FUSIÓN DE IMÁGENES CON ILUMINACIÓN COMPLEMENTARIA

TESIS

Que para obtener el grado de

MAESTRO EN CIENCIAS EN INGENIERÍA ELÉCTRICA

presenta

Alejandro Calderón Carrillo

Dr. Félix Calderón Solorio

Director de Tesis

Universidad Michoacana de San Nicolás de Hidalgo

División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica

Agosto 2022

FUSIÓN DE IMÁGENES CON ILUMINACIÓN COMPLEMENTARIA

Los Miembros del Jurado de Examen de Grado aprueban la **Tesis de Maestría en Ciencias en Ingeniería Eléctrica** de **Alejandro Calderón Carrillo.**

Dr. Jaime Cerda Jacobo
Presidente del Jurado

Dr. Félix Calderón Solorio
Director de Tesis

Dr. José Antonio Camarena Ibarrola
Vocal

Dr. José Ortiz Bejar
Vocal

Dr. Carlos Alberto Júnez Ferreira
Revisor Externo (Fac. Ing. Civil, UMSNH)

Dr. Félix Calderón Solorio
Coordinador de Maestría en Ciencias en Ingeniería Eléctrica de la División de Estudios de Posgrado. UMSNH
(Por reconocimiento de firmas)

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO
Junio 2022

Este trabajo está dedicado a Dios quien me ha dado perseverancia, inteligencia y la oportunidad de completar esta etapa de mi vida. A mi madre y a mi familia, que con su amor, paciencia y esfuerzo me han apoyado para llegar a cumplir hoy un sueño más.

Agradezco al Consejo de Ciencia y Tecnología (CONACYT) por su apoyo financiero para concluir mis estudios de maestría. De igual manera, agradezco a la Universidad Michoacana de San Nicolás de Hidalgo (UMSNH) y a la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica por su contribución en mi formación académica. Finalmente, un gran agradecimiento a mi director de tesis, el Dr. Félix Calderón Solorio, por su tiempo, paciencia y por ser mi guía en este proyecto.

Resumen

La apariencia de una imagen no siempre puede ser la deseada, y algunas veces puede ser producida por la mala distribución de iluminación en la imagen. En este trabajo se presenta un algoritmo con el propósito de mejorar la calidad de un par de imágenes de una misma escena, en donde una imagen sufre parcialmente de una gran cantidad de iluminación y la otra sufre parcialmente de falta de iluminación. Se propone realizar una fusión de imágenes por medio de una segmentación para mejorar la calidad de cada una de estas imágenes, combinando la información deseada de cada imagen fuente para obtener una imagen fusionada con mayor cantidad de información y detalles.

Se hace una revisión de distintos algoritmos, métodos y modelos con el objetivo de realizar una segmentación de imágenes para posteriormente realizar la fusión. Basado en el estado del arte, se abordan cuatro técnicas de segmentación por medio de la binarización de imágenes, siendo estas la binarización por umbralización con el Método de Otsu y por Entropía Cruzada, binarización por agrupación con K-Medias y binarización con el modelo de Mezclas de Gaussianas. Se analizan cada una de estas técnicas de segmentación de imágenes por binarización de forma cuantitativa utilizando la métrica de exactitud para comparar su desempeño. Además, se describe un algoritmo que mejora el desempeño de la segmentación de imágenes por medio de binarización. Para la fusión de imágenes, se utiliza el algoritmo de segmentación por binarización con mejor desempeño y posteriormente se realiza un post proceso de mejora de calidad en la imagen fusionada.

Los resultados del algoritmo de fusión de imágenes propuesto son presentados haciendo pruebas utilizando imágenes sintéticas y reales. Los resultados cuantitativos se realizan utilizando imágenes sintéticas en donde se analizan las métricas de similitud de imágenes del Índice de Similitud Estructural (SSIM), la Razón del Pico de Señal a Ruido (PSNR) y el Índice Universal de Calidad de Imágenes (UQI). Los resultados cualitativos se realizan utilizando imágenes reales. Nuestros resultados muestran que el algoritmo funciona con eficacia para estas imágenes reales y sintéticas comparado con otros algoritmos de fusión. Sin embargo, existe lugar para mejorar el desempeño, la eficiencia, la facilidad y utilidad de este algoritmo propuesto.

Palabras Clave: **Segmentación de Imágenes, Binarización de Imágenes, Mejora de Imágenes, Corrección de Exposición Fotográfica, Mezcla de Imágenes**

Abstract

The appearance of an image may not always be as desired, and can sometimes be caused by poor lighting distribution in the image. In this work, an algorithm is presented with the purpose of improving the quality of a pair of images of the same scene, where one image partially suffers from a large amount of illumination and the other partially suffers from a lack of illumination. It is proposed to carry out a fusion of images by means of a segmentation to improve the quality of each of these images, combining the desired information of each source image to obtain a fused image with a greater amount of information and details.

A review of different algorithms, methods and models is made with the aim of performing image segmentation to later perform the fusion. Based on the state of the art, four segmentation techniques are addressed by means of image binarization, these being binarization by thresholding with the Otsu Method and by Cross Entropy, binarization by grouping with K-Means and binarization with the model of Gaussian mixtures. Each of these binarization image segmentation techniques are analyzed quantitatively using the accuracy metric to compare their performance. In addition, an algorithm that improves the performance of image segmentation through binarization is described. For image fusion, the binarization segmentation algorithm with the best performance is used and subsequently a quality improvement post-process is performed on the merged image.

The results of the proposed image fusion algorithm are presented by testing using synthetic and real images. Quantitative results are performed using synthetic images where the image similarity metrics of the Structural Similarity Index (SSIM), the Peak Signal to Noise Ratio (PSNR) and the Universal Image Quality Index (UQI) are analyzed. Qualitative results are made using real images. Our results show that the algorithm works effectively for these real and synthetic images compared to other fusion algorithms. However, there is room to improve the performance, efficiency, ease and usefulness of this proposed algorithm.
Keywords: **Image Segmentation, Image Binarization, Image Enhancement, Photographic Exposure Correction, Image Blending**

Contenido

Dedicatoria V
Resumen VII
Abstract IX
Contenido X
Lista de Figuras XIII
Lista de Tablas XVII
Lista de Símbolos XIX
Lista de Acrónimos XXI
Lista de Algoritmos 1

1. Introducción 1
1.1. Definición del problema 4
1.2. Marco Teórico 13
1.3. Objetivos 16
1.3.1. Objetivo general 16
1.3.2. Objetivos particulares 16
1.4. Hipótesis 17
1.5. Descripción de capítulos 17

2. Trabajo Relacionado 18
2.1. Diferentes Técnicas de Fusión de Imágenes 18
2.2. Binarización 21
2.2.1. Método de Otsu 22
2.2.2. Umbralización por Entropía Cruzada Mínima 23
2.2.3. K-Medias 25
2.2.4. Mezcla de Gaussianas 27
2.3. Desempeño de los Algoritmos de Binarización 31
2.4. Algoritmo de Ventana Óptima 40
2.4.1. Imágenes Integrales 42
2.4.2. Búsqueda Binaria 43
2.5. Resultados al Aplicar el Algoritmo de Ventana Óptima 45
2.6. Resumen del Capítulo 50

3. Algoritmo Propuesto 52
3.1. Segmentación de Imágenes por Binarización . . . 53
3.2. Fusión Preliminar de Imágenes . . . 57
3.3. Ajuste de Niveles de Iluminación . . . 60
3.4. Corrección de Bordes de las Regiones . . . 62
3.5. Algoritmo de Fusión de Imágenes con Iluminación Complementaria . . . 65
3.6. Resumen del Capítulo . . . 68

4. Pruebas y resultados 69
4.1. Medidas de Similitud de Imágenes . . . 69
4.1.1. Índice de Similitud Estructural . . . 70
4.1.2. Razón del Pico de Señal a Ruido . . . 70
4.1.3. Índice Universal de Calidad de Imágenes . . . 71
4.2. Transformación Afín para la Solución del Problema de Registro . . . 71
4.3. Pruebas con Imágenes Sintéticas . . . 73
4.3.1. Prueba 1 . . . 74
4.3.2. Prueba 2 . . . 77
4.3.3. Prueba 3 . . . 78
4.4. Pruebas con Imágenes Reales . . . 81
4.4.1. Prueba 1 . . . 81
4.4.2. Prueba 2 . . . 83
4.4.3. Prueba 3 . . . 85
4.4.4. Prueba 4 . . . 86
4.4.5. Prueba 5 . . . 87
4.4.6. Prueba 6 . . . 88
4.4.7. Prueba 7 . . . 89
4.5. Resumen del Capítulo . . . 91

5. Conclusiones y Trabajos Futuros 92
5.1. Conclusiones Generales . . . 92
5.2. Trabajos Futuros . . . 94

Referencias 95

Lista de Figuras

1.1. Luz a través de la cámara [theAtlas] . . . 3
1.2. Ejemplo de una Distribución de Probabilidad de una Imagen [thrAtlas] . . 4
1.3. Ejemplos de imágenes con Sobreexposición y Subexposición . . . 5
1.4. Efectos de la Apertura del Diafragma en una Fotografía [Mansurov20] . . . 6
1.5. Efectos de la Velocidad de Obturación en una Fotografía [cap21] . . . 7
1.6. Efectos del Valor ISO en una Fotografía [Fucci15] . . . 8
1.7. Ejemplo de Imagen a Contraluz . . . 8
1.8. Ejemplo de Imagen con Flash . . . 9
1.9. Regiones de las imágenes a fusionar. . . . 10
1.10. Fusión Sencilla . . . 12
1.11. Fusión Ideal . . . 13

2.1. Imagen $\boldsymbol{I_a}$ con $\sigma_0 = \sigma_1 = 10$ (izquierda) y su distribución de probabilidad (derecha) . . . 32
2.2. Imagen $\boldsymbol{I_b}$ con $\sigma_0 = \sigma_1 = 40$ (izquierda) y su distribución de probabilidad (derecha) . . . 32
2.3. Imagen $\boldsymbol{I_c}$ con $\sigma_0 = 10$ y $\sigma_1 = 30$ (izquierda) y su distribución de probabilidad (derecha) . . . 33
2.4. Imagen $\boldsymbol{I_d}$ con $\sigma_0 = 20$ y $\sigma_1 = 50$ (izquierda) y su distribución de probabilidad (derecha) . . . 33
2.5. Imágenes binarizadas utilizando la imagen de la Figura 2.1(a) con $\sigma_0 = \sigma_1 = 10$ 35
2.6. Imágenes binarizadas utilizando la imagen de la Figura 2.2(a) con $\sigma_0 = \sigma_1 = 40$ 36
2.7. Imágenes binarizadas utilizando la imagen de la Figura 2.3(a) con $\sigma_0 = 10$ y $\sigma_1 = 30$ . . . 37
2.8. Imágenes binarizadas utilizando la imagen de la Figura 2.4(a) con $\sigma_0 = 20$ y $\sigma_1 = 50$ . . . 38
2.9. Mapas de Decisión Obtenidas con la Binarización Utilizando Otsu y Otsu con VO para $\sigma_0 = \sigma_1 = 40$ . . . 46
2.10. Mapas de Decisión Obtenidas con la Binarización por Otsu y Otsu con VO para $\sigma_0 = 20$ y $\sigma_1 = 50$ . . . 47
2.11. Mapas de Decisión Obtenidas con la Binarización Utilizando Entropía Cruzada y Entropía Cruzada con VO para $\sigma_0 = \sigma_1 = 40$ . . . 47

2.12. Mapas de Decisión Obtenidas con la Binarización Utilizando Entropía Cruzada y Entropía Cruzada con VO para $\sigma_0 = 20$ y $\sigma_1 = 50$ 48
2.13. Mapas de Decisión Obtenidas con la Binarización de Mezcla de Gaussianas y Mezcla de Gaussianas con VO para $\sigma_0 = \sigma_1 = 40$ 49
2.14. Mapas de Decisión Obtenidas con la Binarización de Mezcla de Gaussianas y Mezcla de Gaussianas con VO para $\sigma_0 = 20$ y $\sigma_1 = 50$ 49

3.1. Ejemplo de Imágenes a Fusionar 52
3.2. Regiones de Píxeles Nítidos y No Nítidos en Imágenes en Escala de Grises . 53
3.3. Distribuciones de Probabilidad de las Imágenes en Escala de Grises 54
3.4. Mapas de Decisión de las Imágenes a Fusionar 55
3.5. Clasificación de Píxeles de la Imágenes a Fusionar con Mapas de Decisión Diferentes 56
3.6. Mapa de Decisión Final $\boldsymbol{P}$ 57
3.7. Clasificación de Píxeles de la Imágenes a Fusionar con el Mismo Mapa de Decisión 59
3.8. Imagen Fusionada Preliminar $\boldsymbol{I_F}$ 59
3.9. Histograma del Canal de Color Rojo de la Región $\boldsymbol{I_{2,1}}$ 60
3.10. Imagen Promedio $\boldsymbol{I_{Prom}}$ 61
3.11. Imagen Promedio $\boldsymbol{I_{Prom}}$ Dividida en las Regiones $\boldsymbol{I_{Prom1}}$ e $\boldsymbol{I_{Prom2}}$ Utilizando $\boldsymbol{P}$ 61
3.12. Imagen Fusionada con Ajuste de Iluminación 62
3.13. Imágenes de Bordes 63
3.14. Imagen con Filtro de Mediana $\boldsymbol{I_m}$ 64
3.15. Imagen Fusionada Final $\boldsymbol{I_F^*}$ 64

4.1. Imágenes para la Fusión de la Prueba 1 74
4.2. Imágenes en el Proceso de Fusión 76
4.3. Imagen Fusionada de las Imágenes Sintéticas de la Prueba 1 77
4.4. Imágenes para la Fusión de la Prueba 2 78
4.5. Imagen Fusionada de las Imágenes Sintéticas de la Prueba 2 78
4.6. Imágenes para la Fusión de la Prueba 3 79
4.7. Imagen Fusionada de las Imágenes Sintéticas de la Prueba 3 80
4.8. Imágenes Reales a Fusionar de la Prueba 1 82
4.9. Imágenes en el Proceso de Fusión 83
4.10. Imágenes Reales a Fusionar de la Prueba 2 84
4.11. Imagen Fusionada de las Imágenes Reales de la Prueba 2 84
4.12. Imágenes Reales a Fusionar de la Prueba 3 85
4.13. Imagen Fusionada de las Imágenes Reales de la Prueba 3 85
4.14. Imágenes Reales a Fusionar de la Prueba 4 86
4.15. Imagen Fusionada de las Imágenes Reales de la Prueba 4 86
4.16. Imágenes Reales a Fusionar de la Prueba 5 87
4.17. Imagen Fusionada de las Imágenes Reales de la Prueba 5 87
4.18. Imágenes Reales a Fusionar de la Prueba 6 88
4.19. Imagen Fusionada de las Imágenes Reales de la Prueba 6 89

4.20. Imágenes Reales a Fusionar de la Prueba 7 90
4.21. Imagen Fusionada de las Imágenes Reales de la Prueba 7 90

Lista de Tablas

2.1. Ventajas y Desventajas de Diferentes Métodos de Fusión de Imágenes [Rani13] 20
2.2. Resultados de Binarización de la Imagen de la Figura 2.1 con $\sigma_0 = \sigma_1 = 10$ 38
2.3. Resultados de Binarización de la Imagen de la Figura 2.2 con $\sigma_0 = \sigma_1 = 40$ 38
2.4. Resultados de Binarización de la Imagen de la Figura 2.3 con $\sigma_0 = 10$ y $\sigma_1 = 30$ 38
2.5. Resultados de Binarización de la Imagen de la Figura 2.4 con $\sigma_0 = 20$ y $\sigma_1 = 50$ 39
2.6. Resultados de Binarización de la Imagen Sintética con $\sigma_0 = \sigma_1 = 40$ 50
2.7. Resultados de Binarización de la Imagen Sintética con $\sigma_0 = 20$ y $\sigma_1 = 50$. 50

4.1. Resultados de Fusión con Imágenes Sintéticas 80

Lista de Símbolos

$\mathbf{I}$	Imagen digital
$\mathbf{I}[r, c]$	Píxel de la imagen $\mathbf{I}$ ubicado en las coordenadas $[r, c]$
$\mathbf{I_1}$	Imagen con sobreexposición.
$\mathbf{I_2}$	Imagen con subexposición.
$\mathbf{I_{1,0}}$	Región de píxeles con sobreexposición de la imagen $\mathbf{I_1}$.
$\mathbf{I_{1,1}}$	Región de píxeles nítidos de la imagen $\mathbf{I_1}$.
$\mathbf{I_{2,0}}$	Región de píxeles con subexposición de la imagen $\mathbf{I_2}$.
$\mathbf{I_{2,1}}$	Región de píxeles nítidos de la imagen $\mathbf{I_2}$.
$\mathbf{G}$	Imagen en escala de grises.
$\mathbf{I_B}$	Imagen binaria.
p_i	Probabilidad de ocurrencia del i-ésimo valor en una imagen.
N_r	Número de renglones de una imagen.
N_c	Número de columnas de una imagen.
n_i	Total de píxeles con valor i.
u	Valor umbral.
w_0	Probabilidad de ocurrencia de la clase 0.
w_1	Probabilidad de ocurrencia de la clase 1.
μ_0	Media de la clase 0.
μ_1	Media de la clase 1.
σ_B^2	Varianza interclase.
K	Cantidad de grupos.
$\mathcal{N}$	Función de una distribución Gaussiana.
$\boldsymbol{B}$	Mapa de bordes.
$\boldsymbol{W}$	Matriz de tamaños de ventana.
w_{max}	Tamaño de ventana máximo.
T	Tolerancia de bordes dentro de una ventana.
$\boldsymbol{I_F}$	Imagen fusionada preliminar.
$\boldsymbol{I_{Prom}}$	Imagen promedio de $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$.
$\boldsymbol{B^*}$	Mapa de bordes con filtro de mediana.
$\boldsymbol{I_m}$	Imagen $\boldsymbol{I_F}$ con filtro de mediana.
$\boldsymbol{I_F^*}$	Imagen fusionada final.
$\boldsymbol{H}$	Matriz de Transformación Afín.

Lista de Acrónimos

RGB: (del inglés "Red, Green and Blue") modelo de color basado en síntesis aditiva.

GMM: (del inglés "Gaussian Mixture Model") Modelo de Mezcla de Gaussianas.

EM: (del inglés "Maximización de Expectativas") algoritmo Expectación Maximización.

ISO: (del inglés "International Organization for Standardization") Organización Internacional de Estandarización.

DWT: (del inglés "Discrete Wavelet Transform") Transformada Wavelet Discreta.

CLI-S: Combinación Lineal de Imágenes Simple.

VO: Algoritmo de Ventana Óptima.

PCA: (del inglés "Principal Component Analysis") Análisis de Componentes Principales.

SSIM: (del inglés "Structural Similarity Index Measure") Indice de Similitud Estructural.

PSNR: (del inglés "Peak Signal to Noise Ratio") Razón del Pico de Señal a Ruido.

UQI: (del inglés "Universal Quality Index") Índice Universal de Calidad de Imágenes.

Lista de Algoritmos

1. Otsu(Entrada: D_P; Salida: u) 23
2. K-Medias(Entrada: $\boldsymbol{G}$, K, $Iter_{max}$; Salida: $\boldsymbol{E}$) 27
3. EM(Entrada: X, K, $Iter_{max}$; Salida: θ) 30
4. Busqueda_Binaria(Entrada: r, c,w_{max},$\boldsymbol{H_B}$, T; Salida: w_m) 44
5. VO(Entrada: $\boldsymbol{P}$,T,w_{max}; Salida: $\boldsymbol{W}$) 44
6. Fusion(Entrada: $\boldsymbol{I_1}$,$\boldsymbol{I_2}$,T,w_b,w_m; Salida: $\boldsymbol{I_F^*}$) 65

Capítulo 1

Introducción

En el ámbito de la fotografía, la baja calidad de una imagen generalmente no es deseable ya que se quiere tener capturado en una fotografía cada detalle de la escena. En computación, una imagen con baja calidad se puede traducir como una imagen que contiene poca información. La baja calidad de una imagen puede darse por varios factores, por ejemplo, la falta o exceso de luz en la escena, las condiciones del ambiente en la escena a capturar (como niebla, humo y contaminación en general) o incluso por la estructura física de la cámara (como su baja resolución, sensores dañados, etc.). Sin embargo, existe una variedad de técnicas para mejorar la calidad de una imagen. Por ejemplo, en este trabajo se presenta una técnica llamada fusión de imágenes utilizada para mejorar la calidad de un par de imágenes.

Al tener distintas tomas de una misma escena se pueden combinar estas imágenes para obtener una imagen de mejor calidad, a esto se le llama fusión de imágenes. Formalmente, la fusión de imágenes es un proceso que combinar la información deseada de un conjunto de imágenes de una misma escena en una sola imagen y la imagen fusionada resultante deberá ser más informativa y completa que cualquiera de las imágenes originales [Rani13].

El sensor de una cámara es un elemento digital que captura la luz y después la cámara lo guarda como un valor numérico de luminosidad, por lo general un valor entero entre 0 y 255, siendo 0 luz nula y 255 la máxima cantidad de luz que puede captar el sensor.

Una cámara contiene un arreglo bidimensional de sensores que captura la luz de una escena y genera un arreglo bidimensional de valores numéricos de luminosidad llamada imagen digital, a la cual denotaremos como $\boldsymbol{I}$. A cada elemento de esta matriz se le conoce como píxel y es representado como $\boldsymbol{I}[r, c]$, donde r es la posición del renglón y c es la posición de la columna. Una imagen a color comúnmente es representada en el modelo RGB (del inglés "Red, Green, Blue") donde cada píxel contiene 3 valores, uno para cada color, esto significa que:

$$\boldsymbol{I}[r, c] = \Big[\boldsymbol{R}[r, c], \boldsymbol{G}[r, c], \boldsymbol{B}[r, c]\Big] \tag{1.1}$$

donde $\boldsymbol{R}[r, c]$ es el valor del color rojo, $\boldsymbol{G}[r, c]$ es el valor del color verde y $\boldsymbol{B}[r, c]$ es el valor del color azul para el píxel de la imagen $\boldsymbol{I}$ en las coordenadas $[r, c]$.

Por otro lado, en una imagen en escala de grises cada píxel tiene solamente un valor. A una imagen en escala de grises la denotaremos como $\boldsymbol{G}$ y podemos realizar la conversión de una imagen RGB a una en escala de grises calculando el piso del promedio de los 3 colores para cada píxel mediante:

$$\boldsymbol{G}[r, c] = \lfloor \frac{\boldsymbol{R}[r, c] + \boldsymbol{G}[r, c] + \boldsymbol{B}[r, c]}{3} \rfloor \tag{1.2}$$

La exposición en fotografía es la cantidad de luz que recibe la cámara a través de su lente [theAtlas]. Cuando la luz pasa a través de la lente hacia el sensor de la cámara durante un período de tiempo determinado, la cantidad de luz que llega a la cámara determinará el aspecto final de la imagen, como se muestra en la Figura 1.1. El triángulo de la exposición fotográfica es una analogía para explicar los elementos principales que afectan a la exposición en una fotografía como es la apertura de diafragma, velocidad de obturación y el ISO (del inglés "International Organization for Standardization").

La apertura de diafragma determinará la cantidad de luz que pasará a través del lente hacia el sensor de la cámara [crehana]. Al usar una apertura más amplia entrará más luz a la cámara y si se cierra la apertura, disminuirá la cantidad de luz y como resultado se obtendrá una imagen más oscura. A la velocidad de obturación también se le conoce como

tiempo de exposición y es el tiempo que el obturador de la cámara estará abierto capturando luz. Mientras mayor tiempo esté el obturador recibiendo luz, la imagen será más brillosa y cuando el obturador se abre y cierra rápidamente, el resultado es una fotografía más oscura. Por último, el ISO no se refiere directamente a una organización, sino a una medida estándar de amplificación de la luz capturada por el sensor de la cámara [theAtlas]. El ISO aumenta de forma artificial la exposición de la imagen que ya ha sido capturada.

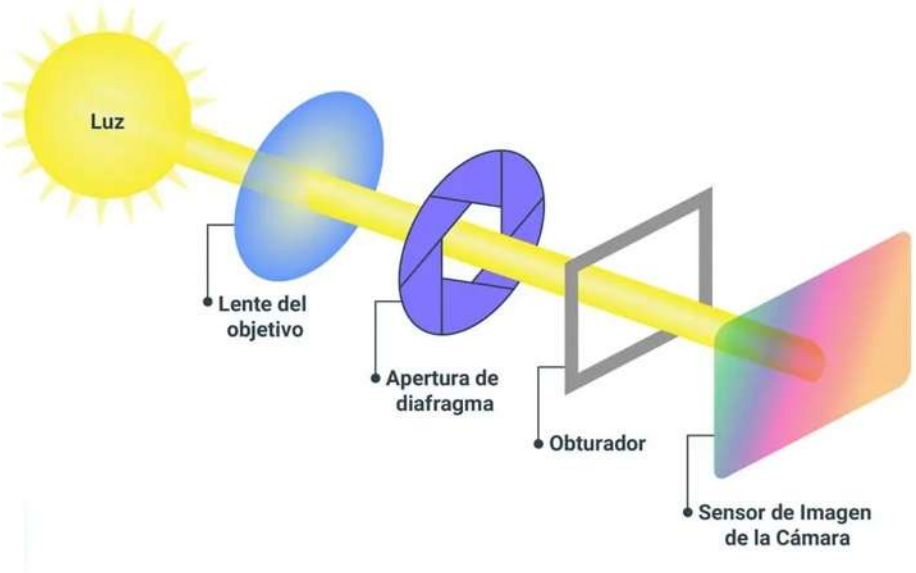


Figura 1.1: Luz a través de la cámara [theAtlas]

Una herramienta útil para observar gráficamente la exposición de iluminación de una imagen es mediante el uso de una distribución de probabilidad. Gráficamente una distribución de probabilidad representa en su eje horizontal el rango de valores que van desde 0 a 255 para cada canal de color y el eje vertical representa la probabilidad de ocurrencia de cada valor. En la Figura 1.2 se muestra un ejemplo de una distribución de probabilidad de una imagen cualquiera. La probabilidad de ocurrencia es una medida de la certidumbre de que ocurra un evento [NCERT19]. La probabilidad de ocurrencia del i-ésimo valor de un píxel es calculada como (1.3).

$$p_i = \frac{n_i}{N_r N_c} \tag{1.3}$$

Donde p_i es la probabilidad de que el i-ésimo valor ocurra, n_i es el total de píxeles con el i-ésimo valor, N_r es el número de renglones de la imagen y N_c es el número de columnas de la imagen.

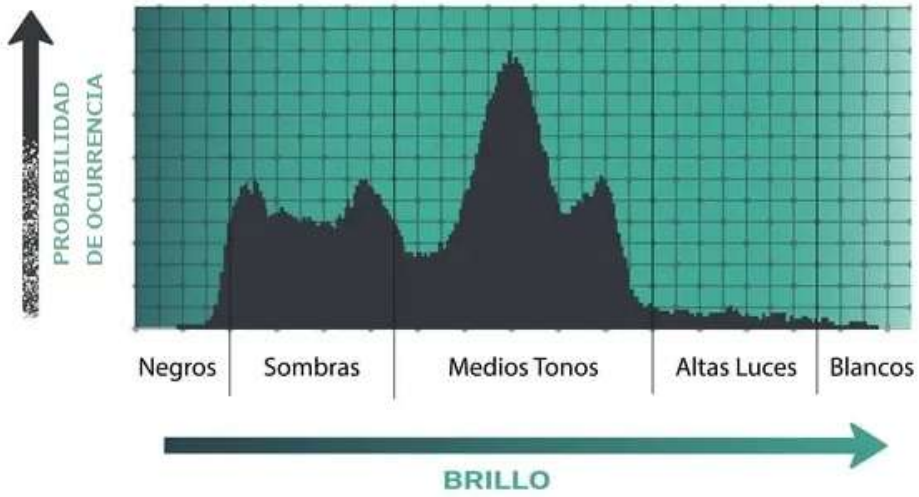
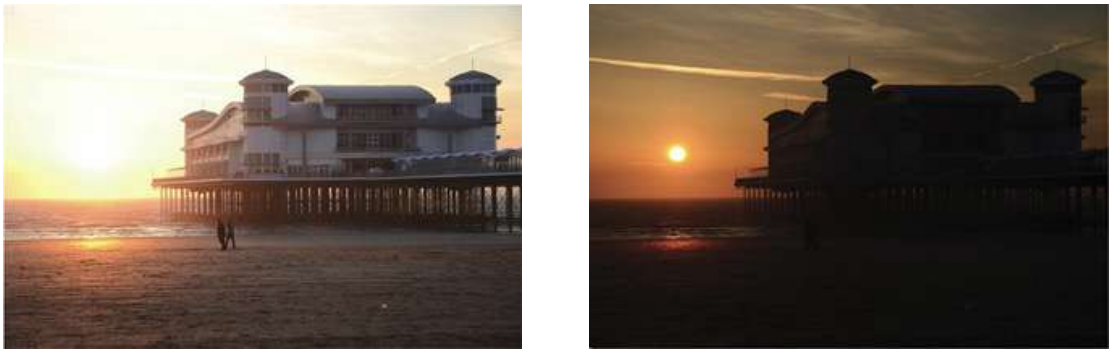


Figura 1.2: Ejemplo de una Distribución de Probabilidad de una Imagen [thrAtlas]

1.1. Definición del problema

Dentro de los factores que contribuyen a la degradación de una imagen se encuentra la mala exposición fotográfica en la escena capturada. Algunos de estos factores de degradación de una imagen son la sobreexposición y la subexposición en los sensores de la cámara. La sobreexposición en una imagen se produce cuando el sensor de la cámara recibe demasiada luz y en las regiones píxeles correspondientes a estos sensores se pierden los detalles de la imagen. Por otro lado, la subexposición de una imagen se da cuando los sensores de la cámara reciben muy poca luz, lo cual hace que las regiones de píxeles correspondientes a estos sensores pierdan los detalles también. A los píxeles que no son afectados por la sobreexposición o la subexposición los denominamos como píxeles nítidos. Estos píxeles son nítidos de forma subjetiva, ya que no utilizamos alguna medida de nitidez y al compararlos con los píxeles que sufren de la sobreexposición y subexposición, los detalles en estas regiones son de mayor calidad y contienen mayor información.

Según Naina Dhingra, las imágenes con subexposición pueden verse afectadas por el ruido, mientras que las imágenes con sobreexposición pueden sufrir opacidad por movimiento [Dhingra15]. Para los propósitos de este trabajo, a una imagen con sobreexposición la denotaremos por $\boldsymbol{I_1}$ y una imagen con subexposición la denotaremos con $\boldsymbol{I_2}$. En la Figura 1.3(a) se muestra un ejemplo de una imagen con sobreexposición y en la Figura 1.3(b) se muestra un ejemplo de una imagen con subexposición.



(a) ***Ejemplo de Imagen con Sobreexposición*** $\boldsymbol{I_1}$

(b) ***Ejemplo de Imagen con Subexposición*** $\boldsymbol{I_2}$

Figura 1.3: Ejemplos de imágenes con Sobreexposición y Subexposición

En la imagen de la Figura 1.3(a) se puede ver que existe una región de píxeles nítidos en la parte de la casa en el muelle junto con la playa y una región de píxeles sobreexpuestos en la parte del cielo. En la imagen de la Figura 1.3(b) se puede ver lo contrario, existe una región de píxeles nítidos en la parte del cielo y una región de píxeles subexpuestos en la parte de la casa y la playa.

Algunas de las causas de la sobreexposición y subexposición de píxeles en una imagen son los parámetros contenidos en el triángulo de la exposición fotográfica, es decir, la apertura de diafragma, velocidad de obturación y el ISO. Cada uno de estos parámetros controlan la apariencia de la fotografía, sin embargo, todos estos 3 parámetros afectan a la distribución de la iluminación en una imagen. Estos parámetros de la cámara no siempre pueden ser modificados, por lo tanto, no siempre se puede capturar la fotografía ideal.

Según Nasim Mansurov, la apertura del diafragma tiene diferentes efectos en la fotografía [Mansurov20]. Uno de los más importantes es el brillo o la exposición de las imágenes. A medida que la apertura cambia de tamaño, altera la cantidad total de luz que llega al sensor de la cámara y, por lo tanto, el brillo de la imagen. Una gran apertura dejará pasar mucha luz, lo que dará como resultado una fotografía más iluminación. Una pequeña apertura hace todo lo contrario, oscureciendo una foto. En la imagen de la Figura 1.4 se muestra el efecto de la exposición de iluminación en una fotografía cuando se tiene una gran apertura (izquierda), una apertura mediana (centro) y una pequeña apertura (derecha). Se puede observar que al tener una gran apertura se obtiene una imagen con píxeles sobreexpuestos y al tener una pequeña apertura se obtiene una imagen con píxeles subexpuestos.

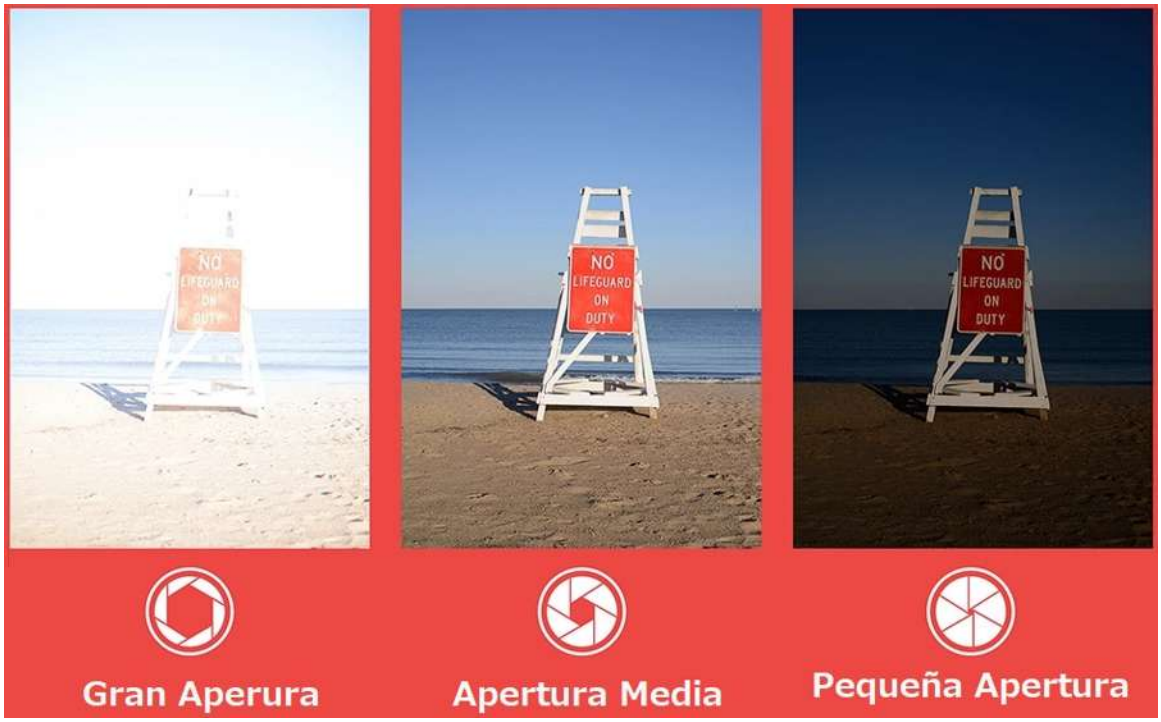


Figura 1.4: Efectos de la Apertura del Diafragma en una Fotografía [Mansurov20]

La velocidad de obturación en fotografía es el tiempo que el obturador de la cámara está abierto para capturar la fotografía [cap21]. El efecto de la velocidad de obturación en la fotografía es el cambio en la exposición. Al dejar el obturador de la cámara abierto por

un período de tiempo largo o corto, el sensor capturará más o menos luz respectivamente, lo que se traducirá en una exposición más brillante o más oscura. Cuanto más rápida sea la velocidad del obturador, más oscura será la imagen y cuanto más lenta sea la velocidad del obturador, más brillante será la imagen. En la Figura 1.5 se muestra un ejemplo del efecto de distintas velocidades de obturación en la exposición de una fotografía. En la figura se puede observar una escala que va desde 2 segundos de exposición en el lado izquierdo de la imagen, hasta $\frac{1}{2000}$ de segundo de exposición en el lado derecho de la imagen.

Figura 1.5: Efectos de la Velocidad de Obturación en una Fotografía [cap21]

El ISO tiene un impacto directo en la sensibilidad del sensor de la cámara a la luz [Fucci15]. Cuanto más alto es el ISO, más sensible es el sensor a la luz, por lo que cuanto mayor es la exposición y da como resultado una imagen con más iluminación. Por otro lado, mientras más bajo es el ISO, la exposición es menor, lo que produce una imagen mas obscura. En la Figura 1.6 se muestran varias imágenes tomadas con distintos valores de ISO.

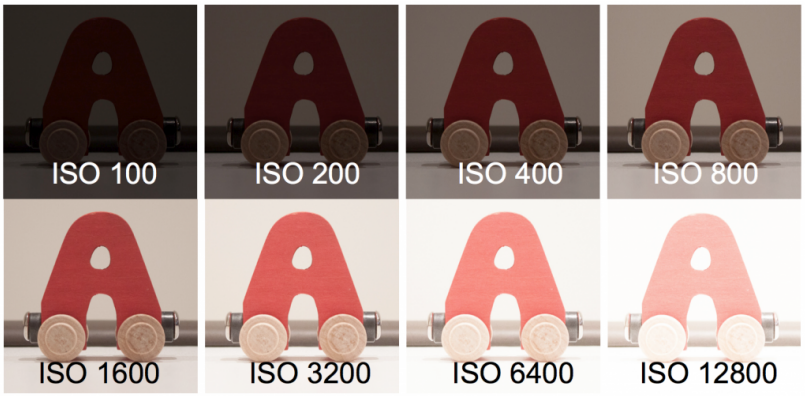


Figura 1.6: Efectos del Valor ISO en una Fotografía [Fucci15]

Otra de las causas que provocan la subexposición en una imagen se le conoce como el efecto de contraluz. De acuerdo con Ian Plant, la contraluz es una técnica fotográfica en la que la cámara apunta directamente hacia una fuente de luz [Plant17]. Este efecto produce diferentes resultados sobre la imagen, como un fuerte contraste entre luz y oscuridad, esconde detalles de la escena, crea siluetas y acentúa las líneas y formas. En la Figura 1.7 se muestra una imagen a contraluz, en donde se tiene como fuente de luz el sol detrás de una persona.

Figura 1.7: Ejemplo de Imagen a Contraluz

La apertura del diafragma, la velocidad de obturación, y el ISO no siempre son modificables en cada cámara, esto es más común en las cámaras antiguas. Las cámaras antiguas no profesionales sólo tenían una luz llamada "Flash"para aumentar la iluminación de la escena. Sin embargo, el Flash ilumina los objetos cerca de la cámara y algunas veces puede obscurecer los objetos lejanos a la cámara. Algunas de las veces se puede obtener también una imagen con sobreexposición al iluminar demasiado la escena con el Flash. Los efectos de utilizar el Flash se pueden observar en la imagen de la Figura 1.8. En esta imagen se puede apreciar que la mujer un el centro de la imagen tiene una sobreexposición y el fondo de la escena esta obscuro.

Figura 1.8: Ejemplo de Imagen con Flash

Si se tiene una imagen sobreexpuesta, como la de la Figura 1.3(a), y una imagen subexpuesta, como la de la Figura 1.3(b), de una misma escena, consideremos realizar una fusión de imágenes en donde combinemos este par de imágenes para obtener una imagen fusionada que contenga los píxeles nítidos de ambas imágenes.

Al analizar una imagen sobreexpuesta o subexpuesta se puede identificar dos regiones de píxeles en la imagen. Una región contiene píxeles nítidos y la otra región contiene

píxeles muy brillantes o muy obscuros en donde es difícil captar los detalles de esa parte de la imagen. A partir de la cantidad de iluminación se pueden separarán los píxeles nítidos de los píxeles subexpuestos o sobreexpuestos. Si dividimos la imagen sobreexpuesta de la Figura 1.3(a) en las regiones de píxeles nítidos y píxeles sobreexpuestos, y la imagen subexpuesta de la Figura 1.3(b) en las regiones de píxeles nítidos y píxeles subexpuestos, obtenemos las imágenes presentadas en la Figura 1.9. En la Figura 1.9(a) se muestra la región de píxeles sobreexpuestos de la imagen $\boldsymbol{I_1}$, a la cual llamaremos $\boldsymbol{I_{1,0}}$. En la Figura 1.9(b) se muestra la región de píxeles nítidos de la imagen $\boldsymbol{I_1}$, a la cual llamaremos $\boldsymbol{I_{1,1}}$. En la Figura 1.9(c) se muestra la región de píxeles subexpuestos de la imagen $\boldsymbol{I_2}$, a la cual llamaremos $\boldsymbol{I_{2,0}}$. En la Figura 1.9(d) se muestra la región de píxeles sobreexpuestos de la imagen $\boldsymbol{I_2}$, a la cual llamaremos $\boldsymbol{I_{2,1}}$. A este proceso de dividir una imagen en diferentes regiones de píxeles se le conoce como segmentación de imágenes.

(a) ***Región Sobreexpuesta*** $\boldsymbol{I_{1,0}}$

(b) ***Región Nítida*** $\boldsymbol{I_{1,1}}$

(c) ***Región Subexpuesta*** $\boldsymbol{I_{2,0}}$

(d) ***Región Nítida*** $\boldsymbol{I_{2,1}}$

Figura 1.9: Regiones de las imágenes a fusionar.

El objetivo de la segmentación de imágenes es simplificar la representación de imágenes en información significativa mediante la partición en regiones de la imagen [Ho11]. Formalmente, la segmentación de imágenes se refiere al proceso de dividir una imagen en varios segmentos o regiones de píxeles [Dass12]. Los píxeles en una región son similares según algunos criterios de homogeneidad como color, intensidad o textura, y son utilizados para localizar e identificar objetos y fronteras en una imagen.

En el caso de las imágenes $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$, se desea realizar una segmentación en dos regiones para cada imagen y de esta manera obtener las regiones de píxeles que se muestran en la Figura 1.9. Al realizar una segmentación de una imagen en dos regiones podemos considerar realizar una binarización. La binarización consiste en separar los píxeles de una imagen en dos grupos. A un grupo se le asigna el color blanco (un valor de 0) y al otro el color negro (un valor de 255), obteniendo de esta forma una imagen binaria a la cual llamaremos Mapa de Decisión y la denotaremos por $\boldsymbol{P}$. A partir de una imagen en escala de grises $\boldsymbol{G}$ es muy simple realizar una binarización. La forma más sencilla es establecer un valor umbral u que permita asociar cada píxel de la imagen al color blanco o al negro. Por lo tanto, la binarización por umbral se define como (1.4). El mapa de decisión $\boldsymbol{P}$ determina qué píxeles corresponden a cada región de las imágenes a fusionar.

$$\boldsymbol{P}[r,c] = \begin{cases} 0 & \text{Si } \boldsymbol{G}[r,c] < u \\ 255 & \text{Si no} \end{cases} \tag{1.4}$$

Al analizar las regiones de píxeles nítidos $\boldsymbol{I_{1,1}}$ e $\boldsymbol{I_{2,1}}$ podemos asumir a simple vista que estas regiones son complementarias en cuanto a su distribución espacial y de iluminación. De aquí el nombre de "Fusión de Imágenes con Iluminación Complementaria". Entonces, podemos considerar un solo mapa de decisión $\boldsymbol{P}$ para ambas imágenes a fusionar ya que son complementarias en el espacio. Para realizar la fusión de una imagen sobreexpuesta y una subexpuesta, una forma muy sencilla e intuitiva consiste en realizar la segmentación de imágenes mediante la binarización de una imagen y posteriormente juntar las regiones $\boldsymbol{I_{1,1}}$ e $\boldsymbol{I_{2,1}}$ en una sola imagen, como se muestra en la Figura 1.10. La nueva imagen fusionada la llamaremos $\boldsymbol{I_F}$. Esta imagen fusionada la podemos calcular por medio del mapa de decisión $\boldsymbol{P}$. Para realizar la fusión de las imágenes se toma en cuenta que si

$\boldsymbol{P}[r,c] = 0$ entonces el píxel $\boldsymbol{I_F}[r,c] = \boldsymbol{I_1}[r,c]$. Por otro lado, si $\boldsymbol{P}[r,c] = 255$ entonces el píxel $\boldsymbol{I_F}[r,c] = \boldsymbol{I_2}[r,c]$. Por lo tanto, el cálculo de la imagen fusionada $\boldsymbol{I_F}$ se puede calcular por medio de (1.5).

$$\boldsymbol{I_F}[r,c] = \frac{\boldsymbol{I_1}[r,c]\,(255 - \boldsymbol{P}[r,c]) + \boldsymbol{I_2}[r,c]\boldsymbol{P}[r,c]}{255} \tag{1.5}$$

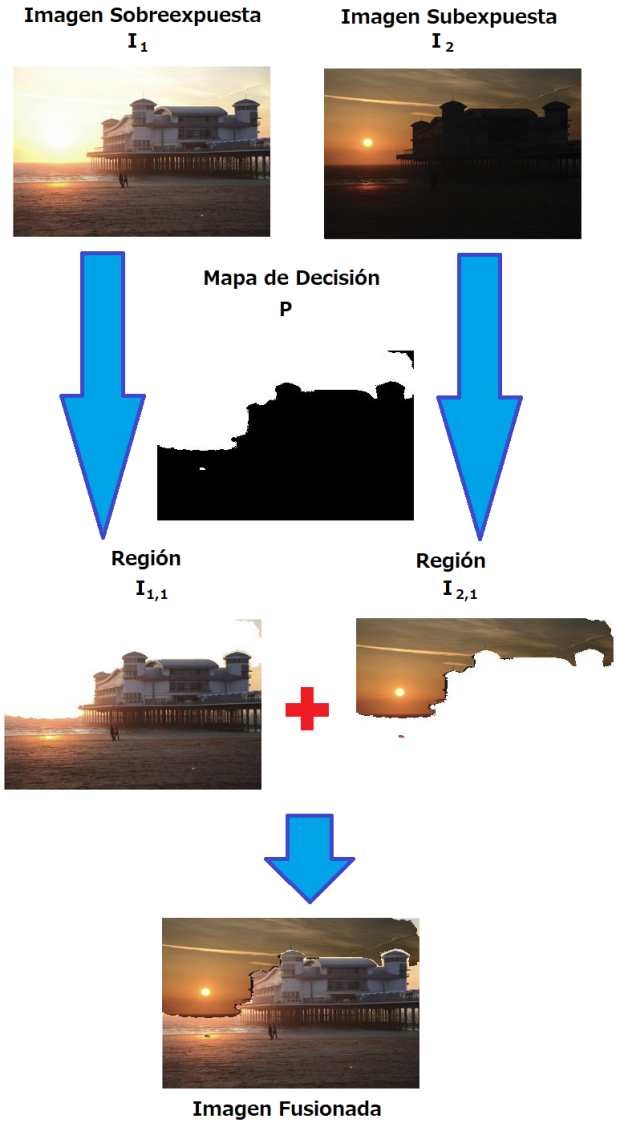


Figura 1.10: Fusión Sencilla

En (1.5) se puede ver en el numerador que si $\boldsymbol{P}[r,c] = 0$, el resultado es $\boldsymbol{I_F}[r,c] = \frac{\boldsymbol{I_1}[r,c](255)}{255} = \boldsymbol{I_1}[r,c]$. En cambio, si $\boldsymbol{P}[r,c] = 255$, el resultado es $\boldsymbol{I_F}[r,c] = \frac{\boldsymbol{I_2}[r,c](255)}{255} = \boldsymbol{I_2}[r,c]$. De esta manera se garantiza que cada píxel de la imagen fusionada tenga un valor perteneciente a alguna de las imágenes fuente en la misma coordenada.

Sin embargo, al observar la imagen resultante de esta fusión en la Figura 1.10 se puede apreciar que la iluminación de cada una de las regiones no es del todo complementaria. Esto es más evidente en la frontera que une a estas regiones. En la borde de las regiones, se puede observar una inconsistencia en la iluminación. Por lo tanto, para obtener una imagen fusionada ideal, como la que se muestra en la Figura 1.11, proponemos realizar un post-procesamiento a la imagen fusionada preliminar para ajustar la iluminación.

Figura 1.11: Fusión Ideal

En este trabajo, el problema de fusión de una imagen con sobreexposición y una imagen con subexposición se reduce a realizar una segmentación de imágenes de forma automática para aislar las regiones de píxeles nítidos de cada una de las imágenes y posteriormente realizar la fusión de imágenes combinando estas dos regiones. Después, será necesario realizar un proceso que ajuste la inconsistencia de la iluminación de la imagen fusionada preliminar.

1.2. Marco Teórico

Los diferentes métodos de fusión de imágenes se dividen en tres niveles: nivel de píxel, nivel de característica y nivel de decisión. En la fusión de imágenes a nivel de píxel

(bajo nivel), el proceso de fusión trabaja directamente en los píxeles de las imágenes fuente y son preferidos cuando las imágenes a fusionar provienen de sensores similares [Dhingra15]. Un ejemplo es la fusión de imágenes multienfoque que consiste en fusionar imágenes que enfocan objetos diferentes de una misma escena para obtener una imagen nítida de todos los objetos [Calderón16]. La fusión de imágenes a nivel de característica (nivel medio) trabaja con las características extraídas de las imágenes fuente y dependen mucho del tipo de imagen [Dhingra15]. Un ejemplo es la fusión de imágenes satelitales a diferentes resoluciones, como Google Maps al ampliar los mapas [Shuang08]. La fusión de imágenes a nivel de decisión (nivel alto) trabajan en el nivel mas alto y combina la interpretación de diferentes objetos obtenidos de las imágenes fuente. Los algoritmos a nivel de decisión son adecuados cuando las imágenes a ser fusionadas son obtenidas por sensores muy diferentes [Dhingra15]. Un ejemplo es la fusión de imágenes visibles e infrarrojas de cámaras de vigilancia [Kolekar15].

El método de fusión de imágenes sobreexpuestas y subexpuestas presentado en el trabajo de Naina Dhingra considera las imágenes como un conjunto difuso para posteriormente realizar la fusión utilizando DWT (del inglés “Discrete Wavelet Transform”) [Dhingra15].

El algoritmo de CLI-S (Combinación Lineal de Imágenes Simple) propuesto por Félix Calderón tiene como propósito fusionar imágenes multifoco [Calderón16]. El algoritmo calcula la nitidez de los píxeles para realizar una fusión a nivel píxel. De ese trabajo extraemos la idea de combinar dos imágenes por medio de la ecuación (1.5).

Dentro de la fusión de imágenes a nivel píxel, comúnmente se utiliza algún algoritmo de segmentación de imágenes. La segmentación de imágenes es el proceso de particionar una imagen en un conjunto de regiones o segmentos de píxeles. Según Linda Shapiro y George Stockman, las regiones de la segmentación de una imagen deben ser uniformes y homogéneas con respecto a alguna característica, como el nivel de gris, el color o la textura [Shapiro00]. Generalmente estas regiones son simples y no tienen muchos huecos dentro de ellas. También, las fronteras de cada región suelen ser suaves, no irregulares y deben ser precisas en cuanto al espacio que ocupan. A cada una de estas regiones de píxeles se les asigna una etiqueta, la cual es única e identifica a cada región. La segmentación de imágenes es muy utilizada en tareas de reconocimiento y detección de objetos. Existen diferentes enfoques

en cuanto a los métodos de segmentación de imágenes como la umbralización, métodos de agrupación, métodos basados en histogramas, segmentación basada en modelos, entre otros. Uno de los algoritmos o métodos más famosos para realizar una segmentación de imagen es el algoritmo K-Medias [MacQueen67]. El algoritmo K-Medias se basa en agrupamiento de datos, sin embargo, se puede utilizar para segmentar imágenes tomando en cuanta que los datos son los valores de los píxeles y cada grupo es una región o segmento. Existes otros algoritmos y modelos que sirven de apoyo al realizar una segmentación tales como el método de Otsu [Otsu79], umbralización por entropía cruzada mínima [Li93] y el modelo de mezcla de Gaussianas [Chatzis08]. Estas técnicas de segmentación se describen brevemente a continuación.

El método de Otsu (llamado así por Nobuyuki Otsu) es un algoritmo utilizado para establecer un umbral de forma automática, el cual puede ser utilizado para realizar una segmentación de imágenes por medio de una binarización [Otsu79]. El algoritmo calcula un umbral maximizando la varianza entre dos clases.

El algoritmo de umbralización por entropía cruzada mínima es un método que utiliza un esquema de iteración de un punto [Li93], similar al método de Otsu. Éste es un método iterativo que minimiza la entropía cruzada entre la imagen segmentada y la imagen original.

El algoritmo de K-Medias divide N puntos en K grupos y representa cada grupo por un punto central [MacQueen67]. El objetivo de K-Medias es minimizar las distancias cuadradas dentro de los cúmulos hacia el punto central.

Un modelo de mezcla de gaussianas, también conocido como GMM (del inglés "Gaussian Mixture Model"), es un modelo probabilístico que asume que los datos se modelarán mediante una mezcla de K distribuciones gaussianas ajustados a la distribución de N datos [Chatzis08]. El algoritmo EM (del inglés "Expectation Maximization") es un método para encontrar las medias y las matrices de covarianza a partir de las observaciones [Dempster97]. Para realizar una segmentación podemos suponer que cada gaussiana representa la probabilidad de pertenecer a una clase.

El algoritmo VO (Ventana Óptima) del trabajo de Cristina Reyes propuesto por Félix Calderón tiene como objetivo calcular de forma automática ventanas de tamaño ópti-

mo sujetas a respetar los bordes del cambio de iluminación en una imagen para su binarización [Reyes21]. Por lo tanto, al aplicar el algoritmo VO podremos aumentar la exactitud de la segmentación de la imagen.

1.3. Objetivos

Desarrollar un algoritmo que realice una fusión de dos imágenes, una imagen que contenga algunos píxeles sobreexpuestos y otra imagen que contenga algunos píxeles subexpuestos.

1.3.1. Objetivo general

El objetivo de este trabajo es implementar un algoritmo que realice automáticamente la fusión de un par de imágenes correspondientes píxel a píxel donde una imagen contenga píxeles sobreexpuestos y la otra imagen contenga píxeles subexpuestos para obtener una imagen fusionada que contenga más detalles e información deseada que las imágenes originales.

1.3.2. Objetivos particulares

- Analizar y comparar diferentes algoritmos de segmentación de imágenes.
- Implementar el algoritmo óptimo de segmentación de imágenes para obtener las regiones de píxeles subexpuestos, píxeles sobreexpuestos y píxeles nítidos.
- Afinar las fronteras de las regiones de píxeles obtenidas en la segmentación de imágenes mediante el algoritmo de Ventanas Óptimas.
- Fusionar las dos regiones de píxeles nítidos de la imagen con sobreexposición y la imagen con subexposición.
- Mejorar la calidad de la imagen fusionada.

1.4. Hipótesis

Al combinar las regiones de píxeles nítidos de una imagen con sobreexposición y una imagen con subexposición en una sola imagen y posteriormente aplicar las correcciones necesarias, se obtendrá una imagen resultante con más detalles e información deseada que cada una de las imágenes originales.

1.5. Descripción de capítulos

En el Capítulo 1 se presenta una introducción acerca de la fusión de imágenes, se describió el problema principal de este trabajo y la propuesta de solución. En el Capítulo 2 se exponen algunos trabajos relacionados con la segmentación y fusión de imágenes, y se describen algunos de los algoritmos más populares. En el Capítulo 3 se presenta el algoritmo propuesto para realizar la fusión de imágenes. En el Capítulo 4 se exponen las pruebas y resultados de la fusión de imágenes con imágenes reales y sintéticas. Por último, en el Capítulo 5 se presentan las conclusiones de este trabajo junto con trabajos que se pueden realizar en un futuro.

Capítulo 2

Trabajo Relacionado

La fusión de imágenes se define como la combinación de información de dos o más imágenes para generar una sola imagen mucho más informativa. La fusión de imágenes se utiliza para integrar la información complementaria multitemporal, multivista y multisensor en una sola imagen con calidad mejorada en la cual se mantiene la integridad de las características importantes. En el estado del arte existe una gran diversidad de métodos, técnicas y algoritmos para realizar fusión de imágenes. En este capítulo se exponen brevemente algunos algoritmos acerca de la fusión de imágenes y algunas otras técnicas muy utilizadas dentro de la fusión de imágenes, como lo es la segmentación de imágenes mediante binarización.

2.1. Diferentes Técnicas de Fusión de Imágenes

El trabajo presentado por Kusum Rani hace una revisión de distintas técnicas de fusión de imágenes [Rani13]. El tipo de técnica a utilizar para realizar una fusión de imágenes depende de la naturaleza de las imágenes a fusionar. Las imágenes a fusionar pueden ser multisensor, multimodal, multifoco o multitemporal. Uno de los pre-procesos a contemplar antes de realizar la fusión de imágenes es el registro de imágenes. Este proceso consiste en transformar los datos en un solo sistema de coordenadas. Los métodos de fusión de imágenes puede dividirse en dos grupos: métodos en el dominio espacial y métodos en el dominio espectral. Los métodos en el dominio espacial trabajan directamente con los píxeles

de las imágenes. Los métodos en el dominio espectral transforman la imagen a un dominio de frecuencias, tomando las imágenes como señales.
Algunos de los algoritmos de fusión de imágenes simples analizados en el trabajo de Kusum Rani son:

- Método Máximo Simple: la imagen fusionada resultante se obtiene seleccionando la máxima intensidad de los píxeles correspondientes de ambas imágenes originales [Rani08].

- Método Mínimo Simple: la imagen fusionada resultante se obtiene seleccionando la mínima intensidad de los píxeles correspondientes de ambas imágenes originales.

- Método Promedio Simple: la imagen fusionada resultante se obtiene tomando el promedio de la intensidad de los píxeles correspondientes en ambas imágenes originales [Malviya09].

- Método Promedio Ponderado: la imagen fusionada resultante se obtiene tomando el promedio ponderado de la intensidad de los píxeles correspondientes en ambas imágenes originales.

- Análisis de Componentes Principales: (PCA por sus siglas en inglés) este método reduce los conjuntos de datos multidimensionales a una dimensión inferior para analizarlos. Este método determina los pesos de cada imagen fuente utilizando el eigenvector correspondiente al eigenvalor más grande de la matriz de covarianza de cada imagen de origen [Naidu08].

- Método con Transformada Discreta Wavelet: (DWT por sus siglas en inglés) descompone las imágenes fuente en diferentes escalas o resoluciones a partir de diferentes bandas de frecuencia [Patel11].

Algunas de las ventajas y desventajas de cada uno de estos métodos se muestran a continuación en la Tabla 2.1.

Método de Fusión	Dominio	Ventaja	Desventaja
Máximo y Mínimo Simple [Rani08]	Espacial	La imagen fusionada está altamente enfocada.	La imagen fusionada puede no ser tan completa o coherente.
Promedio Simple y Promedio Ponderado [Malviya09]	Espacial	Implementación sencilla	La imagen fusionada puede resultar borrosa.
PCA [Naidu08]	Espacial	Proporciona imágenes nítidas con alta resolución espacial.	Se presenta degradación espacial en la imagen fusionada.
DWT [Patel11]	Espectral	La imagen fusionada tiene poco ruido.	Menos resolución espacial.
Combinación de DWT y PCA [M .Chandana11]	Espectral	La imagen fusionada contiene una alta resolución con alta calidad en el contenido espectral.	Es un método muy complejo.

Tabla 2.1: Ventajas y Desventajas de Diferentes Métodos de Fusión de Imágenes [Rani13]

En el trabajo de Félix Calderón se expone la fusión de imágenes multifoco utilizando un algoritmo llamado Combinación Lineal de Imágenes Simple (CLI-S) [Calderón16]. Esta fusión de imágenes combina un par de imágenes de una misma escena para obtener una sola imagen nítida y totalmente enfocada. El par de imágenes a fusionar son una imagen enfocada en primer plano, denotada por $\boldsymbol{I_1}$, y una imagen enfocada en segundo plano, denotada por $\boldsymbol{I_2}$. Cada una de estas imágenes contiene dos tipos de regiones: una región de píxeles nítidos y otra región de píxeles fuera del foco. Utilizando un kernel que modela la perdida de nitidez y considerando la coherencia espacial de las imágenes, se calcula una mapa binario $\boldsymbol{P}$. La coherencia espacial significa ser consistente o conectado, en el contexto de visión computacional significa que dentro de una región pequeña de píxeles no se tiene grandes cambios. Posteriormente, se construye la imagen fusionada $\boldsymbol{I_F}$ mediante (2.1) en donde combina los píxeles nítidos de cada imagen.

$$\boldsymbol{I_F}[r,c] = \boldsymbol{I_1}[r,c]\boldsymbol{P}[r,c] + \boldsymbol{I_2}[r,c](1 - \boldsymbol{P}[r,c]) \qquad \forall [r,c] \tag{2.1}$$

En la Tabla 2.1 se puede observar que la fusión de imágenes en el dominio espacial es mucho más sencilla que en el dominio espectral. Además, las imágenes que deseamos fusionar contienen 2 regiones de píxeles (siendo la región con píxeles nítidos y la región con píxeles sobreexpuestos o subexpuestos) al igual que en la fusión de imágenes multifoco (siendo la región de píxeles enfocados y la región de píxeles fuera del foco). Teniendo en cuenta estas dos características de las imágenes a fusionar, podemos hacer una analogía al algoritmo de fusión de imágenes multifoco y considerar realizar la fusión de imágenes con iluminación complementaria construyendo la imagen fusionada de una forma similar a (2.1). Para esto deberemos construir una mapa binario apropiado que permita construir una imagen fusionada a partir de una imagen sobreexpuesta y otra imagen subexpuesta. Al proceso de construcción de esta matriz binaria a partir de una imagen se le conoce como binarización.

2.2. Binarización

La binarización es el proceso de convertir una imagen $\boldsymbol{G}$ a una imagen en blanco y negro $\boldsymbol{I_B}$. Usualmente la binarización consiste en comparar cada píxel de una imagen con un umbral para determinar si el píxel de la imagen resultante se clasificará como blanco o negro. Si un píxel de una imagen en escala de grises $\boldsymbol{G}[r,c]$ es menor a un valor umbral u, el píxel correspondiente de la nueva imagen es igual a 0, de lo contrario es igual a 255.

$$\boldsymbol{I_B}[r,c] = \begin{cases} 0, & \boldsymbol{G}[r,c] < u \\ 255, & \text{Si no} \end{cases} \tag{2.2}$$

Sin embargo, el problema radica en establecer un valor para u. Para obtener el valor de u, existe un distintos métodos como algoritmo llamado Método de Otsu [Otsu79] y el de Umbralización por Entropía Cruzada Mínima [Li93]. Estos algoritmos calculan automáticamente un umbral para separar un conjunto de datos en dos grupos.

La binarización es una forma de realizar una segmentación de imagen, ya que es equivalente a dividir la imagen en dos regiones (uno representado por el blanco y otro por el negro). Entonces, podemos hacer uso de distintos algoritmos y modelos tales como K-Medias [MacQueen67] y Mezcla de Gaussianas [Chatzis08] para realizar la binarización por medio de una segmentación de imágenes.

2.2.1. Método de Otsu

El método de Otsu [Otsu79] calcula un umbral de forma automática para la binarización de una imagen en escala de grises. El algoritmo calcula un umbral que maximice la varianza interclase, es decir, la separabilidad de dos clases a partir de una distribución de probabilidad de una imagen. Supongamos que dividimos los píxeles de una imagen en 2 clases, la clase C_0 y la clase C_1. La probabilidad de ocurrencia de la clase C_0 esta dada por (2.3) y la probabilidad de ocurrencia de la clase C_1 esta dada por (2.4) donde p_i es la probabilidad de ocurrencia del i-ésimo tono de gris y es calculado como (1.3).

$$w_0(u) = \sum_{i=0}^{u} p_i \tag{2.3}$$

$$w_1(u) = \sum_{i=u+1}^{255} p_i \tag{2.4}$$

La media de la clase C_0 se calcula mediante (2.5) y la media de la clase C_1 mediante (2.6) como una suma ponderada de su valor en la escala de grises y su probabilidad de ocurrencia.

$$\mu_0(u) = \frac{\sum_{i=0}^{u} i p_i}{w_0(u)} \tag{2.5}$$

$$\mu_1(u) = \frac{\sum_{i=u+1}^{255} i p_i}{w_1(u)} \tag{2.6}$$

La varianza interclase está definida como:

$$\sigma_B^2(u) = w_0(u) w_1(u) [\mu_0(u) - \mu_1(u)]^2 \tag{2.7}$$

El valor óptimo de u, denotado por u^*_{max}, es obtenido realizando una búsqueda exhaustiva sobre todos los posibles valores de u que maximice la varianza interclase y es calculado como

(2.8). Para una imagen en escala de grises el rango de búsqueda es de 0 a 255.

$$u^*_{max} = \underset{0 \leq u \leq 255}{\operatorname{argmax}} \left(\sigma^2_B(u)\right) \tag{2.8}$$

El método de Otsu se muestra en el Algoritmo 1. Este algoritmo recibe un arreglo D_P con la distribución de probabilidad de una imagen en escala de grises, es decir $D_P = [p_0, p_1, p_2, \ldots, p_{255}]$, y entrega como resultado el valor óptimo del umbral u^*_{max}. El algoritmo hace uso de una variable auxiliar v_{max} para guardar el valor máximo de la varianza intercalase.

Algoritmo 1: Otsu(Entrada: D_P; Salida: u)

$u^*_{max} \leftarrow 0$
$v_{max} \leftarrow 0$
para $u = 0$ *a* 255 **hacer**
  $w_0(u) \leftarrow \sum_{i=0}^{u} \boldsymbol{D_P}[i]$
  $w_1(u) \leftarrow \sum_{i=u+1}^{255} \boldsymbol{D_P}[i]$
  $\mu_0(u) \leftarrow \frac{\sum_{i=0}^{u} i\boldsymbol{D_P}[i]}{w_0(u)}$
  $\mu_1(u) \leftarrow \frac{\sum_{i=u+1}^{255} i\boldsymbol{D_P}[i]}{w_1(u)}$
  $\sigma^2_B(u) \leftarrow w_0(u)w_1(u)[\mu_0(u) - \mu_1(u)]^2$
  si $\sigma^2_B(u) > v_{max}$ **entonces**
    $u^*_{max} \leftarrow u$
    $v_{max} \leftarrow \sigma^2_B(u)$
  fin
fin
devolver u^*_{max}

2.2.2. Umbralización por Entropía Cruzada Mínima

El algoritmo de Umbralización por Entropía Cruzada Mínima formula un umbral que minimice la distancia teórica de la información entre dos imágenes [Li93]. En teoría de la información, la entropía cruzada entre dos distribuciones de probabilidad mide la diferencia entre estas distribuciones. Esta medida es la distancia Kullback-Leibler, calculada por (2.9). Esta distancia mide que tan diferente es una distribución de probabilidad p de

una distribución de probabilidad q.

$$D(p,q) = p(i)\log\frac{p(i)}{q(i)} \tag{2.9}$$

Para un histograma h definido en el rango de escala de grises $[0, L]$, el cero y primer momento de primer plano y de fondo del histograma con umbral t son respectivamente (2.10) y (2.11).

$$\begin{aligned} m_{0a}(t) &= \sum_{i=0}^{t-1} h(i) \\ m_{0b}(t) &= \sum_{i=t}^{L-1} h(i) \end{aligned} \tag{2.10}$$

$$\begin{aligned} m_{1a}(t) &= \sum_{i=0}^{t-1} ih(i) \\ m_{1b}(t) &= \sum_{i=t}^{L-1} ih(i) \end{aligned} \tag{2.11}$$

Las medias de las porciones están definidas como (2.12).

$$\begin{aligned} \mu_a(t) &= \frac{m_{1a}(t)}{m_{0a}(t)} \\ \mu_b(t) &= \frac{m_{1b}(t)}{m_{0b}(t)} \end{aligned} \tag{2.12}$$

El método de entropía cruzada mínima selecciona un umbral que minimice la entropía cruzada de la imagen y su versión segmentada por el umbral t.La función de criterio es (2.13). El umbral óptimo t_{op} es dado por el minimizador de Ecuación (2.13) como (2.14).

$$\eta(t) = -m_{1a}(t)\log(\mu_a(t)) - m_{1b}(t)\log(\mu_b(t)) \tag{2.13}$$

$$t_{op} = arg\min_{t}\ \eta(t) \tag{2.14}$$

El calculo de t_{op} involucra la evaluación de $\eta(t)$ para todos los posibles valores del umbral t. Se observa que una condición necesaria para el mínimo de $\eta(t)$ se da al establecer

la derivada de $\eta(t)$ en cero. La derivada de $\eta(t)$ se expresa en (2.15). Para que $\eta'(t)$ sea cero, $h(t)$ o el segundo término en la Ecuación (2.15) es cero. La solución que se busca esta en los ceros del segundo término dado que si $h(t) = 0$ seria una solución trivial. Igualando el segundo término y simplificándolo, se obtiene (2.16).

$$\eta'(t) = h(t)\left(t\log\frac{\mu_a(t)}{\mu_b(t)} - (\mu_a(t) - \mu_b(t))\right) \tag{2.15}$$

$$t = \frac{\mu_b(t) - \mu_a(t)}{\log(\mu_b(t)) - \log(\mu_a(t))} \tag{2.16}$$

Aplicando el método de iteración de un punto a la Ecuación (2.16), se obtiene el procedimiento (2.17) para calcular el umbral óptimo, donde la función *round* realiza el redondeo al entero más cercano.

$$t_{n+1} = round\left(\frac{\mu_b(t_n) - \mu_a(t_n)}{\log(\mu_b(t)) - \log(\mu_a(t))}\right) \tag{2.17}$$

2.2.3. K-Medias

El algoritmo de K-Medias consiste en agrupar un conjunto de N elementos en K grupos de forma que se minimice la suma de las distancias de entre los elementos de cada grupo y el centro o media del grupo al que pertenecen [MacQueen67]. El algoritmo inicia asignando cada elemento aleatoriamente a uno los grupos de manera que cada grupo contenga al menos un elemento. K-Medias es un algoritmo iterativo que consiste en dos pasos principales: la asignación y la actualización. En el paso de asignación cada elemento es asignado al grupo con la media más cercana. En el paso de actualización se calcula nuevamente la media de cada grupo. El ciclo puede terminar termina cuando el valor de las medias de cada grupo no tengan cambio alguno de una iteración a la siguiente o cuando se realicen un determinado número de iteraciones.

Para este trabajo se realiza una división de píxeles de una imagen en escala de grises $\boldsymbol{G}$=$[G_0, G_1, G_2, \ldots, G_{K-1}]$ en K grupos. Cada grupo se denota como G_k para $k = 0, 1, 2, \ldots, K-1$ donde k es el identificador del grupo. La media del k-ésimo grupo μ_k se calcula mediante (2.18) como la suma de todos los píxeles en el grupo G_k entre la

cardinalidad del grupo G_k, es decir la cantidad de píxeles en el grupo G_k. En (2.19) se calcula la distancia de un píxel $\boldsymbol{G}[r, c]$ y la media μ_k para $k = 0, 1, 2, \ldots, K-1$ como el valor absoluto de la diferencia de el valor del píxel y la media del grupo, en donde el símbolo # se refiere al operador de cardinalidad.

$$\mu_k = \frac{\sum_{\boldsymbol{G}[r,c] \in G_k} \boldsymbol{G}[r, c]}{\#(G_k)} \tag{2.18}$$

$$d(\mu_k, \boldsymbol{G}[r, c]) = |\boldsymbol{G}[r, c] - \mu_k| \tag{2.19}$$

El algoritmo de K-Medias se muestra en el Algoritmo 2. Este algoritmo recibe una imagen en escala de grises $\boldsymbol{G}$, la cantidad de grupos K y un limite de iteraciones $Iter_{max}$. El algoritmo devuelve una matriz bidimensional $\boldsymbol{E}$ donde $\boldsymbol{E}[r, c]$ contiene la etiqueta del grupo asignado al píxel $\boldsymbol{G}[r, c]$. Además, el algoritmo utiliza una variable auxiliar d_{min} la cual sirve para guardar la distancia hacia la media del grupo más cercano. La función $random(inicio, fin)$ regresa un número entero aleatorio mayor o igual a *inicio* y menor o igual a fin.

Algoritmo 2: K-Medias(Entrada: $\boldsymbol{G}$, K, $Iter_{max}$; Salida: $\boldsymbol{E}$)

para $r = 0$ ***a*** $N_r - 1$ **hacer**
  para $c = 0$ ***a*** $N_c - 1$ **hacer**
    $\boldsymbol{E}[r, c] \leftarrow random(0, K - 1)$; `// Inicialización`
  fin
fin
para $iter = 0$ ***a*** $Iter_{max} - 1$ **hacer**
  para $k = 0$ ***a*** $K - 1$ **hacer**
    $\mu_k \leftarrow \frac{\sum_{\boldsymbol{G}[r,c] \in G_k} \boldsymbol{G}[r,c]}{|G_k|}$; `// Paso de Actualización`
  fin
  para $r = 0$ ***a*** $N_r - 1$ **hacer**
    para $c = 0$ ***a*** $N_c - 1$ **hacer**
      $d_{min} \leftarrow d(\mu_0, \boldsymbol{G}[r, c])$
      para $k = 0$ ***a*** $K - 1$ **hacer**
        si $d(\mu_k, \boldsymbol{G}[r, c]) < d_{min}$ **entonces**
          $d_{min} \leftarrow d(\mu_k, \boldsymbol{G}[r, c])$
          $\boldsymbol{E}[r, c] \leftarrow k$; `// Paso de Asignación`
        fin
      fin
    fin
  fin
fin
devolver $\boldsymbol{E}$

2.2.4. Mezcla de Gaussianas

La Mezcla de Gaussianas es un modelo probabilístico que representa K subgrupos dentro de un grupo de datos, donde cada subgrupo corresponde a una Distribución Gaussiana. Por lo tanto, la distribución de probabilidad de todos los datos será una mezcla o suma ponderada de K distribuciones Gaussianas. El objetivo del modelo de mezcla de

Gaussianas es encontrar una aproximación o estimación de los parámetros $\theta = \{\mu, \sigma^2, \Pi\}$ que describen a cada una de las Gaussianas de la mezcla a partir de las observaciones o datos. Estos parámetros son:

- Las medias $\mu = [\mu_0, \mu_1, \ldots, \mu_{K-1}]$ de cada distribución Gaussiana de la mezcla.
- Las varianzas $\sigma^2 = [\sigma_0^2, \sigma_1^2, \ldots, \sigma_{K-1}^2]$ de cada distribución Gaussiana de la mezcla.
- Las probabilidades $\Pi = [\Pi_0, \Pi_1, \ldots, \Pi_{K-1}]$ de que un dato pertenezca a cada una de las Gaussianas de la mezcla.

En un espacio unidimensional, la función de densidad de probabilidad de una distribución Gaussiana esta dada por (2.20) donde x_n es un dato escalar, μ_k es la media de la Gaussiana y σ_k^2 es la varianza de la Gaussiana.

$$\mathcal{N}(x_n|\mu_k, \sigma_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}} \tag{2.20}$$

La distribución de probabilidad de los datos en la mezcla de Gaussianas se define por (2.21) como una suma ponderada de distribuciones Gaussianas para un grupo de datos $X = [x_0, x_1, \ldots, x_N]$, donde $p(x_n)$ es la probabilidad de ocurrencia de un dato x_n.

$$p(x_n) = \sum_{k=0}^{K-1} \Pi_k \mathcal{N}(x_n|\mu_k, \sigma_k) \tag{2.21}$$

Un algoritmo muy utilizado para encontrar los parámetros θ de las Gaussianas de la mezcla en la distribución de probabilidad es el algoritmo Expectation Maximization (EM) [Dempster97]. El algoritmo EM utiliza los datos existentes para determinar los valores óptimos de los parámetros del modelo de Mezcla de Gaussianas. El algoritmo consiste en dos pasos:

- **Paso de Estimación:** asigna cada dato a una Gaussiana de forma probabilística.
- **Paso de Maximización:** actualiza los parámetros θ para maximizar la log-verosimilitud en función de los datos asignados a cada Gaussiana en el paso de Estimación.

La verosimilitud se calcula mediante (2.22) como la probabilidad conjunta de todos los datos de X.

$$p(X) = \prod_{n=0}^{N-1} p(x_n) = \prod_{n=0}^{N-1} \sum_{k=0}^{K-1} \Pi_k \mathcal{N}(x_n|\mu_k, \sigma_k) \tag{2.22}$$

La log-verosimilitud se calcula mediante (2.23).

$$\ln p(X) = \sum_{n=0}^{N-1} \ln \sum_{k=0}^{K-1} \Pi_k \mathcal{N}(x_n|\mu_k, \sigma_k) \tag{2.23}$$

Antes de empezar con el paso de Estimación, se inicializan los parámetros θ. Estos pueden ser inicializados a partir del resultado del algoritmo de K-Medias suponiendo que cada grupo corresponde a una Gaussiana de la mezcla. En el paso de Estimación, calculamos la probabilidad de que un dato x_n pertenezca a la k-ésima Gaussiana mediante (2.24).

$$p(\Pi_k|x_n) = \frac{\Pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)}{\sum_{j=0}^{K-1} \Pi_k \mathcal{N}(x_n|\mu_j, \sigma_j)} \tag{2.24}$$

En el paso de Maximización, actualizamos Π_k mediante (2.25), μ_k mediante (2.26) y σ_k^2 mediante (2.27).

$$\Pi_k = \frac{\sum_{n=0}^{N-1} p(\Pi_k|x_n)}{N} \tag{2.25}$$

$$\mu_k = \frac{\sum_{n=0}^{N-1} p(\Pi_k|x_n) x_n}{\sum_{n=0}^{N-1} p(\Pi_k|x_n)} \tag{2.26}$$

$$\sigma_k^2 = \frac{\sum_{n=0}^{N-1} p(\Pi_k|x_n)(x_n - \mu_k)^2}{\sum_{n=0}^{N-1} p(\Pi_k|x_n)} \tag{2.27}$$

El algoritmo EM se presenta en el Algoritmo 3, el cual recibe un grupo de datos X, la cantidad de Gaussianas K en la mezcla y un limite de iteraciones $Iter_{max}$. El algoritmo

regresa los parámetros del modelo de Mezcla de Gaussianas θ.

Algoritmo 3: EM(Entrada: X, K, $Iter_{max}$; Salida: θ)

```
Inicializar θ = {μ, σ², Π}
para iter = 0 a Iter_max hacer
    para k = 0 a K − 1 hacer
        para n = 0 a N − 1 hacer
            p(π_k|x_n) ← Π_k N(x_n|μ_k,σ_k) / Σ_{j=0}^{K−1} Π_k N(x_n|μ_j,σ_j)
        fin
    fin
    para k = 0 a K − 1 hacer
        Π_k ← Σ_{n=0}^{N−1} p(Π_k|x_n) / N
        μ_k ← Σ_{n=0}^{N−1} p(Π_k|x_n) x_n / Σ_{n=0}^{N−1} p(Π_k|x_n)
        σ_k² ← Σ_{n=0}^{N−1} p(Π_k|x_n)(x_n − μ_k)² / Σ_{n=0}^{N−1} p(Π_k|x_n)
    fin
fin
devolver θ = {μ, σ², Π}
```

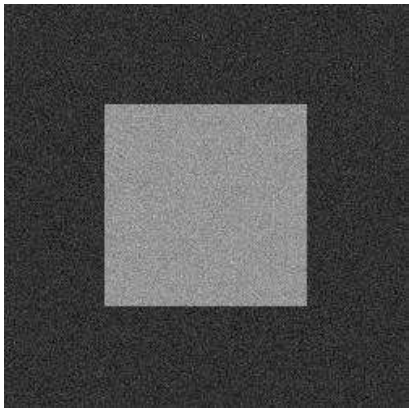
Después de haber obtenido los parámetros θ del modelo de Mezcla de Gaussianas, podemos asignar cada dato x_n a un grupo, es decir a una Gaussiana de la mezcla. Un dato x_n es asignado a la Gaussiana en donde se tenga la mayor verosimilitud. Además, se tiene un arreglo de etiquetas $E = [e_0, e_1, \ldots, e_N]$ tal que la etiqueta e_n determina el grupo que se le asigna a x_n. Entonces, la asignación del grupo un dato x_n esta dado por (2.28).

$$e_n = \begin{cases} k & \text{Si } \underset{0 \leq j \leq K-1}{\operatorname{argmax}} \left(\Pi_j \mathcal{N}(x_n|\mu_j, \sigma_j)\right) = [\mu_k, \sigma_k^2, \Pi_k] \qquad \text{para } k = 0, 1, \ldots, K-1 \\ 0 & \text{Si no} \end{cases} \tag{2.28}$$

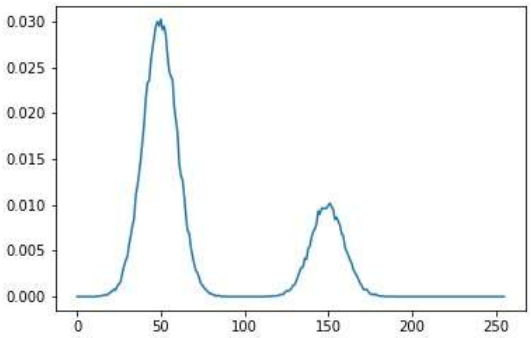
2.3. Desempeño de los Algoritmos de Binarización

Para comparar el desempeño entre los algoritmos de binarización anteriormente mencionados en este capítulo, se generan 4 imágenes sintéticas diferentes. Cada imagen tiene un tamaño de 250 renglones por 250 columnas y una distribución de probabilidad formada por dos distribuciones normales, donde cada distribución normal representa una clase. La distribución de probabilidad de cada una de estas imágenes contiene dos modas, una moda para cada distribución normal o clase, las cuales en un histograma se pueden observar como los dos picos más representativos y evidentes. A estas clases las llamaremos clase 0 y clase 1. La clase 0 se encuentra en los píxeles con las coordenadas $[r, c]$ para $62 \leq r \leq 187$ y $62 \leq c \leq 187$ y el resto de píxeles corresponden a la clase 1. El 25 % de cada imagen corresponde a la clase 0 y el 75 % a la clase 1.

Cada clase contiene valores aleatorios tomados de una distribución normal. La media de la clase 0 la denotaremos por μ_0 y su desviación estándar por σ_0. Para la clase 1, su media la denotaremos por μ_1 y su desviación estándar por σ_1. En las 4 imágenes, la clase 0 tiene una media $\mu_0 = 50$ y la clase 1 tiene una media $\mu_1 = 150$. La desviación estándar de las clases cambia en cada imagen. La primera imagen denotada por $\boldsymbol{I_a}$ tiene las desviaciones estándar $\sigma_0 = \sigma_1 = 10$ y se muestra en la Figura 2.1(a). En la Figura 2.1(b) se muestra la distribución de probabilidad de la imagen $\boldsymbol{I_a}$. La segunda imagen denotada por $\boldsymbol{I_b}$ tiene las desviaciones estándar $\sigma_0 = \sigma_1 = 40$ y se muestra en la Figura 2.2(a). En la Figura 2.2(b) se muestra su distribución de probabilidad de la imagen $\boldsymbol{I_b}$. La tercera imagen denotada por $\boldsymbol{I_c}$ tiene las desviaciones estándar $\sigma_0 = 10$ y $\sigma_1 = 30$, la cual se muestra en la Figura 2.3(a). En la Figura 2.3(b) se muestra su distribución de probabilidad. Por último, la cuarta imagen denotada por $\boldsymbol{I_d}$ tiene las desviaciones estándar $\sigma_0 = 20$ y $\sigma_1 = 50$, y se muestra en la Figura 2.4(a). En la Figura 2.4(b) se muestra la distribución de probabilidad de $\boldsymbol{I_d}$.

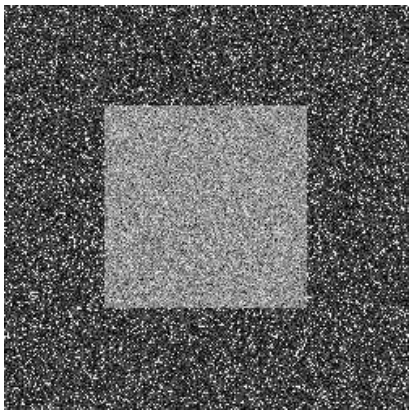


(a) ***Imagen*** I_a ***con*** $\sigma_0 = \sigma_1 = 10$

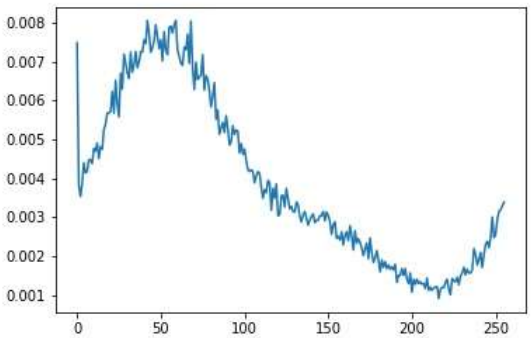


(b) ***Distribución de Probabilidad de*** I_a

Figura 2.1: Imagen $\boldsymbol{I_a}$ con $\sigma_0 = \sigma_1 = 10$ (izquierda) y su distribución de probabilidad (derecha)

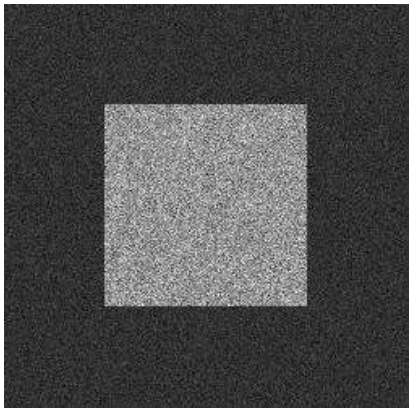


(a) ***Imagen*** I_b ***con*** $\sigma_0 = \sigma_1 = 40$

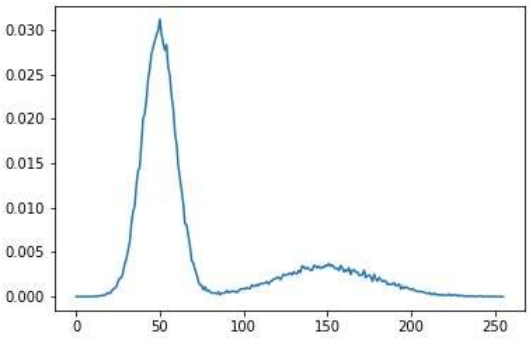


(b) ***Distribución de Probabilidad de*** I_b

Figura 2.2: Imagen $\boldsymbol{I_b}$ con $\sigma_0 = \sigma_1 = 40$ (izquierda) y su distribución de probabilidad (derecha)

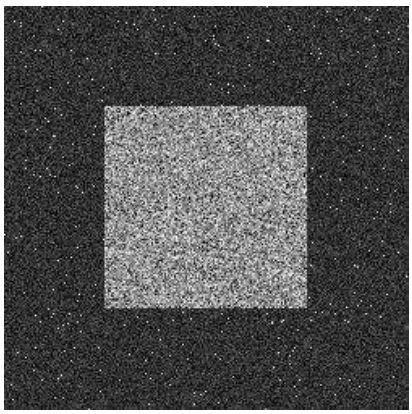


(a) *Imagen I_c con $\sigma_0 = 10$ y $\sigma_1 = 30$*

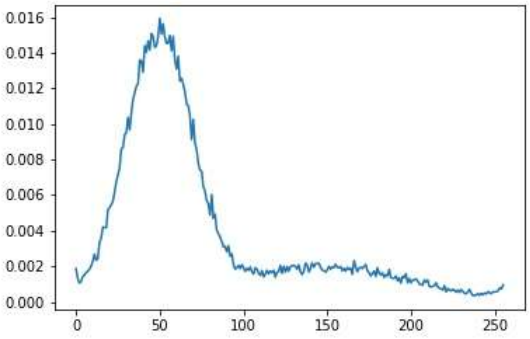


(b) *Distribución de Probabilidad de I_c*

Figura 2.3: Imagen $\boldsymbol{I_c}$ con $\sigma_0 = 10$ y $\sigma_1 = 30$ (izquierda) y su distribución de probabilidad (derecha)



(a) *Imagen I_d con $\sigma_0 = 20$ y $\sigma_1 = 50$*



(b) *Distribución de Probabilidad de I_d*

Figura 2.4: Imagen $\boldsymbol{I_d}$ con $\sigma_0 = 20$ y $\sigma_1 = 50$ (izquierda) y su distribución de probabilidad (derecha)

Al analizar la distribución de probabilidad de $\boldsymbol{I_a}$ en la Figura 2.1(b) podemos ver que las dos curvas que representan a cada una de las clases están completamente separadas. Para la distribución de probabilidad de $\boldsymbol{I_b}$ en la Figura 2.2(b) se puede apreciar que las 2 clases están completamente mezcladas. Cabe mencionar que para $\boldsymbol{I_a}$ e $\boldsymbol{I_b}$ sus desviaciones estándar son iguales para cada clase, es decir que $\sigma_0 = \sigma_1$. En la distribución de probabilidad de $\boldsymbol{I_c}$ en la Figura 2.3(b) se observa que las dos curvas se encuentran una al lado de otra y

se pueden distinguir cada una de las clases. Para la distribución de probabilidad de $\boldsymbol{I_d}$ en la Figura 2.4 se puede ver que las 2 curvas inciden y se mezclan un poco. Además, para $\boldsymbol{I_c}$ e $\boldsymbol{I_d}$ sus desviaciones estándar son diferentes para cada clase, es decir que $\sigma_0 \neq \sigma_1$.

Para comparar el desempeño de los algoritmos anteriormente expuestos, hacemos uso de la medida de exactitud. La exactitud es una medida estadística de que tan bien una prueba de clasificación binaria identifica correctamente. Es decir, la exactitud es la proporción de resultados verdaderos entre el número de casos examinados [Metz78]. Entonces, la exactitud se calcula como (2.29) donde:

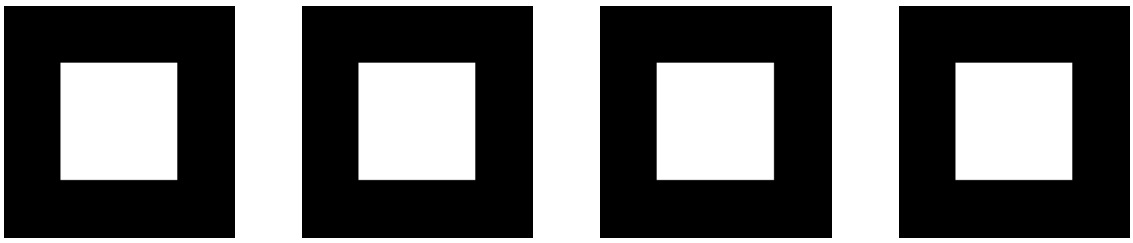
- VN (Verdaderos Negativos): es la cantidad de datos originalmente Falsos que en la predicción resultaron Falsos.
- VP (Verdaderos Positivos): es la cantidad de datos originalmente Verdaderos que en la predicción resultaron Verdaderos.
- FP (Falsos Positivos): es la cantidad de datos originalmente Falsos que en la predicción resultaron Verdaderos.
- FN (Falsos Negativos): es la cantidad de datos originalmente Verdadero que en la predicción resultaron Falsos.

$$Exactitud = \frac{VN + VP}{VP + VN + FP + FN} \times 100\,\% \tag{2.29}$$

Una exactitud del 100 % significa que los valores calculados son iguales a los valores verdaderos.

En la Figura 2.5(a) se muestra la imagen de la Figura 2.1(a) binarizada con el método de Otsu con un umbral $u = 101$ y con una exactitud de 100 %. En la Figura 2.5(b) se muestra la imagen de la Figura 2.1(a) binarizada con Entropía Cruzada con un umbral $u = 90$ y con una exactitud de 100 %. En la Figura 2.5(c) se muestra la imagen de la Figura 2.1(a) binarizada con K-Medias con una exactitud de 100 %. Finalmente, en la Figura 2.5(d) se muestra la imagen de la Figura 2.1(a) binarizada con Mezcla de Gaussianas con una exactitud de 100 %. Para todas estas imágenes binarizadas, la clase 0 tiene una media

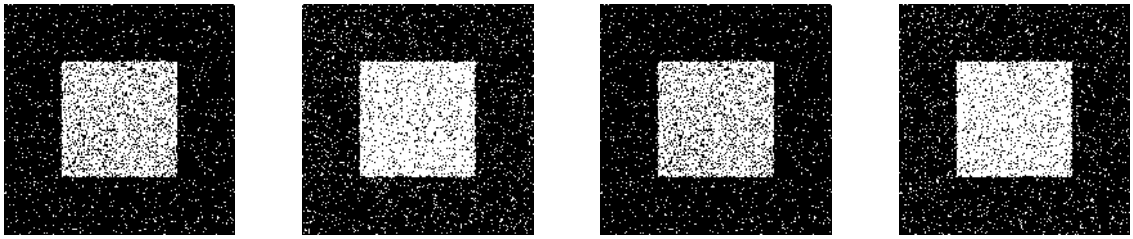
$\mu_0 = 49.492565$ y una desviación estándar $\sigma_0 = 10.023313$ y la clase 1 tiene una media $\mu_1 = 149.520192$ y una desviación estándar $\sigma_1 = 10.023504$. Estos resultados se muestran en la Tabla 2.2 junto con el tiempo de ejecución.



(a) ***Imagen binarizada con Otsu*** **(b)** ***Imagen binarizada con Entropía Cruzada*** **(c)** ***Imagen binarizada con K-Medias*** **(d)** ***Imagen binarizada con Mezcla de Gaussianas***

Figura 2.5: Imágenes binarizadas utilizando la imagen de la Figura 2.1(a) con $\sigma_0 = \sigma_1 = 10$

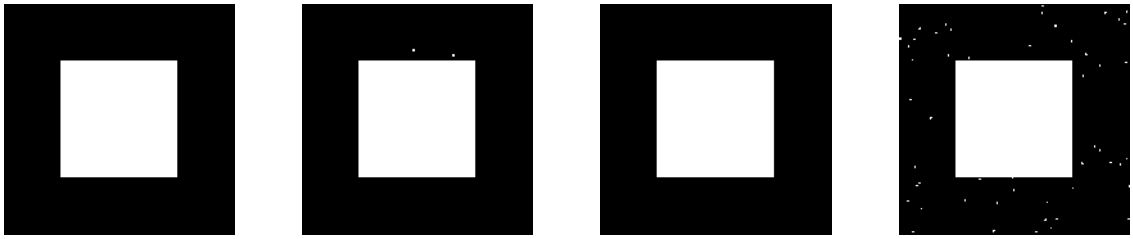
En la Figura 2.6(a) se muestra la imagen de la Figura 2.2(a) binarizada con el método de Otsu con un umbral $u = 117$ y con una exactitud de 83.9072 %, donde la clase 0 tiene una media $\mu_0 = 56.0367835$ y una desviación estándar $\sigma_0 = 30.500425$ y la clase 1 tiene una media $\mu_1 = 177.307852$ y una desviación estándar $\sigma_1 = 43.023291$. En la Figura 2.6(b) se muestra la imagen de la Figura 2.2(a) binarizada con Entropía Cruzada con un umbral $u = 94$ y con una exactitud de 80.1952 %, donde la clase 0 tiene una media $\mu_0 = 48.278506$ y una desviación estándar $\sigma_0 = 25.267710$ y la clase 1 tiene una media $\mu_1 = 160.463859$ y una desviación estándar $\sigma_1 = 49.140442$. En la Figura 2.6(c) se muestra la imagen de la Figura 2.2(a) binarizada con K-Medias con una exactitud de 83.9072 %, donde la clase 0 tiene una media $\mu_0 = 56.036783$ y una desviación estándar $\sigma_0 = 30.500425$ y la clase 1 tiene una media $\mu_1 = 177.307852$ y una desviación estándar $\sigma_1 = 43.023291$. Finalmente, en la Figura 2.6(d) se muestra la imagen de la Figura 2.2(a) binarizada con Mezcla de Gaussianas con una exactitud de 81.7232 %, donde la clase 0 tiene una media $\mu_0 = 50.073862$ y una desviación estándar $\sigma_0 = 26.379738$ y la clase 1 tiene una media $\mu_1 = 163.874293$ y una desviación estándar $\sigma_1 = 47.592655$. Estos resultados se muestran en la Tabla 2.3 junto con el tiempo de ejecución.



(a) ***Imagen binarizada con Otsu*** **(b)** ***Imagen binarizada con Entropía Cruzada*** **(c)** ***Imagen binarizada con K-Medias*** **(d)** ***Imagen binarizada con Mezcla de Gaussianas***

Figura 2.6: Imágenes binarizadas utilizando la imagen de la Figura 2.2(a) con $\sigma_0 = \sigma_1 = 40$

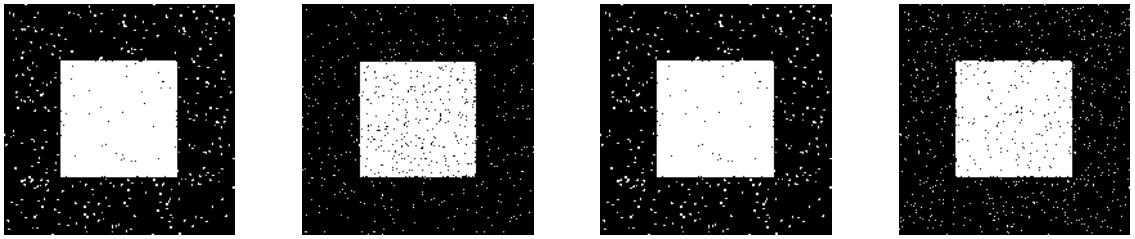
En la Figura 2.7(a) se muestra la imagen de la Figura 2.3(a) binarizada con el método de Otsu con un umbral $u = 102$ y con una exactitud de 98.6432 %, donde la clase 0 tiene una media $\mu_0 = 50.140288$ y una desviación estándar $\sigma_0 = 11.269089$ y la clase 1 tiene una media $\mu_1 = 152.904310$ y una desviación estándar $\sigma_1 = 26.692746$. En la Figura 2.7(b) se muestra la imagen de la Figura 2.3(a) binarizada con Entropía Cruzada con un umbral $u = 91$ y con una exactitud de 99.4096 %, donde la clase 0 tiene una media $\mu_0 = 49.782684$ y una desviación estándar $\sigma_0 = 10.336654$ y la clase 1 tiene una media $\mu_1 = 151.486697$ y una desviación estándar $\sigma_1 = 28.073901$. En la Figura 2.7(c) se muestra la imagen de la Figura 2.3(a) binarizada con K-Medias con una exactitud de 98.6432 %, donde la clase 0 tiene una media $\mu_0 = 50.140288$ y una desviación estándar $\sigma_0 = 11.269089$ y la clase 1 tiene una media $\mu_1 = 152.904310$ y una desviación estándar $\sigma_1 = 26.692746$. Finalmente, en la Figura 2.7(d) se muestra la imagen de la Figura 2.3(a) binarizada con Mezcla de Gaussianas con una exactitud de 99.632 %, donde la clase 0 tiene una media $\mu_0 = 49.488479$ y una desviación estándar $\sigma_0 = 10.010460$ y la clase 1 tiene una media $\mu_1 = 150.106794$ y una desviación estándar $\sigma_1 = 29.012795$. Estos resultados se muestran en la Tabla 2.4 junto con el tiempo de ejecución.



(a) ***Imagen binarizada con Otsu*** (b) ***Imagen binarizada con Entropía Cruzada*** (c) ***Imagen binarizada con K-Medias*** (d) ***Imagen binarizada con Mezcla de Gaussianas***

Figura 2.7: Imágenes binarizadas utilizando la imagen de la Figura 2.3(a) con $\sigma_0 = 10$ y $\sigma_1 = 30$

En la Figura 2.8(a) se muestra la imagen de la Figura 2.4(a) binarizada con el método de Otsu con un umbral $u = 110$ y con una exactitud de 93.72 %, donde la clase 0 tiene una media $\mu_0 = 51.648924$ y una desviación estándar $\sigma_0 = 21.401276$ y la clase 1 tiene una media $\mu_1 = 167.380232$ y una desviación estándar $\sigma_1 = 36.754498$. En la Figura 2.8(b) se muestra la imagen de la Figura 2.4(a) binarizada con Entropía Cruzada con un umbral $u = 94$ y con una exactitud de 94.688 %, donde la clase 0 tiene una media $\mu_0 = 49.747980$ y una desviación estándar $\sigma_0 = 19.347771$ y la clase 1 tiene una media $\mu_1 = 158.414406$ y una desviación estándar $\sigma_1 = 40.479085$. En la Figura 2.8(c) se muestra la imagen de la Figura 2.4(a) binarizada con K-Medias con una exactitud de 93.72 %, donde la clase 0 tiene una media $\mu_0 = 51.648924$ y una desviación estándar $\sigma_0 = 21.401276$ y la clase 1 tiene una media $\mu_1 = 167.380232$ y una desviación estándar $\sigma_1 = 36.754498$. Finalmente, en la Figura 2.8(d) se muestra la imagen de la Figura 2.4(a) binarizada con Mezcla de Gaussianas con una exactitud de 94.8736 %, donde la clase 0 tiene una media $\mu_0 = 49.962523$ y una desviación estándar $\sigma_0 = 19.607454$ y la clase 1 tiene una media $\mu_1 = 159.867395$ y una desviación estándar $\sigma_1 = 40.364210$. Estos resultados se muestran en la Tabla 2.5 junto con el tiempo de ejecución.



(a) ***Imagen binarizada con Otsu*** **(b)** ***Imagen binarizada con Entropía Cruzada*** **(c)** ***Imagen binarizada con K-Medias*** **(d)** ***Imagen binarizada con Mezcla de Gaussianas***

Figura 2.8: Imágenes binarizadas utilizando la imagen de la Figura 2.4(a) con $\sigma_0 = 20$ y $\sigma_1 = 50$

Algoritmo	u	μ_0	σ_0	μ_1	σ_1	Exactitud	Tiempo(segs)
Otsu	101	49.492565	10.023313	149.520192	10.023504	100 %	0.1471
Entropía	90	49.492565	10.023313	149.520192	10.023504	100 %	0.0179
K-Medias		49.492565	10.023313	149.520192	10.023504	100 %	0.1783
GMM		49.492565	10.023313	149.520192	10.023504	100 %	0.1627

Tabla 2.2: Resultados de Binarización de la Imagen de la Figura 2.1 con $\sigma_0 = \sigma_1 = 10$

Algoritmo	u	μ_0	σ_0	μ_1	σ_1	Exactitud	Tiempo(segs)
Otsu	117	56.0367835	30.500425	177.307852	43.023291	83.9072 %	0.1848
Entropía	94	48.278506	25.267710	160.463859	49.140442	80.1952 %	0.0149
K-Medias		56.0367835	30.500425	177.307852	43.023291	83.9072 %	0.3385
GMM		50.073862	26.37973	163.874293	47.592655	81.7232 %	2.5902

Tabla 2.3: Resultados de Binarización de la Imagen de la Figura 2.2 con $\sigma_0 = \sigma_1 = 40$

Algoritmo	u	μ_0	σ_0	μ_1	σ_1	Exactitud	Tiempo(segs)
Otsu	102	50.140288	11.269089	152.904310	26.692746	98.6432 %	0.1531
Entropía	91	49.782684	10.336654	151.486697	28.073901	99.4096 %	0.0169
K-Medias		50.140288	11.269089	152.904310	26.692746	98.6432 %	0.1848
GMM		49.488479	10.010460	150.106794	29.012795	99.6320 %	0.3514

Tabla 2.4: Resultados de Binarización de la Imagen de la Figura 2.3 con $\sigma_0 = 10$ y $\sigma_1 = 30$

Algoritmo	u	μ_0	σ_0	μ_1	σ_1	Exactitud	Tiempo(segs)
Otsu	110	51.648924	21.401276	167.380232	36.754498	93.7200 %	0.1848
Entropía	94	49.747980	19.347771	158.414406	40.479085	94.688 %	0.0156
K-Medias		51.648924	21.401276	167.380232	36.754498	93.7200 %	0.2695
GMM		49.962523	19.607454	159.867395	40.364210	94.8736 %	1.0339

Tabla 2.5: Resultados de Binarización de la Imagen de la Figura 2.4 con $\sigma_0 = 20$ y $\sigma_1 = 50$

Al observar los resultados de la binarización de las imágenes sintéticas con K-Medias, Mezcla de Gaussianas y mediante el umbral obtenido con el Método de Otsu y con la Entropía Cruzada podemos ver que los resultados de Otsu y K-Medias son idénticos, excepto por su tiempo de ejecución. La binarización de Umbralización por Entropía Cruzada es el algoritmo mas rápido. Por último, la Mezcla de Gaussianas tiene un mejor desempeño comparado con los otros algoritmos en la mayoría de las pruebas, ya que en general se acerca más a los valores verdaderos de las medias y las desviaciones estándar de cada clase y su exactitud es mayor que los otros algoritmos. Sin embargo, en la Tabla 2.3 se puede ver que el modelo de Mezcla de Gaussianas es el que tiene peor desempeño. Esto es porque las dos clases están demasiado mezcladas. En el histograma de la Figura 2.2(b) no se puede observar claramente la separación de las dos distribuciones Gaussianas de la mezcla. El Método de Otsu calcula el umbral u que maximice la varianza interclase, la cual sólo considera los pesos y las medias de cada clase. El algoritmo de Umbralización por Entropía Cruzada calcula umbral que minimice la diferencia de la información entre los dos segmentos de la imagen. El algoritmo de K-Medias agrupa los datos de acuerdo a la distancia de cada dato hacia la media de cada grupo. Sin embargo, el modelo de Mezcla de Gaussianas considera las medias y también considera las varianzas de cada clase. Para este trabajo, suponemos que las dos clases no necesariamente deben de tener la misma desviación estándar. Por lo tanto, podemos concluir que el modelo de Mezcla de Gaussianas es el mejor candidato para realizar la binarización de las imágenes a fusionar ya que no solo considera la media sino también la desviación estándar de cada clase. Además, podemos suponer que K-Medias y la binarización con el umbral del Método de Otsu son idénticos ya que ambos consideran la distancia hacia las medias de cada clase. Por lo tanto, solo se tomará en cuanta el Método de Otsu y se descartará K-Medias porque el Método de Otsu cuesta menos tiempo.

Al analizar el desempeño de los algoritmos de binarización, podemos notar que existe poco mas de un 5 % de error al binarizar. Sin embargo, en la Sección 1.2 se mencionó un algoritmo capaz de mejorar el desempeño la binarización de imágenes llamado algoritmo de Ventanas Óptimas. Este algoritmo ayuda a aumentar la exactitud de la binarización al introducir coherencia espacial.

2.4. Algoritmo de Ventana Óptima

El trabajo de tesis de Cristina Reyes [Reyes21] describe el algoritmo de Ventana Óptima (VO). Este proceso se basa en la Combinación Lineal de Imágenes Simple (CLI-S) [Calderón16] para lograr obtener imágenes binarias mucho más exactas. Este algoritmo considera la coherencia espacial en una imagen. Es decir, la condición de binarización del algoritmo considera los píxeles dentro de una vecindad limitada por una ventana. Sin embargo, el algoritmo CLI-S necesita un valor para el tamaño de ventana, el cual para el algoritmo es constante. El algoritmo VO calcula un tamaño de ventana para cada píxel de la imagen de acuerdo a una tolerancia T.

Empecemos definiendo la función de calidad $\boldsymbol{F_0}[r,c]$ (2.30) como la distancia de cada píxel $\boldsymbol{I}[r,c]$ a la media μ_0 de la clase 0 y la función de calidad $\boldsymbol{F_1}[r,c]$ (2.31) como la distancia de cada píxel $\boldsymbol{I}[r,c]$ a la media μ_1 de la clase 1.

$$\boldsymbol{F_0}[r,c] = |\boldsymbol{I}[r,c] - \mu_0| \tag{2.30}$$

$$\boldsymbol{F_1}[r,c] = |\boldsymbol{I}[r,c] - \mu_1| \tag{2.31}$$

Se genera una matriz binaria denominada mapa de decisión $\boldsymbol{P}$ en donde $\boldsymbol{P}[r,c] = 0$ si la distancia entre $\boldsymbol{I}[r,c]$ y μ_0 es menor que la distancia entre $\boldsymbol{I}[r,c]$ y μ_1, de otra forma $\boldsymbol{P}[r,c] = 1$, es decir:

$$\boldsymbol{P}[r,c] = \begin{cases} 0 & \text{Si } \boldsymbol{F_0}[r,c] < \boldsymbol{F_1}[r,c] \\ 1 & \text{Si no} \end{cases} \tag{2.32}$$

Para introducir coherencia espacial, calculamos la suma del vecindario de $\boldsymbol{F_0}[r,c]$ y $\boldsymbol{F_1}[r,c]$ delimitado por una ventana de tamaño $(2w+1)\times(2w+1)$ centrada en las coordenadas $[r,c]$ tal como:

$$\boldsymbol{S_k}[r,c] = \sum_{i=r-w}^{r+w}\sum_{j=c-w}^{c+w} \boldsymbol{F_k}[i,j] \tag{2.33}$$

donde $k = 0, 1$.

Entonces, ahora el mapa de decisión $\boldsymbol{P}$ lo calculamos mediante:

$$\boldsymbol{P}[r,c] = \begin{cases} 0 & \text{Si } \boldsymbol{S_0}[r,c] < \boldsymbol{S_1}[r,c] \\ 1 & \text{Si no} \end{cases} \tag{2.34}$$

Sin embargo, existe el problema de calcular el valor de w. Este valor será diferente para cada elemento $\boldsymbol{S_k}[r,c]$ y depende del vecindario de $\boldsymbol{P}[r,c]$. Para calcular el valor adecuado de w para cada coordenada $[r,c]$ sobre el mapa de decisión $\boldsymbol{P}$ comenzamos definiendo un mapa de bordes $\boldsymbol{B}$. El mapa de bordes $\boldsymbol{B}$ se define por (2.35).

$$\boldsymbol{B}[r,c] = \begin{cases} 1 & \text{Si } \boldsymbol{P}[r,c] \neq \boldsymbol{P}[r-w,c] \text{ ó } \boldsymbol{P}[r,c] \neq \boldsymbol{P}[r,c-1] \text{ ó } \boldsymbol{P}[r,c] \neq \boldsymbol{P}[r-1,c-1] \\ 0 & \text{Si no} \end{cases} \tag{2.35}$$

Definimos una matriz $\boldsymbol{W}$ la cual contiene en $\boldsymbol{W}[r,c]$ el tamaño de ventana adecuado w para las coordenadas $[r,c]$. Para calcular $\boldsymbol{W}[r,c]$ es necesario realizar una búsqueda del valor de w que maximice el área de la ventana. El área de la ventana se define como $A(w) = (2w+1)\times(2w+1)$ donde $0 \leq w \leq w_{max}$. La búsqueda esta sujeta a la restricción de que no sobrepase una determinada cantidad de bordes T a la cual denominaremos como tolerancia. Hacemos la suposición de que al aumentar la ventana w, aumenta la cantidad de bordes. La función a optimizar se define como:

$$\begin{gathered} \boldsymbol{W}[r,c] = \underset{w}{argmax} A(w) \\ \text{sujeto a} \\ 0 \leq w \leq w_{max} \\ \sum_{i=r-w}^{r+w} \sum_{j=c-w}^{c+w} \boldsymbol{B}[i,j] < T \end{gathered} \tag{2.36}$$

Para hacer eficiente la búsqueda del valor de w se utiliza la búsqueda binaria y para disminuir el costo computacional del cálculo de la suma de píxeles se emplean imágenes integrales [Viola01].

2.4.1. Imágenes Integrales

El uso de imágenes integrales [Viola01] permite generar rápida y eficientemente la suma de valores en un subconjunto rectangular de una imagen. Una imagen integral se define mediante (2.37).

$$\boldsymbol{H}[r,c] = \sum_{i=0}^{r} \sum_{j=0}^{c} \boldsymbol{I}[i,j] \tag{2.37}$$

El calculo de la imagen integral mediante (2.37) es de orden $O(N^4)$, sin embargo, éste calculo se puede realizar con la recurrencia (2.38) en donde solamente se realizan 3 operaciones para cada coordenada, haciendo el calculo de la imagen integral de orden $O(N^2)$.

$$\boldsymbol{H}[r,c] = \boldsymbol{I}[r,c] + \boldsymbol{H}[r,c-1] + \boldsymbol{H}[r-1,c] - \boldsymbol{H}[r-1,c-1] \tag{2.38}$$

La suma de los valores dentro de una ventana de tamaño $(2w+1) \times (2w+1)$ centrada en las coordenadas $[r,c]$ es igual a realizar la suma y resta de cuatro elementos en la imagen integral $\boldsymbol{H}$, como se muestra en (2.39). El calculo de una matriz $\boldsymbol{S}$ sin el uso de imágenes integrales es de orden $O(N^4)$. Sin embargo, al utilizar las imágenes integrales el calculo es de orden $O(N^2)$.

$$\boldsymbol{S}[r,c] = \sum_{i=r-w}^{r+w} \sum_{j=c-w}^{c+w} \boldsymbol{I}[i,j] = \boldsymbol{H}[r+w, c+w] + \boldsymbol{H}[r-w-1, c-w-1] - \boldsymbol{H}[r-w-1, c+w] - \boldsymbol{H}[r+w, c-w-1] \quad (2.39)$$

Ahora sólo falta establecer el tamaño óptimo de ventana w. Para esto se hace uso de un algoritmo de búsqueda llamado búsqueda binaria. Se busca un tamaño de ventana w lo más grande posible que cumpla con las restricciones de la Ecuación (2.36). Suponemos que al crecer w, la suma sobre la matriz de bordes $\boldsymbol{B}$ dentro de la ventana con tamaño $(2w+1) \times (2w+1)$ también crece o aumenta. Sin embargo, se tiene la restricción de que esta suma no supere la tolerancia T y el tamaño máximo de ventana w_{max}.

2.4.2. Búsqueda Binaria

La búsqueda binaria es un algoritmo que encuentra la posición de un valor dentro de un arreglo ordenado. La búsqueda binaria compara el valor objetivo con el elemento a la mitad del arreglo. Si el valor objetivo es igual a ese elemento, el algoritmo regresa la posición del elemento. Si el valor objetivo es menor que el elemento, el algoritmo continua buscando en la mitad inferior del arreglo. Si el valor objetivo es mayor al elemento, el algoritmo sigue buscando en la mitad superior del arreglo. En cada iteración, el algoritmo elimina la mitad de elementos en los que no se encuentra el valor objetivo.

Para la búsqueda del tamaño de ventana w, el algoritmo de búsqueda binaria maximiza la función (2.36) realizando una búsqueda en el arreglo ordenado en el rango $[0, w_{max}]$. La condición para el segmento en el que se realiza la búsqueda debe maximizar la suma del vecindario $\boldsymbol{S}[r,c]$ con la restricción de que no supere cierta tolerancia T. Suponemos que al aumentar el tamaño de ventana w, la suma del vecindario aumenta también dado que el área del vecindario es mayor.

El algoritmo de búsqueda binaria para el tamaño de ventana w se ilustra en el Algoritmo 4. El algoritmo recibe las coordenadas $[r,c]$ correspondientes al centro de la ventana, el tamaño máximo de ventana w_{max}, la imagen integral $\boldsymbol{H_B}$ de la imagen de bordes $\boldsymbol{B}$ y la tolerancia de bordes T. El algoritmo entrega el valor óptimo de tamaño de

ventana w_m.

Algoritmo 4: Busqueda_Binaria(Entrada: r, c,w_{max},$\boldsymbol{H_B}$, T; Salida: w_m)

$w_i \leftarrow 1$
$w_f \leftarrow w_{max}$
mientras $w_f - w_i > 1$ **hacer**
  $w_m \leftarrow \lfloor \frac{w_i + w_f}{2} \rfloor$
  Calcular $\boldsymbol{S}[r,c]$ empleando $\boldsymbol{H_B}$ y w_m con (2.39)
  si $\boldsymbol{S}[r,c] < T$ **entonces**
    $w_i \leftarrow w_m$
  en otro caso
    $w_f \leftarrow w_m$
  fin
fin
devolver w_m

El algoritmo VO se describe en el Algoritmo 5 , el cual recibe el mapa de decisión $\boldsymbol{P}$, el número máximo de bordes o tolerancia T y el tamaño máximo de ventana w_{max}. El algoritmo VO regresa una matriz de ventanas óptimas $\boldsymbol{W}$.

Algoritmo 5: VO(Entrada: $\boldsymbol{P}$,T,w_{max}; Salida: $\boldsymbol{W}$)

Calcular $\boldsymbol{B}[r,c]$ con (2.35) $\forall[r,c]$
Calcular $\boldsymbol{H_B}[r,c]$ de $\boldsymbol{B}[r,c]$ con (2.37) $\forall[r,c]$
para $r = 0 \quad \boldsymbol{a} \quad N_r - 1$ **hacer**
  para $c = 0 \quad \boldsymbol{a} \quad N_c - 1$ **hacer**
    $\boldsymbol{W}[r,c] \leftarrow$ Busqueda_Binaria(r,c,w_{max},$\boldsymbol{H_B}$,T)
  fin
fin
devolver $\boldsymbol{W}$

Después de haber obtenido el tamaño de las ventanas óptimas para cada píxel, podemos calcular el mapa de decisión $\boldsymbol{P}$ utilizando las imágenes integrales para el calculo de $\boldsymbol{S_k}[r,c]$ con (2.40) para $k = 0, 1$.

$$\boldsymbol{P}[r,c] = \begin{cases} 0 & \text{Si } \boldsymbol{S_0}[r,c] < \boldsymbol{S_1}[r,c] \\ 1 & \text{Si no} \end{cases} \tag{2.40}$$

2.5. Resultados al Aplicar el Algoritmo de Ventana Óptima

El algoritmo VO requiere de un mapa de decisión inicial P, para esto utilizaremos las imágenes obtenidas de la binarización utilizando el modelo de Mezcla de Gaussianas y de umbralización por el método de Otsu y el de Entropía Cruzada con peores resultados como mapas de decisión inicial. Es decir, utilizaremos las imágenes binarizadas donde $\sigma_0 = \sigma_1 = 40$ y $\sigma_0 = 20$ y $\sigma_1 = 50$. Para el calculo de las funciones de calidad $\boldsymbol{F_0}$ y $\boldsymbol{F_1}$ utilizaremos las medias en lugar de las modas, como se muestra en (2.41).

$$\boldsymbol{F_k}[r,c] = |\boldsymbol{I}[r,c] - \mu_k| \qquad \text{para } k = 0, 1 \tag{2.41}$$

El tamaño máximo de ventana w_{max} lo consideraremos de manera que la ventana máxima cubra todo el lado más pequeño de la imagen. Para nuestro caso, todos los lados son iguales de tamaño por lo que el área de la ventana es:

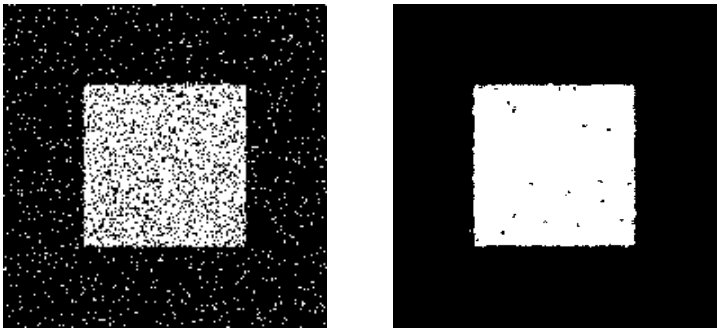
$$A(w) = (2w_{max} + 1) \times (2w_{max} + 1) = N_r \times N_c \tag{2.42}$$

Como $N_r = N_c = 250$, tenemos que:

$$\begin{aligned} (2w_{max} + 1) \times (2w_{max} + 1) &= 250 \times 250 \\ (2w_{max} + 1)^2 &= 250^2 \\ 2w_{max} + 1 &= 250 \\ 2w_{max} &= 249 \\ w_{max} &\approx 124 \end{aligned}$$

Por último, el valor de tolerancia T lo estableceremos igual a 10. Este valor de T se obtuvo de manera empírica al realizar varias pruebas durante el proceso de la codificación del algoritmo de Fusión de Imágenes con Iluminación Complementaria con distintas imágenes.

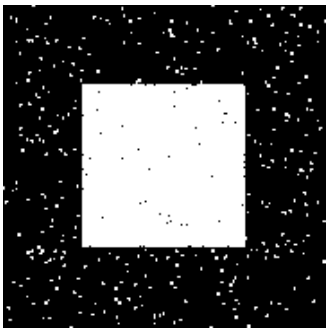
El mapa de decisión inicial obtenido por umbralización con el Método de Otsu para las desviaciones estándar originales de $\sigma_0 = \sigma_1 = 40$ se muestra nuevamente en la Figura 2.9(a) y el mapa de decisión $\boldsymbol{P}$ afinado con el VO se muestra la Figura 2.9(b). Con este mapa de decisión $\boldsymbol{P}$ se obtiene una exactitud de 99.3968 %, donde la clase 0 tiene una media $\mu_0 = 75.976229$ y una desviación estándar $\sigma_0 = 63.193860$ y la clase 1 tiene una media $\mu_1 = 149.197346$ y una desviación estándar $\sigma_1 = 40.1772$. Los resultados de la binarización con Otsu y su afinación con VO se muestran en la Tabla 2.6 junto con el tiempo de ejecución.



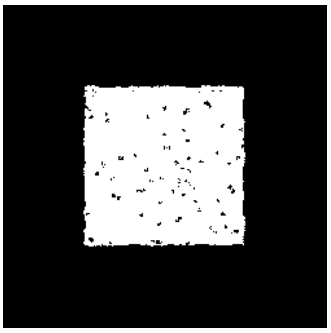
(a) ***Mapa de Decisión Inicial*** **(b)** ***Mapa de Decisión Afinado con VO***

Figura 2.9: Mapas de Decisión Obtenidas con la Binarización Utilizando Otsu y Otsu con VO para $\sigma_0 = \sigma_1 = 40$

El mapa de decisión inicial obtenido por umbralización con el Método de Otsu para las desviaciones estándar originales de $\sigma_0 = 20$ y $\sigma_1 = 50$ se muestra nuevamente en la Figura 2.10(a) y el mapa de decisión $\boldsymbol{P}$ afinado con el VO se muestra la Figura 2.10(b). Con el mapa de decisión $\boldsymbol{P}$ se obtiene una exactitud de 98.496 %, donde la clase 0 tiene una media $\mu_0 = 52.480016$ y una desviación estándar $\sigma_0 = 27.840888$ y la clase 1 tiene una media $\mu_1 = 146.71842$ y una desviación estándar $\sigma_1 = 48.977745$. Los resultados de la binarización con Otsu y su afinación con VO se muestran en la Tabla 2.7 junto con el tiempo de ejecución.



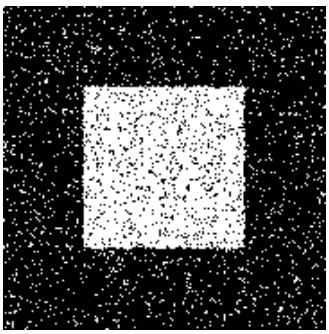
(a) ***Mapa de Decisión Inicial***



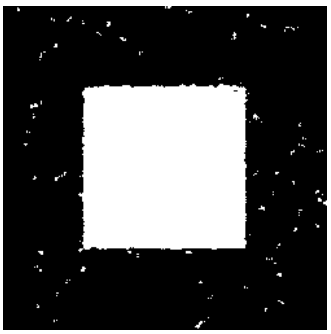
(b) ***Mapa de Decisión Afinado con VO***

Figura 2.10: Mapas de Decisión Obtenidas con la Binarización por Otsu y Otsu con VO para $\sigma_0 = 20$ y $\sigma_1 = 50$

El mapa de decisión inicial obtenido por umbralización por Entupía Cruzada Mínima para las desviaciones estándar originales de $\sigma_0 = \sigma_1 = 40$ se muestra nuevamente en la Figura 2.11(a) y el mapa de decisión $\boldsymbol{P}$ afinado con el VO se muestra la Figura 2.11(b). Con este mapa de decisión $\boldsymbol{P}$ se obtiene una exactitud de 98.8768 %, donde la clase 0 tiene una media $\mu_0 = 75.396284$ y una desviación estándar $\sigma_0 = 63.060106$ y la clase 1 tiene una media $\mu_1 = 147.053933$ y una desviación estándar $\sigma_1 = 42.706221$. Los resultados de la binarización con Entropía Cruzada y su afinación con VO se muestran en la Tabla 2.6 junto con el tiempo de ejecución.



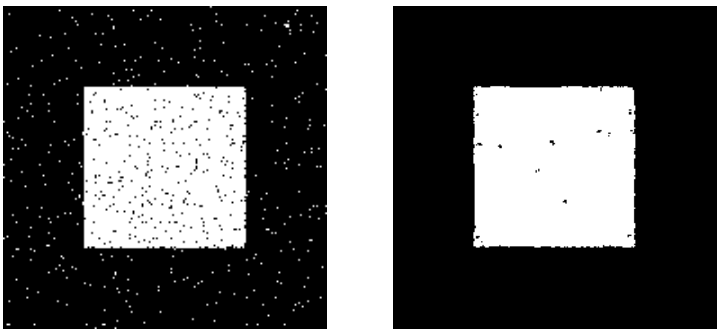
(a) ***Mapa de Decisión Inicial***



(b) ***Mapa de Decisión Afinado con VO***

Figura 2.11: Mapas de Decisión Obtenidas con la Binarización Utilizando Entropía Cruzada y Entropía Cruzada con VO para $\sigma_0 = \sigma_1 = 40$

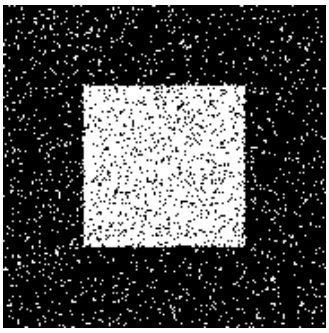
El mapa de decisión inicial obtenido por umbralización por Entupía Cruzada Mínima para las desviaciones estándar originales de $\sigma_0 = 20$ y $\sigma_1 = 50$ se muestra nuevamente en la Figura 2.12(a) y el mapa de decisión $\boldsymbol{P}$ afinado con el VO se muestra la Figura 2.12(b). Con este mapa de decisión $\boldsymbol{P}$ se obtiene una exactitud de 98.3568 %, donde la clase 0 tiene una media $\mu_0 = 51.655054$ y una desviación estándar $\sigma_0 = 25.830118$ y la clase 1 tiene una media $\mu_1 = 145.852197$ y una desviación estándar $\sigma_1 = 49.797313$. Los resultados de la binarización con Entropía Cruzada y su afinación con VO se muestran en la Tabla 2.7 junto con el tiempo de ejecución.



(a) ***Mapa de Decisión Inicial*** **(b)** ***Mapa de Decisión Afinado con VO***

Figura 2.12: Mapas de Decisión Obtenidas con la Binarización Utilizando Entropía Cruzada y Entropía Cruzada con VO para $\sigma_0 = 20$ y $\sigma_1 = 50$

El mapa de decisión inicial obtenido de la binarización con GMM para las desviaciones estándar originales de $\sigma_0 = \sigma_1 = 40$ se muestra nuevamente en la Figura 2.13(a) y el mapa de decisión $\boldsymbol{P}$ afinado con el VO se muestra la Figura 2.13(b). Con el mapa de decisión $\boldsymbol{P}$ se obtiene una exactitud de 99.5536 %, donde la clase 0 tiene una media $\mu_0 = 75.976414$ y una desviación estándar $\sigma_0 = 62.894042$ y la clase 1 tiene una media $\mu_1 = 148.585697$ y una desviación estándar $\sigma_1 = 40.182613$. Los resultados de la binarización con GMM y su afinación con VO se muestran en la Tabla 2.6 junto con el tiempo de ejecución.

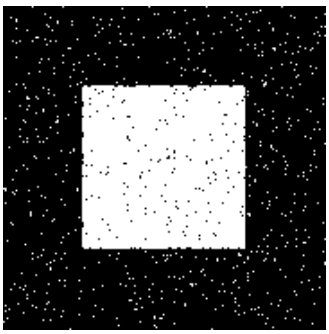


(a) ***Mapa de Decisión Inicial***

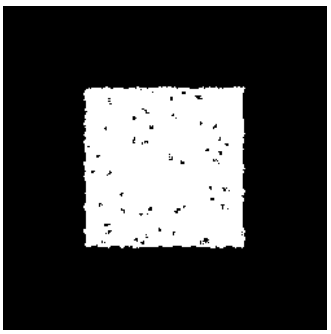
(b) ***Mapa de Decisión Afinado con VO***

Figura 2.13: Mapas de Decisión Obtenidas con la Binarización de Mezcla de Gaussianas y Mezcla de Gaussianas con VO para $\sigma_0 = \sigma_1 = 40$

El mapa de decisión inicial obtenido mediante la binarización con GMM para las desviaciones estándar originales de $\sigma_0 = 20$ y $\sigma_1 = 50$ se muestra nuevamente en la Figura 2.14(a) y el mapa de decisión $\boldsymbol{P}$ mejorado mediante la afinación con VO se muestra la Figura 2.14(b). Con el mapa de decisión $\boldsymbol{P}$ se obtiene una exactitud de 99.5264 %, donde la clase 0 tiene una media $\mu_0 = 52.522386$ y una desviación estándar $\sigma_0 = 28.074650$ y la clase 1 tiene una media $\mu_1 = 145.897306$ y una desviación estándar $\sigma_1 = 49.390332$. Los resultados de la binarización con GMM y su afinación con VO se muestran en la Tabla 2.7 junto con el tiempo de ejecución.



(a) ***Mapa de Decisión Inicial***



(b) ***Mapa de Decisión Afinado con VO***

Figura 2.14: Mapas de Decisión Obtenidas con la Binarización de Mezcla de Gaussianas y Mezcla de Gaussianas con VO para $\sigma_0 = 20$ y $\sigma_1 = 50$

Algoritmo	μ_0	σ_0	μ_1	σ_1	Exactitud	Tiempo(segs)
Otsu	56.0367835	30.500425	177.307852	43.023291	83.9072 %	0.1848
Otsu+VO	75.976229	63.19386	149.197346	40.1772	99.3968 %	0.7032
Entropía	48.278506	25.267710	160.463859	49.140442	80.1952 %	0.0149
Entropía+VO	75.396284	63.060106	147.053933	42.706221	98.8768 %	0.5547
GMM	50.073862	26.37973	163.874293	47.592655	81.7232 %	2.5902
GMM+VO	75.976414	62.894042	148.585697	40.182613	99.5536 %	3.2310

Tabla 2.6: Resultados de Binarización de la Imagen Sintética con $\sigma_0 = \sigma_1 = 40$

Algoritmo	μ_0	σ_0	μ_1	σ_1	Exactitud	Tiempo(segs)
Otsu	51.648924	21.401276	167.380232	36.754498	93.7200 %	0.1848
Otsu+VO	52.480016	27.840888	146.71842	48.977745	98.4960 %	0.7025
Entropía	49.747980	19.347771	158.414406	40.479085	94.688 %	0.0156
Entropía+VO	51.655054	25.830118	145.852197	49.797313	99.3568 %	0.5547
GMM	49.962523	19.607454	159.867395	40.364210	94.8736 %	1.0339
GMM+VO	52.522386	28.074650	145.897306	49.390332	98.5264 %	1.5886

Tabla 2.7: Resultados de Binarización de la Imagen Sintética con $\sigma_0 = 20$ y $\sigma_1 = 50$

Al observar los resultados, podemos darnos cuenta que al afinar los mapas de decisión iniciales la exactitud aumenta considerablemente. Sin embargo, la cercanía a los valores originales de las medias y varianzas no siempre mejoran. Podemos concluir que efectivamente los resultados en cuanto a la exactitud son mejores al afinar con el algoritmo VO.

2.6. Resumen del Capítulo

En este capítulo presentamos algunos algoritmos utilizados para realizar la binarización de imágenes, la cual es un paso importante para realizar la fusión de imágenes. Los algoritmos de Otsu [Otsu79], Entropía Cruzada [Li93] y K-Medias [MacQueen67] son rápidos pero no tienen un buen desempeño en comparación con GMM [Chatzis08] en la mayoría de las pruebas. Otsu y K-Medias toman en cuenta la distancia de cada elemento hacia una media. Por otro lado, GMM considera las medias y también las varianzas de cada clase. El algoritmo de Entropía Cruzada calcula el umbral basado en la cantidad de información

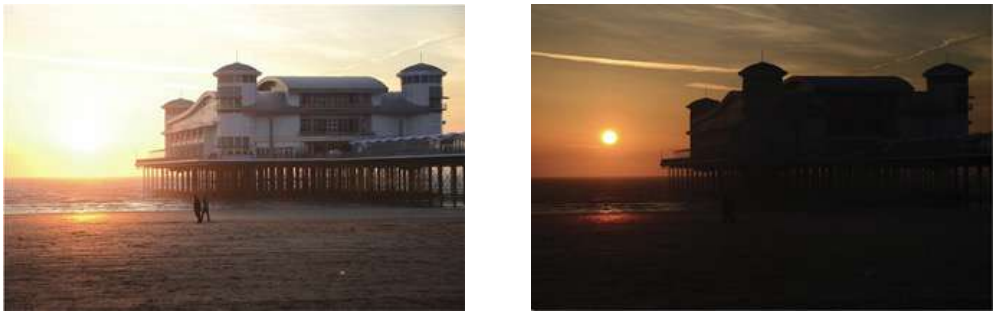
de cada segmento. El algoritmo VO [Reyes21] junto con el algoritmo CLI-S [Calderón16] ayudan a afinar el mapa de decisión para obtener una mayor exactitud.

Capítulo 3

Algoritmo Propuesto

En este Capítulo se describe el algoritmo propuesto para realizar la fusión de imágenes con iluminación complementaria, en otras palabras, la fusión de una imagen con sobreexposición, a la cual llamamos $\boldsymbol{I_1}$, y una imagen con subexposición, a la cual llamamos $\boldsymbol{I_2}$, de una misma escena.

En la Figura 3.1 se muestra un ejemplo de un par de imágenes que a fusionar mediante el algoritmo de fusión de imágenes con iluminación complementaria. En la Figura 3.1(a) se muestra una imagen con algunos píxeles sobreexpuestos $\boldsymbol{I_1}$ y en la Figura 3.1(b) una imagen con algunos píxeles subexpuestos $\boldsymbol{I_2}$. Este par de imágenes se utilizarán para mostrar y describir el algoritmo de fusión en este capítulo.



(a) ***Imagen Sobreexpuesta*** $\boldsymbol{I_1}$ **(b)** ***Imagen Subexpuesta*** $\boldsymbol{I_2}$

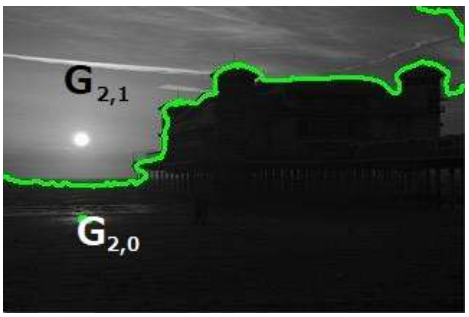
Figura 3.1: Ejemplo de Imágenes a Fusionar

En la Sección 1.1 se mencionó que el problema de fusión de imágenes se aborda utilizando una segmentación de imágenes por medio de una binarización. Con esta segmentación de imágenes se pretende separar los píxeles nítidos de cada una de las imágenes a fusionar.

3.1. Segmentación de Imágenes por Binarización

Con la fusión de imágenes se pretende obtener una imagen que contenga los píxeles nítidos de ambas imágenes a fusionar. Entonces, el primer problema a resolver es segmentar las imágenes $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$ para obtener dos regiones, una región de píxeles nítidos y otra de píxeles no nítidos. Para realizar la fusión, primero transformamos las imágenes a color a imágenes en escala de grises ya que es más fácil trabajar con un solo valor de iluminación por píxel. En la escala de tonos de grises, cada valor representa la intensidad de iluminación en cada sensor de la cámara. Las imágenes $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$ en escala de grises las denominaremos como $\boldsymbol{G_1}$ y $\boldsymbol{G_2}$ respectivamente. En la Figura 3.2(a) se muestra la imagen $\boldsymbol{G_1}$ dividida en 2 regiones: la región de píxeles no nítidos a la que denominaremos como $\boldsymbol{G_{1,0}}$ y la región de píxeles nítidos a la que denominaremos como $\boldsymbol{G_{1,1}}$. De la misma forma, en la Figura 3.2(b) se muestra la imagen $\boldsymbol{G_2}$ dividida en 2 regiones: la región de píxeles no nítidos a la que denominaremos como $\boldsymbol{G_{2,0}}$ y la región de píxeles nítidos a la que denominaremos como $\boldsymbol{G_{2,1}}$.

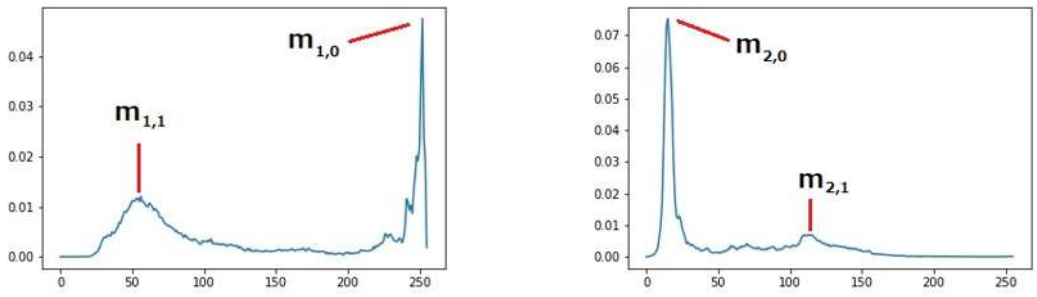
(a) ***Regiones de la Imagen Sobreexpuesta en Escala de Grises***



(b) ***Regiones de la Imagen Subexpuesta en Escala de Grises***

Figura 3.2: Regiones de Píxeles Nítidos y No Nítidos en Imágenes en Escala de Grises

Al analizar las distribuciones de probabilidad de $\boldsymbol{G_1}$ y $\boldsymbol{G_2}$ podemos darnos cuenta que existen dos modas en cada una, las cuales pertenecen a cada región de la imagen. En la Figura 3.3(a) se muestra la distribución de probabilidad de $\boldsymbol{G_1}$ donde se observan 2 modas: la moda $m_{1,0}$ correspondiente a la región $\boldsymbol{G_{1,0}}$ y la moda $m_{1,1}$ correspondiente a la región $\boldsymbol{G_{1,1}}$. De igual forma, en la Figura 3.3(b) se muestra la distribución de probabilidad de $\boldsymbol{G_2}$ donde se observan 2 modas: la moda $m_{2,0}$ correspondiente a la región $\boldsymbol{G_{2,0}}$ y la moda $m_{2,1}$ correspondiente a la región $\boldsymbol{G_{2,1}}$.



(a) ***Distribución de Probabilidad de la Imagen Sobreexpuesta en Escala de Grises***

(b) ***Distribución de Probabilidad de la Imagen Subexpuesta en Escala de Grises***

Figura 3.3: Distribuciones de Probabilidad de las Imágenes en Escala de Grises

Al observar cada una de las distribuciones de probabilidad podemos asumir que es una mezcla de dos Gaussianas. Por lo tanto, la binarización por medio del modelo de Mezcla de Gaussianas parece ser un buen candidato para obtener una imagen binaria para $\boldsymbol{G_1}$ y otra imagen binaria para $\boldsymbol{G_2}$. Al realizar la binarización de las imágenes se obtiene una imagen binaria para cada imagen a fusionar. Estas imágenes binarias, a las cuales denominamos como Mapas de Decisión, con el cual se puede determinar que píxeles son nítidos y cuales no lo son. Además, al tener el par de mapas de decisión se puede obtener un resultado más favorable en la binarización al utilizar el algoritmo VO. Al momento de aplicar el algoritmo VO a los mapas de decisión, se estableció el parámetro de tolerancia de este algoritmo a $T = 10$. En la Figura 3.4(a) se muestra un mapa de decisión para la imagen sobreexpuesta, al que llamaremos $\boldsymbol{P_1}$, obtenido de la binarización de la imagen $\boldsymbol{G_1}$ por medio

del modelo de Mezcla de Gaussianas y el algoritmo VO. En la Figura 3.4(b) se muestra un mapa de decisión para imagen subexpuesta, al que llamaremos $\boldsymbol{P_2}$, obtenido también de la binarización de la imagen $\boldsymbol{G_2}$ por medio de la binarización de Mezcla de Gaussianas y el algoritmo VO. En el mapa de decisión $\boldsymbol{P_1}$, la región en color blanco corresponde a los píxeles no nítidos y la región de color negro a los píxeles nítidos. Por otro lado, en el mapa de decisión $\boldsymbol{P_2}$ el color negro corresponde a los píxeles nítidos y el blanco a los no nítidos.

(a) ***Mapa de Decisión de la Imagen Sobreexpuesta*** $\boldsymbol{P_1}$

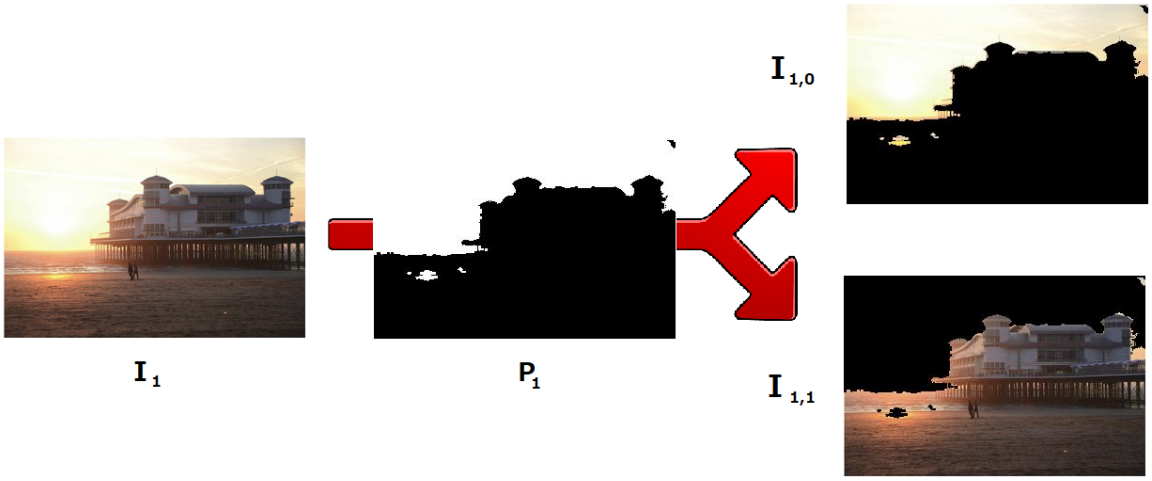
(b) ***Mapa de Decisión de la Imagen Subexpuesta*** $\boldsymbol{P_2}$

Figura 3.4: Mapas de Decisión de las Imágenes a Fusionar

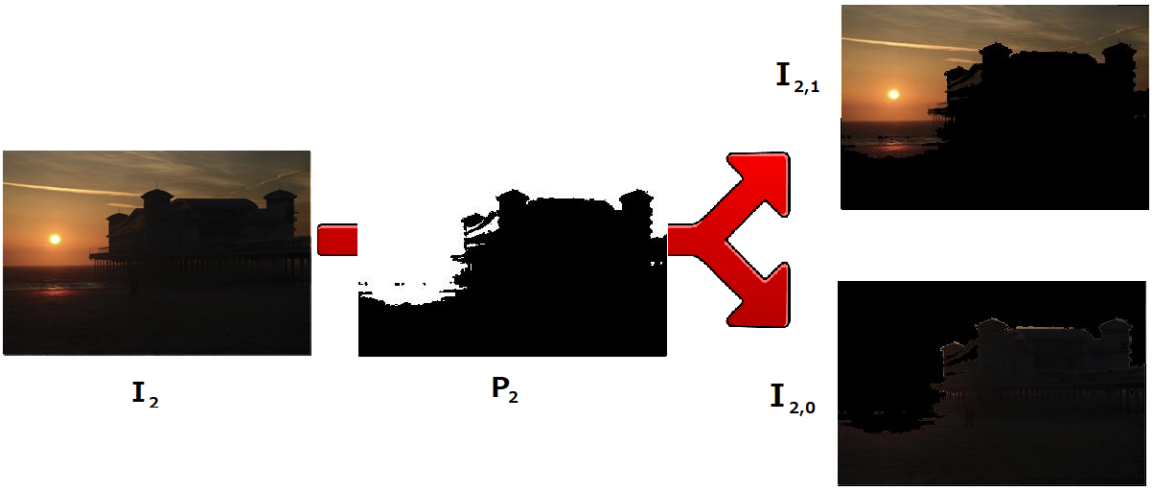
En la Figura 3.5(a) se muestra como se divide la imagen $\boldsymbol{I_1}$ en dos regiones: la región de píxeles no nítidos $\boldsymbol{I_{1,0}}$ y la región de píxeles nítidos $\boldsymbol{I_{1,1}}$, mediante un mapa de decisión $\boldsymbol{P_1}$. De igual forma, en la Figura 3.5(b) se muestra como se divide la imagen $\boldsymbol{I_2}$ en dos regiones: la región de píxeles no nítidos $\boldsymbol{I_{2,0}}$ y la región de píxeles nítidos $\boldsymbol{I_{2,1}}$, mediante un mapa de decisión $\boldsymbol{P_2}$. La división de la imagen $\boldsymbol{I_1}$ en las regiones $\boldsymbol{I_{1,0}}$ e $\boldsymbol{I_{1,1}}$ se calcula con (3.1) y la división de la imagen $\boldsymbol{I_2}$ en las regiones $\boldsymbol{I_{2,0}}$ e $\boldsymbol{I_{2,1}}$ se calcula con (3.2). Los mapas de decisión $\boldsymbol{P_1}$ y $\boldsymbol{P_2}$ son imágenes binarias con ceros y unos, siendo el color blanco el valor de 1 y el color negro el valor de 0.

$$\begin{aligned} \boldsymbol{I_{1,0}}[r,c] &= \boldsymbol{I_1}[r,c]\boldsymbol{P_1}[r,c] \\ \boldsymbol{I_{1,1}}[r,c] &= \boldsymbol{I_1}[r,c](1-\boldsymbol{P_1}[r,c]) \end{aligned} \tag{3.1}$$

$$\begin{aligned} \boldsymbol{I_{2,0}}[r,c] &= \boldsymbol{I_2}[r,c](1-\boldsymbol{P_2}[r,c]) \\ \boldsymbol{I_{2,1}}[r,c] &= \boldsymbol{I_2}[r,c]\boldsymbol{P_2}[r,c] \end{aligned} \tag{3.2}$$



(a) ***Clasificación en Regiones de Píxeles Nítidos y No Nítidos para la Imagen Sobreexpuesta Utilizando P_1***



(b) ***Clasificación en Regiones de Píxeles Nítidos y No Nítidos para la Imagen Subexpuesta Utilizando P_2***

Figura 3.5: Clasificación de Píxeles de la Imágenes a Fusionar con Mapas de Decisión Diferentes

Para realizar la fusión de $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$ proponemos juntar las regiones de píxeles nítidos. Sin embargo, al ver los mapas de decisión $\boldsymbol{P_1}$ y $\boldsymbol{P_2}$ podemos observar que no son iguales. Por lo tanto, las regiones de píxeles nítidos $\boldsymbol{I_{1,1}}$ e $\boldsymbol{I_{2,1}}$ no son complementarias y no podremos juntar estas dos regiones en una sola imagen para realizar la fusión. Entonces, para solucionar este problema calculamos un solo mapa de decisión, al que llamaremos $\boldsymbol{P}$, aplicando el operador AND binario ($\wedge$) utilizando el mapa de decisión $\boldsymbol{P_1}$ y $\boldsymbol{P_2}$ con (3.3). Al calcular el nuevo mapa de decisión $\boldsymbol{P}$ aplicando (3.3), obtenemos la imagen binaria que se muestra en la Figura 3.6. Teniendo el mapa de decisión final $\boldsymbol{P}$ podemos realizar la fusión de imágenes para obtener una imagen fusionada con las dos regiones de píxeles nítidos.

$$\boldsymbol{P}[r,c] = \boldsymbol{P_1}[r,c] \wedge \boldsymbol{P_2}[r,c] \tag{3.3}$$

Figura 3.6: Mapa de Decisión Final $\boldsymbol{P}$

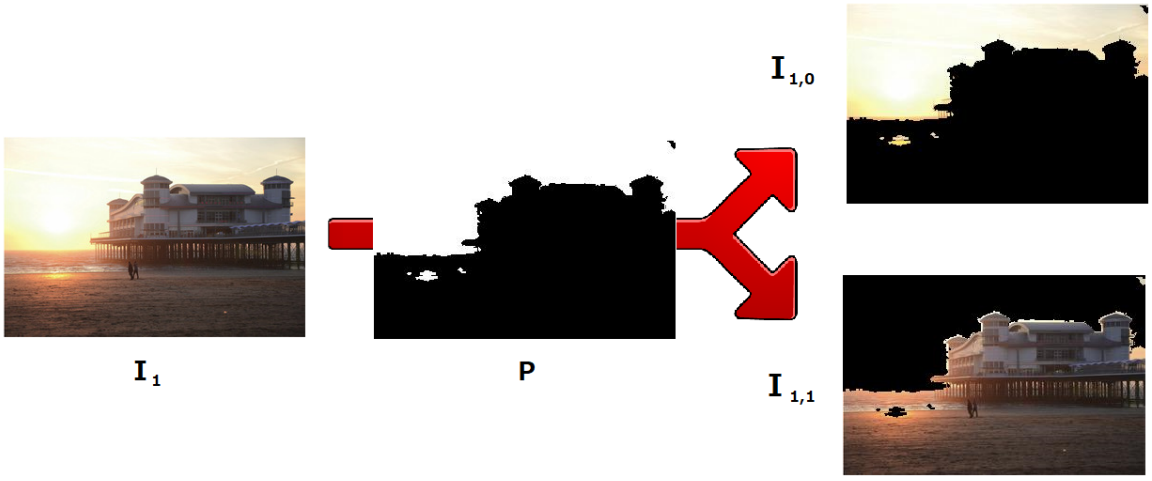
3.2. Fusión Preliminar de Imágenes

Con el nuevo mapa de decisión $\boldsymbol{P}$ podemos dividir las imágenes $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$ en sus respectivas regiones de píxeles nítidos y no nítidos de forma que las regiones de píxeles nítidos ahora sean complementarias. En la Figura 3.7(a) se muestra como la imagen $\boldsymbol{I_1}$ se divide en las regiones $\boldsymbol{I_{1,0}}$ e $\boldsymbol{I_{1,1}}$ utilizando un mapa de decisión $\boldsymbol{P}$. De la misma manera, en la Figura 3.7(b) se muestra como la imagen $\boldsymbol{I_2}$ se divide en las regiones $\boldsymbol{I_{2,0}}$ e $\boldsymbol{I_{2,1}}$ utilizando un mismo mapa de decisión $\boldsymbol{P}$. Ahora, la división de la imagen $\boldsymbol{I_1}$ en las regiones $\boldsymbol{I_{1,0}}$ e $\boldsymbol{I_{1,1}}$ se calcula con (3.4) y la división de la imagen $\boldsymbol{I_2}$ en las regiones $\boldsymbol{I_{2,0}}$ e $\boldsymbol{I_{2,1}}$ con (3.5).

Esto se asimila a la Ecuación (2.1) del trabajo de Félix Calderón [Calderón16].

$$\begin{aligned} \boldsymbol{I_{1,0}}[r,c] &= \boldsymbol{I_1}[r,c]\boldsymbol{P}[r,c] \\ \boldsymbol{I_{1,1}}[r,c] &= \boldsymbol{I_1}[r,c](1-\boldsymbol{P}[r,c]) \end{aligned} \tag{3.4}$$

$$\begin{aligned} \boldsymbol{I_{2,0}}[r,c] &= \boldsymbol{I_2}[r,c](1-\boldsymbol{P}[r,c]) \\ \boldsymbol{I_{2,1}}[r,c] &= \boldsymbol{I_2}[r,c]\boldsymbol{P}[r,c] \end{aligned} \tag{3.5}$$

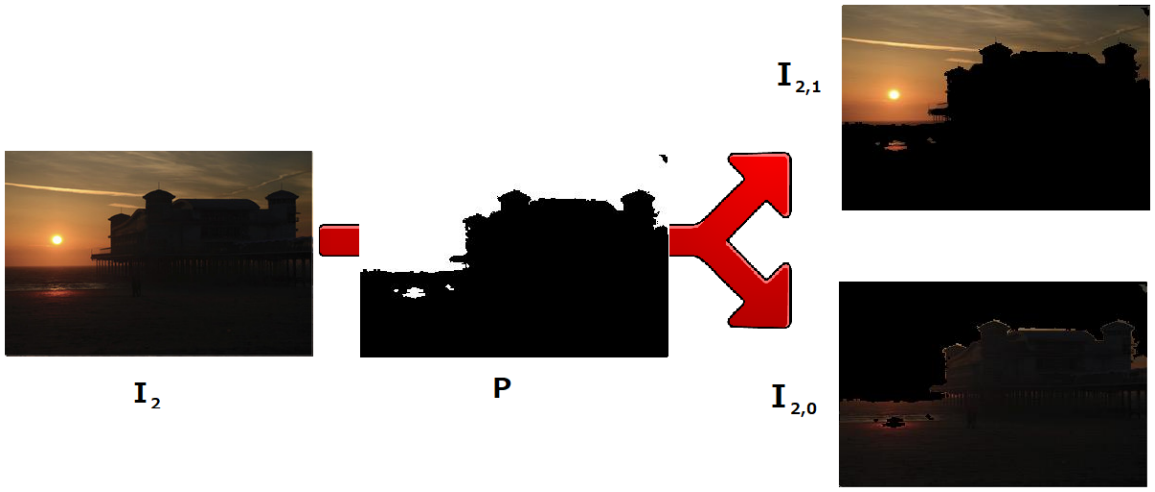


(a) ***Clasificación en Regiones de Píxeles Nítidos y No Nítidos para la Imagen Sobreexpuesta Utilizando P***

Para fusionar la imagen sobreexpuesta $\boldsymbol{I_1}$ y la imagen subexpuesta $\boldsymbol{I_2}$ combinamos las dos regiones de píxeles nítidos, es decir las regiones $\boldsymbol{I_{1,1}}$ e $\boldsymbol{I_{2,1}}$, para generar una sola imagen. La fusión de imágenes se realiza aplicando el criterio de la ecuación (3.6) en donde $\boldsymbol{I_F}$ es la imagen fusionada preliminar.

$$\boldsymbol{I_F}[r,c] = \begin{cases} \boldsymbol{I_1}[r,c] & \text{Si } \boldsymbol{P}[r,c] = 0 \\ \boldsymbol{I_2}[r,c] & \text{Si no} \end{cases} \tag{3.6}$$

Al aplicar el criterio de la ecuación (3.6) se obtiene una imagen fusionada preliminar $\boldsymbol{I_F}$ mostrada en la Figura 3.8.



(b) ***Clasificación en Regiones de Píxeles Nítidos y No Nítidos para la Imagen Subexpuesta Utilizando P***

Figura 3.7: Clasificación de Píxeles de la Imágenes a Fusionar con el Mismo Mapa de Decisión

Figura 3.8: Imagen Fusionada Preliminar $\boldsymbol{I_F}$

Al analizar la imagen fusionada preliminar $\boldsymbol{I_F}$ en la Figura 3.8, podemos observar que existe una inconsistencia entre los niveles de iluminación de las regiones de píxeles $\boldsymbol{I_{1,1}}$ e $\boldsymbol{I_{2,1}}$ que la componen. Esta inconsistencia de iluminación es más evidente en los bordes que unen a estas dos regiones. Entonces, es necesario ajustar estos niveles de iluminación para que el cambio entre las regiones no sea tan contrastante.

3.3. Ajuste de Niveles de Iluminación

Para ajustar los niveles de iluminación de la imagen $\boldsymbol{I_F}$ proponemos realizar una transformación lineal en las regiones $\boldsymbol{I_{1,1}}$ e $\boldsymbol{I_{2,1}}$ para cada canal de color con el propósito de mover los rangos dinámicos de cada región de forma que el cambio de iluminación entre las dos regiones no sea tan evidente. El rango dinámico de una región de píxeles es el intervalo de los valores de sus píxeles. Por ejemplo, en la Figura 3.9 se muestra el histograma del canal de color rojo de la región $\boldsymbol{I_{2,1}}$ donde el rango dinámico es aproximadamente de $[23, 255]$.

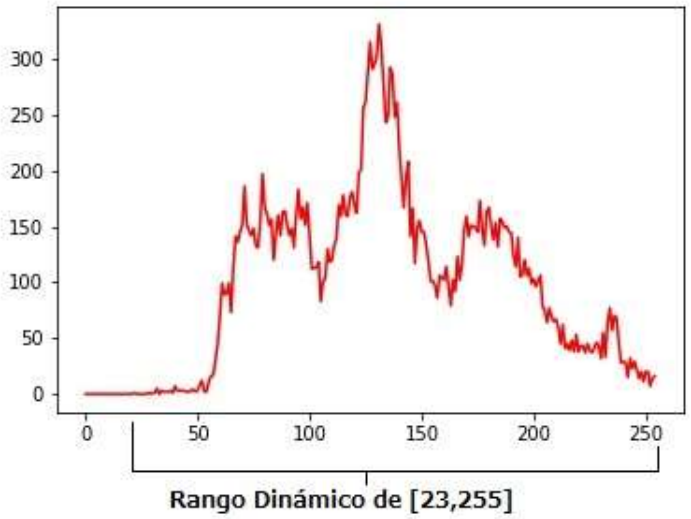


Figura 3.9: Histograma del Canal de Color Rojo de la Región $\boldsymbol{I_{2,1}}$

Al trasladar el rango dinámico para un píxel $\boldsymbol{I}[r, c]$ en el intervalo $[f_{min}, f_{max}]$ a un nuevo intervalo $[g_{min}, g_{max}]$, podemos calcular su nuevo valor $\boldsymbol{I_{Nuevo}}[r, c]$ con (3.7).

$$\boldsymbol{I_{Nuevo}}[r, c] = \frac{(\boldsymbol{I}[r, c] - f_{min})(g_{max} - g_{min})}{f_{max} - f_{min}} + g_{min} \tag{3.7}$$

Entonces, calculamos la imagen promedio $\boldsymbol{I_{Prom}}$ de $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$ con (3.8), la cual se muestra en la Figura 3.10.

$$\boldsymbol{I_{Prom}}[r, c] = \lfloor \frac{\boldsymbol{I_1}[r, c] + \boldsymbol{I_2}[r, c]}{2} \rfloor \tag{3.8}$$

Figura 3.10: Imagen Promedio $\boldsymbol{I_{Prom}}$

Después, utilizando el mapa de decisión $\boldsymbol{P}$ se divide la imagen $\boldsymbol{I_{Prom}}$ en las regiones correspondientes a $\boldsymbol{I_{1,1}}$ e $\boldsymbol{I_{2,1}}$, a las cuales denominaremos como $\boldsymbol{I_{Prom1}}$ e $\boldsymbol{I_{Prom2}}$ respectivamente, como se muestra en la Figura 3.11.

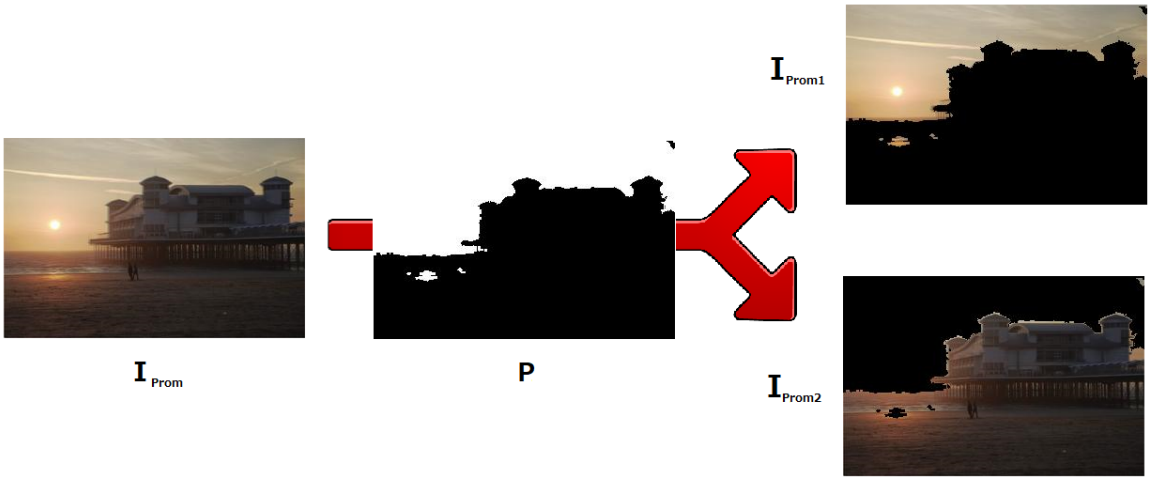


Figura 3.11: Imagen Promedio $\boldsymbol{I_{Prom}}$ Dividida en las Regiones $\boldsymbol{I_{Prom1}}$ e $\boldsymbol{I_{Prom2}}$ Utilizando $\boldsymbol{P}$

Ahora, se traslada el rango dinámico de la región $\boldsymbol{I_{1,1}}$ al rango dinámico de $\boldsymbol{I_{Prom1}}$ y el rango dinámico de la región $\boldsymbol{I_{2,1}}$ al rango dinámico de $\boldsymbol{I_{Prom2}}$ utilizando (3.7) en cada canal de color para obtener una nueva imagen fusionada como la que se muestra en la Figura 3.12.

Figura 3.12: Imagen Fusionada con Ajuste de Iluminación

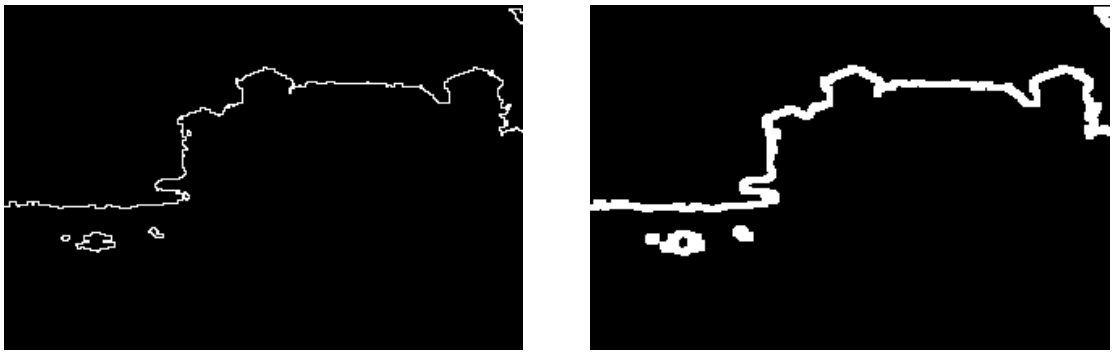
Si analizamos la imagen de la Figura 3.12 podemos observar que la inconsistencia en la iluminación entre las dos regiones ya no es tan contrastante, sin embargo, aun se puede notar en los bordes del par de regiones.

3.4. Corrección de Bordes de las Regiones

Para resolver el problema de los bordes proponemos calcular una imagen binaria de bordes $\boldsymbol{B}$ utilizando el mapa de decisión $\boldsymbol{P}$ de la misma forma que se calculó para el algoritmo VO con la ecuación (2.35). A la imagen de bordes $\boldsymbol{B}$ se le aplica un filtro de dilatación. Un filtro de dilatación para una imagen binaria se calcula aplicando el operador binario OR para los píxeles dentro de una ventana de tamaño $(2w+1) \times (2w+1)$, donde w determina qué tanto se dilatan los bordes en la imagen $\boldsymbol{B}$. La imagen de bordes con el filtro de dilatación la llamaremos $\boldsymbol{B}^*$ y se calcula con (3.9).

$$\boldsymbol{B}^*[r,c] = \bigvee_{i=r-w}^{r+w} \bigvee_{j=c-w}^{c+w} \boldsymbol{B}[i,j] \tag{3.9}$$

En la figura 3.13(a) se muestra la imagen de bordes binaria $\boldsymbol{B}$ y en la Figura 3.13(b) la imagen de bordes $\boldsymbol{B}^*$ aplicando un filtro de dilatación con $w = 2$.



(a) ***Imagen de Bordes B*** **(b)** ***Imagen de Bordes con Filtro de Dilatación B*****

Figura 3.13: Imágenes de Bordes

Después, generamos una nueva imagen aplicando un filtro de mediana a la imagen fusionada preliminar $\boldsymbol{I_F}$. El filtro de mediana calcula la mediana dentro de una ventana de tamaño $(2w+1) \times (2w+1)$ centrada en las coordenadas $[r, c]$ para un valor de w apropiado para tamaño de la imagen. Por ejemplo, para una señal de entrada en una dimensión $X = [2, 3, 80, 6, 2, 3]$ se obtiene una señal de salida del filtro $Y = [3, 6, 6, 3]$ para una $w = 1$, donde cada valor de la señal Y se calcula como:

$$Y[1] = mediana(2, 3, 80) = 3$$

$$Y[2] = mediana(3, 80, 6) = mediana(3, 6, 80) = 6$$

$$Y[3] = mediana(80, 6, 2) = mediana(2, 6, 80) = 6$$

$$Y[4] = mediana(6, 2, 3) = mediana(2, 3, 6) = 3$$

Al aplicar el filtro de mediana a la imagen $\boldsymbol{I_F}$ obtenemos una imagen emborronada a la cual llamaremos $\boldsymbol{I_m}$ mostrada en la Figura 3.14.

Figura 3.14: Imagen con Filtro de Mediana $\boldsymbol{I_m}$

Por último, para calcular la imagen fusionada final con bordes corregidos, a la que denominamos como $\boldsymbol{I_F^*}$, aplicamos el criterio dado en (3.10). La imagen fusionada final $\boldsymbol{I_F^*}$ se muestra en la Figura 3.15.

$$\boldsymbol{I_F^*}[r,c] = \begin{cases} \boldsymbol{I_F}[r,c] & \text{Si } \boldsymbol{B^*}[r,c] = 0 \\ \boldsymbol{I_m}[r,c] & \text{Si no} \end{cases} \tag{3.10}$$

Figura 3.15: Imagen Fusionada Final $\boldsymbol{I_F^*}$

3.5. Algoritmo de Fusión de Imágenes con Iluminación Complementaria

El algoritmo de Fusión de Imágenes con Iluminación Complementaria se muestra en el Algoritmo 6, el cual recibe como parámetros: la imagen sobreexpuesta $\boldsymbol{I_1}$, la imagen subexpuesta $\boldsymbol{I_2}$, la tolerancia T para el algoritmo VO, el tamaño de ventana w_b para el filtro de dilatación aplicado en la imagen de bordes $\boldsymbol{B}$ y el tamaño de ventana w_m para el filtro de mediana. Este algoritmo regresa una imagen que contiene la fusión de las imágenes $\boldsymbol{I_1}$ e $\boldsymbol{I_2}$. En el Algoritmo 6, la función *Grises* recibe una imagen a color y regresa la misma imagen en escala de grises calculada con (1.2). También, el ciclo de la linea 45 itera en cada uno de los canales de color en la escala RGB, es decir, cuando $k = 0$ se refiere al canal de color *rojo*, cuando $k = 1$ se refiere al canal de color *verde* y cuando $k = 2$ se refiere al canal de color *azul*. En las lineas 46 a 53 del Algoritmo 6, las funciones Min y Max devuelven el valor mínimo y máximo respectivamente de la matriz que recibe.

Algoritmo 6: Fusion(Entrada: $\boldsymbol{I_1}$,$\boldsymbol{I_2}$,T,w_b,w_m; Salida: $\boldsymbol{I_F^*}$)

1 $\boldsymbol{G_1} \leftarrow Grises(\boldsymbol{I_1})$

2 $\boldsymbol{G_2} \leftarrow Grises(\boldsymbol{I_2})$

3 $\theta_1 \leftarrow$ EM($\boldsymbol{G_1}$,2) ; `// Algoritmo 3`

4 $\theta_2 \leftarrow$ EM($\boldsymbol{G_2}$,2) ; `// Algoritmo 3`

5 Calcular el mapa de decisión $\boldsymbol{P_1}[r, c]$ con (2.28) utilizando θ_1 $\forall[r, c]$

6 Calcular el mapa de decisión $\boldsymbol{P_2}[r, c]$ con (2.28) utilizando θ_2 $\forall[r, c]$

para $k = 1$ ***a*** 2 **hacer**
  $\boldsymbol{F_0}[r,c] \leftarrow |\boldsymbol{I_k}[r,c] - \mu_0|$ donde $\mu_0 \in \theta_k \ \forall [r,c]$
  $\boldsymbol{F_1}[r,c] \leftarrow |\boldsymbol{I_k}[r,c] - \mu_1|$ donde $\mu_1 \in \theta_k \ \forall [r,c]$
  $\boldsymbol{H_0}[r,c] \leftarrow \boldsymbol{F_0}[r,c] + \boldsymbol{H_0}[r,c-1] + \boldsymbol{H_0}[r-1,c] - \boldsymbol{H_0}[r-1,c-1] \ \forall [r,c]$
  $\boldsymbol{H_1}[r,c] \leftarrow \boldsymbol{F_1}[r,c] + \boldsymbol{H_1}[r,c-1] + \boldsymbol{H_1}[r-1,c] - \boldsymbol{H_1}[r-1,c-1] \ \forall [r,c]$
  si $N_r > N_c$ **entonces**
    $w_{max} \leftarrow \lfloor \frac{N_c}{2} - 1 \rfloor$
  en otro caso
    $w_{max} \leftarrow \lfloor \frac{N_r}{2} - 1 \rfloor$
  fin
  $\boldsymbol{W} \leftarrow \mathrm{VO}(\boldsymbol{P_k}, T, w_{max})$
  para $r = 0$ ***a*** $N_r - 1$ **hacer**
    para $c = 0$ ***a*** $N_c - 1$ **hacer**
      Calcular la suma S_0 con $\boldsymbol{H_0}$ con (2.39) donde $w = \boldsymbol{W}[r,c]$
      Calcular la suma S_1 con $\boldsymbol{H_1}$ con (2.39) donde $w = \boldsymbol{W}[r,c]$
      si $S_1 < S_0$ **entonces**
        $\boldsymbol{P_k}[r,c] \leftarrow 1$
      en otro caso
        $\boldsymbol{P_k}[r,c] \leftarrow 0$
      fin
    fin
  fin
fin
$\boldsymbol{P}[r,c] \leftarrow \boldsymbol{P_1}[r,c] \wedge \boldsymbol{P_2}[r,c] \ \forall [r,c]$

para $r = 0$ ***a*** $N_r - 1$ **hacer**
  para $c = 0$ ***a*** $N_c - 1$ **hacer**
    si $\boldsymbol{P}[r,c] = 0$ **entonces**
      $\boldsymbol{I_F}[r,c] \leftarrow \boldsymbol{I_1}[r,c]$
    en otro caso
      $\boldsymbol{I_F}[r,c] \leftarrow \boldsymbol{I_2}[r,c]$
    fin
    $\boldsymbol{I_{Prom}}[r,c] \leftarrow \lfloor \frac{\boldsymbol{I_1}[r,c]+\boldsymbol{I_2}[r,c]}{2} \rfloor$
    $\boldsymbol{I_{1,1}}[r,c] \leftarrow \boldsymbol{I_1}[r,c](1 - \boldsymbol{P}[r,c])$
    $\boldsymbol{I_{2,1}}[r,c] \leftarrow \boldsymbol{I_2}[r,c]\boldsymbol{P}[r,c]$
    $\boldsymbol{I_{P,1,1}}[r,c] \leftarrow \boldsymbol{I_{Prom}}[r,c](1 - \boldsymbol{P}[r,c])$
    $\boldsymbol{I_{P,2,1}}[r,c] \leftarrow \boldsymbol{I_{Prom}}[r,c]\boldsymbol{P}[r,c]$
  fin
fin
para $k = 0$ ***a*** 2 **hacer**
  $f_{1,min} \leftarrow Min(\boldsymbol{I_{1,1}^{(k)}})$
  $f_{2,min} \leftarrow Min(\boldsymbol{I_{2,1}^{(k)}})$
  $f_{1,max} \leftarrow Max(\boldsymbol{I_{1,1}^{(k)}})$
  $f_{2,max} \leftarrow Max(\boldsymbol{I_{2,1}^{(k)}})$
  $g_{1,min} \leftarrow Min(\boldsymbol{I_{P,1,1}^{(k)}})$
  $g_{2,min} \leftarrow Min(\boldsymbol{I_{P,2,1}^{(k)}})$
  $g_{1,max} \leftarrow Max(\boldsymbol{I_{P,1,1}^{(k)}})$
  $g_{2,max} \leftarrow Max(\boldsymbol{I_{P,2,1}^{(k)}})$
  para $r = 0$ ***a*** $N_r - 1$ **hacer**
    para $c = 0$ ***a*** $N_c - 1$ **hacer**
      si $\boldsymbol{P}[r,c] = 1$ **entonces**
        $\boldsymbol{I_F}[r,c] \leftarrow \frac{(\boldsymbol{I_F}[r,c]-f_{1,min})(g_{1,max}-g_{1,min})}{f_{1,max}-f_{1,min}} + g_{1,min}$
      en otro caso
        $\boldsymbol{I_F}[r,c] \leftarrow \frac{(\boldsymbol{I_F}[r,c]-f_{2,min})(g_{2,max}-g_{2,min})}{f_{2,max}-f_{2,min}} + g_{2,min}$
      fin
    fin
  fin
fin

Calcular la imagen de bordes $\boldsymbol{B}$ de $\boldsymbol{P}$ con (2.35)
Calcular la imagen de bordes dilatada $\boldsymbol{B^*}$ con (3.9) donde $w = w_b$
Calcular la imagen con filtro de mediana $\boldsymbol{I_m}$ de $\boldsymbol{I_F}$ donde $w = w_m$
para $r = 0$ ***a*** $N_r - 1$ **hacer**
    para $c = 0$ ***a*** $N_c - 1$ **hacer**
        si $\boldsymbol{B^*}[r, c] = 0$ **entonces**
            $\boldsymbol{I^*_F}[r, c] \leftarrow \boldsymbol{I_F}[r, c]$
        en otro caso
            $\boldsymbol{I^*_F}[r, c] \leftarrow \boldsymbol{I_m}[r, c]$
        fin
    fin
fin
devolver $\boldsymbol{I^*_F}$

3.6. Resumen del Capítulo

En este capítulo presentamos el algoritmo de fusión para obtener una imagen que contenga los píxeles nítidos de una imagen sobreexpuesta y una imagen subexpuesta. El algoritmo de fusión de imágenes integra un proceso de binarización de imágenes utilizando el modelo GMM [Chatzis08] con el algoritmo EM [Dempster97] y el algoritmo VO [Reyes21] para obtener un mapa de decisión binario. El mapa de decisión permite determinar que píxeles son sobreexpuestos, cuales son nítidos y cuales son subexpuestos a partir de la cantidad de iluminación. La imagen fusionada final es totalmente dependiente del resultado de binarización de imágenes, ya que no solo permite determinar que píxeles son nítidos, sino que también es utilizado en el proceso de ajuste de iluminación para eliminar las inconsistencias de iluminación entre las 2 imágenes a fusionar. En el siguiente capítulo mostramos los resultados de la fusión de imágenes reales y sintéticas.

Capítulo 4

Pruebas y resultados

En este capítulo se muestran los resultados del proceso de fusión de imágenes sintéticas y de imágenes reales. También, se describen distintas medidas para cuantificar la similitud entre la imagen original y la fusión de imágenes para el caso de los pares de imágenes sintéticas. Asimismo, se describe como solucionar el problema de registro de imágenes utilizando una transformación afín a partir de 3 coordenadas de cada imagen a fusionar.

4.1. Medidas de Similitud de Imágenes

Las medidas de similitud de imágenes proporcionan una medida cuantitativa del grado de coincidencia entre dos imágenes. La comparación de dos imágenes es una operación fundamental en muchos algoritmos de fusión de imágenes. Existe una gran variedad de medidas de similitud de imágenes, sin embargo para este trabajo tomamos algunas de las más utilizadas como lo son: el Índice de Similitud Estructural (SSIM por sus siglas en inglés) [Wang04], la Razón del Pico de Señal a Ruido (PSNR por sus siglas en inglés) [Wang04] y el Índice Universal de Calidad de Imágenes (UQI por sus siglas en inglés) [Wang02], los cuales son descritos a continuación.

4.1.1. Índice de Similitud Estructural

El Índice de Similitud Estructural (SSIM) [Wang04] esta basado en la comparación de la iluminación, contraste y estructura entre dos imágenes. El SSIM de una ventana $\boldsymbol{X}$ en una imagen original y una ventana $\boldsymbol{Y}$ en una imagen de prueba con un tamaño en común de $N \times N$ se calcula como:

$$SSIM(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}, c_1, c_2) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.1}$$

Donde μ_x es la media de $\boldsymbol{X}$, μ_y es la media de $\boldsymbol{Y}$, σ_x^2 es la varianza de $\boldsymbol{X}$, σ_y^2 es la varianza de $\boldsymbol{Y}$, $\sigma_{x,y}$ es la covarianza de $\boldsymbol{X}$ y $\boldsymbol{Y}$, c_1 y c_2 son dos variables para prevenir la división entre cero. La variable c_1 y c_2 son por defecto $c_1 = (0.01L)^2$ y $c_2 = (0.03L)^2$, donde L es el valor máximo que puede tomar un píxel, es decir $L = 255$. El valor del SSIM tiene un rango de $[0, 1]$, siendo 1 el valor máximo de similitud indicando que las dos imágenes son idénticas.

4.1.2. Razón del Pico de Señal a Ruido

La Razón del Pico de Señal a Ruido (PSNR por sus siglas en inglés) [Wang04] define la relación entre la máxima energía posible de una señal y el ruido que afecta a esta señal. El PSNR se expresa generalmente con la unidad decibel (dB). El PSNR se calcula con (4.2) para un par de imágenes $\boldsymbol{I}$ y $\boldsymbol{K}$ de tamaño $N_r \times N_c$, donde MAX es el valor máximo que puede tomar un píxel de la imagen y MSE es el error cuadrático medio que es calculado con (4.3).

$$PSNR(\boldsymbol{I}, \boldsymbol{K}) = 20log_{10}\left(\frac{MAX}{\sqrt{MSE(\boldsymbol{I}, \boldsymbol{K})}}\right) \tag{4.2}$$

$$MSE(\boldsymbol{I}, \boldsymbol{K}) = \frac{1}{N_r N_c}\sum_{r=0}^{N_r-1}\sum_{c=0}^{N_c-1}|\boldsymbol{I}[r, c] - \boldsymbol{K}[r, c]|^2 \tag{4.3}$$

El PSNR es muy usado en la codificación para la compresión de imágenes. Los valores aceptables de esta métrica son mayores a 30 para una imagen con ruido e infinito

para una imagen sin ruido. El PSNR puede variar bastante con muy poco ruido en la imagen dado que es un valor logarítmico.

4.1.3. Índice Universal de Calidad de Imágenes

El Índice Universal de Calidad de Imágenes [Wang02] (UQI por sus siglas en inglés) es otra medida de similitud la cual es muy fácil de calcular. El UQI para una señal original X y una señal de prueba Y se calcula con (4.4), donde μ_x es la media de la señal X, μ_y es la media de la señal Y, σ_x^2 es la varianza de la señal X, σ_y^2 es la varianza de la señal Y y $\sigma_{x,y}$ es la covarianza entre la señal X y la señal Y. El UQI tiene un rango de $[0, 1]$, siendo el valor de 1 cuando las imágenes son idénticas.

$$Q(\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}) = \frac{4\sigma_{xy}\mu_x\mu_y}{(\sigma_x^2 + \sigma_y^2)(\mu_x^2 + \mu_y^2)} \tag{4.4}$$

4.2. Transformación Afín para la Solución del Problema de Registro

Para obtener un resultado favorable al utilizar el algoritmo de Fusión de Imágenes con Iluminación Complementaria, se debe tener un par de imágenes a fusionar en donde cada píxel $\boldsymbol{I_1}[r, c]$ corresponda al píxel $\boldsymbol{I_2}[r, c]$. Sin embargo, para una imagen $\boldsymbol{I_1}$ y una imagen $\boldsymbol{I_2}$, los píxeles $\boldsymbol{I_1}[r, c]$ e $\boldsymbol{I_2}[r, c]$ para todo r y para todo c no siempre son correspondientes. Entonces, si consideramos cada imagen como un plano cartesiano y relacionamos cada píxel a una coordenada de su correspondiente plano podemos alinear las coordenadas de cada plano a través de una transformación afín.

La transformación afín de una imagen consiste en mapear las coordenadas de cada píxel de la imagen a las deseadas. Con esta transformación podemos realizar las transformaciones geométricas de translación, rotación, escalamiento y proyección. Dada una coordenada (x, y) podemos realizar la transformación afín multiplicando una matriz $\boldsymbol{H}$ con las coordenadas homogéneas $[x, y, 1]^T$ para obtener las nuevas coordenadas homogéneas $[\hat{x}, \hat{y}, \hat{z}]^T$ como en la Ecuación (4.5).

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \boldsymbol{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.5}$$

Para nuestro problema se desea encontrar la matriz $\boldsymbol{H}$ con el motivo de realizar una transformación afín para empatar las imágenes a fusionar a partir de múltiples píxeles correspondientes. Por lo tanto, se tiene el sistema de ecuaciones (4.6) donde $\boldsymbol{P}$ es la matriz que contiene las coordenadas de los píxeles del origen, $\hat{\boldsymbol{P}}$ es la matriz con las coordenadas de los píxeles en el destino, $\boldsymbol{H}$ es la matriz de la transformación afín y N es la cantidad de coordenadas de los píxeles. Las coordenadas en $\boldsymbol{P}$ son correspondientes a las coordenadas en $\hat{\boldsymbol{P}}$.

$$\hat{\boldsymbol{P}} = \boldsymbol{H}\boldsymbol{P}$$

$$\begin{bmatrix} \hat{x}_1 & \hat{x}_2 & \hat{x}_3 \\ \hat{y}_1 & \hat{y}_2 & \hat{y}_3 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.6}$$

Dadas las coordenadas de los píxeles $\boldsymbol{P}$ y $\hat{\boldsymbol{P}}$, podemos decir que la multiplicación vectorial de $\hat{\boldsymbol{P}}$ y $\boldsymbol{H}\boldsymbol{P}$ es igual a cero, es decir, $\hat{\boldsymbol{P}} \times \boldsymbol{H}\boldsymbol{P} = 0$ dado que $\boldsymbol{H}\boldsymbol{P}$ y $\hat{\boldsymbol{P}}$ son paralelos ya que son correspondientes. Si la matriz $\boldsymbol{H}$ la consideramos como un vector columna de tal forma que $\boldsymbol{H} = [h_1, h_2, h_3, h_4, h_5,\ h_6, h_7, h_8, h_9]^T$ podemos arreglar la ecuación (4.6) de forma que la podemos representarla como (4.7) [Hartley99].

$$\begin{bmatrix} 0 & 0 & 0 & -\hat{x}_1 & -\hat{y}_1 & -1 & y_1\hat{x}_1 & y_1\hat{y}_1 & y_1 \\ \hat{x}_1 & \hat{y}_1 & 1 & 0 & 0 & 0 & -x_1\hat{x}_1 & -x_1\hat{y}_1 & -x_1 \\ 0 & 0 & 0 & -\hat{x}_2 & -\hat{y}_2 & -1 & y_2\hat{x}_2 & y_2\hat{y}_2 & y_2 \\ \hat{x}_2 & \hat{y}_2 & 1 & 0 & 0 & 0 & -x_2\hat{x}_2 & -x_2\hat{y}_2 & -x_2 \\ 0 & 0 & 0 & -\hat{x}_3 & -\hat{y}_3 & -1 & y_N\hat{x}_3 & y_N\hat{y}_3 & y_3 \\ \hat{x}_3 & \hat{y}_3 & 1 & 0 & 0 & 0 & -x_N\hat{x}_3 & -x_N\hat{y}_3 & -x_3 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \boldsymbol{B}\boldsymbol{H} = 0 \tag{4.7}$$

Este sistema de ecuaciones puede ser resuelto utilizando la descomposición en valores singulares (SVD por sus siglas en inglés) sobre la matriz $\boldsymbol{B}$. La SVD calcula tres matrices: $\boldsymbol{U}$,$\boldsymbol{S}$ y $\boldsymbol{V}$. La solución del sistema se encuentra normalizando el último vector columna en $\boldsymbol{V}$ con respecto a último elemento de $\boldsymbol{V}$, es decir, el último elemento de este mismo vector columna posicionado en la esquina inferior derecha de $\boldsymbol{V}$.

4.3. Pruebas con Imágenes Sintéticas

Es difícil determinar si el algoritmo de Fusión de Imágenes con Iluminación Complementaria tiene buenos resultados en imágenes reales dado que no hay una forma concreta de cuantificar el desempeño del algoritmo para imágenes reales. Esto es porque el problema a resolver en este y trabajo es un problema inverso. Sin embargo, dadas las características de las imágenes a fusionar (siendo para la subexposición falta de luz y para la sobreexposición exceso de luz) podemos generar un par de imágenes sintéticas a partir de una imagen original. El par de imágenes sintéticas nos permiten comparar el resultado de su fusión con la original cuantitativamente. Algunas de las métricas que permiten cuantificar la similitud entre dos imágenes son: el SSIM, el PSNR, y el UQI, las cuales se describieron previamente en este capítulo.

Para crear un par imágenes sintéticas generamos manualmente un mapa de decisión. En este mapa de decisión, los píxeles de color negro representan los objetos en primer plano y los píxeles en color blanco representan el fondo. Para la imagen sintética sobreexpuesta, los píxeles negros del mapa de decisión correspondientes se mantienen iguales a los de la imagen original y los píxeles blancos correspondientes en la imagen original les aplicamos una transformación lineal para mover su rango dinámico. Por otro lado, para la imagen sintética subexpuesta los píxeles blancos del mapa de decisión correspondientes son igual a los píxeles de la imagen original y los píxeles negros correspondientes en la imagen original les aplicamos una transformación lineal diferente para mover su rango dinámico.

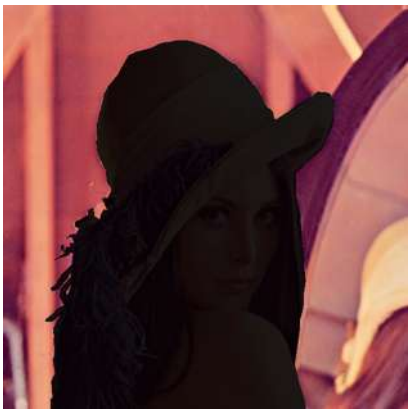
4.3.1. Prueba 1

En la Figura 4.1(a) se muestra la imagen original de donde se generan el par de imágenes a fusionar. En la Figura 4.1(b) se puede observar el mapa de decisión creado manualmente. En la Figura 4.1(c) y 4.1(d) se muestran la imagen sintética sobreexpuesta y la imagen sintética subexpuesta generadas por el mapa de decisión de la Figura 4.1(b) respectivamente.

(a) ***Imagen de Original***

(b) ***Mapa de Decisión***

(c) ***Imagen Sintética Sobreexpuesta***



(d) ***Imagen Sintética Subexpuesta***

Figura 4.1: Imágenes para la Fusión de la Prueba 1

En el proceso de fusión de imágenes, primero se convierten las imágenes a fusionar a imágenes en escala de grises como las que se muestran en las Figuras 4.2(a) y 4.2(b). Luego,

se calculan los mapas de decisión de cada una de estas imágenes en escala de grises mediante el modelo GMM aplicando el algoritmo VO como las que se muestran en las Figuras 4.2(c) y 4.2(d). Después, se calcula un mapa de decisión final realizando el operador binario AND con el par de mapas de decisión para obtener una imagen como la de la Figura 4.2(e). Con este mapa de decisión tomamos los píxeles nítidos de cada una de las imágenes originales para realizar la fusión de estas imágenes para obtener una imagen como la que se muestra en la Figura 4.2(f). Con el mapa de decisión final y esta imagen fusionada preliminar se realiza la transformación lineal para ajustar la iluminación y obtener una imagen como la que se muestra en la Figura 4.2(g). Por último, se suavizan los bordes de las regiones del mapa de decisión sobre la imagen fusionada preliminar para obtener una imagen fusionada final como la que se muestra en la Figura 4.2(h).

(a) ***Imagen de Sobreexpuesta en Escala de Grises***

(b) ***Imagen de Subexpuesta en Escala de Grises***

(c) ***Mapa de Decisión de Imagen Sobreexpuesta***

(d) ***Mapa de Decisión de Imagen Subexpuesta***

(e) ***Mapa de Decisión Final***

(f) ***Imagen Fusionada Preliminar***

(g) ***Imagen Fusionada con Ajuste de Iluminación***

(h) ***Imagen Fusionada Final***

Figura 4.2: Imágenes en el Proceso de Fusión

Al fusionar las imágenes de la Figura 4.1(c) y la Figura 4.1(d) con el Algoritmo 6 con $T = 10$, una $w_b = 1$ y una $w_m = 2$ se obtiene la imagen fusionada de la Figura 4.3. Al comparar cuantitativamente esta imagen fusionada con la imagen original, obtenemos un SSIM = 0.939681, un PSNR = 19.98767 y un UQI = 0.942992.

Figura 4.3: Imagen Fusionada de las Imágenes Sintéticas de la Prueba 1

4.3.2. Prueba 2

En la Figura 4.4(a) se muestra la imagen original de donde se generan el par de imágenes a fusionar. En la Figura 4.4(b) se puede observar el mapa de decisión creado manualmente. En la Figura 4.4(c) y 4.4(d) se muestran la imagen sintética sobreexpuesta y la imagen sintética subexpuesta generadas por el mapa de decisión de la 4.4(b) respectivamente.

(a) ***Imagen de Original***

(b) ***Mapa de Decisión***

(c) ***Imagen Sintética Sobreexpuesta***

(d) ***Imagen Sintética Subexpuesta***

Figura 4.4: Imágenes para la Fusión de la Prueba 2

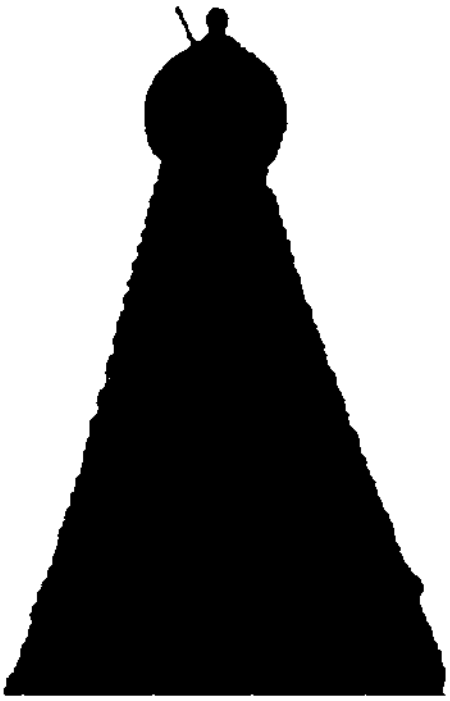
Al fusionar las imágenes de la Figura 4.4(c) y la Figura 4.4(d) con el Algoritmo 6 con $T = 10$, una $w_b = 1$ y una $w_m = 2$ se obtiene la imagen fusionada de la Figura 4.5. Al comparar cuantitativamente esta imagen fusionada con la imagen original, obtenemos un SSIM = 0.987329, un PSNR = 32.18788 y un UQI = 0.980042.

Figura 4.5: Imagen Fusionada de las Imágenes Sintéticas de la Prueba 2

4.3.3. Prueba 3

En la Figura 4.6(a) se muestra la imagen original de donde se generan el par de imágenes a fusionar. En la Figura 4.6(b) se puede observar el mapa de decisión creado manualmente. En la Figura 4.6(c) y 4.6(d) se muestran la imagen sintética sobreexpuesta y la imagen sintética subexpuesta generadas por el mapa de decisión de la 4.6(b) respectivamente.

(a) ***Imagen de Original***



(b) ***Mapa de Decisión***

(c) ***Imagen Sintética Sobreexpuesta***

(d) ***Imagen Sintética Subexpuesta***

Figura 4.6: Imágenes para la Fusión de la Prueba 3

Al fusionar las imágenes de la Figura 4.6(c) y la Figura 4.6(d) con el Algoritmo 6 con $T = 10$, una $w_b = 1$ y una $w_m = 2$ se obtiene la imagen fusionada de la Figura 4.7. Al comparar cuantitativamente esta imagen fusionada con la imagen original, obtenemos un SSIM = 0.989355, un PSNR = 30.753027 y un UQI = 0.999025.

Figura 4.7: Imagen Fusionada de las Imágenes Sintéticas de la Prueba 3

En la Tabla 4.1 se muestra un resumen de los resultados de las pruebas para la fusión con imágenes sintéticas.

Prueba	SSIM	PSNR	UQI
Prueba 1	0.9396	19.9876	0.9429
Prueba 2	0.9873	32.1878	0.9800
Prueba 3	0.9893	30.7530	0.9990

Tabla 4.1: Resultados de Fusión con Imágenes Sintéticas

4.4. Pruebas con Imágenes Reales

En esta sección se muestran distintos pares de imágenes reales a fusionar y el resultado al aplicar la fusión de imágenes con el Algoritmo 6. Para cada prueba del Algoritmo 6 establecemos a $T = 10$, $w_b = 1$ y $w_m = 2$. Para las pruebas con imágenes reales se aplicó una transformación afín a algunas imágenes para lograr que coincidiera la posición con su respectivo par y solucionar el problema de registro.

4.4.1. Prueba 1

En la Figura 4.8(a) se muestra una imagen sobreexpuesta y en la Figura 4.8(b) una imagen subexpuesta de la misma escena las cuales se utilizan para realizar la fusión de imágenes. Se aplicó una transformación afín a la imagen sobreexpuesta con la matriz de la transformación afín $\boldsymbol{H}$ de a continuación.

$$\boldsymbol{H} = \begin{bmatrix} 0.9957 & 3.4653 \times 10^{-3} & 4.1588 \\ 5.4597 \times 10^{-3} & 0.9989 & -7.2587 \\ 1.5022 \times 10^{-5} & -1.4156 \times 10^{-5} & 1.0001 \end{bmatrix} \tag{4.8}$$

En el proceso de fusión, primero se convierten las imágenes a fusionar a escala de grises como las que se muestran en las Figuras 4.9(a) y 4.9(b). Luego, calculamos los mapas de decisión de estas imágenes mediante el modelo GMM junto con el algoritmo VO para obtener un par de imágenes como las que se muestran en las Figuras 4.9(c) y 4.9(d). Después, calculamos un solo mapa de decisión mediante el operador binario AND para obtener una imagen como la de la Figura 4.9(e). Usando este mapa de decisión calculamos la imagen fusionada preliminar como la que se muestra en la Figura 4.9(f). Ahora, se ajusta la iluminación de la imagen fusionada preliminar para obtener una imagen como la que se muestra en la Figura 4.9(g). Por último, se arreglan los bordes entre las regiones del mapa de decisión en la imagen fusionada preliminar para obtener la imagen fusionada final como la que se muestra en la Figura 4.9(h).

(a) ***Imagen Sobreexpuesta Real***

(b) ***Mapa Subexpuesta Real***

Figura 4.8: Imágenes Reales a Fusionar de la Prueba 1

(a) ***Imagen Sobreexpuesta en Escala de Grises***

(b) ***Imagen Subexpuesta en Escala de Grises***

(c) ***Mapa de Decisión de la Imagen Sobreexpuesta***

(d) ***Mapa de Decisión de la Imagen Subexpuesta***

(e) ***Mapa de Decisión Final***

(f) ***Fusión Preliminar***

(g) ***Fusión Preliminar con Ajuste de Iluminación***

(h) ***Imagen Fusionada Final***

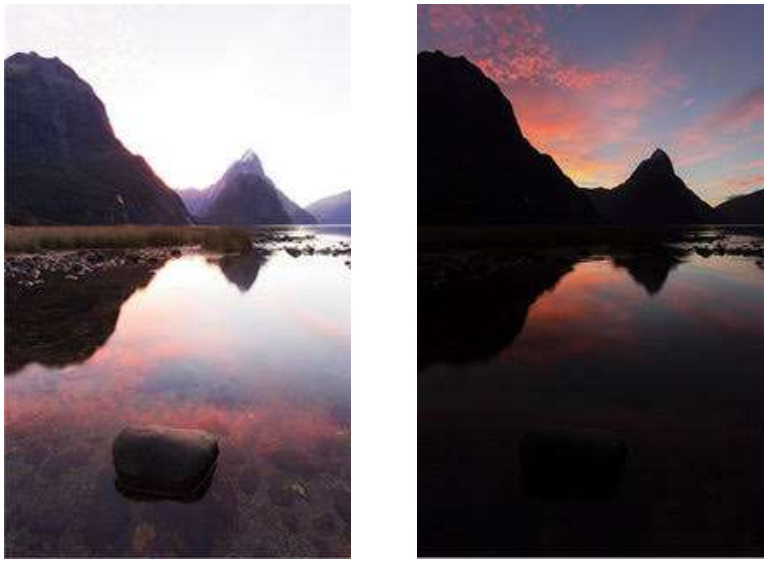
Figura 4.9: Imágenes en el Proceso de Fusión

4.4.2. Prueba 2

En la Figura 4.10(a) se muestra una imagen sobreexpuesta y en la Figura 4.10(b) una imagen subexpuesta de la misma escena para fusionar. Se aplicó una transformación afín a la imagen sobreexpuesta con la matriz de transformación afín $\boldsymbol{H}$ que se muestra a continuación.

$$\boldsymbol{H} = \begin{bmatrix} 0.9924 & 9.7759 \times 10^{-4} & -4.0985 \\ -2.5785 \times 10^{-3} & 0.9844 & 0.4319 \\ -7.8343 \times 10^{-5} & -6.7642 \times 10^{-5} & 1.0003 \end{bmatrix} \tag{4.9}$$

Al fusionar el par de imágenes de la Figura 4.10 con el Algoritmo 6 obtenemos como resultado la imagen fusionada que se muestra en la Figura 4.11.



(a) ***Imagen Sobreexpuesta Real*** **(b)** ***Imagen Subexpuesta Real***

Figura 4.10: Imágenes Reales a Fusionar de la Prueba 2

Figura 4.11: Imagen Fusionada de las Imágenes Reales de la Prueba 2

4.4.3. Prueba 3

En la Figura 4.12(a) se muestra una imagen sobreexpuesta y en la Figura 4.12(b) una imagen subexpuesta de la misma escena para fusionar. Se aplicó una transformación afín a la imagen sobreexpuesta con la matriz de transformación afín $\boldsymbol{H}$ que se muestra a continuación.

$$\boldsymbol{H} = \begin{bmatrix} 1.0057 & 2.7872 \times 10^{-2} & -2.2662 \\ -4.6323 \times 10^{-3} & 1.0295 & 2.8766 \\ -3.8155 \times 10^{-5} & 1.5105 \times 10^{-4} & 1.0005 \end{bmatrix} \tag{4.10}$$

Al fusionar el par de imágenes de la Figura 4.12 con el Algoritmo 6 obtenemos como resultado la imagen fusionada que se muestra en la Figura 4.13.

(a) ***Imagen Sobreexpuesta Real*** **(b)** ***Imagen Subexpuesta Real***

Figura 4.12: Imágenes Reales a Fusionar de la Prueba 3

Figura 4.13: Imagen Fusionada de las Imágenes Reales de la Prueba 3

4.4.4. Prueba 4

En la Figura 4.14(a) se muestra una imagen sobreexpuesta y en la Figura 4.14(b) una imagen subexpuesta de la misma escena para fusionar. Al fusionar este par de imágenes con el Algoritmo 6 obtenemos como resultado la imagen fusionada que se muestra en la Figura 4.15.

(a) ***Imagen Sobreexpuesta Real*** **(b)** ***Imagen Subexpuesta Real***

Figura 4.14: Imágenes Reales a Fusionar de la Prueba 4

Figura 4.15: Imagen Fusionada de las Imágenes Reales de la Prueba 4

4.4.5. Prueba 5

En la Figura 4.16(a) se muestra una imagen sobreexpuesta y en la Figura 4.16(b) una imagen subexpuesta de la misma escena. Al fusionar este par de imágenes con el Algoritmo 6 obtenemos como resultado la imagen fusionada que se muestra en la Figura 4.17.

(a) ***Imagen Sobreexpuesta Real*** **(b)** ***Imagen Subexpuesta Real***

Figura 4.16: Imágenes Reales a Fusionar de la Prueba 5

Figura 4.17: Imagen Fusionada de las Imágenes Reales de la Prueba 5

4.4.6. Prueba 6

Para esta prueba se utilizaron imágenes tomadas con un celular Galaxy A10 en donde se configuro la exposición de la cámara para obtener cada una de las imágenes. En la Figura 4.18(a) se muestra una imagen sobreexpuesta y en la Figura 4.18(b) una imagen subexpuesta de la misma escena. Al fusionar el par de imágenes de la Figura 4.18 con el Algoritmo 6 obtenemos como resultado la imagen fusionada que se muestra en la Figura 4.19.

(a) ***Imagen Sobreexpuesta Real*** **(b)** ***Imagen Subexpuesta Real***

Figura 4.18: Imágenes Reales a Fusionar de la Prueba 6

Figura 4.19: Imagen Fusionada de las Imágenes Reales de la Prueba 6

4.4.7. Prueba 7

De igual forma, esta las imágenes de esta prueba fueron tomadas por un celular Galaxy A10 modificando los valores de exposición de iluminación de la cámara. En la Figura 4.20(a) se muestra una imagen sobreexpuesta y en la Figura 4.20(b) una imagen subexpuesta de la misma escena. Al fusionar este par de imágenes con el Algoritmo 6 obtenemos como resultado la imagen fusionada que se muestra en la Figura 4.21.

(a) ***Imagen Sobreexpuesta Real*** **(b)** ***Imagen Subexpuesta Real***

Figura 4.20: Imágenes Reales a Fusionar de la Prueba 7

Figura 4.21: Imagen Fusionada de las Imágenes Reales de la Prueba 7

4.5. Resumen del Capítulo

En este capítulo mostramos resultados de forma cuantitativa con los pares de imágenes sintéticas y de forma cualitativa con los pares de imágenes reales del algoritmo de fusión de imágenes propuesto. En los resultados de forma cuantitativa observamos que para las medidas de similitud (SSIM, PSNR y UQI) las imágenes fusionadas a partir de las imágenes sintéticas son muy parecidas a sus imágenes originales. En los resultados de forma cualitativa podemos observar que para la fusión de imágenes reales se tienen buenos resultados subjetivamente excepto en algunas imágenes en donde se tiene el reflejo de la luz sobre el agua como en las imágenes fusionadas de las Figuras 4.11 y 4.15.

Capítulo 5

Conclusiones y Trabajos Futuros

En este trabajo se presentó el desarrollo de un algoritmo de fusión de imágenes. Este algoritmo combina un par de imágenes de forma que la imagen fusionada contenga los píxeles con mayor nitidez. El par de imágenes a fusionar tienen características un tanto opuestas ya que una imagen contiene píxeles con exceso de iluminación y la otra contiene píxeles con falta de iluminación. Este algoritmo de Fusión de Imágenes incorpora el modelo GMM [Chatzis08] junto con el algoritmo VO [Reyes21] para binarizar imágenes y clasificar sus píxeles para posteriormente juntar los píxeles nítidos para crear una imagen fusionada preliminar a la que se le aplica un post proceso para mejorar su calidad.

5.1. Conclusiones Generales

En el Capítulo 2 expusimos distintos trabajos relacionados a la binarización de imágenes para clasificar sus píxeles en dos clases ya que es una parte fundamental del algoritmo de fusión de imágenes propuesto. Con la binarización de las imágenes a fusionar se obtuvo un mapa de decisión, el cual es indispensable para la mayoría de los pasos en el algoritmo de fusión de imágenes. Al comparar los algoritmos de binarización, se observó que el modelo GMM [Chatzis08] es el mejor candidato para utilizar para la binarización de imágenes. También, en la parte de binarización se analizó cuantitativamente que el algoritmo VO [Reyes21] ayuda al desempeño de estos algoritmos de binarización. En el Capítulo 4 podemos notar que en los resultados cuantitativos del Algoritmo 6 de fusión de imágenes, en

donde se utilizan las imágenes sintéticas, que son buenos ya que la mayoría de las pruebas tienen un PSNR por arriba de los 30dB y un SSIM y UQI por arriba del 0.9. De forma cualitativa, las fusiones con imágenes reales muestran resultados agradables a la vista del ojo humano en gran parte ya que se obtiene una imagen nítida. El problema que se tiene en algunos casos es que la calidad y el contraste de las imágenes originales a fusionar perjudican a que la imagen fusionada final no tenga buena calidad. Existen casos en los que se observan inconsistencias en los bordes, esto es porque los pares de imágenes a fusionar no están totalmente empatados coordenada a coordenada. Otro motivo por el cual puede existir inconsistencia en los bordes es porque el valor de tolerancia T no es el apropiado.

Rani Kusum hace una comparación de distintos algoritmos de fusión de imágenes en donde muestra de forma cuantitativa el desempeño de cada algoritmo mediante la métrica de PSNR [Rani13], la cual es una de las métricas más populares para la comparación de imágenes. Los algoritmos que se comparan en este trabajo son los mencionados en la Tablas 2.1. La fusión simple por máximos tiene un PSNR de 26.86dB, la fusión por promedio tiene un PSNR de 25.48dB, la fusión por PCA tiene un PSNR de 76.44dB, y por último la fusión por DWT tiene un PSNR de 76.44dB. Es evidente que nuestro algoritmo no es tan bueno comparado con la fusión por PCA y DWT. No obstante, en un trabajo presentado por la institución Computer Graphics and Multimedia Laboratory se analiza una comparación de las métricas SSIM y PSNR, en donde concluyen que a pesar de que el PSNR sea una métrica muy popular, no es una buena métrica para la estimación de la calidad de una imagen [Group19]. Hay demasiados parámetros que influyen mucho en el valor de PSNR y apenas afectan a la calidad visual (por ejemplo, pequeños cambios en el brillo, el contraste, el matiz y la saturación). Por otro lado, la métrica SSIM tiene un rendimiento bastante mejor en comparación con PSNR. Pero SSIM es casi insensible a los cambios de brillo, contraste, matiz y saturación. Además, la mayoría de trabajos de fusión de imágenes presentan sus resultados utilizando el PSNR.

Otra distinción del algoritmo de Fusión de Imágenes con Iluminación Complementaria con respecto a otros algoritmos de fusión es que es mucho mas sencillo y fácil de

entender que la fusión por PCA o por DWT dado que estos algoritmos no están basados en la fusión a nivel píxel. Existen también propuestas también de fusión de imágenes por medio de redes neuronales, sin embargo, estas son muy costosas en cuanto a la cantidad de información que requieres y el tiempo que ocupan para su entrenamiento en comparación con el algoritmo de fusión propuesto.

5.2. Trabajos Futuros

Dado que para poder utilizar el algoritmo de fusión de imágenes con iluminación complementaria se necesita que el par de imágenes a fusionar empaten una con la otra, se propone incorporar al algoritmo de Fusión de Imágenes con Iluminación Complementaria la solución del problema para registro de forma automática en lugar de solucionar este problema manualmente estableciendo las coordenadas necesarias antes de utilizar el algoritmo. La solución al problema de registro de imágenes de forma automática no se incluyó en este trabajo porque generalmente el costo de solucionar este problema es alto en cuanto al tiempo. Otra mejora que se propone es calcular automáticamente los parámetros T, w_m y w_b que necesita el algoritmo de fusión de imágenes descrito en el Algoritmo 6.

También, al ser el algoritmo de Fusión de Imágenes con Iluminación Complementaria un poco costoso en cuanto al tiempo, se propone optimizar el proceso de manera que se pueda paralelizar y de ser posible ser aplicado en tiempo real. De esta manera, el algoritmo puede ser utilizado en celulares inteligentes con múltiples cámaras para aumentar la calidad de las fotografías.

Referencias

[Calderón16] Calderón, F., Carrillo, A. G., y Flores, J. J. 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, 2016. ISSN 1697-7920. doi: 10.1016/j.riai.2016.07.002.
URL https://polipapers.upv.es/index.php/RIAI/article/view/9262

[cap21] What is shutter speed? understanding shutter speed in photography, Feb 2021.
URL https://capturetheatlas.com/what-is-shutter-speed-in-photograph

[Chatzis08] Chatzis, S. P., Kosmopoulos, D. I., y Varvarigou, T. A. Signal modeling and classification using a robust latent space model based on ¡formula formulatype=ïnline»¡tex¿t¡/tex¿¡/formula¿distributions. *IEEE Transactions on Signal Processing*, 56(3):949–963, 2008. doi:10.1109/TSP.2007.907912.

[crehana] crehana. ¿qué es la sobreexposición y subexposición fotografica ?[2021]. https://www.crehana.com/mx/blog/fotografia/sobreexposicion-fotografica/. Accessed: 2021-Aug-09.

[Dass12] Dass, R. Image segmentation techniques. *IJECT*, 2012. ISSN 2230-7109 (Online) — 230-9543 (Print).

[Dempster97] Dempster, A. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, 1997.

[Dhingra15] Dhingra, N. Fusion of fuzzy enhanced overexposed and underexposed images. *ScienceDirect, ELSEVIER, Procedia Computer science*, 2015. ISSN 54(2015)738–745.

[Fucci15] Fucci, V. Iso exposure exercise, Oct 2015.
URL http://www.getoffgreenauto.com/iso-exposure-exercise/

[Group19] Group, M. G. . M. L. V. Ways of cheating on popular objective metrics: blurring, noise, super-resolution and others, 2019.
URL https://videoprocessing.ai/metrics/ways-of-cheating-on-popular-objective-metrics.html

[Hartley99] Hartley, R. y Zisserman, A. Multiple view geometry. 1999.

[Ho11] Ho, P.-G. *Image Segmentation*. IntechOpen, 2011.

[Kolekar15] Kolekar, N. B. Decision level based image fusion using wavelet transform and support vector machine. *IJSER*, 2015. ISSN 2347-3878.

[Li93] Li, C. y Lee, C. Minimum cross entropy thresholding. *Pattern Recognition*, 26(4):617–625, 1993. ISSN 0031-3203. doi:https://doi.org/10.1016/0031-3203(93)90115-D.

[M .Chandana11] M .Chandana, S. A. y Kumar, N. A hybrid multi- focus medical image fusion based on wavelet transform. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, 2, 08 2011. doi:ISSN:2079-2557.

[MacQueen67] MacQueen, J. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[Malviya09] Malviya, A. y Bhirud, S. G. Bhirud," image fusion of digital images.

India, International Journal of Recent Trends in Engineering, págs. 3–146, 2009.

[Mansurov20] Mansurov, N. What is aperture? understanding aperture in photography, Apr 2020.
URL https://photographylife.com/what-is-aperture-in-photography

[Metz78] Metz, C. E. Basic principles of roc analysis. *Seminars in Nuclear Medicine*, 8(4), 1978. doi:283-98.

[Naidu08] Naidu, D. V. y Raol, J. Pixel-level image fusion using wavelets and principal component analysis. *Defence Science Journal*, 58, 05 2008. doi: 10.14429/dsj.58.1653.

[NCERT19] NCERT, N. C. o. E. R. y Training. Mathematics textbook for class xi. 2019.

[Otsu79] Otsu, N. A threshold selection method from gray-level histograms. *IEEE*, 1979. doi:10.1117/1.1631315.

[Patel11] Patel, C. K. S. M. Pixel based and wavelet based image fusion methods with their comparative study. *National Conference on Recent Trends in Engineering Technology*, 05 2011.

[Plant17] Plant, I. Backlighting photography tips for photo magic: Opg, Ene 2017.
URL https://www.outdoorphotographyguide.com/article/backlight-photography-tips-for-photo-magic/#

[Rani08] Rani, K. y Sharma, R. Study of different image fusion algorithm. 9001, 04 2008.

[Rani13] Rani, K. Study of different image fusion algorithm. *IJETAE*, 2013. ISSN 2250-2459.

[Reyes21] Reyes, C. Binarización de imágenes de texto con iluminación no uniforme. 2021.

[Shapiro00] Shapiro, L. y Stockman, G. Computer vision. 2000.

[Shuang08] Shuang, L. A region-based technique for fusion of high resolution images using mean shift segmentation. *Commission VII, WG VII/6*, 2008.

[theAtlas] the Atlas, C. ¿qué es la exposición en fotografía? ajustes básicos de exposición. https://capturetheatlas.com/es/que-es-la-exposicion-en-fotografia/#definicion. Accessed: 2021-Aug-09.

[thrAtlas] thr Atlas, C. ¿qué es el histograma en fotografía? aprendiendo a leer el histograma. https://capturetheatlas.com/es/histograma-fotografia/#que. Accessed: 2021-Aug-09.

[Viola01] Viola, P. y Jones, M. Rapid object detection using a boosted cascade of simple features. *En Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition*, CVPR 2001, págs. I–I. IEEE, 2001.

[Wang02] Wang, Z. y Bovik, A. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002. doi:10.1109/97.995823.

[Wang04] Wang, Z., Bovik, A., Sheikh, H., y Simoncelli, E. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi:10.1109/TIP.2003.819861.