



Universidad Michoacana de San Nicolás de Hidalgo

División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica

Binarización de imágenes de texto con iluminación no uniforme

TESIS

Que para obtener el grado de
Maestro en Ciencias en Ingeniería Eléctrica

presenta
Cristina Reyes Zúñiga

Dr. Félix Calderón Solorio
Director de Tesis

Morelia, Michoacán a Agosto del año 2021

Resumen

La binarización de imágenes es una técnica utilizada en el procesamiento de imágenes que convierte una imagen en tono de gris a una imagen de dos colores representados por blanco y negro. La binarización de imágenes presenta problemas en imágenes con iluminación no uniforme debido a que se generan áreas claras y oscuras en la imagen. En el estado del arte se han propuesto distintos algoritmos para binarizar imágenes los cuales emplean umbral global o local. Para abordar el problema de iluminación no uniforme se utilizan algoritmos que calculan umbral local; entre ellos están los que emplean ventanas para delimitar el área sobre la que se calculará el umbral, sin embargo el tamaño de las ventanas que se utiliza es constante o fijo. En esta tesis proponemos un algoritmo para calcular el tamaño óptimo de las ventanas que estarán sujetas a respetar los bordes donde ocurre el cambio de iluminación en la imagen. El algoritmo que proponemos consiste en obtener un mapa de decisión que identifica el área clara y oscura de la imagen con el cual calculamos un mapa de bordes del área clara y oscura con el objetivo de crear ventanas que contengan solamente píxeles del área clara o del área oscura. Las ventanas serán pequeñas cerca de los bordes donde ocurre el cambio de iluminación y grandes donde no existen bordes, debido a que si se mezclan píxeles del área clara y oscura el criterio de decisión se ve afectado. Presentamos dos algoritmos para binarización de imágenes en los cuales empleamos el algoritmo que calcula ventanas de tamaño óptimo. Mostramos resultados de forma cualitativa y cuantitativa de nuestros algoritmos propuestos para binarización de imágenes y de algoritmos del estado del arte.

Palabras clave: **binarización de imágenes, umbral local, ventana adaptable, iluminación no uniforme, imágenes integrales, búsqueda binaria.**

Abstract

Image binarization is a technique to convert a grayscale image to black and white image. Non-uniform illumination is a problem in image binarization because it generates light and dark areas in the image. The algorithms of state of the art to image binarization calculate a local or global threshold. To solve the non-uniform illumination problem algorithms employ local threshold, some of them employ windows to identify area and calculate the local threshold. In this document we present an algorithm to calculate optimal size windows which are subject to respect the illumination change in the image. Our algorithm gets a decision map to identify dark and light areas in original image, after it compute a borders map to recognize pixels that belong area where occurs illumination change in the image. The algorithm computes small windows close to illumination change areas and big windows far to avoid mixing pixels of the light area and dark area. We create two algorithms to image binarization, and they use optimal size windows. We present qualitative and quantitative results of our algorithms and state of the art algorithms.

Keywords: **image binarization, local threshold, adaptive window, non-uniform illumination, binary search.**

Contenido

Resumen	III
Abstract	V
Contenido	VII
Lista de Figuras	IX
Lista de Tablas	XI
Lista de Algoritmos	XIII
Lista de Símbolos	XV
Lista de Abreviaturas	XVII
1. Introducción	1
1.1. Planteamiento del problema	2
1.2. Marco teórico	8
1.2.1. Métricas de calidad	9
1.3. Estado del arte	11
1.4. Objetivos de la tesis	13
1.4.1. Objetivo general	13
1.4.2. Objetivos particulares	14
1.5. Justificación	14
1.6. Hipótesis	14
1.7. Variables de estudio	14
1.8. Preguntas de Investigación	15
1.9. Metodología	15
1.10. Descripción de Capítulos	15
2. Trabajo relacionado	17
2.1. Método de Otsu	17
2.2. Método de k-medias	20
2.3. Método de Bradley y Roth	22
2.4. Algoritmo CLI-S para fusión de imágenes multi-foco	26
2.5. Algoritmo para binarización de imágenes propuesto por Edgar Molina	28
2.6. Conclusión	31
3. Algoritmos propuestos	33
3.1. Cálculo eficiente del tamaño de la ventana	42
3.2. Binarización de Imágenes con Promedios Ponderados	49
3.3. Binarización de Imágenes con Ventanas Adaptables	56
3.4. Conclusión	64

4. Resultados	67
4.1. Imágenes con iluminación no uniforme real y sintética	67
4.2. Resultados cuantitativos mostrados mediante gráficas	69
4.3. Resultados utilizando diferentes parámetros en los algoritmos	80
4.4. Resultados de forma cualitativa	83
4.5. Conclusiones	91
5. Conclusiones	93
5.1. Conclusiones Generales	93
5.2. Trabajos futuros	94
Referencias	95
Glosario	96

Lista de Figuras

1.1. Imagen de texto en tono de gris binarizada utilizando (1.1)	2
1.2. Imágenes con iluminación no uniforme	3
1.3. Imágenes de texto escrito a mano con degradaciones en el fondo de la hoja	3
1.4. Imagen binarizada eligiendo un umbral basado en su distribución de probabilidad	4
1.5. Imagen con iluminación no uniforme binarizada eligiendo un umbral a la mitad de las modas de su distribución de probabilidad	5
1.6. Imagen dividida en áreas con el mismo tipo de iluminación	6
1.7. Distribuciones de probabilidad de áreas de las imágenes de la Figura 1.6	7
1.8. Imágenes binarizada por áreas con el mismo tipo de iluminación	8
1.9. Vecindario delimitado por una ventana	9
2.1. Suma de tonos de gris de píxeles delimitados por una ventana utilizando imágenes integrales	24
2.2. Componente de luminancia de I_0 utilizando un kernel morfológico de tamaño $C = 20$	29
2.3. Luminancia inversa de la Figura 2.3 a)	30
3.1. A la izquierda imagen I_1 y a la derecha su distribución de probabilidad	34
3.2. A la izquierda imagen I_1 binarizada con Otsu, 1979 en medio binarizada con Bradley and Roth, 2007 y a la derecha con Molina et al., 2018	35
3.3. Exactitud obtenida al binarizar I_1 con Bradley and Roth, 2007 y Molina et al., 2018 utilizando diferentes valores de w	36
3.4. A la izquierda imagen I_2 y a la derecha su distribución de probabilidad	37
3.5. A la izquierda imagen I_2 binarizada con Otsu, 1979, en medio binarizada con Bradley and Roth, 2007 y a la derecha con Molina et al., 2018	38
3.6. Exactitud obtenida al binarizar I_2 con Bradley and Roth, 2007 y Molina et al., 2018 utilizando diferentes valores de w	39
3.7. A la izquierda imagen I_3 y a la derecha su distribución de probabilidad	40
3.8. A la izquierda imagen I_3 binarizada el algoritmo de Otsu, 1979, en medio binarizada con Bradley and Roth, 2007 y a la izquierda con Molina et al., 2018	40
3.9. Exactitud obtenida al binarizar I_3 con los algoritmos Bradley and Roth, 2007 y Molina et al., 2018	41
3.10. Imagen I_3 binarizada con (3.7)	43
3.11. Binarización de I_3 utilizando (3.9) con diferentes valores de w	44
3.12. Gráfica de la exactitud de I_3 binarizada con (3.9) utilizando diferentes valores de w	45
3.13. A la izquierda se muestra el mapa de bordes de la imagen I_3 , en el centro el tamaño de las ventanas alrededor de los bordes y a la derecha la imagen binarizada	49
3.14. Imagen con iluminación no uniforme, su componente de iluminación y su iluminación inversa	50
3.15. Imagen inversamente iluminada de la Figura 3.14 a)	51

3.16. Ventanas calculadas con VO, mapa de decisión y la imagen fusionada	52
3.17. Diagrama con los procesos de BIPP y las imágenes que va calculando en cada uno de ellos	53
3.18. Ejemplo que muestra una imagen binarizada con BIPP y la misma imagen binarizada con Molina et al., 2018	56
3.19. A la izquierda imagen con texto y la derecha su distribución de probabilidad	57
3.20. Ejemplo de imagen con iluminación no uniforme y su mapa de decisión	58
3.21. Mapa de decisión y ventanas calculadas con VO de la Figura 3.7	59
3.22. Diagrama con los proceso que realiza BIVA y la imagen que obtiene en cada uno de ellos	61
3.23. Imagen con iluminación no uniforme binarizada con BIVA y Bradley and Roth, 2007	64
4.1. Imagen empleada para aplicar distintos patrones de iluminación no uniforme	67
4.2. Imágenes con diferentes patrones de iluminación sintéticas no uniforme	68
4.3. Imágenes con iluminación no uniforme generada a partir de iluminación real	69
4.4. Gráficas con resultados utilizando AC y F-score de imágenes con iluminación sintética binarizadas	72
4.5. Gráficas con resultados utilizando PSNR y NRM de imágenes con iluminación sintética binarizadas	73
4.6. Gráficas de barras con promedios donde Br=Bradle and Roth, 2007, Ot=Otsu, 1079, Molina + Br=Molina et al., 2018 utilizando Bradle and Roth, 2007 para binarizar, Molina+Otsu= Molina et al., 2018 utilizando Otsu, 1079	74
4.7. Gráficas que muestran la exactitud de las imágenes con iluminación real binarizadas con los diferentes algoritmos	77
4.8. Gráficas con resultado de las imágenes con iluminación real binarizadas con los diferentes algoritmos	78
4.9. Gráficas de barras con promedios donde Br=Bradle and Roth, 2007, Ot=Otsu, 1079, Molina + Br=Molina et al., 2018 utilizando Bradle and Roth, 2007 para binarizar, Molina+Otsu= Molina et al., 2018 utilizando Otsu, 1079	79
4.10. A la izquierda imagen con texto y a la derecha la misma imagen con iluminación no uniforme generada con una función de Gauss	80
4.11. Imágenes con iluminación no uniforme	84
4.12. Binarización de la Figura 4.11 a) con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA	85
4.13. Binarización de la Figura 4.11 b) con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA	87
4.14. Binarización de la Figura 4.11 c) con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA	88
4.15. Binarización de la Figura 4.11 d) binarizada con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA	89
4.16. Binarización de la Figura 4.11 e) con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA	90
4.17. Binarización de la Figura 4.11 f) binarizada con Molina et al., 2018, BIPP, Bradley and Roth, 2017 y BIVA	91

Lista de Tablas

4.1. Exactitud de la Figura 4.10 b) binarizada con Bradley and Roth, 2007 utilizando diferentes valores de τ y w	81
4.2. Exactitud de la Figura 4.10 b) binarizada con BIVA utilizando diferentes valores de τ y w_{max}	82
4.3. Exactitud de la Figura 4.10 b) binarizada con Molina et al., 2018 utilizando diferentes valores C y w	82
4.4. Exactitud de la Figura 4.10 b) binarizada con BIPP utilizando diferentes valores en los parámetros τ y w_{max}	83

Lista de Algoritmos

1.	Otsu(I)	20
2.	k-medias(I, N_{iter})	22
3.	BradleyRoth(I, τ, w)	25
4.	CLI-S(I_0, I_1, w)	28
5.	Algoritmo de Edgar Molina(I_0, C)	31
6.	VO (P, T, w_{max})	48
7.	BIPP(I_0, C, w_{max}, T, τ)	55
8.	BIVA(I, w_{max}, T, τ)	63

Lista de Símbolos

I_B	Imagen binarizada.
p_i	Probabilidad de nivel de gris en la imagen.
n_i	Total de píxeles de tono i .
ω_0	Probabilidad de la clase 0.
ω_1	Probabilidad de la clase 1.
μ_0	Media de la clase 0.
μ_1	Media de clase 1.
σ_0^2	varianza de clase 0.
σ_1^2	varianza de clase 1.
σ_W^2	varianza intra-clase.
σ_B^2	varianza inter-clase.
μ_T	Media de la imagen.
σ_T^2	varianza de la imagen.
N_r	Número de renglones.
N_c	Número de columnas.
I	Imagen en escala de grises que se quiere binarizar.
$I[r, c]$	Píxel de la imagen I ubicado en las coordenadas $[r, c]$.
u	valor del umbral para clasificar como blanco o negro no o más píxeles.
m_0	Moda de la clase de las letras.
m_1	Moda de la clase de del fondo.
I_r	Imagen de referencia, correctamente binarizada.
D	Distribución de probabilidad.
H	Imagen integral.
u_{rc}	Umbral para binarizar el píxel en las coordenadas $[r, c]$.
τ	Parámetro para calcular el porcentaje que se tomara en cuenta del umbral u_{rc} .
$w[r, c]$	Parámetro para determinar el tamaño de ventana $(2w[r, c] + 1)$ en las coordenadas $[r, c]$.
h	kernel laplaciano.
F_0	Función de calidad de la clase de las letras.
F_1	Función de calidad de la clase del fondo.
P	Mapa de decisión.
ΔF	Matriz de la resta de F_0 con F_1 .
I_f	Imagen fusionada.
I_1	Imagen con iluminación no uniforme.
I_2	Imagen inversamente iluminada a I_1 .
L_1	Componente de luminancia de la imagen I_1 .
R_1	Componente de reflectancia de la imagen I_1 .
L_2	Componente de luminancia de la imagen I_2 .
\bullet	Operador morfológico de cerradura.
\oplus	Operador de dilatación.

\ominus	Operador de erosión.
E	Porcentaje del error en la imagen binarizada.
AC	Porcentaje de aciertos en la imagen binarizada.
B	Mapa de bordes del mapa de decisión.
T	Máximo número de bordes que puede contener una ventana.

Lista de Abreviaturas

BIPP Binarización de Imágenes con Promedios Ponderados

BIVA Binarización de Imágenes con Ventanas Adaptables

CLI-S Combinación Lineal de Imágenes Simple

NRM Negative Rate Metric

PSNR Peak Signal to Noise Ratio

RGB Red Green Blue

VO Ventana Óptima

Capítulo 1

Introducción

La binarización de imágenes es una técnica con aplicaciones en diferentes áreas del procesamiento de imágenes como en identificación de objetos de interés en imágenes médicas, procesamiento de vídeos, detección de rostros [Singh and Hruaia, 2015](#), reconocimiento de huellas dactilares, reconocimiento de iris, entre otros [Chaki et al., 2014](#). En segmentación de imágenes se utiliza para separar el área de interés del resto de la imagen [LaTorre et al., 2013](#). En reconocimiento óptico de caracteres la binarización de imágenes se emplea para separar el texto del fondo de la imagen con el objetivo de obtener una imagen con fondo limpio sin manchas, sombras o exceso de luz. Por lo que, de una correcta binarización depende el rendimiento de la segmentación, el reconocimiento óptico de caracteres, entre otros [Cherala and Rege, 2008](#).

La imagen digital es una representación bidimensional definida como una matriz I con N_r número de renglones y N_c número de columnas. La matriz esta conformada por píxeles donde un píxel es representado mediante $I[r, c]$ en el que r representa la posición en los renglones y c la posición en las columnas. En una imagen en tono de gris el valor de un píxel contiene tonos que van de $[0 - 255]$ correspondiente a la escala de grises. El proceso de binarizar imágenes consiste en convertir una imagen en tono de gris a una imagen en blanco y negro comparando cada píxel con un valor llamado umbral para determinar si será clasificado como blanco o negro. Dada una imagen I en tono de gris y un umbral u , la binarización se puede realizar haciendo que los píxeles $I[r, c]$ con valor menor a u sean considerados como 0 que representa el color negro, de lo contrario como 255 que representa el color blanco tal como en (1.1), donde I_B representa la imagen binarizada.

$$I_B[r, c] = \begin{cases} 255 & I[r, c] > u \\ 0 & \text{si no} \end{cases} \quad (1.1)$$

La Figura 1.1 a) muestra una imagen de texto en tono de gris. La Figura 1.1 b) muestra la imagen binarizada de la Figura 1.1 a) utilizando (1.1) la cual es aceptable para realizar un post-procesamiento. Sin embargo, para obtener una correcta binarización al utilizar (1.1) es necesario que la imagen no tenga manchas o cambios de tono debido a la iluminación no uniforme.

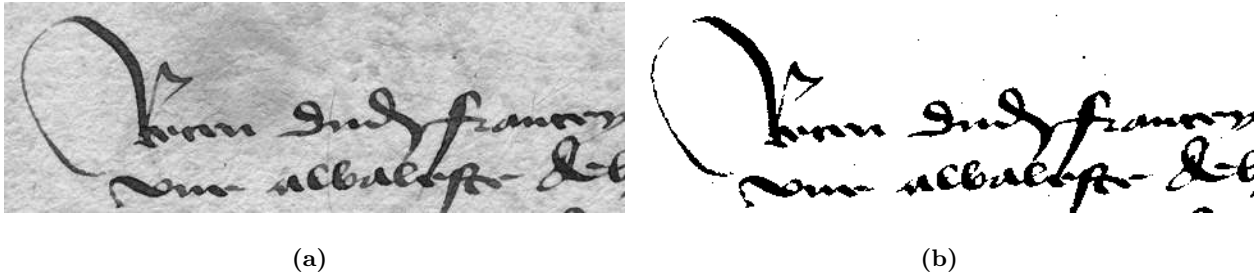


Figura 1.1: Imagen de texto en tono de gris binarizada utilizando (1.1)

1.1. Planteamiento del problema

Debido al avance de la tecnología es común capturar imágenes con cámaras de diferentes dispositivos como celulares, tabletas, entre otros. La calidad de las imágenes capturadas con dichos dispositivos muchas veces depende del lugar donde se capturó la imagen o de la hora del día, por lo que pueden presentar iluminación no uniforme lo cual genera imágenes con áreas claras y oscuras, bajo contraste o sombras [Chen et al., 2012](#). La Figura 1.2 muestra tres ejemplos de imágenes que presentan iluminación no uniforme. La Figura 1.2 a) muestra la imagen de una carta donde la parte superior derecha está bastante iluminada y el resto de la imagen se ve oscura. La Figura 1.2 b) muestra una serie de marcadores que se utilizan en realidad aumentada que son detectados por algún dispositivo, dicha imagen tiene un área clara y una oscura debido a la iluminación no uniforme. La Figura 1.2 c) muestra una imagen de texto con iluminación no uniforme generada con la herramienta Photo Shop.

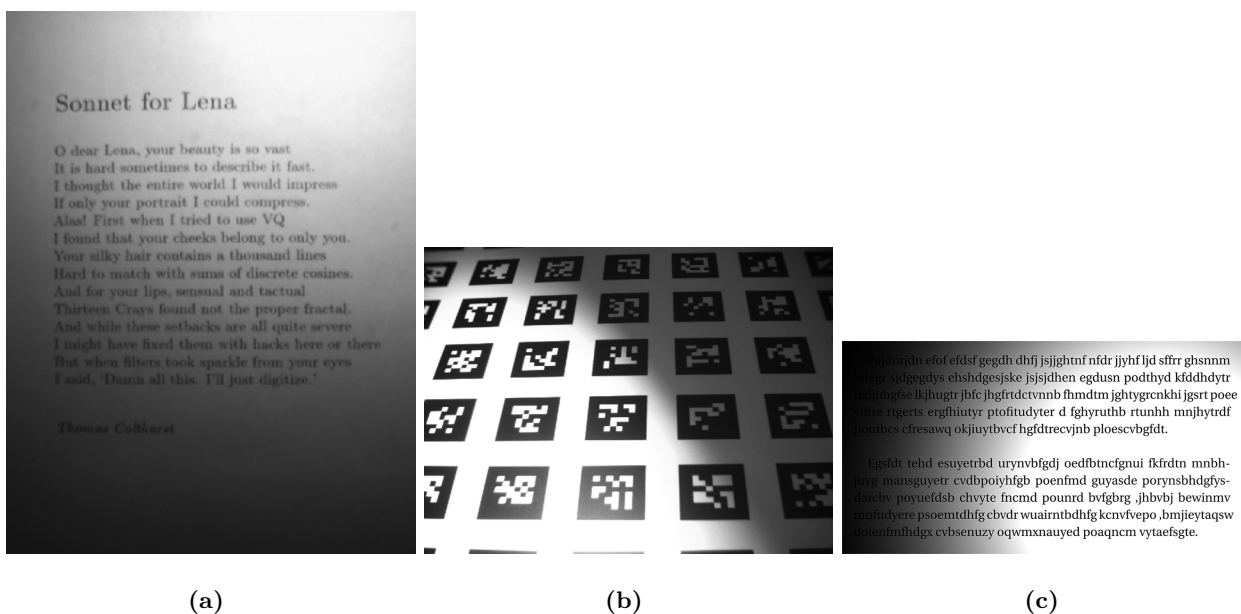


Figura 1.2: Imágenes con iluminación no uniforme

En el proceso de digitalización de libros las hojas pueden presentar degradaciones que ocurren debido a varias razones que van desde el tipo de fuente de adquisición hasta las condiciones ambientales en las que se encontraban [Jana et al., 2017](#). La Figura 1.3 muestra imágenes de texto escrito a mano que presentan manchas generadas en el fondo, lo cual afecta el proceso de binarización.

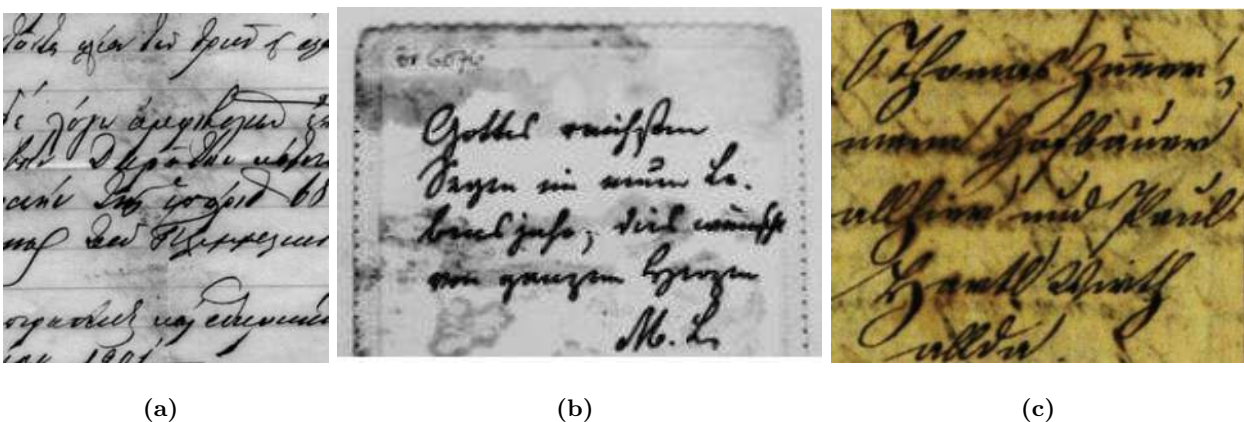
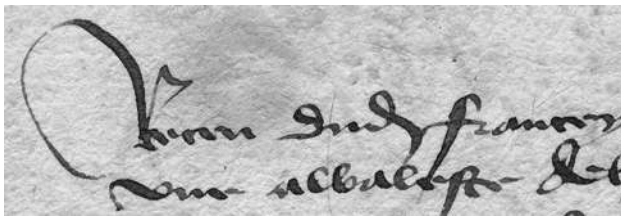


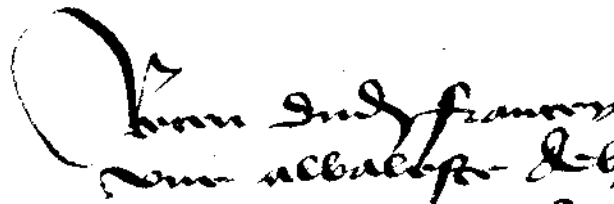
Figura 1.3: Imágenes de texto escrito a mano con degradaciones en el fondo de la hoja

Una herramienta muy utilizada en el procesamiento de imágenes es la distribución de probabilidad de la imagen, la cual asigna a cada tono de gris la probabilidad de ocurrencia. Al representar la distribución de probabilidad de la imagen en forma gráfica el eje horizontal muestra los diferentes tonos de gris y el eje vertical la probabilidad de cada tono de gris. En la distribución

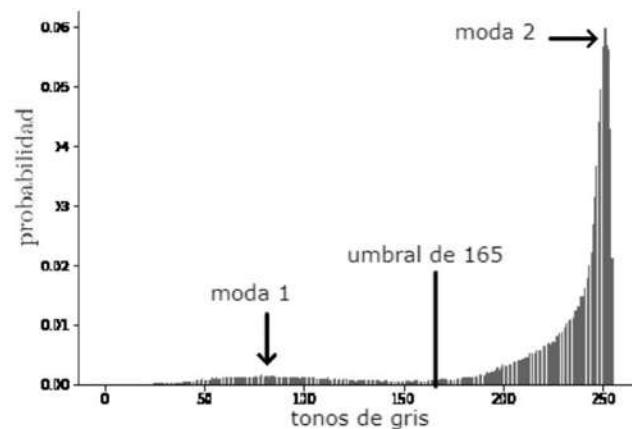
de probabilidad de la imagen se puede ver la dispersión, tendencia y forma de distribución de los tonos de gris. En una imagen de texto con iluminación uniforme la distribución de probabilidad generalmente presenta dos modas las cuales pertenecen a los píxeles del texto y del fondo. Entre las modas se genera un valle que hace que las clases del texto y fondo se vean separadas y definidas. La Figura 1.4 a) muestra una imagen con texto y la Figura 1.4 c) muestra su distribución de probabilidad en la que señalamos la moda de los píxeles que pertenecen al texto la cual corresponde al valor 75 (moda 1) y la moda de los píxeles que pertenecen al fondo la cual corresponde al valor 250 (moda 2). En la distribución de probabilidad de la Figura 1.4 c) las probabilidades de los tonos de gris con valor de 120 a 140 son muy bajas por lo que generan el valle que hace que las clases del texto y fondo se vean separadas. Por lo tanto la binarización de la Figura 1.4 a) con un umbral elegido entre 120 y 140 será adecuada, sin embargo cada valor generará una imagen diferente. Para binarizar la imagen de la Figura 1.4 a) se elige el umbral 165 que corresponde al valor a la mitad de las modas 75 y 250. En la Figura 1.4 b) se muestra la Figura 1.4 a) binarizada con el umbral que corresponde al tono de gris con valor 165 y vemos que la imagen ha sido bien binarizada debido a que el fondo de la imagen esta en color blanco y las letras en color negro.



(a) Imagen de texto en tono de grises



(b) Imagen binarizada utilizando el umbral 165



(c) Distribución de probabilidad de la imagen de la Figura 1.4 a)

Figura 1.4: Imagen binarizada eligiendo un umbral basado en su distribución de probabilidad

Sin embargo, es común que las imágenes que se van a binarizar presenten iluminación no uniforme, manchas, sombras o bajo contraste. La Figura 1.5 a) muestra una imagen con iluminación no uniforme con el mismo texto de la Figura 1.4 a). La Figura 1.5 b) muestra la distribución de probabilidad de la Figura 1.5 a) donde vemos que no existe un valle que separe las clases del texto y fondo, en dicha distribución de probabilidad señalamos las modas que predominan las cuales son el tono de gris con valor 150 (moda 1) y el tono de gris con valor 250 (moda 2). Sin embargo, la moda con valor 150 se ha generado por la iluminación no uniforme de la imagen por lo que no representa la clase de las letras ni del fondo. Si para binarizar la Figura 1.5 a) se utiliza el umbral 165 que se empleó para binarizar la imagen de la Figura 1.4 a) que está adecuadamente iluminada, se tendrá una imagen binarizada como la mostrada en la Figura 1.5 c) donde vemos que la parte derecha de la imagen que pertenece al fondo se clasificó como letras. Si se elige como umbral el valor 200 que se encuentra a la mitad de la moda 150 y la moda 250 de la distribución de probabilidad, el resultado será como el que se muestra en la Figura 1.5 d) en este caso la mayor parte del fondo fue clasificada como texto. Por lo tanto en imágenes con iluminación no uniforme resulta inadecuado utilizar la distribución de probabilidad para elegir un umbral que binarice la imagen.

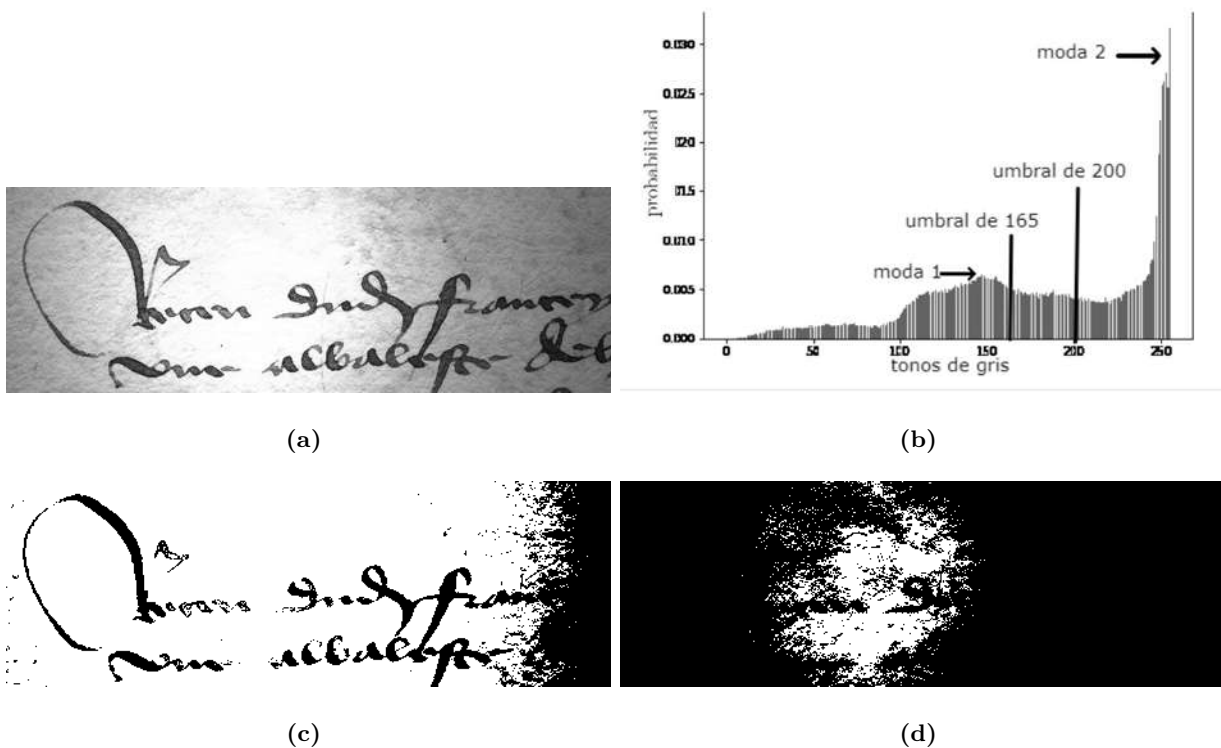


Figura 1.5: Imagen con iluminación no uniforme binarizada eligiendo un umbral a la mitad de las modas de su distribución de probabilidad

Para el caso de iluminación no uniforme de la Figura 1.5 a) otra alternativa es que antes de binarizar la imagen se divida en áreas en las que cada una tenga iluminación uniforme como se muestra en las imágenes de la Figura 1.6. La Figura 1.7 incisos a), b) y c) muestran la distribución de probabilidad de las imágenes de la Figura 1.6 incisos a), b) y c) respectivamente, en dichas distribuciones de probabilidad vemos que las clases que representan los píxeles del texto y fondo están definidas y separadas por un valle, a diferencia de la distribución de probabilidad de la Figura 1.5 b) de la imagen completa que presenta iluminación no uniforme. En la Figura 1.7 a) la moda 1 que representa los píxeles del texto corresponde al valor 75 y la moda 2 de los píxeles del fondo es el valor 175 por lo que se elige como umbral el valor a la mitad de las modas que corresponde al tono de gris con valor 122. Si binarizamos la Figura 1.6 a) con el umbral 122 se obtiene la Figura 1.8 a). En la Figura 1.7 b) la moda 1 es 75 y la moda 2 es 250 en este caso se elige el valor 165 como umbral (mitad de las modas). Si binarizamos la Figura 1.6 b) con el umbral 165 se obtiene la Figura 1.8 b). En la Figura 1.7 c) la moda 1 corresponde al valor 75 y la moda 2 al valor 135 por lo tanto se elige el tono de gris 88 como umbral. Si binarizamos la Figura 1.6 c) con el umbral 88 se obtiene la Figura 1.8 c). En la Figura 1.8 se pueden apreciar las 3 áreas binarizadas adecuadamente.

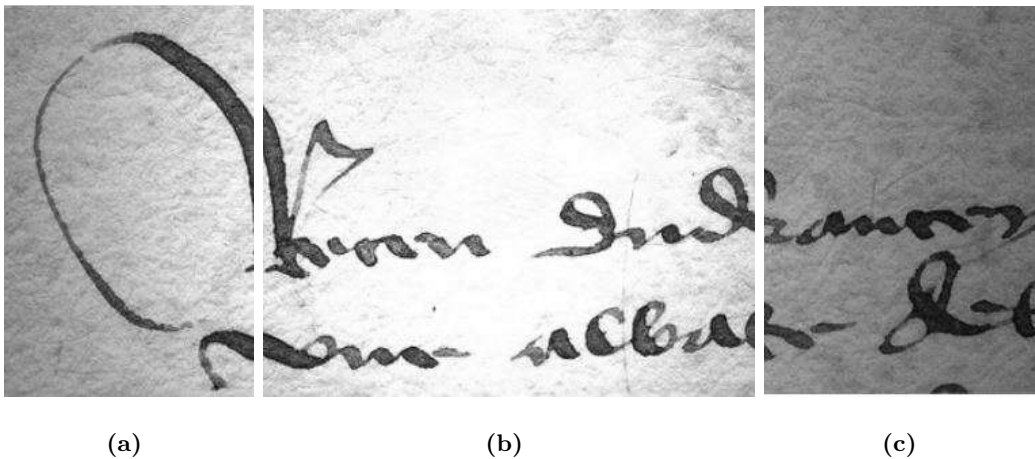


Figura 1.6: Imagen dividida en áreas con el mismo tipo de iluminación

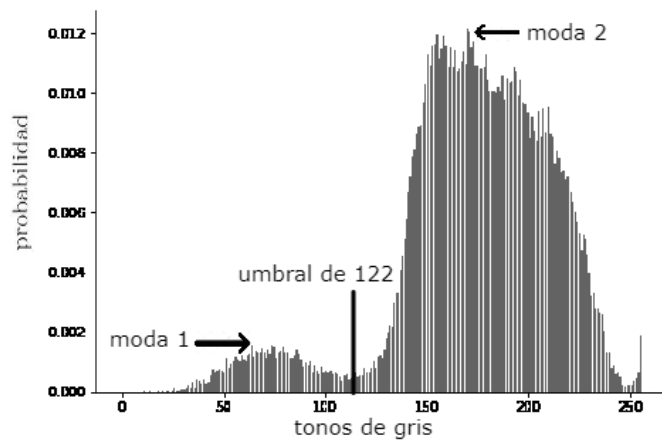
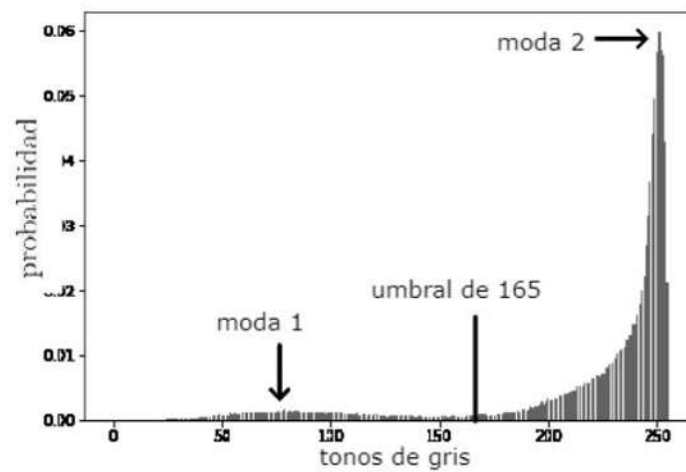
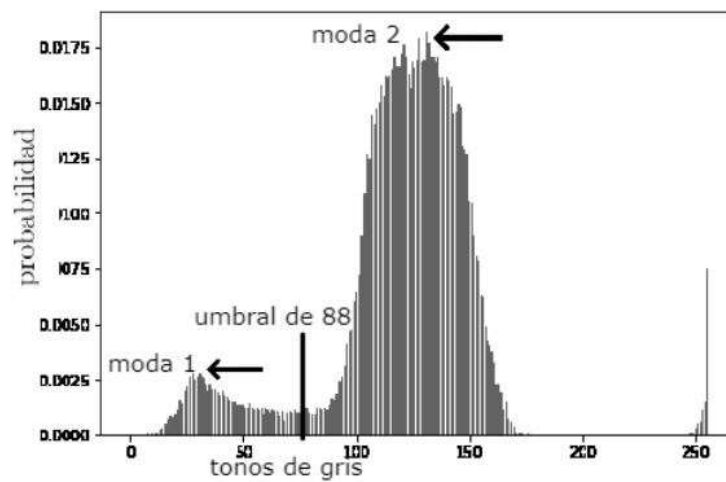
(a) *Distribución de probabilidad de 1.6 a)*(b) *Distribución de probabilidad de 1.6 b)*(c) *Distribución de probabilidad de 1.6 c)*

Figura 1.7: Distribuciones de probabilidad de áreas de las imágenes de la Figura 1.6

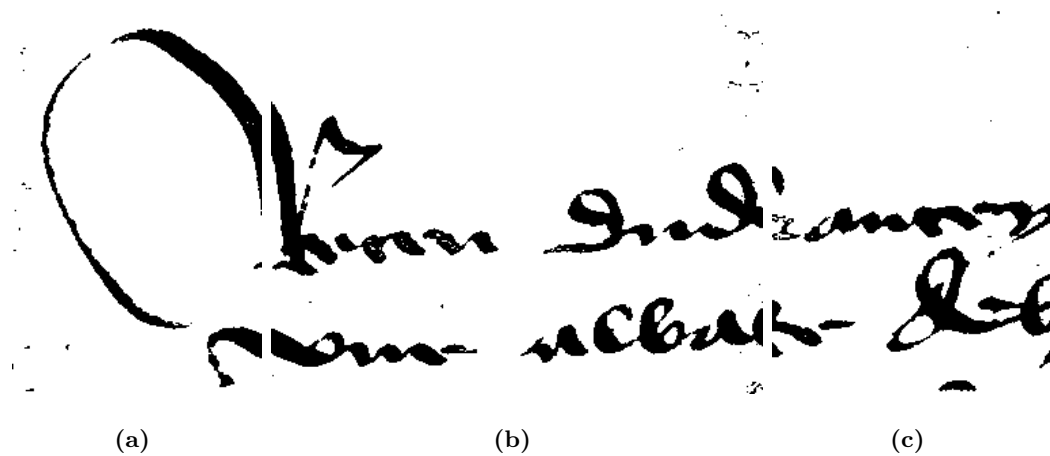


Figura 1.8: Imágenes binarizada por áreas con el mismo tipo de iluminación

Por lo tanto el problema de binarización de imágenes con iluminación no uniforme se reduce a dividir la imagen en áreas con el mismo tipo de iluminación y calcular un umbral en cada área. El problema de binarización de imágenes con iluminación no uniforme en esta tesis lo abordamos obteniendo automáticamente cada área de la imagen mediante ventanas de tamaño óptimo, las cuales se podrán traslapar y se calcularán en cada píxel de la imagen.

1.2. Marco teórico

Los distintos colores de una imagen se pueden obtener con el modelo RGB (por sus siglas en inglés de Red, Green, Blue) que es la composición del color en términos de la intensidad de los colores primarios de la luz. En una imagen denotada por I utilizando el modelo RGB para representar los colores, a cada píxel $I[r, c]$ corresponden tres componentes (R, G, B) los cuales son los colores rojo, verde y azul que son mezclados en diferentes intensidades para obtener el color del píxel. Cada componente puede tener 256 tonos diferentes por lo que se tienen $256 \times 256 \times 256 = 16,777,216$ colores posibles que podrá tener un píxel. El color rojo más intenso de un píxel tendrá en sus componentes los valores $(255, 0, 0)$, el verde más intenso $(0, 255, 0)$ y el azul más intenso tendrá $(0, 0, 255)$. La ausencia de color o color negro se obtiene con $(0, 0, 0)$ y el color blanco se forma con los tres colores a su máximo nivel $(255, 255, 255)$. Para convertir una imagen de color a su equivalente en escala de grises se obtiene el promedio de los tres componentes (R, G, B) en cada píxel y a cada componente se le asigna dicho promedio, por lo tanto un píxel en tono de gris tendrá el mismo valor en sus tres componentes.

En el proceso de digitalización de libros las imágenes binarias son una herramienta ideal para ahorrar una gran cantidad de recursos, debido a que en una imagen binaria los colores blanco

y negro se pueden representar en un bit a diferencia de una imagen de color donde el color de cada píxel se representan en 24 bits. Para binarizar imágenes se han ido desarrollando diferentes técnicas. La técnica que se utilizó para binarizar la Figura 1.1 a) consiste en determinar un umbral basándose en la distribución de probabilidad de la imagen, por lo que basta elegir un umbral a la mitad de las modas que representan el texto y fondo para binarizar la imagen. Otra forma de determinar el umbral es minimizando la varianza de las clases texto y fondo tal como en Otsu, 1979. Las técnicas que utilizan un umbral para binarizar la imagen son llamadas *binarización de imágenes con umbral global*. Cuando se calculan varios umbrales para binarizar la imagen la técnica se denomina *binarización de imágenes con umbral local*. Una manera de determinar umbrales locales en la imagen es haciendo uso de áreas con un tamaño adecuado de tal manera que se tenga una iluminación uniforme en cada área. Otra manera de calcular umbrales locales es obteniendo un umbral para cada píxel de la imagen basándose en las propiedades del vecindario del píxel, tal como se hace en Bradley and Roth, 2007, Sauvola and Pietikäinen, 2000 o Bernsen, 1986. La Figura 1.9 muestra un ejemplo del vecindario del píxel con valor 74 utilizando una ventana de tamaño $(2w + 1) \times (2w + 1)$ donde con $w = 1$.

72	73	79	82	81
71	90	80	83	81
70	70	74	76	80
69	30	70	70	71
68	69	71	73	72

Figura 1.9: Vecindario delimitado por una ventana

La evaluación de las imágenes binarizadas la podemos realizar de forma cualitativa utilizando diferentes métricas de calidad. Las métricas comparan una imagen binarizada (I_B) con determinado algoritmo, con una imagen de referencia (I_r) para medir la calidad de binarización de la imagen. En los siguientes párrafos se describen las métricas de calidad más utilizadas en binarización de imágenes.

1.2.1. Métricas de calidad

La métrica F denominada también F -score es definida en función de los píxeles clasificados como falsos positivos (FP), falsos negativos (FN) y verdaderos positivos (VP). En una imagen

binaria se define como clase positiva las letras y como clase negativa el fondo. Un valor alto de la métrica F indica una correcta binarización y es definida como:

$$F = \frac{2 * VP}{2VP + FP + FN} \quad (1.2)$$

El $PSNR$ (por sus siglas en inglés de Peak Signal-to-Noise Ratio) define la relación entre la máxima energía posible de una señal y el ruido que afecta a su representación fidedigna. Se basa en el error cuadrático medio (MSE) el cual se calcula como (1.3).

$$MSE = \frac{1}{N_r N_c} \sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} (I_B[r, c] - I_r[r, c])^2 \quad (1.3)$$

El $PSNR$ se define como:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (1.4)$$

donde MAX es el máximo valor que puede tomar un píxel en la imagen. Un valor pequeño de MSE se traduce en poco error en la imagen binarizada lo cual genera un valor grande de $PSNR$.

La métrica NRM (por sus siglas en inglés de Negative Rate Metric) evalúa los errores de la imagen binarizada I_B respecto a I_r , combinando el rango de falsos positivos NR_{FP} calculado mediante (1.5) donde FP (Falsos Positivos) y VN (Verdaderos Negativos)

$$NR_{FP} = \frac{FP}{FP + VN} \quad (1.5)$$

con el rango de falsos negativos NR_{FN} calculado mediante (1.5) donde FN (Falsos Negativos) y VP (Verdaderos Positivos).

$$NR_{FN} = \frac{FN}{FN + VP} \quad (1.6)$$

La métrica NRM es definida como (1.7) y entre más cercano a cero sea el valor la calidad de la binarización es mayor.

$$NRM = \frac{NR_{FP} + NR_{FN}}{2} \quad (1.7)$$

La métrica DRD (por sus siglas de Distorsión Recíproca de Distancia) mide la distorsión de la imagen binarizada I_B respecto a la imagen de referencia I_r . Esta métrica utiliza una matriz de pesos m donde cada peso $m[r, c]$ es determinado por la distancia recíproca medida respecto al píxel del centro. La matriz m de tamaño $(2w + 1) \times (2w + 1)$ donde w es definido para determinar el tamaño de la matriz se calcula con (1.8):

$$m[r, c] = \begin{cases} 0 & r = w, c = w \\ \frac{1}{\sqrt{(r-w)^2 + (c-w)^2}} & \text{si no} \end{cases} \quad (1.8)$$

Para cada píxel mal clasificado de I_B respecto a I_r calcula la distorsión DRD la cual esta dada por (1.9):

$$DRD_k = \sum_{r=-w}^w \sum_{c=-w}^w |I_B[r, c] - I_r[r, c]| \cdot m[r, c] \quad (1.9)$$

La métrica DRD se calcula mediante:

$$DRD = \frac{\sum_{k=1}^S DRD_k}{NUBN} \quad (1.10)$$

donde $NUBN$ estima el área válida (no vacía de letras) definida como el número de bloques de 8×8 no uniformes en I_r . La calidad de una imagen binarizada es mejor cuanto más se acerque a cero el valor de la métrica DRD .

1.3. Estado del arte

El algoritmo que se propone en [Bernsen, 1986](#) consiste en calcular un umbral en cada píxel de la imagen con información del contraste de los píxeles del vecindario delimitado por una ventana de tamaño constante. Se obtienen los valores máximo y mínimo en cada vecindario y el valor umbral corresponde al promedio de estos dos valores. La imagen es binarizada comparando cada píxel con su valor umbral el cual en las coordenadas $[r, c]$ está dado por:

$$T[r, c] = \frac{\max(w[r, c]) + \min(w[r, c])}{2} \quad (1.11)$$

El algoritmo propuesto en [Otsu, 1979](#) calcula de forma automática un umbral maximizando la separabilidad de las clases texto y fondo de la distribución de probabilidad de la imagen, haciendo una búsqueda exhaustiva en los valores de la escala de grises.

En el algoritmo para binarización de imágenes presentado en [Sauvola and Pietikäinen, 2000](#) la imagen es dividida en áreas de 10×20 píxeles. En cada área se calcula el promedio y el cambio en el contraste para determinar si el área pertenece al texto o al fondo de la imagen y así utilizar el algoritmo adecuado en cada área. Si el área pertenece al fondo de la imagen se utiliza el algoritmo Mamdani descrito en [Welstead, 1994](#) p. 494 para calcular el umbral de cada píxel, de lo contrario si el área pertenece al texto de la imagen se utiliza el algoritmo Niblack (1986) al cual se le agrega un parámetro que representa el rango dinámico $R = 128$. El umbral del píxel $I[r, c]$ de la imagen I está dado por (1.12) donde $\mu[r, c]$ es la media del vecindario del píxel, $\sigma[r, c]$ es la desviación estándar y $k = 0.5$.

$$T[r, c] = \mu[r, c] \cdot \left[1 + k \left(\frac{\sigma[r, c]}{R} - 1 \right) \right] \quad (1.12)$$

El algoritmo de binarización de imágenes que se presenta en [Gatos et al., 2006](#) es utilizado en documentos degradados y se realiza en diferentes fases. Inicia con un preprocesamiento de la imagen I para obtener una imagen I_w con un mejor contraste entre el fondo y el área de texto utilizando un filtro de Wiener. Utilizan el algoritmo de [Sauvola and Pietikäinen, 2000](#) para obtener una aproximación de la imagen binarizada S con la cual se calcula una aproximación del fondo F de la imagen I_w . Con el fondo F se calcula una segunda imagen binarizada mejorada B_g con (2.22) donde se asigna 0 si la distancia de la imagen preprocesada $I_w[r, c]$ con el fondo $F[r, c]$ sobrepasa el umbral $d_{rc} = q \times \delta_{rc}$. El parámetro δ_{rc} representa la distancia promedio entre el fondo de la imagen y el texto en caso de presentar iluminación uniforme, de lo contrario se utiliza una función sigmoide para calcular dicho parámetro. Finalmente se hace un post procesado con información del vecindario de cada píxel para mejorar la calidad de la imagen binarizada.

$$B_g[r, c] = \begin{cases} 0 & F[r, c] - I_w[r, c] > d_{rc} \\ 0 & \text{si no} \end{cases} \quad (1.13)$$

En [Bradley and Roth, 2007](#) se toma en cuenta la variación espacial de iluminación en la imagen por lo que se calcula un umbral para cada píxel de la imagen con el valor medio de los píxeles que pertenecen a su vecindario. El vecindario de cada píxel se delimita por una ventana de tamaño constante centrada en el píxel y la suma de los píxeles del vecindario se calcula utilizando imágenes integrales. Cada píxel es comparado con su umbral para ser clasificado como blanco o negro. El umbral está dado por (1.14) donde τ puede tomar diferentes valores.

$$T[r, c] = \mu[r, c] \cdot \left(\frac{100 - \tau}{100} \right) \quad (1.14)$$

El algoritmo k-medias [Jana et al., 2017](#) tiene como objetivo dividir un conjunto de elementos en un determinado número de grupos, donde cada grupo es representado por el valor medio (denotado por k) de sus elementos. El algoritmo inicia con k medias elegidas aleatoriamente para que cada elemento se asigne al grupo con la media más cercana. Se actualizan las medias con los elementos que se asignaron a cada grupo y se vuelven a reasignar los elementos al grupo con la media más cercana. El proceso de actualizar las medias y reasignar los elementos se realiza de forma iterativa hasta que las medias no cambien.

En el algoritmo propuesto por [Molina et al., 2018](#) la idea principal es obtener una imagen con iluminación apropiada para binarización a partir de la fusión de dos imágenes inversamente iluminadas. Para obtener la imagen inversamente iluminada de la imagen original calculan una aproximación de los componentes de iluminación (l_0) y reflectancia (R_0) de la imagen original la cual es definida como $I_0[r, c] = l_0[r, c]R_0[r, c]$ de acuerdo a [Rafael C. Gonzalez, 2002](#). La iluminación de la imagen se obtiene utilizando el operador morfológico de cerradura (denotado como \bullet) para remover el texto de la imagen y preservar la función de iluminación. La aproximación de la reflectancia la calculan con el cociente de la imagen original y su componente de luminancia. La luminancia inversa la obtienen como $l_1[r, c] = 255 - l_0[r, c]$. Para obtener la imagen inversamente iluminada se multiplica la reflectancia (R_0) y la luminancia inversa (l_1) para formar una imagen con el mismo texto pero con un patrón de iluminación inverso. Para detectar las áreas claras de las imágenes inversamente iluminadas que formarán parte de la imagen fusionada utilizan un filtro laplaciano. La imagen fusionada es binarizada con el algoritmo de [Otsu, 1979](#), [Bradley and Roth, 2007](#) o algún otro algoritmo del estado del arte de binarización de imágenes.

1.4. Objetivos de la tesis

Fundamentar e implementar algoritmo para binarizar imágenes con iluminación no uniforme utilizando ventanas de tamaño óptimo.

1.4.1. Objetivo general

Fundamentar e implementar un algoritmo para calcular de forma automática ventanas de tamaño óptimo que respeten los bordes del cambio de iluminación en la imagen. Implementar algoritmos de binarización de imágenes que utilicen ventanas adaptables de manera que el umbral

correspondiente a cada píxel sea calculado adecuadamente.

1.4.2. Objetivos particulares

- Analizar e implementar algoritmos de binarización de imágenes del estado del arte.
- Implementar algoritmo para calcular el tamaño óptimo de las ventanas.
- Implementar algoritmos propuestos de binarización de imágenes.
- Crear imágenes con diferentes patrones de iluminación no uniforme
- Implementar métricas de la calidad.
- Diseñar experimentos para probar resultados de los algoritmos propuestos.
- Comparar de forma cuantitativa y cualitativa los resultados de los algoritmos propuestos y del estado del arte

1.5. Justificación

Los algoritmos de binarización de imágenes del estado del arte utilizan ventanas de tamaño fijo o constante para delimitar el área sobre la que se calcula el umbral para binarizar la imagen. Al utilizar ventanas de tamaño constante en imágenes que presentan iluminación no uniforme, manchas o sombras el cálculo del umbral se ve afectado.

1.6. Hipótesis

Al utilizar ventanas de tamaño óptimo sujetas a respetar los bordes del cambio de iluminación en la imagen el cálculo del criterio de decisión no se verá afectado por el cambio de iluminación debido a que no se mezclarán píxeles de áreas claras y oscuras.

1.7. Variables de estudio

Exactitud de imágenes binarizadas, distribución de probabilidad de la imagen, valor medio del vecindario de cada píxel de la imagen y tamaño óptimo de las ventanas.

1.8. Preguntas de Investigación

¿Por qué utilizar de ventanas de tamaño constante afecta el criterio de decisión con el que se binarizan imágenes con iluminación no uniforme?

¿Utilizar ventanas de tamaño óptimo en cada píxel de la imagen mejora los resultados al binarizar imágenes con iluminación no uniforme?

1.9. Metodología

Examinar el marco teórico y los algoritmos del estado del arte sobre binarización de imágenes. Análisis e implementación de algoritmos de binarización de imágenes del estado del arte utilizando el lenguaje de programación *C++*. Implementación del algoritmo propuesto para el cálculo de tamaño óptimo de ventanas así como los propuestos para binarización de imágenes. Crear imágenes sintéticas y reales con distintos patrones de iluminación no uniforme. Implementar diferentes métricas de calidad para medir la exactitud de las imágenes binarizadas con los algoritmos propuestos. Comparar de manera cuantitativa y cualitativa las imágenes binarizadas con algoritmos del estado del arte y nuestras propuestas.

1.10. Descripción de Capítulos

En el capítulo 1 presentamos una introducción al tema de binarización de imágenes así como el problema que identificamos acerca de binarizar imágenes que presentan iluminación no uniforme y la forma en que abordaremos el problema. En el capítulo 2 describimos el trabajo relacionado a binarización de imágenes en el cual presentamos los algoritmos más utilizados para binarización de imágenes y los que utilizan ventanas. En el capítulo 3 presentamos nuestros algoritmos propuestos los cuales consisten en un algoritmo para el cálculo óptimo de ventanas y dos algoritmos para binarización de imágenes. En el capítulo 4 mostramos los resultados de imágenes reales y sintéticas con distintos patrones de iluminación no uniforme binarizadas con nuestros algoritmos propuestos y con algoritmos del estado del arte. En el capítulo 5 describimos las conclusiones obtenidas del trabajo realizado y el trabajo futuro.

Capítulo 2

Trabajo relacionado

La binarización de imágenes consiste en convertir una imagen en escala de gris a una imagen en blanco y negro. En el estado del arte existen diferentes algoritmos para binarizar imágenes, en este capítulo describimos los más utilizados así como los que utilizan ventanas para delimitar el área sobre la que se calcula el umbral.

2.1. Método de Otsu

El método de binarización de imágenes propuesto en [Otsu, 1979](#) calcula un umbral de forma automática para binarizar la imagen haciendo una búsqueda exhaustiva sobre los valores de los tonos de gris minimizando la varianza interclase de la imagen. El método tiene el propósito de maximizar la separabilidad de las clases correspondientes a los píxeles que pertenecen al texto y los que pertenecen al fondo empleando la distribución de probabilidad de la imagen.

Definimos el valor de $I[r, c]$ como un tono de gris correspondiente a un píxel de la imagen I el cual puede tener valores entre 0 y 255 que corresponden a la escala de grises. El total de píxeles con i -ésimo tono de la escala de grises en la imagen lo representamos con n_i . Por lo tanto en una imagen con N_r renglones y N_c columnas el total de píxeles se puede representar con la suma $N_r N_c = n_0 + n_1 + n_2 + n_3 + \dots + n_{255}$ y la probabilidad del i -ésimo tono de gris en la imagen como:

$$p_i = \frac{n_i}{N_r N_c} \quad (2.1)$$

El valor del umbral será un valor de la escala de grises denotado como u por lo que los píxeles con valor menor a u los llamamos clase $C_0(u)$ y los píxeles con valor mayor o igual a u los llamamos clase $C_1(u)$. El porcentaje de píxeles de la imagen en la clase $C_0(u)$ lo calculamos mediante (2.2) y

el porcentaje de píxeles que pertenecen a la clase $C_1(u)$ mediante (2.3).

$$\omega_0(u) = \sum_{i=0}^u p_i \quad (2.2)$$

$$\omega_1(u) = \sum_{i=u+1}^{255} p_i \quad (2.3)$$

El valor medio de la clase $C_k(u)$ es determinado como una suma ponderada de los diferentes tonos de gris en la clase cuyos pesos son las probabilidades de ocurrencia. El valor medio de la clase $C_0(u)$ lo calculamos con (2.4) y el de la clase $C_1(u)$ mediante(2.5).

$$\mu_0(u) = \frac{\sum_{i=0}^u ip_i}{\omega_0(u)} \quad (2.4)$$

$$\mu_1(u) = \frac{\sum_{i=u+1}^{255} ip_i}{\omega_1(u)} \quad (2.5)$$

La varianza es una medida de dispersión que representa la variabilidad de un conjunto de datos y se define como el valor medio del cuadrado de desviaciones con respecto la media. La varianza de la clase $C_0(u)$ la calculamos con (2.6) y de la clase $C_1(u)$ con (2.7).

$$\sigma_0^2(u) = \frac{\sum_{i=0}^u [\mu_0(u) - i]^2 p_i}{\omega_0(u)} \quad (2.6)$$

$$\sigma_1^2(u) = \frac{\sum_{i=u+1}^{255} [\mu_1(u) - i]^2 p_i}{\omega_1(u)} \quad (2.7)$$

La varianza intra-clase se define como una suma ponderada de la varianza de las clases y representa la variabilidad de los elementos de cada clase. Un valor pequeño de varianza intra-clase se traduce como poca dispersión en los elementos de las clases. La varianza intra-clase de las clases $C_0(u)$ y $C_1(u)$ divididas por el umbral u es definida como:

$$\sigma_W^2(u) = \sigma_0^2(u) \cdot \omega_0(u) + \sigma_1^2(u) \cdot \omega_1(u) \quad (2.8)$$

La varianza inter-clase calcula la dispersión de las clases midiendo la distancia de la media de las clases a la media de todo el conjunto de datos. La varianza inter-clase de $C_0(u)$ y $C_1(u)$ divididas por el umbral u es definida como:

$$\sigma_B^2(u) = \omega_0[\mu_0(u) - \mu_T]^2 + \omega_1[\mu_1(u) - \mu_T]^2 \quad (2.9)$$

donde μ_T representa la media de la imagen y se calcula como:

$$\mu_T = \sum_{i=0}^{255} ip_i \quad (2.10)$$

El umbral óptimo u para binarizar la imagen es encontrado mediante una búsqueda exhaustiva sobre los valores de la escala de grises minimizando la varianza intra-clase o maximizando la varianza inter-clase de la imagen. El umbral óptimo calculado con la varianza intra-clase lo representamos como (2.11) y calculado con la varianza inter-clase como (2.12).

$$u_{min}^* = \operatorname{argmin}_{0 \leq u \leq 255} \sigma_W^2(u) \quad (2.11)$$

$$u_{max}^* = \operatorname{argmax}_{0 \leq u \leq 255} \sigma_B^2(u) \quad (2.12)$$

El valor de u_{min}^* y u_{max}^* será el mismo valor debido a que donde se minimiza la varianza intra-clase se maximiza varianza inter-clase. El método *Otsu* se muestra en Algoritmo 1 el cual recibe la imagen en escala de grises I . En la línea 1 se inicializan variables y de la 2 a la 9 se calcula la varianza inter-clase. De la línea 10 a la 12 se actualiza el umbral óptimo. De la línea 13 a la 18 se binariza la imagen y en la línea 19 se regresa la imagen binarizada.

Algoritmo 1: Otsu(I)

```

1  inicializar variables
2  para  $u = 0$  a  $255$  hacer
3       $\omega_0(u) \leftarrow \sum_{i=0}^k p_i$ 
4       $\omega_1(u) \leftarrow \sum_{i=u+1}^{255} p_i$ 
5       $\mu_0(u) \leftarrow \frac{\sum_{i=0}^u ip_i}{\omega_0(u)}$ 
6       $\mu_1(u) \leftarrow \frac{\sum_{i=u+1}^{255} ip_i}{\omega_1(u)}$ 
7       $\sigma_0^2(u) \leftarrow \frac{\sum_{i=0}^u [\mu_0(u) - i]^2 p_i}{\omega_0(u)}$ 
8       $\sigma_1^2(u) \leftarrow \frac{\sum_{i=u+1}^{255} [\mu_1(u) - i]^2 p_i}{\omega_1(u)}$ 
9       $\sigma_W^2(u) \leftarrow \sigma_0^2(u) \cdot \omega_0(u) + \sigma_1^2(u) \cdot \omega_1(u)$ 
10     si  $\sigma_W^2(u) \leq var_m$  entonces
11          $u^* \leftarrow u$ 
12          $var_m \leftarrow \sigma_W^2(u)$ 
13  para  $r = 0$  a  $N_r - 1$  hacer
14     para  $c = 0$  a  $N_c - 1$  hacer
15         si  $I[r, c] < u^*$  entonces
16              $I_B[r, c] \leftarrow 0$ 
17         en otro caso
18              $I_B[r, c] \leftarrow 255$ 
19  Regresa la imagen binarizada  $I_B$ 

```

2.2. Método de k-medias

El método de k-medias propuesto por [MacQueen et al., 1967](#) consiste en agrupar un conjunto de elementos en k número de grupos representados por sus medias de manera que los elementos pertenezcan al grupo con la media más cercana, con el objetivo de que se minimice la suma de las distancias entre los elementos de cada grupo. Al inicio del algoritmo el valor de las medias que representan los grupos se asignan aleatoriamente y cada elemento es asignado al grupo con la media más próxima, posteriormente se actualizan las medias con los elementos que se asignaron a cada grupo. En la siguiente iteración se vuelven a reasignar los elementos al grupo con la media más cercana y se vuelven a actualizar las medias, este proceso se realiza de forma iterativa hasta que las medias ya no cambian o después de un determinado número de iteraciones. Para binarizar una

imagen con el algoritmo de *k*-medias se establecen dos grupos, uno para los píxeles que pertenecen al fondo y otro para los que pertenecen al texto. En (2.13) se muestra la distancia del valor del píxel $I[r, c]$ a la media μ_0 de los píxeles de las letras y en (2.14) a la media μ_1 de los píxeles del fondo.

$$d(\mu_0, I[r, c]) = |I[r, c] - \mu_0| \quad (2.13)$$

$$d(\mu_1, I[r, c]) = |I[r, c] - \mu_1| \quad (2.14)$$

El método de *k*-medias se muestra en el Algoritmo 2 el cual recibe la imagen I en escala de grises y el número de iteraciones N_{iter} . En las líneas 1 a 3 se inicializan variables y en la línea 4 inicia el ciclo de las iteraciones. En las líneas 5 y 6 inician los ciclos para recorrer la imagen. En la línea 7 se compara la distancia del valor $I[r, c]$ a las medias μ_0 y μ_1 , si el píxel está más cerca de μ_0 se clasifica como negro. En la línea 9 se cuentan los píxeles clasificados como negros. En la línea 10 se suman los valores de la imagen original que se han clasificado como negros, en caso de que el píxel este más cerca de μ_1 , se clasifica como blanco. En la línea 13 se cuentan los píxeles clasificados como blancos. En la línea 14 se suman los píxeles de la imagen original que se clasificaron como blancos y en las líneas 18 y 19 se actualizan las medias de los grupos. En la línea 20 finaliza el ciclo de las iteraciones y en la línea 21 se regresa la imagen binarizada.

Algoritmo 2: k-medias(I, N_{iter})

```

1   $\mu_0 \leftarrow random()$ 
2   $\mu_1 \leftarrow random()$ 
3  Inicializa variables
4  para  $iter = 0$  a  $N_{iter} - 1$  hacer
5      para  $r = 0$  a  $N_r - 1$  hacer
6          para  $c = 0$  a  $N_c - 1$  hacer
7              si  $|I[r, c] - \mu_0| \leq |I[r, c] - \mu_1|$  entonces
8                   $I_B[r, c] \leftarrow 0$ 
9                   $suma_n \leftarrow suma_n + 1$ 
10                  $total_n \leftarrow total_n + I[r, c]$ 
11                 en otro caso
12                      $I_B[r, c] \leftarrow 255$ 
13                      $suma_b \leftarrow suma_b + 1$ 
14                      $total_b \leftarrow total_b + I[r, c]$ 
15                 fin
16             fin
17         fin
18          $\mu_0 \leftarrow \frac{suma_n}{total_n}$ 
19          $\mu_1 \leftarrow \frac{suma_b}{total_b}$ 
20 fin
21 Regresa imagen binarizada  $I_B$ 

```

2.3. Método de Bradley y Roth

Derek Bradley y Gerhard Roth en 2007 presentaron una técnica enfocada en calcular umbrales adaptativos sobre imágenes en transmisión de vídeo en tiempo real [Bradley and Roth, 2007](#). El método calcula un umbral en cada píxel de la imagen que obtienen con el promedio de los tonos de gris que pertenecen al vecindario del píxel delimitado mediante un cuadrado llamado ventana de tamaño $(2w + 1) \times (2w + 1)$. Utilizan el parámetro τ con el que se determina que si el tono de gris del píxel $I[r, c]$ es τ por ciento menor que el umbral es clasificado como negro, de lo contrario como blanco. En [Bradley and Roth, 2007](#) utilizan imágenes integrales [Viola and Jones, 2001](#) para calcular la suma de los tonos de gris dentro de las ventanas. La imagen integral H es una

transformación de la imagen original I y en las coordenadas $H[r, c]$ es definida como la suma de los valores arriba y a la izquierda de I y está dada por:

$$H[r, c] = \sum_{k=0}^r \sum_{l=0}^c I[k, l] \quad (2.15)$$

El cálculo de la imagen integral utilizando (2.15) resulta en un algoritmo de orden $O(N^4)$ por lo que se aplican las propiedades de asociatividad sobre (2.15) y se obtiene:

$$H[r, c] = \sum_{k=0}^{r-1} \sum_{l=0}^c I[k, l] + \sum_{l=0}^c I[r, l] \quad (2.16)$$

$$H[r, c] = \sum_{k=0}^{r-1} \sum_{l=0}^c I[k, l] + \sum_{l=0}^{c-1} I[r, l] + I[r, c] \quad (2.17)$$

$$H[r, c] = \sum_{k=0}^{r-1} \sum_{l=0}^c I[k, l] + \left(\sum_{k=0}^{c-1} \sum_{l=0}^r I[r, l] + \sum_{k=0}^{c-1} \sum_{l=0}^{r-1} I[r, l] \right) + I[r, c] \quad (2.18)$$

sustituyen la definición de imagen integral (2.15) en (2.18) para obtener la siguiente recurrencia:

$$H[r, c] = I[r, c] + H[r - 1, c] + H[r, c - 1] - H[r - 1, c - 1] \quad (2.19)$$

Se utiliza (2.19) para calcular la imagen integral H que permite calcular de forma eficiente la suma de los tonos de gris de los píxeles dentro de una ventana empleando un número constante de operaciones independientemente del tamaño de la ventana [Viola and Jones, 2001](#). La Figura 2.1 a) muestra una ventana de tamaño $(2w + 1) \times (2w + 1)$ centrada en las coordenadas $[r, c]$ de la imagen I . Para calcular la suma de los tonos de gris de los píxeles dentro de la ventana se obtienen cuatro valores de la imagen integral H de la imagen I , los cuales representan la suma sobre las áreas: $A_1 = H[r + w, c + w]$ mostrada en la Figura 2.1 b), el área $A_2 = H[r + w, c]$ mostrada en la Figura 2.1 c), el área $A_3 = H[r - w, c + w]$ mostrada en la Figura 2.1 e) y el área $A_4 = H[r - w, c - w]$ mostrada en la Figura 2.1 e).

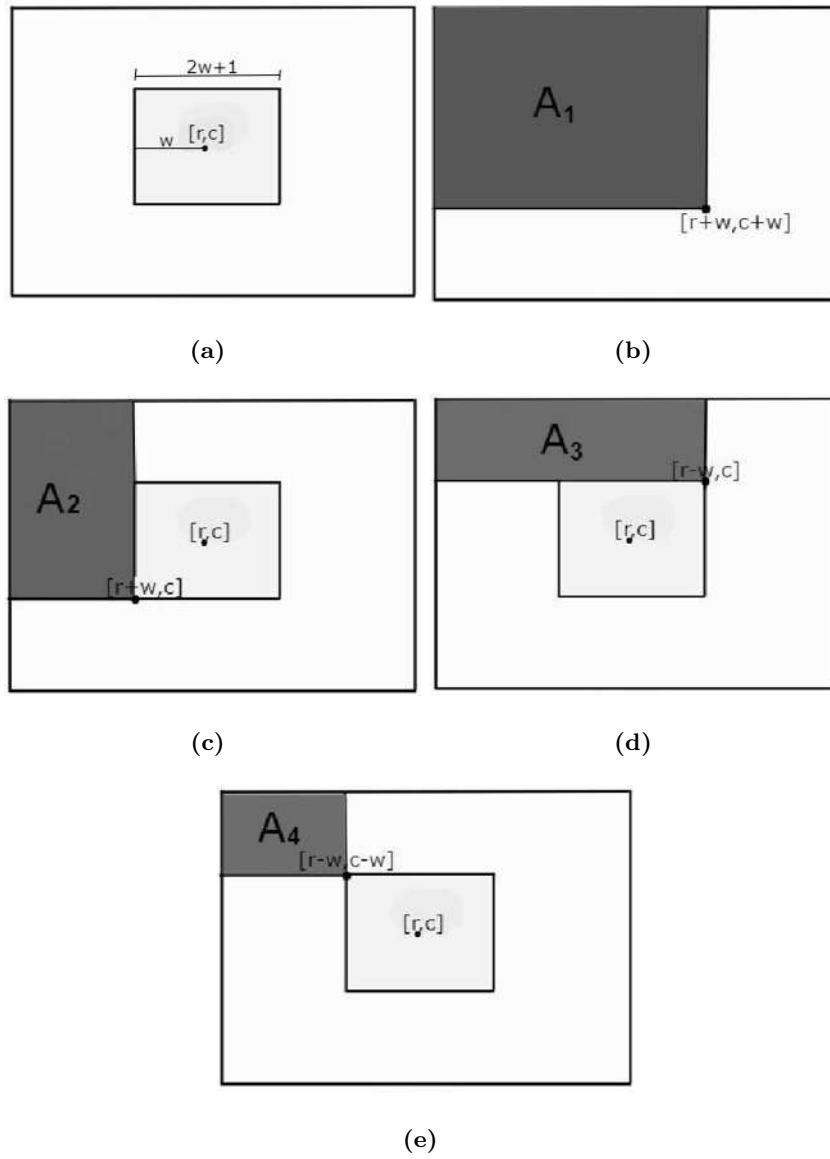


Figura 2.1: Suma de tonos de gris de píxeles delimitados por una ventana utilizando imágenes integrales

Por lo tanto la suma de los tonos de gris dentro de la ventana de tamaño $(2w+1) \times (2w+1)$ centrada en las coordenadas $I[r, c]$ se representa como $S[r, c] = A_1 - A_2 - A_3 + A_4$ y se calcula mediante (2.20) donde H es la imagen integral de a I :

$$S[r, c] = H[r + w, c + w] - H[r + w, c - w - 1] - H[r - w - 1, c + w] + H[r - w - 1, c - w - 1] \quad (2.20)$$

Dada la imagen integral y el tamaño de la ventana w definimos el promedio del vecindario de $I[r, c]$ como:

$$\mu_{rc} = \frac{S[r, c]}{(2w + 1)^2} \quad (2.21)$$

Con el promedio μ_{rc} y el parámetro τ se determina si el píxel $I[r, c]$ es clasificado como blanco o negro tal como:

$$I_B[r, c] = \begin{cases} 0 & I[r, c] < \frac{100-\tau}{100} \mu_{rc} \\ 255 & \text{si no} \end{cases} \quad (2.22)$$

El método de [Bradley and Roth, 2007](#) se muestra en el Algoritmo 3 el cual recibe la imagen en escala de grises I , τ y w . En las líneas 2 a 5 se calcula la imagen integral H con (2.19). En las líneas 7 y 8 inician los ciclos para recorrer la imagen I . En la línea 9 se obtiene el promedio de los tonos de gris del vecindario del píxel $I[r, c]$ mediante (2.21). En las líneas 10 a 14 se binariza la imagen y en la línea 17 se regresa la imagen binarizada I_B .

Algoritmo 3: BradleyRoth(I, τ, w)

```

1  inicializa variables
2  para  $r = 0$  a  $N_r - 1$  hacer
3      para  $c = 0$  a  $N_c - 1$  hacer
4          Calcular  $H$  mediante (2.19)
5      fin
6  fin
7  para  $r = 0$  a  $N_r - 1$  hacer
8      para  $c = 0$  a  $N_c - 1$  hacer
9          Calcular  $\mu_{rc}$  mediante (2.28)
10         si  $I[r, c] < \frac{100-\tau}{100} \mu_{rc}$  entonces
11              $I_B[r, c] \leftarrow 0$ 
12         en otro caso
13              $I_B[r, c] \leftarrow 255$ 
14         fin
15     fin
16 fin
17 Regresa imagen  $I_B$ 

```

2.4. Algoritmo CLI-S para fusión de imágenes multi-foco

El algoritmo de Combinación Lineal de Imágenes Simple (CLI-S) propuesto por [Calderon et al., 2016](#) tiene como propósito fusionar imágenes multifoco con el fin de combinar la información nítida presente en imágenes de una misma escena. Las imágenes multifoco son capturadas bajo diferentes condiciones de enfoque con el propósito de obtener una imagen con la mayor nitidez posible. CLI-S calcula una matriz de pesos binaria P donde el color blanco representa el área en la que I_0 es más nítida y en color negro el áreas en la que I_1 es más nítida, tal como:

$$P[r, c] = \begin{cases} 1 & I_0[r, c] \text{ es más nítido que } I_1[r, c] \\ 0 & \text{si no} \end{cases} \quad (2.23)$$

la nitidez de un píxel de las imágenes multi-foco la calculan utilizando el kernel laplaciano h :

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

por lo que en las coordenadas $[r, c]$ la nitidez es obtenida como el valor absoluto de la convolución con h tal como:

$$F_0[r, c] = |I_0[r, c] * h| \quad (2.24)$$

$$F_1[r, c] = |I_1[r, c] * h| \quad (2.25)$$

si el valor de $F_0[r, c]$ es mayor que $F_1[r, c]$ el valor del píxel en las coordenadas $[r, c]$ de la imagen I_0 es más nítido que el de la imagen I_1 , de lo contrario el píxel de la imagen I_1 es más nítido que el de I_0 por lo tanto P se calcula como:

$$P[r, c] = \begin{cases} 1 & F_0[r, c] > F_1[r, c] \\ 0 & \text{si no} \end{cases} \quad (2.26)$$

Con el propósito de obtener coherencia espacial se calcula el promedio del vecindario de $F_0[r, c]$ y $F_1[r, c]$ delimitado por una ventana de tamaño $(2w + 1) \times (2w + 1)$ centrada en las coordenadas $[r, c]$ para lo cual calculan la matriz ΔF mediante (2.27). Si el signo del valor del píxel $\Delta F[r, c]$ es negativo quiere decir que $I_0[r, c]$ es más nítido de lo contrario que $I_1[r, c]$ es más nítido.

$$\Delta F[r, c] = F_0[r, c] - F_1[r, c] \quad (2.27)$$

Para calcular la suma de los píxeles dentro de la ventana centrada en $\Delta F[r, c]$ utilizan imágenes integrales Viola and Jones, 2001. La imagen integral de ΔF la obtienen con (2.19) y la suma $S[r, c]$ mediante 2.20, si el valor $S[r, c]$ es positivo el valor que maximiza la nitidez en la imagen fusionada $I_f[r, c]$ es 1 en la matriz de pesos binaria $P[r, c]$ de lo contrario 0. De esta manera la matriz de pesos binaria P la calculan como:

$$P[r, c] = \begin{cases} 1 & S[r, c] > 0 \\ 0 & \text{si no} \end{cases} \quad (2.28)$$

Con la matriz binaria de pesos P la imagen fusionada I_f es calculada mediante:

$$I_f[r, c] = I_0[r, c]P[r, c] + I_1[r, c](1 - P[r, c]) \quad (2.29)$$

CLI-S se muestra en el Algoritmo 4 y recibe las imágenes multi-foco I_0 e I_1 y el parámetro w para el tamaño de las ventanas. En las líneas 2 y 3 se calculan las funciones de calidad F_0 y F_1 . En la línea 4 se calcula ΔF y en la línea 5 la imagen integral de la matriz ΔF . En las líneas 6 a 10 se obtienen las sumas del vecindario de cada píxel de ΔF y de las líneas 11 a 20 se obtiene la imagen fusionada I_f .

Algoritmo 4: CLI-S(I_0, I_1, w)

```

1 inicializa variables
2 Calcula  $F_0$  con (2.24)
3 Calcula  $F_1$  con (2.25)
4 Calcular  $\Delta F$  con (2.27)
5 Calcular imagen integral  $H$  de  $\Delta F$  con (2.19)
6 para  $r = 0$  a  $N_r - 1$  hacer
7   | para  $c = 0$  a  $N_c - 1$  hacer
8   |   | Calcular  $S[r, c]$  con (2.20)
9   |   fin
10  fin
11 para  $r = 0$  a  $N_r - 1$  hacer
12   | para  $c = 0$  a  $N_c - 1$  hacer
13   |   | si  $S[r, c] < 0$  entonces
14   |   |   |  $P[r, c] \leftarrow 0$ 
15   |   |   en otro caso
16   |   |   |  $P[r, c] \leftarrow 1$ 
17   |   |   fin
18   |   fin
19 fin
20  $I_f[r, c] \leftarrow I_1[r, c]P[r, c] + I_0[r, c](1 - P[r, c])$ 
21 Regresa imagen fusionada  $I_f$ 

```

2.5. Algoritmo para binarización de imágenes propuesto por Edgar Molina

En el algoritmo propuesto en [Molina et al., 2018](#) dada una imagen I_0 con iluminación no uniforme se calcula la imagen I_1 con condiciones de iluminación inversas a I_0 . Las imágenes inversamente iluminadas I_0 e I_1 son fusionadas con el algoritmo CLI-S de [Calderon et al., 2016](#) con lo cual tratan de obtener una imagen con iluminación adecuada para binarización. Sin embargo consideramos que el kernel laplaciano utilizado en CLI-S el cual es empleado para medir la propiedad de nitidez de las imágenes multi-foco no es el adecuado para medir la iluminación de las imágenes inversamente iluminadas I_0 e I_1 . [Molina et al., 2018](#) obtiene la imagen inversamente iluminada

I_1 desde el punto de vista físico donde una imagen puede ser caracterizada por la multiplicación de dos componentes básicos: la cantidad de iluminación incidente en una escena y la cantidad de iluminación reflejada por los objetos en dicha escena, dichos componentes son llamados *luminancia* y *reflectancia* [Rafael C. Gonzalez, 2002](#). De acuerdo a esta definición el valor del píxel $I_0[r, c]$ es definido como (2.30) donde L_0 es el componente de *luminancia* y R_0 el componente de *reflectancia* de la imagen I_0 .

$$I_0[r, c] = L_0[r, c]R_0[r, c] \quad (2.30)$$

En [Molina et al., 2018](#) utilizan el operador morfológico de cerradura (denotado por \bullet) para obtener una aproximación de la *luminancia* de la imagen. El operador morfológico de cerradura es definido como: la aplicación del operador de dilatación (denotado por \oplus) sobre la imagen I_0 seguido del operador de erosión (denotado por \ominus) [Soille, 2004](#). El operador morfológico de cerradura es definido como:

$$L_0[r, c] = I_0[r, c] \bullet C = (I_0[r, c] \oplus C) \ominus C \quad (2.31)$$

donde C es el kernel morfológico.

El operador de dilatación \oplus se calcula con (2.32) y el operador de erosión \ominus con (2.33).

$$\hat{I}_0[r, c] = I_0[r, c] \oplus C = \max \left(\sum_{k=r-C}^{r+C} \sum_{l=c-C}^{c+C} I_0[k, l] \right) \quad (2.32)$$

$$L_0[r, c] = \hat{I}_0[r, c] \ominus C = \min \left(\sum_{k=r-C}^{r+C} \sum_{l=c-C}^{c+C} \hat{I}_1[k, l] \right) \quad (2.33)$$

La Figura 2.2 a) muestra una imagen de texto I_0 que presenta iluminación no uniforme. La Figura 2.2 b) muestra una aproximación del componente de *luminancia* de la Figura 2.2 a) calculada al aplicar el operador morfológico de cerradura con un kernel morfológico de tamaño $C = 20$.



Figura 2.2: Componente de luminancia de I_0 utilizando un kernel morfológico de tamaño $C = 20$

El componente de *reflectancia* lo calculan utilizando (2.30) mediante el cociente:

$$R_0[r, c] = \frac{I_0[r, c]}{L_0[r, c]} \quad (2.34)$$

y la luminancia inversa la calculan como:

$$L_1[r, c] = 255 - L_0[r, c] \quad (2.35)$$

La Figura 2.3 a) muestra la imagen I_0 que presenta iluminación no uniforme y la Figura 2.3 b) muestra una aproximación de su luminancia inversa.

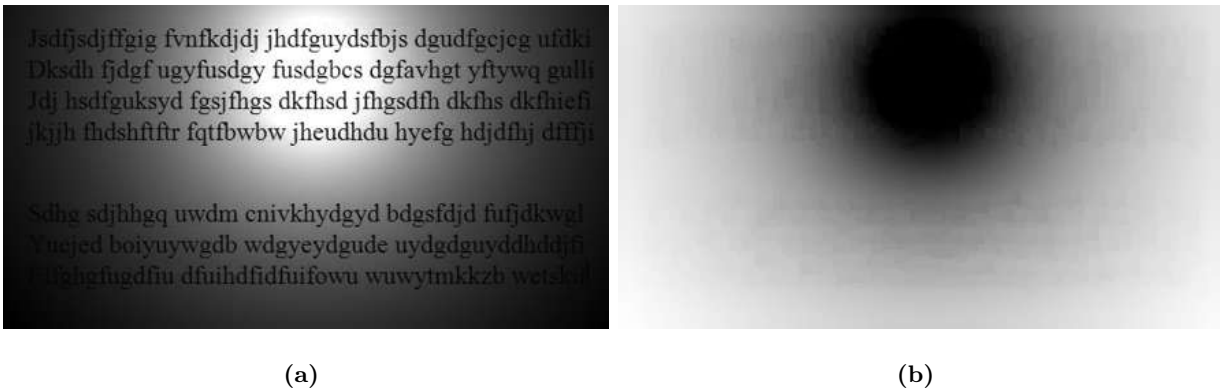


Figura 2.3: Luminancia inversa de la Figura 2.3 a)

La imagen con condiciones de iluminación inversas a la imagen original I_0 es obtenida al multiplicar los componentes de *reflectancia* R_0 y luminancia inversa L_1 tal como:

$$I_1[r, c] = R_0[r, c]L_1[r, c] \quad (2.36)$$

Las imágenes inversamente iluminadas I_0 e I_1 son fusionadas con el algoritmo CLI-S por medio de:

$$I_f[r, c] = I_0[r, c]P[r, c] + I_1[r, c](1 - P[r, c]) \quad (2.37)$$

El método propuesto por Molina et al., 2018 se muestra en el Algoritmo 5 y recibe la imagen I_0 con iluminación no uniforme y el kernel morfológico C . En la línea 1 se inicializan variables y en la línea 2 se obtiene el componente de luminancia L_0 . En las líneas 3 a 9 se obtienen la imagen inversamente iluminada I_1 . En la línea 10 se emplea CLI-S para obtener la imagen fusionada I_f con iluminación adecuada para binarización y en la línea 11 se binariza la imagen con el Algoritmo Otsu, 1979.

Algoritmo 5: Algoritmo de Edgar Molina(I_0, C)

```

1 inicializa variables
2  $L_0 = (I_1 \oplus C) \ominus C$ 
3 para  $r = 0$  a  $N_r - 1$  hacer
4   para  $c = 0$  a  $N_c - 1$  hacer
5      $R_0[r, c] \leftarrow \frac{I_0[r, c]}{L_0[r, c]}$ 
6      $L_1[r, c] \leftarrow 255 - L_0[r, c]$ 
7      $I_1[r, c] \leftarrow R_0[r, c]L_1[r, c]$ 
8   fin
9 fin
10 Fusionar  $I_0$  e  $I_2$  mediante CLI-S Algoritmo 4
11 Binariza  $I_f$ 
12 Regresa imagen binarizada  $I_B$ 

```

2.6. Conclusión

En este capítulo presentamos diferentes algoritmos de binarización de imágenes entre ellos uno de clasificación y algunos que utilizan umbral global o local. Algoritmos como el que fue propuesto en [Otsu, 1979](#) que es un algoritmo rápido y k-medias que se utiliza en clasificación tienen problemas para binarizar imágenes con iluminación no uniforme debido a que utilizan umbral global. Para abordar los problemas de iluminación no uniforme se utilizan otros métodos como calcular umbral local para diferentes áreas de la imagen tal como el algoritmo [Bradley and Roth, 2007](#) el cual utiliza ventanas y es fácil de implementar. El algoritmo de [Molina et al., 2018](#) calcula una imagen fusionada con iluminación uniforme empleando el algoritmo CLI-S, con el objetivo de binarizarla con [Otsu, 1979](#) o [Bradley and Roth, 2007](#). Sin embargo vimos que los algoritmos del estado del arte para binarización de imágenes que utilizan ventanas lo hacen empleando el tamaño de las ventanas fijo o constante.

Capítulo 3

Algoritmos propuestos

En este capítulo describimos dos algoritmos para binarización de imágenes con iluminación no uniforme los cuales denominamos Binarización de Imágenes con Promedios Ponderados (BIPP) y Binarización de Imágenes con Ventanas Adaptables (BIVA). También presentamos el algoritmo Ventana Óptima (VO) que tiene como objetivo calcular el tamaño adecuado de las ventanas de cada píxel de la imagen. En BIPP y BIVA empleamos VO para calcular el tamaño de las ventanas con las que delimitamos el vecindario de los píxeles con el propósito de calcular adecuadamente el criterio de decisión para binarizar la imagen. En el estado del arte existen diferentes algoritmos de binarización de imágenes sin embargo solamente algunos toman en cuenta el problema de iluminación no uniforme. Dos algoritmos de binarización de imágenes muy utilizados por su eficiencia son [Otsu, 1979](#) y [Bradley and Roth, 2007](#) el cual utiliza ventanas de tamaño constante. Para mostrar el desempeño de [Bradley and Roth, 2007](#), [Otsu, 1979](#) y [Molina et al., 2018](#) al binarizar imágenes con iluminación no uniforme hicimos un experimento que tiene como objetivo clasificar los píxeles de tres imágenes distintas en blancos y negros. Generamos las imágenes I_1 , I_2 e I_3 cada una con $N_r = 150$ renglones y $N_c = 150$ filas, dichas imágenes tienen distribución de probabilidad normal y son bimodales tal como en las imágenes con texto y fondo blanco. En la distribución de probabilidad de una imagen de texto y fondo blanco las letras son representadas por los tonos de gris más oscuros los cuales denominamos clase 0 y los asignamos a las coordenadas $[r, c]$ con valores $50 < r < 100$ y $50 < c < 100$ y el fondo es representado con tonos de gris más claros los denominamos clase 1 y los asignamos al resto de los píxeles. En las 3 imágenes la clase 1 tiene media $\mu = 50$ y la clase 2 media $\mu = 100$ lo que cambiamos es la varianza en cada imagen. La imagen I_1 tiene clase 0 con media $\mu = 50$ y varianza $\sigma = 3$, y clase 1 con media $\mu = 100$ y varianza $\sigma = 3$. La imagen I_2 tiene clase 0 con media $\mu = 50$ y varianza $\sigma = 15$, y clase 1 con media $\mu = 100$ y varianza $\sigma = 15$. La imagen I_3 tiene clase 0 con media $\mu = 50$ y varianza $\sigma = 30$, y clase 1 con media $\mu = 100$ y varianza $\sigma = 30$.

Para medir de manera cuantitativa las imágenes I_1 , I_2 e I_3 binarizadas con los algoritmos [Otsu, 1979](#), [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) utilizamos el porcentaje de aciertos o exactitud. Para obtener el porcentaje de aciertos calculamos el error mediante (3.1) donde \oplus es la operación or exclusivo, I_B representa la imagen binarizada e I_r representa la imagen de referencia.

$$E = \frac{\sum_{r=0}^{N_r-1} \sum_{c=0}^{N_c-1} I_B[r, c] \oplus I_r[r, c]}{N_r N_c} \quad (3.1)$$

el porcentaje de aciertos (AC) lo calculamos con (3.2).

$$AC = 100(1 - E) \quad (3.2)$$

La Figura 3.1 a) muestra la imagen I_1 donde la clase 0 tiene media $\mu = 50$ con varianza $\sigma = 3$ y clase 1 con media $\mu = 100$ y varianza $\sigma = 3$. La Figura 3.1 b) muestra la distribución de probabilidad de la Figura 3.1 a) en la que se puede identificar la clase 0 donde los píxeles tienen valor de 47 a 53 y la clase 1 donde los píxeles tienen valor de 97 a 103.

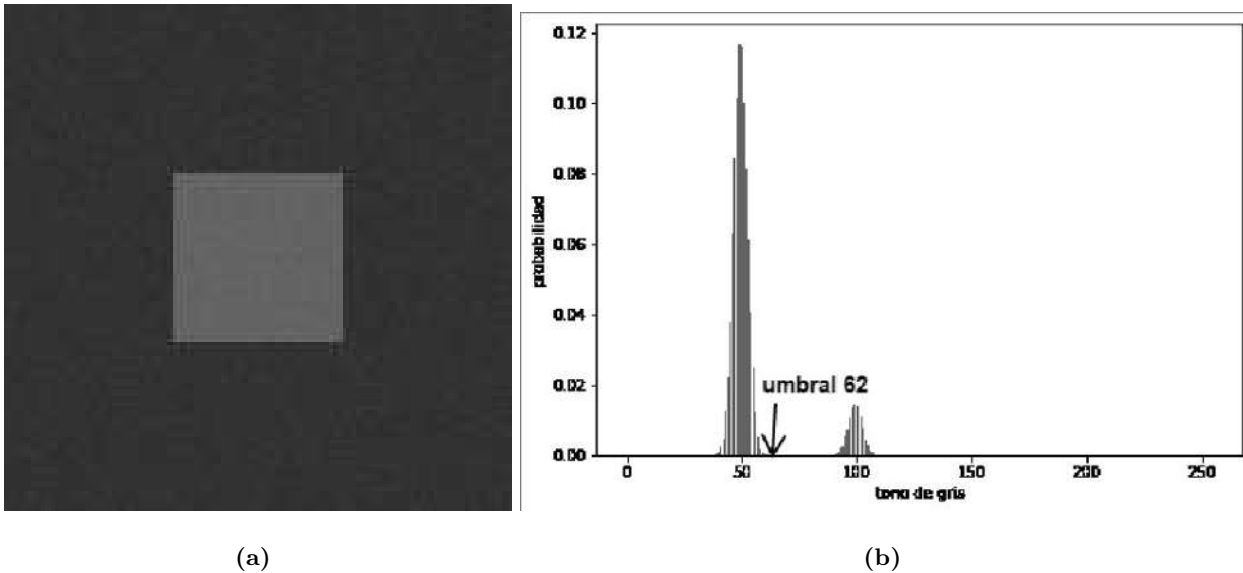


Figura 3.1: A la izquierda imagen I_1 y a la derecha su distribución de probabilidad

La Figura 3.2 a) muestra la imagen I_1 binarizada con el algoritmo [Otsu, 1979](#) la cual tiene 100% de exactitud. En la Figura 3.1 b) mostramos el umbral correspondiente al tono de gris con valor 62 con el que [Otsu, 1979](#) clasifica los píxeles y como podemos ver dicho valor separa adecuadamente las clases. La Figura 3.2 b) muestra la imagen I_1 binarizada con el algoritmo [Bradley and Roth, 2007](#) donde $\tau = 10$ y $w = 60$, dicha imagen tiene 82.58% de exactitud. La Figura 3.2 c) muestra el resultado de binarizar I_1 con [Molina et al., 2018](#) donde $w = 80$ y $C = 80$ la cual tiene

31.08 % de exactitud. En las tres imágenes vimos que la binarizada con [Otsu, 1979](#) es la que tiene mayor exactitud, y que a pesar de que en la distribución de probabilidad de la imagen I_1 las clases están separadas y se esperaría un buen desempeño de los algoritmos, las imágenes binarizadas con [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) tuvieron poca exactitud.

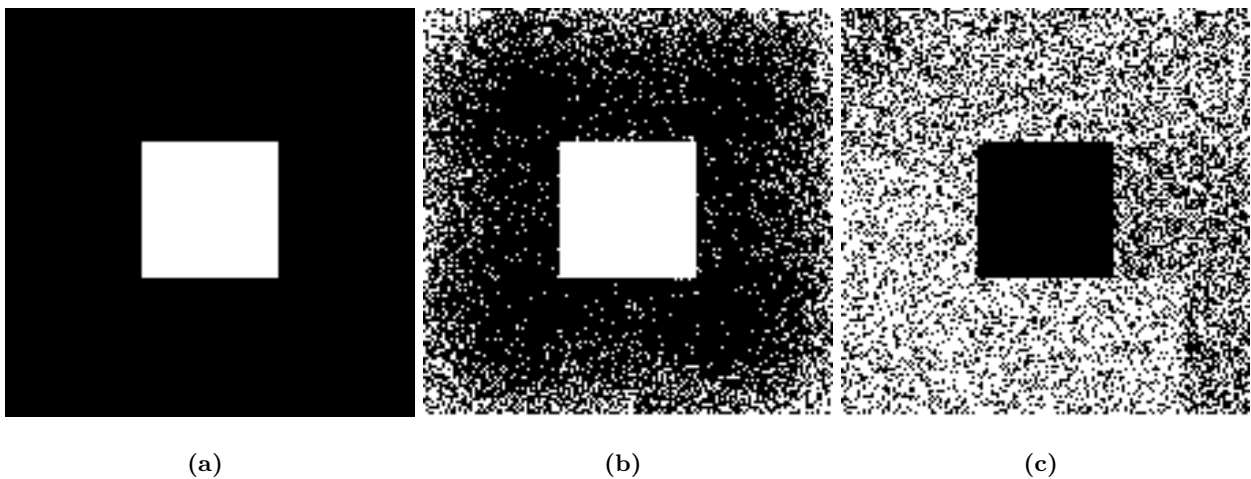


Figura 3.2: A la izquierda imagen I_1 binarizada con [Otsu, 1979](#) en medio binarizada con [Bradley and Roth, 2007](#) y a la derecha con [Molina et al., 2018](#)

En el capítulo 2 explicamos que en los algoritmos [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) se establece el parámetro w para determinar el tamaño de las ventanas utilizadas. La Figura 3.3 muestra la gráfica con exactitud de la imagen I_1 binarizada con [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) utilizando diferentes valores de w , con el objetivo de mostrar que al utilizar diferentes valores de w la exactitud en la imagen binarizada cambia por lo que elegir el tamaño adecuado es importante. En la gráfica se puede notar que con el algoritmo [Bradley and Roth, 2007](#) la exactitud más alta es de 82 % y con [Molina et al., 2018](#) la exactitud más alta es 29 %.

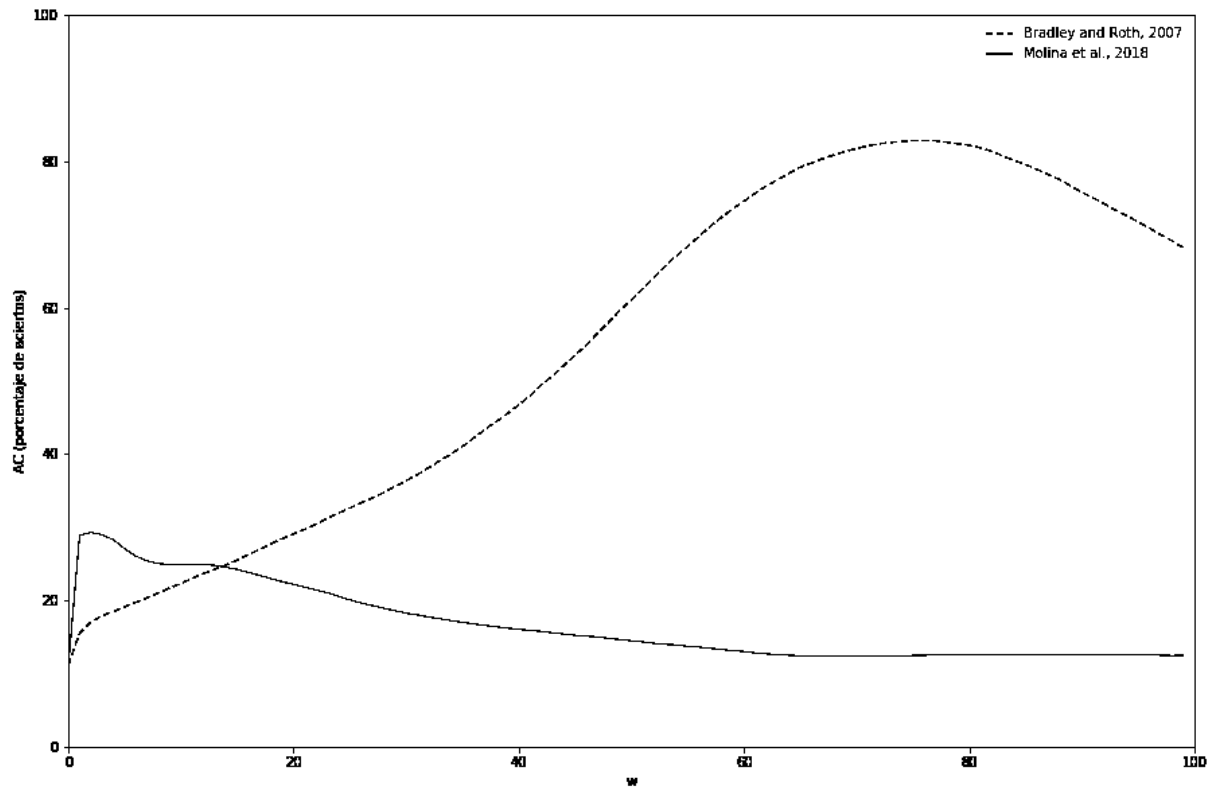


Figura 3.3: Exactitud obtenida al binarizar I_1 con Bradley and Roth, 2007 y Molina et al., 2018 utilizando diferentes valores de w

La Figura 3.4 a) muestra la imagen I_2 donde la clase 0 tiene media $\mu = 50$ con varianza $\sigma = 15$ y la clase 1 tiene media $\mu = 100$ con varianza $\sigma = 15$. La Figura 3.4 b) muestra la distribución de probabilidad de la Figura 3.4 a) donde podemos notar que no se distingue donde termina la clase 0 ni donde inicia la clase 1, esto se debe a que la varianza de las clases aumentó de $\sigma = 3$ a $\sigma = 15$.

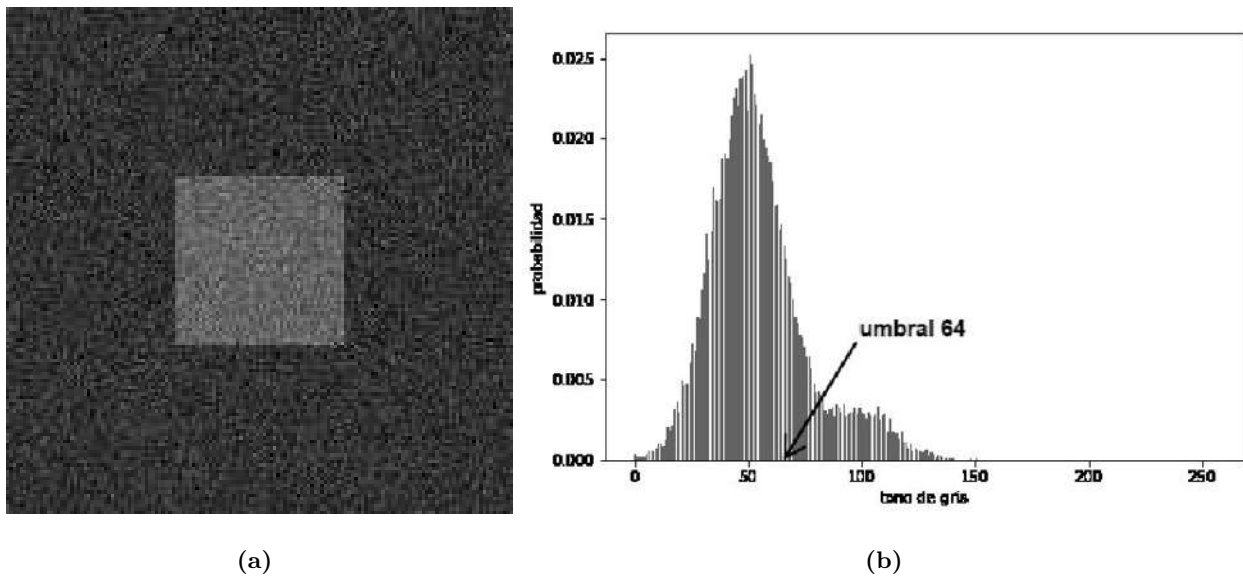


Figura 3.4: A la izquierda imagen I_2 y a la derecha su distribución de probabilidad

La Figura 3.5 a) muestra la imagen I_2 binarizada con el algoritmo [Otsu, 1979](#) la cual tiene 84.69% de exactitud. En la distribución de probabilidad mostrada en la Figura 3.4 b) señalamos el tono de gris 64 correspondiente al umbral que [Otsu, 1979](#) obtuvo para binarizar la imagen. La Figura 3.5 b) muestra la imagen I_2 binarizada con [Bradley and Roth, 2007](#) utilizando $\tau = 3$ y $w = 60$ la cual tiene 61.65% de exactitud. La Figura 3.5 c) muestra el resultado de binarizar I_2 con [Molina et al., 2018](#) utilizando $w = 80$ y $C = 80$ la cual tiene 41.58% de exactitud. Podemos notar que las exactitudes obtenida al binarizar I_2 con [Otsu, 1979](#), [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) es menor que la obtenida al binarizar la imagen I_1 con dichos algoritmos debido a que tienen dificultad para hacer una correcta binarización cuando las clases de la distribución de probabilidad de la imagen están traslapadas o no están definidas, como es el caso de la distribución de probabilidad de la imagen I_2 .

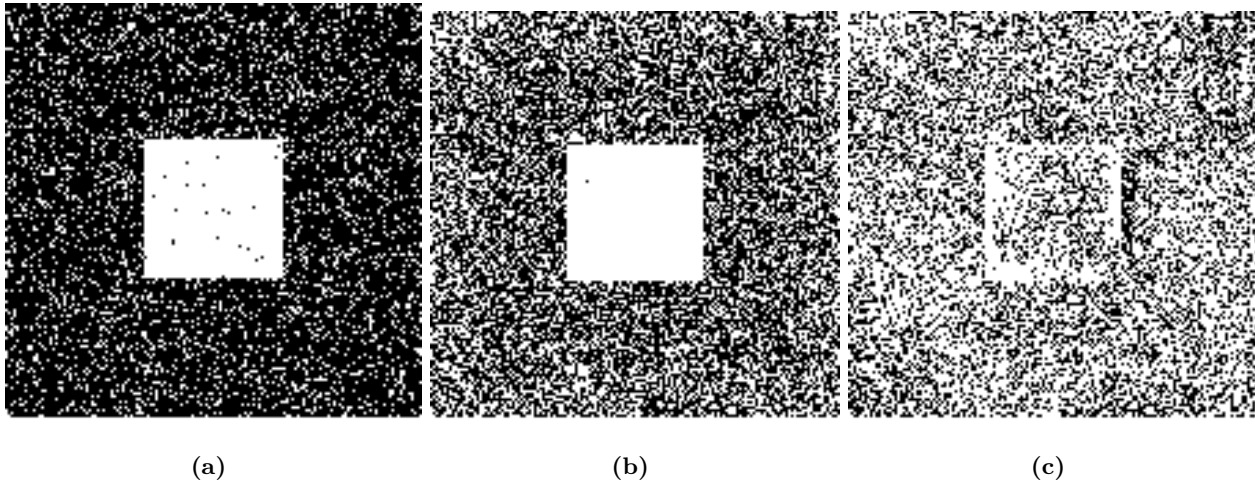


Figura 3.5: A la izquierda imagen I_2 binarizada con Otsu, 1979, en medio binarizada con Bradley and Roth, 2007 y a la derecha con Molina et al., 2018

La Figura 3.6 muestra la gráfica con la exactitud de la imagen I_2 binarizada utilizando diferentes valores del parámetro w empleando los algoritmos Bradley and Roth, 2007 y Molina et al., 2018. En la gráfica se puede notar que con Bradley and Roth, 2007 el valor máximo de exactitud es 62 % y con Molina et al., 2018 es 41 %, a diferencia de la gráfica mostrada en la Figura 3.3 obtenida al binarizar I_1 con diferentes valores de w con Bradley and Roth, 2007 y Molina et al., 2018 en la cual el valor máximo es 64 % de exactitud. Esta disminución de la exactitud se debe a que la imagen I_1 tiene varianza $\sigma = 3$ en las clases por lo que las clases están separadas y definidas y en la imagen I_2 la varianza es $\sigma = 15$ y las clases no se distinguen o se encuentran traslapadas.

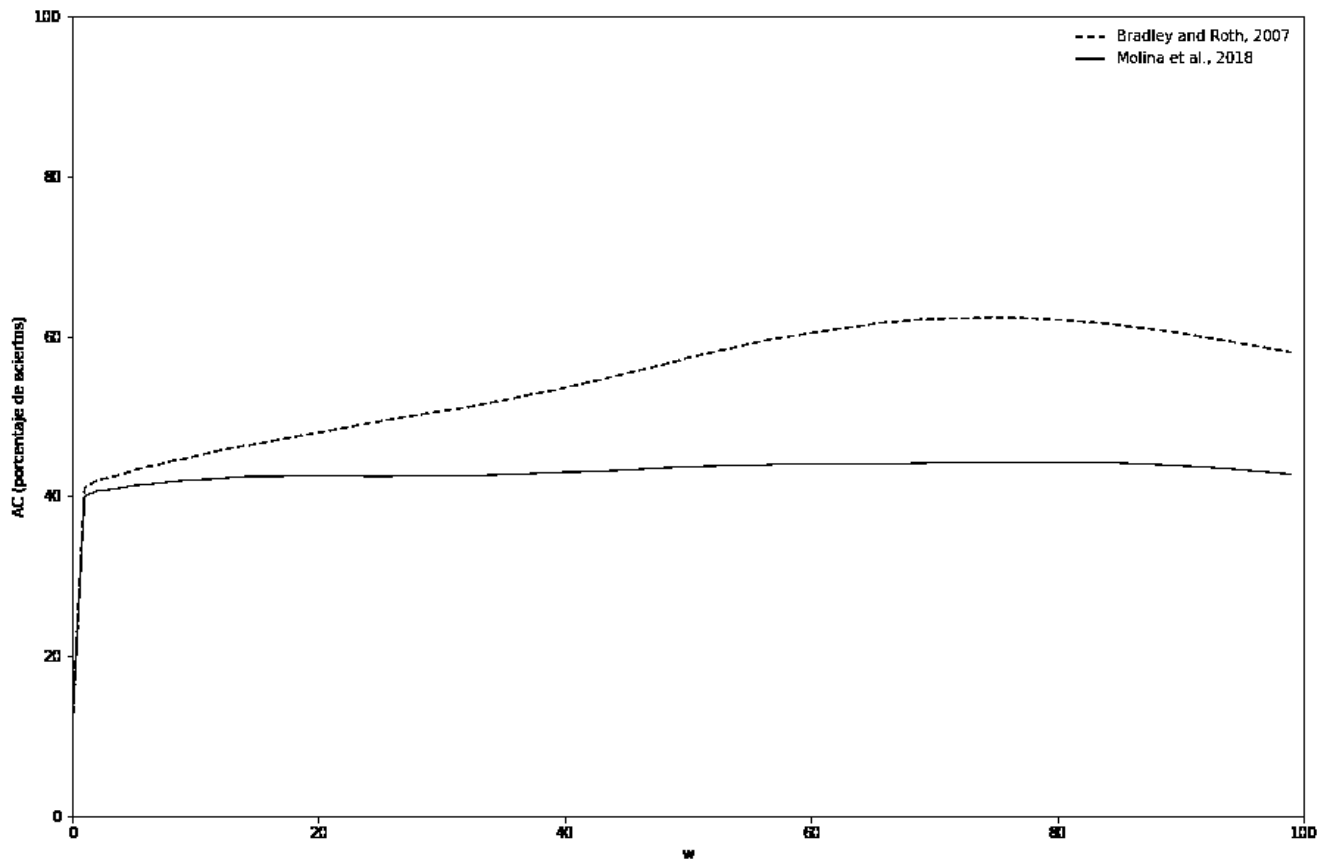


Figura 3.6: Exactitud obtenida al binarizar I_2 con Bradley and Roth, 2007 y Molina et al., 2018 utilizando diferentes valores de w

La Figura 3.7 a) muestra la imagen I_3 donde la clase 0 tiene media $\mu = 50$ con varianza $\sigma = 30$ y la clase 1 tiene media $\mu = 100$ y varianza $\sigma = 30$. La Figura 3.7 b) muestra la distribución de probabilidad de la Figura 3.7 a) en la que vemos que la clase 0 y la clase 1 no se distinguen debido al valor de sus medias y que la varianza de las clases es $\sigma = 30$, a diferencia de la distribución de probabilidad de la imagen I_1 donde la varianza es $\sigma = 3$ y vemos que las clases si están separadas.

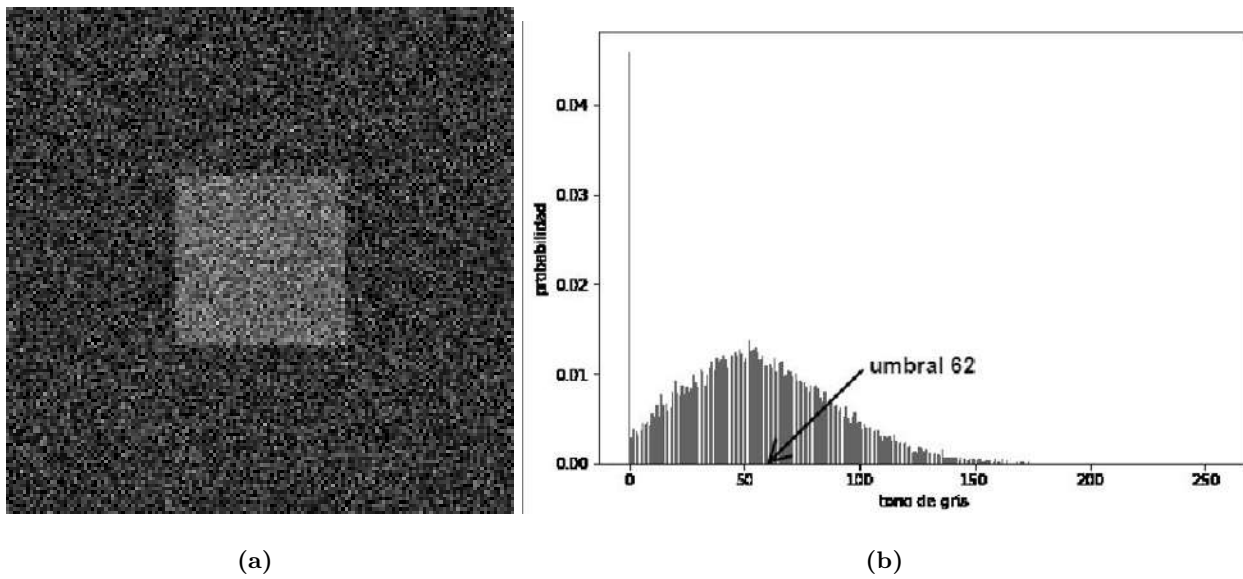


Figura 3.7: A la izquierda imagen I_3 y a la derecha su distribución de probabilidad

La Figura 3.8 a) muestra la imagen I_3 binarizada con el algoritmo [Otsu, 1979](#) la cual tiene 68.70% de exactitud utilizando un umbral correspondiente al tono de gris con valor 62 señalado en la distribución de probabilidad mostrada en la Figura 3.7 b). La Figura 3.8 b) muestra la imagen I_3 binarizada con [Bradley and Roth, 2007](#) utilizando $\tau = 3$ y $w = 50$ la cual tiene 52.56% de exactitud. La Figura 3.8 c) muestra la imagen I_3 binarizada con [Molina et al., 2018](#) utilizando $w = 80$ y $C = 80$ la cual tiene 43.85% de exactitud. Podemos ver que las tres imágenes binarizadas tienen exactitud menor a 69% debido a que la varianza de la clase 0 y la clase 1 de la distribución de probabilidad de I_3 es $\sigma = 30$ lo que genera que las clases se encuentren traslapadas.

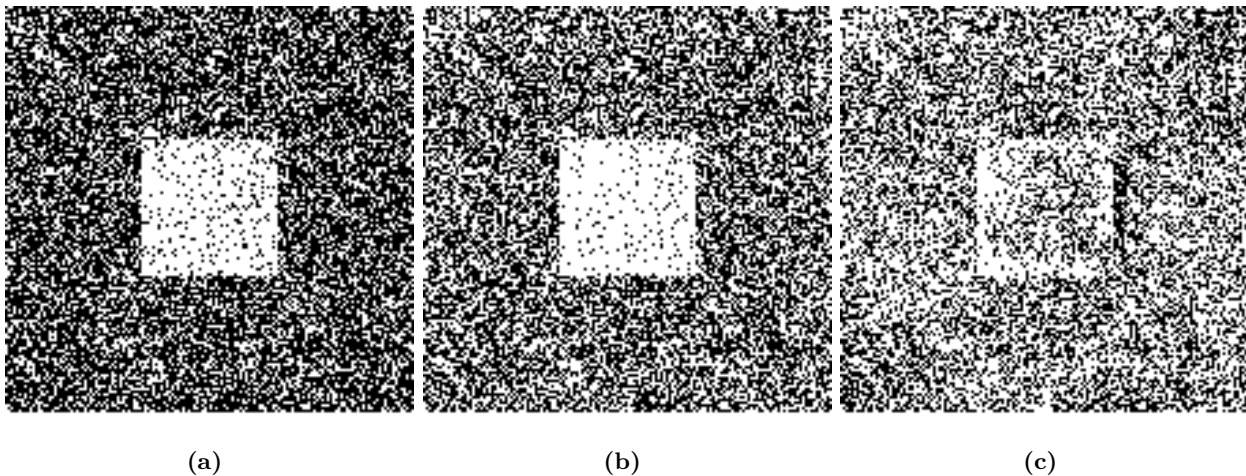


Figura 3.8: A la izquierda imagen I_3 binarizada el algoritmo de Otsu, 1979, en medio binarizada con Bradley and Roth, 2007 y a la izquierda con Molina et al., 2018

La Figura 3.9 muestra la gráfica de la exactitud obtenida al binarizar la imagen I_3 con los algoritmos Bradley and Roth, 2007 y Molina et al., 2018 utilizando diferentes valores del parámetro w . Se puede notar que con el algoritmo de Bradley and Roth, 2007 el valor máximo de exactitud que se obtiene es 52.77% y con Molina et al., 2018 es 43.87% de exactitud las cuales representan exactitudes bajas en una imagen binaria.

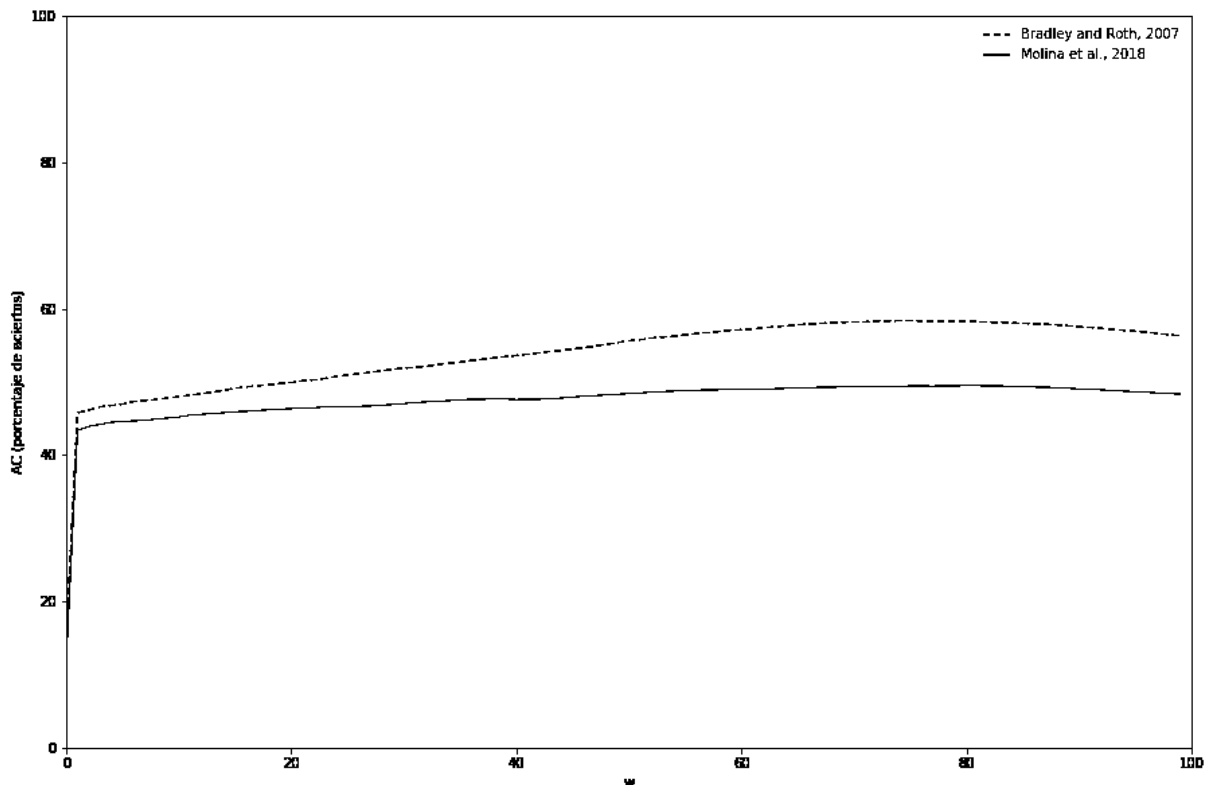


Figura 3.9: Exactitud obtenida al binarizar I_3 con los algoritmos Bradley and Roth, 2007 y Molina et al., 2018

En las Figuras 3.2, 3.5 y 3.8 que muestran las imágenes I_1 , I_2 e I_3 binarizadas con los algoritmos Otsu, 1979, Bradley and Roth, 2007 y Molina et al., 2018 vimos que la exactitud disminuye si en la distribución de probabilidad de la imagen las clases están traslapadas o no están definidas. Además, en los algoritmos que utilizan ventanas como Bradley and Roth, 2007 y Molina et al., 2018 si no se tiene la imagen de referencia para generar una gráfica, es difícil identificar cual será el valor w de ventana que conviene utilizar. En la siguiente sección describimos nuestra propuesta para calcular ventanas adaptables sujetas a respetar el cambio de iluminación de la imagen.

3.1. Cálculo eficiente del tamaño de la ventana

Para implementar nuestra propuesta de ventanas adaptables comenzamos analizando la distribución de probabilidad de una imagen en tono de gris que contenga información de letras sobre un fondo blanco. Podemos considerar que la distribución de probabilidad de una imagen de texto sobre fondo blanco es bimodal lo que significa que habrá una moda correspondiente a los píxeles de las letras la cual denominamos m_0 y otra correspondiente a los píxeles del fondo que denominamos m_1 . Para obtener m_0 y m_1 calculamos la distribución de probabilidad (D) de la imagen I dada y obtenemos el umbral (u) con el que separamos los píxeles que pertenecen al texto de los que pertenecen al fondo utilizando el método de [Otsu, 1979](#). Los píxeles de la imagen con valor inferior a u pertenecerán a la clase 0 (texto) y obtenemos m_0 mediante:

$$m_0 = \operatorname{argmax}_{0 \leq i \leq u} D(i) \quad (3.3)$$

y los píxeles con valor mayor o igual a u pertenecerán a la clase 1 (fondo) y obtenemos m_1 mediante:

$$m_1 = \operatorname{argmax}_{u < i \leq 255} D(i) \quad (3.4)$$

Definimos la función de calidad F_0 como la distancia de cada píxel de la imagen I a la moda m_0 la cual calculamos con (3.5) y F_1 como la distancia de cada píxel de la imagen I a la moda m_1 y la obtenemos con (3.6), de manera que podremos determinar a cual de las dos modas esta más cerca el valor del píxel $I[r, c]$. El algoritmo k-medias utiliza un concepto similar a las funciones de calidad que proponemos sin embargo en k-medias utilizan las medias y nosotros las modas para hacer más robusto el proceso.

$$F_0[r, c] = |I[r, c] - m_0| \quad (3.5)$$

$$F_1[r, c] = |I[r, c] - m_1| \quad (3.6)$$

Generamos la matriz binaria denominada mapa de decisión P en la cual asignamos el valor 0 en las coordenadas $[r, c]$ si la distancia $F_0[r, c]$ es menor a $F_1[r, c]$ de lo contrario asignamos 1 tal como:

$$P[r, c] = \begin{cases} 1 & F_0[r, c] > F_1[r, c] \\ 0 & \text{si no} \end{cases} \quad (3.7)$$

Para probar el desempeño de (3.7) lo aplicamos sobre la imagen I_3 con el objetivo de clasificar los píxeles en blancos y negros para lo cual utilizamos $m_0 = 52$ que obtuvimos con (3.3) y $m_1 = 63$

que obtuvimos con (3.4). La Figura 3.10 muestra el resultado de binarizar la imagen I_3 con (3.7) la cual tiene 69.65 % de exactitud.

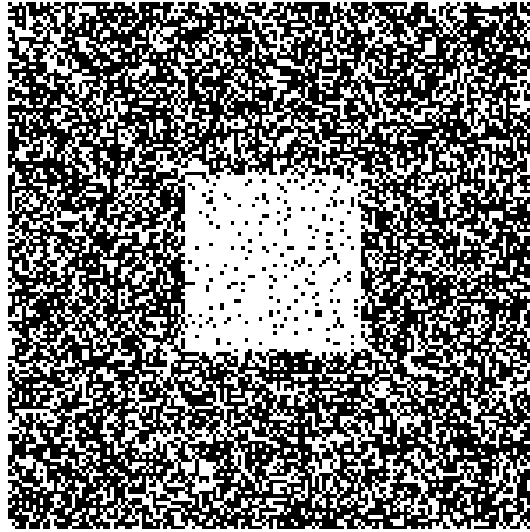


Figura 3.10: Imagen I_3 binarizada con (3.7)

Con el fin de introducir coherencia espacial y hacer más robusto (3.7) calculamos la suma (que es proporcional a calcular el promedio) del vecindario de $F_0[r, c]$ y $F_1[r, c]$ delimitado por una ventana de tamaño $(2w + 1) \times (2w + 1)$ centrada en las coordenadas $[r, c]$ tal como:

$$S_m[r, c] = \sum_{k=r-w}^{r+w} \sum_{l=c-w}^{c+w} F_m[k, l] \quad (3.8)$$

donde $m = \{0, 1\}$.

Bajo la aseveración de que vamos a calcular la suma, ahora el mapa de decisión P lo calculamos mediante:

$$P[r, c] = \begin{cases} 1 & S_0[r, c] > S_1[r, c] \\ 0 & \text{si no} \end{cases} \quad (3.9)$$

La Figura 3.11 a) muestra la imagen I_3 binarizada con (3.9) donde utilizamos $w = 1$ la cual tiene 76.52 % de exactitud que es mayor a la obtenida al binarizar la misma imagen pero sin utilizar la suma del vecindario. La Figura 3.11 b) muestra la imagen I_3 binarizada empleando $w = 2$ en la que se obtuvo 86.12 % de exactitud la cual es mayor a la obtenida con $w = 1$. La Figura 3.11 c) muestra la imagen I_3 binarizada empleando $w = 4$ con 94.39 % de exactitud y la Figura 3.11 d) muestra la imagen I_3 binarizada utilizando $w = 15$ la cual tiene 92.63 % de exactitud. En las cuatro imágenes binarizadas podemos observar que al ir aumentando el valor w la exactitud mejora, sin embargo si aumenta indefinidamente decrece debido a que se está redondeando el cuadro del centro porque

hay demasiada coherencia espacial es decir es un valor de coherencia espacial muy alto y deforma los bordes.

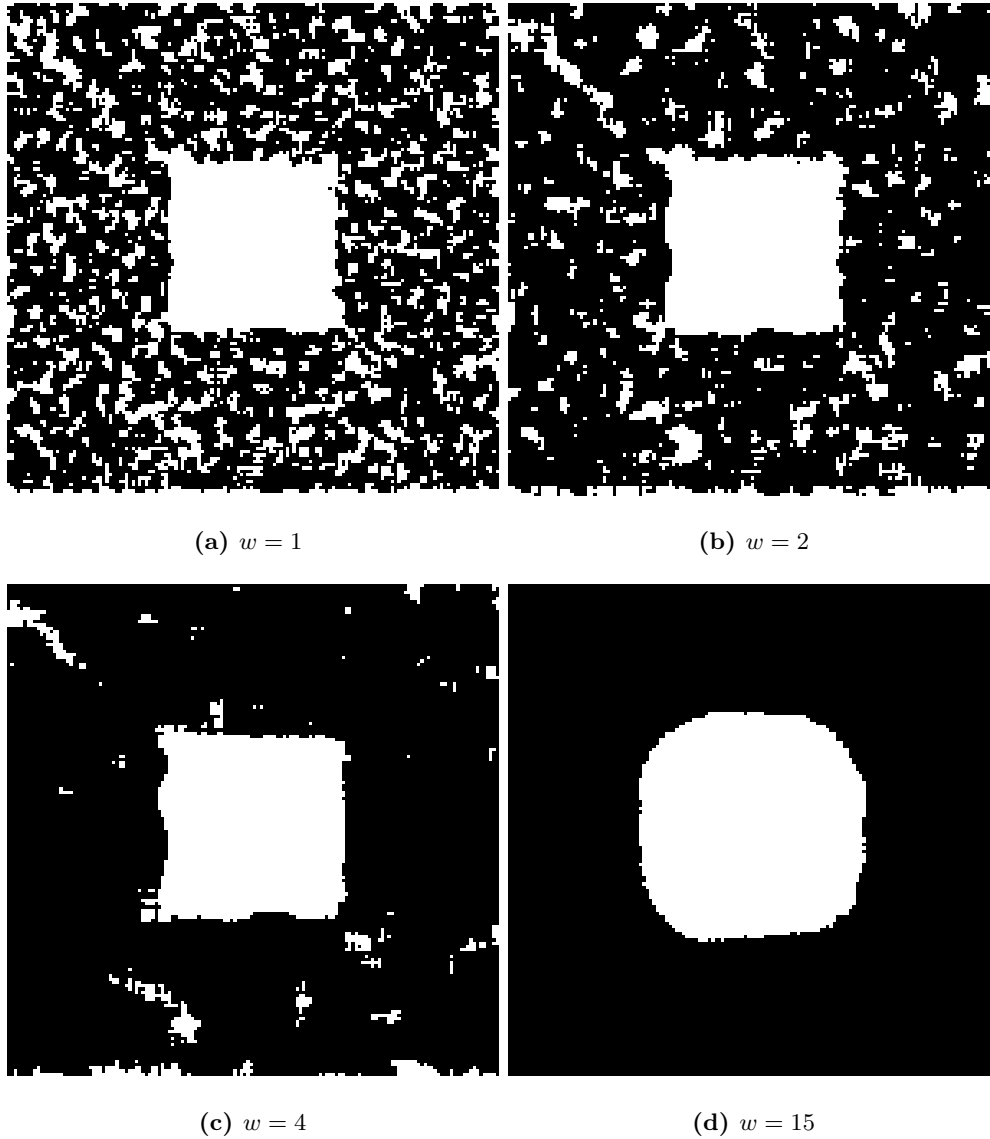


Figura 3.11: Binarización de I_3 utilizando (3.9) con diferentes valores de w

En la Figura 3.11 a) vemos que utilizando $w = 1$ se identificaron los bordes del cuadro del centro sin embargo en el área negra hay píxeles clasificados como blancos. Mientras que en la Figura 3.11 d) donde empleamos $w = 15$ los píxeles del área negra y blanca se clasificaron correctamente, sin embargo el cuadro del centro está redondeado. Por lo que conviene utilizar un valor w pequeño cerca de los bordes del cuadro del centro y un valor grande lejos de los bordes, esto se debe a que el promedio o la suma de los tonos de gris de los píxeles dentro de una ventana se ve afectado cuando la ventana contiene píxeles del área oscura y clara, de modo que si el promedio representa el elemento

del centro de la ventana se clasificara como elemento de la clase a la que no pertenece. La Figura 3.12 muestra la gráfica con la exactitud de la imagen I_3 binarizada con (3.9) utilizando diferentes valores de w . En la gráfica vemos que con $w = 5$ se obtiene la mayor exactitud y que desde $w = 2$ a $w = 20$ las exactitudes son mayores a 90%, las cuales son mayores a las obtenidas al binarizar la misma imagen I_3 utilizando diferentes valores de w con los algoritmos Otsu, 1979, Bradley and Roth, 2007 y Molina et al., 2018.

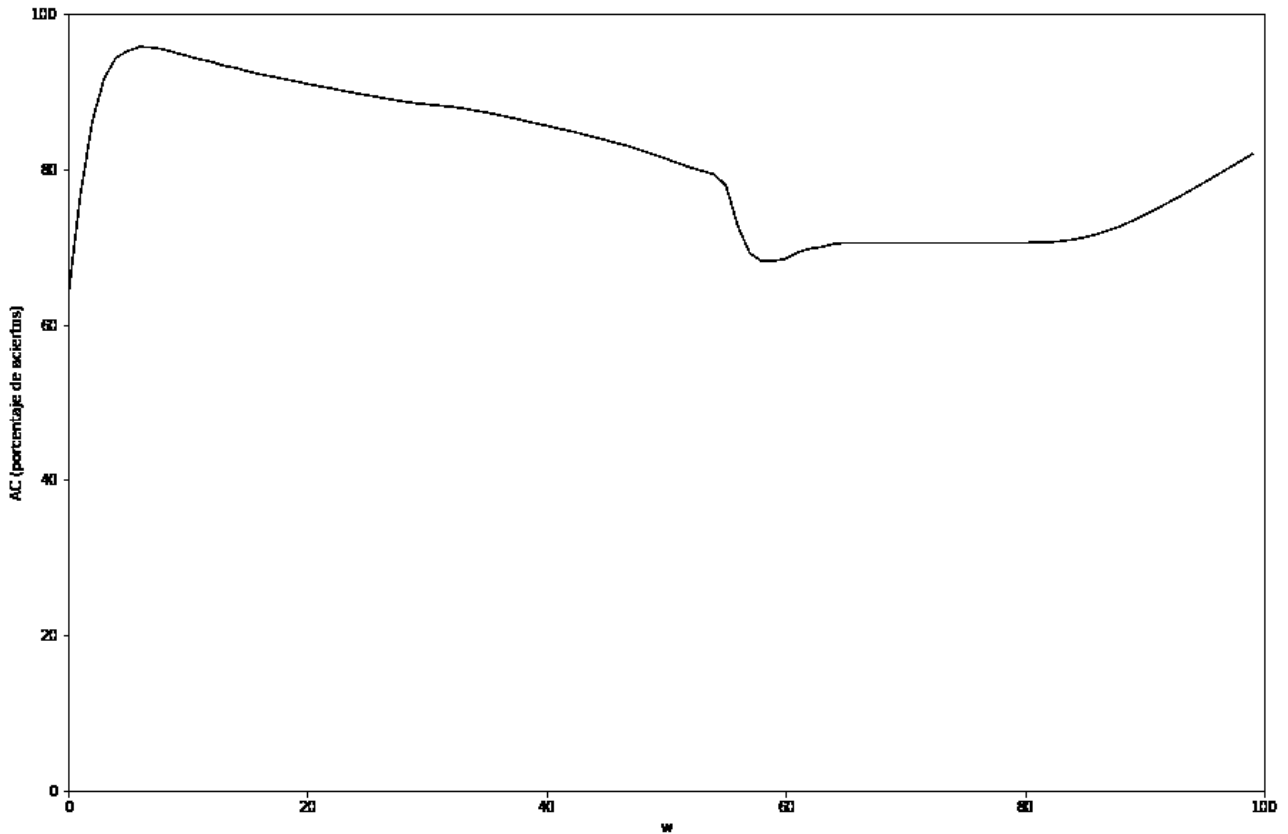


Figura 3.12: Gráfica de la exactitud de I_3 binarizada con (3.9) utilizando diferentes valores de w

En la Figura 3.11 vimos que utilizar ventanas con el fin de obtener la suma para clasificar un píxel como blanco o negro es mejor que hacer la comparación a nivel píxel, y nos dimos cuenta que se obtiene mayor exactitud al utilizar el tamaño de las ventanas de manera que no se mezclen los píxeles de áreas claras y oscuras. Para calcular automáticamente el valor w adecuado en cada coordenada de la imagen I dada comenzamos definiendo un mapa de decisión P el cual obtenemos mediante 3.9 con el que calculamos un mapa de bordes B , con el objetivo de identificar los píxeles donde ocurre el cambio de iluminación de la región clara y oscura. A continuación mostramos la forma de obtener el mapa de bordes y el cálculo automático de w .

Mapa de bordes

Determinamos que existe un borde en el mapa de decisión en las coordenadas $P[r, c]$ si existe algún valor adyacente diferente a él, por lo que en el mapa de bordes denotado como B asignamos el valor 1 en las coordenadas $[r, c]$, de lo contrario asignamos el valor 0 tal como:

$$B[r, c] = \begin{cases} 1 & \text{si } P[r, c] \neq P[r-1, c] \vee P[r, c] \neq P[r, c-1] \vee P[r, c] \neq P[r+1, c-1] \\ 0 & \text{si no} \end{cases} \quad (3.10)$$

Con el objetivo de que las ventanas sean grandes en áreas de la imagen donde no hay cambio de iluminación pero que al mismo tiempo abarquen únicamente píxeles del área oscura o clara, maximizamos el tamaño de las ventanas y limitamos a que tengan un número máximo de bordes. Determinamos que las ventanas pueden tener un valor mínimo de 0 y máximo de w_{max} con el propósito de que sean de tamaño acotado. Representamos el área de la ventana centrada en las coordenadas $B[r, c]$ como $A(w[r, c]) = (2w + 1) \times (2w + 1)$ y denotamos con T al número de bordes dentro de la ventana. Para calcular el valor w asociado a $A(w[r, c])$ de manera que se maximice el área de la ventana sujeta a la restricción de que no sobrepase determinada cantidad de bordes T , optimizamos la función:

$$\begin{aligned} w[r, c] &= \underset{w}{\operatorname{argmax}} A(w) \\ &\text{sujeto a} \\ &0 \leq w \leq w_{max} \\ &\sum_{k=r-w}^{r+w} \sum_{l=c-w}^{c+w} B[k, l] < T \end{aligned} \quad (3.11)$$

Para hacer eficiente la búsqueda del tamaño w en cada una de las coordenadas del mapa de decisión optimizando la función (3.11), proponemos utilizar búsqueda binaria y emplear imágenes integrales [Viola and Jones, 2001](#) (descritas en el Capítulo 2) para calcular la sumatoria las cuales tienen complejidad constante debido a que solamente se realizan cuatro operaciones para calcular la suma de los píxeles dentro de una ventana independientemente del tamaño.

Búsqueda binaria

Es un algoritmo de búsqueda que encuentra la posición de un valor en un array ordenado el cual almacena una serie de elementos del mismo tipo. La búsqueda binaria inicia comparando el

valor objetivo con el elemento ubicado a la mitad del array, si los valores coinciden se regresa la posición del elemento ubicado a la mitad de lo contrario se elimina la mitad de los elementos en la que el valor objetivo no puede estar, es decir si el valor objetivo es mayor que el elemento del centro la mitad inferior es eliminada por lo que la búsqueda continua en la mitad superior y el proceso se repite hasta que la posición del valor objetivo es encontrado [Lin et al., 2019](#). Para calcular el tamaño de las ventanas utilizando búsqueda binaria el valor objetivo es que se cumplan las restricciones de la función a optimizar (3.11) y la lista ordenada de elementos donde se realiza la búsqueda es el rango 0 a w_{max} . El cálculo del tamaño óptimo de la ventana en cada coordenada $[r, c]$ de la imagen lo denominamos algoritmo de Ventana Óptima (VO) el cual se muestra en el Algoritmo 6 y recibe el mapa de decisión P , el número máximo de bordes T y el tamaño máximo de ventana w_{max} . En las líneas 1 a 5 se calcula la matriz de bordes B . En la línea 6 se calcula la imagen integral y en las líneas 7 a 22 se realiza la búsqueda binaria en cada una de las coordenadas de la imagen. En la línea 23 se regresa la matriz W con el tamaño óptimo de ventana en cada una de las coordenadas de la imagen.

Algoritmo 6: VO (P, T, w_{max})

```

1  para  $r = 0$  a  $N_r - 1$  hacer
2  |   para  $c = 0$  a  $N_c - 1$  hacer
3  |   |   Calcular  $B[r, c]$  con (3.10)
4  |   fin
5  fin
6   $H_B[r, c] = B[r, c] + H_B[r - 1, c] + H_B[r, c - 1] - H_B[r - 1, c - 1] \forall [r, c]$ 
7  para  $r = 0$  a  $N_r - 1$  hacer
8  |   para  $c = 0$  a  $N_c - 1$  hacer
9  |   |   inicio  $\leftarrow 0$ 
10  |   |   fin  $\leftarrow w_{max}$ 
11  |   |   hacer
12  |   |   |    $w \leftarrow \frac{inicio+fin}{2}$ 
13  |   |   |   Calcular  $S[r, c]$  empleando  $H_B[r, c]$  con (2.20)
14  |   |   |   si  $S[r, c] > T$  entonces
15  |   |   |   |   fin  $\leftarrow w$ 
16  |   |   |   en otro caso
17  |   |   |   |   inicio  $\leftarrow w$ 
18  |   |   |   fin
19  |   |   mientras fin-inicio  $> 1$ ;
20  |   |    $W[r, c] \leftarrow w$ 
21  |   fin
22 fin
23 Resultado  $W$ 

```

Utilizamos el algoritmo VO para calcular ventanas de tamaño óptimo en la imagen I_3 mostrada en la Figura 3.7 a) empleando $T = 10$, $\tau = 1$ y $w_{max} = 16$. Obtenemos F_0 con (3.5) y F_1 con (3.6) utilizando $m_0 = 52$ y $m_1 = 63$. La Figura 3.13 a) muestra el mapa de bordes de la Figura 3.7 a) obtenido con el algoritmo VO. La Figura 3.13 b) muestra una imagen en la que se representan los tamaños de las ventanas calculadas alrededor de los bordes en la cual vemos ventanas pequeñas cerca de los bordes y grandes lejos de los bordes. La Figura 3.13 c) muestra la imagen I_3 binarizada con (3.9) utilizando los tamaños de ventana calculados con VO la cual tiene 98.94% de exactitud.

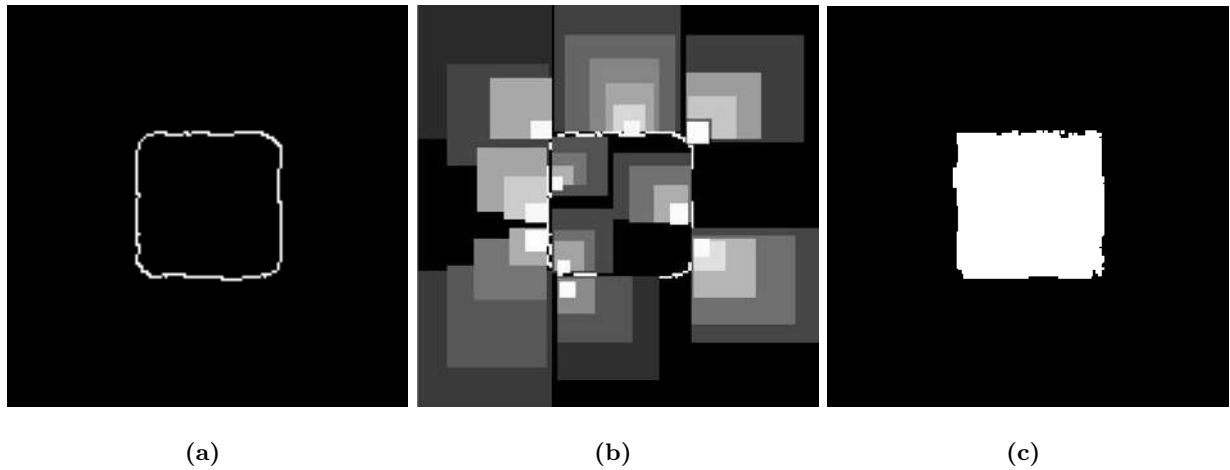


Figura 3.13: A la izquierda se muestra el mapa de bordes de la imagen I_3 , en el centro el tamaño de las ventanas alrededor de los bordes y a la derecha la imagen binarizada

En las siguientes secciones presentamos los algoritmos denominados Binarización de Imágenes con Promedios Ponderados (BIPP) y Binarización de Imágenes con Ventanas Adaptables (BIVA) los cuales utilizan VO para calcular ventanas de tamaño óptimo.

3.2. Binarización de Imágenes con Promedios Ponderados

En esta sección describimos el algoritmo BIPP (Binarización de Imágenes con Promedios Ponderados) el cual emplea VO (Ventana Óptima) con el propósito de obtener una imagen fusionada sin áreas oscuras a partir de imágenes inversamente iluminadas. Otro propósito de BIPP al utilizar el algoritmo VO es emplear las ventanas de tamaño óptimo para binarizar la imagen fusionada con [Bradley and Roth, 2007](#) de manera que las ventanas utilizadas ya no sean de tamaño constante. BIPP inicia calculando la imagen inversamente iluminada I_1 de la imagen original I_0 de acuerdo a [Rafael C. Gonzalez, 2002](#), quien explica que una imagen puede ser caracterizada por la multiplicación de dos componentes básicos: la cantidad de iluminación incidente en una escena y la cantidad de iluminación reflejada por los objetos en dicha escena. Estos dos componentes los llaman luminancia (L_0) y reflectancia (R_0) que al multiplicarse generan la imagen tal como $I_0[r, c] = L_0[r, c]R_0[r, c]$. El componente de luminancia lo obtenemos como lo hace [Molina et al., 2018](#) utilizando el operador morfológico de cerradura (denotado por \bullet) mediante (2.33). Con el componente de luminancia calculamos la iluminación inversa mediante $L_1[r, c] = 255 - L_0[r, c]$ que corresponderá al componente de luminancia de la imagen I_1 . El componente de reflectancia lo obtenemos con el cociente de la imagen original en escala de grises y su luminancia tal como $R_0[r, c] = \frac{I_0[r, c]}{L_0[r, c]}$ el cual será el mismo

en la imagen inversamente iluminada debido a que dicho componente contiene la información de reflexión con la que están formados los caracteres del texto, por lo tanto la imagen inversamente iluminada la obtenemos como $I_1[r, c] = R_0[r, c]L_1[r, c]$. La Figura 3.14 a) muestra una imagen con iluminación no uniforme denotada como I_0 y la Figura 3.14 b) muestra su componente de iluminación utilizando un kernel morfológico de tamaño 80. La Figura 3.14 c) muestra la iluminación inversa de la Figura 3.14 a).

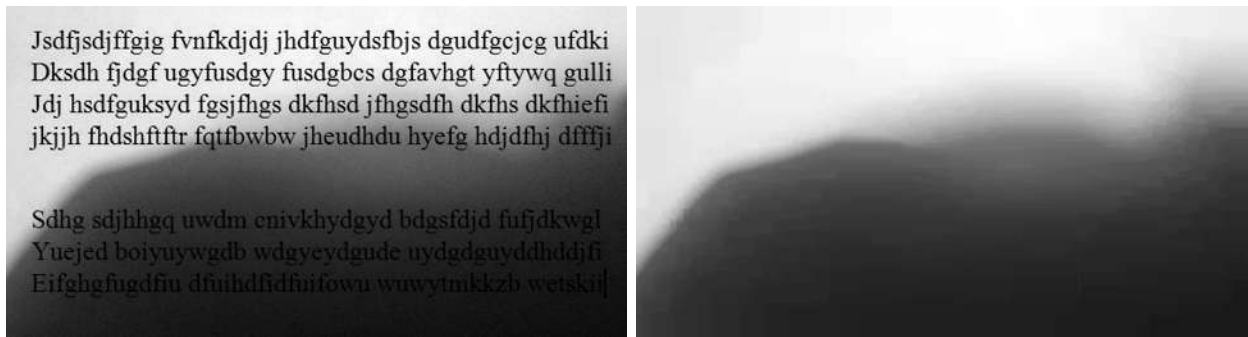
(a) Imagen con iluminación no uniforme I_0 (b) Componente de iluminación de I_0 (c) iluminación inversa de I_0

Figura 3.14: Imagen con iluminación no uniforme, su componente de iluminación y su iluminación inversa

La Figura 3.15 muestra la imagen inversamente iluminada de la Figura 3.14 a) representada como la imagen I_1 en la que podemos ver que el área clara de I_0 es oscura en I_1 y el área oscura de I_0 es clara en I_1 .

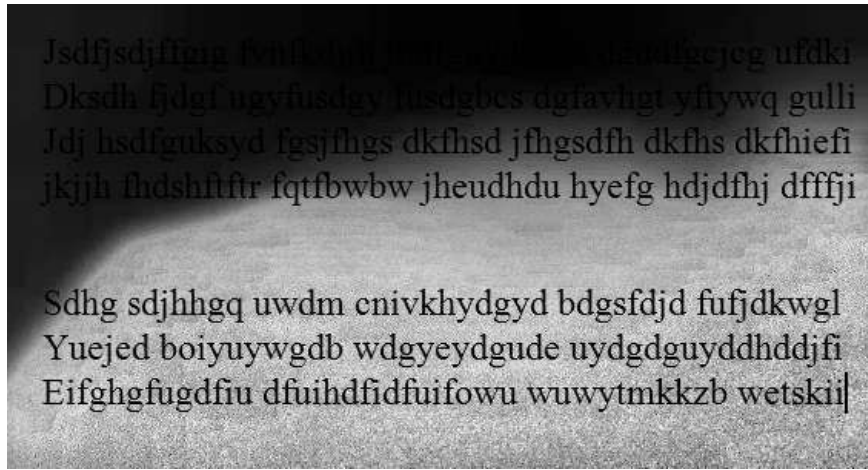


Figura 3.15: Imagen inversamente iluminada de la Figura 3.14 a)

La forma en que el algoritmo VO permite obtener una imagen fusionada con áreas claras es calculando ventanas sujetas a respetar los bordes donde ocurre el cambio de iluminación en las imágenes inversamente iluminadas I_0 e I_1 , con las que podremos determinar de manera más precisa los píxeles que conformarán la imagen fusionada. BIPP calcula los promedios de las imágenes inversamente iluminadas y los emplea como funciones de calidad F_0 y F_1 con el objetivo de medir la iluminación de cada imagen. Con las funciones de calidad VO calcula un mapa de decisión P para identificar las regiones claras de I_0 e I_1 . El mapa de decisión P tendrá el valor 0 en las coordenadas $[r, c]$ si el promedio del vecindario del píxel $I_0[r, c]$ es más claro que el del píxel $I_1[r, c]$ de lo contrario tendrá el valor 1. El algoritmo VO obtiene P mediante (2.28) donde $S[r, c]$ es la suma (proporcional al promedio) del vecindario de $\Delta F[r, c] = F_0[r, c] - F_1[r, c]$ la cual obtiene utilizando imágenes integrales Viola and Jones, 2001 por lo que se calcula la matriz integral $H_F[r, c]$ de $\Delta F[r, c]$ mediante (2.19) y de esta manera obtenemos $S[r, c]$ con (2.20). Para delimitar el vecindario de $\Delta F[r, c]$ se utilizan las ventanas calculadas con VO que están sujetas a respetar los bordes del cambio de iluminación. El algoritmo VO se realiza de forma iterativa con el objetivo de que las ventanas y el mapa de decisión se ajusten en cada iteración, el algoritmo termina con un mapa de decisión más preciso calculado con ventanas de tamaño óptimo. El mapa de decisión P de la última iteración del algoritmo VO lo utilizamos para calcular la imagen fusionada mediante $I_f[r, c] = I_0[r, c]P[r, c] + I_1[r, c](1 - P[r, c])$ y las ventanas de tamaño óptimo las empleamos para hacer la binarización de la imagen. La Figura 3.16 a) muestra una imagen en la que cada coordenada tiene el valor w correspondiente a un tono de gris calculado con VO utilizando la Figura 3.14 a) como I_0 y la Figura 3.15 como I_1 empleando $T = 15$ y $w_{max} = 80$. En la imagen vemos un área más oscura debido a que los valores de w son mas pequeños y un valor pequeño de w equivale a un tono de gris casi negro en cambio un valor

alto corresponde al color blanco. La Figura 3.16 b) muestra el mapa de decisión P calculado con las ventanas de la Figura 3.16 a) en el cual el área blanca representa el área clara de la imagen I_0 y el color negro el área clara de la imagen I_1 . La Figura 3.16 c) muestra la imagen fusionada con áreas claras de I_0 e I_1 utilizando el mapa de decisión de la Figura 3.16 b).

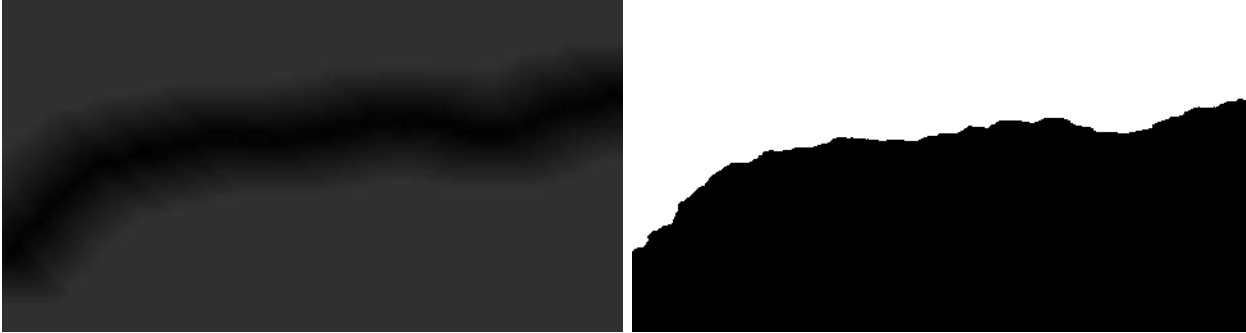
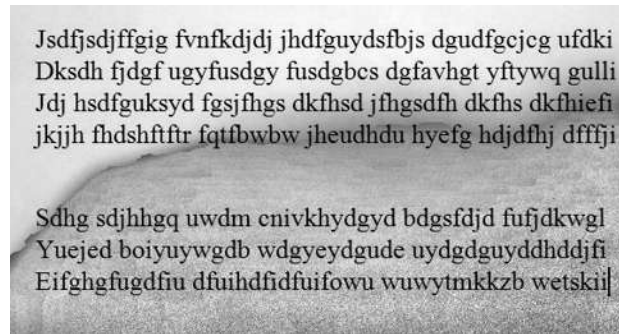
(a) *ventanas calculadas con VO*(b) *mapa de decisión*(c) *imagen fusionada*

Figura 3.16: Ventanas calculadas con VO, mapa de decisión y la imagen fusionada

Además de obtener la imagen fusionada otro de los propósitos de BIPP al utilizar VO es emplear las ventanas adaptables obtenidas en la última iteración para binarizar la imagen con [Bradley and Roth, 2007](#). De manera que la binarización de la imagen fusionada se hará con el Algoritmo 3 descrito en el Capítulo 2 utilizando las ventanas sujetas a respetar los bordes del cambio de iluminación. La Figura 3.17 muestra un diagrama con los procesos que realiza BIPP y la imagen que se obtiene en cada uno de ellos, en el paso 1 se recibe la imagen original y se asignan los parámetros tamaño máximo de ventanas w_{max} , τ , tolerancia T y kernel morfológico C . El paso 2 calcula la imagen inversamente iluminada y el 3 las funciones de calidad F_0 y F_1 . En el 4 se calculan las ventanas de tamaño óptimo, en el 5 el mapa de decisión y se realizan iterativamente con el algoritmo VO. El paso 6 calcula la imagen fusionada y el paso 7 binariza la imagen con [Bradley and Roth, 2007](#) utilizando la imagen fusionada y las ventanas de tamaño óptimo.

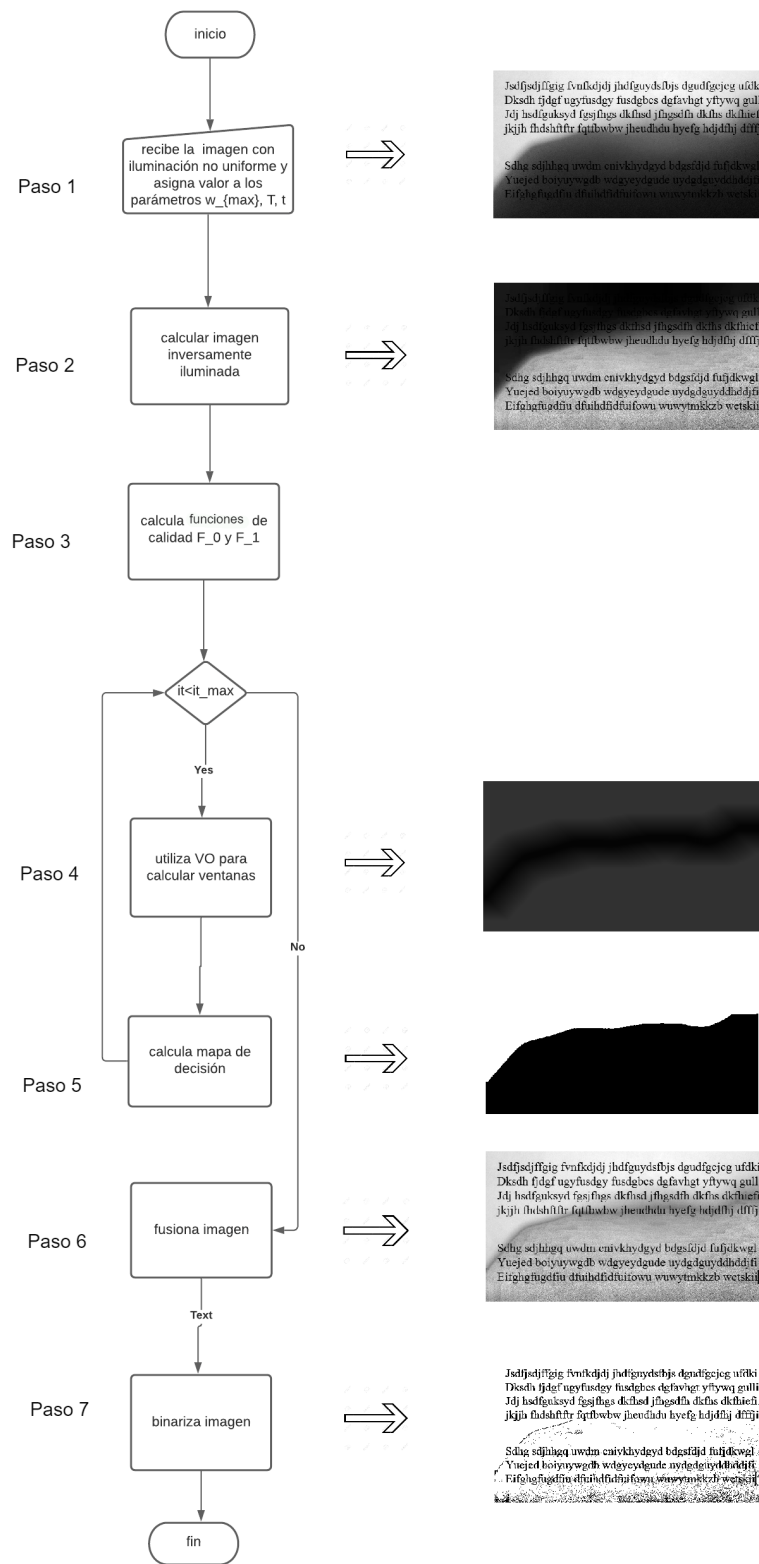


Figura 3.17: Diagrama con los procesos de BIPP y las imágenes que va calculando en cada uno de ellos

BIPP se muestra en el Algoritmo 7 con las operaciones que se realizan en cada uno de los pasos mostrados en el diagrama de la Figura 3.17. El Algoritmo 7 recibe la imagen I_0 , C , w_{max} , T y τ que corresponde al paso 1. En las líneas 1 a 9 se calcula la imagen con iluminación inversa y $\Delta F[r, c]$ correspondiente al paso 2. En la línea 10 se calcula la imagen integral de la matriz ΔF . De las líneas 11 a 23 se realiza el proceso iterativo para calcular el mapa de decisión P y el tamaño de las ventanas correspondientes al paso 4 y 5. En la línea 24 se calcula la imagen fusionada I_f correspondiente al paso 6 y en la línea 25 se calcula la imagen binarizada utilizando las ventanas adaptables con [Bradley and Roth, 2007](#) correspondiente al paso 7.

Algoritmo 7: BIPP(I_0, C, w_{max}, T, τ)

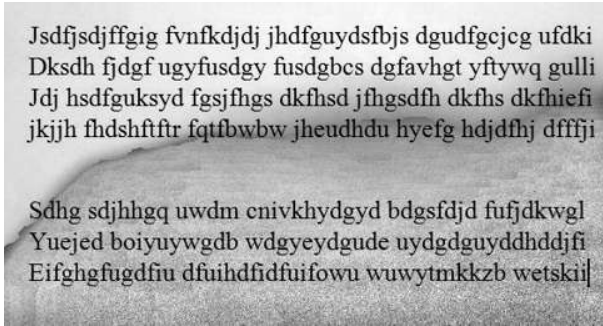
```

1  Calcular la luminancia  $L_0$  de la imagen  $I_0$ 
2  para  $r = 0$  a  $N_r - 1$  hacer
3      para  $c = 0$  a  $N_c - 1$  hacer
4           $R[r, c] \leftarrow \frac{I_0[r, c]}{L_0[r, c]}$ 
5           $L_1[r, c] \leftarrow 255 - L_0[r, c]$ 
6           $I_1[r, c] \leftarrow R[r, c]L_1[r, c]$ 
7           $\Delta F[r, c] \leftarrow I_0[r, c] - I_1[r, c]$ 
8      fin
9  fin
10  $H_F[r, c] = \Delta F[r, c] + H_F[r - 1, c] + H_F[r, c - 1] - H_F[r - 1, c - 1] \forall [r, c]$ 
11 para  $it = 0$  a  $Max$  hacer
12      $W \leftarrow VO(P^{(it)}, T, w_{max})$  Algoritmo 6
13     para  $r = 0$  a  $N_r - 1$  hacer
14         para  $c = 0$  a  $N_c - 1$  hacer
15              $S[r, c] =$ 
16                  $H_F[r+w, c+w]H_F[r+w, c-w-1]H_F[r-w-1, c+w] + H_F[r-w-1, c-w-1]$ 
17             si  $S[r, c] < 0$  entonces
18                  $P^{(it)}[r, c] \leftarrow 1$ 
19             en otro caso
20                  $P^{(it)}[r, c] \leftarrow 0$ 
21         fin
22     fin
23 fin
24  $I_f[r, c] \leftarrow I_0[r, c]P^{(it)}[r, c] + I_1[r, c](1 - P^{(it)}[r, c])$ 
25  $I_B \leftarrow$  Algoritmo 3: BradleyRoth( $I_f, \tau, W$ )
26 Regresa  $I_B$ 

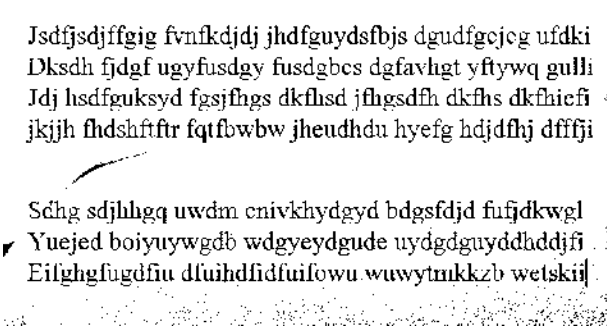
```

En la Figura 3.18 comparamos los resultados de la Figura 3.14 a) binarizada con Bradley and Roth, 2007 y con Molina et al., 2018 debido a que ambos utilizan ventanas y calculan una imagen fusionada sin embargo BIPP lo hace utilizando ventanas adaptables. La Figura 3.18 a) muestra la imagen fusionada y la Figura 3.18 b) muestra la imagen binarizada obtenidas con BIPP

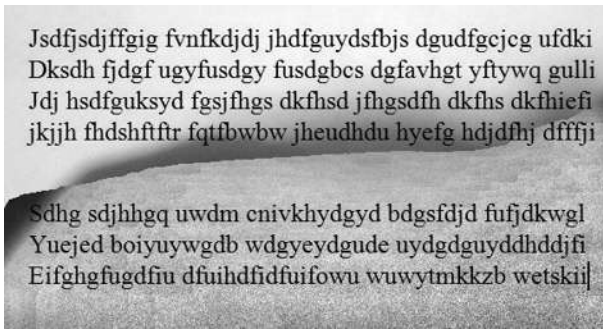
utilizando $w_{max} = 80$, $\tau = 30$, $T = 15$ y $C = 80$. La Figura 3.18 c) y la Figura 3.18 d) muestra la imagen fusionada y la imagen binarizada con Molina et al., 2018 empleando $C = 80$ y $w = 80$. Se puede ver que en la imagen fusionada con BIPP la iluminación es más uniforme que la fusionada con Molina et al., 2018 y la imagen binarizada con BIPP no tiene áreas oscuras.



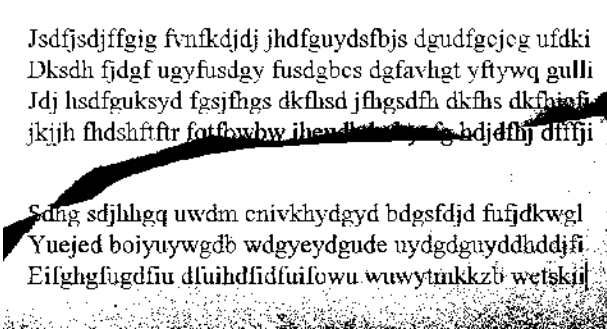
(a) imagen fusionada



(b) Imagen binarizada con BIPP



(c) Imagen binarizada con Molina et al., 2018



(d) Imagen binarizada con Molina et al., 2018

Figura 3.18: Ejemplo que muestra una imagen binarizada con BIPP y la misma imagen binarizada con Molina et al., 2018

3.3. Binarización de Imágenes con Ventanas Adaptables

En esta sección presentamos el algoritmo BIVA (Binarización de Imágenes con Ventanas Adaptables) el cual al igual que BIPP utiliza VO para calcular el tamaño de las ventanas, sin embargo BIVA calcula funciones de calidad utilizando un criterio de decisión parecido al que k -medias emplea para clasificar los píxeles de la imagen. Las funciones de calidad calcularán una medida de iluminación de la imagen por lo que contendrán información del área clara y oscura, con el propósito de que el algoritmo VO obtenga ventanas de tamaño óptimo para binarizar la imagen con Bradley and Roth, 2007. BIVA inicia calculando las funciones de calidad F_0 y F_1 analizando la distribución de probabilidad de la imagen original I tal como lo hacemos al inicio de la sección

3.1. párrafo 2 en (3.5) y (3.6), donde mencionamos que si la imagen I contiene información de letras sobre un fondo blanco podemos considerar que su distribución de probabilidad es bimodal. La Figura 3.19 a) muestra una imagen con texto sobre una hoja y la Figura 3.19 b) muestra su distribución de probabilidad bimodal, en la cual existe una moda m_0 que corresponde al valor que más se repite en los píxeles que pertenecen a las letras que tienen un valor de 0 a 20 en la escala de grises, y una moda m_1 que corresponde al valor que más se repite en los píxeles que pertenecen al fondo que tienen valor de 140 a 250 en la escala de grises.

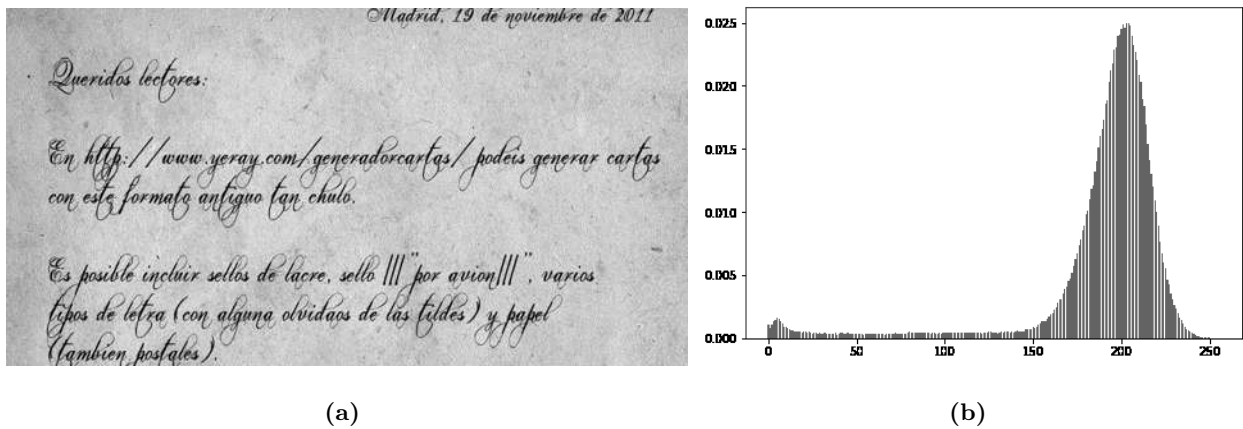


Figura 3.19: A la izquierda imagen con texto y la derecha su distribución de probabilidad

Definimos la función de calidad F_0 como la distancia del valor de los píxeles de la imagen I a la moda m_0 y la calculamos como $F_0[r, c] = |I[r, c] - m_0|$ y la función de calidad F_1 como la distancia del valor de los píxeles de I a la moda m_1 y la calculamos como $F_1[r, c] = |I[r, c] - m_1|$. Generamos un mapa de decisión P donde asignamos el valor 0 a las coordenadas $[r, c]$ si la distancia $F_0[r, c]$ es menor a $F_1[r, c]$ de lo contrario tendrán el valor 1, con el objetivo de identificar el área clara de la imagen con el color blanco y el área oscura con negro. Para hacer más robusto el cálculo del mapa de decisión obtenemos las sumas (que son proporcionales al promedio) del vecindario de cada coordenada $[r, c]$ de las funciones de calidad F_0 y F_1 con el objetivo de tener coherencia espacial. Una manera eficiente de calcular las sumas para obtener el mapa de decisión es restando las funciones de calidad $\Delta F[r, c] = F_0[r, c] - F_1[r, c]$ y obtener la suma del vecindario representada como $S[r, c]$, si el signo de dicha suma es positivo a $P[r, c]$ se asigna el valor 0 de lo contrario el valor 1. Para calcular las sumas utilizamos imágenes integrales como lo propone Viola and Jones, 2001 por lo que obtenemos la matriz $H_F[r, c]$ mediante (2.19) que representa la imagen integral de $\Delta F[r, c]$ y calculamos la suma $S[r, c]$ con (2.20) como lo describimos en el Capítulo 2 sección 2.3. La Figura 3.20 a) muestra una imagen con iluminación no uniforme con un área clara y una oscura.

La Figura 3.20 b) muestra el mapa de decisión donde el color blanco representa el área clara de la Figura 3.20 a) y el color negro el área oscura.

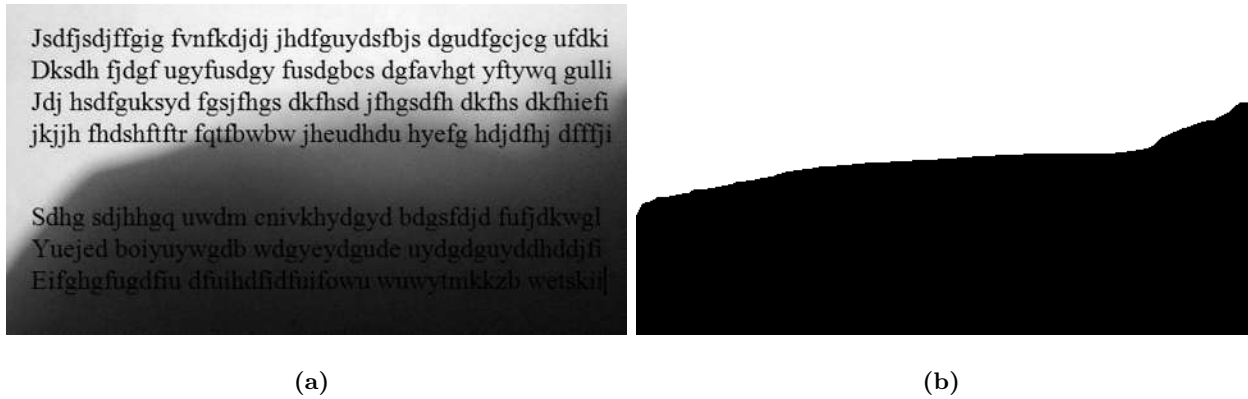


Figura 3.20: Ejemplo de imagen con iluminación no uniforme y su mapa de decisión

El objetivo del mapa de decisión es identificar el área clara y oscura de la imagen con iluminación no uniforme de manera que podamos utilizarlo para calcular ventanas sujetas a respetar los bordes del cambio de iluminación utilizando el algoritmo Ventana Óptima (VO). Para mostrar el proceso iterativo de VO hicimos un ejemplo empleando la Figura 3.20 donde mostramos el mapa de decisión y las ventanas calculadas en cada una de las iteraciones. La Figura 3.21 a) muestra el mapa de decisión inicial en el que se asigna el valor 0 en todas las coordenadas y la Figura 3.21 b) muestra las ventanas calculadas utilizando $w_{max} = 50$ y $T = 10$, en la cual se ha calculado el valor máximo que pueden tener las ventanas debido a que el mapa de decisión solamente tiene un área y el tamaño máximo corresponde al tono de gris con valor 50. La segunda iteración calcula el mapa de decisión con las ventanas obtenidas en la primera iteración, dicho mapa de decisión se muestra en la Figura 3.21 c) con el cual se obtienen las ventanas de tamaño óptimo mostradas en la Figura 3.21 d) donde vemos un área con color más oscuro correspondiente al borde del cambio de iluminación de la imagen, debido a que en dicha área las ventanas son pequeñas y un valor pequeño de ventana corresponde a un tono de gris casi negro. La Figura 3.21 e) y la Figura 3.21 f) muestran el mapa de decisión y ventanas calculados en la última iteración.

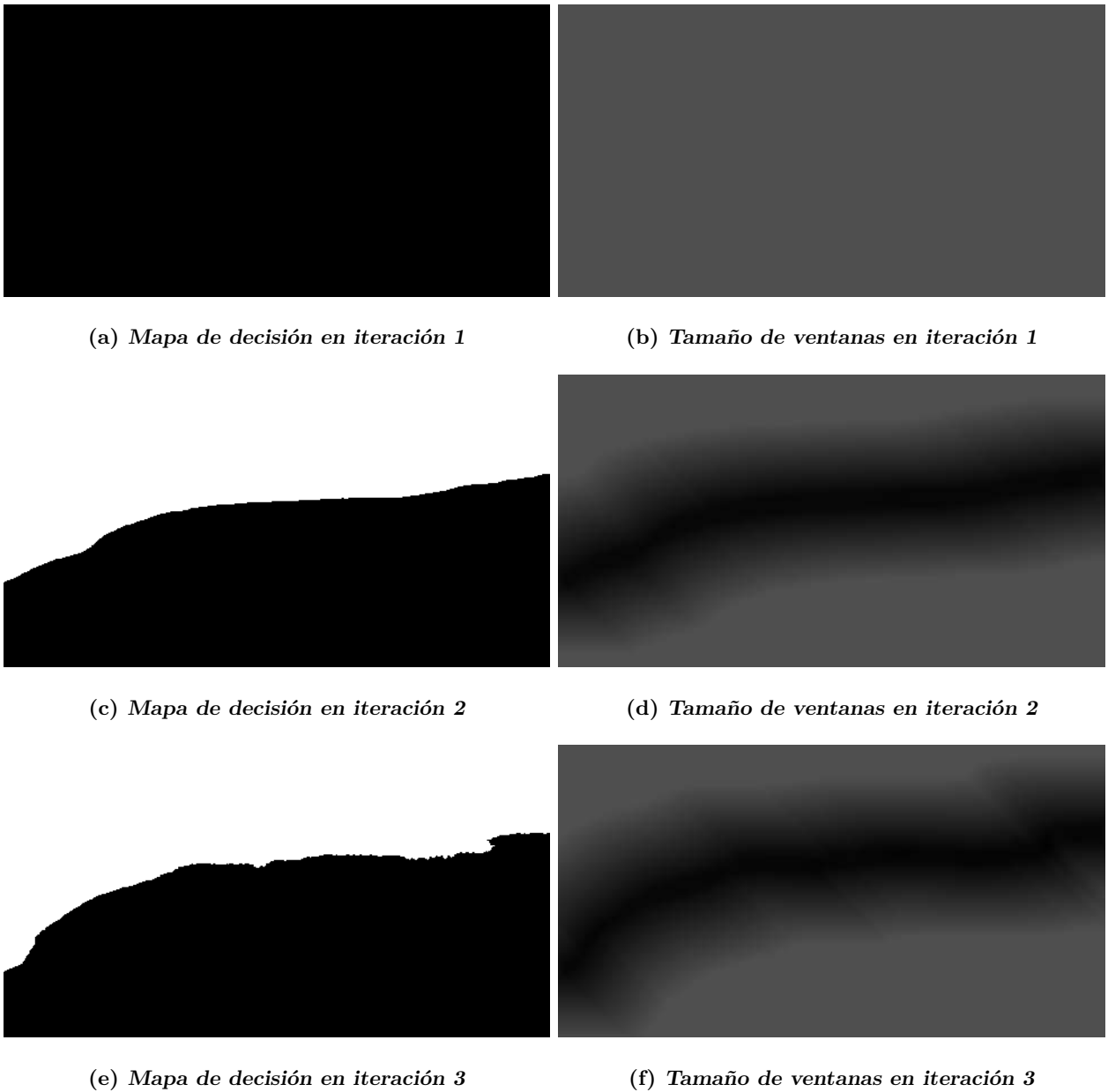


Figura 3.21: Mapa de decisión y ventanas calculadas con VO de la Figura 3.7

Binarizamos la imagen con iluminación no uniforme con las ventanas calculadas en la última iteración del algoritmo VO utilizando [Bradley and Roth, 2007](#) mostrado en el Algoritmo 3, el cual recibirá la imagen original y las ventanas de tamaño óptimo. La Figura 3.22 muestra un diagrama con los procesos que realiza BIVA y la imagen que se genera en cada uno de ellos. El paso 1 corresponde a recibir la imagen y asignar valor a los parámetros tamaño máximo de las ventanas w_{max} , τ y tolerancia T . El paso 2 calcula las funciones de calidad F_0 y F_1 . El paso 3 corresponde a calcular las ventanas, el 4 calcula el mapa de decisión y se realizan de manera iterativa con el algoritmo VO. En el paso 5 se binariza la imagen con [Bradley and Roth, 2007](#) utilizando las ventanas

de tamaño óptimo.

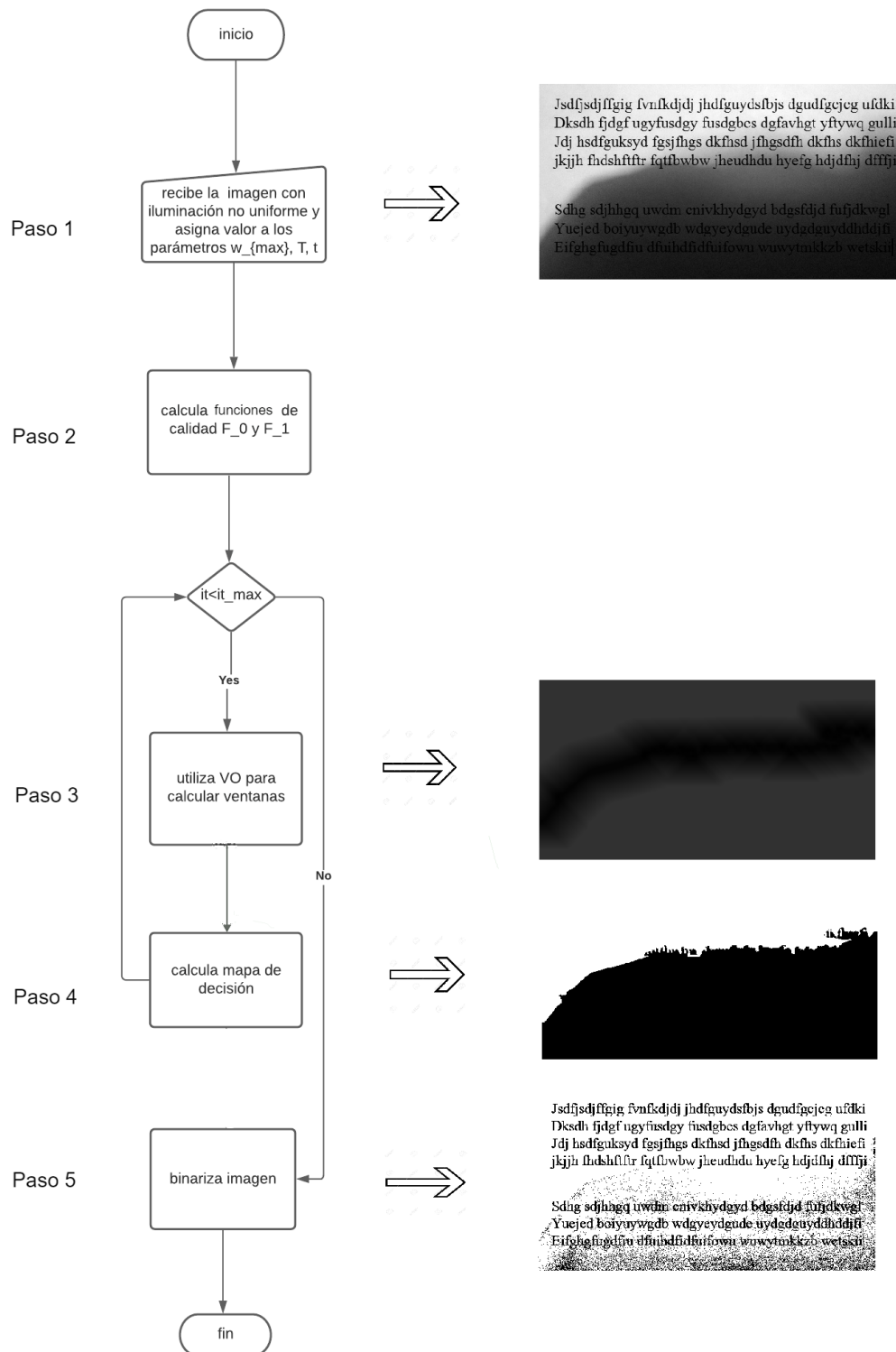


Figura 3.22: Diagrama con los procesos que realiza BIVA y la imagen que obtiene en cada uno de ellos

BIVA se muestra en el Algoritmo 8 con las operaciones realizadas en cada proceso del diagrama de la Figura 3.22. El Algoritmo 8 recibe la imagen I , w_{max} , T y τ correspondiente al paso 1. En las líneas 4 y 5 se obtienen las funciones de calidad del paso 2, en la línea 6 se calcula la diferencia entre las funciones de calidad y en la línea 7 se inicializa el mapa de decisión. En las líneas 11 a 23 se realiza el procedimiento iterativo de VO correspondiente al paso 3 y 4. En la línea 24 se obtiene la imagen binarizada utilizando [Bradley and Roth, 2007](#) con los tamaños de ventana calculados por BIVA correspondiente al paso 5.

Algoritmo 8: BIVA(I, w_{max}, T, τ)

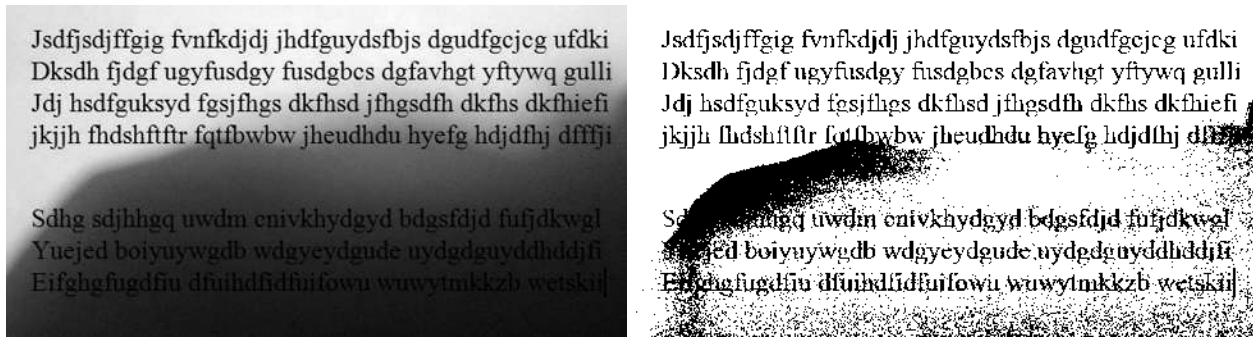
```

1  calcular  $m_1$  y  $m_2$  de  $I$ 
2  para  $r = 0$  a  $N_r - 1$  hacer
3      para  $c = 0$  a  $N_c - 1$  hacer
4           $F_0[r, c] = m_0 - I[r, c]$ 
5           $F_1[r, c] = m_1 - I[r, c]$ 
6           $\Delta F[r, c] = F_0[r, c] - F_1[r, c]$ 
7           $P[r, c] = 1$ 
8      fin
9  fin
10  $H_F[r, c] = \Delta F[r, c] + H_F[r - 1, c] + H_F[r, c - 1] - H_F[r - 1, c - 1] \forall [r, c]$ 
11 para  $it = 0$  a  $Max$  hacer
12      $W \leftarrow VO(P^{(it)}, T, w_{max})$  Algoritmo 6
13     para  $r = 0$  a  $N_r - 1$  hacer
14         para  $c = 0$  a  $N_c - 1$  hacer
15              $S[r, c] =$ 
16                  $H_F[r+w, c+w]H_F[r+w, c-w-1]H_F[r-w-1, c+w] + H_F[r-w-1, c-w-1]$ 
17             si  $S[r, c] < 0$  entonces
18                  $P^{(it)}[r, c] \leftarrow 1$ 
19             en otro caso
20                  $P^{(it)}[r, c] \leftarrow 0$ 
21         fin
22     fin
23 fin
24  $I_B \leftarrow$  Algoritmo 3:BradleyRoth( $I_f, \tau, W$ )
25 Regresa  $I_B$ 

```

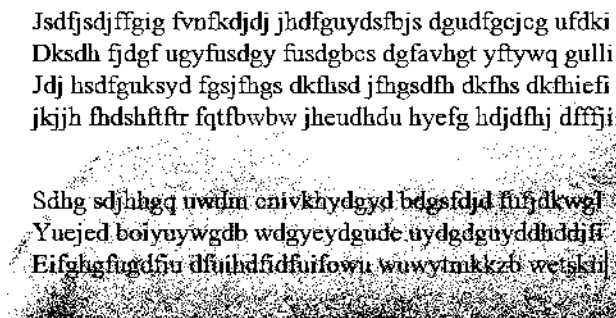
La Figura 3.23 a) muestra una imagen de texto con iluminación no uniforme con el propósito de binarizarla con el algoritmo BIVA el cual utiliza ventanas adaptables y con Bradley and Roth, 2007 que utiliza ventanas de tamaño constante. La Figura 3.23 b) muestra la imagen binarizada con el algoritmo Bradley and Roth, 2007 empleando $w = 50$ y $\tau = 10$ y la Figura 3.23 c) muestra la imagen binarizada con BIVA donde $T = 10$, $\tau = 10$ y $w_{max} = 50$. Como se puede ver,

BIVA ha tenido un mejor desempeño al binarizar la imagen con iluminación no uniforme.



(a) Imagen con iluminación no uniforme

(b) Imagen binarizada con Bradley and Roth, 2007



(c) Imagen binarizada con BIVA

Figura 3.23: Imagen con iluminación no uniforme binarizada con BIVA y Bradley and Roth, 2007

3.4. Conclusión

En este capítulo presentamos el algoritmo VO para calcular ventanas adaptables que pueden ser utilizadas para resolver el problema de clasificación tal como lo mostramos en la Figura 3.13, donde vemos que VO clasifica los píxeles con mayor exactitud que los algoritmos del estado del arte Otsu, 1979, Molina et al., 2018 y Bradley and Roth, 2007. BIVA es un algoritmo de binarización robusta de imágenes que calcula ventanas de tamaño óptimo empleando VO y funciones de calidad que miden la distancia de los píxeles de la imagen a las modas del texto y del fondo con el objetivo de detectar el área clara y oscura de la imagen. BIVA utiliza las ventanas calculadas con VO para binarizar la imagen con Bradley and Roth, 2007. En la Figura 3.23 mostramos que BIVA obtiene mejores resultados al binarizar una imagen con iluminación no uniforme que Bradley and Roth, 2007, esto se debe a que en BIVA utilizamos ventanas adaptables sujetas a respetar los bordes del cambio de iluminación. BIPP es otro algoritmo que presentamos para binarización robusta de imágenes en el cual utilizamos las ventanas calculadas con VO con dos propósitos, el primero es

calcular una imagen fusionada con regiones claras adecuada para binarización a partir de imágenes inversamente iluminadas, y el segundo propósito es utilizar las ventanas calculadas con el algoritmo VO para binarizar la imagen fusionada con el algoritmo de [Bradley and Roth, 2007](#). En la Figura [3.18](#) mostramos una imagen con iluminación no uniforme binarizada con BIPP y [Molina et al., 2018](#), vimos que la exactitud es mayor en la binarizada con BIPP por lo que es una buena idea calcular la imagen inversamente iluminada de la imagen original para identificar las áreas claras en ambas imágenes, con el propósito de obtener una imagen fusionada y binarizarla utilizando las ventanas de tamaño óptimo con [Bradley and Roth, 2007](#). En el siguiente capítulo mostramos resultados de BIPP, BIVA y algoritmos del estado del arte utilizando diferentes imágenes con iluminación no uniforme sintética y real.

Capítulo 4

Resultados

En este capítulo mostramos resultados de los algoritmos BIVA y BIPP presentados en esta tesis los cuales hacen uso de nuestra propuesta VO (Ventana Óptima). Las imágenes que utilizamos tienen diferentes patrones de iluminación no uniforme sintética y real. Presentamos los resultados de forma cuantitativa mediante gráficas y tablas utilizando diferentes métricas de calidad (descritas en el Capítulo 1.2.1) y de forma cualitativa mostrando imágenes binarizadas.

4.1. Imágenes con iluminación no uniforme real y sintética

Las imágenes que utilizamos las obtuvimos de [Molina et al., 2018](#) las cuales consisten en imágenes con iluminación no uniforme sintética y real. La Figura 4.1 muestra la imagen a partir de la que fueron generadas las imágenes.

Jsdjfsdjffgig fvnfkddj jhdfguydsfbjs dgudfgcjcg ufdki
Dksdh fjdgf ugyfusdgy fusbgbcs dgfavhgt yftywq gulli
Jdj hsdfguksyd fgsjfhgs dkfhsd jfhgsdfh dkfhs dkfhiefi
jkjjh fhdshftfr fqtfbwbw jheudhdu hyefg hdjdfhj dfffji

Sdhg sdjhhgq uwdm enivkhydgyd bdgsfdjd fufjdkwgl
Yuejed boiyuywgdb wdgzeydgude uydgdguyddhddjfi
Eifghgfugdfiu dfuihdfidfuifowu wuwytmkkzb wetskij

Figura 4.1: Imagen empleada para aplicar distintos patrones de iluminación no uniforme

Para generar las imágenes con iluminación sintética se aplicó iluminación no uniforme con la herramienta Adobe Photoshop CS6 sobre la Figura 4.1 usando diferentes niveles de intensidad y exposición. La Figura 4.2 muestra las 15 imágenes sintéticas generadas con distintos patrones de iluminación no uniforme.



Figura 4.2: Imágenes con diferentes patrones de iluminación sintéticas no uniforme

La Figura 4.3 muestra 30 imágenes con diferentes patrones de iluminación real no uniforme sobre la imagen de la Figura 4.1 .



Figura 4.3: Imágenes con iluminación no uniforme generada a partir de iluminación real

4.2. Resultados cuantitativos mostrados mediante gráficas

En esta sección presentamos resultados de forma cuantitativa de las imágenes sintéticas y reales binarizadas con BIVA, BIPP, [Molina et al., 2018](#) utilizando [Otsu, 1979](#) para binarizar la imagen, [Molina et al., 2018](#) utilizando [Bradley and Roth, 2007](#) para binarizar la imagen, [Otsu, 1979](#) y [Bradley and Roth, 2007](#). Mostramos los resultados en gráficas de líneas y de barras utilizando las métricas de calidad Porcentaje de Aciertos (AC), F-score, PSNR y NRM de manera que los resultados se puedan visualizar de distintas formas.

Resultados de imágenes con iluminación sintética

Las Figuras 4.4 y 4.5 muestran gráficas de líneas utilizando diferentes métricas de calidad para mediar la calidad de las imágenes sintéticas binarizadas. El eje vertical representa la métrica de calidad que se utilizó, el eje horizontal el número de la imagen sintética y en la parte inferior izquierda se muestra la línea que representa el algoritmo con el que se binarizaron las imágenes.

La Figura 4.4 a) muestra una gráfica de líneas con el porcentaje de aciertos calculado mediante (3.2) de las 15 imágenes sintéticas binarizadas con los diferentes algoritmos. De las 15 imágenes binarizadas con Otsu, 1979 12 tienen porcentaje de aciertos menor a 60 % y solamente 3 tienen mayor a 60 %. Con el algoritmo Molina et al., 2018 utilizando Otsu, 1979 para binarizar, una imagen tiene 71.24 % de porcentaje de aciertos y las 14 restantes tienen entre 84 % y 98 %. Al binarizar las imágenes con Molina et al., 2018 utilizando Bradley and Roth, 2007 2 imágenes tienen exactitud menor a 90 % y las 13 restantes tienen porcentaje de aciertos mayor a 90 %. Con el algoritmo Bradley and Roth, 2007 6 imágenes tienen porcentaje de aciertos entre 60 % y 90 % y 9 imágenes tienen porcentaje de aciertos mayores a 90 %. Con el algoritmo BIPP las 15 imágenes tienen porcentaje de aciertos entre 98 % y 100 %. Con BIVA solamente una tiene 96.7 % de porcentaje de aciertos y las 14 restantes tienen porcentaje de aciertos mayor a 98 %.

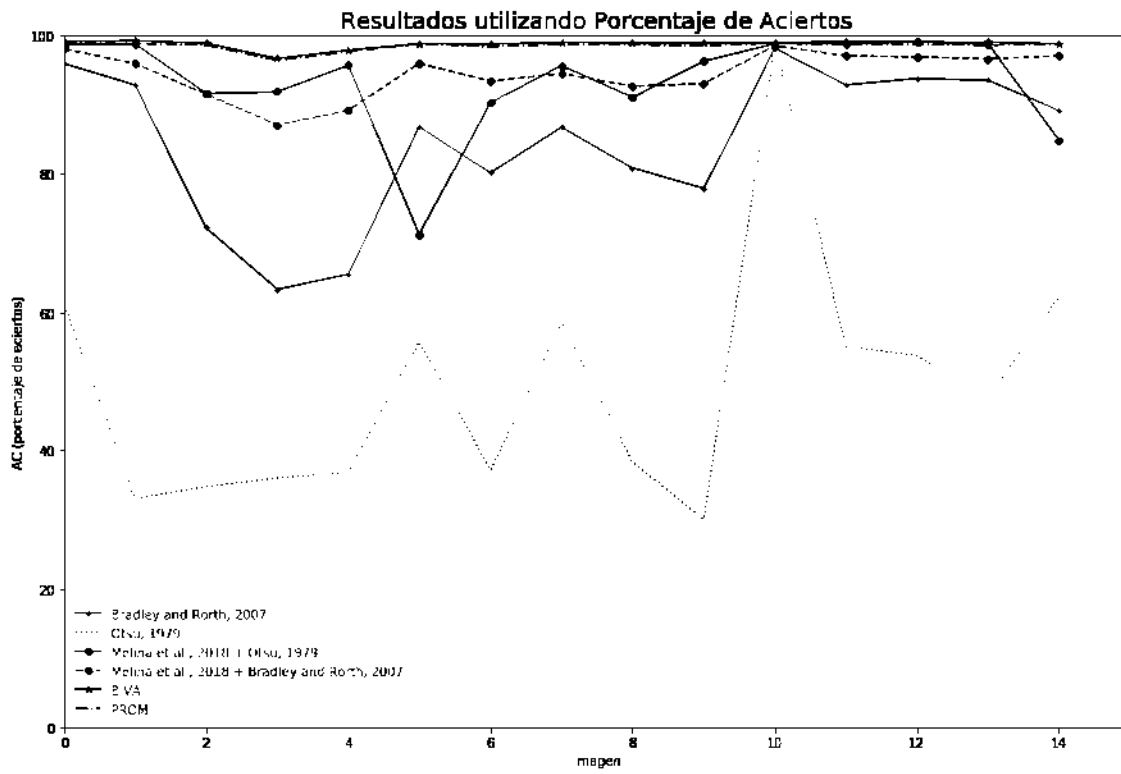
La Figura 4.4 b) muestra el resultado de las imágenes sintéticas binarizadas utilizando la métrica de calidad F-Score calculada mediante (1.2). Al binarizar las imágenes con Otsu, 1979 7 tienen F-score menor a 0.60, 7 imágenes tienen entre 0.6 y 0.8, solamente la una tiene 0.80 de F-score. Con el algoritmo Bradley and Roth, 2007 2 imágenes F-score entre 0.7 y 0.8, 4 entre 0.8 y 0.90 y el resto tienen F-score entre 0.90 y 1.0. Al binarizar las imágenes con Molina et al., 2018 utilizando Otsu, 1979 una imagen tiene 0.83 de F-score y de las 14 imágenes restantes 7 tienen F-score entre 0.90 y 0.98, y 7 tienen F-score de 0.99. Con Molina et al., 2018 utilizando Bradley and Roth, 2007 3 imágenes tienen F-score entre 0.93 y 0.96, de las 12 restantes 6 entre 0.96 y 0.98 y 6 tienen F-score mayor a 0.98. Con el algoritmo BIVA 2 imágenes tienen F-score de 0.98, las 13 imágenes restantes tienen F-score mayor a 0.99. Al binarizar las imágenes con BIPP 2 imágenes tienen F-score de 0.9880, las 13 restantes tienen F-score mayor a 0.99.

La Figura 4.4 c) muestra los resultados de las imágenes binarizadas utilizando el PSNR calculado con (1.4). Las imágenes binarizadas con Otsu, 1979 tienen un valor mínimo de PSNR de 1.55 y un máximo de 19.29, con el algoritmo Bradley and Roth, 2007 el valor mínimo de PSNR es 4.36 y el máximo de 17.354, con Molina et al., 2018 utilizando Otsu, 1979 el valor mínimo de PSNR es 5.4 y el máximo 20.59, con Molina et al., 2018 utilizando Bradley and Roth, 2007 el mínimo es 8.88 y el máximo de 18.29, con BIVA el valor mínimo de PSNR es 14.8093 y el máximo de 21.32, y con

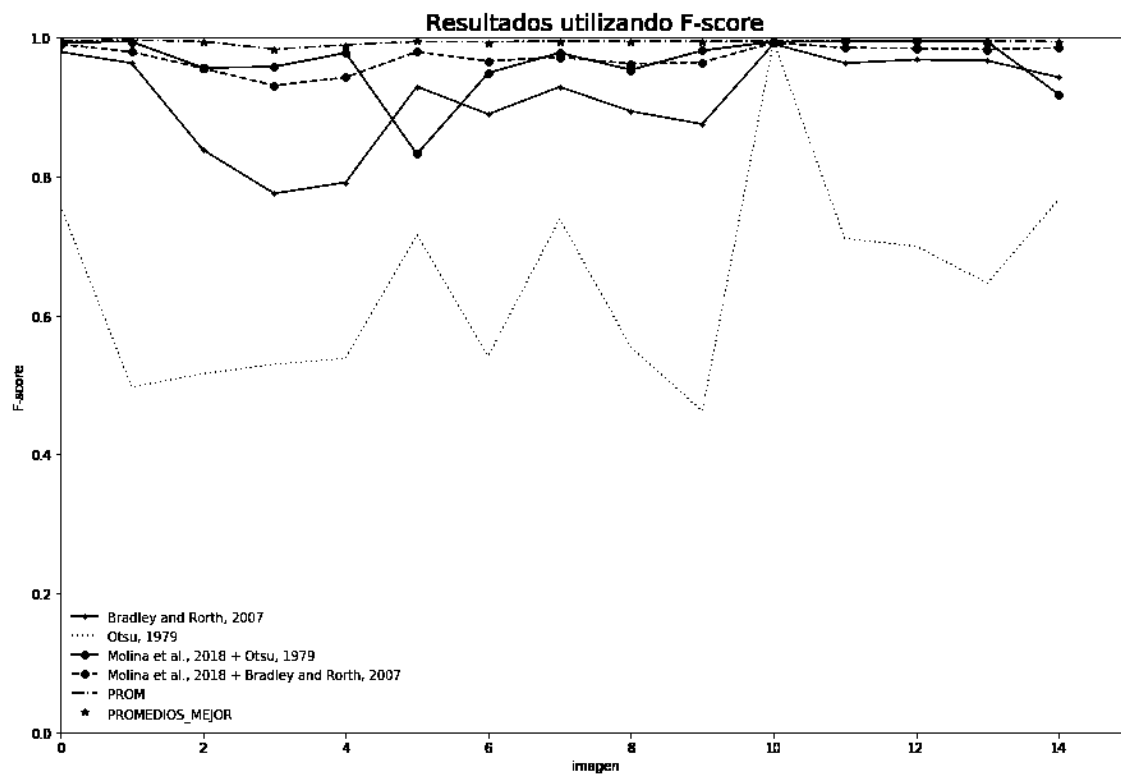
BIPP el mínimo es 14.47 y el valor máximo de PSNR es 20.07.

La Figura 4.4 d) muestra los resultados de las imágenes binarizadas utilizando la métrica de calidad NRM calculado mediante (1.7). Con el algoritmo [Bradley and Roth, 2007](#) 3 imágenes tienen NRM entre 0.250 y 0.125, 6 tienen NRM entre 0.125 y 0.0625 y 6 tienen NRM menor a 0.0625. Con [Otsu, 1979](#) 11 imágenes tienen NRM entre 0.5 y 0.250, 3 entre 0.250 y 0.125 y 1 solamente una tiene NRM menor a 0.0625. Con el algoritmo [Molina et al., 2018](#) utilizando [Otsu, 1979](#) 2 imágenes tienen NRM entre 0.5 y 0.250, 6 entre 0.250 y 0.125, una imagen entre 0.125 y 0.0625 y 6 imágenes tienen NRM menor a 0.0625. Al binarizar las imágenes con [Molina et al., 2018](#) utilizando [Bradley and Roth, 2007](#) 2 imágenes tienen NRM entre 0.5 y 0.250, 7 entre 0.250 y 0.125, 5 entre 0.125 y 0.0625 y una imagen tiene NRM menor a 0.0625. Al binarizar las imágenes con BIVA 2 tienen NRM entre 0.125 y 0.0625 y 13 tienen NRM menor a 0.0625. Al binarizar las imágenes con BIPP 3 tienen NRM entre 0.125 y 0.0625 y 12 tienen NRM menor a 0.0625.

En las 4 gráficas de líneas con los resultados utilizando diferentes métricas de calidad, notamos un mismo comportamiento en los algoritmos al binarizar las imágenes sintéticas. El algoritmo [Otsu, 1979](#) fue el que tuvo más bajo desempeño en la mayoría de las 15 imágenes. [Bradley and Roth, 2007](#) binarizó las imágenes mejor que el algoritmo [Otsu, 1979](#). El algoritmo de [Molina et al., 2018](#) utilizando [Otsu, 1979](#) y [Molina et al., 2018](#) utilizando [Bradley and Roth, 2007](#) tienen mejor desempeño que [Bradley and Roth, 2007](#) al binarizar las imágenes sin embargo, BIPP y BIVA son los algoritmos que tuvieron mejor desempeño al binarizar las 15 imágenes sintéticas con iluminación no uniforme.

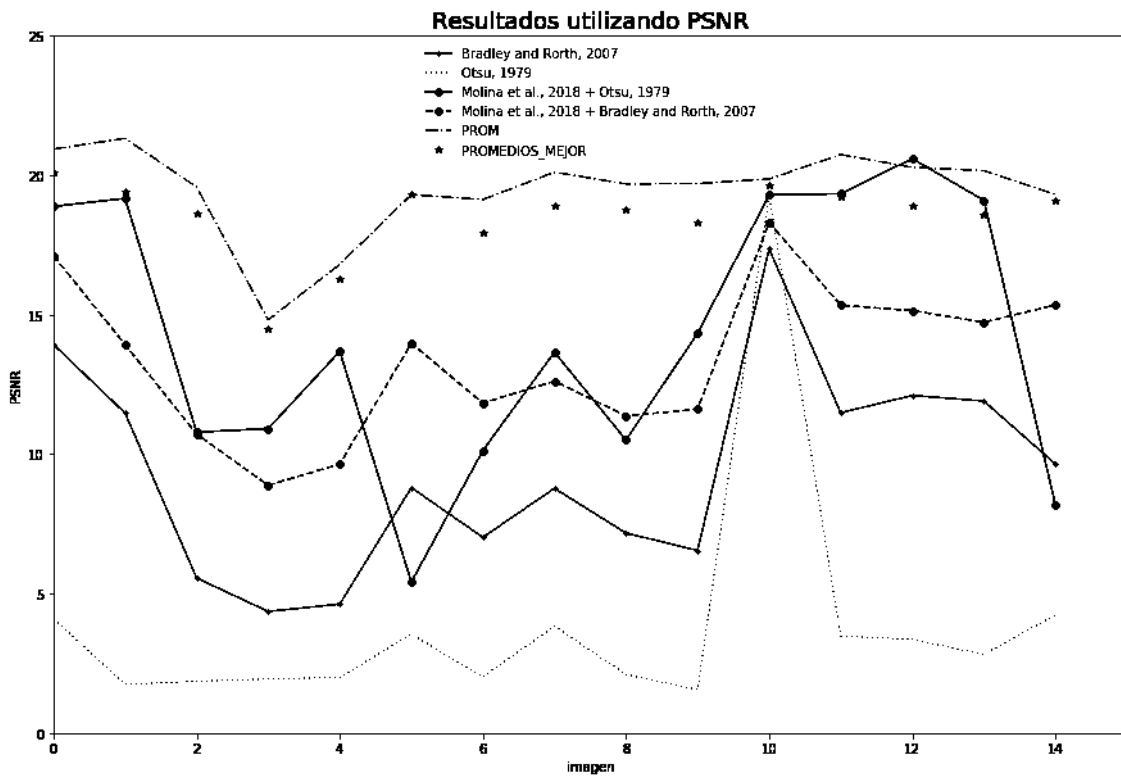


(a)

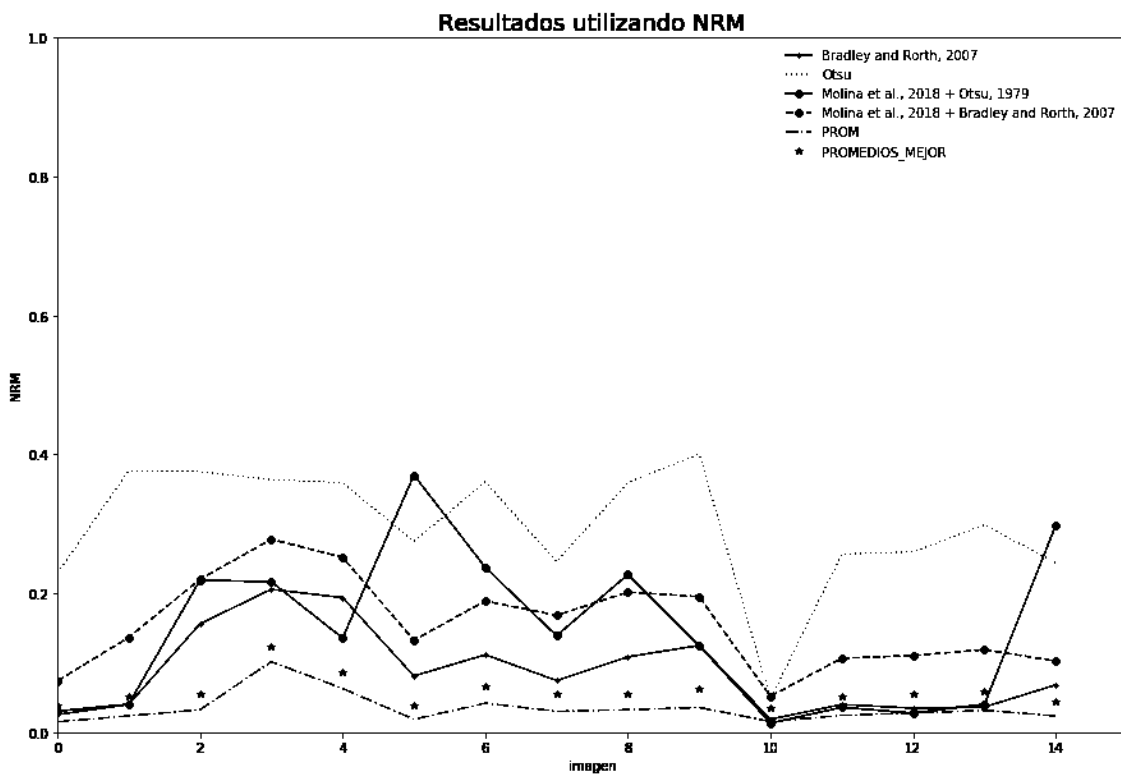


(b)

Figura 4.4: Gráficas con resultados utilizando AC y F-score de imágenes con iluminación sintética binarizadas



(a) PSNR



(b) NRM

Figura 4.5: Gráficas con resultados utilizando PSNR y NRM de imágenes con iluminación sintética binarizadas

La Figura 4.6 muestra gráficas de barras con el promedio de los resultados de las 15 imágenes sintéticas binarizadas con Bradley and Roth, 2007, Otsu, 1979, Molina et al., 2018 utilizando Otsu, 1979 para binarizar la imagen, Molina et al., 2018 utilizando Bradley and Roth, 2007, BIVA y BIPP. En las cuatro gráficas de barras marcamos con color negro los algoritmos que obtuvieron mayor promedio. La Figura 4.6 a) muestra el promedio del porcentaje de aciertos de las 15 imágenes binarizadas en la cual los algoritmos que tienen mayor promedio son BIVA con 97.225 % y BIPP con 96.7699 %. La Figura 4.6 b) muestra el promedio utilizando del F-score de las imágenes binarizadas el la cual BIVA tiene 0.9858 y BIPP 0.9835. La Figura 4.6 c) muestra el promedio del PSNR de las 15 imágenes binarizadas, donde BIVA tiene 16.1619 y BIPP 15.3198. La Figura 4.6 d) muestra el promedio de las 15 imágenes utilizando NRM donde BIVA tiene 0.09376 y BIPP 0.1175. En las 4 gráficas de barras vemos que el promedio de las imágenes sintéticas binarizadas utilizando diferentes métricas de calidad es mayor con los algoritmos BIVA y BIPP.

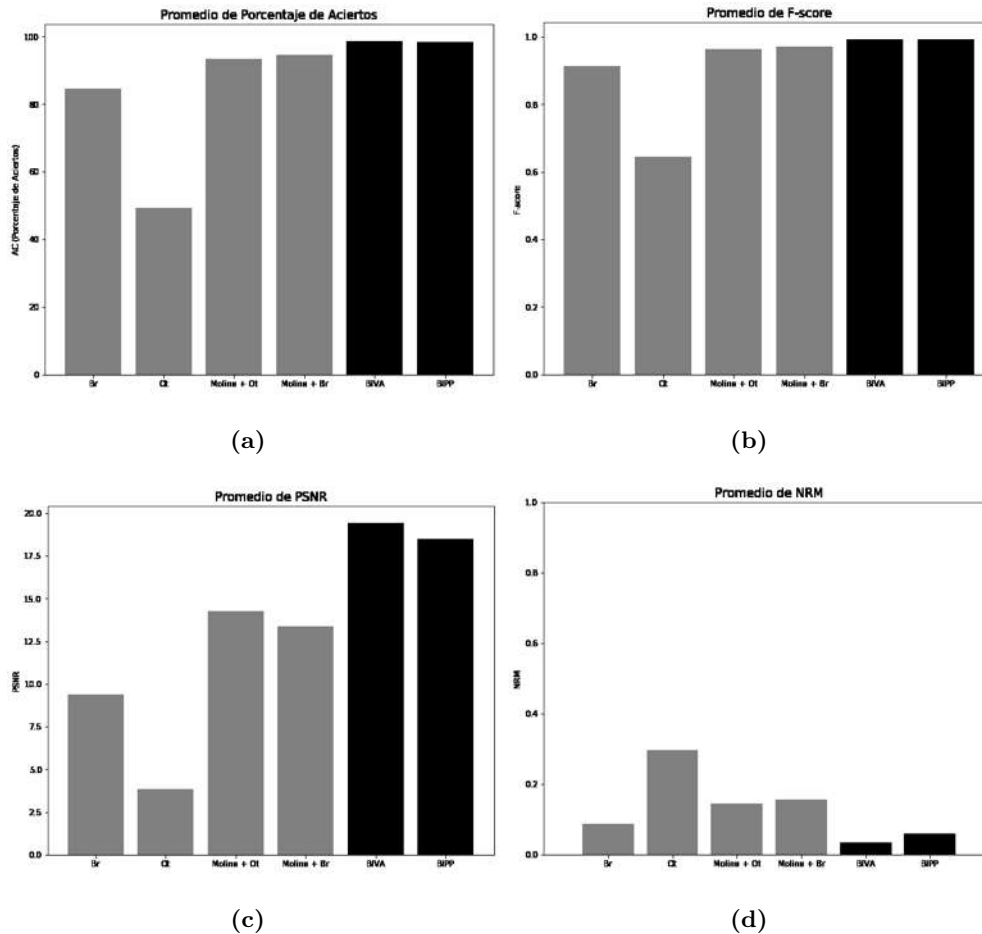


Figura 4.6: Gráficas de barras con promedios donde Br=Bradle and Roth, 2007, Ot=Otsu, 1079, Molina + Br=Molina et al., 2018 utilizando Bradle and Roth, 2007 para binarizar, Molina+Otsu=Molina et al., 2018 utilizando Otsu, 1079

Resultados de imágenes con iluminación real

La Figura 4.7 a) muestra el porcentaje de aciertos calculado con (3.2) de las imágenes reales con iluminación no uniforme binarizadas. Al binarizar las imágenes reales con Otsu, 1979 27 de las 30 imágenes tienen exactitud menor a 60 %, una imagen tiene 69.30 % y 2 imágenes tienen entre 71.69 % y 82.664 % de porcentaje de aciertos. Al binarizar las imágenes con el algoritmo Bradley and Roth, 2007 4 imágenes tienen exactitud menor a 60 %, 14 imágenes tienen entre 60 % y 70 % de porcentaje de aciertos, 7 imágenes tienen entre 70 % y 80 %, 4 imágenes entre 80 % y 90 % y solamente una tiene porcentaje de aciertos mayor a 90 %. Con el algoritmo Molina et al., 2018 utilizando Otsu, 1979 10 tienen porcentaje de aciertos entre 80 % y 90 %, 20 imágenes entre 90 % y 100 % de las cuales 10 imágenes tienen entre 90 % y 95 % y 10 entre 95 % y 100 % de porcentaje de aciertos. Con Molina et al., 2018 utilizando Bradley and Roth, 2007 20 imágenes tienen entre 80 % y 90 % de porcentaje de aciertos y 10 entre 90 % y 100 %. Al binarizar las imágenes con BIVA 3 tienen entre 90 % y 95 % y 27 imágenes tienen entre 95 % y 100 % de porcentaje de aciertos. Al binarizar las imágenes con BIPP 3 tienen entre 90 % y 95 % porcentaje de aciertos y 27 imágenes entre 95 % y 100 %.

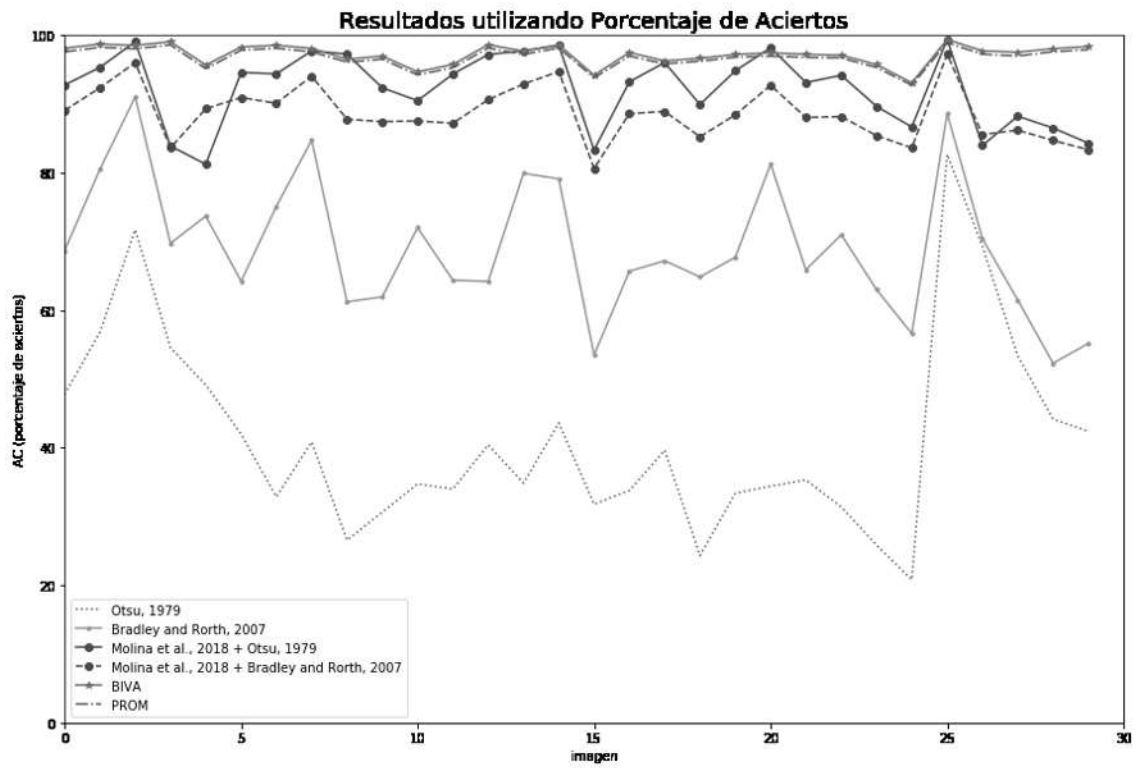
La Figura 4.7 b) muestra el F-score de las imágenes reales binarizadas. Al binarizar las imágenes con Otsu, 1979 20 imágenes tienen F-score menor a 0.6, 5 imágenes tienen entre 0.6 y 0.7, 2 entre 0.7 y 0.8, 2 entre 0.8 y 0.9 y una imagen entre 0.90 y 1.0 de F-score. Con el algoritmo Bradley and Roth, 2007 2 imágenes tienen entre 0.6 y 0.7 de F-score, 12 imágenes entre 0.70 y 0.80, 13 entre 0.80 y 0.90, 2 imágenes entre 0.90 y 0.95 y una imagen entre 0.95 y 1.0 de F-score. Con Molina et al., 2018 utilizando Otsu, 1979 una imagen tiene F-score entre 0.80 y 0.90, 10 entre 0.90 y 0.95 y 19 entre 0.95 y 1.0 de F-score. Al binarizar las imágenes con Molina et al., 2018 utilizando Bradley and Roth, 2007 solamente una imagen tiene F-score entre 0.80 y 0.90, 20 entre 0.90 y 0.95 y 9 imágenes entre 0.95 y 1.0 de F-score. Con BIVA las 30 tienen entre 0.95 y 1.0.

La Figura 4.7 c) muestra los resultados utilizando el PSNR calculado con (1.4) de las imágenes binarizadas. Al binarizar las imágenes con Otsu, 1979 el valor mínimo de PSNR es 1.013 y el máximo es 7.610, con Bradley and Roth, 2007 el mínimo es 3.219 y el máximo 10.474, con Molina et al., 2018 utilizando Otsu, 1979 el valor mínimo de PSNR es 7.263 y el máximo 22.067, con Molina et al., 2018 utilizando Bradley and Roth, 2007 el valor mínimo de PSNR es 7.122 y el máximo 15.58, con BIPP el mínimo es 11.59 y el máximo 21.875 y con BIVA el mínimo es 11.368 y el máximo 19.582.

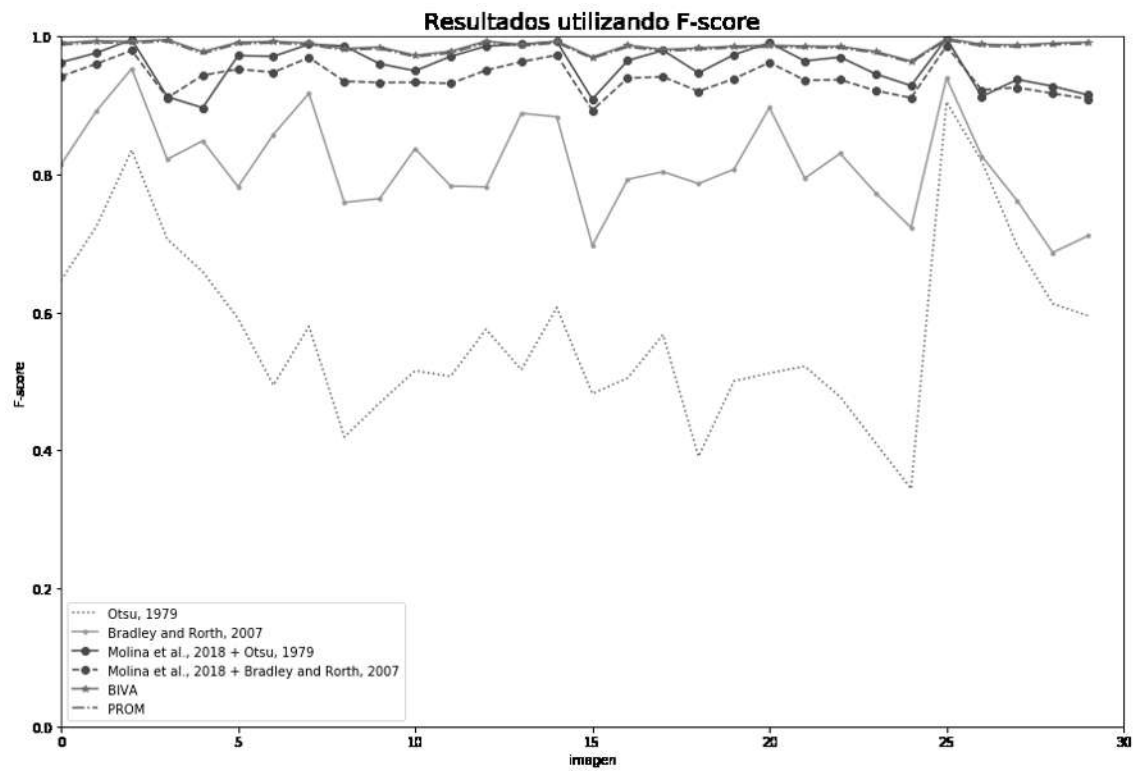
La Figura 4.7 d) muestra el NRM de las 30 imágenes reales binarizadas calculado con (1.7). Al binarizar las imágenes con Otsu, 1979 26 tienen NRM entre 0.5 y 0.250, 3 entre 0.250 y 0.125 y una

entre 0.125 y 0.0625. Con el algoritmo [Bradley and Roth, 2007](#) 3 imágenes tienen NRM entre 0.5 y 0.250, 20 entre 0.250 y 0.125, 6 entre 0.125 y 0.0625 y una tiene NRM menor a 0.0625. Con [Molina et al., 2018](#) utilizando [Otsu, 1979](#) 8 imágenes tienen NRM entre 0.5 y 0.250, 14 entre 0.250 y 0.125, 4 entre 0.125 y 0.0625 y 4 tienen NRM menor a 0.0625. Al binarizar las imágenes con [Molina et al., 2018](#) utilizando [Bradley and Roth, 2007](#) 19 tienen NRM entre 0.5 y 0.250, 10 entre 0.250 y 0.125 y una entre 0.125 y 0.0625. Con el algoritmo BIPP 10 tienen NRM entre 0.250 y 0.125, 18 entre 0.125 y 0.0625 y 2 tienen NRM menor a 0.0625. Al binarizar las imágenes con BIVA 7 entre 0.250 y 0.125, 14 entre 0.125 y 0.0625 y 9 tienen NRM menor a 0.0625.

En las 4 gráficas de líneas las cuales tienen los resultados utilizando diferentes métricas de calidad de las 30 imágenes binarizadas vemos que [Otsu, 1979](#) y [Bradley and Roth, 2007](#) fueron los algoritmos que tuvieron más bajo desempeño en las 30 imágenes. El algoritmo de [Molina et al., 2018](#) utilizando [Otsu, 1979](#) y [Molina et al., 2018](#) utilizando [Bradley and Roth, 2007](#) tienen un buen desempeño ya que tienen mejores resultados que [Bradley and Roth, 2007](#) sin embargo los algoritmos BIPP y BIPP tienen mejor desempeño que [Molina et al., 2018](#) en las 30 imágenes binarizadas.

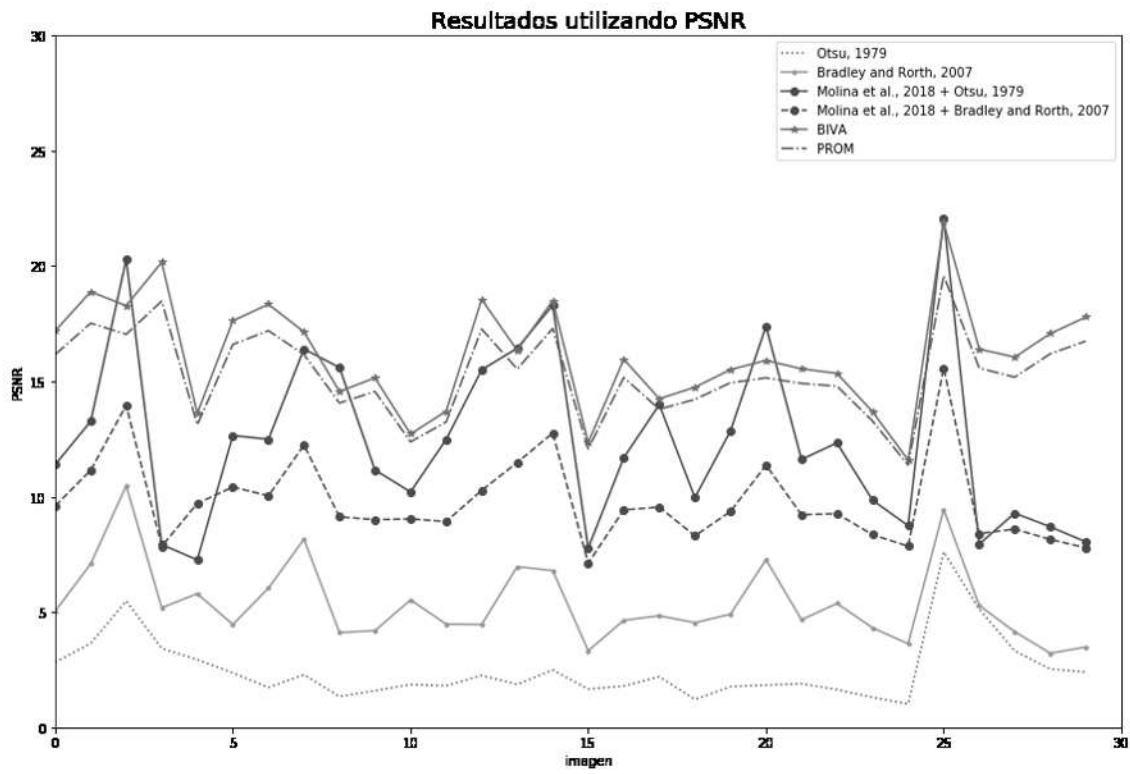


(a)

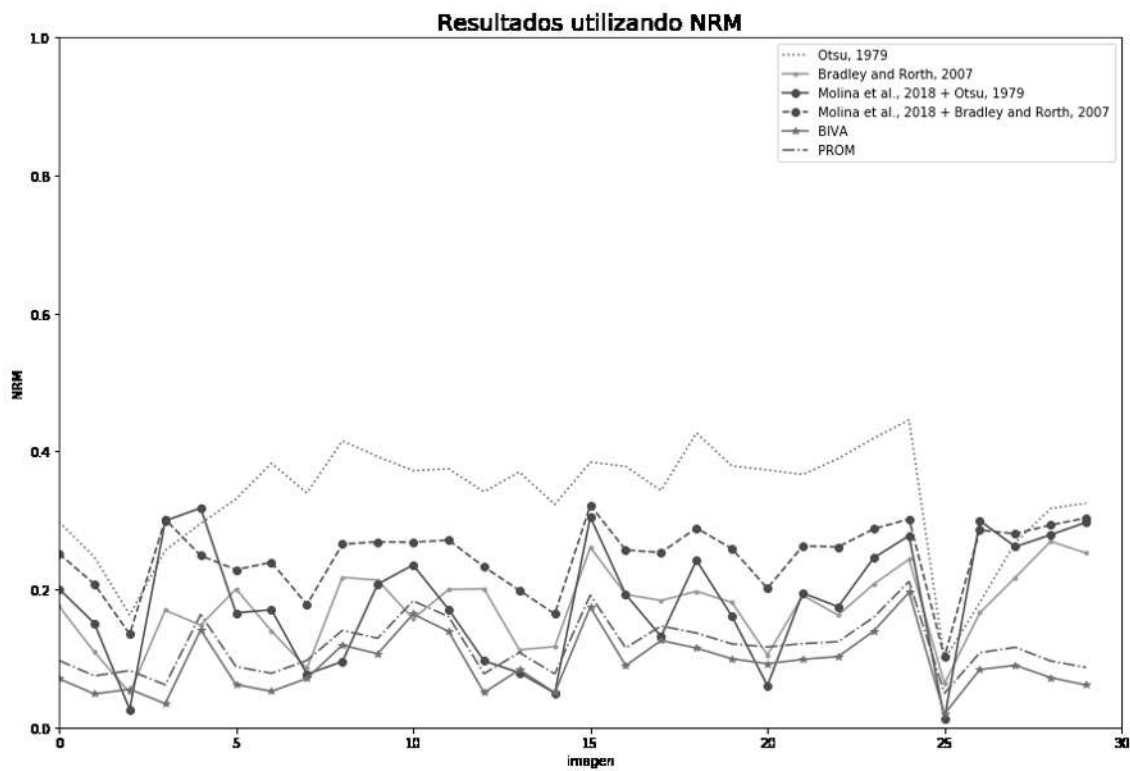


(b)

Figura 4.7: Gráficas que muestran la exactitud de las imágenes con iluminación real binarizadas con los diferentes algoritmos



(a)



(b)

Figura 4.8: Gráficas con resultado de las imágenes con iluminación real binarizadas con los diferentes algoritmos

La Figura 4.9 muestra gráficas de barras con el promedio de los resultados de las 30 imágenes reales binarizadas con Bradley and Roth, 2007, Otsu, 1979, Molina et al., 2018 utilizando Otsu, 1979 para binarizar la imagen y Molina et al., 2018 utilizando Bradley and Roth, 2007, BIVA y BIPP. En las gráficas de barras marcamos los algoritmos que tuvieron promedio más alto. La Figura 4.6 a) muestra el promedio del porcentaje de aciertos donde BIVA tiene 98.765 % y BIPP 98.497 % de porcentaje de aciertos. La Figura 4.6 b) muestra el promedio del F-score donde BIVA tiene 0.994 y BIPP 0.992. La Figura 4.6 c) muestra el promedio del PSNR donde BIVA tiene 19.445 y BIPP 18.489 y la Figura 4.6 d) muestra el promedio del NRM donde BIVA tiene 0.0344 y BIPP 0.0584. En las cuatro gráficas de barras marcamos los algoritmos BIVA y BIPP debido a que son los que tienen mayor promedio en los resultados de las imágenes binarizadas.

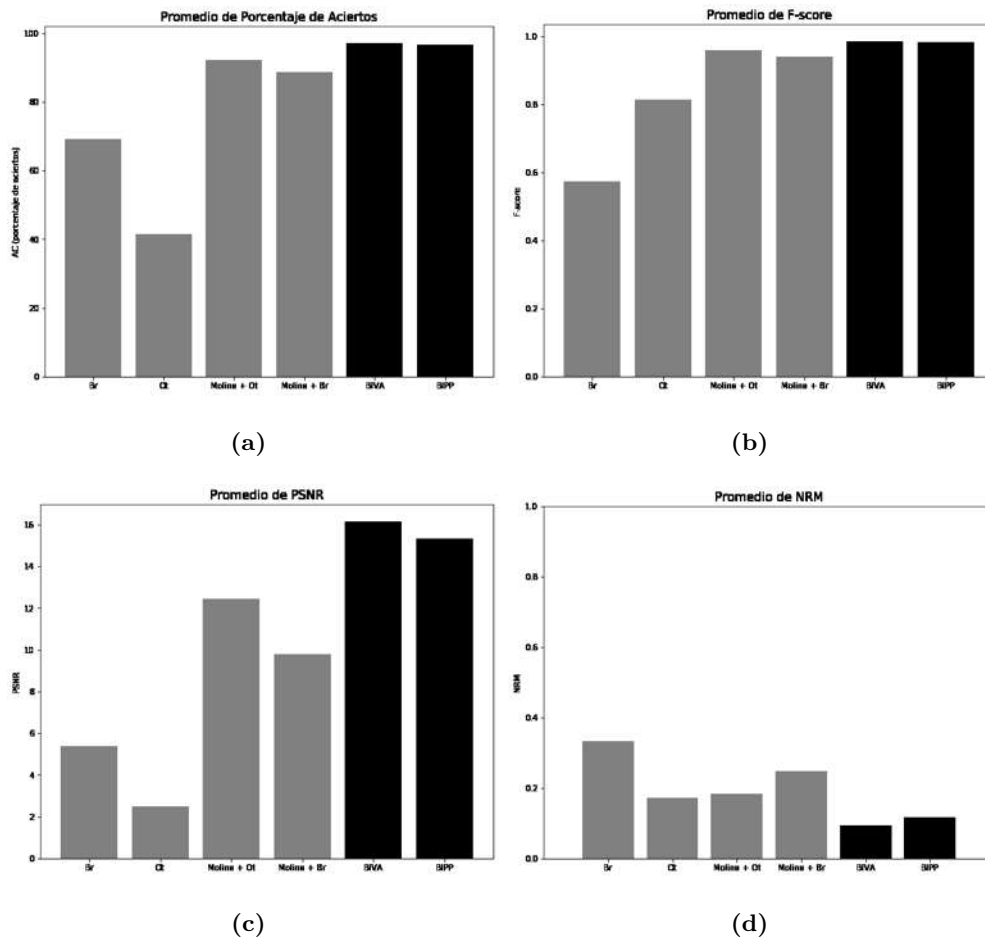


Figura 4.9: Gráficas de barras con promedios donde Br=Bradle and Roth, 2007, Ot=Otsu, 1079, Molina + Br=Molina et al., 2018 utilizando Bradle and Roth, 2007 para binarizar, Molina+Otsu=Molina et al., 2018 utilizando Otsu, 1079

Otsu, 1979 es un algoritmo utilizado por su simplicidad y rapidez al binarizar las imágenes,

sin embargo en las gráficas de líneas y de barras que muestran los resultados de las imágenes sintéticas y reales vimos que tuvo bajo desempeño, esto se debe a que los algoritmos que utilizan umbral global tienen problemas para binarizar imágenes con iluminación no uniforme. [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) son algoritmos que utilizan umbral local haciendo uso de ventanas de tamaño constante, en las gráficas de líneas y de barras vimos que las imágenes binarizadas con dichos algoritmos tuvieron mayor exactitud que las binarizadas con [Otsu, 1979](#) que utiliza umbral global. Sin embargo las imágenes binarizadas con los algoritmos BIPP y BIVA tienen mayor exactitud que las binarizadas con [Molina et al., 2018](#) y [Bradley and Roth, 2007](#), esto se debe a que BIVA y BIPP tienen la ventaja de utilizar ventanas adaptables sujetas a respetar los bordes donde ocurre el cambio de iluminación.

4.3. Resultados utilizando diferentes parámetros en los algoritmos

En la sección 4.2 mostramos gráficas de líneas y de barras con resultados de imágenes sintéticas y reales binarizadas mediante [Molina et al., 2018](#), [Otsu, 1979](#), [Bradley and Roth, 2007](#), BIPP y BIVA con el propósito de medir y visualizar el desempeño de los algoritmos. Sin embargo BIVA, BIPP, [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) utilizan parámetros que se tienen que establecer antes de binarizar la imagen. En [Bradley and Roth, 2007](#) los parámetros son τ y w , en BIVA son w_{max} , τ y T , en [Molina et al., 2018](#) C y w y en BIPP son C , w_{max} y τ . En esta sección binarizamos una imagen que presenta iluminación no uniforme utilizando diferentes valores en los parámetros de BIVA, [Bradley and Roth, 2007](#), [Molina et al., 2018](#) y BIPP. La Figura 4.10 a) muestra una imagen de texto y la Figura 4.10 b) muestra la misma imagen con iluminación no uniforme generada con una función de Gauss con media $\mu = 100$ y sigma $\sigma = 50$.

Este un documento de prueba para ver cómo funcionan los algoritmos de Binarización de Imágenes.

Se prueban los algoritmos de Otsu, Bradley-Roth y BI-VA.

(a)

Este un documento de prueba para ver cómo funcionan los algoritmos de Binarización de Imágenes.

Se prueban los algoritmos de Otsu, Bradley-Roth y BI-VA.

(b)

Figura 4.10: A la izquierda imagen con texto y a la derecha la misma imagen con iluminación no uniforme generada con una función de Gauss

La Tabla 4.1 contiene exactitudes de la Figura 4.10 b) binarizada con Bradley and Roth, 2007 utilizando diferentes valores en τ y w . De la misma manera la Tabla 4.2 contiene exactitudes de la Figura 4.10 b) binarizada con BIVA utilizando diferentes valores en w_{max} y τ . Comparamos las exactitudes de la Tabla 4.1 con las de la Tabla 4.2 para determinar si se obtiene mayor exactitud en la imagen binarizada cuando se utilizan ventanas adaptables. La forma de hacer la comparación fue con las exactitudes ubicadas en la misma posición es decir en la misma fila y en la misma columna. Si la exactitud ubicada en cierta columna y fila de la Tabla 4.2 es mayor que la exactitud ubicada en la misma fila y columna de la Tabla 4.1 la marcamos con gris para identificar que BIVA a tenido mejor desempeño, de lo contrario marcamos la exactitud de la Tabla 4.1 para identificar que Bradley and Roth, 2007 ha tenido mejor desempeño, con lo que estamos haciendo una comparación justa al utilizar los mismos valores en los parámetros de ambos algoritmos. Vemos que hemos marcado más exactitudes en la Tabla 4.2 que corresponde a las exactitudes obtenidas al binarizar la imagen con BIVA con lo que podemos ver la ventaja de utilizar ventanas adaptables.

	$\tau = 0$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 40$	$\tau = 50$	$\tau = 60$	$\tau = 70$	$\tau = 80$	$\tau = 90$
$w = 0$	70.87	88.39	88.39	88.39	88.39	88.39	88.39	88.39	88.39	88.388
$w = 10$	74.0	99.38	99.38	99.38	99.38	99.38	99.38	99.38	99.38	99.38
$w = 20$	85.59	98.87	99.39	99.38	99.38	99.38	99.38	99.38	99.38	99.38
$w = 30$	88.5	97.73	99.38	99.38	99.38	99.38	99.38	99.38	99.38	99.38
$w = 40$	83.30	93.2	97.75	99.38	99.38	99.38	99.38	99.38	99.38	99.38
$w = 50$	79.17	86.77	93.98	98.56	99.38	99.38	99.38	99.38	99.38	99.38
$w = 60$	77.38	82.28	88.98	94.81	99.38	99.38	99.38	99.38	99.38	99.37
$w = 70$	74.74	79.46	84.72	90.81	96.27	99.38	99.38	99.38	99.38	99.38
$w = 80$	71.73	76.5	81.5	87.15	92.69	98.43	99.38	99.38	99.38	99.38
$w = 90$	69.38	73.81	78.73	84.78	90.97	96.67	99.39	99.39	99.39	99.39

Tabla 4.1: Exactitud de la Figura 4.10 b) binarizada con Bradley and Roth, 2007 utilizando diferentes valores de τ y w

	$\tau = 0$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 40$	$\tau = 50$	$\tau = 60$	$\tau = 70$	$\tau = 80$	$\tau = 90$
$w = 0$	70.87	88.41	88.41	88.41	88.41	88.41	88.41	88.41	88.41	88.41
$w = 10$	70.57	99.39	99.39	99.39	99.39	99.39	99.39	99.39	99.39	99.39
$w = 20$	81.82	98.97	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.31
$w = 30$	88.32	98.67	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4
$w = 40$	89.08	98.64	99.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4
$w = 50$	89.8	98.6	99.39	99.39	99.39	99.39	99.39	99.39	99.39	99.39
$w = 60$	90.45	98.64	99.39	99.39	99.39	99.39	99.39	99.39	99.39	99.39
$w = 70$	90.81	98.64	99.39	99.39	99.39	99.39	99.39	99.39	99.39	99.39
$w = 80$	90.82	98.63	99.39	99.39	99.39	99.39	99.39	99.39	99.39	99.39
$w = 90$	90.8	98.64	99.39	99.39	99.39	99.39	99.39	99.39	99.39	99.39

Tabla 4.2: Exactitud de la Figura 4.10 b) binarizada con BIVA utilizando diferentes valores de τ y w_{max}

La Tabla 4.3 contiene exactitudes de la Figura 4.10 b) binarizada con Molina et al., 2018 utilizando diferentes valores en los parámetros y la Tabla 4.4 al binarizarla con BIPP utilizando diferentes valores en los parámetros. La Tabla 4.4 tiene 58 exactitudes marcadas con gris, con lo que vemos que BIPP tiene mejor desempeño que Molina et al., 2018 al binarizar la imagen con iluminación no uniforme.

	$w = 0$	$w = 10$	$w = 20$	$w = 30$	$w = 40$	$w = 50$	$w = 60$	$w = 70$	$w = 80$	$w = 90$
$C = 0$	88.8	89.676	94.68	98.69	99.53	98.67	96.81	96.6	96.59	96.59
$C = 10$	83.48	90.85	96.62	99.53	99.7	98.67	96.81	96.6	96.6	96.6
$C = 20$	83.05	88.13	94.3	98.72	99.53	98.67	96.81	96.6	96.6	96.6
$C = 30$	83.88	92.0	97.47	99.85	99.72	98.69	96.88	96.59	96.59	96.6
$C = 40$	84.76	91.61	97.28	99.76	99.72	98.69	96.9	96.67	96.67	96.67
$C = 50$	84.78	87.77	94.08	98.8	99.57	98.74	96.91	96.71	96.71	96.71
$C = 60$	84.60	87.99	94.99	99.09	99.65	98.64	96.53	96.32	96.32	96.32
$C = 70$	84.49	88.87	93.83	97.5	98.16	97.12	94.99	94.74	94.74	94.74
$C = 80$	82.82	84.97	91.20	96.0756	96.62	95.719	93.65	93.32	93.32	93.32
$C = 90$	82.76	88.85	93.2	95.63	95.28	94.48	91.06	90.65	90.65	90.64

Tabla 4.3: Exactitud de la Figura 4.10 b) binarizada con Molina et al., 2018 utilizando diferentes valores C y w .

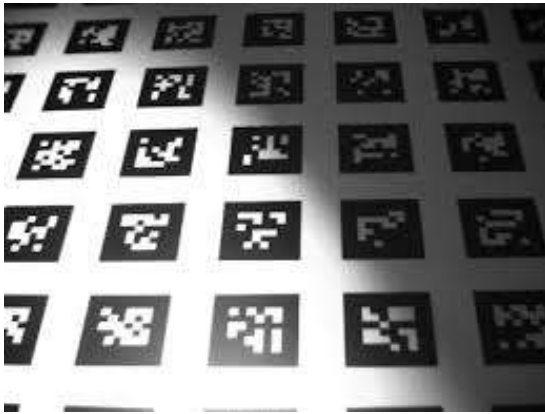
	$\tau = 0$	$\tau = 10$	$\tau = 20$	$\tau = 30$	$\tau = 40$	$\tau = 50$	$\tau = 60$	$\tau = 70$	$\tau = 80$	$\tau = 90$
$w_{max} = 0$	60.23	90.52	90.52	90.52	90.52	90.52	90.52	90.52	90.52	90.52
$w_{max} = 10$	80.47	85.94	86.6382	87.40	88.44	89.76	91.8	93.38	97.01	99.97
$w_{max} = 20$	85.88	89.01	89.59	90.30	91.27	92.47	93.90	95.68	98.39	99.97
$w_{max} = 30$	88.54	89.91	90.62	91.5	92.47	93.54	94.72	96.17	98.58	99.97
$w_{max} = 40$	90.52	92.74	93.36	94.12	95.0	95.9	96.68	97.568	99.14	99.97
$w_{max} = 50$	94.78	95.94	96.47	97.03	97.66	98.23	98.78	99.3	99.86	99.97
$w_{max} = 60$	94.7	96.19	96.73	97.22	97.80	98.32	98.8	99.29	99.87	99.97
$w_{max} = 70$	95.08	96.02	96.84	97.56	98.21	98.75	99.23	99.62	99.94	99.97
$w_{max} = 80$	94.71	95.73	96.61	97.32	97.97	98.54	99.06	99.52	99.90	99.97
$w_{max} = 90$	94.35	95.42	96.36	97.12	97.81	98.43	98.97	99.45	99.85	99.97

Tabla 4.4: Exactitud de la Figura 4.10 b) binarizada con BIPP utilizando diferentes valores en los parámetros τ y w_{max}

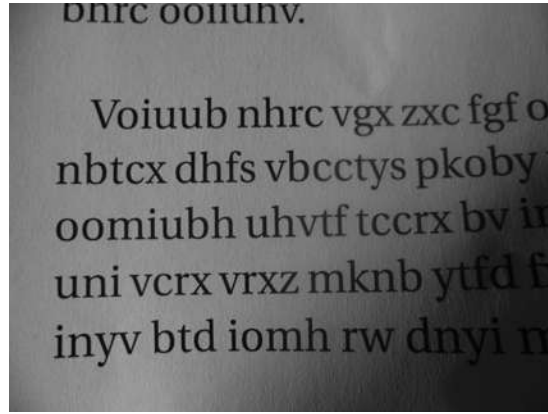
En esta sección comparamos el desempeño de los algoritmos que utilizan ventanas BIVA, Molina et al., 2018, BIPP y Bradley and Roth, 2007 utilizando diferentes valores en sus parámetros y mostramos los resultados en tablas en las que pudimos ver que aunque cambie el valor de los parámetros sigue existiendo la ventaja de utilizar ventanas de tamaño óptimo empleadas en BIVA y BIPP.

4.4. Resultados de forma cualitativa

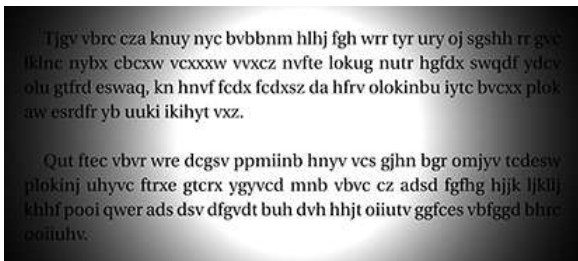
El objetivo de esta sección es mostrar imágenes binarizadas con los algoritmos BIVA, BIPP, Molina et al., 2018 y Bradley and Roth, 2007. La Figura 4.11 muestra las imágenes que utilizamos.



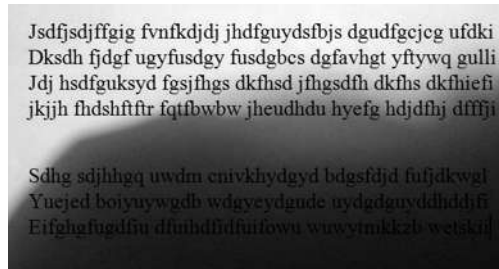
(a) imagen 1



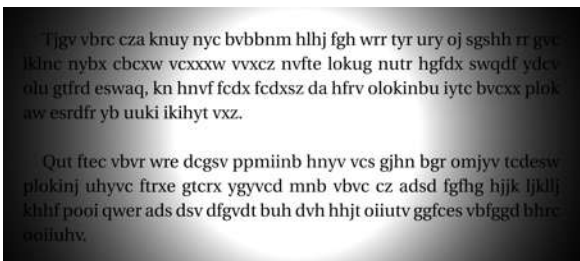
(b) imagen 2



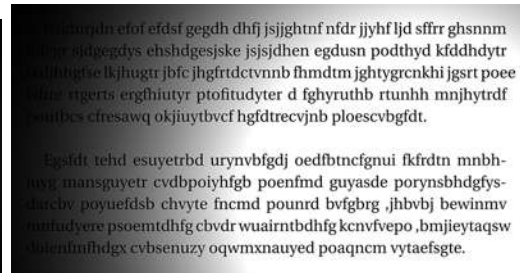
(c) imagen 3



(d) imagen 4



(e) imagen 5



(f) imagen 6

Figura 4.11: Imágenes con iluminación no uniforme

La Figura 4.12 a) muestra una imagen con iluminación no uniforme y la Figura 4.12 b) muestra la imagen con el valor w en cada coordenada calculado con VO donde se nota que el área donde ocurre el cambio de iluminación es más oscura esto se debe a que el tamaño de las ventanas en dicha área es más pequeño y en el área gris el valor es más grande los cuales corresponden a tonos de grises claros. La Figura 4.12 c) muestra la imagen binarizada con [Molina et al., 2018](#) donde $C = 50$ y la Figura 4.12 d) muestra la imagen binarizada con BIPP donde utilizamos $C = 50$, $w_{max} = 80$, $\tau = 10$ y $T = 20$, podemos notar en las imágenes que tienen una línea donde se produce el cambio de iluminación en la imagen original sin embargo, en la imagen binarizada con BIPP la línea es más delgada. La Figura 4.12 e) muestra la imagen binarizada con [Bradley and Roth, 2007](#)

donde $w = 80$, $\tau = 10$ y la Figura 4.12 f) muestra la imagen binarizada con BIVA donde $w_{max} = 80$, $\tau = 10$ y $T = 20$, se puede ver que la imagen binarizada con Bradley and Roth, 2007 tiene una mancha en la esquina superior izquierda y la binarizada con BIVA no la tiene.

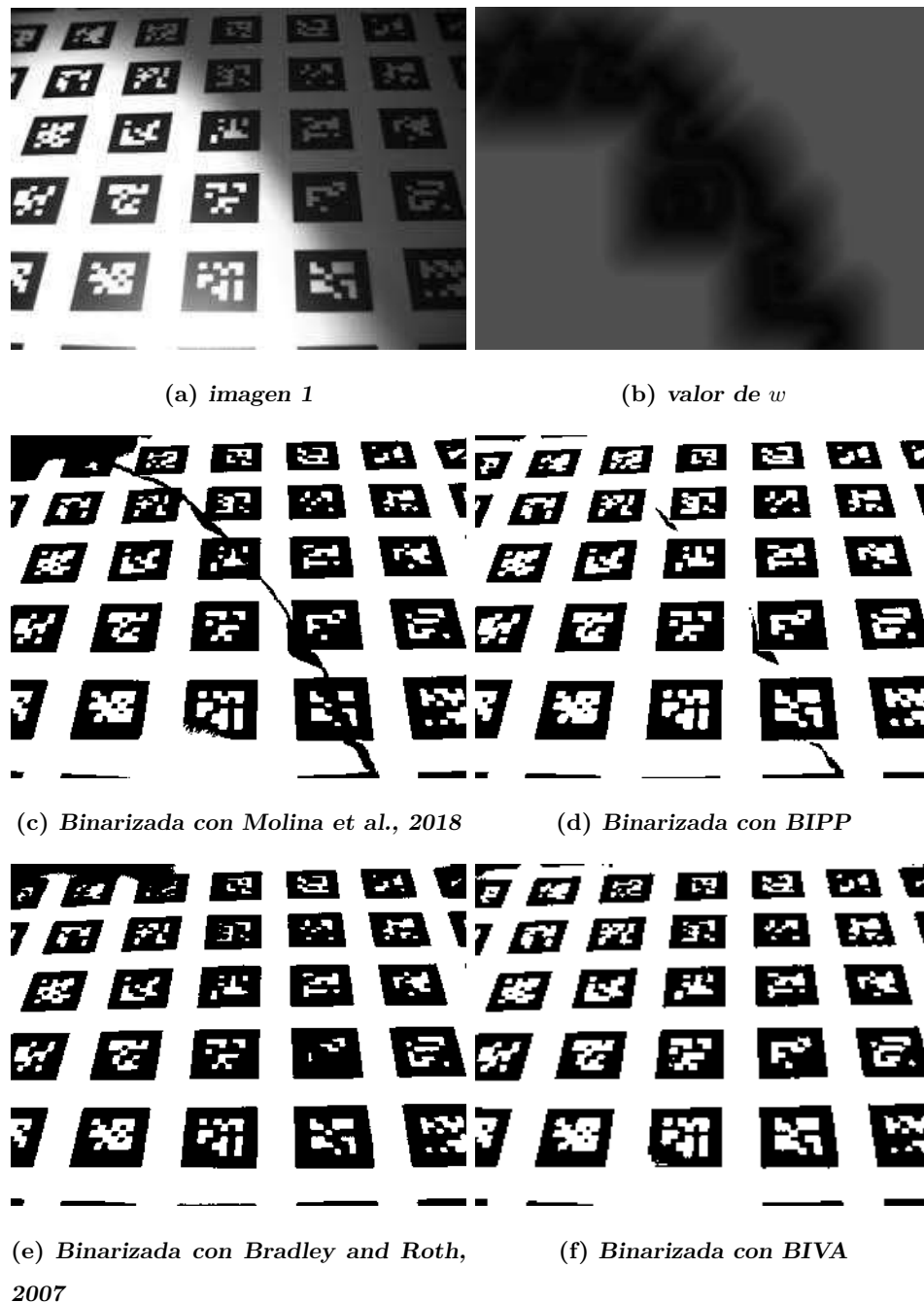


Figura 4.12: Binarización de la Figura 4.11 a) con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA

La Figura 4.13 a) muestra una imagen con iluminación no uniforme y la Figura 4.13 b) la imagen en la que cada coordenada $[r, c]$ tiene el valor w calculado con VO. La Figura 4.13 c)

muestra la imagen binarizada con [Molina et al., 2018](#) donde $C = 50$ y la Figura 4.13 d) muestra la imagen binarizada con BIPP donde utilizamos $C = 50$, $w_{max} = 90$, $\tau = 20$ y $T = 20$, podemos notar que la imagen binarizada con BIPP no tiene manchas sobre las letras y la binarizada con [Molina et al., 2018](#) si tiene. La Figura 4.13 e) muestra la imagen binarizada con [Bradley and Roth, 2007](#) donde utilizamos $w = 90$, $\tau = 20$ y la Figura 4.13 f) muestra la imagen binarizada con BIVA donde utilizamos $w = 90$, $\tau = 20$ y $T = 15$, en la imagen binarizada con BIVA podemos ver que no tiene manchas en las letras a diferencia de la imagen binarizada con [Bradley and Roth, 2007](#) que si tiene.

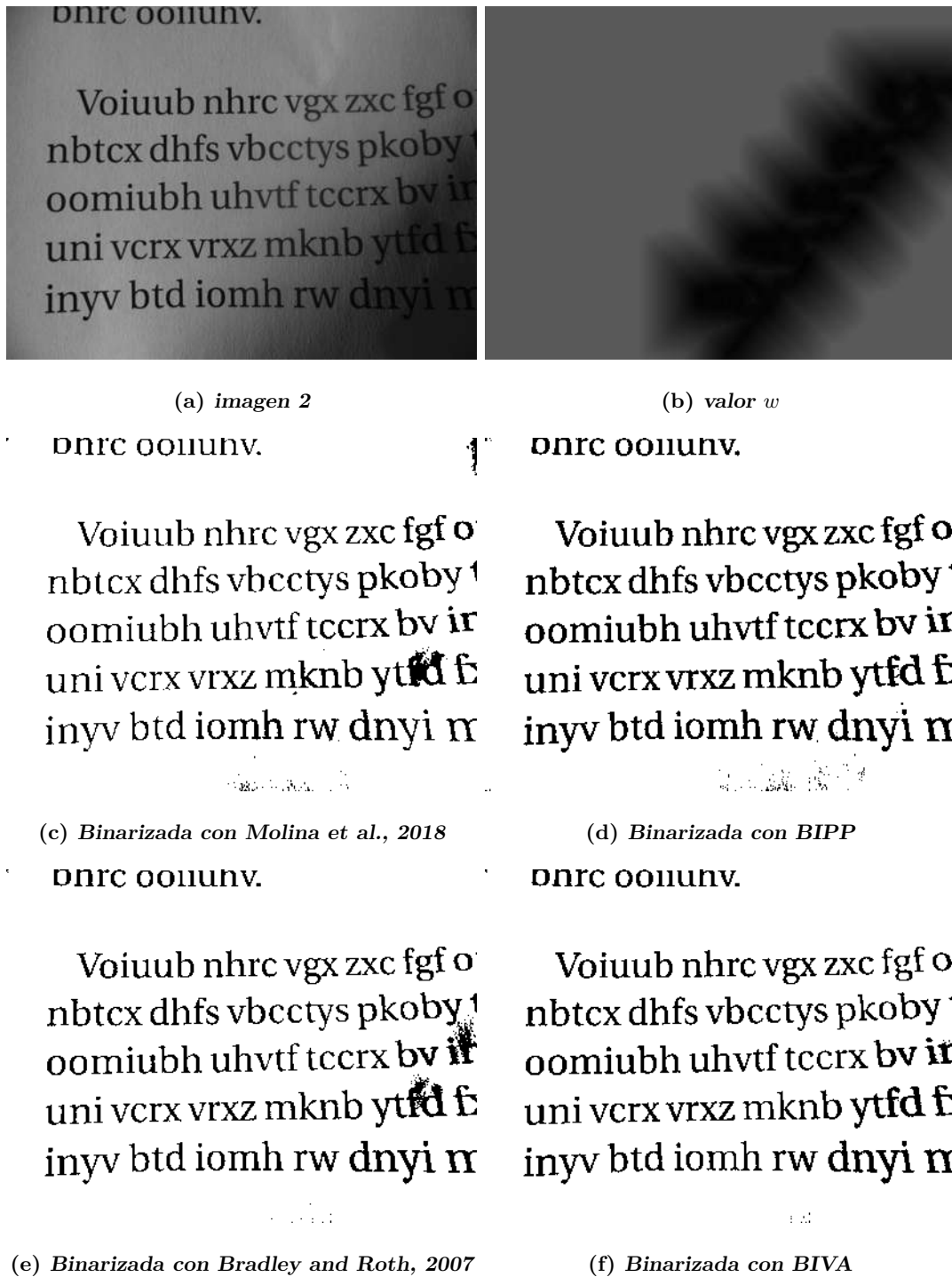


Figura 4.13: Binarización de la Figura 4.11 b) con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA

La Figura 4.14 a) muestra una imagen con iluminación no uniforme y la Figura 4.14 b) la imagen en la que cada coordenada tiene el valor w calculado con VO. La Figura 4.14 c) muestra la imagen binarizada con Molina et al., 2018 donde $C = 50$ y la Figura 4.14 d) la imagen binarizada

mediante BIPP donde utilizamos $C = 50$, $w_{max} = 100$, $\tau = 30$ y $T = 20$, en la imagen binarizada con Molina et al., 2018 se pueden ver áreas totalmente oscuras que corresponden a las áreas oscuras de la imagen original sin embargo, en la imagen binarizada con BIPP dichas áreas son más pequeñas. La Figura 4.14 e) muestra la imagen binarizada con Bradley and Roth, 2007 donde $w = 100$, $\tau = 30$ y la Figura 4.14 f) la imagen binarizada con BIVA donde $w = 100$, $\tau = 30$ y $T = 15$, en la imagen binarizada con BIVA vemos que las áreas oscuras son más pequeñas.

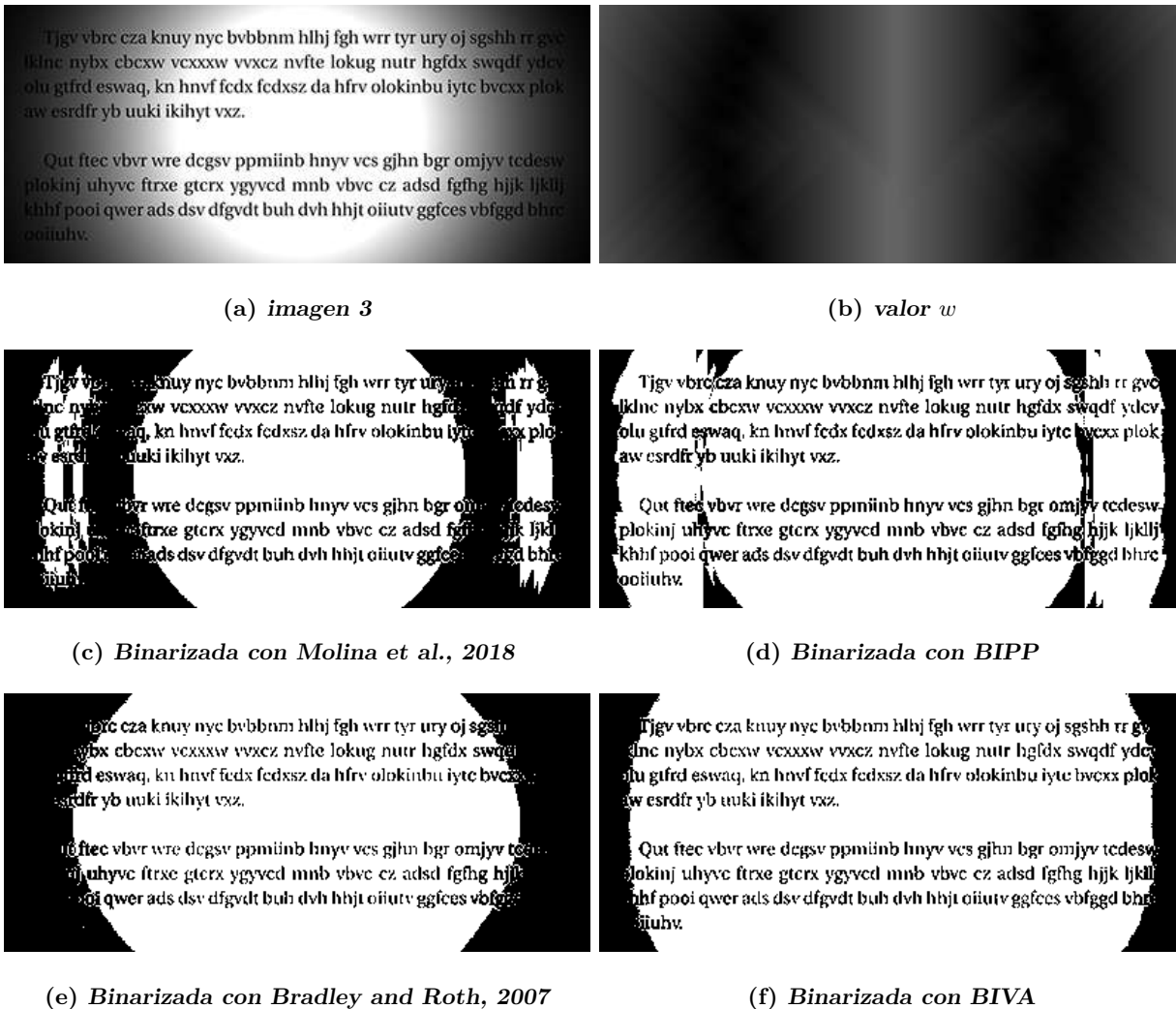


Figura 4.14: Binarización de la Figura 4.11 c) con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA

La Figura 4.15 a) muestra una imagen con iluminación no uniforme y la Figura 4.15 b) una imagen en la que cada coordenada tiene el valor w calculado con VO. La Figura 4.15 c) muestra la imagen binarizada con Molina et al., 2018 donde $C = 50$ y la Figura 4.15 d) muestra la imagen binarizada con BIPP donde utilizamos $C = 50$, $w_{max} = 60$, $\tau = 30$ y $T = 20$, en la imagen binarizada

con [Molina et al., 2018](#) vemos que tiene una línea donde se produce el cambio de iluminación y la binarizada con BIPP no la tiene. La Figura 4.15 e) muestra la imagen binarizada con [Bradley and Roth, 2007](#) donde utilizamos $w = 60$, $\tau = 30$ y la Figura 4.15 f) muestra la imagen binarizada con BIVA donde $w = 60$, $\tau = 30$ y $T = 15$, se puede notar que la imagen binarizada con BIVA no tiene manchas sobre las letras.

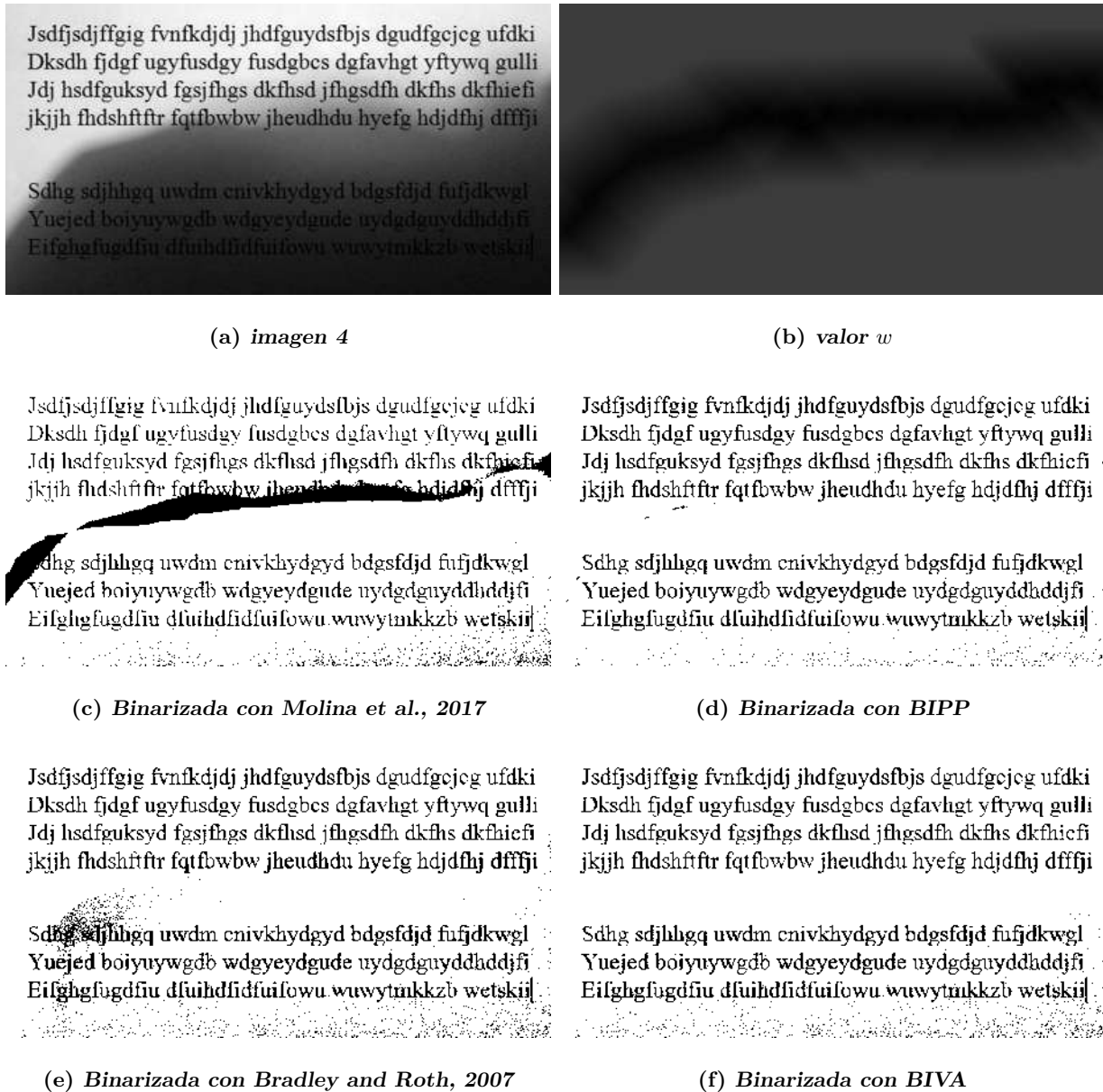


Figura 4.15: Binarización de la Figura 4.11 d) binarizada con Molina et al., 2018, BIPP, Bradley and Roth, 2007 y BIVA

La Figura 4.16 a) muestra una imagen con iluminación no uniforme y la Figura 4.16 b) muestra la imagen en la que cada coordenada tiene el valor w calculado con VO. La Figura 4.16

c) muestra la imagen binarizada con [Molina et al., 2018](#) $C = 50$ y la Figura 4.16 d) muestra la imagen binarizada con BIPP donde utilizamos $C = 50$, $w_{max} = 60$, $\tau = 30$ y $T = 20$, se puede ver que en ambas imágenes se binarizaron correctamente las letras, pero en el área del centro de la imagen, BIPP tuvo mejor desempeño. La Figura 4.16 e) muestra la imagen binarizada con [Bradley and Roth, 2007](#) donde utilizamos $w = 60$, $\tau = 30$ y la Figura 4.16 f) muestra la imagen binarizada con BIVA donde $w = 60$, $\tau = 30$ y $T = 15$, en dichas imágenes se observa que la binarizada con BIVA tiene las áreas oscuras más pequeñas.

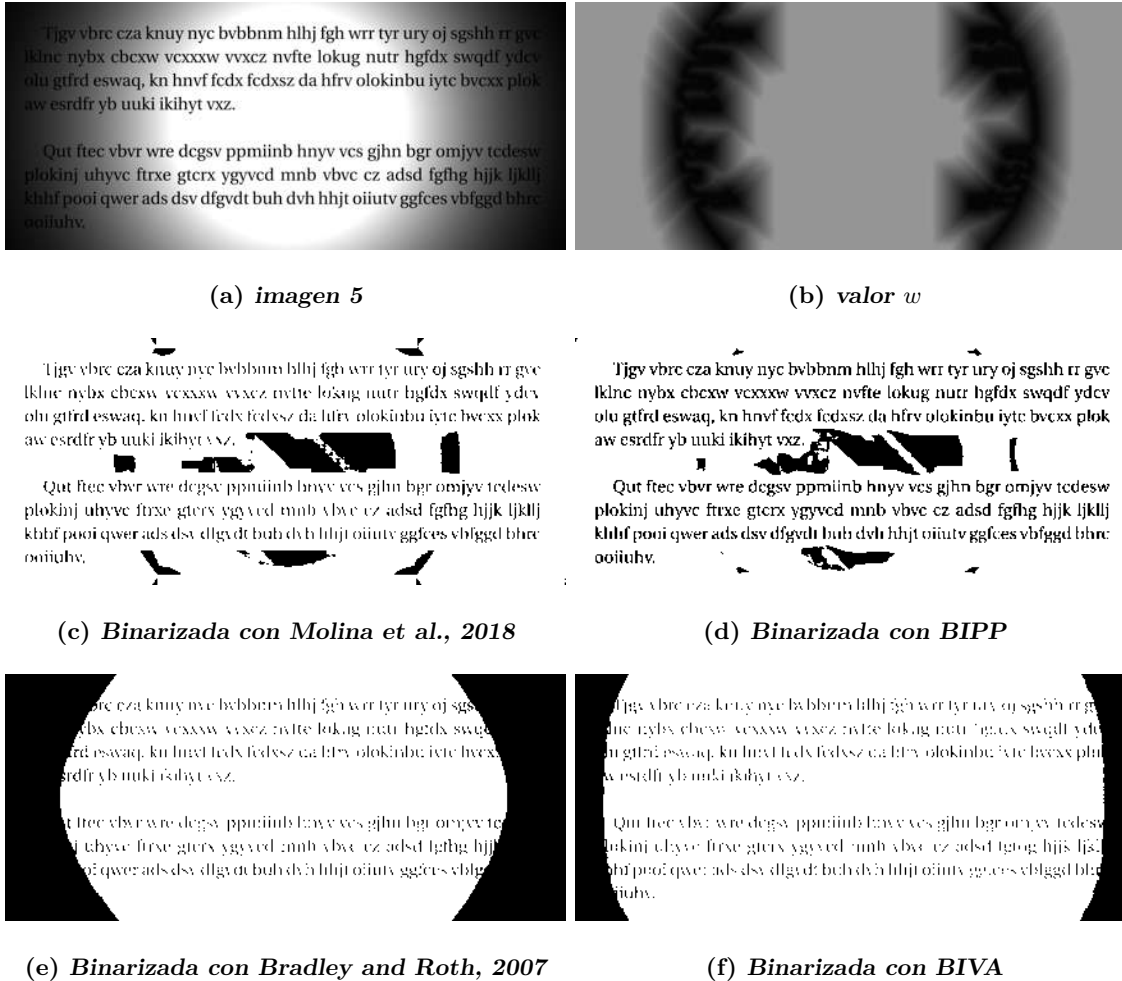
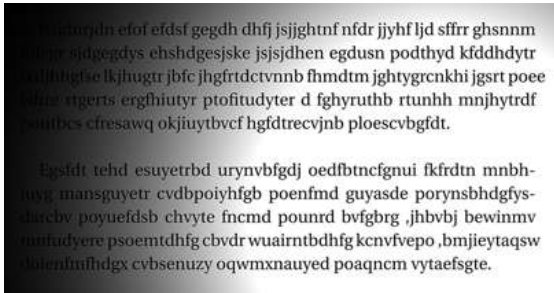


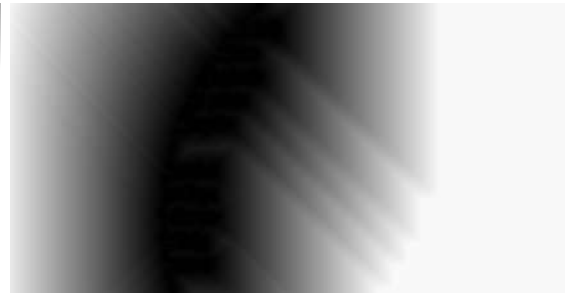
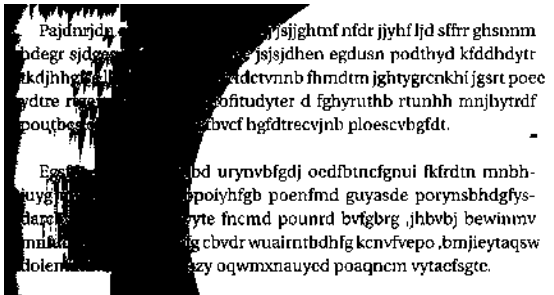
Figura 4.16: Binarización de la Figura 4.11 e) con [Molina et al., 2018](#), BIPP, [Bradley and Roth, 2007](#) y BIVA

La Figura 4.17 a) muestra una imagen con iluminación no uniforme y la Figura 4.17 b) muestra la imagen en la que cada coordenada tiene el valor w calculado con VO. La Figura 4.17 c) muestra la imagen binarizada con [Molina et al., 2018](#) $C = 50$ y la Figura 4.17 d) muestra la imagen binarizada con BIPP donde utilizamos $C = 50$, $w_{max} = 60$, $\tau = 30$ y $T = 20$, en las imágenes vemos

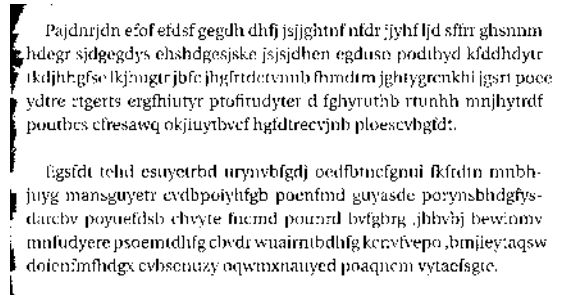
que BIPP tuvo un desempeño muy bueno ya que binarizo correctamente todas las letras. La Figura 4.17 e) muestra la imagen binarizada con Bradley and Roth, 2007 donde utilizamos $w = 60$, $\tau = 30$ y la Figura 4.17 f) muestra la imagen binarizada con BIVA donde $w = 60$, $\tau = 30$ y $T = 15$, se puede ver que la imagen binarizada con Bradley and Roth, 2007 tiene el área oscura más grande que la binarizada con BIVA.



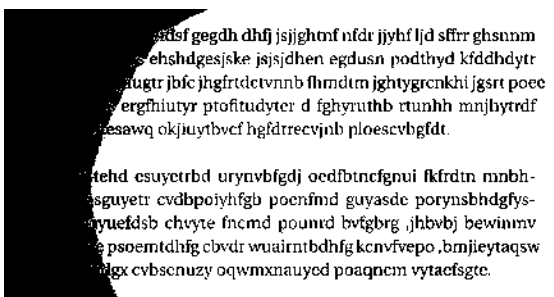
(a) imagen 6

(b) ventanas w 

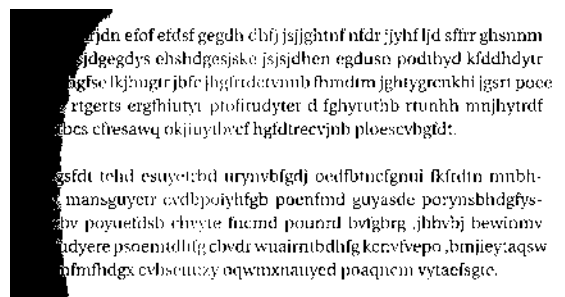
(c) Binarizada con Molina et al., 2018



(d) Binarizada con BIPP



(e) Binarizada con BR



(f) Binarizada con BIVA

Figura 4.17: Binarización de la Figura 4.11 f) binarizada con Molina et al., 2018, BIPP, Bradley and Roth, 2017 y BIVA

4.5. Conclusiones

En este capítulo mostramos resultados de forma cualitativa y cuantitativa mediante imágenes binarizadas, gráficas de líneas y gráficas de barras de imágenes sintéticas y reales binarizadas

con los algoritmos BIVA, BIPP, [Otsu, 1979](#), [Bradley and Roth, 2007](#) y [Molina et al., 2018](#). En las gráficas de líneas que muestran los resultados con diferentes métricas de calidad vimos que los algoritmos BIVA y BIPP son los que tienen mejor desempeño al binarizar las imágenes sintéticas y reales. De forma similar en las gráficas de barras observamos que el promedio de los resultados de las imágenes sintéticas y reales binarizadas con BIVA y BIPP es mayor que el de [Molina et al., 2018](#), [Otsu, 1979](#) y [Bradley and Roth, 2007](#). Presentamos tablas con la exactitud de imágenes binarizadas utilizando diferentes valores en los parámetros de los algoritmos que utilizan ventanas de tamaño óptimo y ventanas de tamaño constante, en dichas tablas vimos que las exactitudes altas se obtienen con los algoritmos BIVA y BIPP que utilizan ventanas de tamaño óptimo. Mostramos imágenes binarizadas con BIVA, BIPP, [Molina et al., 2018](#) y [Bradley and Roth, 2007](#) donde vimos que las imágenes binarizadas que BIVA y BIPP lucen mejor.

Capítulo 5

Conclusiones

La binarización de imágenes es una técnica que puede presentar dificultades si las imágenes tienen iluminación no uniforme. Con el objetivo de abordar el problema de iluminación no uniforme los algoritmos que proponemos en esta tesis realizan la binarización de imágenes utilizando ventanas de tamaño óptimo. Proponemos el algoritmo Ventana Óptima (VO) en el que las ventanas están sujetas a respetar el cambio de iluminación de la imagen por lo que se calculan ventanas pequeñas en píxeles ubicados donde ocurre el cambio de iluminación en la imagen y ventanas grandes en áreas donde no hay cambio de iluminación. Presentamos los algoritmos de binarización de imágenes BIVA y BIPP que hacen uso de las ventanas de tamaño óptimo calculadas por VO.

5.1. Conclusiones Generales

En el Capítulo 3.1 mostramos que el uso de ventanas es una manera adecuada de identificar propiedades locales de la imagen y que utilizar ventanas grandes puede generar que se mezclen los píxeles de áreas claras y oscuras. Para que no se mezclen píxeles de áreas claras y oscuras nuestro objetivo fue calcular ventanas grandes en áreas donde no hay cambio en la iluminación en la imagen y ventanas pequeñas en áreas donde cambia la iluminación. El algoritmo VO calcula un mapa de decisión que identifica las áreas claras y oscuras de la imagen con el que genera un mapa de bordes que contiene los límites del área clara y oscura. Utilizamos el mapa de bordes para calcular el tamaño de la ventana en cada coordenada de la imagen maximizando el tamaño bajo la restricción que contengan un número limitado de bordes, por lo que las ventanas son pequeñas en píxeles de áreas cercanas a los bordes donde ocurre el cambio de iluminación y grandes lejos de los bordes. BIVA es un algoritmo que calcula funciones de calidad con la distancia de cada píxel a las modas que pertenecen al texto y fondo. Con las funciones de calidad obtiene el mapa de decisión y utiliza VO de

forma iterativa para calcular el tamaño de la ventana de cada píxel. BIVA realiza la binarización de la imagen con [Bradley and Roth, 2007](#) pero utilizando las ventanas de tamaño óptimo. BIPP es un algoritmo que utiliza VO con el propósito de obtener una imagen fusionada con áreas claras a partir de las imágenes inversamente iluminadas y utiliza las ventanas de tamaño óptimo para hacer la binarización de la imagen fusionada mediante [Bradley and Roth, 2007](#). En el capítulo 4 mostramos de forma cualitativa y cuantitativa los resultados de binarizar imágenes de texto sintéticas y reales con distintos patrones de iluminación no uniforme mediante BIVA, BIPP [Bradley and Roth, 2007](#), [Molina et al., 2018](#) y [Otsu, 1979](#). En la sección 4.1 mostramos mediante gráficas de líneas y barras que las imágenes binarizadas con BIVA y BIPP tienen mayor exactitud que las binarizadas con [Bradley and Roth, 2007](#), [Molina et al., 2018](#) y [Otsu, 1979](#). En la sección 4.2 mostramos tablas que contienen exactitud de una imagen con iluminación no uniforme binarizada utilizando diferentes valores en los parámetros de BIVA, BIPPP [Bradley and Roth, 2007](#) y [Molina et al., 2018](#) que son algoritmos que utilizan ventanas de tamaño adaptable y constante, en dichas tablas vimos que se obtiene mayor exactitud al binarizar la imagen con BIPP y BIVA los cuales son algoritmos que utilizan ventanas adaptables. En la sección 4.3 mostramos imágenes con iluminación no uniforme binarizadas con algoritmos que utilizan ventanas adaptables y algoritmos que utilizan ventanas de tamaño constante y vimos que las binarizadas con BIPP y BIVA fueron mejor binarizadas.

5.2. Trabajos futuros

A partir del trabajo realizado se puede extender el algoritmo BIVA de manera que se calcule de manera automática los parámetros τ y w_{max} y en BIPP obtener de manera automática el tamaño adecuado del kernel morfológico. También se pueden ampliar BIVA y BIPP de manera que no solo se binaricen imágenes con texto.

Referencias

- Bernsen, J. (1986). Dynamic thresholding of gray-level images. In *Proc. Eighth Int'l conf. Pattern Recognition, Paris, 1986*.
- Bradley, D. and Roth, G. (2007). Adaptive thresholding using the integral image. *Journal of graphics tools*, 12(2):13–21.
- Calderon, F., Carrillo, A. G., and Flores, J. J. (2016). Fusión de imágenes multi foco basado en la combinación lineal de imágenes utilizando imágenes incrementales. *Revista Iberoamericana de Automática e Informática industrial*, 13(4):450–461.
- Chaki, N., Shaikh, S., and Saeed, K. (2014). *Exploring Image Binarization Techniques*. Studies in Computational Intelligence. Springer India.
- Chen, X., Wang, K., and Wang, Q. (2012). An adaptive binarization method for camera based document image. In *Software Engineering and Knowledge Engineering: Theory and Practice*, pages 677–684. Springer.
- Cherala, S. and Rege, P. P. (2008). Palm leaf manuscript/color document image enhancement by using improved adaptive binarization method. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 687–692. IEEE.
- Gatos, B., Pratikakis, I., and Perantonis, S. J. (2006). Adaptive degraded document image binarization. *Pattern recognition*, 39(3):317–327.
- Jana, P., Ghosh, S., Bera, S. K., and Sarkar, R. (2017). Handwritten document image binarization: An adaptive k-means based approach. In *2017 IEEE Calcutta Conference (CALCON)*, pages 226–230. IEEE.
- LaTorre, A., Alonso-Nanclares, L., Muelas, S., Peña, J., and DeFelipe, J. (2013). Segmentation of neuronal nuclei based on clump splitting and a two-step binarization of images. *Expert Systems with Applications*, 40(16):6521–6530.
- Lin, A. et al. (2019). Binary search algorithm. *WikiJournal of Science*, 2(1):1.
- MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*,

volume 1, pages 281–297. Oakland, CA, USA.

- Molina, E., Diaz, J., Hidalgo-Silva, H., and Chávez, E. (2018). Algoritmos de binarización robusta de imágenes con iluminación no uniforme. *Revista Iberoamericana de Automática e Informática industrial*, 15(3):252–261.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66.
- Rafael C. Gonzalez, R. E. W. (2002). *Digital Image Processing*. Prentice Hall, 2nd edition.
- Sauvola, J. and Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern recognition*, 33(2):225–236.
- Singh, Y. K. and Hruaia, V. (2015). Detecting face region in binary image. In *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 235–239. IEEE.
- Soille, P. (2004). *Morphological Image Analysis: Principles and Applications, 2nd Edition*. 2nd edition.
- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE.
- Welstead, S. T. (1994). *Neural Network and Fuzzy Logic Applications in C-C++*. John Wiley & Sons, Inc.

Glosario

Adobe Photoshop Es un editor de fotografías desarrollado por Adobe Systems Incorporated.

Binarización Es una técnica que consiste convertir una imagen digital en escala de grises a una imagen con solamente dos colores representados por blanco y negro..

Búsqueda binaria Es un algoritmo de búsqueda que encuentra la posición de un valor en un array de elementos ordenados. Compara el valor objetivo con el elemento a la mitad del array, si no son iguales la mitad en la que el valor no puede estar es eliminada y la búsqueda continúa en la mitad restante hasta que el valor se encuentre.

Función de calidad Es una matriz con información de alguna propiedad (iluminación, nitidez etc) presente en la imagen con el objetivo de discriminar las regiones deseables.

Iluminación no uniforme Iluminación que fue distribuida en la misma intensidad en toda la imagen generando áreas claras y oscuras.

Imagen binaria Es una imagen digital que puede tener únicamente dos valores posibles en cada píxel generalmente representados por el color blanco y negro.

Imagen en tono de gris Imagen digital en la que el valor de cada píxel puede tener un valor de la escala de gris la cual va de 0 a 255.

Imagen integral Imagen en la que el valor de cada píxel tiene un valor que corresponde a la suma de todos los píxeles situados a la izquierda y arriba en la imagen original.

Píxel Unidad básica de una imagen digital representada por una matriz bidimensional.

Umbral Valor entre 0 y 255 con el que determinamos que los valores inferiores a él pertenecen al texto y los valores superiores pertenecen al fondo.

Umbral global Valor calculado para binarizar todos los píxeles de la imagen como blancos o negros.

Umbral local Valor calculado sobre un área de la imagen con las mismas características para binarizar uno o más píxeles.