilitud usadas por estos algoritmos para obtener el umbral óptimo, las cuales son obtenidas a partir de una función de verosimilitud base. Además, se detalla la manera en que estos algoritmos estiman los parámetros óptimos $\hat{P}$.

3.1 Función de verosimilitud base

Las funciones de verosimilitud de los algoritmos de Kittler, Kurita y Otsu, así como del algoritmo propuesto, son derivadas de una función de verosimilitud base, la cual es una versión unificada de dos funciones de verosimilitud base explicadas vagamente en [14]. La función de verosimilitud base es un resultado muy importante obtenido en esta investigación, ya que unifica todos los algoritmos de segmentación basados en una umbralización por máxima verosimilitud descritos aquí. Dada la distribución de probabilidad de la fuente de información propuesta en este trabajo (Ecuación (1.1))

$$PDF^S(i;P) = [u(a_0) - u(b_0)] * w_0 * PDF_0^S(i;S_0) + [u(a_1) - u(b_1)] * w_1 * PDF_1^S(i;S_1) \tag{3.1}$$

y la probabilidad de ocurrencia de la imagen de intensidad I

$$p(I|P) = \prod_{n=0}^{N-1} [PDF^S(i_n;P)] \tag{3.2}$$

se obtiene la función de verosimilitud base de la manera explicada a continuación. Usando el histograma $h(i)$, la Ecuación (3.2) es equivalente a

$$p(I|P) = \prod_{i=0}^{255} [PDF^S(i;P)]^{h(i)} \tag{3.3}$$

Insertando la Ecuación (3.1) en ella

$$p(I|P) = \prod_{i=0}^{255} \left[[u(a_0) - u(b_0)] * w_0 * PDF_0^S(i; S_0) + [u(a_1) - u(b_1)] * w_1 * PDF_1^S(i; S_1)\right]^{h(i)} \quad (3.4)$$

Escogiendo por conveniencia: $a_0 = 0$, $b_0 = th$, $a_1 = th + 1$, y $b_1 = 255$

$$p(I|P) = \prod_{i=0}^{th} [w_0 * PDF_0^S(i; S_0)]^{h(i)} * \prod_{i=th+1}^{255} [w_1 * PDF_1^S(i; S_1)]^{h(i)} \quad (3.5)$$

Dado que a partir de th el resto de parámetros P pueden ser obtenidos directamente, se intercambia a continuación $p(I|P)$ por $p(I|th)$ para resaltar la dependencia de I en th. Además, debido a que el computador no puede manejar los números tan pequeños resultantes de la productoria de N probabilidades, se introduce el logaritmo natural a ambos lados de la ecuación

$$\ln\big(p(I|th)\big) = \ln\left(\prod_{i=0}^{th} [w_0 * PDF_0^S(i; S_0)]^{h(i)} * \prod_{i=th+1}^{255} [w_1 * PDF_1^S(i; S_1)]^{h(i)}\right) \quad (3.6)$$

Aplicando el logaritmo natural a ambos términos del lado derecho de la ecuación

$$\ln\big(p(I|th)\big) = \ln\left(\prod_{i=0}^{th} [w_0 * PDF_0^S(i; S_0)]^{h(i)}\right) + \ln\left(\prod_{i=th+1}^{255} [w_1 * PDF_1^S(i; S_1)]^{h(i)}\right) \quad (3.7)$$

Insertando los logaritmos naturales en las productorias (por lo que estas se convierten en sumatorias)

$$\ln\big(p(I|th)\big) = \sum_{i=0}^{th} \ln\big([w_0 * PDF_0^S(i; S_0)]^{h(i)}\big) + \sum_{i=th+1}^{255} \ln\big([w_1 * PDF_1^S(i)]^{h(i)}\big) \quad (3.8)$$

Bajando los exponentes $h(i)$

$$\ln\big(p(I|th)\big) = \sum_{i=0}^{th} h(i) \ln\big(w_0 * PDF_0^S(i; S_0)\big) + \sum_{i=th+1}^{255} h(i) \ln\big(w_1 * PDF_1^S(i; S_1)\big) \quad (3.9)$$

se obtiene finalmente que la Ecuación (3.9) es la función de verosimilitud base. Se puede también obtener una versión extendida de esta función de la siguiente manera. Aplicando el logaritmo natural a los términos dentro de los paréntesis

$$\ln\big(p(I|th)\big) = \sum_{i=0}^{th} h(i)\big(\ln(w_0) + \ln\big(PDF_0^S(i;S_0)\big)\big) + \sum_{i=th+1}^{255} h(i)\big(\ln(w_1) + \ln\big(PDF_1^S(i;S_1)\big)\big) \quad (3.10)$$

Aplicando propiedad distributiva y sacando términos constantes

$$\ln\big(p(I|th)\big) = \ln(w_0)\sum_{i=0}^{th} h(i) + \sum_{i=0}^{th} h(i)\ln\big(PDF_0^S(i;S_0)\big) + \ln(w_1)\sum_{i=th+1}^{255} h(i) + \sum_{i=th+1}^{255} h(i)\ln\big(PDF_1^S(i;S_1)\big) \quad (3.11)$$

Introduciendo: $c_{0,th} = \sum_{i=0}^{th} h(i)/N$ y $c_{1,th} = \sum_{i=th+1}^{255} h(i)/N$, la versión extendida de la función de verosimilitud base es

$$\ln\big(p(I|th)\big) = c_{0,th} * N * \ln(w_0) + c_{1,th} * N * \ln(w_1) + \sum_{i=0}^{th} h(i)\ln\big(PDF_0^S(i;S_0)\big) + \sum_{i=th+1}^{255} h(i)\ln\big(PDF_1^S(i;S_1)\big) \quad (3.12)$$

3.2 Funciones de verosimilitud de los algoritmos de Kittler, Kurita y Otsu a partir de la función de verosimilitud base

Kittler, Kurita y Otsu asumen distribuciones de probabilidad Gaussianas (ver sección 2.2.1)

$$PDF_0^S(i) = \frac{1}{\sqrt{2\pi\sigma_0^2}} e^{\frac{-(i-\mu_0)^2}{2\sigma_0^2}}, para\ i \in \{0,1,\dots,th\}, \quad (3.13)$$

$$PDF_1^S(i) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{\frac{-(i-\mu_1)^2}{2\sigma_1^2}}, para\ i \in \{th+1,\dots,254,255\}, \quad (3.14)$$

Si estas se insertan en la función de verosimilitud extendida descrita por la Ecuación (3.12) se obtiene

$$\ln\big(p(I|th)\big) = \big\{c_{0,th} * N * \ln(w_0) + c_{1,th} * N * \ln(w_1)\big\} + \left\{\sum_{i=0}^{th} h(i)\ln\left(\frac{1}{\sqrt{2\pi\sigma_0^2}} e^{\frac{-(i-\mu_0)^2}{2\sigma_0^2}}\right)\right\}$$
$$+ \left\{\sum_{i=th+1}^{255} h(i)\ln\left(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{\frac{-(i-\mu_1)^2}{2\sigma_1^2}}\right)\right\} \quad (3.15)$$

El lado derecho de esta ecuación tiene tres términos. El primero es simplificado de la siguiente manera

$$c_{0,th} * N * \ln(w_0) + c_{1,th} * N * \ln(w_1) = N \sum_{j=0}^{1} c_{j,th} * \ln(w_j) \tag{3.16}$$

El segundo y tercero

$$\sum_{i=0}^{th} h(i) \ln\left(\frac{1}{\sqrt{2\pi\sigma_0^2}} e^{\frac{-(i-\mu_0)^2}{2\sigma_0^2}}\right) + \sum_{i=th+1}^{255} h(i) \ln\left(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{\frac{-(i-\mu_1)^2}{2\sigma_1^2}}\right) \tag{3.17}$$

son simplificados de la siguiente manera. Aplicando el logaritmo natural a los términos de los paréntesis

$$= \sum_{i=0}^{th} h(i) \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right) + \ln\left(\frac{1}{\sqrt{\sigma_0^2}}\right) + \frac{-(i-\mu_0)^2}{2\sigma_0^2}\right) + \sum_{i=th+1}^{255} h(i) \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right) + \ln\left(\frac{1}{\sqrt{\sigma_1^2}}\right) + \frac{-(i-\mu_1)^2}{2\sigma_1^2}\right) \tag{3.18}$$

Separando los términos de los paréntesis

$$= \left(\sum_{i=0}^{th} h(i) \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right)\right)\right) + \left(\sum_{i=0}^{th} h(i) \left(\ln\left(\frac{1}{\sqrt{\sigma_0^2}}\right)\right)\right) + \left(\sum_{i=0}^{th} h(i) \left(\frac{-(i-\mu_0)^2}{2\sigma_0^2}\right)\right) + \left(\sum_{i=th+1}^{255} h(i) \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right)\right)\right)$$
$$+ \left(\sum_{i=th+1}^{255} h(i) \left(\ln\left(\frac{1}{\sqrt{\sigma_1^2}}\right)\right)\right) + \left(\sum_{i=th+1}^{255} h(i) \left(\frac{-(i-\mu_1)^2}{2\sigma_1^2}\right)\right) \tag{3.19}$$

Sacando los términos constantes de las sumatorias

$$= \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right) * \sum_{i=0}^{th} h(i)\right) + \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right) * \sum_{i=th+1}^{255} h(i)\right) + \left(\ln\left(\frac{1}{\sqrt{\sigma_0^2}}\right) * \sum_{i=0}^{th} h(i)\right) + \left(\ln\left(\frac{1}{\sqrt{\sigma_1^2}}\right) * \sum_{i=th+1}^{255} h(i)\right)$$
$$+ \left(\sum_{i=0}^{th} h(i) \left(\frac{-(i-\mu_0)^2}{2\sigma_0^2}\right)\right) + \left(\sum_{i=th+1}^{255} h(i) \left(\frac{-(i-\mu_1)^2}{2\sigma_1^2}\right)\right) \tag{3.20}$$

Introduciendo $c_{0,th} = \sum_{i=0}^{th} h(i)/N$ y $c_{1,th} = \sum_{i=th+1}^{255} h(i)/N$, y sacando más términos constantes de las sumatorias

$$= \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right) * N * c_{0,th}\right) + \left(\ln\left(\frac{1}{\sqrt{2\pi}}\right) * N * c_{1,th}\right) + \left(\ln\left(\frac{1}{\sqrt{\sigma_0^2}}\right) * N * c_{0,th}\right) + \left(\ln\left(\frac{1}{\sqrt{\sigma_1^2}}\right) * N * c_{1,th}\right) + \left(-\frac{1}{2\sigma_0^2} \sum_{i=0}^{th} h(i)((i-\mu_0)^2)\right)$$
$$+ \left(-\frac{1}{2\sigma_1^2} \sum_{i=th+1}^{255} h(i)((i-\mu_1)^2)\right) \tag{3.21}$$

Simplificando

$$= \left(-\frac{N * c_{0,th}}{2} \ln(2\pi)\right) + \left(-\frac{N * c_{1,th}}{2} \ln(2\pi)\right) + \left(-\frac{N * c_{0,th}}{2} \ln(\sigma_0^2)\right) + \left(-\frac{N * c_{1,th}}{2} \ln(\sigma_1^2)\right) + \left(-\frac{1}{2\sigma_0^2} \sum_{i=0}^{th} h(i)((i-\mu_0)^2)\right)$$
$$+ \left(-\frac{1}{2\sigma_1^2} \sum_{i=th+1}^{255} h(i)((i-\mu_1)^2)\right) \tag{3.22}$$

Uniendo el primer término con el segundo, y el tercero con el cuarto (dado que $c_{0,th} + c_{1,th} = 1$)

$$= -\frac{N}{2}\ln[2\pi] - \frac{N}{2}\sum_{j=0}^{1} c_{j,th}\ln\left[\sigma_j^2\right] - \frac{1}{2\sigma_0^2}\sum_{i=0}^{th} h(i)(i-\mu_0)^2 - \frac{1}{2\sigma_1^2}\sum_{i=th+1}^{255} h(i)(i-\mu_1)^2 \tag{3.23}$$

Sustituyendo: $\sigma_0^2 = \frac{1}{c_{0,th}} * \sum_{i=0}^{th}\frac{h(i)(i-\mu_0)^2}{N}$ y $\sigma_1^2 = \frac{1}{c_{1,th}} * \sum_{i=th+1}^{255}\frac{h(i)(i-\mu_1)^2}{N}$ (ver ecuaciones equivalentes en Ecuación (3.31)

$$= -\frac{N}{2}\ln[2\pi] - \frac{N}{2}\sum_{j=0}^{1} c_{j,th}\ln\left[\sigma_j^2\right] - \frac{\sum_{i=0}^{th} h(i)(i-\mu_0)^2}{2 * \frac{1}{c_{0,th}} * \sum_{i=0}^{th}\frac{h(i)(i-\mu_0)^2}{N}} + -\frac{\sum_{i=th+1}^{255} h(i)(i-\mu_1)^2}{2 * \frac{1}{c_{1,th}} * \sum_{i=th+1}^{255}\frac{h(i)(i-\mu_1)^2}{N}} \tag{3.24}$$

Cancelando términos

$$= -\frac{N}{2}\ln[2\pi] - \frac{N}{2}\sum_{j=1}^{2} c_{j,th}\ln\left[\sigma_j^2\right] - \frac{c_{0,th} * N}{2} - \frac{c_{1,th} * N}{2} \tag{3.25}$$

Uniendo el tercer término con el cuarto

$$= -\frac{N}{2}\ln[2\pi] - \frac{N}{2}\sum_{j=1}^{2} c_{j,th}\ln\left[\sigma_j^2\right] - \frac{N}{2} \tag{3.26}$$

Así, la Ecuación (3.15) se convierte en

$$\ln\big(p(I|th)\big) = N\sum_{j=0}^{1} c_{j,th} * \ln\big(w_j\big) - \frac{N}{2} * \ln[2\pi] - \frac{N}{2}\sum_{j=1}^{2} c_{j,th}\ln\left[\sigma_j^2\right] - \frac{N}{2} \tag{3.27}$$

La ecuación (3.27) es la función de verosimilitud de Kittler. Si a partir de esta se hace $\sigma_0^2 = \sigma_1^2 = \sigma_w^2$, entonces la función de verosimilitud de Kurita es obtenida

$$\ln\big(p(I|th)\big) = N\sum_{j=0}^{1} c_{j,th} * \ln\big(w_j\big) - \frac{N}{2} * \ln[2\pi] - \frac{N}{2}\ln[\sigma_W^2] - \frac{N}{2} \tag{3.28}$$

Finalmente, si a partir de la función de verosimilitud de Kurita se hace $w_0 = w_1 = 0.5$, entonces la función de verosimilitud de Otsu es obtenida

$$\ln\big(p(I|th)\big) = N * \ln(0.5) - \frac{N}{2} * \ln[2\pi] - \frac{N}{2}\ln[\sigma_W^2] - \frac{N}{2} \tag{3.29}$$

3.3 Estimación de parámetros

Los algoritmos de Kittler, Kurita y Otsu obtienen para cada posible umbral $th \in \{0,255\}$ un conjunto preliminar de parámetros estimados

$$\hat{P}_{th}: \{th, S_{0,th}: \{\mu_{0,th}, \sigma_{0,th}^2\}, S_{1,th}: \{\mu_{1,th}, \sigma_{1,th}^2\}, w_{0,th}, w_{1,th}\}$$

De los conjuntos de parámetros estimados $\{\hat{P}_0, \hat{P}_1, \dots, \hat{P}_{th}, \dots, \hat{P}_{255}\}$, el que maximiza la función de verosimilitud es considerado $\hat{P}$. Para la obtención de cada $\hat{P}_{th}$, estos tres algoritmos encuentran los parámetros de la manera descrita a continuación.

Primero, en cuanto a la estimación de los pesos, Kittler y Kurita encuentran los MLEs (parámetros estimados por máxima verosimilitud, ver Apéndice A), que en este caso son: $w_{0,th} = c_{0,th}$ y $w_{1,th} = c_{1,th}$, donde $c_{0,th} = \sum_{i=0}^{th} PDF^I(i)$ y $c_{1,th} = \sum_{i=th+1}^{255} PDF^I(i)$; mientras que Otsu siempre asume que $w_{0,th} = 0.5$ y $w_{1,th} = 0.5$. Después, en cuanto a la estimación de las medias, los tres algoritmos estiman los MLEs (basado en 2.2.1.1)

$$\mu_{0,th} = \frac{1}{c_{0,th}} \sum_{i=0}^{th} i * PDF^I(i), \quad \mu_{1,th} = \frac{1}{c_{1,th}} \sum_{i=th+1}^{255} i * PDF^I(i) \tag{3.30}$$

Finalmente, en cuanto a la estimación de las varianzas, el algoritmo de Kittler obtiene los MLEs (basado en 2.2.1.1)

$$\sigma_{0,th}^2 = \frac{1}{c_{0,th}} \sum_{i=0}^{th} (i - \mu_{0,th})^2 * PDF^I(i), \quad \sigma_{1,th}^2 = \frac{1}{c_{1,th}} \sum_{i=th+1}^{255} (i - \mu_{1,th})^2 PDF^I(i) \tag{3.31}$$

mientras que Kurita y Otsu toman $\sigma_{0,th}^2$ y $\sigma_{1,th}^2$ como iguales al promedio de los MLEs de las dos varianzas que Kittler estima, denominado $\sigma_{W,th}^2$

$$\sigma_{0,th}^2 = \sigma_{1,th}^2 = \sigma_{W,th}^2 = \sum_{i=0}^{th} (i - \mu_{0,th})^2 * PDF^I(i) + \sum_{i=th+1}^{255} (i - \mu_{1,th})^2 * PDF^I(i) \tag{3.32}$$

CAPÍTULO 4: ALGORITMO PROPUESTO

En este capítulo se presenta la derivación matemática de la función de verosimilitud usada por el algoritmo propuesto en este trabajo, la cual es obtenida a partir de la función de verosimilitud base. Después, se detallan dos métodos usados por el algoritmo propuesto para estimar los parámetros óptimos $\hat{P}$ que maximizan la function de verosimilitud. Finalmente, se presentan las funciones de verosimilitud de Kittler, Kurita y Otsu como casos especiales de la función de verosimilitud del algoritmo propuesto.

4.1 Función de verosimilitud del algoritmo propuesto a partir de la función de verosimilitud base

En esta investigación se propone usar las siguientes distribuciones gamma de cuatro parámetros (ver sección 2.2.2)

$$PDF_0^S(i) = \frac{1}{{\beta_0}^{\alpha_0} * \Gamma(\alpha_0)} \left(\lambda_0(i-\varphi_0)\right)^{\alpha_0 - 1} e^{\frac{-\lambda_0(i-\varphi_0)}{\beta_0}} \quad , para\ i \in \{0,1,\dots,th\} \tag{4.1}$$

$$PDF_1^S(i) = \frac{1}{{\beta_1}^{\alpha_1} * \Gamma(\alpha_1)} \left(\lambda_1(i-\varphi_1)\right)^{\alpha_1 - 1} e^{\frac{-\lambda_1(i-\varphi_1)}{\beta_1}} \quad , para\ i \in \{th+1,\dots,254,255\} \tag{4.2}$$

donde el parámetro α_j (real positivo) describe la forma, β_j (real positivo) describe la escala, φ_j (real positivo o negativo) describe la locación, y λ_j (1 o -1) describe el signo, el cual indica si la PDF_j^S , $j = \{0,1\}$, no está reflejada o sí está reflejada en el eje $\boldsymbol{y}$, respectivamente. La Figura 5 muestra un ejemplo de una PDF^S de distribuciones gamma con parámetros $P: \{th = 140, S_0: \{\lambda_0 = -1, \varphi_0 = 147, \beta_0 = 25, \alpha_0 = 2\}, S_1: \{\lambda_1 = 1, \varphi_1 = 130, \beta_1 = 16, \alpha_1 = 3\}, w_0 = 0.56, w_1 = 0.44\}$.

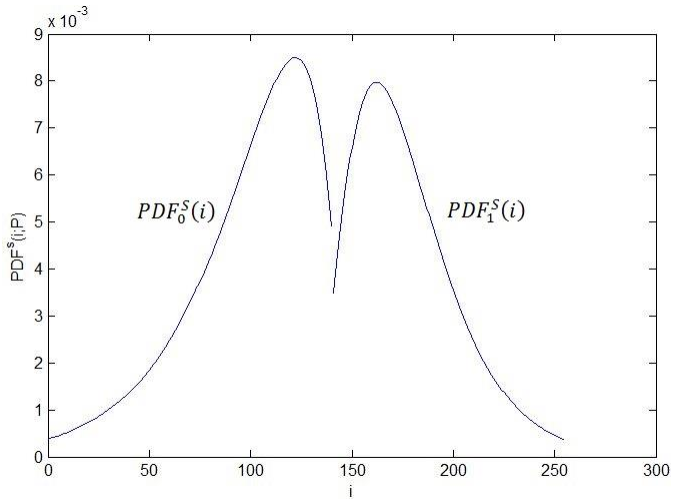

Figura 5: Ejemplo de una $\boldsymbol{PDF^S}$ de distribuciones gamma con parámetros $\boldsymbol{P}$: $\{\boldsymbol{th = 140}, \boldsymbol{S_0}: \{\boldsymbol{\lambda_0 = -1}, \boldsymbol{\varphi_0 = 147}, \boldsymbol{\beta_0 = 25}, \boldsymbol{\alpha_0 = 2}\}, \boldsymbol{S_1}: \{\boldsymbol{\lambda_1 = 1}, \boldsymbol{\varphi_1 = 130}, \boldsymbol{\beta_1 = 16}, \boldsymbol{\alpha_1 = 3}\}, \boldsymbol{w_0 = 0.56}, \boldsymbol{w_1 = 0.44}\}$.

Insertando las ecuaciones (4.1) y (4.2) en la función de verosimilitud base descrita en (3.9) se obtiene la función de verosimilitud del algoritmo propuesto

$$\ln\big(p(I|th)\big) = \sum_{i=0}^{th} h(i) \ln\left(\frac{w_0}{{\beta_0}^{\alpha_0} * \Gamma(\alpha_0)} \big(\lambda_0(i-\varphi_0)\big)^{\alpha_0 - 1} e^{\frac{-\lambda_0(i-\varphi_0)}{\beta_0}}\right) + \sum_{i=th+1}^{255} h(i) \ln\left(\frac{w_1}{{\beta_1}^{\alpha_1} * \Gamma(\alpha_1)} \big(\lambda_1(i-\varphi_1)\big)^{\alpha_1 - 1} e^{\frac{-\lambda_1(i-\varphi_1)}{\beta_1}}\right) \tag{4.3}$$

Una versión extendida de esta función, la cual puede ser usada por conveniencia para la programación del algoritmo, se obtiene inicialmente insertando (4.1) y (4.2) en la función de verosimilitud base extendida descrita en (3.12)

$$\ln\big(p(I|th)\big) = \{c_{0,th} * N * \ln(w_0) + c_{1,th} * N * \ln(w_1)\} + \left\{\sum_{i=0}^{th} h(i) \ln\left(\frac{1}{{\beta_0}^{\alpha_0} * \Gamma(\alpha_0)} \big(\lambda_0(i-\varphi_0)\big)^{\alpha_0 - 1} e^{\frac{-\lambda_0(i-\varphi_0)}{\beta_0}}\right)\right\} + \left\{\sum_{i=th+1}^{255} h(i) \ln\left(\frac{1}{{\beta_1}^{\alpha_1} * \Gamma(\alpha_1)} \big(\lambda_1(i-\varphi_1)\big)^{\alpha_1 - 1} e^{\frac{-\lambda_1(i-\varphi_1)}{\beta_1}}\right)\right\} \tag{4.4}$$

y después expandiendo el segundo y tercer término

$$\sum_{i=0}^{th} h(i) \ln\left(\frac{1}{{\beta_0}^{\alpha_0} * \Gamma(\alpha_0)} \left(\lambda_0(i-\varphi_0)\right)^{\alpha_0-1} e^{\frac{-\lambda_0(i-\varphi_0)}{\beta_0}}\right)$$
$$+ \sum_{i=th+1}^{255} h(i) \ln\left(\frac{1}{{\beta_1}^{\alpha_1} * \Gamma(\alpha_1)} \left(\lambda_1(i-\varphi_1)\right)^{\alpha_1-1} e^{\frac{-\lambda_1(i-\varphi_1)}{\beta_1}}\right) \quad (4.5)$$

de la siguiente manera. Aplicando el logaritmo natural a los términos de los paréntesis

$$= \sum_{i=0}^{th} h(i) \left(\ln\left(\frac{1}{{\beta_0}^{\alpha_0} * \Gamma(\alpha_0)}\right) + \ln\left(\left(\lambda_0(i-\varphi_0)\right)^{\alpha_0-1}\right) + \ln\left(e^{\frac{-\lambda_0(i-\varphi_0)}{\beta_0}}\right)\right)$$
$$+ \sum_{i=th+1}^{255} h(i) \left(\ln\left(\frac{1}{{\beta_1}^{\alpha_1} * \Gamma(\alpha_1)}\right) + \ln\left(\left(\lambda_1(i-\varphi_1)\right)^{\alpha_1-1}\right) + \ln\left(e^{\frac{-\lambda_1(i-\varphi_1)}{\beta_1}}\right)\right) \quad (4.6)$$

Aplicando propiedades del logaritmo natural

$$= \sum_{i=0}^{th} h(i) \left(-\ln({\beta_0}^{\alpha_0} * \Gamma(\alpha_0)) + (\alpha_0 - 1)\ln\left(\lambda_0(i-\varphi_0)\right) + \frac{-\lambda_0(i-\varphi_0)}{\beta_0}\right)$$
$$+ \sum_{i=th+1}^{255} h(i) \left(-\ln({\beta_1}^{\alpha_1} * \Gamma(\alpha_1)) + (\alpha_1 - 1)\ln\left(\lambda_1(i-\varphi_1)\right) + \frac{-\lambda_1(i-\varphi_1)}{\beta_1}\right) \quad (4.7)$$

Sacando los términos constantes

$$= -\ln\left({\beta_0}^{\alpha_0} * \Gamma(\alpha_0)\right) \sum_{i=0}^{th} h(i) + \sum_{i=0}^{th} h(i) \left((\alpha_0 - 1)\ln\left(\lambda_0(i-\varphi_0)\right) + \frac{-\lambda_0(i-\varphi_0)}{\beta_0}\right) - \ln({\beta_1}^{\alpha_1} * \Gamma(\alpha_1)) \sum_{i=th+1}^{255} h(i)$$
$$+ \sum_{i=th+1}^{255} h(i) \left((\alpha_1 - 1)\ln\left(\lambda_1(i-\varphi_1)\right) + \frac{-\lambda_1(i-\varphi_1)}{\beta_1}\right) \quad (4.8)$$

Introduciendo $c_{0,th} = \sum_{i=0}^{th} h(i)/N$ y $c_{1,th} = \sum_{i=th+1}^{255} h(i)/N$

$$= -\ln\left({\beta_0}^{\alpha_0} * \Gamma(\alpha_0)\right) c_{0,th} * N + \sum_{i=0}^{th} h(i) \left((\alpha_0 - 1)\ln\left(\lambda_0(i-\varphi_0)\right) + \frac{-\lambda_0(i-\varphi_0)}{\beta_0}\right) - \ln({\beta_1}^{\alpha_1} * \Gamma(\alpha_1))\, c_{1,th} * N$$
$$+ \sum_{i=th+1}^{255} h(i) \left((\alpha_1 - 1)\ln\left(\lambda_1(i-\varphi_1)\right) + \frac{-\lambda_1(i-\varphi_1)}{\beta_1}\right) \quad (4.9)$$

Se obtiene entonces la versión extendida de la función de verosimilitud del algoritmo propuesto

$$\ln\left(p(I|th)\right) = \left\{c_{0,th} * N * \ln(w_0) + c_{1,th} * N * \ln(w_1)\right\}$$
$$+ \left\{-\ln\left({\beta_0}^{\alpha_0} * \Gamma(\alpha_0)\right) c_{0,th} * N + \sum_{i=0}^{th} \left(h(i) \left((\alpha_0 - 1)\ln\left(\lambda_0(i-\varphi_0)\right) + \frac{-\lambda_0(i-\varphi_0)}{\beta_0}\right)\right)\right\}$$
$$+ \left\{-\ln({\beta_1}^{\alpha_1} * \Gamma(\alpha_1))\, c_{1,th} * N + \sum_{i=th+1}^{255} \left(h(i) \left((\alpha_1 - 1)\ln\left(\lambda_1(i-\varphi_1)\right) + \frac{-\lambda_1(i-\varphi_1)}{\beta_1}\right)\right)\right\} \quad (4.10)$$

4.2 Estimación de parámetros

Los parámetros estimados $\hat{P}: \{th_{op}, \hat{S}_0: \{\hat{\lambda}_0, \hat{\varphi}_0, \hat{\beta}_0, \hat{\alpha}_0\}, \hat{S}_1: \{\hat{\lambda}_1, \hat{\varphi}_1, \hat{\beta}_1, \hat{\alpha}_1\}, \hat{w}_0, \hat{w}_1\}$ pueden ser obtenidos mediante diversos métodos numéricos [4, 8, 11, 19] que usualmente son complicados, sin embargo, en este trabajo se obtuvieron de manera relativamente sencilla. Primeramente, se desarrolló un método de búsqueda exhaustiva de parámetros. Después, dado que los tiempos de procesamiento eran altos, se desarrolló un método basado en el algoritmo de optimización *Downhill Simplex* [22]. Irónicamente, los resultados obtenidos con el método basado en el algoritmo de optimización *Downhill Simplex* fueron ligeramente mejores a los obtenidos con el método de búsqueda exhaustiva de parámetros, por lo que este fue el usado al ser comparado con los algoritmos de Kittler, Kurita y Otsu.

4.2.1 Método exhaustivo

El algoritmo propuesto basado en el método exhaustivo empieza por encontrar los siguientes términos

$first0$ Primer valor de intensidad del histograma (de izquierda a derecha) mayor a cero.

$last0$ Último valor de intensidad del histograma (de izquierda a derecha) mayor a cero perteneciente a la clase 0 (valores de intensidad menores o iguales al umbral th).

$first1$ Primer valor de intensidad del histograma (de izquierda a derecha) mayor a cero perteneciente a la clase 1 (valores de intensidad mayores al umbral th).

$last1$ Último valor de intensidad del histograma (de izquierda a derecha) mayor a cero.

Después, para cada posible umbral $th = \{first0, first0 + 1, first0 + 2, \dots, < last1\}$, el algoritmo encuentra en conjunto de parámetros estimados preliminar $\hat{P}_{th}$. De todos los conjuntos de parámetros preliminares obtenidos, el $\hat{P}_{th}$ que maximiza la función de

verosimilitud extendida (Ecuación (4.10)) es tomado como el conjunto de parámetros óptimo $\hat{P}$, del cual se extrae el umbral óptimo th_{op}.

Para encontrar el conjunto $\hat{P}_{th}$ asociado a cada th, el algoritmo debe maximizar el primero, segundo y tercer término de la función de verosimilitud extendida (Ecuación (4.10)). Para maximizar el primer término

$$\{c_{0,th} * N * \ln(w_{0,th}) + c_{1,th} * N * \ln(w_{1,th})\} \tag{4.16}$$

el algoritmo toma los MLEs (parámetros estimados por máxima verosimilitud, ver Apéndice A) de los pesos: $w_{0,th} = c_{0,th}$ y $w_{1,th} = c_{1,th}$, donde $c_{0,th} = \sum_{i=0}^{th} PDF^I(i)$ y $c_{1,th} = \sum_{i=th+1}^{255} PDF^I(i)$. Para maximizar el segundo término

$$\left\{-\ln\left(\beta_{0,th}{}^{\alpha_{0,th}} * \Gamma(\alpha_{0,th})\right) c_{0,th} * N + \sum_{i=0}^{th}\left(h(i)\left((\alpha_{0,th}-1)\ln\left(\lambda_{0,th}(i-\varphi_{0,th})\right) + \frac{-\lambda_{0,th}(i-\varphi_{0,th})}{\beta_{0,th}}\right)\right)\right\} \tag{4.17}$$

se evalúan 200 combinaciones de los parámetros $S_{0,th}$: $\{\lambda_{0,th}, \varphi_{0,th}, \beta_{0,th}, \alpha_{0,th}\}$, de manera que la combinación cuyos parámetros den un resultado mayor en la evaluación es la que contiene los parámetros óptimos. A continuación se muestran las 200 combinaciones en dos conjuntos de 100 combinaciones cada uno

$$\{\lambda_{0,th}: \{1\}, \varphi_{0,th}: \{first0-1, first0-2, \dots, first0-100\}, \beta_{0,th}, \alpha_{0,th}\}$$
$$\{\lambda_{0,th}: \{-1\}, \varphi_{0,th}: \{last0+1, last0+2, \dots, last0+100\}, \beta_{0,th}, \alpha_{0,th}\}$$

donde $\beta_{0,th}$ y $\alpha_{0,th}$ son dependientes del valor de $\varphi_{0,th}$ de la siguiente manera

$$\beta_{0,\mathrm{th}} = \frac{\sigma_{0,th}^2}{|\mu_{0,th} - \varphi_{0,th}|}, \qquad \alpha_{0,\mathrm{th}} = \frac{|\mu_{0,th} - \varphi_{0,th}|}{\beta_{0,\mathrm{th}}} \tag{4.18}$$

donde a su vez $\mu_{0,th}$ es la media de la clase 0 dado el umbral, y σ_{0,th_A}^2 es la varianza de la clase 0 dado el umbral, descritas a continuación

$$\mu_{0,th} = \frac{1}{c_{0,th}} \sum_{i=0}^{th} i * PDF^I(i), \qquad \sigma_{0,th}^2 = \frac{1}{c_{0,th}} \sum_{i=0}^{th} (i - \mu_{0,th})^2 * PDF^I(i) \tag{4.19}$$

Las primeras 100 combinaciones modelan la clase 0 del histograma con distribuciones gamma de signo positivo. Estas varían su parámetro de locación de $first0 - 1$ hasta $first0 - 100$. Empieza en $first0 - 1$ porque la clase 0 va de $first0$ a $last0$, y termina en $first0 - 100$ porque se comprobó durante la evaluación que ubicar el parámetro de locación a una distancia más lejana a esta computacionalmente no hace una diferencia significativa debido al redondeo. Las segundas 100 combinaciones modelan la clase 0 del histograma con distribuciones gamma de signo negativo. Estas varían su parámetro de locación de $last0 + 1$ hasta $last0 + 100$. Empieza en $last0 + 1$ porque la clase 0 va de $first0$ a $last0$, y termina en $last0 + 100$ porque también se comprobó durante la evaluación que ubicar el parámetro de locación a una distancia más lejana a esta computacionalmente no hace una diferencia significativa debido al redondeo. Análogamente a la maximización del segundo término, para maximizar el tercer término

$$\left\{-\ln\left(\beta_1{}^{\alpha_1} * \Gamma(\alpha_1)\right) c_{1,th} * N + \sum_{i=th+1}^{255} \left(h(i) \left((\alpha_1 - 1) \ln\left(\lambda_1 (i - \varphi_1)\right) + \frac{-\lambda_1 (i - \varphi_1)}{\beta_1} \right) \right) \right\} \tag{4.20}$$

se evalúan 200 combinaciones de los parámetros $S_{1,th}: \{\lambda_{1,th}, \varphi_{1,th}, \beta_{1,th}, \alpha_{1,th}\}$, de manera que la combinación cuyos parámetros den un resultado mayor en la evaluación es la que contiene los parámetros óptimos. A continuación se muestran las 200 combinaciones en dos conjuntos de 100 combinaciones cada uno

$$\{\lambda_{1,th}: \{1\}, \varphi_{1,th}: \{first1 - 1, first1 - 2, \dots, first1 - 100\}, \beta_{1,th}, \alpha_{1,th}\}$$
$$\{\lambda_{1,th}: \{-1\}, \varphi_{1,th}: \{last1 + 1, last1 + 2, \dots, last1 + 100\}, \beta_{1,th}, \alpha_{1,th}\}$$

donde $\beta_{1,th}$ y $\alpha_{1,th}$ son dependientes del valor de $\varphi_{1,th}$ de la siguiente manera

$$\beta_{1,\mathrm{th}} = \frac{\sigma_{1,th}^2}{\left|\mu_{1,th} - \varphi_{1,th}\right|}, \qquad \alpha_{1,\mathrm{th}} = \frac{\left|\mu_{1,th} - \varphi_{1,th}\right|}{\beta_{1,\mathrm{th}}} \tag{4.21}$$

donde a su vez $\mu_{1,th}$ es la media de la clase 1 dado el umbral, y $\sigma_{1,th}^2$ es la varianza de la clase 1 dado el umbral, descritas a continuación (basado en 2.2.1.1)

$$\mu_{1,th} = \frac{1}{c_{1,th}} \sum_{i=th+1}^{255} i * PDF^{I}(i) \ , \quad \sigma_{1,th}^{2} = \frac{1}{c_{1,th}} \sum_{i=th+1}^{255} \left(i - \mu_{1,th}\right)^{2} PDF^{I}(i) \qquad (4.22)$$

Las tres combinaciones que maximizaron los tres términos de la función de verosimilitud son tomadas como parte de los parámetros óptimos preliminares

$$\hat{P}_{th}: \{th, \hat{S}_{0,th}: \{\hat{\lambda}_{0,th}, \hat{\varphi}_{0,th}, \hat{\beta}_{0,th}, \hat{\alpha}_{0,th}\}, \hat{S}_{1,th}: \{\hat{\lambda}_{1,th}, \hat{\varphi}_{1,th}, \hat{\beta}_{1,th}, \hat{\alpha}_{1,th}\}, \hat{w}_{0,th}, \hat{w}_{1,th}\}$$

4.2.2 Método basado en el algoritmo *Downhill Simplex*

El algoritmo *Downhill Simplex* [22] es un popular algoritmo de optimización matemática. Este optimiza una función dependiente de M parámetros a un costo computacional muy bajo, usando solo evaluaciones de la función y no derivadas. El algoritmo hace uso de una figura geométrica llamada simplex. Esta es de M dimensiones y $M + 1$ vértices; por ejemplo, en dos dimensiones es un triángulo y en tres dimensiones es un tetraedro. Las coordenadas de cada uno de los $M + 1$ vértices están descritas por distintos valores de los M parámetros a optimizar. Para la optimización, el algoritmo construye un simplex, cuyos vértices son inicializados en ciertos valores predeterminados de los parámetros. A partir de aquí, el algoritmo hace su camino a través de la compleja topografía M-dimensional dada por la función a optimizar hasta llegar a un mínimo (o máximo). Para recorrer este camino, en cada iteración el algoritmo toma el vértice cuya evaluación en la función a optimizar (función de verosimilitud en este caso) es mayor (o menor) y le hace una modificación a sus coordenadas, cambiando así la forma del simplex y la ubicación de su centroide. Cuando las modificaciones ya no hacen cambios significativos en las evaluaciones de la función de verosimilitud entonces el algoritmo se detiene y da como salida un simplex optimizado. Las coordenadas del centroide del simplex optimizado son los parámetros óptimos.

En la Figura 6 se muestran los cuatro tipos de modificaciones hechos al simplex: a. *reflexión*, el vértice a modificar (color rojo) es reflejado respecto a los demás vértices; b.

reflexión y expansión, se hace la reflexión descrita en "a." pero de manera extendida; c. *contracción*, se acorta la distancia entre el vértice a modificar y los otros vértices; y d. *contracción múltiple*, se hace el simplex más pequeño, manteniendo fijo aquel vértice que evaluado en la función de verosimilitud sea menor (o mayor).

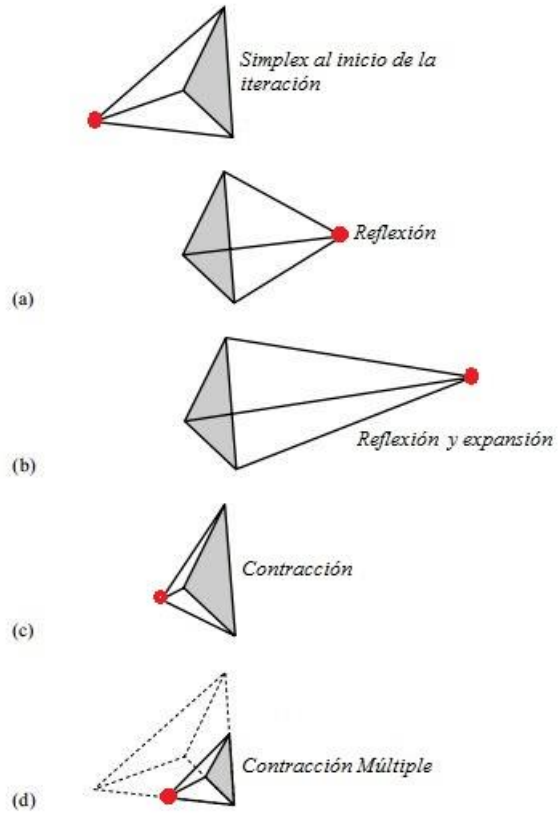

Figura 6: Tipos de modificaciones hechos al simplex en cada iteración [22].

El algoritmo propuesto basado en el algoritmo *Downhill Simplex* empieza por encontrar los siguientes términos

$first0$ Primer valor de intensidad del histograma (de izquierda a derecha) mayor a cero.

$last0$ Último valor de intensidad del histograma (de izquierda a derecha) mayor a cero perteneciente a la clase 0 (valores de intensidad menores o iguales al umbral th).

$first1$ Primer valor de intensidad del histograma (de izquierda a derecha) mayor a cero perteneciente a la clase 1 (valores de intensidad mayores al umbral th).

$last1$ Último valor de intensidad del histograma (de izquierda a derecha) mayor a cero.

Entonces, de manera heurística, para cada umbral $th = \{first0, first0 + 24, first0 + 48, first0 + 72, \dots, < (last1 + 8)\}$, el algoritmo encuentra a partir del uso del algoritmo *Downhill Simplex* con $M = 3$ un conjunto preliminar de parámetros estimados $\hat{P}_{th}$. De todos los conjuntos obtenidos, el $\hat{P}_{th}$ obtenido que maximiza la función de verosimilitud extendida (Ecuación (4.10)) es tomado como el conjunto de parámetros óptimo $\hat{P}$, del cual se extrae el umbral óptimo th_{op}.

Para encontrar el conjunto preliminar $\hat{P}_{th}$ de cada umbral th probado, se construye un simplex tridimensional compuesto por $M + 1 = 4$ vértices $\{A, B, C, D\}$ del tipo $V: \{th, \varphi_0, \varphi_1\}$, los cuales en este trabajo son inicializados de manera heurística de la siguiente manera

$$V_A: \{th_A = th - 16, \quad \varphi_{0,A} = first0 - 5, \quad \varphi_{1,A} = th - 16 - 5\}$$
$$V_B: \{th_B = th - 12, \quad \varphi_{0,B} = first0 - 5, \quad \varphi_{1,B} = last1 + 5\}$$
$$V_C: \{th_C = th + 12, \quad \varphi_{0,C} = th + 12 + 5, \quad \varphi_{1,C} = th + 12 - 5\}$$
$$V_D: \{th_D = th + 16, \quad \varphi_{0,D} = th + 16 + 5, \quad \varphi_{1,D} = last1 + 5\}$$

El vértice V_A es inicializado de esa manera para modelar la clase 0 y la clase 1 con distribuciones gamma de signos positivos; el vértice V_B para modelar la clase 0 con una distribución gamma de signo positivo y la clase 1 con una distribución gamma de signo negativo; el vértice V_C para modelar la clase 0 con una distribución gamma de signo negativo y la clase 1 con una distribución gamma de signo positivo; y el vértice V_D para modelar la clase 0 y la clase 1 con distribuciones gamma de signos negativos. Es importante notar que los umbrales $\{th_A, th_B, th_C, th_D\}$ varian en su inicialización respecto a th para que los vértices del simplex estén más separados, y así evitar que el simplex se convierta en un simplex degenerado [22]. A partir de los valores de los parámetros (o

coordenadas) de cada vértice, el resto de parámetros de la función de verosimilitud R: $\{\lambda_0, \beta_0, \alpha_0, \lambda_1, \beta_1, \alpha_1, w_0, w_1\}$ pueden ser obtenidos directamente de la siguiente manera. Usando el vértice V_A como ejemplo, los pesos $w_{0,A}$ y $w_{1,A}$ son los MLEs (parámetros estimados por máxima verosimilitud, ver Apéndice A)

$$w_{0,A} = c_{0,th_A} \qquad w_{1,A} = c_{1,th_A} \tag{4.11}$$

donde $c_{0,th_A} = \sum_{i=0}^{th_A} PDF^I(i)$ y $c_{1,th_A} = \sum_{i=th_A+1}^{255} PDF^I(i)$. Para obtener el resto de parámetros se necesita antes calcular las medias (μ_{0,th_A} y μ_{1,th_A}) y varianzas (σ^2_{0,th_A} y σ^2_{1,th_A}) de ambas clases dado el umbral, de la siguiente manera

$$\mu_{0,th_A} = \frac{1}{c_{0,th_A}} \sum_{i=0}^{th_A} i * PDF^I(i), \quad \mu_{1,th_A} = \frac{1}{c_{1,th_A}} \sum_{i=th_A+1}^{255} i * PDF^I(i) \tag{4.12}$$

$$\sigma^2_{0,th_A} = \frac{1}{c_{0,th_A}} \sum_{i=0}^{th_A} \left(i - \mu_{0,th_A}\right)^2 * PDF^I(i), \quad \sigma^2_{1,th_A} = \frac{1}{c_{1,th_A}} \sum_{i=th_A+1}^{255} \left(i - \mu_{1,th_A}\right)^2 PDF^I(i) \tag{4.13}$$

A partir de estos, el parámetro de signo $\lambda_{0,A}$ se toma como igual a 1 si $\varphi_{0,A} \leq \mu_{0,th_A}$, o igual a -1 en caso contrario. Análogamente, el parámetro de signo $\lambda_{1,A}$ se toma como igual a 1 si $\varphi_{1,A} \leq \mu_{1,th_A}$, o igual a -1 en caso contrario. Antes de continuar, se hace el siguiente reajuste de los parámetros de locación $\varphi_{0,A}$ y $\varphi_{1,A}$ para asegurarse que la PDF^S vaya a ser mayor a cero en todo el rango de valores mayores a cero de la PDF^I; esto porque las funciones gamma no van hacia infinito en ambos sentidos como la Gaussiana, más bien, son iguales a cero a la izquierda del parámetro de locación cuando el parámetro de signo es positivo, e iguales a cero a la derecha del parámetro de locación cuando el parámetro de signo es negativo.

Si $first0 \leq \varphi_{0,A} \leq \mu_{0,th_A}$, entonces $\varphi_{0,A} = first0 - 1$

Si $\mu_{0,th_A} < \varphi_{0,A} \leq last0$, entonces $\varphi_{0,A} = last0 + 1$

Si $first1 \leq \varphi_{1,A} \leq \mu_{1,th_A}$, entonces $\varphi_{1,A} = first1 - 1$

Si $\mu_{1,th_A} < \varphi_{1,A} \leq last1$, entonces $\varphi_{1,A} = last1 + 1$

Después del reajuste, los parámetros $\{\beta_{0,A}, \alpha_{0,A}, \beta_{1,A}, \alpha_{1,A}\}$ se calculan de la siguiente manera

$$\beta_{0,A} = \frac{\sigma_{0,th_A}^2}{\left|\mu_{0,th_A} - \varphi_{0,A}\right|}, \qquad \alpha_{0,A} = \frac{\left|\mu_{0,th_A} - \varphi_{0,A}\right|}{\beta_{0,A}} \tag{4.14}$$

$$\beta_{1,A} = \frac{\sigma_{1,th_A}^2}{\left|\mu_{1,th_A} - \varphi_{1,A}\right|}, \qquad \alpha_{1,A} = \frac{\left|\mu_{1,th_A} - \varphi_{1,A}\right|}{\beta_{1,A}} \tag{4.15}$$

El algoritmo *Downhill Simplex* puede hacer que se obtengan valores de $\beta_{0,A}$ y $\beta_{1,A}$ muy bajos, o valores de $\alpha_{0,A}$ y $\alpha_{1,A}$ muy altos. Dado esto, se encontró de manera experimental que con los siguientes reajustes se resolvía este problema

Si $\beta_{0,A} < 0.05$, entonces $\beta_{0,A} = 0.05$

Si $\beta_{1,A} < 0.05$, entonces $\beta_{1,A} = 0.05$

Si $\alpha_{0,A} > 30$, entonces $\alpha_{0,A} = 30$

Si $\alpha_{1,A} > 30$, entonces $\alpha_{1,A} = 30$

Entonces, para obtener $\hat{P}_{th}$, el algoritmo hace dos acciones de manera iterativa: 1) evalúa los vértices V (con sus respectivos R) en la función de verosimilitud y 2) modifica las coordenadas de alguno de sus vértices; esto lo hace hasta lograr convergencia. A partir del simplex óptimo (convergencia), se calcula el vértice óptimo: $V_{op} = \frac{1}{4}(V_A + V_B + V_C + V_D)$ con su respectivo R, denominado R_{op}, de donde se obtiene $\hat{P}_{th}: \{V_{op}, R_{op}\}$.

A pesar de que la función de verosimilitud del algoritmo propuesto tiene 11 parámetros, se decidió utilizar el algoritmo *Downhill Simplex* para $M = 3$ parámetros $V: \{th, \varphi_0, \varphi_1\}$, donde el resto de parámetros $R: \{\lambda_0, \beta_0, \alpha_0, \lambda_1, \beta_1, \alpha_1, w_0, w_1\}$ se hicieron dependientes de estos. Esto se hizo por dos razones: 1. al ser los parámetros λ_0 y $\lambda_1$binarios no podían ser incluidos directamente en el algoritmo *Downhill Simplex* (y por consiguiente $\beta_0, \alpha_0, \beta_1, \alpha_1$ tampoco por ser dependientes de estos), ya que este algoritmo está hecho para parámetros continuos; y 2. los pesos w_0 y w_1 están condicionados a sumar 1, pero el algoritmo *Downhill Simplex* no toma en cuenta este tipo de restricciones, por lo que tampoco podían ser incluidos directamente.

4.3 Funciones de verosimilitud de Kittler, Kurita y Otsu como casos especiales de la función de verosimilitud del algoritmo propuesto

Basado en [3], la función de verosimilitud de Kittler, y por consiguiente las de Kurita y Otsu, son casos especiales de la función de verosimilitud (versión normal o extendida) del algoritmo propuesto cuando φ tiende a $+\infty$ (para un valor positivo del parámetro de signo) o a $-\infty$ (para un valor negativo del parámetro de signo). Para lograr obtener la función de verosimilitud de Kittler a partir de la del algoritmo propuesto se debe aplicar la siguiente transformación

$$\mu = \alpha * \beta + \varphi \tag{4.23}$$

$$\sigma^2 = \beta^2 * \alpha \tag{4.24}$$

a los parámetros de la función de verosimilitud extendida del algoritmo propuesto (Ecuación (4.10)). Después se debe hacer una simplificación, en la cual se usa la Fórmula de *Stirling* [7] para valores grandes de α

$$\Gamma(\alpha + 1) \sim \sqrt{2\pi\alpha}\left(\frac{\alpha}{e}\right)^{\alpha} \tag{4.25}$$

Haciendo esta transformación y simplificación, de acuerdo a [3], se obtiene la función de verosimilitud de Kittler

$$\ln\big(p(I|th)\big) = N\sum_{j=0}^{1} c_{j,th} * \ln\big(w_j\big) - \frac{N}{2} * \ln[2\pi] - \frac{N}{2}\sum_{j=1}^{2} c_{j,th}\ln\big[\sigma_j^2\big] - \frac{N}{2} \tag{4.26}$$

de la cual como ya se mencionó, se puede obtener a partir de ella la de Kurita si se hace $\sigma_0^2 = \sigma_1^2 = \sigma_w^2$

$$\ln\big(p(I|th)\big) = N\sum_{j=0}^{1} c_{j,th} * \ln\big(w_j\big) - \frac{N}{2} * \ln[2\pi] - \frac{N}{2}\ln[\sigma_W^2] - \frac{N}{2} \tag{4.27}$$

y posteriormente a partir de la de Kurita la de Otsu si se hace $w_0 = w_1 = 0.5$

$$\ln\big(p(I|th)\big) = N * \ln(0.5) - \frac{N}{2} * \ln[2\pi] - \frac{N}{2}\ln[\sigma_W^2] - \frac{N}{2} \tag{4.28}$$

CAPÍTULO 5: RESULTADOS EXPERIMENTALES

5.1 Implementación

El algoritmo propuesto (ambos métodos), así como los de Kittler, Kurita y Otsu, fueron implementados en el lenguaje de programación C bajo la plataforma Visual C++ 2008, usando una computadora portátil con Windows 7, procesador Intel® core i3 de velocidad 4 GHz y una memoria RAM 4 GB.

5.2 Biblioteca de imágenes *IPCV-LAB Image Data Base*

Posterior a la implementación, se aplicaron estos cuatro algoritmos a las 132 imágenes de la biblioteca de imágenes llamada *IPCV-LAB Image Data Base*, la cual fue elaborada exclusivamente para este trabajo. Antes de elaborar esta biblioteca, se evaluaron otras bibliotecas de imágenes en Internet, sin embargo, en todas se encontraron por lo menos alguna de las siguientes razones que impidieron usarlas: 1. biblioteca que contiene imágenes constituidas por tres o más tipos de objetos distintos, y no por dos (objeto(s) y fondo); 2. biblioteca que contiene segmentaciones manuales en formato JPG, donde las líneas que delimitan la silueta o forma de los objetos en las segmentaciones manuales se ven borrosas; 3. biblioteca que contiene imágenes de documentos de texto sin segmentaciones manuales, las cuales se duraría demasiado tiempo segmentando manualmente; y 4. biblioteca protegida por derechos de autor o cuyos dueños no respondieron a la solicitud de compartirla. La *IPCV-LAB Image Data Base* está subdividida en siete distintas categorías de imágenes de acuerdo a la Tabla 1, las cuales son muy variadas y cubren una amplia gama de aplicaciones en ingeniería. Cada imagen cuenta con su respectiva segmentación manual, hecha pixel por pixel por un humano, y considerada como una segmentación perfecta.

Tabla 1: Categorías de la *IPCV-LAB Image Data Base*

No.	Categoría	Descripción	Imágenes
1	Imágenes sintéticas	Histogramas hechos por generador de números aleatorios basado en distribuciones gamma y Gaussianas [13, 22]	12
2	Gráficos por computador	Los gráficos fueron generados por un programa de animación digital	8
3	Imagenología	Imágenes capturadas con técnicas de imagenología (termografía, radiografía, rayos-x y sensado remoto)	13
4	Letras	Imágenes conteniendo letras hechas por computadora, crayola, lapicero, grafiti, etc.	33
5	Fotografías	Fotografías de personas, objetos, animales y naturaleza	42
6	Objetos pequeños	Objetos con un área menor al 4% del área total de la imagen	4
7	Bajo contraste	Imágenes de bajo contraste	20

5.3 Ejemplos de segmentaciones

En las Figuras 7-14, presentadas a continuación, se detallan los resultados de la aplicación de los cuatro algoritmos a 8 de las 132 imágenes (resultados del algoritmo propuesto obtenidos con método basado en el algoritmo *Downhill Simplex*). Las Figuras 7-14(a) muestran las imágenes originales a color; las Figuras 7-14(b) muestran las imágenes originales a escala de grises (imágenes de intensidad); las Figuras 7-14(c) muestran las segmentaciones manuales; las Figuras 7-14(d) muestran las PDF^{I} de las imágenes originales a escala de grises; las Figuras 7-14(e) muestran las segmentaciones hechas por el algoritmo propuesto; las Figuras 7-14(f,g,h) muestran las segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; las Figuras 7-14(i) muestran los modelos óptimos $PDF^{S,op}$ obtenidos por el algoritmo propuesto superpuestos sobre las PDF^{I} de las imágenes originales a escala de grises; y finalmente las Figuras 7-14(j,k,l)

muestran los modelos óptimos $PDF^{S,op}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre las PDF^{I} de las imágenes originales a escala de grises.

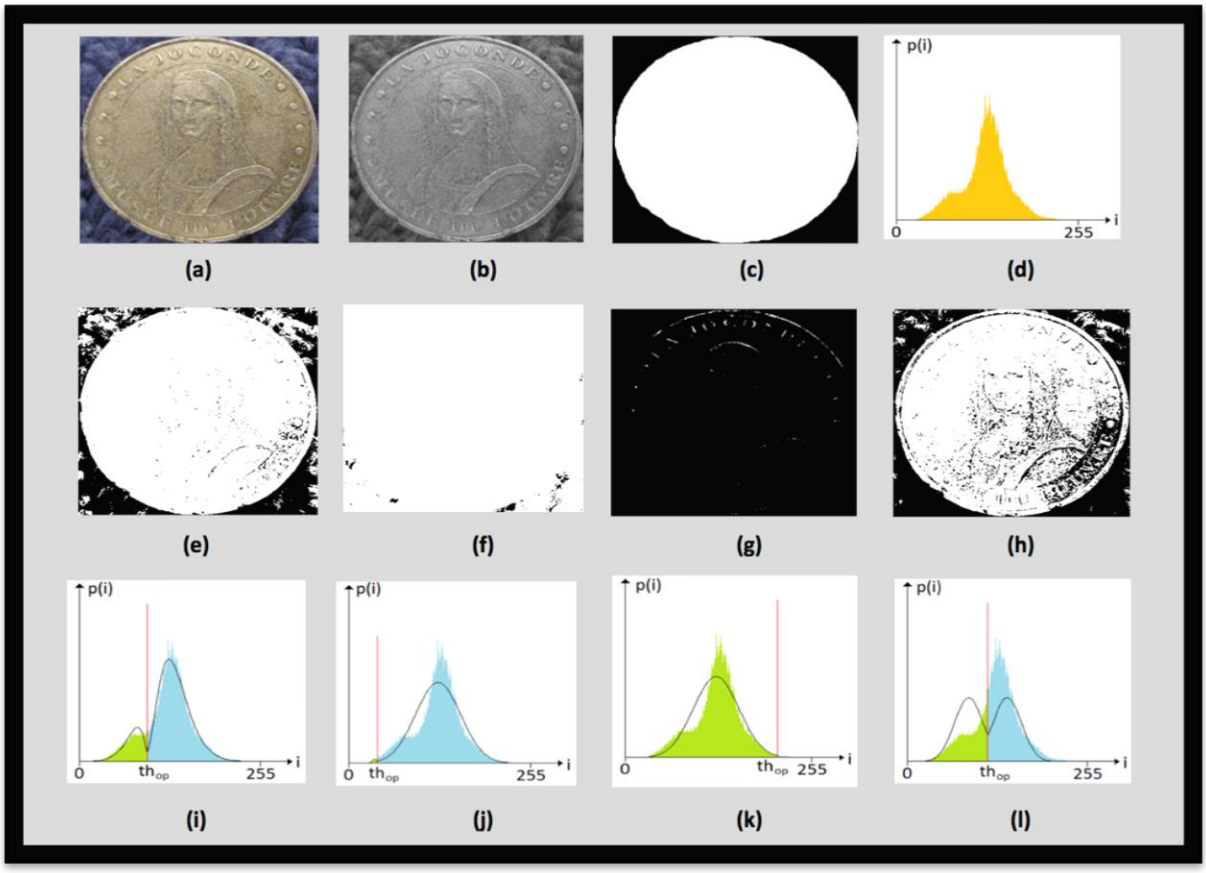

Figura 7: a) Imagen original a color de una moneda; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

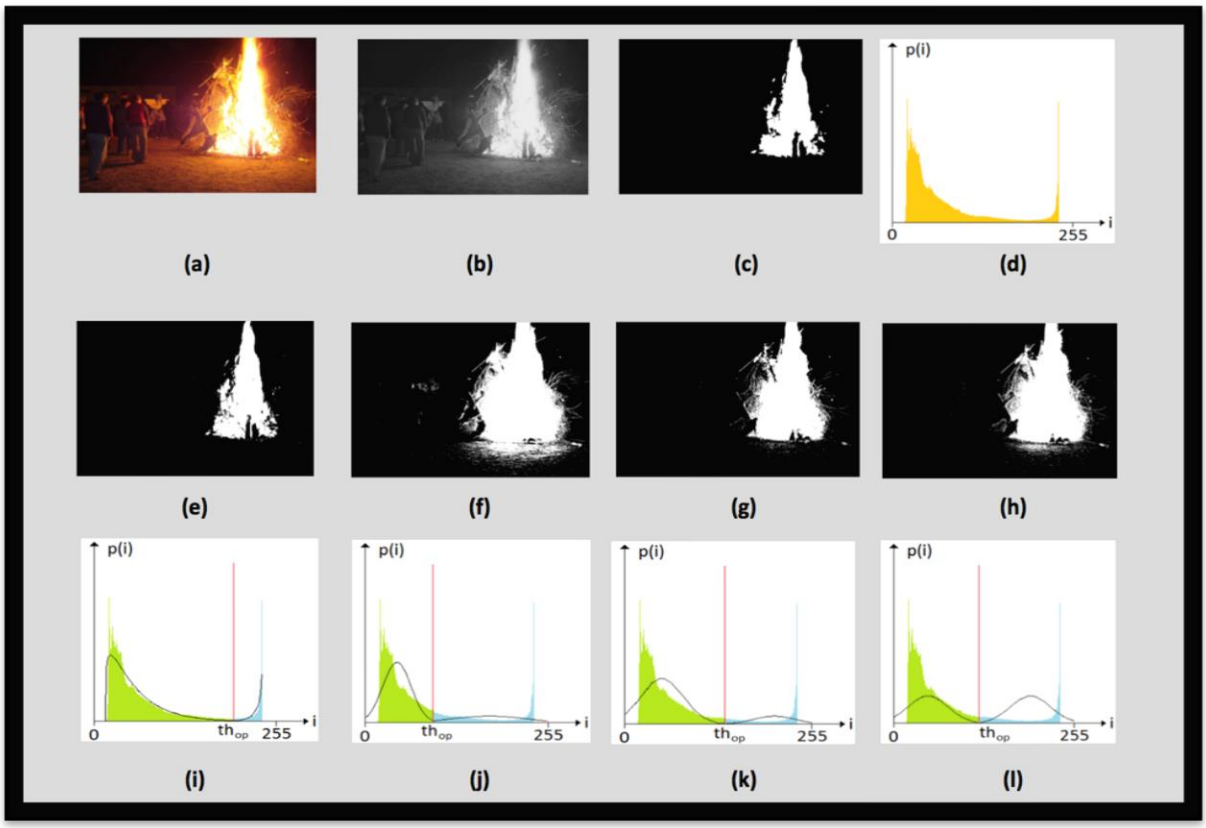

Figura 8: a) Imagen original a color de una fogata; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

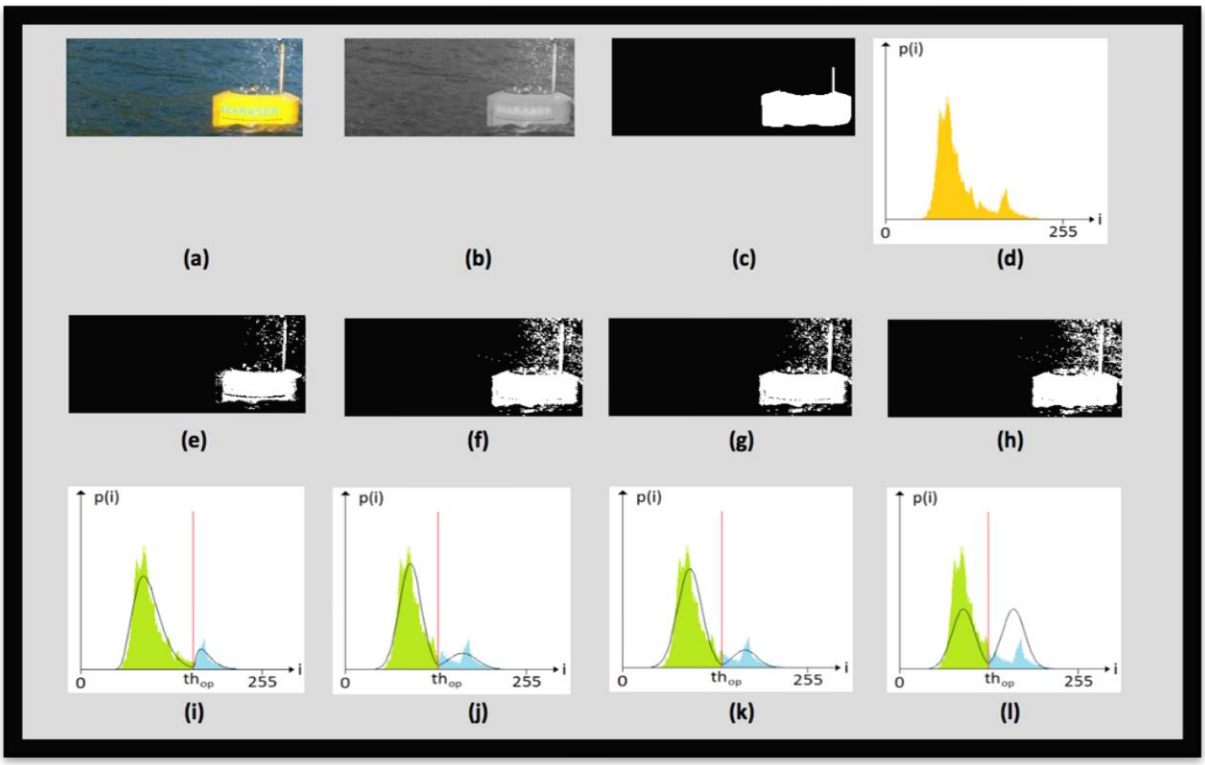

Figura 9: a) Imagen original a color de una boya; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

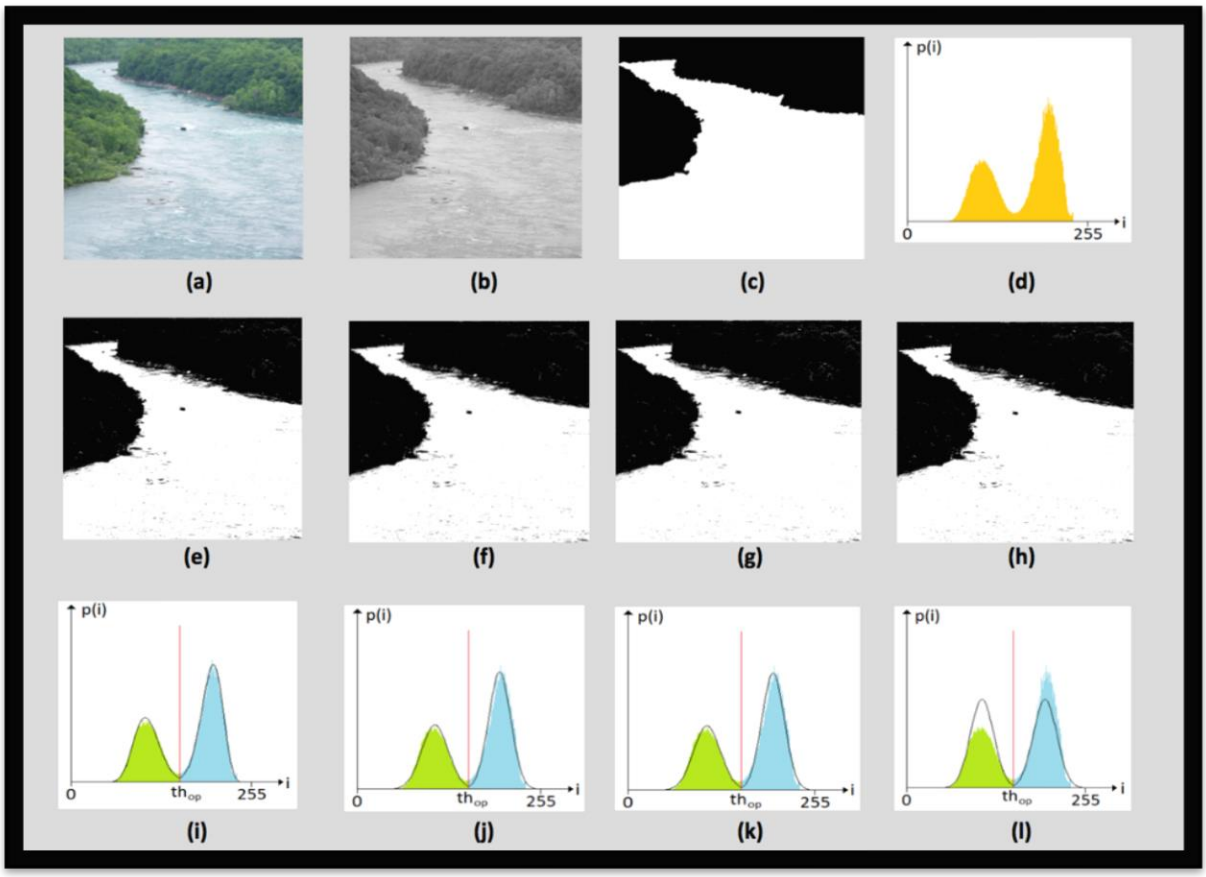

Figura 10: a) Imagen original a color de un rio y un bosque; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

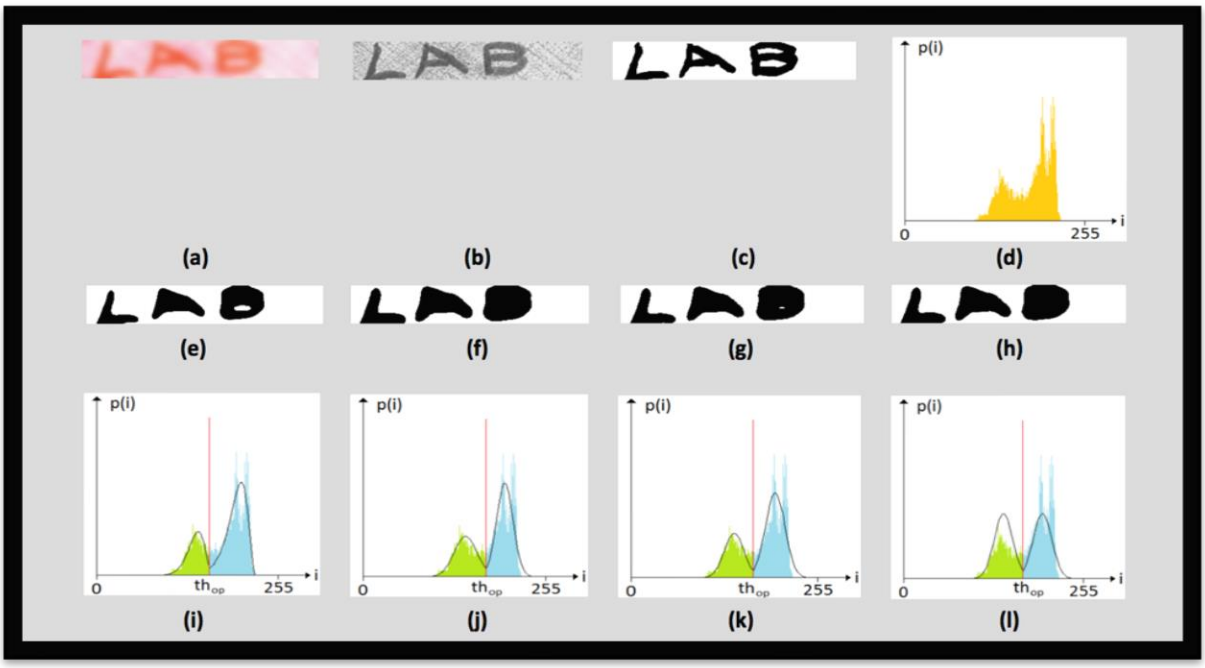

Figura 11: a) Imagen original a color de unas letras hechas con crayola; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

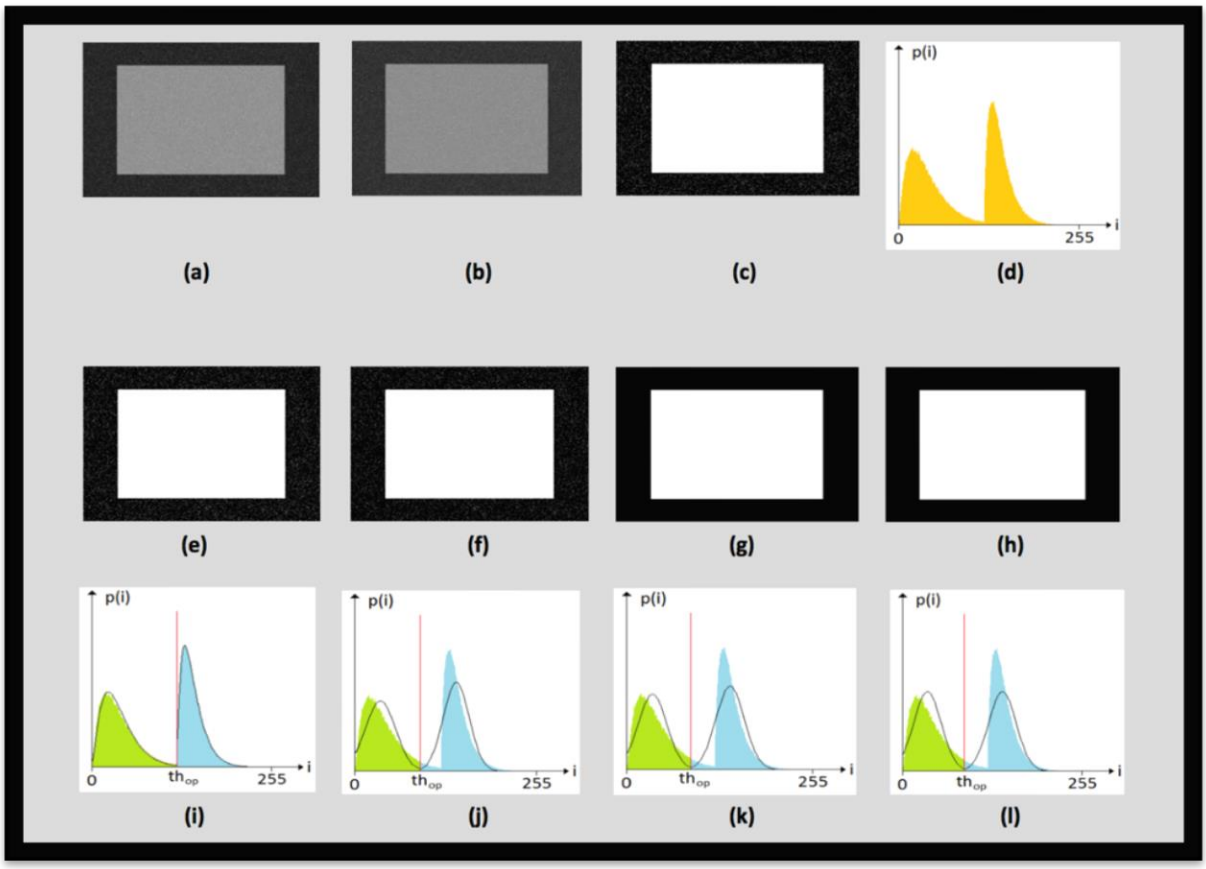

Figura 12: a) Imagen original a color de un cuadrado cuyo histograma fue hecho sintéticamente; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

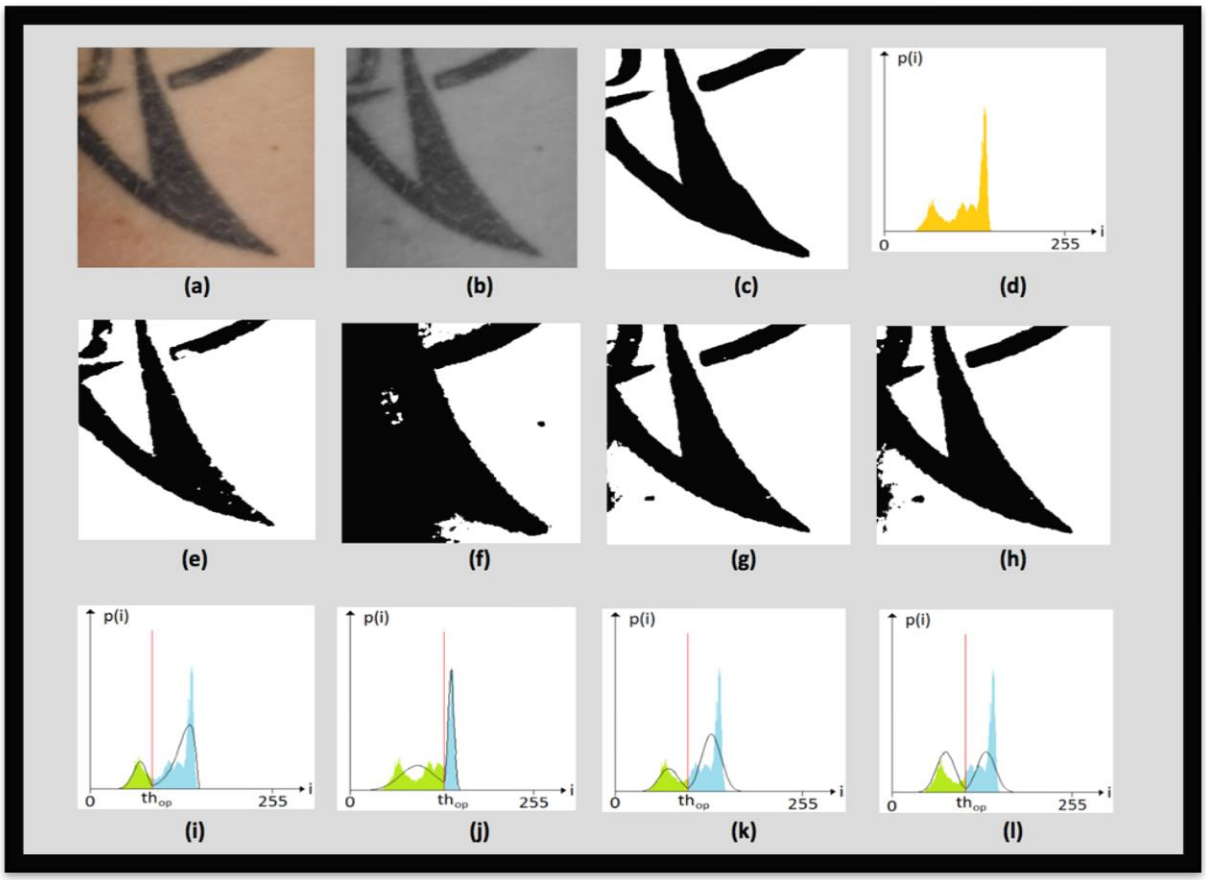

Figura 13: a) Imagen original a color de un tatuaje; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

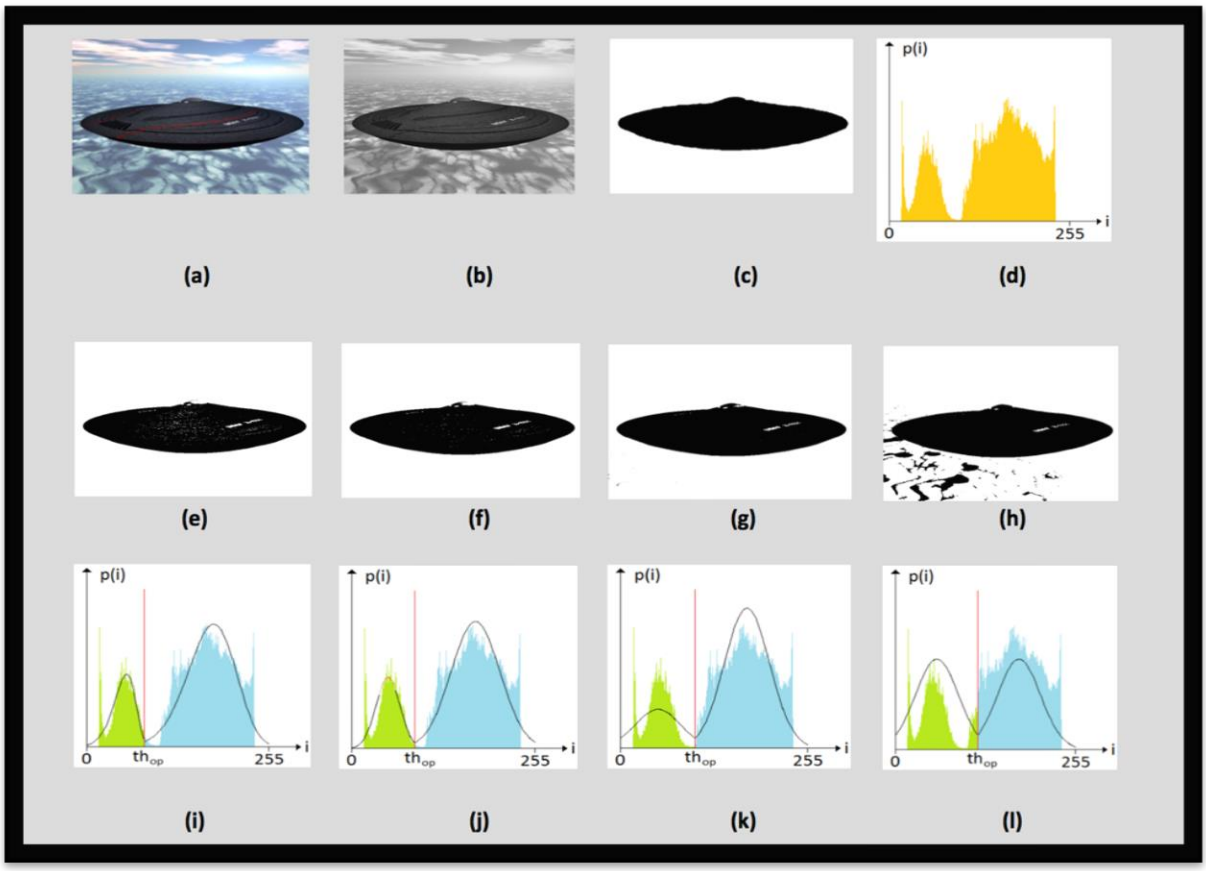

Figura 14: a) Imagen original a color de un OVNI; b) Versión a escala de grises de la imagen original; c) Segmentación manual; d) $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; e) Segmentación hecha por el algoritmo propuesto; f,g,h) Segmentaciones hechas por los algoritmos de Kittler, Kurita y Otsu, respectivamente; i) Modelo óptimo $\boldsymbol{PDF^{S,op}}$ obtenido por el algoritmo propuesto superpuesto sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises; j,k,l) Modelos óptimos $\boldsymbol{PDF^{S,op}}$ obtenidos por los algoritmos de Kittler, Kurita y Otsu, respectivamente, superpuestos sobre la $\boldsymbol{PDF^{I}}$ de la imagen original a escala de grises.

5.4 Evaluación de los resultados

Entre las varias maneras de evaluar el desempeño de los cuatro algoritmos en la segmentación de las 132 imágenes [24, 27, 28], se escogieron dos evaluaciones objetivas y una evaluación subjetiva. La primera evaluación objetiva es mediante el Error Cuadrático Medio (MSE), el cual se usa para medir el error entre la PDF^S obtenida y la PDF^I, con el propósito de evaluar qué tan bien se ajusta el modelo propuesto por cada algoritmo a los comportamientos estadísticos observados en las 132 imágenes de prueba. La segunda evaluación objetiva es mediante la métrica F1-score [6], la cual se usa para determinar la precisión de las segmentaciones hechas por los algoritmos. La evaluación subjetiva es mediante ocho humanos que califican de manera independiente la calidad de las segmentaciones, esto con el fin de corroborar los resultados obtenidos con el F1-score y medir cómo los humanos perciben las segmentaciones hechas por los algoritmos. Se reitera de nuevo, que para el algoritmo propuesto, los resultados evaluados fueron los obtenidos con el método basado en el algoritmo *Downhill Simplex*, debido a que este dio resultados ligeramente mejores que el método exhaustivo con tiempos de procesamiento mucho más bajos.

5.4.1 Evaluación objetiva usando el Error Cuadrático Medio (MSE)

En estadística, el Error Cuadrático Medio (MSE, por sus siglas en inglés: “Mean Squared Error”) cuantifica la diferencia entre los valores dados por un estimador y los valores reales de la cantidad estimada. En este trabajo, se calculó para cada una de las 132 imágenes el MSE entre los modelos óptimos $PDF^{S,op}$ (estimador) obtenidos por cada uno de los cuatro algoritmos y la PDF^I (cantidad estimada) de la imagen original [16]. La fórmula del MSE usada es

$$MSE = \frac{1}{255} * \sum_{i=0}^{255} \left(PDF^S(i) - PDF^I(i)\right)^2 \tag{5.2}$$

La Tabla 2 muestra el MSE promedio (escalado por un factor de10^7) para cada categoría de imágenes.

Tabla 2: Promedios de los MSE obtenidos para cada categoría de imágenes (escalados por un factor de $\mathbf{10^7}$)

Categoría	Propuesto	Kittler	Kurita	Otsu
Imágenes sintéticas	79	115	121	195
Gráficos por computador	301	350	687	894
Imagenología	147	215	291	509
Letras	60	86	203	407
Fotografías	58	86	150	271
Objetos pequeños	15	20	21	365
Bajo contraste	43	51	66	190
PROMEDIO	84	113	178	377

En la Tabla 2 se puede observar que el algoritmo propuesto obtuvo contundentemente los MSE más bajos para cada categoría de imágenes debido a la flexibilidad de la PDF^S compuesta por distribuciones gamma. En las Figuras 8 y 9, las cuales muestran imágenes que presentan comportamientos gamma en sus PDF^I, se puede observar cómo la $PDF^{S,op}$ del algoritmo propuesto hizo una mucho mejor aproximación que las $PDF^{S,op}$ de los otros tres algoritmos. En la Figura 10, la cual muestra una imagen que presenta comportamientos Gaussianos en su PDF^I , se puede notar cómo el algoritmo propuesto hizo una aproximación igual de buena que la de los modelos Gaussianos, ya que el modelo Gaussiano es un caso especial del gamma. En la Figura 12, la cual muestra una imagen con una PDF^I sintética compuesta por dos distribuciones gamma, se puede observar cómo el algoritmo propuesto logra aproximar las distribuciones gamma de una manera precisa mientras que los modelos Gaussianos de una manera imprecisa.

5.4.2 Evaluación objetiva usando la métrica F1-score

En teoría de la detección, el F1-score [6] es una métrica que mide la precisión con la cual un algoritmo detecta algo, por ejemplo un Objeto. Cuando un algoritmo detecta algo, se pueden obtener cuatro posibles resultados

1. Verdadero Positivo: lo que fue detectado se clasificó como Objeto, y la clasificación fue correcta.
2. Verdadero Negativo: lo que fue detectado se clasificó como No Objeto, y la clasificación fue correcta.
3. Falso Positivo: lo que fue detectado se clasificó como Objeto, pero la clasificación fue incorrecta.
4. Falso Negativo: lo que fue detectado se clasificó como No Objeto, pero la clasificación fue incorrecta.

Entonces, si se detectan uno o varios objetos, la precisión del algoritmo es la siguiente

$$F1 = \frac{VP}{2 * VP + FN + FP} \tag{5.1}$$

siendo VP la cantidad de Verdaderos Positivos obtenidos, FN la cantidad de Falsos Negativos obtenidos, y FP la cantidad de Falsos Positivos obtenidos. Esta métrica varía de 0 a 1, siendo 1 la precisión máxima.

En este trabajo, por conveniencia, los pixeles blancos de la imagen segmentada manualmente que fueron clasificados como blancos por los algoritmos representan los Verdaderos Positivos, y los que fueron clasificados como negros representan los Falsos Negativos. Por otro lado, los pixeles negros de la imagen segmentada manualmente que fueron clasificados como negros por los algoritmos representan los Verdaderos Negativos, y los que fueron clasificados como blancos representan los Falsos Positivos.

En la Figura 15 se ilustra esto de la siguiente manera. Los círculos blancos (a la derecha) representan los píxeles blancos de la imagen segmentada manualmente, y los círculos negros (a la izquierda) representan los pixeles negros de la imagen segmentada manualmente. Los círculos ubicados dentro del óvalo representan los pixeles clasificados

correctamente por los algoritmos, mientras los que están fuera representan los que fueron clasificados erróneamente.

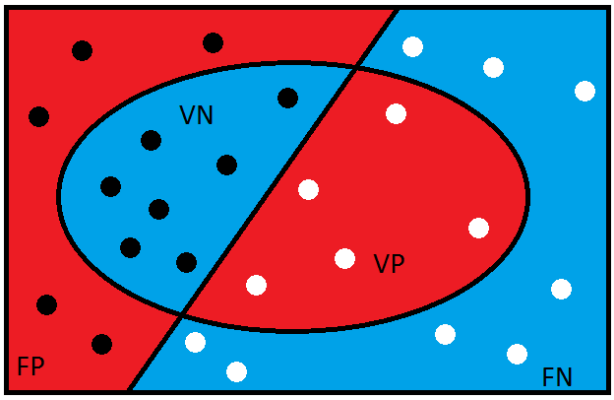

Figura 15: Ilustración de la detección basada en la métrica F1-score aplicada a imágenes. Los círculos blancos (a la derecha) representan los píxeles blancos de la imagen segmentada manualmente, y los círculos negros (a la izquierda) representan los pixeles negros de la imagen segmentada manualmente. Los círculos ubicados dentro del óvalo representan los pixeles clasificados correctamente por el algoritmo, mientras los que están afuera representan los que fueron clasificados erróneamente.

En la evaluación se calculó para cada una de las imágenes el F1-score entre la imagen segmentada por cada uno de los cuatro algoritmos y la correspondiente imagen segmentada manualmente por un humano. La Tabla 3 muestra los promedios de los F1-score obtenidos para cada categoría de imágenes.

Tabla 3: Promedios de los F1-score obtenidos para cada categoría de imágenes

Categoría	Propuesto	Kittler	Kurita	Otsu
Imágenes sintéticas	0,947	0,950	0,984	0,974
Gráficos por computador	0,960	0,882	0,963	0,940
Imagenología	0,952	0,892	0,965	0,960
Letras	0,922	0,912	0,914	0,936
Fotografías	0,853	0,815	0,783	0,798
Objetos pequeños	0,999	0,997	0,999	0,670
Bajo contraste	0,785	0,631	0,614	0,690
PROMEDIO	0,92	0,87	0,89	0,85

En la Tabla 3 se puede observar que el promedio total del F1-score obtenido por el algoritmo propuesto fue ligeramente mejor que el obtenido por los algoritmos de Kittler, Kurita y Otsu. Debido a esto se concluye que el algoritmo propuesto es ligeramente más preciso que los otros tres algoritmos. Además, se puede observar que en la categoría de imágenes de bajo contraste el algoritmo propuesto tuvo un mucho mejor desempeño que los otros tres algoritmos, por lo que se considera el algoritmo propuesto la mejor opción para aplicaciones en las que se desea segmentar imágenes de bajo contraste. Por ejemplo, en la Figura 7 se puede observar cómo el algoritmo propuesto se desempeñó mucho mejor que los otros tres en una imagen de bajo contraste. Finalmente, la razón por la cual el algoritmo propuesto no ganó en todas las categorías puede deberse a que las segmentaciones manuales no son perfectas debido a que fueron hechas por un humano (componente subjetivo).

5.4.3 Evaluación subjetiva por parte de 8 humanos

La evaluación subjetiva se hizo usando el criterio de 8 humanos de distintas edades, géneros, carreras, etc. Estos evaluaron de manera independiente la calidad de las segmentaciones de las 132 imágenes hechas por los cuatro algoritmos con una nota de 0 a 100, donde 0 representa una segmentación muy mala y 100 representa una segmentación excelente. Los humanos usaron como referencia las segmentaciones manuales de cada imagen para poder evaluar la calidad de las segmentaciones hechas por los algoritmos. La Tabla 4 muestra los promedios de las evaluaciones de los humanos para cada categoría de imágenes.

Tabla 4: Promedios de las evaluaciones de los humanos para cada categoría de imágenes

Categoría	Propuesto	Kittler	Kurita	Otsu
Imágenes sintéticas	90	86	88	88
Gráficos por comp.	87	88	81	78
Imagenología	79	77	86	84
Letras	77	77	75	77
Fotografías	78	77	76	76
Objetos pequeños	94	90	96	61
Bajo contraste	65	49	42	53
PROMEDIO	81	78	78	74

De la Tabla 4 se deduce la misma conclusión obtenida en la evaluación del F1-score: que el algoritmo propuesto es ligeramente más preciso que los otros tres. Además, se observa cómo los humanos calificaron al algoritmo propuesto como el mejor para imágenes de bajo contraste.

5.5 Tiempos de procesamiento

En la Tabla 5 se muestran los tiempos de procesamiento promedio (medidos en milisegundos) de los cuatro algoritmos para cada una de las categorías de imágenes. Es importante reiterar que los tiempos de procesamiento del algoritmo propuesto son los obtenidos con el método basado en el algoritmo *Downhill Simplex*.

Tabla 5: Tiempos de procesamiento promedio para cada categoría de imágenes (medidos en milisegundos)

Categoría	Propuesto	Kittler	Kurita	Otsu
Imágenes sintéticas	9	3	3	3
Gráficos por computador	12	4	3	3
Imagenología	9	3	3	2
Letras	7	2	2	2
Fotografías	12	4	4	4
Objetos pequeños	6	3	2	3
Bajo contraste	7	3	3	2
PROMEDIO	9	3	3	3

Se puede observar en la Tabla 5 que el algoritmo propuesto requiere en promedio tres veces más tiempo de procesamiento que los algoritmos de Kittler, Kurita y Otsu. Esto se debe a que el algoritmo propuesto usa un modelo con mayor cantidad de parámetros, lo que hace que la estimación de estos y la evaluación de ellos en la función de verosimilitud consuma más tiempo en comparación con los otros tres que usan modelos más sencillos. Los tiempos de procesamiento son tan bajos (ámbito de los milisegundos) que el factor de tres no hace una diferencia significativa a la hora de escoger cuál algoritmo usar para determinada aplicación.

CAPÍTULO 6: CONCLUSIONES

En este trabajo se propuso un algoritmo de umbralización por máxima verosimilitud más preciso que sus similares descritos en la literatura profesional, cuya función de verosimilitud fue derivada de un modelo de distribución de probabilidad que hace uso de distribuciones gamma de cuatro parámetros. La función de verosimilitud obtenida es genérica, ya que a partir de ella las funciones de verosimilitud de los algoritmos basados en los modelos Gaussianos pueden ser derivadas.

Las funciones de verosimilitud de los algoritmos de Kittler, Kurita y Otsu, así como la del algoritmo propuesto, fueron derivadas a partir de una función de verosimilitud base obtenida en este trabajo. Este es un resultado teórico muy importante de esta investigación, ya que la función de verosimilitud base obtenida es una versión unificada de las dos funciones de verosimilitud base explicadas vagamente en [14].

El algoritmo propuesto, así como los de Kittler, Kurita y Otsu, se aplicaron a las 132 imágenes de la biblioteca de imágenes llamada *IPCV-LAB Image Data Base*. El promedio total del F1-score obtenido por el algoritmo propuesto fue 5%, 3% y 7% mejor al de los algoritmos de Kittler, Kurita y Otsu, respectivamente. Basados en estos resultados, se concluye que el algoritmo propuesto es en promedio ligeramente más preciso que los otros tres cuando se aplica a imágenes de cualquier tipo. Además, para imágenes de bajo contraste, el promedio total del F1-score obtenido por el algoritmo propuesto fue 15%, 17% y 10% mejor al de los algoritmos de Kittler, Kurita y Otsu, respectivamente. De aquí se concluye que el algoritmo propuesto es sin duda la mejor opción para segmentar imágenes de bajo contraste. Estas últimas dos conclusiones fueron corroboradas en la evaluación subjetiva de las segmentaciones por parte de ocho humanos, ya que se obtuvieron resultados muy similares.

Se evaluó también la calidad en la que las $PDF^{S,op}$ obtenidas por los algoritmos propuestos aproximaron cada PDF^{I} de las imágenes probadas mediante el cálculo del MSE. El algoritmo propuesto tuvo un MSE 26%, 53% y 78% mejor que los algoritmos de Kittler,

Kurita y Otsu, respectivamente. Estos resultados confirmaron contundentemente la hipótesis de que el modelo propuesto es capaz de aproximar de mejor manera comportamientos estadísticos más complejos y variados presentes en las PDF^I, lo cual resulta en un incremento en la precisión de la segmentación.

Adicionalmente, a partir de los resultados experimentales, se concluyó que una de la razones por la cuales el modelo propuesto es más apto que el Gaussiano no es solo porque tiene más parámetros, sino porque la distribución gamma se extiende hasta el infinito solo en un sentido, y no en ambos como la distribución Gaussiana. Esto hace que su forma se pueda parecer más a la PDF^I, la cual no se extiende al infinito en ningún sentido, sino que se encuentra circunscrita en el rango de 0 a 255.

El costo de usar un modelo más complejo se reflejó en que el tiempo de procesamiento promedio del algoritmo propuesto fue tres veces mayor al de los otros tres algoritmos. Sin embargo, debido a que los tiempos de procesamiento son muy bajos, el tiempo de procesamiento extra no justifica descartar al algoritmo propuesto. Una posible mejora al algoritmo propuesto sería implementarlo con un método de optimización tal que el tiempo de procesamiento promedio por imagen sea igual que el de los otros tres algoritmos.

REFERENCIAS

Artículos y libros

1. Alison, J. y Boukerrui, D. (2006). Ultrasound Image Segmentation: A Survey. IEEE Transactions on Medical Imaging, Vol. 25, No. 8 (2006), pp. 985-1010.

2. Bazi, Y., Bruzzone, L., y Melgani, F. (2006). Image thresholding based on the EM algorithm and the generalized Gaussian distribution. Pattern Recognition 40 (2007) 619 – 634.

3. Cheng, R. C. H. e Iles, T. C. (1990). Embedded Models in Three-Parameter Distributions and Their Estimation. Journal of the Royal Statistical Society. Series B (Methodological), Vol. 52, No.1 (1990), pp. 135-149.

4. Choi, S. C. y Wette, R. (1969). Maximum Likelihood Estimation of the Parameters of the Gamma Distribution and Their Bias. Technometrics, Vol. 11, No. 4 (Nov., 1969), pp. 683-690.

5. El Zaart, A. y Ziou, D. (2010). Statistical modelling of multimodal SAR images. International Journal of Remote Sensing, 28: 10, 2277 — 2294.

6. Goutte, C. y Gaussier, E. (2005). A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation. Proceedings of the European Colloquium on IR Research (ECIR'05), LLNCS 3108 (Springer), pp.345-359, 2005.

7. Hazewinkel y Michiel. Stirling formula. Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4.

8. Hirose, H. (1994). Computational Statistics & Data Analysis 20 (1995) 343-354.

9. Jiulun, F. (1998). Notes on Poisson distribution-based minimum error thresholding. Pattern Recognition Letters 19 1998 425–431.

10. Jiulun, F. y Winxin, X. (1997). Minimun Error Thresholding: A note. Pattern Recognition Letters 18 (1997) 705-709.

11. Kappenman, R. F. (1985). Estimation for the three-parameter Weibull, lognormal, and Gamma distributions. Computational Statistics & Data Analysis 3 (1985) 11-23.

12. Kittler, J e Illingworth, J (1984). Minimum Error Thresholding. Pattern Recognition Society, Vol 19, No. 1.

13. Kundu, D. y Gupta, R. D. (2007). A convenient way of generating gamma random variables using generalized exponential distribution. Computational Statistics & Data Analysis 51 (2007) 2796 – 2802.

14. Kurita, T., Otsu, N., y Abdelmalek, N. (1991). Maximum Likelihood Thresholding Based on Population mixture models. Pattern Recognition Society, Vol 25, No. 10.

15. Le Cam, L. (1990). Maximum Likelihood: An Introduction. Revue Internationale de Statistique, Vol. 58, No. 2 (Aug., 1990), pp. 153-171.

16. Martinez, G. (2005). Criterion for Automatic Selection of the Most Suitable Maximum-Likelihood Thresholding Algorithm for Extracting Object from their Background in a still image. Conference on Machine Vision Applications, 13-10.

17. Mendel, J. M.. Lessons in Estimation Theory for Signal Processing, Communications, and Control. Prentice Hall.

18. Myung, I. (2003). Tutorial on maximum likelihood estimation. Journal of Mathematical Psychology 47 (2003) 90–100.

19. Norden, H. (1972). A Survey of Maximum Likelihood Estimation. Revue Internationale de Statistique, Vol. 40, No. 3 (Dec., 1972), pp. 329-354.

20. Otsu, N. (1977). A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics, Vol SMS-9, No. 1.

21. Pal, N. R. y Pal, S. K. (1993). A Review on Image Segmentation Techniques. Pattern Recognition, vol 26, No. 9.

22. Press, W. H., Vetterling, W. T., Teukolsky, S. A. and Flannery, B. P. (2002). Numerical Recipes in C, The Art of Scientic Computing, Second Edition, Cambridge University Press (2002).

23. Raut, S., Raghuvanshi, M., Dharaskar, R. y Raut, A. (2009). Image Segmentation – A State-Of-Art Survey for Prediction. International Conference on Advanced Computer Control, 2009.

24. Sezgin, M. y Sankur, B. (2004). Survey over Image Thresholding Techniques and Quantitative Performance Evaluation. Journal of Electronic Imaging 13(1), 146-165, January 2004.

25. Solomon, F. (1946). Probability and Stochastic Processes. Prentice Hall, ISBN 0-13-711961-5.

26. Tao, Z., Tagare, H., y Beaty, J. (2006). Evaluation of Four Probability Distribution Models for Speckle in Clinical Cardiac Ultrasound Images. IEEE Transactions On Medical Imaging, Vol. 25, No. 11, November 2006.

27. Udupa, J. K., LeBlanc, V. R., Zhuge, Y., Imielinska, C., Schmidt, H., Currie, L. M., Hirsch, B. E., y Woodburn, J. (2006). A framework for evaluating image segmentation algorithms. Computerized Medical Imaging and Graphics 30 (2006) 75–87.

28. Zhang, Y. J. (1996). A Survey on evaluation methods for image segmentation. Pattern Recognition, Vol. 29, No. 8, pp. 1335-1346, 1996.

Páginas de Internet

29. Xaraxone.com, accesada el 27-7-2013.

http://www.xaraxone.com/webxealot/workbook02/page_3.htm

30. MIT Edu, accesada el 27-7-2013.

http://web.mit.edu/newsoffice/2012/explained-sigma-0209.html

31. Wikipedia.com, accesada el 27-7-2013.

http://en.m.wikipedia.org/wiki/File:Inverse_gamma_pdf.png

APÉNDICE

A. Estimación por máxima verosimilitud de los pesos de la función de verosimilitud base dado el umbral

Basado en el método de máxima verosimilitud, los pesos óptimos (MLEs) que maximizan la función de verosimilitud base de la Ecuación (3.9) dado el umbral th son los que maximizan su primera parte

$$f(w_0, w_1) = c_{0,th} * N * \ln(w_0) + c_{1,th} * N * \ln(w_1)$$

donde $c_{0,th}$, $c_{1,th}$ y N son constantes, $c_{0,th} + c_{1,th} = 1$, y $w_0 + w_1 = 1$. Eliminando la N por ser una constante que multiplica ambos términos, poniendo $c_{1,th}$ en términos de $c_{0,th}$, y w_1 en términos de w_0 se obtiene

$$f(w_0) = c_{0,th} * ln(w_0) + \left(1 - c_{0,th}\right) * ln(1 - w_0)$$

Para encontrar el máximo, se inserta la derivada parcial respecto a w_0 y se iguala a 0

$$0 = \frac{\partial}{\partial w_0}\left[c_{0,th} * ln(w_0) + \left(1 - c_{0,th}\right) * ln(1 - w_0)\right]$$

Aplicando la derivada

$$0 = c_{0,th} * \frac{1}{w_0} + \left(1 - c_{0,th}\right) * \frac{-1}{1 - w_0}$$

Pasando el segundo término presente en el lado derecho hacia el lado izquierdo

$$\left(1 - c_{0,th}\right) * \frac{1}{1 - w_0} = c_{0,th} * \frac{1}{w_0}$$

Pasando los divisores a sus lados opuestos

$$\left(1 - c_{0,th}\right) * (w_0) = c_{0,th} * (1 - w_0)$$

Aplicando propiedad distributiva

$$w_0 - c_{0,th} * w_0 = c_{0,th} - c_{0,th} * w_0$$

Se obtiene

$$w_0 = c_{0,th}$$

y por lo tanto

$$w_1 = c_{1,th}$$

Es importante mencionar que a pesar de que los MLEs de los pesos suman 1, la PDF^S no necesariamente tendrá un área bajo la curva igual a 1. Esto se debe a que la PDF_0^S es ventaneada (recortada) de 0 a th, y a que la PDF_1^S es ventaneada (recortada) de $th + 1$ a 255. Sin embargo, el hecho de que el área bajo la curva no sea igual a 1 puede ser ignorado para propósitos de implementación, ya que las áreas recortadas suelen ser muy pequeñas. Inclusive, se probó hacer el reajuste y dio resultados iguales o ligeramente peores.