Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica

Desarrollo de una etapa de pre-procesamiento para reducción de ruido, mejoramiento de contraste y de nitidez en imágenes digitalizadas de geles de electroforesis

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado acedémico de Licenciatura

Miguel Angel Aguilar Ulloa

Cartago, 19 de diciembre del 2007

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA EN ELECTRÓNICA

PROYECTO DE GRADUACIÓN

TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

Ing. Arys Carrasquilla Batista

Profesora lectora

Ing. Eduardo Interiano Salguero

Profesor lector

Dr. Pablo Alvarado Moya

Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería en Electrónica.

Cartago, 19 de diciembre del 2007.

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introducimiento conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la resposabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 19 de diciembre del 2007

Miguel Angel Aguilar Ulloa

Ced: 304020248

Resumen

Hoy en día el área de procesamiento de imágenes es utilizada en disciplinas como la medicina, la biología, ingeniería entre otras. El interés de éste proyecto se concentra en una aplicación para caracterización molecular de organismos, particularmente en lo referente al mejoramiento de la calidad de imágenes de geles de electroforesis.

En general, las imágenes de geles digitalizadas poseen defectos como el ruido que proviene del proceso de adquisición de la imagen, bajo contraste y baja nitidez que provienen tanto del proceso de adquisición debido a una inadecuada configuración de la cámara, así como de las mismas imágenes de geles que intrínsecamente se caracterizan por poseer un bajo contraste y baja nitidez.

En éste sentido el presente trabajo tiene como objetivo principal implementar y comparar diferentes algoritmos que reduzcan el ruido, mejoren el contraste y la definición de las imágenes de geles de electroforesis. En el caso de reducción de ruido se comparan los algoritmos: *reductor de ruido gaussiano*, *filtro de mediana*, *reductor de ruido SUSAN* y *medias no locales*. En el caso de mejoramiento de contraste se comparan: *ecualización de histograma* y *mejoramiento de gradiente*. Finalmente, en el caso de mejoramiento de nitidez se comparan dos variantes del algoritmo *enmascarado de desenfoque*, donde en una se emplea una máscara gaussiana para el desenfoque y en otra una máscara laplaciana.

En éste trabajo se propone un enfoque objetivo para la evaluación de los algoritmos mediante el llamado *Frente de Pareto* que permite un análisis multiobjetivo, es decir, considera simultáneamente varias medidas de aptitud que compiten entre sí. Así, se definen diferentes medidas de aptitud para la reducción de ruido, mejoramiento de contraste y nitidez, para cuantificar el rendimiento de los algoritmos. Así, el Frente de Pareto permite comparar "todas" las parametrizaciones de un algoritmo por un lado, y comparar diferentes algoritmos por otro.

[0] ***Palabras Claves:*** *procesamiento de imágenes, reducción de Ruido, mejoramiento de contraste, mejoramiento de nitidez, evaluación multiobjetivo, frente de Pareto*

Abstract

The area of image processing is used in disciplines such as medicine, biology, engineering, etc. This work focuses on the field of molecular characterization of organisms, particularly in referring to the improvement of the quality of images of electrophoresis gels.

In general, the digitized gels images exhibit defects like noise originated in the process of acquisition, low contrast and sharpness that come both processing acquisition due to inadequate configuration of the camera, and of the same images of gels which are inherently characterized by a low contrast and low sharpness.

In this sense this work's main objective is to implement and compare different algorithms that reduce noise, enhance contrast and sharpening of images of electrophoresis gels. In the case of noise reduction the following algorithms are compared: gaussian denoiser, median filter, SUSAN denoiser and Non Local Means denoiser. In the case of contrast enhancement the following algoritms are compared: histogram equalization and improvement gradient. Finally, for sharpening it compares two variants of unsharp masking algorithm, in the first case is used a Gaussian mask for blur and another case a Laplacian mask is used.

A key aspect of this work is that proposes an objective approach for the evaluation of algorithms by the Pareto Front, which allows a multiobjetive analysis, it considers diferents aptitude measures that competing among they. In this way, this work defines various aptitude measures for noise reduction, contrast enhancemente and sharpness, to quantify the performance of the algorithms. Thus, the Pareto Front enables comparisons between "all" parametrizations of an algorithm on the one hand and compare different algorithms on the other.

[0] ***Keywords:*** *Image Processing, Image Enhancement, Denoising, Contrast Enhancement, Sharpening.*

A mi querida madre,
y a mis abuelos

Agradecimiento

Primero y antes que nada, dar gracias a Dios y la Virgen de la Ángeles, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Éste proyecto de graduación está enteramente dedicado a mi familia. A mi madre Leda María quien siempre me ha apoyado y me ha dado el cariño necesario para crear las condiciones idóneas para el éxito en mis estudios, además por brindarme un hogar cálido y enseñarme que la perseverancia y el esfuerzo son el camino para lograr las metas que me proponga.

A mi tía Rosa Lorena por estar siempre a mi lado en todo momento como una madre, desde que inicié mi carrera universitaria, por creer en mí y que yo podía alcanzar la meta que culmino con el presente proyecto.

A mis abuelos Dulce María y Marco Aurelio por contribuir en mi formación desde muy pequeño. Principalmente a mi abuelo por inculcarme valores indispensables para ser un hombre de bien, y aunque ya no me acompaña, su recuerdo y enseñanzas siempre están en mi mente dándome fuerzas para seguir adelante.

A mi madrina Cecilia por sus invaluables consejos en momentos difíciles que me inspiraron paz y tranquilidad. También a mi padrino Marco Aurelio y a su esposa Amparito por tratarme como su hijo, por estar siempre pendientes de mí, dándome apoyo y seguridad.

A mis tías abuelas en especial a Rosa por estar siempre al tanto de mis estudios, y por el cariño que me brindaron como si fuera un nieto suyo.

A mi novia María le agradezco su infinita compresión en aquellos momentos, que por motivos de estudio tuvimos que sacrificar el compartir tiempo juntos. Además, por ser un apoyo incondicional, porque su amor y cariño fueron y son motivación para seguir adelante.

A mi profesor asesor el Dr. Pablo Alvarado por su generosidad al brindarme la oportunidad de realizar este Proyecto y por permitirme recurrir a su capacidad y experiencia en

un marco de confianza y amistad, fundamentales para la concreción de este trabajo.

A mi compañero Róger por ser un excelente socio de estudio y trabajo. Además, por colaborar con el éxito a lo largo de toda mi carerra universitaria.

Y en general a toda mi familia que siempre han estado pendientes, apoyándome en el estudio y en mis proyectos. ¡Gracias!.

Miguel Angel Aguilar Ulloa
Diciembre 2007

Índice general

Índice de figuras v

Índice de Tablas xi

Glosario xiii

Lista de símbolos y abreviaciones xv

1 Introducción 1

1.1 Caracterización molecular de organismos 1

1.2 Problemática del procesamiento de imágenes con mala calidad 3

1.3 Solución propuesta para el mejoramiento de la calidad de imágenes 4

1.4 Objetivos y estructura de éste trabajo 6

2 Marco teórico 7

2.1 Fundamentos de procesamiento digital de imágenes 7

2.1.1 Conceptos generales 7

2.1.2 Mejoramiento de imágenes 8

2.2 Trabajos previos en mejoramiento de imágenes 18

2.2.1 Reducción de ruido 18

2.2.2 Trabajos realizados en mejoramiento de contraste 23

2.2.3 Mejoramiento de nitidez 26

3 Método de evaluación de algoritmos **29**

3.1 Evaluación de algoritmos 29

3.2 Frente de Pareto 31

3.3 Funciones de aptitud 33

3.3.1 Funciones de aptitud para algoritmos de reducción de ruido 34

3.3.2 Funciones de aptitud de contraste en una imagen 35

3.3.3 Funciones de aptitud de nitidez en una imagen 36

4 Algoritmos Implementados **39**

4.1 Algoritmos para la reducción de ruido 39

4.1.1 Modelos de Ruido 39

4.1.2 Reductor de Ruido Gaussiano 42

4.1.3 Filtro de Mediana 43

4.1.4 Reductor de Ruido SUSAN 45

4.1.5 Medias no Locales 48

4.2 Algoritmos para el mejoramiento de contraste 53

4.2.1 Ecualización de Histograma 54

4.2.2 Mejoramiento de Gradiente 61

4.3 Algoritmos para el mejoramiento de nitidez 67

4.3.1 Enmascaramiento de Desenfoque 67

5 Análisis de Resultados **71**

5.1 Reducción de ruido 72

5.2 Mejoramiento de contraste 87

5.3 Mejoramiento de nitidez 95

5.4 Combinación de algoritmos de mejoramiento de imágenes 99

6 Conclusiones y Recomendaciones **103**

6.1 Conclusiones 103
6.2 Recomendaciones 106

Bibliografía **107**

A LTI-Lib **111**

B Referencia de Clases **115**

B.1 Medias no Locales 115
B.1.1 Descripción Detallada 116
B.1.2 Documentación para el Constructor y el Destructor 117
B.1.3 Documentación de las Funciones Miembro 118
B.1.4 Documentación de los Datos Miembro 121
B.2 Mejoramiento de Gradiente 123
B.2.1 Descripción Detallada 123
B.2.2 Documentación del Constructor y Destructor 125
B.2.3 Documentación de las Funciones Miembro 126
B.2.4 Documentación de los Datos Miembro 128
B.3 Máscaras para Mejoramiento de Nitidez 130
B.3.1 Descripción detallada 131
B.3.2 Documentación de Enumeraciones Miembro 134
B.3.3 Documentación del Constructor y Destructor 134
B.3.4 Documentación de las Funciones Miembro 135
B.4 Máscara de Desenfoque 136
B.4.1 Descripción detallada 136
B.4.2 Documentación del Constructor y Destructor 138
B.4.3 Documentación de las Funciones Miembro 138
B.4.4 Documentación de los Datos Miembro 141

Índice alfabético **143**

Índice de figuras

1.1 Ejemplo de imagen de gel de electroforesis [5] utilizando AFLP 2

1.2 Proceso de binarización de gel [5] 2

1.3 Diagrama de bloques del sistema automatizado de análisis de imágenes [5] 3

1.4 Etapa de preprocesamiento del sistema de análisis automático de imágenes de geles de electroforesis 5

1.5 Elementos del mejoramiento de la calidad en una imagen 5

2.1 Rejilla $\mathbb{G}^2$ 8

2.2 Vecindario de 3x3 de un píxel (i, j) en una imagen 9

2.3 Proceso de filtrado lineal espacial 11

2.4 Tipos de expansiones de las fronteras de una imagen: (a) cero, (b) reflejada, (c) periódica, (d) constante. 12

2.5 Diagrama de bloques del proceso de filtrado en el dominio de la frecuencia [21]. 14

2.6 Estrutura piramidal de imágenes [21]. 16

2.7 Proceso de degradación y mejoramiento de imágenes [21]. 18

2.8 Clasificación de algoritmos de reducción de ruido [27]. 20

2.9 Clasificación de algoritmos de mejoramiento de contraste. 23

2.10 Clasificación de algoritmos de mejoramiento de nitidez. 26

3.1 Alternativas de evaluación de algoritmos según Zhang [41] 29

3.2 Frente de Pareto 32

3.3 Diagrama de dispersión de la magnitud del gradiente de la imagen original $\mathcal{N}$ que se supone con baja nitidez versus la magnitud del gradiente de la imagen filtrada con máscara de desenfoque $\widehat{\mathcal{I}} = A_{\mathbf{u}}[\mathcal{N}]$ 37

4.1 Imagen con ruido Gaussiano y Blanco con $\sigma = 0.1$ 40

4.2 Imagen con ruido impulsional 41

4.3 Distribución Gaussiana de una dimensión, con media 0 y $\sigma = 1$ 42

4.4 Distribución Gaussiana de dos dimensiones, con media $(0, 0)$ y $\sigma = 1$ . . . 43

4.5 Máscara Gaussiana 5×5, con $\sigma = 1$ 43

4.6 Cálculo de mediana, con una máscara de 3×3 44

4.7 Cuatro máscaras circulares en diferentes lugares de una imagen [33]. 45

4.8 Cuatro máscaras circulares con los USAN's mostrados como partes blancas de las máscaras [33]. 46

4.9 Esquema de la estrategía del algoritmo medias no locales. Vecindarios similares tienen un peso mayor, $w(p_m, p_{n_1})$ y $w(p_m, p_{n_2})$, que otros con vecindarios muy diferentes que tienen un peso mucho menor $w(p_m, p_{n_3})$ . . 51

4.10 Algoritmo multiescala para el mejoramiento del rendimiento del algoritmo de medias no locales 52

4.11 Algoritmo de bloques para el mejoramiento del rendimiento del algoritmo de medias no locales 53

4.12 Cuatro tipos básicos de imagen con sus correspondientes histogramas: (a) imagen oscura, (b) imagen clara, (c) imagen con bajo contraste, (d) imagen con alto contraste 56

4.13 Función de transformación de nivel de gris que satisface las condiciones de valor único e incremento monotónico 57

4.14 Funciones de transformación típicas: (a) oscurecimiento, (b) aclarado, (c) comprimido a los oscuros, (d) comprimido a los claros, (e) alto contraste, (f) bajo contraste, (g) enfatizado de sombras, (h) enfatizado de claros . . . 58

4.15 Mapeo de un píxel r a uno s mediante la función de transformación $s = A(r)$, en donde se cumple $P_s(s)ds = P_r(r)dr$ 59

4.16 Proceso de mejoramiento de contraste en una dimensión aplicado por el algoritmo [36, 36] 64

4.17 Pseudocódigo del algoritmo principal de mejoramiento de contraste 65

4.18 Pseudocódigo del algoritmo secundario que escala el valor de las colinas . . 66

4.19 Máscaras Laplacianas: (a) sin términos diagonales (b) con términos diagonales 69

4.20 Máscaras laplacianas para alto impulso: (a) sin términos diagonales (b) con términos diagonales 69

5.1 Imagen de gel de electroforesis empleada para la evaluación de los algoritmos. 71

5.2 Frentes de Pareto de los algoritmos de reducción de ruido con $\sigma = 0,01$. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Ruido de Método Escalar* (f_{rme}). 73

5.3 Imágenes procesadas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original $\mathcal{N}$ con un ruido gaussiano con $\sigma = 0,01$, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales. . . 75

5.4 Frente de Pareto de los algoritmos de reducción de ruido con $\sigma = 0,05$. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Método de Ruido Escalar* (f_{mre}). 76

5.5 Imágenes filtradas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original $\mathcal{N}$ con un ruido gaussiano con $\sigma = 0,05$, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales. 78

5.6 Frente de Pareto de los algoritmos de reducción de ruido con $\sigma = 0.10$. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Método de Ruido Escalar* (f_{mre}). 80

5.7 Imágenes filtradas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original $\mathcal{N}$ con un ruido gaussiano con $\sigma = 0,10$, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales. 82

5.8 Frente de Pareto de los algoritmos de reducción de ruido aplicados a imágenes con ruido impulsional. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Método de Ruido Escalar* (f_{rme}). 83

5.9 Imágenes filtradas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original con un ruido impulsional que cubre el 10% de la imagen, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales. 86

5.10 Frentes de pareto de los algoritmos de mejoramiento de contraste. *Mejoramiento de Contraste Promedio* (f_{mcp}) versus *Entropía* (f_e). 87

5.11 Imágenes procesadas por los algoritmos de mejoramiento de contraste y su correspondiente histograma: (a) imagen de gel de electroforesis con bajo contraste, (b) ecualizador de histograma, (c) mejoramiento de gradiente. . 90

5.12 Imágenes procesadas por los algoritmos de mejoramiento de contraste y su correspondiente histograma: (a) imagen de gel de electroforesis clara con bajo contraste, (b) ecualizador de histograma, (c) mejoramiento de gradiente. 92

5.13 Imágenes procesadas por los algoritmos de mejoramiento de contraste y su correspondiente histograma: (a) imagen de gel de electroforesis oscura con bajo contraste, (b) ecualizador de histograma, (c) mejoramiento de gradiente. 94

5.14 Frente de Pareto de las dos variantes de *enmascarado de desenfoque*. *Difuminación* (f_d) versus *Nitidez* (f_n). 96

5.15 Imágenes filtradas mediante el algoritmo enmascarado de desenfoque y su correspondiente diagrama de dispersión de la magnitud del gradiente de la imagen original $|\mathcal{N}|$ versus la magnitud del gradiente de la imagen filtrada $|\widehat{\mathcal{I}}|$: (a) imagen original con baja nitidez $\mathcal{N}$, (b) imagen procesada con máscara gaussiana, (c) imagen procesada con máscara laplaciana. 98

5.16 Secuencia de algoritmos para el mejoramiento de las imágenes de geles de electroforesis 99

5.17 Secuencia de mejoramiento de imágenes: (a) imagen de gel de electroforesis original, (b) imagen con disminución de ruido, (c) imagen mejorada en contraste, (d) imagen mejorada en nitidez. 100

A.1 Arquitectura de una función objeto. El usuario puede cambiar su comportamiento mediante los parámetros. La función objeto también tiene un estado, que enventualmente al igual que los parámetros pueden ser guardados[4] 114

B.1 Diagrama de herencia de la clase lti::nonLocalMeansDenoising 115

B.2 Diagrama de colaboración de la clase lti::nonLocalMeansDenoising 116

B.3 Diagrama de herencia de la clase lti::galContrastEnhancement 123

B.4 Diagrama de colaboración de la clase lti::galContrastEnhancement 123

B.5 Diagrama de herencia de la clase lti::sharpeningKernels 130

B.6 Diagrama de colaboración de la clase lti::sharpeningKernels 131

B.7 Diagrama de herencia de la clase lti::unsharpMasking 136

B.8 Diagrama de colaboración de la clase lti::unsharpMasking 136

Índice de Tablas

4.1 Parámetros del Reductor de Ruido Gaussiano 43

4.2 Parámetros del Filtro de Mediana 44

4.3 Parámetros del Reductor de Ruido SUSAN 48

4.4 Parámetros del algoritmo de Medias no Locales 54

4.5 Parámetros del algoritmo de Ecualización de Histograma 61

4.6 Parámetros del algoritmo de Máscara de Desenfoque 69

5.1 Medidas de aptitud de las imágenes de la Figura 5.3 74

5.2 Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.3 74

5.3 Medidas de aptitud de las imágenes de la Figura 5.5 77

5.4 Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.5 79

5.5 Medidas de aptitud de las imágenes de la Figura 5.7 81

5.6 Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.7 81

5.7 Medidas de aptitud de las imágenes de la Figura 5.9 84

5.8 Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.9 85

5.9 Medidas de aptitud de las imágenes de la Figura 5.11 89

5.10 Parámetros empleados en los algoritmos de mejoramiento de contraste para obtener las imágenes de la Figura 5.11 89

5.11 Medidas de aptitud de las imágenes de la Figura 5.12 91

5.12 Parámetros empleados en los algoritmos de mejoramiento de contraste para obtener las imágenes de la Figura 5.12 93

5.13 Medidas de aptitud de las imágenes de la Figura 5.13 93

5.14 Parámetros empleados en los algoritmos de mejoramiento de contraste para obtener las imágenes de la Figura 5.13 95

5.15 Parámetros empleados en el algoritmo de enmascarado de desenfoque para obtener las imágenes de la Figura 5.15 97

5.16 Medidas de aptitud de las imágenes de la Figura 5.15 97

5.17 Parámetros empleados por los algoritmos para obtener las imágenes de la Figura 5.17 101

Glosario

Adquisición

Es el conjunto de pasos requeridos para capturar una imagen mediante una cámara digital.

Contraste

Es la diferencia de tonos que hay entre las distintas zonas de la imagen. Los objetos en una imagen resultan visibles gracias a su diferencia de contraste respecto a los valores de los tonos que la rodean.

Electroforesis

La electroforesis es una técnica de separación de moléculas basada en su tamaño molecular y carga eléctrica. Ésta generalmente se utiliza con propósitos analíticos, pero puede ser una técnica preparativa para purificar moléculas parcialmente antes de aplicar procesos tales como: una espectroscopía de masas, una clonación o una secuenciación de ADN.

Frente de Pareto

Es un conjunto de puntos en un espacio multidimensional de funciones de aptitud de un algoritmo que describe las mejores configuraciones de éste.

Función de Aptitud

Una función de aptitud es un tipo particular de función objeto, que cuantifica que tan óptima es una solución brindada por un determinado algoritmo. Una función de aptitud ideal debe estar correlacionada con el objetivo del algoritmo.

Mejoramiento de Imágenes

Es un área del procesamiento digital de imágenes, cuyo objetivo principal es procesar una imagen para que el resultado sea más adecuado que la imagen original para una aplicación específica.

Nitidez

Describe nivel de definición de los detalles y contornos en una imagen.

Procesamiento Digital de Imágenes

El campo de procesamiento digital de imágenes se refiere al procesamiento de imágenes digitales mediante una computadora, en donde tanto la entrada como la salida son imágenes.

Ruido

Son píxeles con valores erróneos que se presentan aleatoreamente durante el proceso de adquisición de una imagen y transmisición de datos.

Lista de símbolos y abreviaciones

Notación general

$\mathbf{A}$	Matriz. $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$
$\mathcal{C}$	Conjunto.
$\mid \mathcal{C} \mid$	Cardinalidad del conjunto $\mathcal{C}$.
$\mathbb{R}$	Conjunto de los números reales.
$p(\cdot)$	Densidad de probabilidad
$\mathbf{x}$	Vector. $\mathbf{x} = [x_1\ x_2\ \ldots\ x_n]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$
y	Escalar
μ	Media de una distribución gaussiana.
σ	Desviación estándar de una distribución gaussiana.
σ^2	Varianza de una distribución gaussiana.
∇	Operador gradiente.
∇^2	Operador laplaciano.
ξ	Mediana de un conjunto de datos.

Procesamiento de imágenes

$A[\cdot]$	Transformación realizada por un algoritmo a una imagen.
$\mathbb{G}^2$	Rejilla bidimensional de una imagen $\mathbb{G}^2 = [0...M-1][0...N-1]$ para $< M, N > \in \mathbb{N}^+$
$\mathcal{I}$	Imagen.
$\widehat{\mathcal{I}}$	Imagen procesada por un algoritmo.
$\mathcal{M}$	Máscara.
$\mathcal{N}$	Imagen desgradada, ya sea con ruido, con un bajo contraste o bien una

	baja nitidez.
p	Píxel o elemento de imagen que pertenece a una rejilla $\mathbb{G}^2$
$\mathcal{V}$	Vecindario de un píxel.
r, s	Valores de niveles de gris de dos píxeles.
(i, j)	Par ordenado que denota la posición de un píxel, donde i es la fila y j la columna.

Evaluación

$A_{\mathbf{u}}$	Algoritmo A con parametrización $\mathbf{u}$.
$\widehat{\mathcal{P}}$	Frente de pareto.
$F(A_{\mathbf{u}}, \mathcal{G})$	Agregado de funciones de aptitud de un algoritmo A considerando el conjunto de datos de referencia $\mathcal{G}$.
$f_{ips}(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}})$	Imágenes procesadas por segundo.
$f_{ecmi}(A_{\mathbf{u}}, \mathcal{G})$	Error cuadrático medio inverso.
$f_{rme}(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}})$	Método de ruido escalar.
$f_{mcp}(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}})$	Mejoramiento de contraste promedio.
$f_e(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}})$	Entropía.
$f_d(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}})$	Difuminación.
$f_n(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}})$	Nitidez.
$\mathcal{G}$	Conjunto de datos de referencia $\{\langle \mathcal{I}_k, \mathcal{N}_k \rangle \mid k = 1...n\}$.
$\mathcal{RM}$	Ruido de método.
$\mathcal{I}_k$	Imagen de referencia.
$\mathcal{N}_k$	Versión degradada de la imagen de referencia $\mathcal{I}_k$.

Abreviaciones

AFLP	Polimorfismo de longitud de fragmentos amplificados (*Amplified Fragment Length Polymorphism*).
ALHS	Especificación Automática Local de Histograma (*Automatic Local Histogram Specification*).
CDF	Función de Densidad de Acumulada (*Commulative Density Function*).
DFT	Transformada Discreta de Fourier (*Discrete Fourier Transform*).
ECMI	Error cuadrático medio inverso.
HE	Ecualización de Histograma (*Histogram Equalization*).
HS	Especificación de Histograma (*Histogram Specification*).
IPS	Imágenes Procesadas por Segundo.
HS	Ecualización Local de Histograma (*Local Histogram Equalization*).
MCP	Mejoramiento de Contraste Promedio.
PDF	Función de Densidad de Probabilidad (*Probability Density Function*).
RM	Ruido de Método.
RME	Ruido de Método Escalar.

SUSAN	Núcleo de Asimilación del Segmento Univalor Menor (*Smallest Univalue Segment Assimilating Nucleus*).
UDWT	Transformada de Wavelet no Diezmada (*Undecimated Wavelet Transform*).
USAN	Núcleo de Asimilación del Segmento Univalor (*Univalue Segment Assimilating Nucleus*).

Capítulo 1

Introducción

El procesamiento digital de imágenes se define como la manipulación de imágenes mediante una computadora, en donde tanto la entrada como la salida son imágenes [21]. Ésta se divide en dos categorías: el mejoramiento de la calidad de imágenes y la ejecución de acciones o toma de decisiones con base en la información de la imagen. En éste sentido el presente trabajo se concentra en la primera de ellas. A continuación se enmarca el trabajo en una aplicación en particular que es la caracterización molecular de organismos, posteriormente se presenta la importancia de mejorar la calidad de las imágenes en general. Seguidamente, se plantea la solución al problema y finalmente se presenta el objetivo del trabajo así como la estructura de éste documento.

1.1 Caracterización molecular de organismos

El ADN de organismos se puede caracterizar molecularmente, ésto se puede efectuar mediante un análisis de imágenes de geles, que son el resultado de procesos de electroforesis. La electroforesis es una técnica de separación de moléculas basada en su tamaño molecular y carga eléctrica. Ésta generalmente se utiliza con propósitos analíticos, pero puede ser una técnica preparativa para purificar moléculas parcialmente antes de aplicar procesos tales como: una espectroscopía de masas, una clonación o una secuenciación de ADN. El gel se emplea como una matriz para separar las moléculas, éste generalmente es un polímero y la electroforesis se refiere a la fuerza electromotriz que se usa para desplazar las moléculas a través del gel. En la Figura 1.1 se muestra un ejemplo de una imagen de gel de electroforesis, obtenido mediante una técnica llamada *polimorfismo de longitud de fragmentos amplificados* (en inglés Amplified Fragment Length Polymorphism - AFLP)[3].

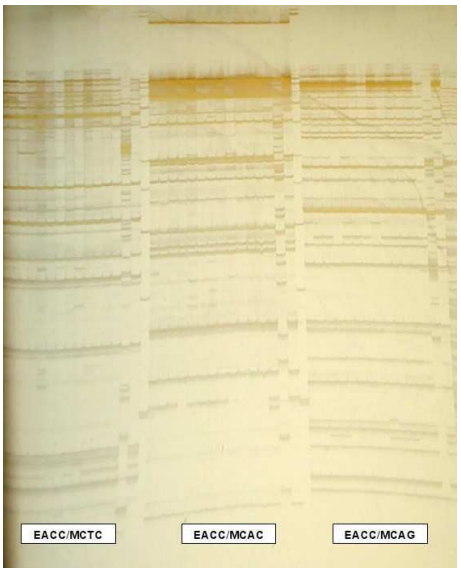


Figura 1.1: Ejemplo de imagen de gel de electroforesis [5] utilizando AFLP

En ésta imagen se aprecian carriles (columnas verticales) y bandas (pequeñas líneas horizontales en cada carril). Además, se aprecia que existe una distorsión geométrica y un bajo contraste que dificulta su interpretación. El análisis de las imágenes es realizado por biólogos moleculares manualmente e inicia con la binarización de la imagen, donde se determinan las bandas ausentes y las presentes, ésta información se representa de forma binaria. En la Figura 1.2 se muestra cómo una banda presente en la imagen de gel se representa con un uno binario y cómo una banda ausente se representa con un cero binario.

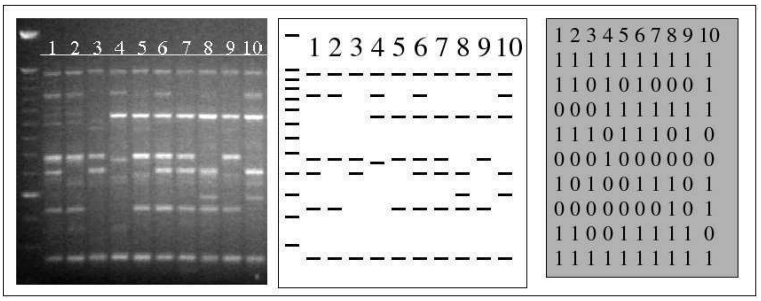


Figura 1.2: Proceso de binarización de gel [5]

Posteriormente, se obtiene una matriz que organiza y resume la información extraída. El objetivo del análisis posterior depende de la tarea. Como se ha mostrado, el análisis de estas imágenes está compuesto de varias etapas y se efectúa manualmente, por lo que errores humanos son introducidos por las limitaciones en la calidad de la imagen. Ésto pone en evidencia la necesidad de herramientas que permitan mejorar las imágenes

adquiridas de forma digital para analizarlas de manera automática.

El sistema propuesto de análisis automático de las imágenes de geles se muestra en la Figura 1.3. El primer elemento de este sistema es la adquisición de imágenes de geles mediante una cámara. Este proceso de digitalización de la imagen introduce imperfecciones a la imagen, y aunado a las imperfecciones intrínsecas de las imágenes de geles, hace necesaria una etapa de pre-procesamiento que reduzca el ruido, mejore el contraste, la nitidez y compense distorsiones geométricas de la imagen. Seguidamente la etapa de extracción de descriptores consiste en transformar cada carril en la imagen del gel, a una representación vectorial adecuada para el posterior análisis. El módulo de reconocimiento de patrones relaciona la imagen adquirida con meta-información, almacenada en una base de datos. La verificación evalúa utilizando la información disponible *a priori* si los datos que provee la etapa de reconocimiento de patrones tienen sentido o no. Una etapa con que se deberá contar es una interfaz para facilitar al experto su interacción con el sistema, en relación con la recolección, almacenamiento, manipulación, recuperación de información y acceso a la base de datos.

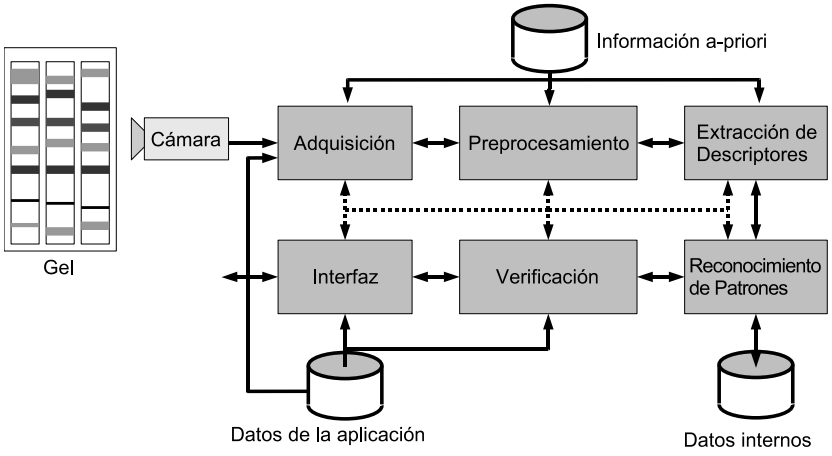


Figura 1.3: Diagrama de bloques del sistema automatizado de análisis de imágenes [5]

1.2 Problemática del procesamiento de imágenes con mala calidad

En el proceso de adquisición en sistemas de procesamiento digital de imágenes, la información digitalizada de primera mano posee imperfecciones. Éstas pueden provenir del elemento a ser digitalizado, como es el caso de las imágenes de los geles, que en sí mismas poseen imperfecciones como bajo contraste, baja nitidez y distorsiones geométricas de las bandas, que son problemas propios del objeto. Otro origen de imperfecciones es el

proceso de adquisición por medio de la cámara, que puede introducir ruido y afectar tanto el contraste como la nitidez de la imagen, ésto causado por una inadecuada configuración de la cámara, defectos físicos en ésta y errores en la transmisión de datos.

Motivos como los expuestos anteriormente sugieren que para un adecuado análisis automático de la imagen, se debe contar con una etapa de pre-procesamiento, que está compuesta por tres elementos: reducción de ruido, mejoramiento de contraste y mejoramiento de nitidez. Ésta etapa de pre-procesamiento es una necesidad común de sistemas de procesamiento digital de imágenes, que dependerá de las características propias de cada aplicación en particular. Así, para el caso del análisis de imágenes de geles de electroforesis, la etapa de pre-procesamiento además de contar con los tres elementos básicos, deberá contar con un módulo de compensación de distorsiones geométricas de las bandas. Dentro de éste contexto la problemática en general de una inadecuada calidad de las imágenes se sintetiza de la siguiente manera:

> *Una imagen digitalizada con un nivel de ruido, de contraste y de nitidez inadecuados, aumenta la inducción de error en su posterior análisis e interpretación.*

1.3 Solución propuesta para el mejoramiento de la calidad de imágenes

El presente trabajo se localiza en la etapa de preprocesamiento mostrada en el diagrama funcional de la Figura 1.3. Tal y como se discutió en la sección anterior ésta etapa para el caso particular del sistema de análisis automático de imágenes de geles de electroforesis está conformada por los siguientes módulos: reducción de ruido, mejoramiento de contraste, mejoramiento de nitidez y compensación de la distorsión geométrica de las bandas. Como se muestra en la Figura 1.4 éstos se pueden conjuntar en dos subetapas en donde la primera de ellas contiene los módulos de reducción de ruido, mejoramiento de contraste y mejoramiento de nitidez y la segunda contiene el módulo de compensación de distorsión geométrica.

El interés de éste trabajo está particularmente en la primera subetapa de la Figura 1.4, en donde se pretenden desarrollar módulos de software que permitan reducir el ruido, mejorar el contraste y la nitidez. El mejoramiento en la calidad de una imagen dependerá en conjunto de éstos tres elementos, tal y como se expresa conceptualmente en la Figura 1.5.

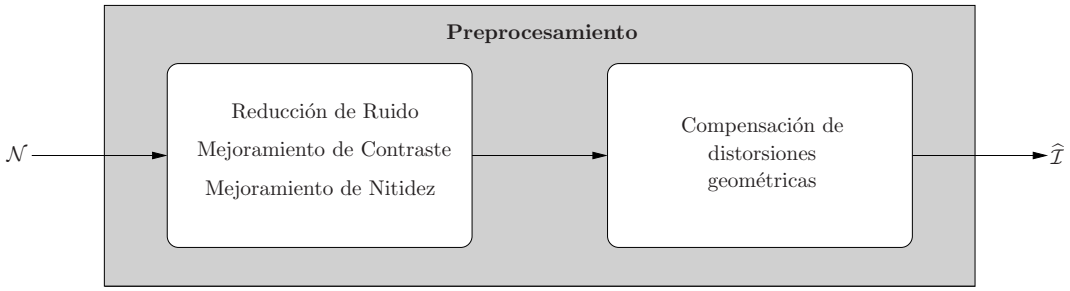


Figura 1.4: Etapa de preprocesamiento del sistema de análisis automático de imágenes de geles de electroforesis

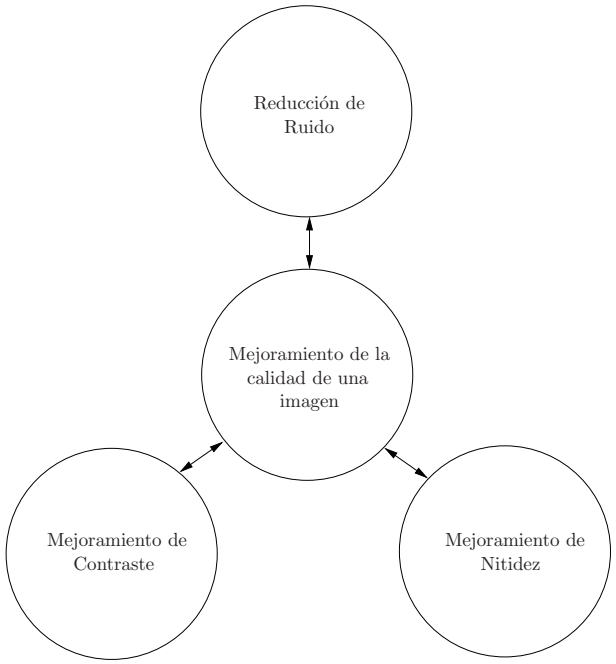


Figura 1.5: Elementos del mejoramiento de la calidad en una imagen

En principio la atención se concentra en efectuar una investigación bibliográfica en artículos que propongan métodos o algoritmos que permitan disminuir el ruido, mejorarar el contraste y la nitidez de una imagen, de ésta manera se elegirá y se implementará el mejor algoritmo para la mejora de cada parámetro.

Los algoritmos implementados formarán parte de una biblioteca de software para visión por computadora y procesamiento digital de imágenes llamada LTI-Lib con el objetivo de poner a libre disposición los algoritmos implementados; para mayor detalle de ésta biblioteca consulte el Apéndice A. Por otra parte, para poder evaluar de una manera objetiva el desempeño de cada uno de los algoritmos se hace necesario contar con medidas cuantitativas o tambíen llamadas *medidas o funciones de aptitud* de ruido, contraste y

nitidez, ya que, no basta con una comparación visual, que es muy subjetiva. Se pretende que los algoritmos que se implementen presenten mejores resultados que los ya existentes en la LTI-Lib, como es en el caso de reducción de ruido y mejoramiento de contraste. Por otra parte, en el caso de mejoramiento de nitidez, la biblioteca no cuenta actualmente con este tipo de algoritmos.

Finalmente, para valorar el desempeño de los algoritmos implementados de una manera integral con las medidas de aptitud encontradas para cada elemento de calidad de imagen, se hará uso de la evaluación por medio de Frentes de Pareto que permiten un análisis multi-objetivo, es decir, consideran a la vez varias medidas de aptitud que compiten entre sí. Ésto permitirá evaluar cada algoritmo todas sus parametrizaciones, que hace que la evaluación sea más robusta que una simple comparación con un solo conjunto de parámetros. Así, el Frente de Pareto además de comparar todas las parametrizaciones de un algoritmo para tener un conocimiento completo del comportamiento del algoritmo, permite hacer una comparación objetiva entre diferentes algoritmos. Un algoritmo genético se empleará aquí para encontrar el Frente de Pareto.

1.4 Objetivos y estructura de éste trabajo

El objetivo principal de este trabajo es mejorar la calidad de imágenes digitalizadas de geles de electroforesis, disminuyendo el ruido, mejorando el contraste y la nitidez de las mismas permitir lograr el análisis automático de ellas. Para lograr ésto se implementarán tres algoritmos: uno para reducir el ruido, uno para mejorar el contraste y otro para mejorar la nitidez en una imagen. Finalmente, se determinarán medidas aptitud de ruido, contraste y nitidez en una imagen que permitan evaluar objetivamente mediante el Frente de Pareto los algoritmos implementados.

En el Capítulo 2 se presentan conceptos generales de procesamiento digital de imágenes y del área de mejoramiento de imágenes. Además, se hará una revisión sobre trabajos previos de reducción de ruido, mejoramiento de contraste y de nitidez en imágenes digitalizadas. Ésto permitirá conocer los diferentes caminos que se han recorrido en la implementación de algoritmos de mejoramiento de la calidad en imágenes y así mismo conocer cuáles son los que han presentado mejores resultados. En el Capítulo 3 se presenta el método de evaluación del rendimiento de los algoritmos, además del concepto de medidas de aptitud, que son las medidas cuantitativas que medirán el nivel de ruido, de contraste y de nitidez en imágenes digitalizadas. En el Capítulo 4 se presentan en detalle los algoritmos y su implementación; en el Capítulo 5 se presentan los resultados obtenidos con su correspondiente análisis. Finalmente, la conclusiones y recomendaciones para el trabajo futuro se encuentran en el Capítulo 6.

Capítulo 2

Marco teórico

Como punto de partida de éste trabajo en éste capítulo se presentarán conceptos fundamentales de procesamiento de imágenes, así como conceptos relacionados particularmente con el área de mejoramiento de imágenes. Posteriormente, se presentan diferentes trabajos que han sido realizados en relación con la reducción del nivel de ruido, mejoramiento de contraste y mejoramiento de nitidez en imágenes, ésto con el fin de conocer diferentes enfoques que se han desarrollado y así determinar entre ellos cuál es la alternativa idónea para ser implementada en cada uno de los tres casos.

2.1 Fundamentos de procesamiento digital de imágenes

2.1.1 Conceptos generales

Antes de presentar los conceptos relacionados con el área de mejoramiento de imágenes se definen conceptos básicos de procesamiento digital de imágenes, éstos son fundamentales para la adecuada compresión de éste trabajo.

Definición 2.1 (Elemento de imagen) *Un elemento de imagen o píxel p, se define como $p = (i, j) \in \mathbb{G}^2$.*

$\mathbb{G}^2$, es una rejilla bidimensional, finita, discreta y compacta que se define como $\mathbb{G}^2 = [0...M-1] \times [0...N-1]$, con $M, N \in \mathbb{N}^+$. En la Figura 2.1 se muestra la rejilla $\mathbb{G}^2$.

Definición 2.2 (Imagen digital) *Una imagen $\mathcal{I}$ es una función que se denota como*

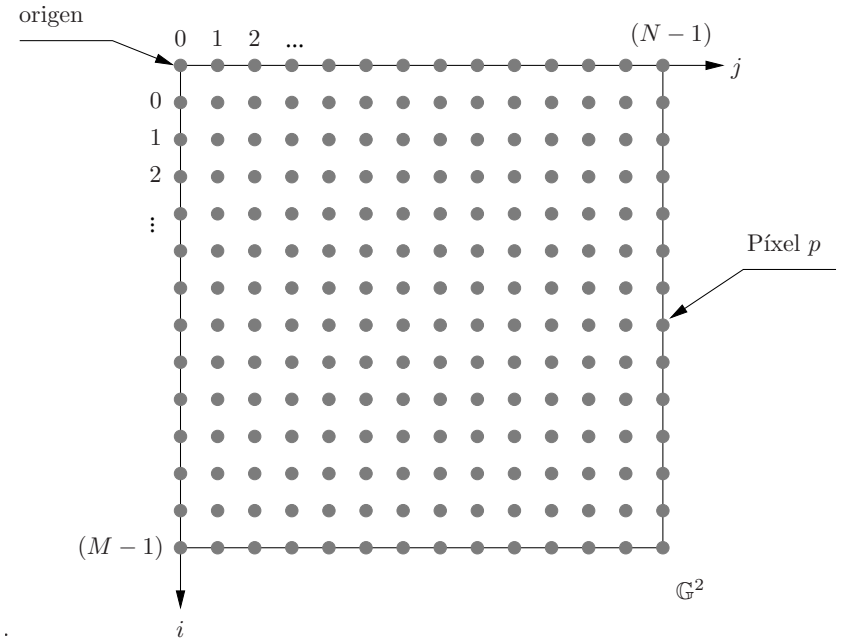


Figura 2.1: Rejilla $\mathbb{G}^2$

$\mathcal{I}(p)$ *o* $\mathcal{I}(i,j)$*, tal que* $\mathcal{I} : \mathbb{G}^2 \to \mathbb{R}$.

Otro concepto de procesamiento digital de imágenes, y de interés para el área de mejoramiento de imágenes es el *vecindario de un píxel* que se define de la siguiente manera:

Definición 2.3 (Vecindario de un píxel) *Un píxel* p *con coordenadas* (i,j) *tiene vecinos verticales y horizontales con las siguientes coordenadas*

$$(i+1,j),(i-1,j),(i,j+1),(i,j-1)$$

Éste conjunto de píxeles se le llama vecindad de 4 y se denota como $\mathcal{V}_4(p)$*. Los cuatro píxeles diagonales de* p *tienen las siguientes coordenadas*

$$(i+1,j+1),(i+1,j-1),(i-1,j+1),(i-1,j-1)$$

y éste conjunto se denota como $\mathcal{V}_D(p)$*. Éstos puntos junto con la vecindad de 4 de* p *forman un conjunto llamado vecindad de 8 de* p *y se denota como* $\mathcal{V}_8(p)$*.*

2.1.2 Mejoramiento de imágenes

La intensión de ésta sección es definir los conceptos básicos de los principales ámbitos en los que se desarrollan los algoritmos de mejoramiento de calidad de imágenes. El

mejoramiento de imágenes es el interés de éste trabajo, su objetivo principal es procesar una imagen para que el resultado sea más adecuado que la imagen original, de acuerdo a la aplicación de interés.

El mejoramiento de imágenes se divide en general en tres categorías[27]: dominio espacial, dominio de la frecuencia, y el más reciente, el dominio de la transformada wavelet (onditas). Una vez definidos éstos tres dominios se presenta el modelo del proceso de degradación y mejoramiento de imágenes.

Dominio Espacial

El presente trabajo se concentra enteramente en éste dominio. El término dominio espacial se refiere al plano de la imágen en sí mismo y los algoritmos basados en éste operan directamente sobre los píxeles, y se denotan mediante la siguiente expresión

$$\widehat{\mathcal{I}}(i,j) = A[\mathcal{I}(i,j)] \tag{2.1}$$

donde $\mathcal{I}(i,j)$ es la imagen de entrada, $\widehat{\mathcal{I}}(i,j)$ es la imagen procesada y A es un operador en $\mathcal{I}$, definido sobre algún vecindario de (i,j). Tal y como se describió en la Definición 2.3 un vecindario es una subimagen cuadrada o rectangular centrada en (i,j), como se muestra en la Figura 2.2.

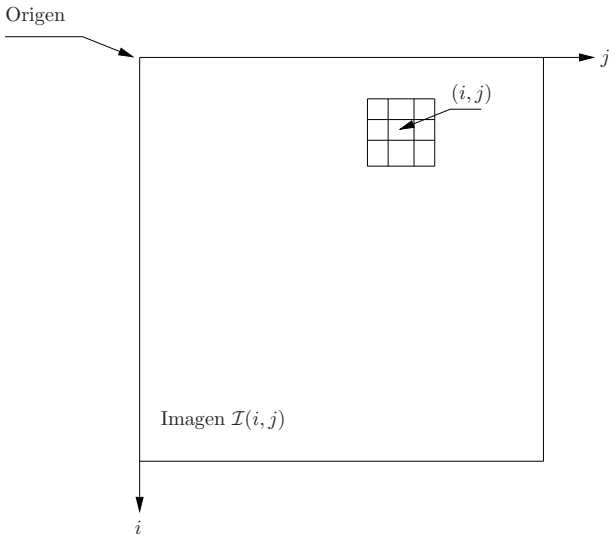


Figura 2.2: Vecindario de 3x3 de un píxel (i,j) en una imagen

El centro de la subimagen se mueve através de todos los píxeles de la imagen. El operador A es aplicado a cada posición (i,j) para la salida, $\widehat{\mathcal{I}}$ en ese punto. La forma más simple

de A es cuando el vecindario de $\mathcal{I}$ tiene un tamaño de 1×1. En éste caso, $\widehat{\mathcal{I}}$ depende solo del valor de $\mathcal{I}$ en (i, j), y A se convierte en una función de transformación de intensidad (o nivel de gris) de la forma

$$s = A(r) \tag{2.2}$$

donde, r y s son variables que denotan respectivamente, la intensidad o el nivel de gris de $\mathcal{I}(i, j)$ y $\widehat{\mathcal{I}}(i, j)$ en cualquier punto (i, j). Vecindarios más grandes permiten una mayor flexibilidad. En general, el proceso consiste en aplicar una función a los valores de $\mathcal{I}$ en un vecindario determinado de (i, j) para determinar el valor de $\widehat{\mathcal{I}}$ en (i, j).

El principal ejemplo del proceso descrito anteriormente es el uso de las llamadas máscaras, conocidas también como filtros, ventanas o núcleos (en inglés *masks, filters, windows, kernels*, respectivamente). Una máscara $\mathcal{M}$ es un pequeño arreglo en dos dimensiones centrado en el píxel (i, j), en donde los valores de los coeficientes de la máscara determinan la naturaleza del proceso. Las técnicas de mejoramiento de imágenes que se basan en éste enfoque se conoce como *procesamiento de máscara* o de *filtrado*.

El concepto de filtrado proviene del uso de la transformada de Fourier en procesamiento de señales en el llamado *dominio de la frecuencia* [2, 26, 29]. Se emplea el término de *filtrado espacial* para diferenciar este proceso del filtrado tradicional en el dominio de la frecuencia.

Dos de los tipos más comunes de filtros espaciales son [21]: los lineales y los no lineales, en donde ambos recorren todos los píxeles de una imagen con sus correspondientes vecindarios, pero la diferencia está en cómo aplican el filtrado a la imagen.

La respuesta del filtrado espacial *lineal* está dada por convolución de la imagen con una máscara. Ésto se calcula mediante la suma de los productos de los coeficientes del filtro con sus correspondientes píxeles en el área abarcada por la máscara. En general, el filtrado lineal de una imagen $\mathcal{I}(i, j)$ de tamaño $M \times N$, con una máscara $\mathcal{M}(i, j)$ de tamaño $m \times n$ está dado por la expresión

$$\begin{aligned} \widehat{\mathcal{I}}(i, j) &= \mathcal{I}(i, j) * \mathcal{M}(i, j) \\ \widehat{\mathcal{I}}(i, j) &= \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t)\mathcal{I}(i - s, j - t) \end{aligned} \tag{2.3}$$

donde $a = (m - 1)/2$ y $b = (n - 1)/2$. Para generar un filtrado completo de la imagen ésta ecuación debe ser aplicada para $i = 0, 1, 2, ..., M - 1$ y $j = 0, 1, 2, ..., N - 1$, así se asegura que la máscara procesa todos los píxeles de una imagen. El proceso de filtrado lineal se ilustra en la Figura 2.3.

El segundo tipo de filtrado es el filtrado espacial *no lineal*, que opera en los vecindarios, y el proceso de recorrido através de los píxeles de una imagen es igual al caso lineal.

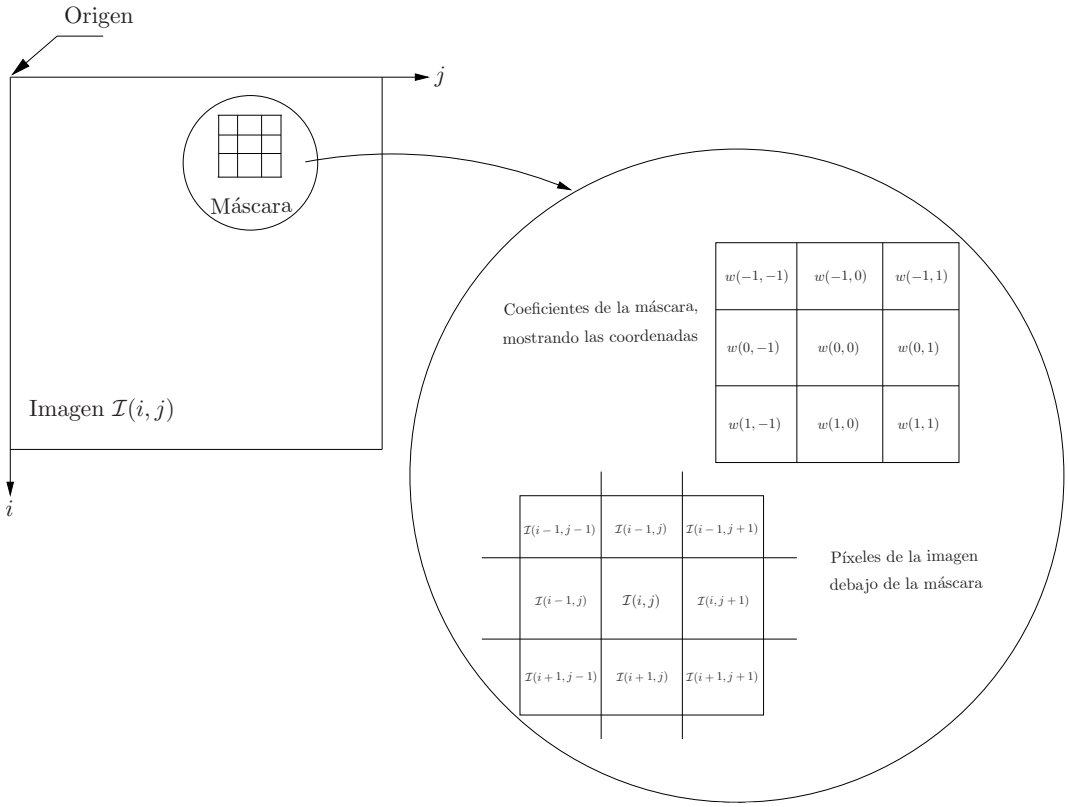
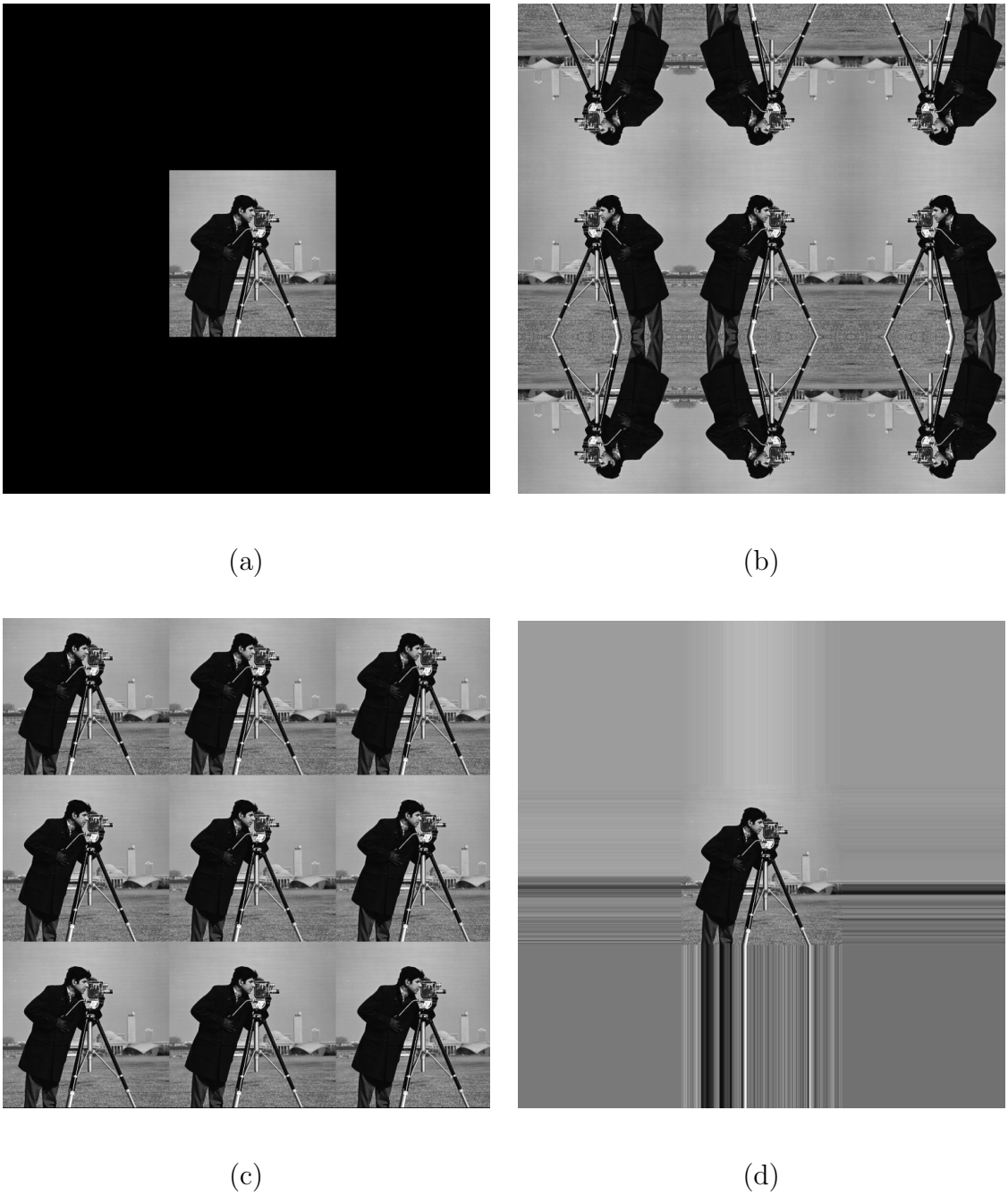


Figura 2.3: Proceso de filtrado lineal espacial

Sin embargo, la diferencia radica en que la operación de filtrado está condicionada a los valores de los píxeles en el vecindario en consideración.

Con respecto a la implementación de los filtros espaciales surge una pregunta: ¿Qué sucede cuando una máscara tiene como píxel central uno que se encuentra en el borde de la imagen o cerca, y por tanto algunos de los píxeles que abarca ésta no se encuentran definidos porque no pertenecen a la imagen?. En éste caso el enfoque empleado en éste trabajo consiste en expandir la frontera de la imagen y rellenar esa área mediante varias alternativas. La primera de ellas consiste asignar un valor de cero a esa área expandida tal y como se ilustra en la Figura 2.4(a). Otro tipo de expansión se basa en reflejar la imagen en cada uno de los bordes como se muestra en 2.4(b). El tercer tipo de expansión consiste en repetir periódicamente la imagen como se ilustra en la Figura 2.4(c). Finalmente, en la Figura 2.4(d) se muestra el último tipo de expansión que consiste es repetir de manera constante el valor del píxel del borde. Éstas diferentes alternativas de fronteras son empleadas de acuerdo con la naturaleza de la máscara que se le aplique a la imágen.



(a) (b)

(c) (d)

Figura 2.4: Tipos de expansiones de las fronteras de una imagen: (a) cero, (b) reflejada, (c) periódica, (d) constante.

Dominio de la Frecuencia

Las técnicas de procesamiento en el dominio de la frecuencia se basan en la la transformada de Fourier de una imagen [2, 26, 29]. El interés de éste trabajo se concentra en la transformada discreta de Fourier (en inglés Discrete Fourier Transform - DFT) que es la base del estudio del dominio de la frecuencia en procesamiento digital de imágenes. La transformada discreta de Fourier en dos dimensiones de una función $f(x, y)$ (o bien una imagen $\mathcal{I}(i, j)$) de tamaño $M \times N$ está dada por la siguiente ecuación

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)} \tag{2.4}$$

Ésta expresión debe ser calculada para los valores de $u = 0, 1, 2, ..., M - 1$ y también para $v = 0, 1, 2, ..., N - 1$. Similarmente, dada $F(u, v)$, se obtiene $f(x, y)$ mediante la transformada inversa de Fourier, mediante la siguiente expresión

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{-j2\pi(ux/M+vy/N)} \tag{2.5}$$

para $x = 0, 1, 2, ..., M - 1$ y $y = 0, 1, 2, ..., N - 1$. Las ecuaciones 2.4 y 2.5 conforman la *pareja de transformadas discretas bidimensionales de Fourier*. Las variables u y v son las *variables de frecuencia*. y x y y son las *variables espaciales*.

- **Filtrado en el Dominio de la Frecuencia**

El domino de la frecuencia es más que el espacio definido por los valores de la transformada de Fourier y sus variables de frecuencia (u, v), a continuación se presenta el significado del dominio de la frecuencia para el procesamiento de imágenes.

A partir de la información de la imagen en el dominio de la frecuencia se puede interpretar de manera general características espaciales ésta. La frecuencia se asocia directamente con la tasa de cambio, y así se puede asociar frecuencias con características de la intensidad de una imagen. Así, la componente de frecuencia más baja $u = v = 0$ corresponde al valor promedio de la intensidad o nivel de gris en una imagen. Al desplazarse en el espectro de frecuencias de una imagen desde el origen hacia las altas frecuencias, se encuentra que las bajas frecuencias son las responsables de regiones planas en donde no hay gran variación de el nivel de gris. Por otra parte, hacia las altas frecuencias se encuentran cambios rápidos en el nivel de gris de una imagen. Éstos cambios rápidos pueden ser bordes de objetos y otros componentes caracterizados por cambios abruptos en el nivel de gris como el ruido.

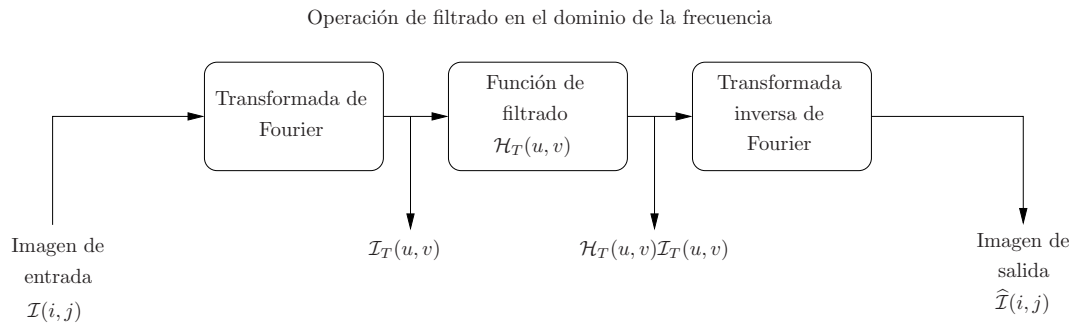


Figura 2.5: Diagrama de bloques del proceso de filtrado en el dominio de la frecuencia [21].

Sea $\mathcal{I}(i,j)$ una imagen y $\mathcal{I}_T(u,v)$ su correspondiente transformada de Fourier. En [21] se propone el siguiente esquema de filtrado en la frecuencia:

1. Calcular $\mathcal{I}_T(u,v)$, la DFT de la imagen.
2. Multiplicar $\mathcal{I}_T(u,v)$ por una función de *filtrado* $\mathcal{H}_T(u,v)$.
3. Calcular la DFT inversa del resultado en (2).
4. Obtener la parte real del resultado en (3).

La función $\mathcal{H}_T(u,v)$ es llamada filtro porque éste suprime ciertos componentes de frecuencia mientras que otros no se alteran. Los tipos de filtros más comunes empleados en procesamiento digital de imágenes son: los *filtros paso bajo* que atenúan las altas frecuencias, y los *filtros paso alto* que atenúan las bajas frecuencias. Es así como que una imagen procesada con un filtro paso bajo presenta menor detalle y luce más difuminada, en contraposición con una imagen procesada con un filtro paso alto que presenta mayor detalle y definición.

Sea $\mathcal{I}(i,j)$ representa la imagen de entrada y $\mathcal{I}_T(u,v)$ es su transformada, entonces la transformada de Fourier de la imagen de salida está dada por

$$\widehat{\mathcal{I}}_T(u,v) = \mathcal{H}_T(u,v)\mathcal{I}_T(u,v) \tag{2.6}$$

La imagen filtrada es obtenida simplemente tomando la transformada inversa de $\widehat{\mathcal{I}}_T(u,v)$

$$\widehat{\mathcal{I}}(i,j) = \mathcal{F}^{-1}[\widehat{\mathcal{I}}_T(u,v)] \tag{2.7}$$

El procedimiento de filtrado comentado anteriormente se resume en el diagrama de bloques de la Figura 2.5, en el que se sintetiza el concepto del filtrado como un proceso que se basa

en la modificación de la transformada de una imagen, mediante alguna función de filtrado, y posteriormente se toma el inverso del resultado para obtener la imagen de salida final.

La relación fundamental entre el domino de espacial y el dominio de la frecuencia es el teorema de la convolución. Éste concepto ya se introdujo anteriormente en el caso del dominio espacial, cuando se describió el proceso mediante el cual una máscara se mueve através de todos los píxeles de una imagen, y así calcula una cantidad predeterminada para cada píxel. Sea $F(u, v)$ y $H(u, v)$, las transformadas de Fourier de $f(x, y)$ y $h(x, y)$, respectivamente. El primer término del teorema de la convolución es $f(x, y) * h(x, y)$, y el segundo término es la multiplicación de las transformadas $F(u, v)H(u, v)$. La convolución en el dominio espacial se define formalmente como

$$f(x, y) * h(x, y) \Leftrightarrow \frac{1}{2\pi} F(u, v)H(u, v) \tag{2.8}$$

La doble flecha es usada para indicar que la expresión de la izquierda que es la convolución espacial puede ser obtenida tomando la transformada inversa de la transformada de Fourier de la expresión de la derecha que es el producto $F(u, v)H(u, v)$. Un resultado análogo es la convolución en el dominio de la frecuencia, que es simplemente una multiplicación en el dominio espacial

$$f(x, y)h(x, y) \Leftrightarrow F(u, v) * H(u, v) \tag{2.9}$$

Dominio de la transformada Wavelet

La transformada de wavelet es una técnica reciente de transformación utilizada en el procesamiento de imágenes. A diferencia de la transformada de Fourier en donde las funciones base son ondas senoidales, la transformada de wavelet se basa la descomposición lineal en ondas llamadas *wavelets* u *onditas*, de frecuencia variable y duración limitada. Un wavelet u ondita $\psi(x)$ es un tipo de función matemática utilizada para dividir una función dada en diferentes componentes de frecuencia y estudia cada componente con una resolución que se ajuste a su escala.

Los wavelets son copias escaladas y trasladadas conocidas como *wavelets hijas* de una forma de onda oscilatoria de longitud finita llamada *wavelet madre*. La transformada de wavelet tiene ventajas sobre la transformada de Fourier para representar funciones que tienen discontinuidades y picos, y para destruir o reconstruir señales finitas, no periódicas y/o no estacionarias.

- **Análisis Multiresolucional (MRA)**

Los wavelets fueron primero presentados como el fundamento de un nuevo enfoque para el procesamiento de señales llamado teoría *multiresolucional*. Como su nombre lo indica, la teoría multiresolucional se relaciona con señales (o imágenes en éste caso) en más de una resolución. Ésto con el objetivo de detectar características de las señales que en una resolución son más fáciles de detectar que en otras. El análisis multiresolucional se fundamenta principalmente en tres técnicas de imagen: *pirámide de imágenes, codificación subbanda* y la *transformada de Haar*.

Pirámide de imágenes Es una estrutura conceptualmente simple para representar imágenes con más de una resolución es la pirámide de imágenes. Como se muestra en la Figura 2.6 la base de la pirámide contiene la representación de mayor resolución de la imagen que está siendo procesada. Al moverse hacia los niveles superiores de la pirámide tanto el tamaño como la resolución de la imagen decrecen.

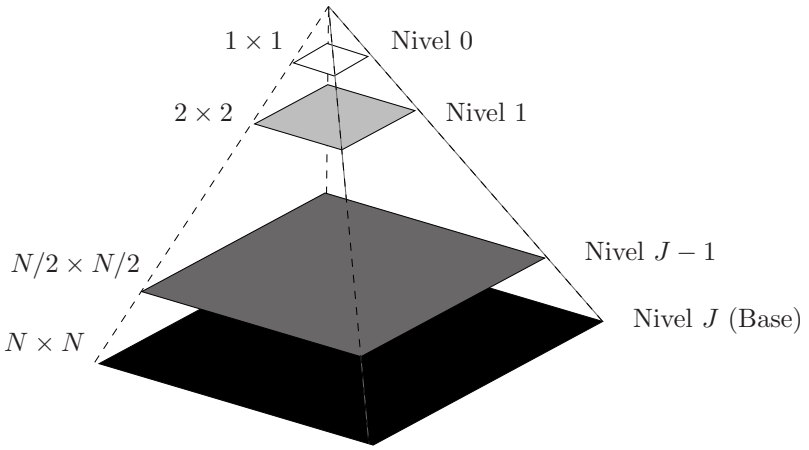


Figura 2.6: Estrutura piramidal de imágenes [21].

Codificación subbanda En la codificación subbanda, una imagen es descompuesta en un conjunto de componentes de banda limitada, llamadas *subbandas*, que pueden ser reensambladas para reconstruir la imagen original sin error. Cada subbanda es generada por el filtro pasabanda que filtra la entrada. Debido a que el ancho de banda de las subbandas resultantes es más pequeño que el de la imagen original, las subbandas pueden ser submuestreadas sin perder información. La reconstrucción de la imagen original se logra sobremuestreando, filtrando y sumando las subbandas individuales.

La transformada de Haar Ésta es la tercera y última de las técnicas de imagen que se relaciona con el análisis multiresolucional. La importancia de ésta transformada

radica en el hecho de que sus funciones base son los wavelets ortonormales conocidos más viejos y simples. La transformada de Haar, es tanto separable como simétrica y puede ser expresada en forma de matriz como

$$\mathbf{T} = \mathbf{HFH} \tag{2.10}$$

donde $\mathbf{F}$ es una matriz de la imagen de tamaño $N \times N$, $\mathbf{H}$ es una matriz de transformación de tamaño $N \times N$. Para la transformada de Haar, la matriz de transformación $\mathbf{H}$ contiene las funciones base de Haar, $h_k(z)$. Ellas se definen sobre un intervalo continuo y cerrado $z \in [0, 1]$, para $k = 0, 1, 2, ..., N-1$, donde $N = 2^n$. Para generar $\mathbf{H}$, se define un entero k, tal que $k = 2^p + q - 1$, donde $0 \leq p \leq n-1$, $q = 0$ o 1 para $p = 0$, y $1 \leq q \leq 2^p$ para $p \neq 0$. Entonces las funciones base de Haar son

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, \quad z \in [0, 1] \tag{2.11}$$

y

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & (q-1)/2^p \leq z \leq (q-0.5)/2^p \\ -2^{p/2} & (q-0.5)/2^p \leq z \leq q/2^p \\ 0 & \textit{de otra manera}, \ z \in [0, 1] \end{cases} \tag{2.12}$$

Para un mayor detalle del dominio wavelet aplicado a procesamiento de imágenes refiérase a [21].

Proceso de degradación y mejoramiento de imágenes

Como se muestra en la Figura 2.7, el proceso de degradación es modelado como una función de degradación que representa los factores de desmejoramiento de una imagen, como errores en el proceso de adquisión, mala iluminación al momento de la captura de la imagen y otros factores más, que resultan en una imagen con inadecuado contraste y nitidez. Lo anterior junto con el ruido aditivo operan en la imagen de entrada deseada idealmente $\mathcal{I}(i, j)$ para producir una imagen degradada $\widehat{\mathcal{I}}(i, j)$. Teniendo algún conocimiento de la función de degradación $\mathcal{H}$, y del ruido aditivo $\eta(i, j)$, es posible obtener una imagen estimada $\hat{\mathcal{I}}(i, j)$ que se acerque lo más posible a la imagen deseada, éste es precisamente el objetivo principal del mejoramiento de imagen y a su vez de éste trabajo.

Basado en los conceptos discutidos en la sección anterior sobre el dominio espacial y de la frecuencia, junto con el concepto de convolución se puede modelar en el *dominio espacial* el proceso de degradación de una imagen de la siguiente manera

$$\widehat{\mathcal{I}}(i, j) = \mathcal{I}(i, j) * \mathcal{H}(i, j) + \eta(i, j) \tag{2.13}$$

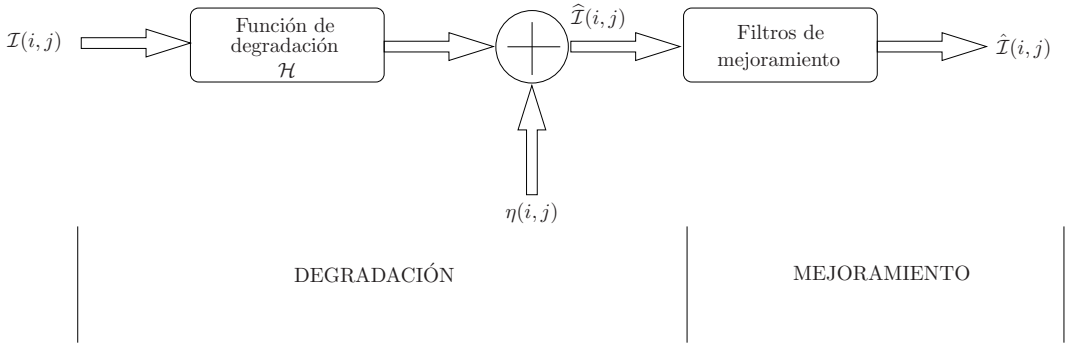


Figura 2.7: Proceso de degradación y mejoramiento de imágenes [21].

donde $\mathcal{H}(i,j)$ es la representación espacial de la función de degradación y el símbolo $*$ indica la convolución entre la imagen de entrada y la función de degradación. Basado en el teorema de la convolución se sabe que una convolución en el dominio del espacio es una multiplicación en el domino de la frecuencia, de tal manera que el proceso de degradación se modela en el dominio de la frecuencia de la siguiente manera

$$\widehat{\mathcal{I}}_T(u,v) = \mathcal{I}_T(u,v)\mathcal{H}_T(u,v) + N(u,v) \tag{2.14}$$

donde los términos en mayúsculas son las transformadas de Fourier de los correspondientes términos de (2.13). Por otra parte, los filtros de mejoramiento son los algoritmos que se desarrollan ya sea en el dominio espacial, dominio de la frecuencia o dominio de wavelet. El presente trabajo se concentra completamente en el dominio espacial.

2.2 Trabajos previos en mejoramiento de imágenes

2.2.1 Reducción de ruido

El proceso de adquisición de imágenes mediante sensores y cámaras generalmente contamina las imágenes con imperfecciones como el ruido. Defectos en instrumentos, problemas con el proceso de adquisición, ruido térmico, así como errores de transmisión y compresión degradan la calidad de las imágenes digitalizadas, de tal manera que el primer paso antes de analizar las imágenes adquiridas es aplicar técnicas de reducción de ruido.

La forma en que el ruido se presenta en las imágenes digitales se ve influenciada por los instrumentos de captura, la transmisión de datos y la cuantización de la imagen, entre otros y según ésta así será la técnica a emplear para reducir el ruido. En general, el ruido en imágenes naturales se asume como aditivo y aleatorio, y es modelado como Gausiano

y blanco, y éste es en el que se enfoca principalmente el presente trabajo. En la Sección 4.1.1 se presentan con detalle modelos comunes de ruido en procesamiento de imágenes.

Evolución de la investigación en reducción de ruido en imágenes

La reducción de ruido sigue siendo un problema fundamental en el área de procesamiento de imágenes. Una de las principales preocupaciones que concentra los esfuerzos de la mayor parte de los trabajos es reducir el ruido sin afectar la imagen original, debido a que muchos enfoques de reducción de ruido tienden a difuminar las imágenes. Por otra parte, al comparar los algoritmos desarrollados para la reducción de ruido en una imagen con los de mejoramiento de contraste y de definición, se encuentra que los esfuerzos de investigación se han concentrado mayormente en la reducción de ruido.

En general los wavelets brindan un rendimiento superior dedido a las propiedades de multiresolución comentadas en la sección anterior. Con la Transformada de Wavelet ganando popularidad en los últimos años, muchos algoritmos basados en ésta han sido propuestos. De tal manera que el interés en la investigación ha pasado del dominio espacial y de Fourier al dominio de la Transformada de wavelet. Desde que Donoho en [17] propuso un enfoque de umbral para reducción de ruido, se han publicado gran cantidad de artículos en éste sentido. A pesar, de que el concepto propuesto por Donoho [18] no fue revolucionario, demostró un enfoque simple a un problema difícil [27].

Lo anterior no quiere decir que se ha dejado de lado el desarrollo de algoritmos en el dominio espacial y de la frecuencia. El enfoque propuesto por Buades en [9] en el 2004, llamado *medias no locales* (en inglés Non-Local Means) es un algoritmo de reducción de ruido desarrollado en el dominio espacial que aprovecha la redundancia en las imágenes, y su principal virtud es el no afectar la imagen original para eliminar el ruido; éste enfoque es de especial interés para el presente trabajo.

Clasificación de los algoritmos de reducción de ruido

Como se muestra en la Figura 2.8, existen dos enfoques básicos para reducción de ruido en una imagen: *métodos en el dominio espacial* y *métodos en el dominio de las transformadas* [27].

- **Filtrado en el Dominio Espacial**

Una forma tradicional para remover ruido de una imagen es emplear filtros espaciales. Como se discutió en la sección anterior, éstos filtros pueden ser clasificados en lineales y no lineales.

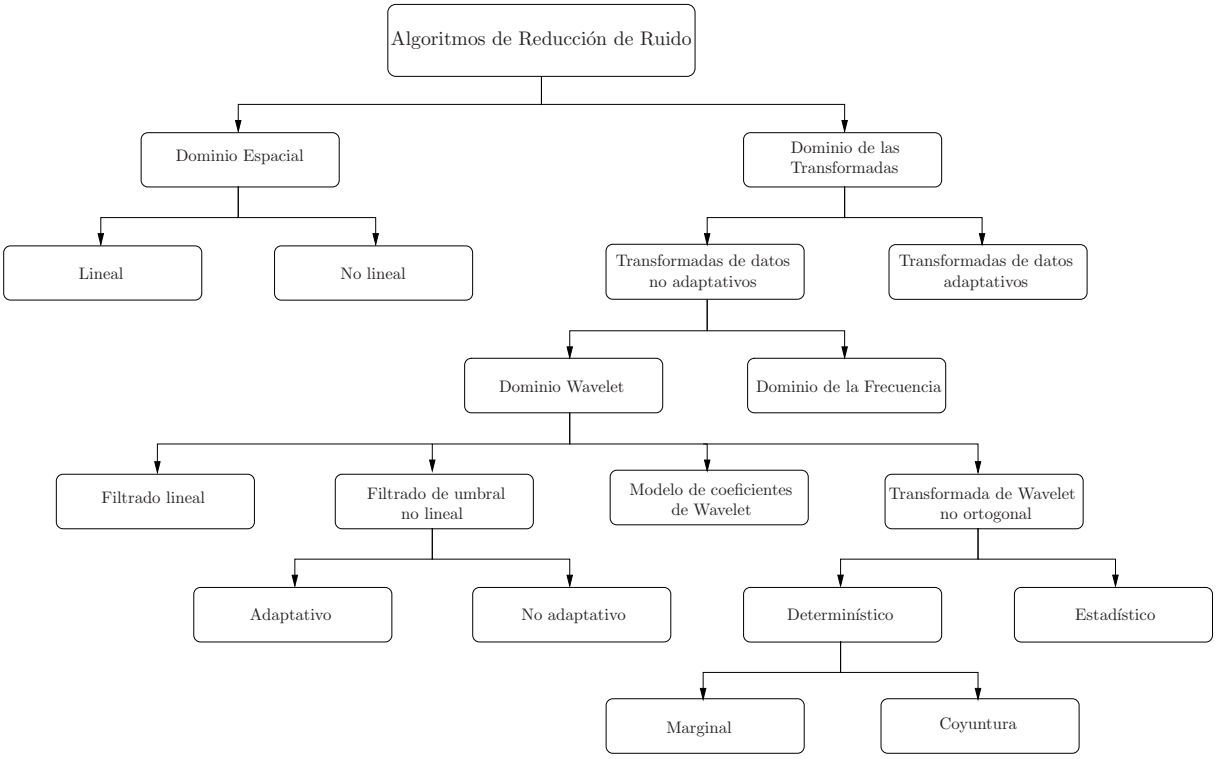


Figura 2.8: Clasificación de algoritmos de reducción de ruido [27].

Filtros lineales En general los filtros lineales tienden a difuminar las imágenes, afectando los bordes, líneas y detalles finos. Un filtro de media es el filtro lineal óptimo para ruido Gaussiano, en el sentido de error cuadrático medio. La idea de éste filtro es sustituir el valor de cada píxel, por el promedio de los valores de los vecinos, definidos por la máscara. El filtro Gaussiano es otro filtro lineal, que convuciona una máscara gaussiana con la imagen para reducir el ruido; éste se describe en la Sección 4.1.2.

Filtros no lineales En éste tipo de filtros el ruido se remueve sin identificarlo. Los filtros espaciales utilizan filtros pasobajo, en grupos de píxeles definidos por las máscaras, asumiendo que ocupan la parte alta del espectro de frecuencia. El filtro de mediana es un ejemplo de filtro no lineal en donde el valor de píxel en estudio es sustituido con la mediana del vecindario definido en la máscara; éste filtro se describe en la Sección 4.1.3. Otro enfoque que evita el difuminado de la imagen es el propuesto por Buades en [9, 10, 11, 12]. Éste algoritmo se basa en el promediado no local de todos los píxeles de una imagen; en la Sección 4.1.5 se describe detalladamente éste algoritmo.

- **Filtrado en el Dominio de las Transformadas**

Los métodos de filtrado en el dominio de las transfomadas pueden ser clasificados de acuerdo a la elección de las funciones base. Las funciones base pueden ser datos adaptativos y no adaptativos.

Filtrado en la frecuencia En algoritmos del dominio de la frecuencia, la reducción de ruido se logra mediante el diseño de un filtro con una frecuencia de corte determinada, cuando el ruido se encuentra descorrelacionado de la imagen original. El ruido se encuentra en las altas frecuencias, por tal motivo, que el tipo de filtros que se utilizan son pasobajo. Los filtros se implementan mediante la *Transformada Rápida de Fourier* (en inglés Fast Fourier Transform - FFT). Éstos métodos podrían causar frecuencias artificales en la imagen procesada.

Dominio Wavelet El filtrado en el dominio Wavelet se divide en dos categorías: métodos lineales y no lineales.

I. Filtros Lineales Los filtros lineales en el dominio wavelet proveen resultados óptimos cuando el ruido, puede ser modelado como un proceso Gaussiano y el criterio de exactitud es el error cuadrático medio. Sin embargo, el diseño de filtros basado en éste criterio frecuentemente resulta en una imagen que es más desagradable que la imagen original con ruido, a pesar de ésto el proceso de filtrado reduce exitosamente el error cuadrático medio.

II. Filtros de Umbral no Lineales El área más investigada de la transformada wavelet aplicada a la reducción de ruido es la de los métodos basados en el umbral de coeficientes no lineales. Éste procedimiento aprovecha que la transformada de wavelet mapea ruido blanco del dominio de la señal a ruido blanco en el dominio de la transformada. Así, mientras que la energía de la señal se concentra más en pocos coeficientes en el dominio de la transformada, el ruido no lo hace. Éste es un importante principio para separar el ruido de la señal de interés [17].

El procedimiento descrito anteriormente, en donde los pequeños coeficientes son removidos, mientras que otros se mantienen se llama *umbral fuerte*. Pero genera desperfectos en las imágenes como resultado de un fracasado intento de remover los coeficientes de ruido moderamente grandes.Para superar éste inconveniente, en [17] se introduce el *umbral liviano*, en el que los coeficientes sobre el umbral se reducen en el valor absoluto del umbral en sí.

a. Umbrales no adaptativos VISUShrink [18] es un umbral universal no adaptativo, que depende solo del número de puntos de datos. Tiene equivalencia asintótica, sugiriendo el mejor rendimiento en términos de error cuadrático medio cuando el número de píxeles alcanza infinito.

b. Umbrales adaptativos SUREShrink [18] usa un híbrido del umbral universal y el umbral SURE (en inglés Stein's Unbiased Risk Estimator) y se desempeña mejor que VISUShrink.

III. Transformadas de wavelet no ortogonales La transformada de wavelet no diezmada (en inglés Undecimated Wavelet Transform - UDWT) es usada también para descomponer una señal para proveer una mejor solución visualmente. Aunque la mejora en los resultados es superior, el uso de UDWT incluye gran cantidad cálculos, lo que lo hace ser un proceso más lento [27].

Adicionalmente al uso de UDWT, el uso de *multiwavelets* mejora el desempeño pero también incrementa la complejidad computacional. Los multiwavelets son obtenidos mediante la aplicación de más de una función madre (función de escalado) a un conjunto de datos dado.

IV. Modelo de Coeficientes de Wavelet Éste enfoque se concentra en explotar las propiedades de multiresolución de la transformada de wavelet. Ésta técnica identifica una estrecha correlación de la señal en diferentes resoluciones observando la señal en múltiples resoluciones. Éste método provee excelentes resultados pero computacionalmente es mucho más complejo y costoso. El modelado de los coeficientes de wavelet puede ser determinístico o estadístico [27].

a. Determinístico El método determinístico de modelado implica crear una estructura de árbol de los coeficientes de wavelet, en donde cada nivel del árbol representa la escala de transformación y cada nodo representa los coeficientes de wavelet [6]. La aproximación óptima del árbol muestra una interpretación jerárquica de la descomposición de wavelet. Los coeficientes de wavelet de singularidades son los más grandes a lo largo de las ramas del árbol. Así, si un coeficiente de wavelet tiene una fuerte presencia en un nodo particular, entonces en el caso de ser una señal su presencia debería ser más pronunciada en sus ramas padre. Si es un coeficiente ruidoso, no se encontrará una presencia consistente.

b. Estadístico Éste enfoque se concentra en las propiedades más interesantes de la transformada de wavelet, como lo son la correlación multiescala entre los coeficientes de wavelet, la correlación local entre los coeficientes del vecindario, entre otros. Éste enfoque tiene el objetivo intrínseco de perfeccionar el modelado exacto de los datos de una imagen con el uso de la transformada de wavelet. En [13] y [30] se puede encontrar una amplia revisión sobre las propiedades estadísticas de los coeficientes de wavelet. En general, los

enfoques se clasifican en dos modelos probabilísticos: marginal y coyuntura.

2.2.2 Trabajos realizados en mejoramiento de contraste

Las imágenes con bajo contraste pueden resultar de una pobre iluminación, falta de un rango dinámico adecuado en el sensor de imagen, o también apertura inadecuada del objetivo durante la adquisición, o bien que el objeto que se desea adquirir posee en sí mismo un bajo contraste como es el caso de las imágenes de geles de electroforesis. El propósito de los métodos de mejoramiento de contraste es mejorar la visibilidad y los detalles de una imagen.

- **Clasificación de los algoritmos de mejoramiento de contraste**

En general, muchos autores [35, 36, 25] concuerdan como se muestra en la Figura 2.9, que los métodos de mejoramiento de contraste se clasifican en dos grandes categorías: *globales* y *locales*.

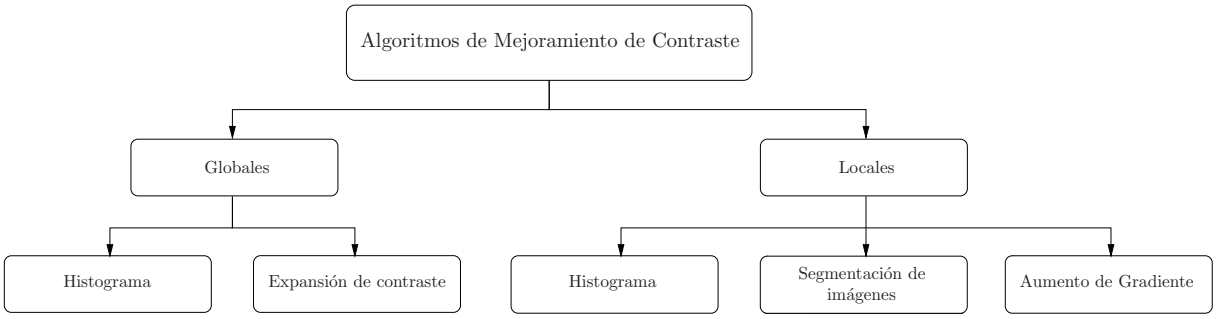


Figura 2.9: Clasificación de algoritmos de mejoramiento de contraste.

Globales Las técnicas de mejoramiento de contraste globales, remedian problemas como condiciones excesivas o pobres de iluminación en el ambiente de origen. Sin embargo, éstas técnicas no siempre producen buenos resultados especialmente en imágenes que tienen una gran variación espacial en el contraste.

I. Histograma La *ecualización de histograma* (en inglés Histogram Equalization - HE) [21] es un método ampliamente utilizado para el mejoramiento de contraste. En principio, la ecualización de histograma incrementa el contraste de una imagen mediante la transformación de su histograma en uno uniforme que abarque el rango dinámico completo de niveles de gris. Ésto se basa en la suposición que para una máxima transmisión de información, la distribución percibida (histograma) de niveles de gris en una imagen debe ser uniforme.

A pesar de la simplicidad y la definición implícita de la función de transformación en el método de ecualización de histograma, hay algunos problemas asociados a ello. El primero, es que resulta en un cambio significativo en el valor medio del brillo de la imagen. Ésto debido a que la ecualización de histograma transforma el histograma original de la imagen en una distribución uniforme, que tiene un valor medio en la mitad del rango de niveles de gris independientemente del valor medio de la imagen original. Segundo, el método de ecualización de histograma puede causar un sobre contraste o bien una saturación de los niveles de gris fuera del rango. Y tercero, la función de transformación de la ecualización de histograma es capaz de mejorar el contraste global de una imagen y podría o no aumentar el contraste local, debido a que está basado en el contenido global de la imagen.

Como se señaló en la discusión anterior la ecualización de histograma determina automáticamente la función de transformación que permite que la imagen resultante tenga un histograma uniforme. En éste sentido, es útil también poder especificar la forma del histograma que se desea que tenga la imagen procesada. El método que se utiliza para generar una imagen procesada con un histograma especificado se llama *especificación de histograma* (en inglés Histogram Specification - HS) [21]. En éste enfoque el contraste de la imagen original se modifica mediante la especificación del histograma deseado. La ecualización de histograma se puede considerar como un caso especial de especificación de histograma en donde el histograma deseado tiene una distribución uniforme. La ecualización de histograma se describe detallamente en la Sección 4.2.1.

II. Expansión de contraste La expansión de contraste [21] es una técnica simple de mejoramiento de contraste que intenta aumentar el contraste de una imagen "espandiendo" su rango de niveles de gris a un rango deseado de valores, como puede ser el rango completo de valores que el tipo de imagen permita. Éste método se diferencia de la ecualización de histograma en que solo puede aplicar una función lineal de escalado a los píxeles de la imagen.

Locales Los métodos globales son apropiados para mejoramiento en general, pero hay casos en que es necesario mejorar los detalles en pequeñas áreas de la imagen. En términos generales las técnicas locales presentan mejores resultados que las globales.

I. Histograma El procedimiento global de ecualización de histograma puede ser fácilmente extendido a mejoramiento local de contraste llamado *ecualización local de histograma* (en inglés Local Histogram Ecualization - LHE) [21]. En la ecualización local de histograma, se define primero un vecindario cuadrado o rectangular y se mueve el centro del mismo píxel por píxel através de toda la imagen. Ésta extensión de la ecualización de histograma permite a cada píxel adaptarse a su vecindario, así se puede lograr un mayor contraste para todas las ubicaciones de la imagen. Sin embargo, la ecualización local de histograma

resulta en una modificación no natural de la imagen. También éste método produce bordes en puntos donde la transformación local cambia abruptamente, debido al rápido cambio del histograma local. Ésto se debe a que la ecualización local de histograma es solo la extensión local de la ecualización de histograma, por tanto, hereda sus problemas de sobremejoramiento y saturación.

La especificación de histograma también tiene su contraparte en los enfoques locales de mejoramiento de contraste. La especificación de histograma es una técnica con aspecto inherente: ¿Cómo se define el histograma deseado?. En éste sentido Jafar et. al. en [25] proponen un nuevo método llamado *Especificación Local Automática de Histograma* (en inglés Automatic Local Histogram Specification - ALHS), que proporciona automáticamente el óptimo mejoramiento de contraste con una distorsión mínima en la apariencia de la imagen. Básicamente, la especificación local automática del histograma es aplicado localmente como la ecualización local de histograma. Sin embargo, para modificar el píxel en el centro del bloque, se especifica el histograma de salida deseado para ese bloque, así el método de especificación de histograma se utiliza para encontrar el nuevo valor del píxel. La idea central de la especificación automática del histograma es la determinación del histograma deseado para cada bloque. La especificación local automática del histograma define automáticamente el histograma, de tal manera que éste sea lo más cercano a la distribución uniforme y al mismo tiempo su valor medio del brillo tenga una mínima desviación respecto a su valor original.

II. Segmentación de Imágenes Muchos de los enfoques de mejoramiento de contraste local se basan en la segmentación de imágenes, para ésto se emplean métodos en el dominio espacial y en dominio de la frecuencia, seguidos de un operador de mejoramiento de contraste en cada segmento. Los enfoques se diferencian principalmente en la forma en que escoge para la representación multi-escala o multi-resolución de la imagen (difusión anisotrópica [20], técnicas piramidales no lineales [38], técnicas morfológicas multi-escala [37, 28]) o en la forma en que mejoran el contraste después de la segmentación (operadores morfológicos [28], transformadas de wavelet [40], transformadas curvelet [34], lógica difusa [23, 22] y algoritmos genéticos [31]).

III. Aumento de Gradiente En general, el aumento de gradiente se encuentra asociado con un aumento de contraste. En éste sentido Subr et. al. [35, 36] proponen un algoritmo que logra el mejoramiento del gradiente mediante la maximización de una función objetivo, obteniendo así una imagen resultante con un mayor contraste, que se controla con un sólo parámetro de éste algoritmo. Además, se basa en restricciones que evitan que los valores de los niveles de gris se saturen en la imagen procesada. En particular entre los algoritmos de mejoramiento de contraste, es de especial interés para el presente trabajo. En la Sección 4.2.2 se describe detalladamente éste algoritmo.

2.2.3 Mejoramiento de nitidez

El principal objetivo del mejoramiento de la nitidez de imágenes es resaltar los detalles finos en una imagen o mejorar los detalles que fueron difuminados, ya sea, por un error o por un efecto natural del proceso de adquisición. En general, los esfuerzos de investigación en el área de mejoramiento de nitidez de imágenes son más escasos que en el caso de reducción de ruido y de mejoramiento de contraste.

- **Clasificación de los algoritmos de mejoramiento de nitidez**

Al igual que en el caso de los algoritmos de reducción de ruido los algoritmos de mejoramiento de nitidez como se muestra en la Figura 2.10 se pueden clasificar en: *dominio espacial*, *dominio de la transformada de Fourier* y *dominio de la transformada de wavelet*.

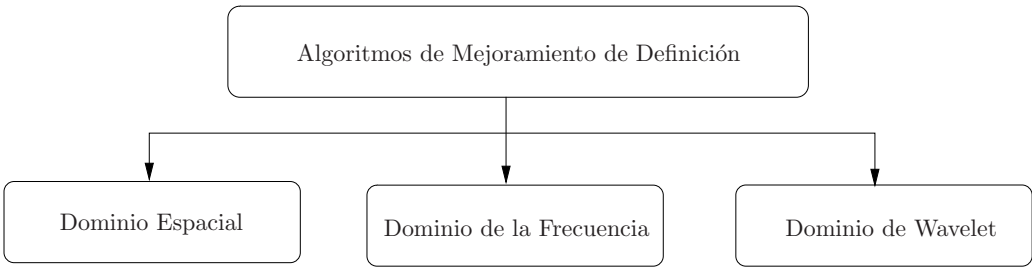


Figura 2.10: Clasificación de algoritmos de mejoramiento de nitidez.

Dominio Espacial En general, la difuminación de una imagen se puede lograr en el dominio espacial mediante el promediado de los valores de un vecindario. Debido a que el promediado es análogo a la integración, es posible concluir que el mejoramiento de la nitidez en una imagen se puede alcanzar mediante la diferenciación espacial. Ésto porque la diferenciación de imágenes mejora los bordes y otras discontinuidades (como el ruido desafortunadamente) y desenfatiza áreas con variación lenta en los valores del nivel de gris. Uno de los métodos más ampliamente utilizados en el mejoramiento de nitidez se llama *enmascaramiento de desenfoque* [21], que consiste en sustraer una versión difuminada de la imagen original de la imagen en sí misma. La imagen que se resta es filtrada mediante la aplicación de una máscara ya sea gaussiana o laplaciana. Ésta técnica se presenta con más detalle en la Sección 4.3.1.

I. Dominio de la Frecuencia Como se discutió en la Sección 2.1.2, los bordes y otros cambios abruptos en los niveles de grises están asociados con los componentes de alta frecuencia, de tal manera que el mejoramiento de la nitidez de una imagen puede ser logrado en el dominio de la frecuencia mediante proceso de *filtrado pasoalto*, que atenúa las bajas frecuencias sin modificar la información de alta frecuencia. Algunos de los filtros pasoalto que se emplean son: Ideal, Butterworth, y el Gaussiano [21].

II. Dominio de Wavelet Donoho en [16] propone un algoritmo basado en wavelets. La idea principal de éste enfoque es que a un nivel de resolución dado, tanto la transformada de wavelet, como su inversa pueden ser expresadas como una convolución con ciertas funciones wavelet, así lograr enfatizar ciertas componentes para la mejora de nitidez.

Capítulo 3

Método de evaluación de algoritmos

En general ningún algoritmo de mejoramiento de calidad es aplicable a todo tipo de imágenes y tampoco es apto para todo tipo aplicación, es por ésto que se hace necesario evaluar los algoritmos objetivamente, con el fin de conocer por completo el comportamiento de cada algoritmo y así mismo, para efectuar comparaciones directas entre ellos. La alternativa empleada en éste trabajo es la optimización multi-objetivo que no solo se limita a encontrar una única solución, sino que más bien busca el conjunto de las soluciones "óptimas" para un determinado algoritmo y sus parametrizaciones.

3.1 Evaluación de algoritmos

Zhang en [41] propuso una categorización para la evaluación de algoritmos de segmentación que se muestra en la Figura 3.1, sin embargo, ésta puede ser aplicada a otro tipo de algoritmos como lo son los de mejoramiento de calidad de imágenes.

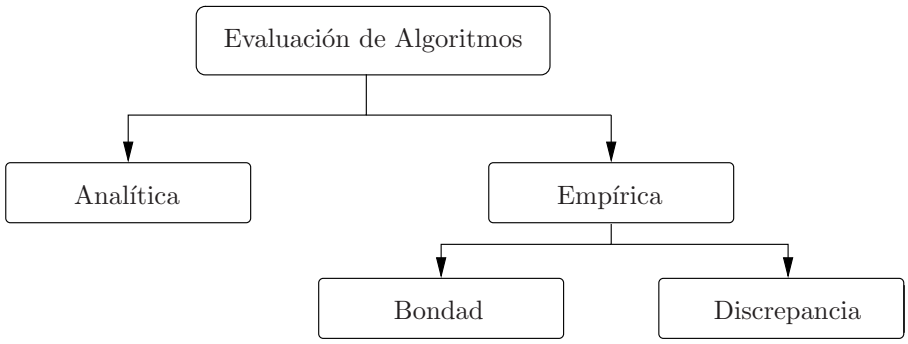


Figura 3.1: Alternativas de evaluación de algoritmos según Zhang [41]

Los métodos analíticos tratan con los algoritmos en sí mismos considerando principios subyacentes, hipótesis, limitaciones, complejidad, entre otros. Éstas técnicas no requieren la implementación de los algoritmos y están exentas de los efectos negativos de la experimentación. Sin embargo, ellos no pueden describir aspectos del desempeño de los algoritmos.

Los métodos empíricos trabajan con la salida de los algoritmos, requiriendo explícitamente la implementación de los mismos. Las medidas de *bondad* definen medidas cuantitativas que juzgan la concordancia de los resultados de los algoritmos con el concepto idealizado, sin la necesidad de datos de referencia. Así las medidas de bondad son absolutas, como por ejemplo el tiempo de ejecución de un determinado algoritmo. Finalmente, la última categoría son las medidas de *discrepancia*. En éstas a diferencia de las medidas de bondad, la salida de un algoritmo es comparada con datos de referencia, llamados *datos de oro* (en inglés golden data), y las diferencias son codificadas en alguna métrica.

El mayor problema en la evaluación de técnicas de mejoramiento de imágenes es la diversidad de resultados de un algoritmo dependiendo de la parametrización elegida. Éste aspecto es tradicionalmente descuidado en la literatura de mejoramiento de imágenes e inclusive en otras áreas, así, cuando un nuevo enfoque es propuesto, sus ventajas sobre otras técnicas son comúnmente mostradas bajo la consideración de un solo conjunto de parámetros que raramente está optimizado. Para evitar tal situación, el presente trabajo plantea la evaluación objetiva de los algoritmos y sus parametrizaciones.

Everingham et. al. en [19] señala el riesgo de intentar concentrar las bondades de un algoritmo es una sola medida, debido a que es imposible reflejar todas las dependencias entre parámetros y propiedades del resultado de un algoritmo en un solo valor escalar. Un compromiso entre varias medidas de aptitud o de costo puede ser hecho solo si suficiente información acerca de los efectos de diferentes parametrizaciones están disponibles.

La evaluación a través de discrepancia empírica está asociada con la generación de un conjunto de datos de referencia, que en éste caso son las imágenes originales. La generación de éste conjunto de *oro* dependiendo de la aplicación, puede ser una difícil tarea, debido a que la evaluación es significativa solo si el conjunto de referencia es suficientemente representativo para el contexto. Por ésta razón, muchos autores optan por mostrar simplemente uno o dos ejemplos, usando solo un conjunto de parámetros [9, 35]. En el caso particular de éste trabajo, el conjunto de referencia está constituido por algunas imágenes de geles de electroforesis.

En éste trabajo el enfoque seleccionado es la evaluación empírica propuesta por Everingham et al. [19], que está basada en la optimización de desempeño multi-objetivo. El resultado de tal mecanismo de evaluación es un *frente*, que describe las "mejores" configuraciones en un espacio de aptitudes multi-dimensional. Para un compromiso dado entre

medidas de aptitud, éste enfoque permite seleccionar no solo objetivamente el algoritmo, sino también la parametrización que produce un punto de operación deseado.

3.2 Frente de Pareto

El agregado de funciones de aptitud F para una algoritmo A con parametrización $\mathbf{u}$, evaluada usando como referencia un conjunto de datos $\mathcal{G}$ se define como

$$F(A_{\mathbf{u}}, \mathcal{G}) = \Phi(f_1(A_{\mathbf{u}}, \mathcal{G}), ..., f_n(A_{\mathbf{u}}, \mathcal{G})) \tag{3.1}$$

con funciones de aptitud individuales $f_k(A_{\mathbf{u}}, \mathcal{G})$ que son monotónicamente crecientes con la aptitud de algún aspecto particular del comportamiento del algoritmo. Las funciones f_k generan un espacio multidimensional de aptitudes, donde cada punto representa el desempeño de un algoritmo parametrizado con un punto $\mathbf{u}$ en un espacio de parámetros. La forma general de Φ es asumida como desconocida, pero se incrementa monotónicamente con aumentos de todas las funciones de aptitud f_k. Ésta condición asegura que ese punto en el espacio de aptitud puede ser considerado como más apto que todos los otros puntos con valores menores en todas las dimensiones.

En la Figura 3.2, por ejemplo, el punto p_1 es más apto que el punto p_4 y todos los demás puntos que se encuentren dentro del rectángulo gris. En éste contexto, el punto p_1 se dice domina a p_4. Todos los puntos no dominados en un conjunto definen el frente de pareto de ese conjunto. En el ejemplo de la Figura 3.2, el frente está definido por los puntos p_1, p_2 y p_3. Elegir una parametrización que no está en el frente de Pareto será siempre una mala elección, debido a que un punto que forme parte del frente de pareto tendrá un mejor rendimiento.

En la Figura 3.2 el punto p_1 domina la región marcada por el rectágulo gris. Las líneas punteadas delimitan la región de dominancia de los puntos p_2, p_3 y p_4. La línea sólida señala el frente de Pareto para el conjunto de puntos mostrado. Los conceptos previos son matemáticamente expresados por la expresión

$$\widehat{\mathcal{P}} = \{\langle \mathbf{u} \in \mathbb{P}_A, \mathbf{f}(A_{\mathbf{u}}, \mathcal{G})\rangle | \neg\exists \mathbf{v} \in \mathbb{P}_A : \mathbf{f}(A_{\mathbf{v}}, \mathcal{G}) \succ \mathbf{f}(A_{\mathbf{u}}, \mathcal{G})\} \tag{3.2}$$

donde $\widehat{\mathcal{P}}$ es el frente de Pareto, $\mathbf{f}$ es el vector de funciones de aptitud $[f_1, ..., f_n]^T$ y $\mathbb{P}_A$ es el espacio de parámetros del algoritmo A. La relación parcial de ordenamiento "$\succ$" describe la propiedad de dominancia, en donde el vector de funciones de aptitud $\mathbf{f}(A_{\mathbf{v}}, \mathcal{G})$ domina a $\mathbf{f}(A_{\mathbf{u}}, \mathcal{G})$. A lo anterior se le conoce como dominancia de Pareto, y se define como

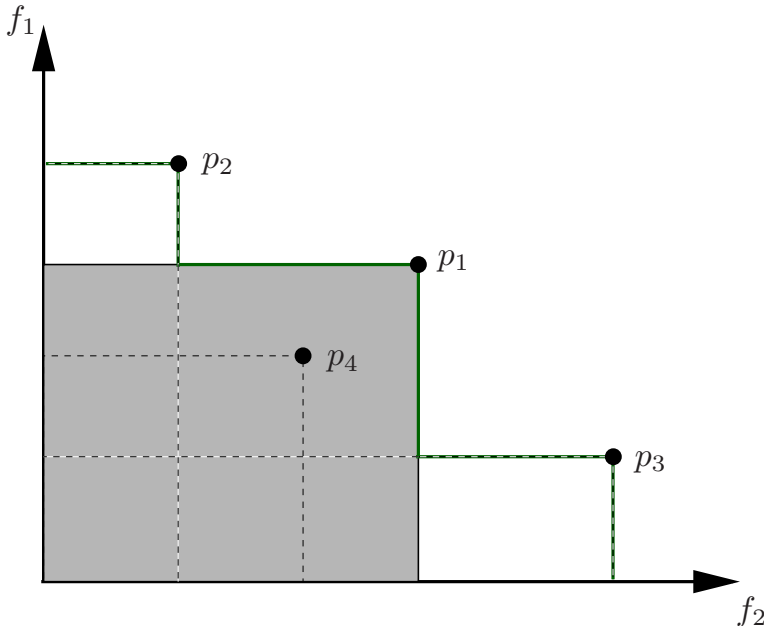


Figura 3.2: Frente de Pareto

$$\mathbf{f}(A_{\mathbf{v}}, \mathcal{G}) \succ \mathbf{f}(A_{\mathbf{u}}, \mathcal{G}) \Leftrightarrow \forall k : f_k(A_{\mathbf{v}}, \mathcal{G}) \geq f_k(A_{\mathbf{u}}, \mathcal{G}) \wedge \exists k : f_k(A_{\mathbf{v}}, \mathcal{G}) > f_k(A_{\mathbf{u}}, \mathcal{G}) \quad (3.3)$$

Cualquier algoritmo que encuentre el frente de Pareto para un conjunto de puntos de aptitud implementa (3.2) y (3.3). Debido a que el espacio de parámetros $\mathbb{P}_A$, usualmente contiene un número infinito de paremetrizaciones, el siguiente problema consiste en elegir un conjunto representativo de muestras de $\mathbb{P}_A$, de tal manera, que su frente de Pareto se pueda considerar como una fiel aproximación del frente exacto para el espacio completo.

Una aproximación ingenua sería muestrear regularmente los valores de cada parámetro, ya que el número de evaluaciones necesarias podría crecer exponencialmente con el número de parámetros. Así por ejemplo, un algoritmo con 4 parámetros, cada uno muestreado diez veces, requeriría $10^4 = 10000$ evaluaciones. Debido a que una sola evaluación, comprende cálculos para un conjunto completo de datos, el tiempo de ejecución requerido para ésta alternativa es enorme.

En éste trabajo se empleó la optimización evolutiva multi-objetivo, cuya principal finalidad es encontrar el frente de Pareto. Ésta es una alternativa genética, que elimina los cálculos innecesarios y concentra su atención en aquellas regiones del espacio de parámetros en donde se obtienen los mejores resultados. El número de evaluaciones requeridas en ésta alternativa genética es proporcional al número de bits usados para representar la parametrización.

Todos los algoritmos de optimización multi-objetivo intentan encontrar el frente que contiene las parametrizaciones mejor optimizadas para el conjunto de datos de referencia $\mathcal{G}$. Por lo tanto, como se mencionó en la sección anterior es importante que en la evaluación se utilicen datos representativos del contexto de la aplicación; para el caso particular de éste trabajo éstos datos representativos son las imágenes de geles de electroforesis.

3.3 Funciones de aptitud

Una función de aptitud es un tipo particular de función objetiva, qué cuantifica que tan "óptima" es una solución. Una función de aptitud ideal debe estar correlacionada con el objetivo del algoritmo.

Para algoritmos de evaluación multi-objetivo, se deben seleccionar varias funciones de aptitud, que pueden ser medidas de bondad o de discrepancia. Independientemente de la selección final, un conjunto de datos de referencia $\mathcal{G}$ debe ser elegido. En el caso de mejoramiento de imágenes éste conjunto está definido como

$$\mathcal{G} = \{\langle \mathcal{I}_k, \mathcal{N}_k \rangle | k = 1...n\} \tag{3.4}$$

donde $\mathcal{I}_k$ es la imagen de referencia, y $\mathcal{N}_k$ es la imagen de referencia degradada, ya sea, con ruido, con un mal contraste o con una mala definición. El conjunto $\mathcal{G}_\mathcal{I} = \{\mathcal{I}_k | k = 1...n\}$ contiene las imágenes de referencia. $\mathcal{G}_\mathcal{N} = \{\mathcal{N}_k | k = 1...n\}$ contiene las imágenes de referencia degradadas.

Sea $\widehat{\mathcal{I}}_k$ una imagen mejorada mediante un algoritmo A parametrizado con $\mathbf{u}$ a partir de la imagen desgradada $\mathcal{N}_k$

$$\widehat{\mathcal{I}}_k = A_\mathbf{u}[\mathcal{N}_k] \tag{3.5}$$

El objetivo es buscar funciones de aptitud que evalúen la diferencia entre la imagen original $\mathcal{I}_k$ y la imagen mejorada $\widehat{\mathcal{I}}_k$ mediante el algoritmo A.

Para cada uno de los algoritmos que mejoran los parámetros de calidad de una imagen, a saber, ruido, contraste y nitidez se emplearon diferentes medidas de aptitud. También, se empleó una medida aptitud en común, que es el inverso del tiempo que toma cada algoritmo en ejecutarse.

Imágenes procesadas por segundo El inverso del tiempo de ejecución es una medida de aptitud que permite evaluar la eficiencia de los algoritmos. Éste dato es relevante en

aplicaciones que se deben ejecutar en tiempo real. Se emplea el inverso puesto que es deseable un bajo tiempo de ejecución y así se utiliza el tiempo de ejecución como aptitud y no como una medida de costo, y al mismo tiempo ésto permite obtener una medida de la cantidad de imágenes que un algoritmo procesa por segundo. Sea $te(A_{\mathbf{u}}[\mathcal{N}_k])$ el tiempo de ejecución requerido para que el algoritmo mejore la calidad de la imagen $\mathcal{N}_k$. Así, ésta es una medida de bondad definida como

$$f_{ips}(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}}) = \frac{|\mathcal{G}_{\mathcal{N}}|}{\sum\limits_{\mathcal{N}_k \in \mathcal{G}_{\mathcal{N}}} te(A_{\mathbf{u}}[\mathcal{N}_k])} \tag{3.6}$$

3.3.1 Funciones de aptitud para algoritmos de reducción de ruido

Para evaluar aspectos particulares de los algoritmos de reducción de ruido en una imagen se emplearon dos medidas de discrepancia: la primera de ellas es el *error cuadrático medio inverso* y la segunda se llama *ruido de método escalar*.

Error cuadrático medio inverso El error cuadrático medio mide el cuadrado de la distancia entre una imagen y su estimación, que éste caso ésta última imagen es la obtenida a partir del algoritmo de reducción de ruido. Como se busca medidas de aptitud se emplea el inverso del error cuadrático medio. Ésta es una medida de discrepancia y se define como

$$f_{ecmi}(A_{\mathbf{u}}, \mathcal{G}) = \frac{1}{|\mathcal{G}_{\mathcal{I}}|} \sum_{\mathcal{I}_k, \mathcal{N}_k \in \mathcal{G}} \frac{|\mathbb{G}_k^2|}{\sum\limits_{<p_m, p_n> \in \mathbb{G}_k^2} ||\mathcal{I}_k(p_m) - \widehat{\mathcal{I}}_k(p_n)||_2^2} \tag{3.7}$$

donde $\mathcal{I}_k$ es la imagen original, $\widehat{\mathcal{I}}_k = A_{\mathbf{u}}[\mathcal{N}_k]$ es la imagen mejorada mediante el algoritmo, p_m es un píxel que pertenece a $\mathbb{G}_k^2$, donde $\mathcal{I}_k : \mathbb{G}_k^2 \to \mathbb{R}$, $\widehat{\mathcal{I}}_k : \mathbb{G}_k^2 \to \mathbb{R}$, $|\mathbb{G}_k^2|$ es la cardinalidad de $\mathbb{G}_k^2$ y $|\mathcal{G}_{\mathcal{I}}|$ es la cardinalidad de $\mathcal{G}_{\mathcal{I}}$.

Ruido de método (*Method noise*) En principio el ruido de método (en inglés method noise) no es una medida escalar, sino más bien es una medida de diferencia entre imágenes. Éste fue propuesto por Buades et. al. en [9] y se define de la siguiente manera:

Definición 3.1 (Ruido de método escalar) *Sea una imagen (ruidosa o no)* $\mathcal{N}$ *y* $A_{\mathbf{u}}$ *un operador de reducción de ruido que depende de un parámetro de filtrado* $\mathbf{u}$. *Entonces,*

se define como el ruido de método como una diferencia entre imágenes.

$$rm(A_{\mathbf{u}}[\mathcal{N}], \mathcal{N}) = \mathcal{N} - A_{\mathbf{u}}[\mathcal{N}] \tag{3.8}$$

En general uno de los principales problemas de los algoritmos de reducción de ruido es que éstos difuminan las imágenes, porque el ruido al contener sus componentes en altas frecuencias es fácilmente confundido con bordes y finas estructuras, que también contienen altas frecuencias. El ruido de método nos dice cuáles de las características geométricas o detalles de la imagen a mejorar son preservados por el algoritmo de reducción de ruido y cuáles son eliminados. La diferencia entre la imagen original y su versión filtrada, muestra el ruido que fue removido por el algoritmo. Con el fin de preservar la mayoría de detalles posibles de la imagen original, el ruido de método debe acercarse lo más cercano posible a ruido blanco y gaussiano. El ruido de método permite evaluar otro aspecto de los algoritmos de reducción de ruido diferente al evaluado con el error cuadrático medio y también es una medida de discrepancia.

El ruido de método no es un valor escalar sino una imagen, por tanto, se determinó una medida que evalúa el contenido geométrico y de detalles de la imagen original que elimina el algoritmo de reducción de ruido. En principio se evalúa la varianza media local de la imagen del ruido de método, así cuanto más detalles elimina el algoritmo de la imagen original mayor es la varianza, siendo ésta medida un costo. Sin embargo, se utiliza su inverso como una medida de aptitud, y se define el *ruido de método escalar* como

$$f_{rme}(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}}) = \frac{1}{|\mathcal{G}_{\mathcal{N}}|} \sum_{\mathcal{N}_k \in \mathcal{G}_{\mathcal{N}}} \frac{8|\mathbb{G}_k^2|}{\sum\limits_{p_m \in \mathbb{G}_k^2} \sum\limits_{p_n \in \mathcal{V}_8(p_m)} (\mathcal{RM}_k(p_m) - \mathcal{RM}_k(p_n))^2} \tag{3.9}$$

con

$$\mathcal{RM}_k = rm(A_{\mathbf{u}}[\mathcal{N}_k], \mathcal{N}_k) \tag{3.10}$$

donde $\mathcal{RM}_k$ es la imagen del ruido de método, p_m es un píxel que pertenece a $\mathbb{G}_k^2$, tal que $\mathcal{RM}_k : \mathbb{G}_k^2 \to \mathbb{R}$, p_n es un píxel vecino de p_m, $\mathcal{V}_8(p_m)$ es una vecindad de 8 de p_m, $|\mathbb{G}_k^2|$ es la cardinalidad de $\mathbb{G}_k^2$ y $|\mathcal{G}_{\mathcal{N}}|$ es la cardinalidad de $\mathcal{G}_{\mathcal{N}}$.

3.3.2 Funciones de aptitud de contraste en una imagen

Para evaluar el rendimiento de los algoritmos de mejoramiento de contraste se empleó una medida de discrepancia llamada *mejoramiento de contraste promedio*, y una medida

de bondad "llamada" *entropía*.

Mejoramiento de Contraste Promedio Ésta medida de aptitud se basa en la función objetivo en la cual se fundamenta el algoritmo de mejoramiento de gradiente descrito en la Sección 4.2.2. Ésta función de aptitud es una medida de discrepancia que cuantifica la razón de mejoramiento de contraste de una imagen que se supone con bajo constraste con respecto a su versión mejorada y se define de la siguiente manera

$$f_{mcp}(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}}) = \frac{1}{|\mathcal{G}_{\mathcal{N}}|} \sum_{\mathcal{N}_k \in \mathcal{G}_{\mathcal{N}}} \frac{1}{4|\mathbb{G}_k^2|} \sum_{p_m \in \mathbb{G}_k^2} \sum_{p_n \in \mathcal{V}_4(p_m)} \frac{\widehat{\mathcal{I}}_k(p_m) - \widehat{\mathcal{I}}_k(p_n)}{\mathcal{N}_k(p_m) - \mathcal{N}_k(p_n)} \tag{3.11}$$

donde $\widehat{\mathcal{I}}_k = A_{\mathbf{u}}[\mathcal{N}_k]$, p_m es un píxel que pertenece a $\mathbb{G}_k^2$, tal que $< \mathcal{I}_k, \mathcal{N}_k >: \mathbb{G}_k^2 \to \mathbb{R}$, p_n es un vecino de p_m, $\mathcal{V}_4(p_m)$ es un conjunto de cuatro vecinos de p_m, $|\mathbb{G}_k^2|$ es la cardinalidad de $\mathbb{G}_k^2$ y $|\mathcal{G}_{\mathcal{N}}|$ es la cardinalidad de $\mathcal{G}_{\mathcal{N}}$.

Entropía La entropía discreta es una medida de la información contenida en una imagen, en donde valores altos indican imágenes ricas en detalles, y por tanto poseen alto contraste. Ésta es una medida de bondad y se define de la siguiente manera

$$f_e(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}}) = \frac{1}{|\mathcal{G}_{\mathcal{N}}|} \sum_{\mathcal{N}_k \in \mathcal{G}_{\mathcal{N}}} \sum_{n=0}^{255} h_k(n) \log_2 h_k(n), \quad \forall h_k(n) \neq 0 \tag{3.12}$$

donde h_k es el histograma normalizado de $\widehat{\mathcal{I}}_k = A_{\mathbf{u}}[\mathcal{N}_k]$. Para más detalles sobre el histograma refiérase a la Sección 4.2.1.

3.3.3 Funciones de aptitud de nitidez en una imagen

Para evaluar la mejora de nitidez introducida por un filtro a una imagen, se empleará una pareja de medidas complementarias entre sí, que miden la nitidez y la difuminación, y fueron presentadas por Dijik et. al. [15]. Ésta pareja de mediciones se basa en un *diagrama de dispersión* en donde se grafican la magnitud del gradiente de los píxeles en la imagen original versus la magnitud de los correspondientes píxeles en la imagen mejorada. Para ilustrar en la Figura 3.3 se muestra un ejemplo de un diagrama de dispersión de una imagen filtrada con un algoritmo de mejoramiento de contraste llamado enmascarado de desenfoque descrito detalladamente en la Sección 4.3.1.

A partir de los valores de la magnitud de los gradientes los píxeles son clasificados en los que fueron difuminados y los que incrementaron su nitidez. En el primer caso el gradiente de los píxeles aumenta; dichos puntos se muestran bajo la línea $x = y$ en el escatergrama.

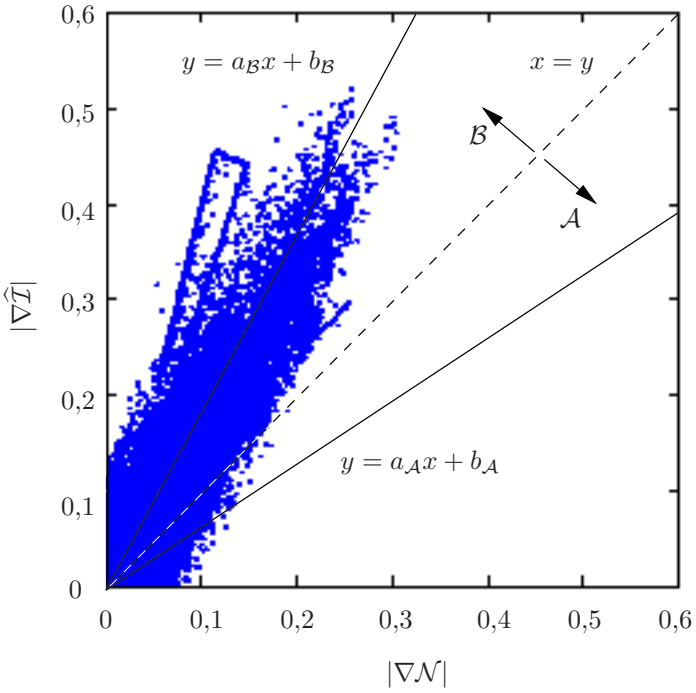


Figura 3.3: Diagrama de dispersión de la magnitud del gradiente de la imagen original $\mathcal{N}$ que se supone con baja nitidez versus la magnitud del gradiente de la imagen filtrada con máscara de desenfoque $\widehat{\mathcal{I}} = A_{\mathbf{u}}[\mathcal{N}]$

Los píxeles que aumentaron su gradiente se grafican sobre ésta línea. Éstos dos grupos de píxeles se clasifican en dos conjuntos denotados como $\mathcal{A}$ y $\mathcal{B}$, respectivamente

$$\mathcal{A} = \{(|\nabla\mathcal{N}|, |\nabla\widehat{\mathcal{I}}|) \mid |\nabla\mathcal{N}| \geq |\nabla\widehat{\mathcal{I}}|) \tag{3.13}$$

$$\mathcal{B} = \{(|\nabla\mathcal{N}|, |\nabla\widehat{\mathcal{I}}|) \mid |\nabla\mathcal{N}| < |\nabla\widehat{\mathcal{I}}|) \tag{3.14}$$

Las rectas $y = ax + b$ representan las líneas de mejora ajuste para cada uno de los dos conjuntos de datos, esto con el objetivo de obtener factores que indiquen el grado en que los bordes incrementaron su gradiente y las regiones planas fueron difuminadas. Éstas rectas se calculan minizando la desviación absoluta de la siguiente manera

$$(a_{\mathcal{A}}, b_{\mathcal{A}}) = \texttt{arg}\min_{(a,b)} \sum_{(x,y)\in\mathcal{A}} |y - (ax + b)| \tag{3.15}$$

$$(a_{\mathcal{B}}, b_{\mathcal{B}}) = \texttt{arg}\min_{(a,b)} \sum_{(x,y)\in\mathcal{B}} |y - (ax + b)| \tag{3.16}$$

De donde se obtiene lo siguiente

$$a_{\mathcal{A}} = \frac{\sum\limits_{(x,y)\in\mathcal{A}} xy}{\sum\limits_{(x,y)\in\mathcal{A}} x^2} \tag{3.17}$$

$$a_{\mathcal{B}} = \frac{\sum\limits_{(x,y)\in\mathcal{B}} xy}{\sum\limits_{(x,y)\in\mathcal{B}} x^2} \tag{3.18}$$

La pendiente $a_{\mathcal{A}}$ indica el grado de difuminación introducido por el filtro. De la misma manera, $a_{\mathcal{B}}$ indica el grado de nitidez introducido por el filtro. Los valores de $b_{\mathcal{A}}$ y $b_{\mathcal{B}}$ se descartan. Nótese que $a_{\mathcal{A}} \leq 1$ y $a_{\mathcal{B}} \geq 1$.

Para tomar en cuenta el número de píxeles empleados para estimar éstos valores, las pendientes son ponderadas con el número relativo de puntos utilizados para la estimación, de la siguiente manera

Difuminación

$$f_e(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}}) = \frac{1}{|\mathcal{G}_{\mathcal{N}}|} \sum_{\mathcal{N}_k \in \mathcal{G}_{\mathcal{N}}} (a'_{\mathcal{A}} - 1) \frac{|\mathcal{A}|}{|\mathcal{A}| + |\mathcal{B}|} \tag{3.19}$$

donde se emplea $a'_{\mathcal{A}} = \frac{1}{a_{\mathcal{A}}}$ para obtener números en el mismo rango $[1, \infty)$.

Nitidez

$$f_n(A_{\mathbf{u}}, \mathcal{G}_{\mathcal{N}}) = \frac{1}{|\mathcal{G}_{\mathcal{N}}|} \sum_{\mathcal{N}_k \in \mathcal{G}_{\mathcal{N}}} (a_{\mathcal{B}} - 1) \frac{|\mathcal{B}|}{|\mathcal{A}| + |\mathcal{B}|} \tag{3.20}$$

Éstos dos valores pueden ser considerados como un factor de amplificación de bordes y otros detalles, y un factor de atenuación de las regiones planas, respectivamente.

Capítulo 4

Algoritmos Implementados

En éste capítulo se presentan los algoritmos de mejoramiento de imágenes comparadas en éste trabajo. En el caso de la reducción de ruido se comparan: el *Reductor de Ruido Gaussiano*, el *Filtro de Mediana*, el *Reductor de Ruido Susan* [33], y el algoritmo llamado *Medias no Locales* [9, 11, 12, 10, 8], éste último fue el que se implementó en éste trabajo. Con respecto al mejoramiento de contraste se comparan: *Ecualización de Histograma* y un algoritmo de *mejoramiento de gradiente* [35, 36], éste último fue el que se implementó. Finalmente, para el mejoramiento de nitidez se implementaron y compararon dos variantes del algoritmo llamado *enmascaramiento de desenfoque*. Las variantes en éste algoritmo se basan en el tipo de máscara empleada para el desenfoque, en la primera se utilizó una máscara *laplaciana* y en la segunda una máscara *gaussiana*.

4.1 Algoritmos para la reducción de ruido

En ésta sección primero se presentan los modelos de ruido más comunes en procesamiento digital de imágenes, posteriormente se describen los algoritmos de reducción de ruido.

4.1.1 Modelos de Ruido

La principal fuente de ruido en imágenes digitales se encuentra en el proceso de adquisión de éstas, así como en el proceso de transmisión de datos. El desempeño de los sensores de imágenes se ven afectados por una variedad de factores como condiciones ambientes y la calidad de los componentes de los sensores de imágenes en sí. Antes de describir cada uno de los algoritmos de reducción de ruido, se definen algunas de las funciones de densidad de probabilidad (en inglés Probability Density Function - PDF) de ruido más comunes en

aplicaciones de procesamiento de imágenes. El interés en éste trabajo se concentra en el ruido *gaussiano* y en el ruido *impulsional*.

Ruido Gaussiano y Blanco

El modelo de ruido Gaussiano (también llamado *normal*) es usado frecuentemente en la práctica. La PDF de una variable aleatoria Gaussiana, z, está dada por

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{(z-\mu)^2/2\sigma^2} \tag{4.1}$$

donde z representa el nivel de gris, μ es el valor promedio de z, y σ es la desviación estándar. La desviación estándar cuadrática, σ^2, se llama *varianza* de z. El ruido gaussiano tiene un efecto general en toda la imagen, es decir, la intensidad de cada píxel de la imagen se ve alterada en cierta medida con respecto a la intensidad en la imagen original. En la Figura 4.1 se muestra un ejemplo de una imagen con ruido gaussiano y blanco con una desviación estándar de $\sigma = 0.1$.

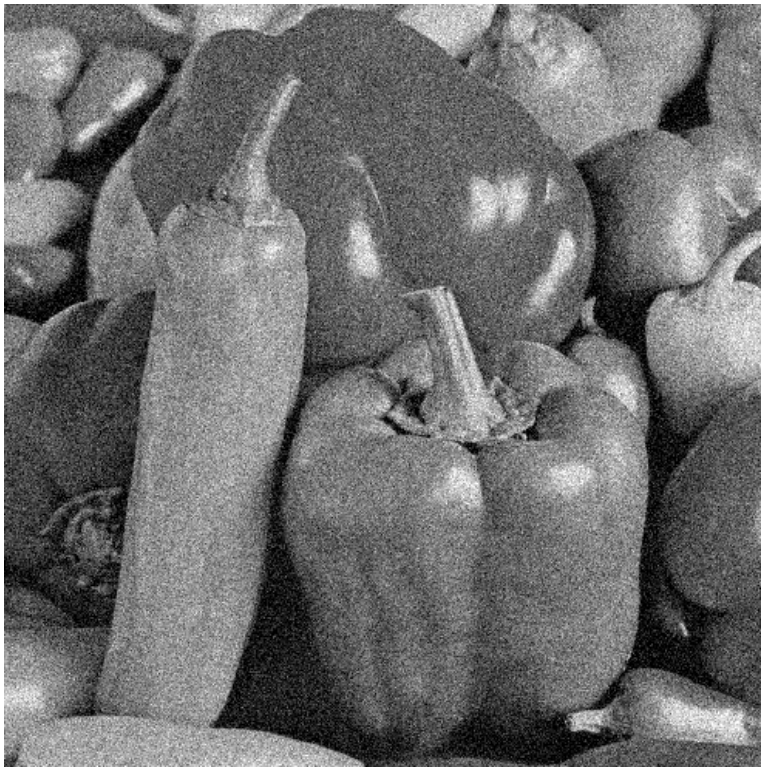


Figura 4.1: Imagen con ruido Gaussiano y Blanco con $\sigma = 0.1$

Ruido Impulso (Sal y Pimienta)

La PDF del ruido impulsional (bipolar) está dada por

$$p(z) = \begin{cases} P_a & para\ z = a \\ P_b & para\ z = b \\ 0 & en\ otro\ caso \end{cases} \tag{4.2}$$

Si $b > a$, el nivel de gris b aparecerá como un punto claro en la imagen, y el nivel a aparecerá como un punto oscuro. Si P_a o P_b es cero, el ruido impulsional se llama *unipolar*. Si ninguna de las probabilidades son cero, y especialmente si son aproximadamente iguales, el ruido impulsional tendrá la apariencia de gránulos de sal y pimienta distribuidos aleatoriamente por toda la imagen. A diferencia del ruido gaussiano el ruido impulsional tiene un efecto sobre un subconjunto del total de píxeles de la imagen. En la Figura 4.2 se muestra un ejemplo de una imagen con ruido impulsional.

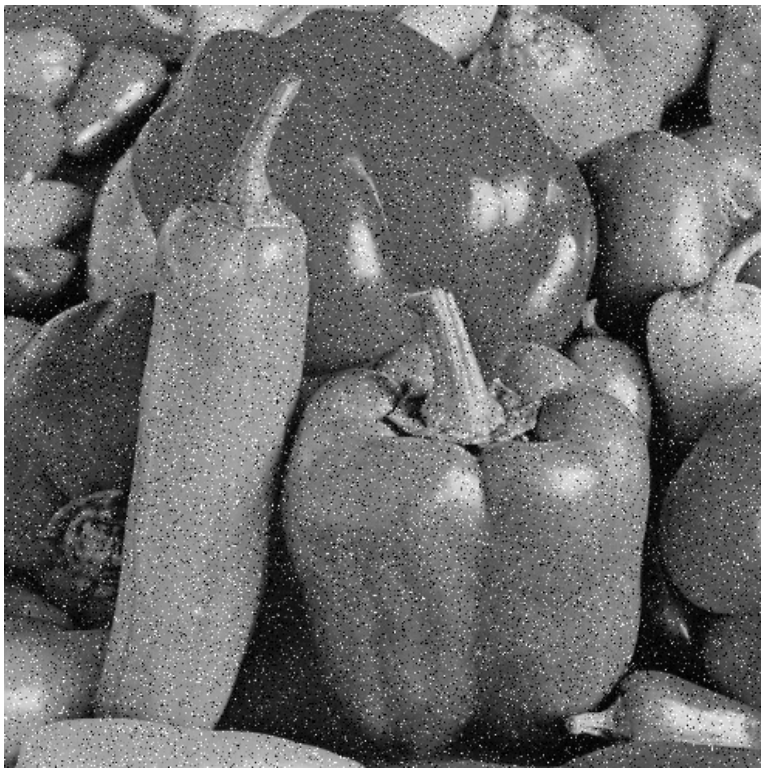


Figura 4.2: Imagen con ruido impulsional

4.1.2 Reductor de Ruido Gaussiano

El *reductor de ruido Gaussiano* se basa en una máscara que se convoluciona con una imagen, para efectos de reducir el ruido. Sin embargo, éste tipo de máscara remueve también detalles de la imagen y la difumina. La distribución Gaussiana de 1D tiene la siguiente forma

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{4.3}$$

donde σ es la desviación estándar de la distribución. También se asume que ésta distribución tiene una valor medio de cero. Ésta distribución es ilustrada en la Figura 4.3.

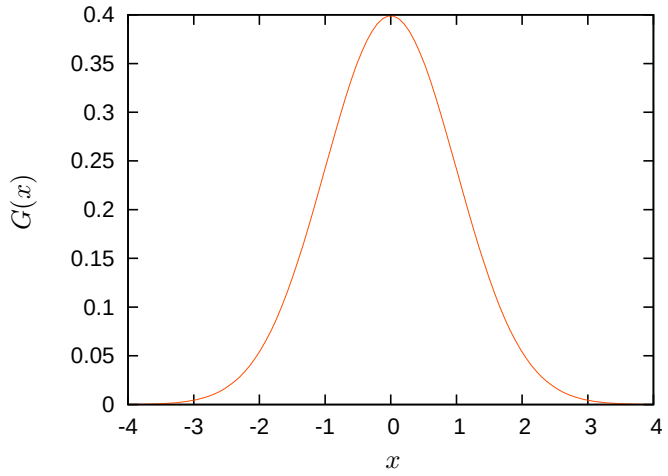


Figura 4.3: Distribución Gaussiana de una dimensión, con media 0 y $\sigma = 1$

El interés para procesamiento de imágenes es una máscara en dos dimensiones. En éste sentido, una máscara Gaussiana de 2D isotrópica se define de la siguiente manera

$$G(x, y) = G(x)G(y) = \frac{1}{\sigma^2 2\pi} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.4}$$

Ésta distribución en dos dimensiones se muestra en la Figura 4.4.

Debido a que las imágenes son una colección discreta de píxeles, es necesario obtener una aproximación discreta de la función Gaussiana para poder implementar la convolución. En la Figura 4.5, se muestra una máscara Gaussiana de valores enteros de 5×5 píxeles con σ de 1.

Una vez que la máscara ha sido calculada, el reductor de ruido Gaussiano puede ser implementado con la técnica de convolución discutida en la Sección 2.1.2. En la Tabla

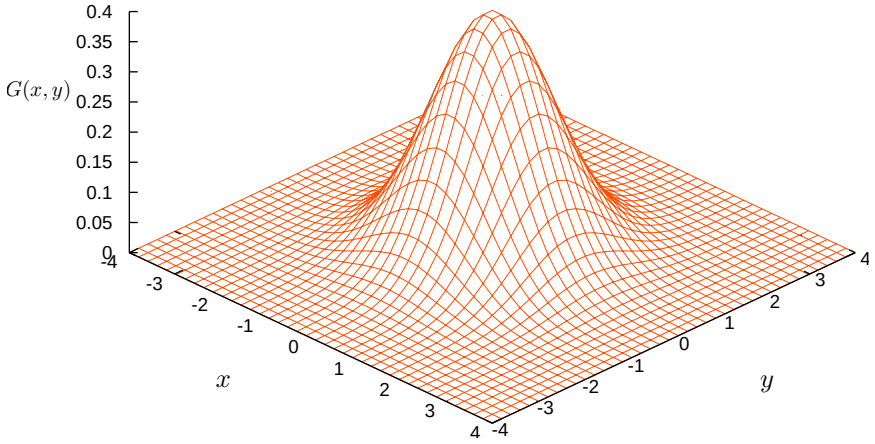


Figura 4.4: Distribución Gaussiana de dos dimensiones, con media $(0, 0)$ y $\sigma = 1$

$$\frac{1}{273} \times \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline \end{array}$$

Figura 4.5: Máscara Gaussiana 5×5, con $\sigma = 1$

4.1 se resumen los tres parámetros de éste algoritmo, que son el tamaño de la máscara, la varianza y el tipo de frontera empleado para la convolución.

Tabla 4.1: Parámetros del Reductor de Ruido Gaussiano

Parámetro	Rango de Valores
Tamaño de la máscara Gaussiana	3, 5, 7, ...
Varianza (σ^2)	0 - 100 (16 bits)
Tipo de frontera de la imagen	Constante, Periódica, Reflejada, Cero y Sin Frontera

4.1.3 Filtro de Mediana

El *filtro de mediana* es un filtro espacial no lineal de ordenamiento estadístico, cuya respuesta se basa en el ordenamiento de los píxeles contenidos en el área definida por el filtro. El filtro de mediana es uno de los filtros de ordenamiento estadístico más conocido,

y como su nombre lo indica, reemplaza el valor del píxel por la mediana de los valores del nivel de gris en el vecindario del ese píxel definido por el filtro. Matemáticamente se expresa de la siguiente manera

$$\widehat{\mathcal{I}}(i,j) = \texttt{mediana}_{(i,j)\in\mathcal{V}}\{\mathcal{N}(i,j)\} \tag{4.5}$$

donde $\mathcal{V}$ es el vecindario definido por el filtro de mediana. El valor original del píxel es incluido en el cálculo de la mediana. Los filtros de mediana tienen ventajas sobre el reductor de ruido Gaussiano, debido a que para cierto tipo de ruido como el impulsional tanto bipolar como unipolar son particularmente efectivos, además ésto se logra con una difuminación menos considerable que los filtros lineales.

La mediana, ξ, de un conjunto de valores es tal que la mitad de valores del conjunto son menores o iguales que ξ, y la otra mitad de valores son mayores o iguales que ξ. Para efectos de calcular la mediana, primero se ordenan ascendentemente los elementos del conjunto, que en éste caso son los valores del nivel de gris contenidos en el área definida por el filtro, posteriormente se define la mediana de ese conjunto y se asigna ese valor al píxel central del vecindario. Así por ejemplo, en un vecindario de 3×3 la mediana es el quinto valor más grande, en la Figura 4.6 se muestra un ejemplo numérico.

50	61	55
39	46	64
57	40	49

Valores del vecindario:
39,40,46,47,49,50,55,61,64

Mediana: 49

Figura 4.6: Cálculo de mediana, con una máscara de 3×3

En la Tabla 4.2 se presenta los parámetros de éste algoritmo que son el tamaño del vecindario en donde se aplica la mediana y el tipo de frontera empleado por el filtro.

Tabla 4.2: Parámetros del Filtro de Mediana

Parámetro	Rango de Valores
Tamaño de la máscara	3, 5, 7, 9, 11, 13, ...
Tipo de frontera de la imagen	Constante, Periódica, Reflejada, Cero y Sin Frontera

4.1.4 Reductor de Ruido SUSAN

SUSAN (con siglas en inglés *Smallest Univalue Segment Assimilating Nucleus*) es un enfoque no lineal presentado por Smith en [33], para el procesamiento de imágenes a bajo nivel, en relación con detección de bordes, detección de esquinas y reducción de ruido con preservación de estructuras; ésto último es el interés de éste trabajo. A continuación se explica el principio de SUSAN y luego se presenta el algoritmo de reducción de ruido.

Principio de SUSAN

Considere la Figura 4.7 que muestra un rectágulo oscuro en un fondo blanco. Una máscara circular que tiene como centro un píxel llamado *núcleo* se muestra en cinco posiciones. Si el brillo de cada píxel dentro de una máscara es comparado con el núcleo de esa máscara, entonces un área de la máscara puede ser definida con el mismo o similar brillo que el núcleo. Ésta área se llama *USAN* (que es el acrónimo en inglés para *Univalue Segment Assimilating Nucleus*). En la Figura 4.8, cada máscara de la Figura 4.7 es representada con su USAN en blanco.

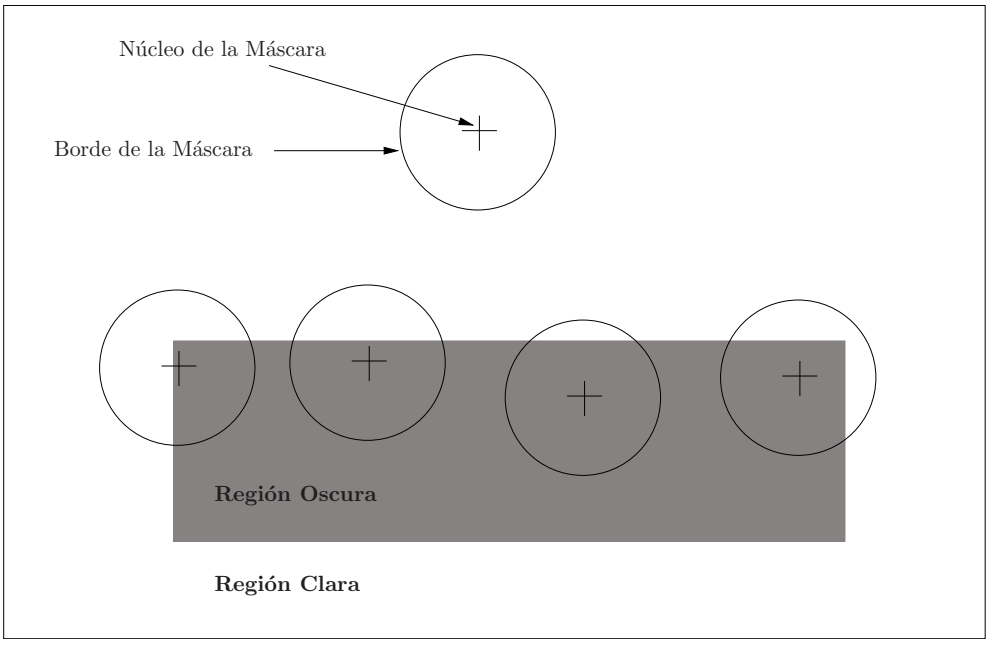


Figura 4.7: Cuatro máscaras circulares en diferentes lugares de una imagen [33].

Éste concepto de que cada píxel de la imagen tenga asociado una área con brillo similar es la base del principio de SUSAN: *Una imagen procesada para dar como salida el área USAN, tiene bordes y características en dos dimensiones fuertemente reforzadas, con éstas*

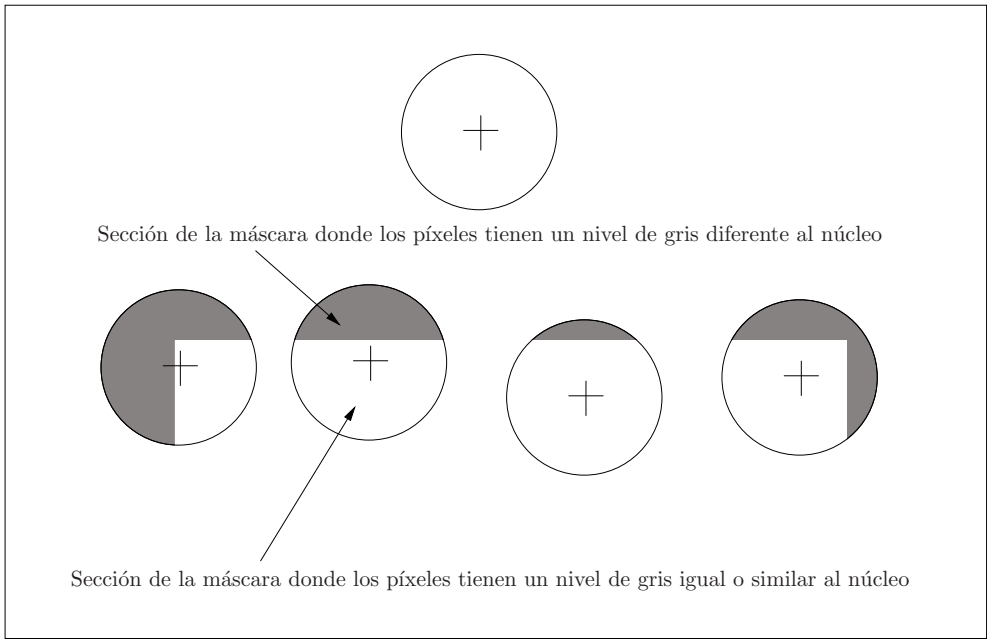


Figura 4.8: Cuatro máscaras circulares con los USAN's mostrados como partes blancas de las máscaras [33].

últimas con más énfasis que los bordes [33].

El hecho de que el mejoramiento de bordes y esquinas SUSAN no use derivadas de la imagen explica porqué el rendimiento en presencia del ruido es bueno. El efecto de integración del principio junto con su respuesta no lineal, da como resultado un fuerte rechazo al ruido. El algoritmo de reducción de ruido SUSAN se relaciona con el principio de SUSAN en que la USAN se emplea para escoger el mejor vecindario local de difuminado.

Algoritmo SUSAN para reducción de ruido

El algoritmo SUSAN para filtrado de ruido, como otros algoritmos preserva la estructura de la imagen, difuminando solo los vecinos que forman parte de la *misma región* que el píxel central. En general ésta región se asume constante en valor, sin embargo, éste algoritmo la asume aproximadamente constante.

El filtro SUSAN trabaja tomando el promedio sobre todos los píxeles que se encuentran en la USAN. Es evidente que ésto le dará el máximo número de los vecinos adecuados con los cuales se calculará el promedio, mientras que no implique vecinos de regiones no relacionadas. Así la estructura de la imagen se mantiene intacta.

El algoritmo SUSAN se relaciona con el operador gradiente inverso ponderado (4.6), que forma una media ponderada de los píxeles locales, en donde la ponderación depende de

la distancia entre el valor del píxel central y el valor de los píxeles locales

$$\widehat{\mathcal{I}}(i,j) = \frac{\sum \frac{\mathcal{N}(i+a,j+b)}{\max\{\frac{1}{2},|\mathcal{N}(i+a,j+b)-\mathcal{N}(i,j)|\}}}{\sum \frac{1}{\max\{\frac{1}{2},|\mathcal{N}(i+a,j+b)-\mathcal{N}(i,j)|\}}} \tag{4.6}$$

donde $\mathcal{N}(i,j)$ es la imagen original que se supone ruidosa, $\widehat{\mathcal{I}}(i,j)$ es la imagen filtrada, y la sumatoria es tomada sobre un vecindario de 3×3. La diferencia entre el operador gradiente inverso ponderado y el filtro SUSAN, radica en el hecho que el primero emplea el gradiente inverso para el ponderado de cada vecindario, a diferencia del SUSAN en donde la ponderación se deriva de la ponderación de la USAN, de la siguiente manera

$$c(\vec{r},\vec{r_0}) = e^{\left(\frac{\mathcal{N}(\vec{r})-\mathcal{N}(\vec{r_0})}{t}\right)^6} \tag{4.7}$$

Mientras que el método de gradiente inverso ponderado no usa *umbral de brillo* del todo, la ecuación de similitud de brillo (4.10) es fuertemente dependiente del umbral. Ésto tiene como ventaja que los píxeles que forma parte de USAN tienen ponderaciones similares, mientras que los que están fuera de la USAN tienen valores de ponderación de practicamente cero. Para efectos de la implementación del filtro SUSAN, se emplea una forma más suavizada de la ecuación de ponderación, de la siguiente manera

$$c(\vec{r},\vec{r_0}) = e^{\left(\frac{\mathcal{N}(\vec{r})-\mathcal{N}(\vec{r_0})}{t}\right)^2} \tag{4.8}$$

A parte de la diferencia en la ecuación de ponderación, otra diferencia de importancia entre el filtro SUSAN y el método gradiente inverso ponderado es que las sumas sobre el vecindario local no incluyen el píxel central (núcleo), ésto permite una reducción mucho mayor del ruido impulsional. Si el área de USAN es cero como podría suceder en el caso del ruido impulsional, el valor del píxel central (núcleo) se calcula mediante la mediana de los 8 píxeles más cercanos a éste.

Así la ecuación completa del filtro SUSAN es

$$\widehat{\mathcal{I}}(i,j) = \frac{\sum\limits_{(a,b)\neq(0,0)} \mathcal{N}(i+a,j+b) e^{-\frac{r^2}{2\sigma^2} - \frac{(\mathcal{N}(i+a,j+b)-\mathcal{N}(i,j))^2}{t^2}}}{\sum\limits_{(a,b)\neq(0,0)} e^{-\frac{r^2}{2\sigma^2} - \frac{(\mathcal{N}(i+a,j+b)-\mathcal{N}(i,j))^2}{t^2}}} \tag{4.9}$$

donde $r = \sqrt{a^2+b^2}$, σ controla la escala del difuminado espacial y t es el umbral de brillo que controla la escala de defuminado de brillo. Con respecto a la mejora de bordes en imágenes, es claro que en un borde difuminado los píxeles llevados a los vecindarios que son más cercanos en valor. Así, lejos de destruir la estructura de una imagen, el filtro

SUSAN mejora la calidad de la imagen. Los parámetros del algoritmo de reducción de ruido SUSAN son: el tamaño del vecindario que determina la calidad de la reducción de ruido y la velocidad de ejecución, el umbral t, el tipo de frontera y el factor de forma f de la siguiente forma exponencial

$$c(\vec{r}, \vec{r_0}) = e^{\left(\frac{\mathcal{N}(\vec{r})-\mathcal{N}(\vec{r_0})}{t}\right)^f} \tag{4.10}$$

Los parámetros del algoritmo reductor de ruido SUSAN se muestran en la Tabla 4.3

Tabla 4.3: Parámetros del Reductor de Ruido SUSAN

Parámetro	Rango de Valores
Tamaño de la máscara	3, 7
Umbral	0-15 (8 bits)
Factor de forma	2-6
Tipo de frontera de la imagen	Constante, Periódica, Reflejada, Cero y Sin Frontera

4.1.5 Medias no Locales

El algoritmo *Medias no Locales* (en inglés NL-Means) presentado por Buades et. al. en [9, 11, 12, 10, 8], aprovecha la alta redundancia de las imágenes naturales. En éste sentido se sabe que una pequeña ventana en una imagen natural tiene muchas otras ventanas similares en una misma imagen. Ahora en un sentido muy general, se puede definir como *vecindario de un píxel* p_m cualquier conjunto de píxeles p_n en una imagen, tal que una ventana alrededor de p_n se vea como una ventana alrededor de p_m, así los valores de el vecindario de p_n pueden ser empleados para predecir el valor de p_m. Ésto quiere decir que el vecindario de un determinado píxel no está aislado de otros vecindarios de la misma imagen, y es precisamente éste el principio del algoritmo de medias no locales.

Sea $\mathcal{N}$ una imagen ruidosa definida en una región delimitada $\mathbb{G}^2 \to \mathbb{R}$, sea $p \in \mathbb{G}^2$. El algoritmo medias no locales estima el valor de p como un promedio de los valores de todos los píxeles cuyos vecindarios Gaussianos sean similares al vecindario de p

$$NL(\mathcal{N})(p) = \frac{1}{C(p)} \int_{\mathbb{G}^2} e^{-\frac{(G_\sigma * |\mathcal{N}(x+.)-\mathcal{N}(y+.)|^2)(0)}{h^2}} \mathcal{N}(y)dy \tag{4.11}$$

donde G_σ es una máscara Gaussiana con una desviación estándar σ, h actúa como un

parámetro del grado de filtrado y $C(p)$ es el factor de normalización definido de la siguiente manera

$$C(p) = \int_{\mathbb{G}^2} e^{-\frac{(G_\sigma * |\mathcal{N}(x+.) - \mathcal{N}(y+.)|^2)(0)}{h^2}} \mathcal{N}(y) dy \tag{4.12}$$

Se debe recordar que

$$(G_\sigma * |\mathcal{N}(x+.) - \mathcal{N}(y+.)|^2)(0) = \int_{\mathbb{R}^2} G_\sigma(t) |\mathcal{N}(x+t) - \mathcal{N}(y+t)|^2 dt \tag{4.13}$$

Debido a que se está trabajando con imágenes digitales, es necesaria una descripción discreta del algoritmo que se desarrolla en la siguiente sección.

Descripción

Dada una imagen ruidosa $\mathcal{N} = \{\mathcal{N}(p_m) \mid p_m \in \mathbb{G}^2 \, , \, \mathcal{N} : \mathbb{G}^2 \to \mathbb{R}\}$, el valor estimado de $NL(\mathcal{N})(p_m)$ es calculado como el promedio ponderado de todos lo píxeles en la imagen

$$NL(\mathcal{N})(p_m) = \sum_{p_n \in \mathbb{G}^2} w(p_m, p_n) \mathcal{N}(p_n) \tag{4.14}$$

donde los pesos $\{w(p_m, p_n)\}_{p_n}$ dependen de la similitud entre los píxeles p_m y p_n y satisfacer las siguientes dos condiciones

$$0 \leq w(p_m, p_n) \leq 1 \tag{4.15}$$

y

$$\sum_{p_n} w(p_m, p_n) = 1 \tag{4.16}$$

Con el objetivo de calcular la simitud entre los píxeles de una imagen, se define un *sistema de vecindarios* en una imagen $\mathcal{I}$ como

Definición 4.1 (Sistema de Vecindarios) *Un sistema de vecindarios en una imagen $\mathcal{I} : \mathbb{G}^2 \to \mathbb{R}$ es una familia $\mathcal{V} = \{\mathcal{V}_{p_m}\}_{p_m \in \mathbb{G}^2}$ de subconjuntos de $\mathbb{G}^2$ tal que para todo $p_m \in \mathbb{G}^2$,*

1. $p_m \in \mathcal{V}_{p_m}$,
2. $p_n \in \mathcal{V}_{p_m} \Rightarrow p_m \in \mathcal{V}_{p_n}$

Además como se definió en la sección 2.1.1, $\mathcal{V}_{p_m}$ es un vecindario o ventada de similud de p_m.

Las ventanas de similitud pueden tener diferentes formas y tamaños para adaptarse mejor a la imagen. Por simplicidad se emplean ventanas cuadradas de tamaño fijo. La restricción de $\mathcal{N}$ a un vecindario $\mathcal{V}_{p_m}$ se denota como $\mathcal{N}(\mathcal{V}_{p_m})$

$$\mathcal{N}(\mathcal{V}_{p_m}) \Leftrightarrow \mathcal{N}(p_n), p_n \in \mathcal{V}_{p_m} \tag{4.17}$$

La similitud de dos píxeles p_m y p_n dependerá de la similitud de las intensidades del nivel de gris de los vectores $\mathcal{N}(\mathcal{V}_{p_m})$ y $\mathcal{N}(\mathcal{V}_{p_n})$, donde $\mathcal{V}_k$ denota un vecindario cuadrado de tamaño fijo centrado en k. Los píxeles con niveles de gris del vecindario similares a $\mathcal{N}(\mathcal{V}_{p_m})$ tendrán pesos mayores en el promediado. El algoritmo de medias no locales no solo compara el nivel de gris en un solo punto sino también que compara el vecindario completo. Como se muestra en la Figura 4.9 de Lena, p_{n_1} y p_{n_2} tienen un mayor peso porque sus ventanas de similitud son similares a la de p_m. Por otro lado, el peso $w(p_m, p_{n3})$ es mucho menor. Aunque el nivel de gris de p_{n_3} es similar al del píxel p_m, los valores de intensidad en las ventanas de similitud son muy diferentes.

Con el fin de calcular la similitud de intensidades del nivel de gris en los vectores $\mathcal{N}(\mathcal{V}_{p_m})$ y $\mathcal{N}(\mathcal{V}_{p_n})$, se calcula la distancia Euclideana Gaussiana Ponderada, $||\mathcal{N}(\mathcal{V}_{p_m}) - \mathcal{N}(\mathcal{V}_{p_n})||^2_{2,\sigma}$. Ahora ésta medida está más adaptada para cualquier ruido blanco aditivo tal que el ruido altere la distancia entre ventanas en una forma uniforme. La aplicación de la distancia Euclideana a vecindarios ruidosos plantea la siguiente igualdad

$$E||\mathcal{N}(\mathcal{V}_{p_m}) - \mathcal{N}(\mathcal{V}_{p_n})||^2_{2,\sigma} = ||\mathcal{I}(\mathcal{V}_{p_m}) - \mathcal{I}(\mathcal{V}_{p_n})||^2_{2,\sigma} + 2\sigma^2 \tag{4.18}$$

donde $\mathcal{I}$ y $\mathcal{N}$ son, respectivamente, las imágenes original y ruidosa, y σ^2 es la varianza del ruido. Ésta igualdad muestra la robustez del algoritmo debido a que la distancia Euclideana conserva el orden de similitud entre píxeles. Así, los píxeles con mayor similitud a p_m en $\mathcal{N}$ serán también los píxeles con mayor similitud a p_m en $\mathcal{I}$. Los pesos asociados con las distancias cuadráticas se definen como

$$w(p_m, p_n) = \frac{1}{Z(p_m)} e^{-\frac{||\mathcal{N}(\mathcal{V}_{pm}) - \mathcal{N}(\mathcal{V}_{pn})||}{h^2}} \tag{4.19}$$

donde $Z(p_m)$ es el factor de normalización

$$Z(p_m) = \sum_{p_n} e^{-\frac{||\mathcal{N}(\mathcal{V}_{pm}) - \mathcal{N}(\mathcal{V}_{pn})||}{h^2}} \tag{4.20}$$

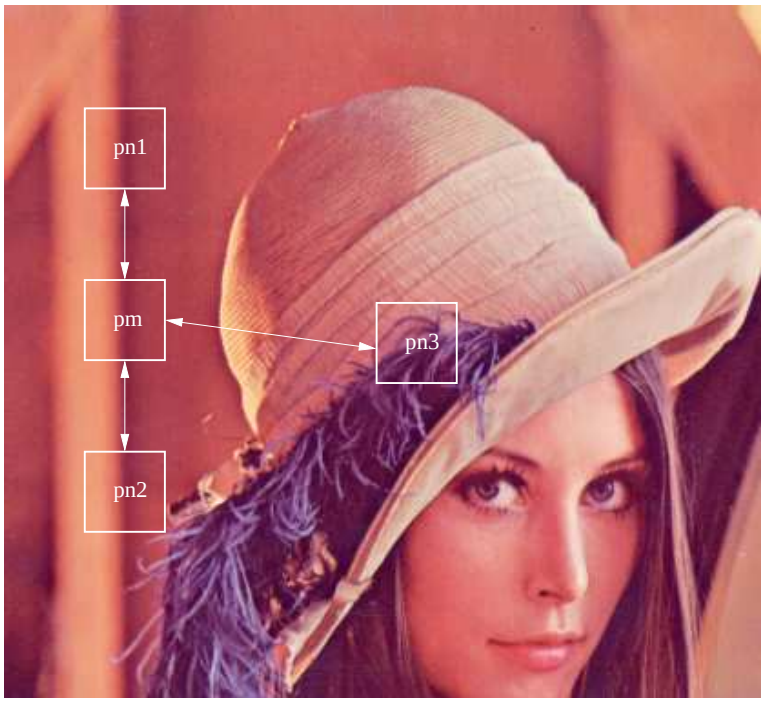


Figura 4.9: Esquema de la estrategía del algoritmo medias no locales. Vecindarios similares tienen un peso mayor, $w(p_m, p_{n_1})$ y $w(p_m, p_{n_2})$, que otros con vecindarios muy diferentes que tienen un peso mucho menor $w(p_m, p_{n_3})$

y h actúa como el grado de filtrado, que controla la caída de la función exponencial y así mismo la caída de los pesos como una función de la distancia Euclideana.

Versiones más rápidas del algoritmo

El algoritmo de medias no locales en principio presenta gran complejidad computacional, de tal manera que para su implementación se debe emplear una alternativa menos costosa en cuanto a tiempo de ejecución. Ésta consiste en establecer dos ventanas: la primera se llama *ventana de similitud* de tamaño $(2f+1)^2$, típicamente de 5×5 a 9×9 píxeles, ésta ya fue definida anteriormente en (4.17), y la segunda se llama *ventana de búsqueda* de tamaño $(2s+1)^2$, en la que se restringen las búsquedas de las ventanas de similitud; formalmente a partir de la definición del algoritmo de medias no locales en (4.14) ésta ventana de búsqueda es la imagen completa. Sin embargo, ésto es lo que hace costoso al algoritmo de tal manera que la ventana de búsqueda se restringe a un tamaño menor, típicamente de 11×11 a 21×21 píxeles. La complejidad del algoritmo bajo éstas condiciones es $N^2 \times (2f+1)^2 \times (2s+1)^2$, donde N^2 es el número de píxeles de la imagen. Es muy conveniente ampliar el tamaño de la ventana de búsqueda tanto como se pueda, sin

embargo, existe un compromiso con el tiempo de ejecución. A continuación se presentan dos enfoques para lograr lo anteriormente discutido.

- **Multiescala**

Lo discutido anteriormente se puede lograr con una estrategia multiescala. Éste algorimo se muestra en la Figura 4.10.

Algoritmo Multiescala

1: Se amplía la imagen u_0 por un factor de 2 mediante el procedimiento de submuestreo de Shannon estándar Ésto produce una nueva imagen u_1. Por conveniencia, se denotan los píxeles de u_1 como (i, j), y los píxeles de u_0 como $(2i, 2j)$.
2: Se aplica el algoritmo de medias no locales a u_1, así que con cada píxel (i, j) de u_1, una lista de ventanas centradas en $(i_1, j_1), ..., (i_k, j_k)$ está asociada.
3: Para cada píxel de u_0, $(2i + a, 2j + b)$ con $a, b \in \{0, 1\}$, se aplica el algoritmo de medias no locales. Pero en lugar de comparar con todas la ventanas en la zona de búsqueda, se compara solo con las nueve ventanas vecinas de cada píxel $(2i_l, 2j_l)$ para $l = 1, ..., k$.
4: Éste procedimiento puede aplicarse en una estructura piramidal, submuestreando u_1 en u_2, y así sucesivamente. De hecho, no es aconsejable aplicar una ampliación más de dos veces.[11]

Figura 4.10: Algoritmo multiescala para el mejoramiento del rendimiento del algoritmo de medias no locales

Mediante la aplicación por solo un factor de 2, el tiempo de cálculo se divide en aproximadamente 16.

- **Mediante Bloques**

Sea $\mathcal{N}$ una imagen ruidosa, donde $p_{m_k} \in \mathbb{G}^2$, $\mathcal{N} : \mathbb{G}^2 \to \mathbb{R}$. Para cada p_{m_k}, sea $W_k \subset \mathbb{G}^2$ la ventana de búsqueda centrada en p_{m_k}, $W_k = p_{m_k} + B_k$ donde B_k define el tamaño y la forma de la ventana. Supóngase que cada W_k es un subconjunto conectado de $\mathbb{G}^2$, tal que $\mathbb{G}^2 = W_1 \cup W_2 \cup ... \cup W_n$ y donde las intersecciones entre ventanas no deben ser nulas. Entonces para cada W_k se define el algoritmo de medias no locales como

$$NL(W_k)(p_{m_k}) = \sum_{p_n \in W_k} w(p_{m_k}, p_n)\mathcal{N}(p_n) \tag{4.21}$$

Los pesos están definidos como

$$w(p_k, p_n) = \frac{1}{C_k} e^{-\frac{||\mathcal{N}(p_{m_k}+B_k)-\mathcal{N}(p_n+B_k)||^2_{2,\sigma}}{h^2}} \tag{4.22}$$

donde C_k es el factor de normalización

$$C_k = \sum_{p_n \in W_k} e^{-\frac{||\mathcal{N}(p_{m_k}+B_k)-\mathcal{N}(p_n+B_k)||^2_{2,\sigma}}{h^2}} \tag{4.23}$$

y h actúa un parámetro del grado de filtrado. El hecho de que todas éstas ventanas se encuentren traslapadas permite una transición regular en la imagen restaurada. Ésta variante mediante bloques del algoritmo resulta en una mejor adaptación a la configuración local de la imagen y al mismo tiempo reduce el costo computacional. La implementación recién discutida se ilustra a continuación:

Algoritmo de Bloques

1: Sea $N \times N$ el tamaño de la imagen, y sea $p_{m_k} \in \mathbb{G}^2$, $\mathcal{N} : \mathbb{G}^2 \to \mathbb{R}$.
2: Se define la ventana de búsqueda $W_k = p_{m_k} + B_k$ de tamaño $(2s+1)^2$.
3: Para cada ventana de búsqueda W_k se aplica el algoritmo de medias no locales y se asigna el nuevo valor al píxel p_{m_k}.
4: Tomando la ventana de similitud de tamaño $(2f+1)^2$ y la ventana de búsqueda de tamaño $(2s+1)^2$ la complejidad del algoritmo es $(2f+1)^2 \times (2s+1)^2 \times N^2$. Que evidentemente es mucho menor que la complejidad inicial $(2f+1)^2 \times N^2 \times N^2$.

Figura 4.11: Algoritmo de bloques para el mejoramiento del rendimiento del algoritmo de medias no locales

Ésta alternativa de bloques fue la que se empleó en éste trabajo. En el Apéndice B.1 se presenta una referencia de la clase implementada que contiene el algoritmo medias no locales. En la Tabla 4.4 se muestran los parámetros del algoritmo de medias no locales y su respectivo rango de valores.

4.2 Algoritmos para el mejoramiento de contraste

En ésta sección se describen los algoritmos de mejoramiento de contraste que se compararon: *Ecualización de Histograma Global* y *Mejoramiento de Gradiente Local*, siendo éste último el que se implementó en éste trabajo.

Tabla 4.4: Parámetros del algoritmo de Medias no Locales

Parámetro	Rango de Valores
h	0-1
Tamaño de ventana de similitud	3,5,7,9
Tamaño de ventana de búsqueda	11,13,15,17,19,21
Varianza de la máscara Gaussiana	0-100
Tipo de frontera de la imagen	Constante, Periódica, Reflejada, Cero y Sin Frontera

4.2.1 Ecualización de Histograma

Antes de describir el algoritmo de equalización de histograma, se define el histograma.

Histograma

El histograma de una imagen digital con niveles de gris en el rango $[0, L-1]$, es una función discreta $h(r_k) = n_k$, donde r_k es el k-ésimo nivel de gris y n_k es el número de píxeles en la imagen que tienen el nivel de gris r_k. Es común en la práctica normalizar el histograma dividiendo cada uno de los valores por el total de píxeles, n, en la imagen. Así, un histograma normalizado está dado por $p(r_k) = n_k/n$, para $k = 0, 1, ..., L-1$; $p(r_k)$ brinda un estimado de la probabilidad de ocurrencia de un valor de nivel de gris r_k.

Los histogramas son la base de numerosas técnicas de procesamiento en el dominio espacial. La manipulación de histogramas puede ser usado efectivamente para el mejoramiento de imágenes. Adicionalmente, el histograma provee información estadística útil, para aplicaciones de procesamiento de imágenes como compresión y segmentación.

Considere la Figura 4.12, que es una imagen de un planeta, acá se muestran cuatro características básicas del nivel de gris en una imagen: imagen oscura, imagen clara, imagen con bajo contraste e imagen con alto contraste. El extremo derecho de ésta figura muestra los histogramas correspondientes a éstas imágenes. El eje horizontal de cada gráfico corresponde a valores de $h(r_k) = n_k$ o $p(r_k) = n_k/n$ si los valores son normalizados.

Como se muestra en la Figura 4.12(a), en el caso de la imagen oscura los componentes de su histograma se concentran en el extremo menor (oscuro) de la escala de grises, que además presenta bajo contraste. Similarmente, los componentes del histograma de la imagen clara, que además presenta bajo contraste se muestra en la Figura 4.12(b) se encuentran en el extremo mayor de la escala de grises. La imagen con bajo contraste

como se muestra en la Figura 4.12(c) tiene un histograma concentrado en el medio de la escala de grises. Finalmente, se observa en la Figura 4.12(d) que una imagen con alto contraste cubre un amplio rango de la escala de grises, y la distribución no está muy lejos de ser uniforme, con muy pocas líneas más altas que otras. Intuitivamente, es razonable concluir que una imagen cuyos píxeles tienden a ocupar el rango completo de los posibles valores de niveles de gris, y que además su distribución tiende a ser uniforme, tendrá una apariencia de alto contraste y exhibirá una amplia variedad de tonos de gris. El efecto neto será una imagen que muestra un amplio rango dinámico. Como se mostrará en el siguiente sección, es posible desarrollar una función de transformación que pueda lograr automáticamente éste efecto, basado solo en la información disponible del histograma de entrada.

Descripción de Ecualización de Histograma

En principio el principal objetivo de la ecualización de histograma es producir un histograma con una distribución uniforme, que como se comentó en la sección anterior éste tipo de distribución es una característica de imágenes con alto contraste.

Sea una variable continua r que representa los niveles de gris de una imagen a ser mejorada. Se asume que r ha sido normalizada en el intervalo $[0, 1]$, con $r = 0$ representando el negro y $r = 1$ representado el blanco. Posteriormente, se considerará un planteamiento discreto, en donde los valores de los píxeles estén en el intervalo $[0, L - 1]$. La siguiente es la forma de la función de transformación de interés

$$s = A(r) \quad 0 \leq r \leq 1 \tag{4.24}$$

que produce un nivel s para cada píxel r en la imagen original. Ésta función de transformación satisface las siguientes condiciones

1. $A(r)$ es de valor único y se incrementa monotónicamente en el intervalo $0 \leq r \leq 1$.
2. $0 \leq A(r) \leq 1$ para $0 \leq r \leq 1$.

El requisito en 1 que $A(r)$ sea de valor único se necesita para garantizar que la transformada inversa exista, y la condición monotónica preserva el incremento de negro a blanco en la imagen de salida. Una función de transformación que no se incrementa monotónicamente podría resultar en una imagen de salida con secciones invertidas, lo que no es deseable en éste caso. Finalmente, la condición 2 asegura que los niveles de gris de la salida se encontrarán en el mismo rango que los de la entrada. La transformación inversa

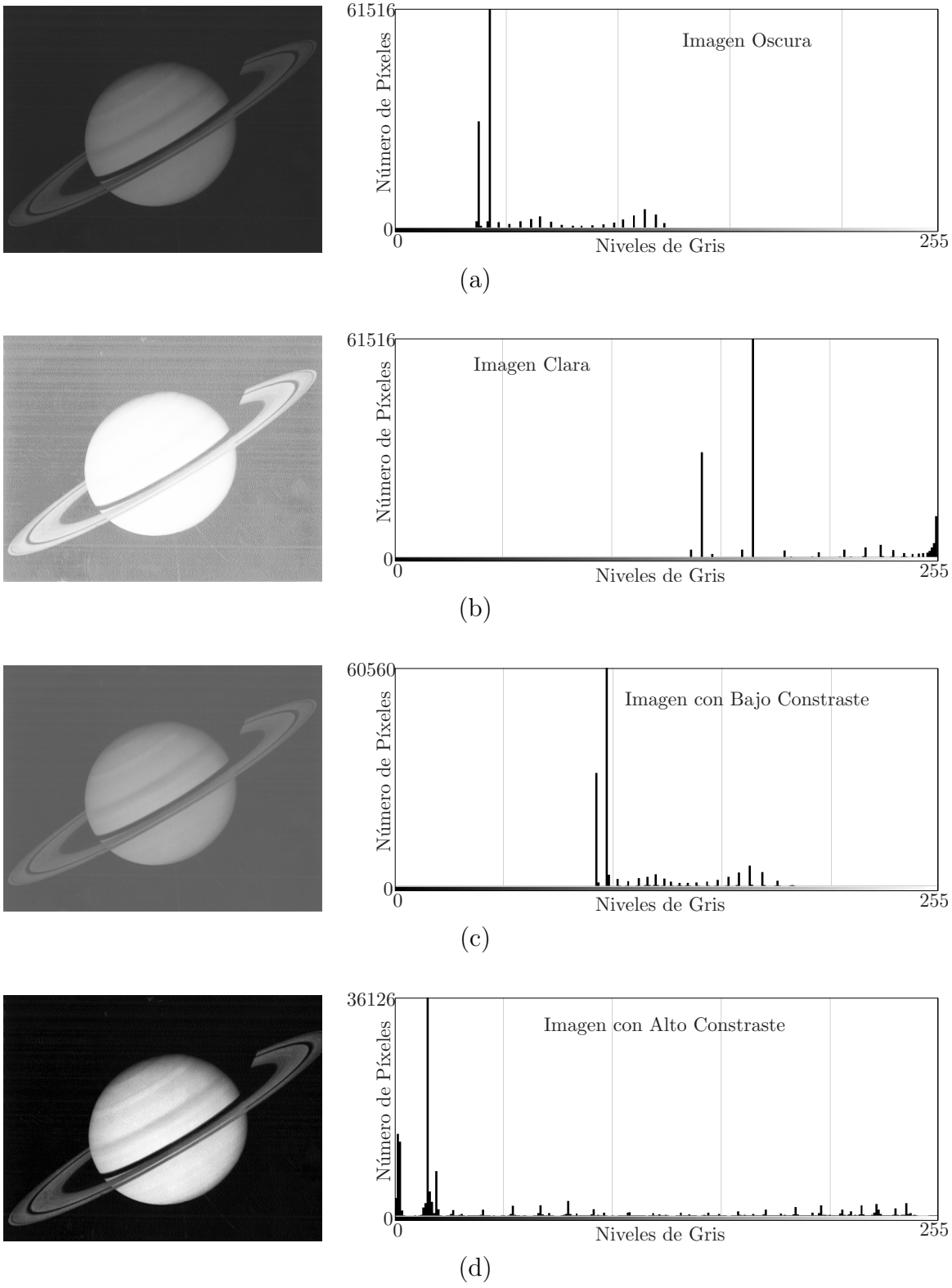


Figura 4.12: Cuatro tipos básicos de imagen con sus correspondientes histogramas: (a) imagen oscura, (b) imagen clara, (c) imagen con bajo contraste, (d) imagen con alto contraste

se denota de la siguiente manera

$$r = A(s)^{-1} \tag{4.25}$$

En la Figura 4.13, se muestra una función que satisface las dos condiciones discutidas anteriormente.

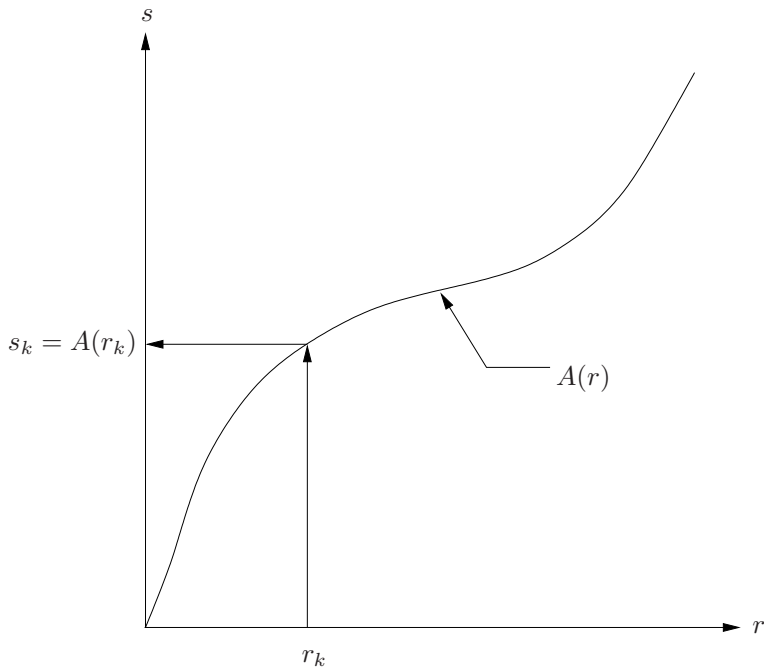


Figura 4.13: Función de transformación de nivel de gris que satisface las condiciones de valor único e incremento monotónico

Por otra parte en la Figura 4.14, se muestran algunas funciones de transformación típicas.

Los niveles de gris en una imagen pueden ser vistos como variables aleatorias en el intervalo $[0, 1]$. Como ya se discutió en la sección 4.1.1 uno de los principales descriptores de una variable aleatoria es la función de densidad de probabilidad (PDF). Sean $p_r(r)$ y $p_s(s)$ funciones de densidad de probabilidad de las variables aletorias r y s, respectivamente. A partir de la teoría de probabilidad se sabe que, si $p_r(r)$ y $A(r)$ son conocidos y $A^{-1}(s)$ satisface la condición 1, entonces la función de densidad de probabilidad $p_s(s)$ de la variable transformada s puede ser obtenida usando la siguiente relación

$$\begin{aligned} p_s(s)ds &= p_r(r)dr \\ p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \end{aligned} \tag{4.26}$$

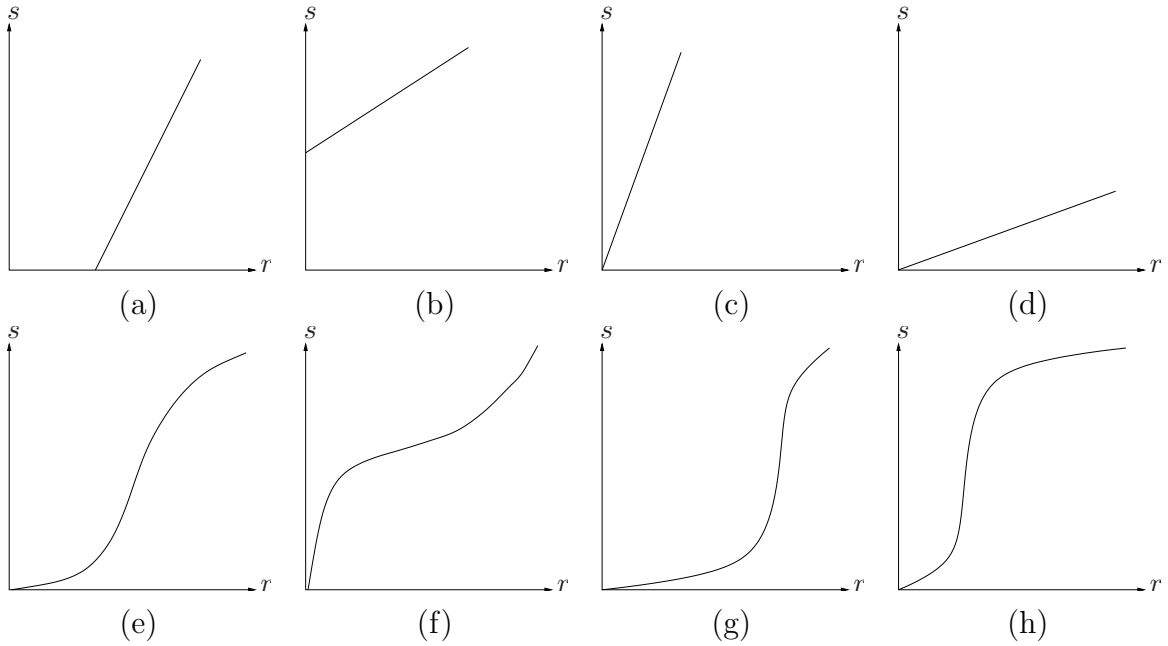


Figura 4.14: Funciones de transformación típicas: (a) oscurecimiento, (b) aclarado, (c) comprimido a los oscuros, (d) comprimido a los claros, (e) alto contraste, (f) bajo contraste, (g) enfatizado de sombras, (h) enfatizado de claros

Así, la densidad de probabilidad de la función de la variable transformada, s, está determinada por la PDF del nivel de gris de la imagen de entrada y la función de transformación elegida. Lo anterior se ilustra en la Figura 4.15.

Una función de transformación de particular importancia en procesamiento de imágenes tiene la siguiente forma

$$s = A(r) = \int_0^r p_r(w)dw \tag{4.27}$$

donde w es una variable de integración. El miembro derecho de la ecuación se le llama *función de distribución acumulada* (en inglés cumulative distribution function - CDF) de una variable aleatoria r.

Dada una función de transformación $A(r)$, $p_s(s)$ se encuentra aplicando (4.26). El desarrollo se muestra a continuación

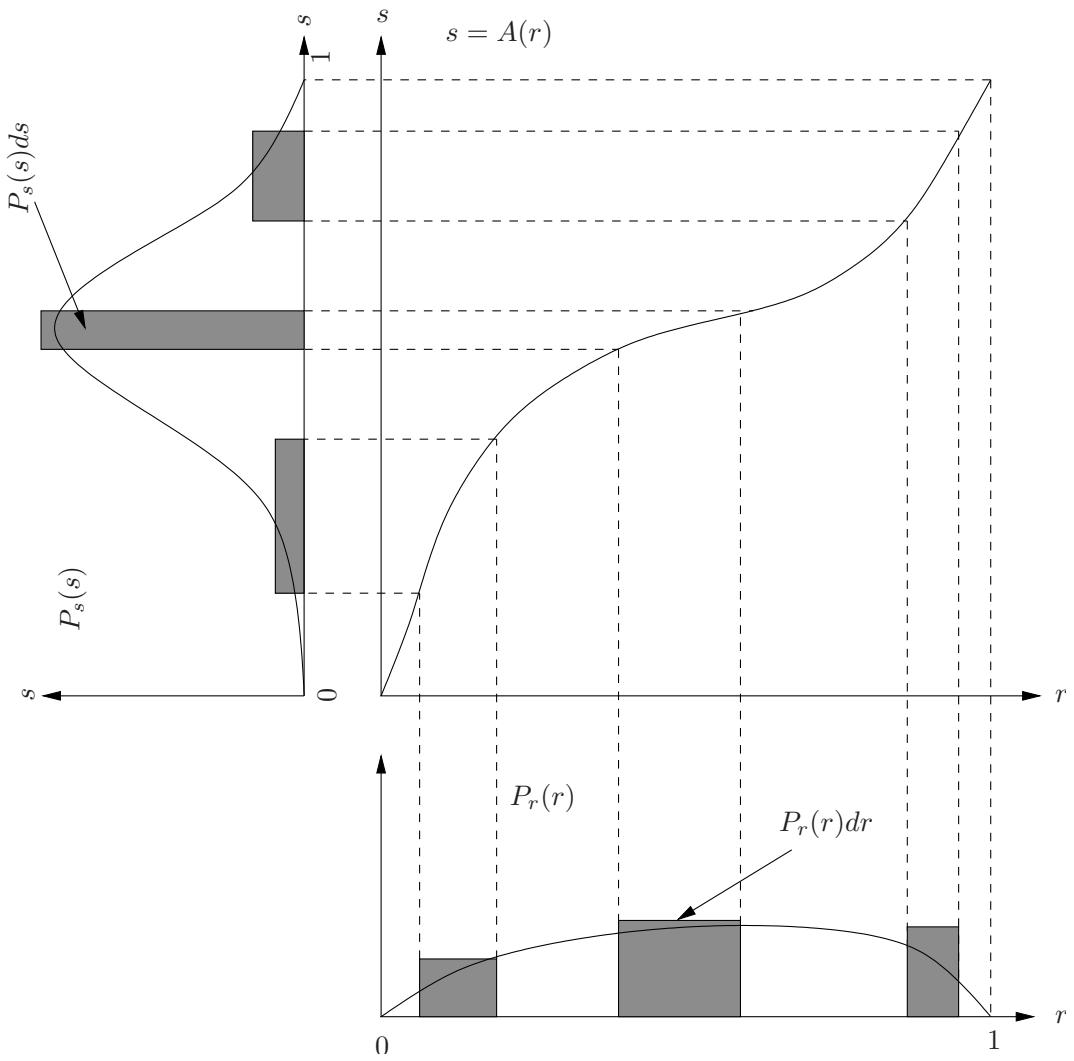


Figura 4.15: Mapeo de un píxel r a uno s mediante la función de transformación $s = A(r)$, en donde se cumple $P_s(s)ds = P_r(r)dr$

$$\begin{aligned} \frac{ds}{dr} &= \frac{dA(r)}{dr} \\ &= \frac{d}{dr}\left[\int_0^r p_r(w)dw\right] \\ &= p_r(r) \end{aligned} \tag{4.28}$$

Sustituyendo éste resultado por dr/ds en (4.26), y teniendo en cuenta que todos los valores de probabilidad son positivos, se obtiene

$$\begin{aligned} p_s(s) &= p_r(r)\left|\frac{dr}{ds}\right| \\ &= p_r(r)\left|\frac{1}{p_r(r)}\right| \\ &= 1 \quad 0 \leq s \leq 1 \end{aligned} \tag{4.29}$$

Debido a que $p_s(s)$ es la función de densidad de probabilidad, debe ser cero fuera del intervalo $[0, 1]$, de tal manera que la integral sobre todos los valores de s debe ser igual a 1. De la forma de $p_s(s)$ dada en (4.29) se observa que ésta es una función de densidad de probabilidad *uniforme*, y es precisamente ésto lo que se esperaba de la ecualización de histograma.

Para valores discretos que es el interés en procesamiento de imágenes, se emplea probabilidades y sumatorias, en lugar de densidades de probabilidad e integrales. La probabilidad de ocurrencia de un nivel de gris r_k en una imagen se aproxima mediante

$$p_r(r_k) = \frac{n_k}{n} \quad k = 0, 1, 2, ..., L-1 \tag{4.30}$$

donde, como se señaló anteriormente, n es el número total de píxeles en la imagen, n_k es el número de píxeles que tiene un nivel de gris r_k, y L es el número total de posibles niveles de gris en la imagen. La versión discreta de la función de distribución acumulada de (4.27) es

$$\begin{aligned} s_k &= A(r_k) = \sum_{j=0}^{k} p_r(r_j) \\ &= \sum_{j=0}^{k} \frac{n_j}{n} \quad k = 0, 1, 2, ..., L-1 \end{aligned} \tag{4.31}$$

Así, una imagen procesada es obtenida mediante el mapeo de cada píxel con el nivel r_k en la imagen de entrada en un píxel correspondiente con un nivel s_k en la imagen de salida mediante (4.31), en donde ésta ecuación se le llama *ecualización de histograma*. En la Tabla 4.5 se muestran los parámetros del algoritmo de ecualización de histograma.

Tabla 4.5: Parámetros del algoritmo de Ecualización de Histograma

Parámetro	Rango de Valores
Valor menor que es ecualizado desde la entrada	0 (0-1)
Valor mayor que es ecualizado desde la entrada	1 (0-1)
Valor menor que es ecualizado hacia la salida	0 (0-1)
Valor mayor que es ecualizado hacia de salida	1 (0-1)
Número de celdas que el histograma usa para la ecualización	256

4.2.2 Mejoramiento de Gradiente

La percepción humana del contraste se encuentra relacionada con el gradiente local de una imagen [39]. En éste sentido Subr et al. proponen en [35, 36] un algoritmo de mejoramiento local de contraste, logrado mediante la maximización de una función objetivo escalar que estima el promedio local de contraste en la imagen. Así, el objetivo de éste método es aumentar los gradientes locales. Además, se establecen dos condiciones que controlan la magnitud de la mejora en la imagen, y así mismo previene la saturación de niveles de gris en la imagen como resultado de la mejora. Las características más importantes de éste algoritmo se resumen de la siguiente manera:

- Se emplea una función objetivo escalar para *estimar* y *evaluar* el promedio local de contraste en una imagen.
- Para resolver el problema de maximización de la función objetivo se emplea un algoritmo voraz.
- Se establecen dos condiciones para éste algoritmo: la primera de ellas controla la magnitud del mejoramiento de contraste logrado, y la segunda asegura que en la imagen procesada no presente valores saturados en el nivel de gris.

El problema de optimización

La "mejora" de los gradientes locales de una imagen está sujeta a dos condiciones estrictas que previenen la saturación y un contraste excesivo. Así, la función objetivo a maximizar es la siguiente

$$f(\mathcal{N}) = \frac{1}{4|\mathbb{G}^2|} \sum_{p_m \in \mathbb{G}^2} \sum_{p_n \in \mathcal{V}_4(p_m)} \frac{\widehat{\mathcal{I}}(p_m) - \widehat{\mathcal{I}}(p_n)}{\mathcal{N}(p_m) - \mathcal{N}(p_n)} \tag{4.32}$$

sujeta a las siguientes condiciones

$$1 \leq \frac{\widehat{\mathcal{I}}(p_m) - \widehat{\mathcal{I}}(p_n)}{\mathcal{N}(p_m) - \mathcal{N}(p_n)} \leq (1 + \delta) \tag{4.33}$$

y

$$L \leq \widehat{\mathcal{I}}(p_m) \leq U \tag{4.34}$$

donde $\mathcal{N}(p_m)$ y $\widehat{\mathcal{I}}(p_m)$ representan los niveles de gris en el píxel p_m de las imágenes de entrada que se supone con bajo contraste y salida, respectivamente, donde $p_m \in \mathbb{G}^2$ tal que $\mathcal{N}, \widehat{\mathcal{I}} : \mathbb{G}^2 \rightarrow \mathbb{R}$. Así mismo $\mathcal{N}(p_n)$ y $\widehat{\mathcal{I}}(p_n)$ representan los niveles de gris en el píxel p_n de las imágenes de entrada y salida, respectivamente, donde p_n es un píxel vecino de p_m, $|\mathbb{G}^2|$ denota la cardinalidad de $\mathbb{G}^2$, $\mathcal{V}_4(p_m)$ denota la vecindad de cuatro de p_m, L y U son el límite menor y mayor de los niveles de gris ($L = 0$ y $U = 255$ en una imagen de 8 bits), respectivamente y $\delta > 0$ es el único parámetro que controla la cantidad de mejoramiento de contraste logrado. El objetivo de éste algoritmo es maximizar la función objetivo mediante el aumento del gradiente local alrededor de un píxel en la imagen de entrada en el mayor grado posible. La condición definida en (4.33) asegura un mejoramiento de gradiente delimitado. El límite menor asegura que los signos de los gradientes son preservados. El límite superior asegura un límite de mejoramiento de contraste controlado por el parámetro δ. Finalmente, la condición definida por (4.34) asegura que la imagen de salida no tendrá valores saturados de intensidad.

El Algoritmo

Para resolver el problema de optimización de la sección anterior se emplea un algoritmo voraz. Éste algoritmo se basa en el hecho de que dados dos píxeles vecinos con niveles de gris r y s, con $s \neq r$, cuando se escalan por un factor de $(1+\delta)$ resultan en r' y s', tal que

$$\frac{r' - s'}{r - s} = (1 + \delta) \tag{4.35}$$

Así, si simplemente se escalan los valores $\mathcal{N}(p_m)$, $\forall p_m \in \mathcal{N}$, por un factor de $(1 + \delta)$, se obtiene el valor máximo posible para $f(\mathcal{N})$. Sin embargo, esto podría violar (4.34) en p_m, saturando la intensidad en ese punto. Para evitar ésto, éste algoritmo adopta una estrategia iterativa, empleado un algoritmo voraz.

Para entender adecuadamente éste algoritmo se debe visualizar una imagen $\mathcal{N}$ en tres dimensiones en donde la base es una cuadrícula de $m \times n$, y cada punto de la cuadrícula tiene una altura proporcional al nivel de gris de ese píxel. Así la altura de cada píxel p_m se encuentra dentro del rango definido entre L y U.

Para cada iteración del algoritmo, el umbral u se encuentra en el rango $L \leq u \leq U$. Después, se genera una matriz (B) de $m \times n$, marcando regiones de $\mathcal{N}$ que se encuentran sobre el plano definido por el umbral u como

$$\mathbf{B}(i,j) = \begin{cases} 1 & \texttt{si}\ \mathcal{N}(i,j) > u \\ 0 & \texttt{si}\ \mathcal{N}(i,j) \leq u \end{cases} \tag{4.36}$$

Posteriormente, se localizan los componentes conectados que no son cero en $\mathbf{B}$, y se etiquetan como únicas. A estos componentes se llama *pequeñas colinas* o *colinas* (en inglés hillocks), y se denotan como h_k^u, donde k denota el número de componente y u denota el valor del umbral empleado para definir las colinas. Seguidamente, los píxeles en cada colina son escalados por una cantidad tal que no sobrepase U y al mismo tiempo el gradiente alrededor de ese píxel no se mejore por más del factor $(1+\delta)$. El factor de escala se elige individualmente para cada colina.

Éste método involucra un barrido sucesivo de planos definidos por umbrales u_k, tal que, $L \leq u_k \leq U$ y en cada barrido, se escalan las colinas respetando las dos condiciones definidas anteriormente. Nótese que como se barren planos sucesivos, una colina h_i^u se puede dividir en h_n^{u+1}, con $n = 0, 1, 2, ...$, o permanecer sin cambio. Pero, dos colinas h_l^u y h_t^u nunca se pueden fusionar para formar h_k^{u+1}. Éstos resultados indican que el proceso de definir planos en cada umbral es estrictamente creciente de una etapa a la siguiente, y además los píxeles examinados en una etapa son un subconjunto de los píxeles examinados en la etapa anterior. La observación anterior permite optimizar los gradientes de cada píxel basado en información almacenada acerca de qué tanto se ha escalado una determinada colina en iteraciones anteriores, así se puede asegurar que éste valor no exceda $(1+\delta)$.

Para los valores iniciales de u, el área abarcada por las colinas es amplia. Sin embargo, en éstos niveles el grado de mejoramiento que se logra no es tan amplio porque muchos de los píxeles ya se encuentran saturados. Conforme el valor de u se incrementa, las colinas conectadas se dividen y es posible mejorar aún más los gradientes de los píxeles.

Ésta primera etapa descrita anteriormente solo mejora las colinas locales de la imagen de entrada $\mathcal{N}$ con bajo contraste, generando una imagen que se denota como $\mathcal{N}_1$, de tal manera que para lograr un mejoramiento completo de la imagen se deben procesar también los valles. Para ello se debe invertir $\mathcal{N}_1$, mediante $U - \mathcal{N}_1$ para generar una imagen $\mathcal{N}_2$, y a ésta se le aplica la misma técnica que se le aplicó a $\mathcal{N}_1$, para así mejorar los valles también. Finalmente, se invierte $\mathcal{N}_2$, mediante $U - \mathcal{N}_2$ para obtener la imagen mejorada final.

En la Figura 4.16, se muestra el proceso desarrollado por éste algoritmo ilustrado en una dimensión por simplicidad. En la Figura 4.16(a), el umbral es cero así que solo hay una colina etiquetada como 1, que contiene todos los píxeles de la imagen de entrada. Ésta

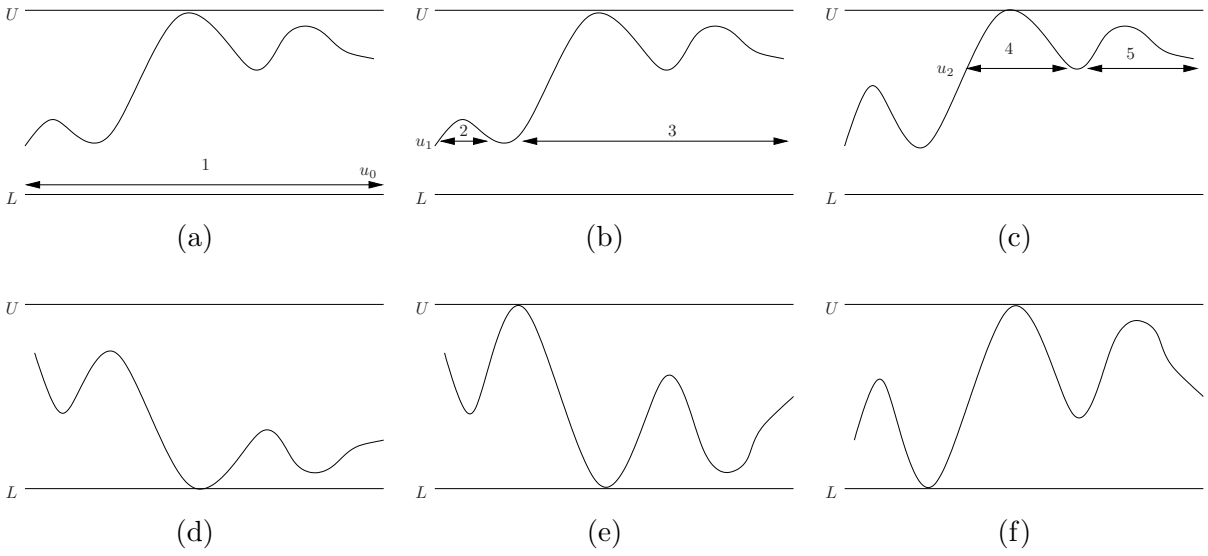


Figura 4.16: Proceso de mejoramiento de contraste en una dimensión aplicado por el algoritmo [36, 36]

colina ya está expandida, así que el píxel con el máximo valor ya alcanza la saturación. En la Figura 4.16(b), el umbral se incrementa a u_1 y así la colina 1, es dividida ahora en dos colinas 2 y 3. La colina 3 no puede ser expandida más debido a que el mayor píxel en valor ya está saturado. Sin embargo, la colina 2 puede ser expandida, de tal manera que el mejoramiento local de cada píxel alcance $(1 + \delta)$. Debido a que éste es la máxima magnitud de mejoramiento, la colina 2 no será expandida más. En la Figura 4.16(c), el umbral es u_2 y la colina 3 se divide en las colinas 4 y 5. En éste caso solo la colina 5 podrá ser mejorada debido a que la colina 4 ya se encuentra saturada. En la segunda pasada, la imagen de la Figura 4.16(c) se invierte y produce la Figura 4.16(d). Las colinas son procesadas y expandidas como en la primera pasada para producir la Figura 4.16(e). Finalmente, la Figura 4.16(e) es invertida para obtener la imagen final mejorada en la Figura 4.16(f).

Para resumir, las Figuras 4.17 y 4.18 muestran el pseudocódigo de las dos rutinas que componen el algoritmo de mejora de gradiente. En primera instancia, hay una rutina principal llamada *mejoraContraste*, que se muestra en la Figura 4.17, ésta rutina le aplica a una imagen de entrada el escalado de las colinas, luego invierte éste resultado para escalar los valles y finalmente, vuelve a invertir éste segundo resultado para obtener la imagen mejorada.

Por otra parte, una rutina secundaria que se muestra en la Figura 4.18 llamada *escalado*, se encarga de escalar las colinas. Ésta rutina hace un barrido a lo largo de todos los mínimos de la imagen, de tal manera que en cada plano se aplica el escalado a las colinas.

Ésta rutina almacena la historia del escalado para cada píxel de tal manera que no se sobrepase el valor $(1+\delta)$. El único parámetro del algoritmo de mejoramiento de gradiente es el factor *delta*. En la Sección B.2 se encuentra la referencia de la clase que se implementó con el algoritmo de mejoramiento de gradiente.

Algoritmo Principal: Mejora Contraste	
Algoritmo	mejoraContraste($\mathcal{N}$, δ, L, U)
Entradas:	Imagen de Entrada $\mathcal{N}$
	Parámetro: δ
Salida:	Imagen de Salida $\widehat{\mathcal{I}}$

Inicio
1: $\widehat{\mathcal{I}} \leftarrow \mathcal{N}$
2: $\widehat{\mathcal{I}} = escala(\widehat{\mathcal{I}}, \delta)$
3: $\widehat{\mathcal{I}} \leftarrow U - \widehat{\mathcal{I}}$
4: $\widehat{\mathcal{I}} = escala(\widehat{\mathcal{I}}, \delta)$
5: $\widehat{\mathcal{I}} \leftarrow U - \widehat{\mathcal{I}}$
3: **Retornar** $\widehat{\mathcal{I}}$
Fin

Figura 4.17: Pseudocódigo del algoritmo principal de mejoramiento de contraste

Algoritmo Secundario: Escala la magnitud de las colinas

Algoritmo	escala($\mathcal{N}$, δ, L, U)
Entradas:	Imagen de Entrada $\mathcal{N}$
	Parámetro: δ
Salida:	Imagen de Salida $\widehat{\mathcal{I}}$

Inicio

$\widehat{\mathcal{I}} \leftarrow \mathcal{N}$
$u = 0$
mientras $u < U$ **hacer**
    Obtenga la matriz binaria
    si $\mathcal{N}(i,j) \geq u$ **entonces**
        $\mathbf{B}(i,j) = 1$
    fin si
    Identifique el conjunto de colinas H en $\mathbf{B}$
    para cada $H_k \in H$ **hacer**
        encontar $p_{\max}$, $\widehat{\mathcal{I}}(p_{\max}) \geq \widehat{\mathcal{I}}(p) \quad \forall p \in H_k$
        $\delta_{\max} = min(\delta, (U-u))/(\widehat{\mathcal{I}}(p_{\max}) - u) - 1)$
        para cada $p \in H_k$ **hacer**
            $\delta_{aplicar} = \delta_{\max}$
            buscar la historia del escaldo de p, $\delta_h(p)$
            si $\delta_{aplicar} + \delta_h(p) > \delta$ **entonces**
                $\delta_{aplicar} = \delta - \delta_h(p)$
            fin si
            $\widehat{\mathcal{I}}(p) = (1 + \delta_{aplicar})(\widehat{\mathcal{I}}(p) - u) + u$
            $\delta_h = \delta_h(p) + \delta_{aplicar}$
        fin para
    fin para
    asignar a u el menor valor del píxel sobre el plano actual
fin mientras

Fin

Figura 4.18: Pseudocódigo del algoritmo secundario que escala el valor de las colinas

4.3 Algoritmos para el mejoramiento de nitidez

En ésta sección se describen dos variantes de un algoritmo de mejoramiento de nitidez llamado *enmascaramiento de desenfoque*.

4.3.1 Enmascaramiento de Desenfoque

El *enmascaramiento de desenfoque* es una técnica ampliamente utilizada para el mejoramiento de la nitidez en imágenes, que proviene de un proceso fotográfico, donde se crea una copia del negativo original por contacto sobre una película, situando un cristal fino de plata entre ambos, esto produce una copia positiva desenfocada por la difusión de la luz. Posteriormente, se sitúan ambas películas haciéndolas corresponder exactamente en un ampliador para reproducirlas en papel. Las áreas oscuras de la película positiva desenfocada, opuestas a las áreas claras del negativo original impedirán que la luz pase y así se sustraerá de la luz que pasa a través de la película original. Debido a que el positivo es intensionalmente difuminado, solo las bajas frecuencias se cancelarán, de tal manera que las altas frecuencias se enfatizarán y así se obtiene una imagen con mayor nitidez y detalle.

En el ámbito digital, muchas aplicaciones de software de manipulación de imágenes emplean ésta técnica para el mejoramiento de la nitidez. El nombre "enmascaramiento de desenfoque" se deriva del hecho de que ésta técnica sustrae una versión difuminada o "desenfocada" de la imagen original de la imagen en sí. Ésto se expresa de la siguiente manera

$$\widehat{\mathcal{I}}(i,j) = \mathcal{N}(i,j) - \mathcal{N}_d(i,j) \tag{4.37}$$

donde $\widehat{\mathcal{I}}(i,j)$ es la imagen mejorada en nitidez y $\mathcal{N}_d(i,j)$ es la versión difuminada de $\mathcal{N}(i,j)$, siendo ésta la imagen de entrada que se supone con baja nitidez. Una generalización del enmascaramiento de desenfoque es el *filtrado de alto impulso*. Una imagen filtrada con alto impulso, $\widehat{\mathcal{I}}(i,j)$, se define como

$$\widehat{\mathcal{I}} = C\mathcal{N} - \mathcal{N}_d \tag{4.38}$$

donde $C \geq 1$ es una constante. Reordenando se obtiene

$$\begin{aligned} \widehat{\mathcal{I}}(i,j) &= \mathcal{N}(i,j) * C\delta(i,j) - \mathcal{N}(i,j) * \mathcal{M}(m,n) \\ \widehat{\mathcal{I}}(i,j) &= \mathcal{N}(i.j) * (C\delta(i,j) - \mathcal{M}(m,n)) \end{aligned} \tag{4.39}$$

Así, para mejorar la nitidez se debe convolucionar la imagen $\mathcal{N}(i,j)$ con la máscara $(C\delta(i,j) - \mathcal{M}(m,n))$, donde $\mathcal{M}(m,n)$ es una máscara *gaussiana* o *laplaciana*. El caso

de la máscara gaussiana ya se discutió en la Sección 4.1.2, mientras que el laplaciano se describirá a continuación.

En general, una imagen difuminada se logra en el dominio espacial mediante el promediado de sus píxeles en un vecindario. Debido a que el promediado es análogo a la integración, se puede esperar que un mejoramiento de la nitidez en una imagen se puede lograr mediante la diferenciación. El operador *Laplaciano* para una función de dos variables (imagen), se define como

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial x^2} \tag{4.40}$$

Debido a que las derivadas de cualquier orden son lineales, el laplaciano es un operador lineal. Para efectos de procesamiento de imágenes es necesario expresar el laplaciano en una forma discreta. Sea una imagen $\mathcal{I}$ con $(i,j) \in \mathbb{G}^2$. Debido a que la imagen se compone de dos variables primero se define la derivada de segundo orden en la dirección i

$$\frac{\partial^2 \mathcal{I}(i,j)}{\partial i^2} = \mathcal{I}(i+1,j) + \mathcal{I}(i-1,j) - 2\mathcal{I}(i,j) \tag{4.41}$$

y de manera similar en la dirección j

$$\frac{\partial^2 \mathcal{I}(i,j)}{\partial j^2} = \mathcal{I}(i,j+1) + \mathcal{I}(i,j-1) - 2\mathcal{I}(i,j) \tag{4.42}$$

La implementación digital del laplaciano se obtiene sumando las derivadas de cada una de las dos direcciones, de la siguiente manera

$$\nabla^2 \mathcal{I}(i,j) = [\mathcal{I}(i+1,j) + \mathcal{I}(i-1,j) + \mathcal{I}(i,j+1) + \mathcal{I}(i,j-1)] - 4\mathcal{I}(i,j) \tag{4.43}$$

Las direcciones diagonales pueden ser incorporadas agregando dos términos más a (4.43), uno para cada diagonal. Debido a que cada uno de los términos diagonales contienen también un término $-2\mathcal{I}(i,j)$, el total sustraído de los términos de diferencia es ahora $-8\mathcal{I}(i,j)$. En la Figura 4.19(a) se muestra la máscara laplaciana definida en (4.43) y en la Figura 4.19(b) se muestra la máscara laplaciana con los términos diagonales incluidos.

Ahora bien, como se definió en (4.39) para el caso de alto impulso, una imagen nítida se logra mediante la convolución de ésta con $(C\delta(i,j) - \mathcal{M}(m,n))$, donde $\mathcal{M}(m,n)$ es una máscara gaussiana o laplaciana. Así, en la Figura 4.20 se muestra la máscara Laplaciana para el caso de alto impulso. En la Sección B.3 se encuentra la referencia de la clase que contiene las máscaras laplaciana y gaussiana, y en la Sección B.4 se encuentra la referencia de la clase que contiene el algoritmo de enmascaramiento de desenfoque. En la Tabla 4.6 se muestra los parámetros del algoritmo de enmascaramiento de desenfoque.

0	1	0
1	-4	1
0	1	0

(a)

1	1	1
1	-8	1
1	1	1

(b)

Figura 4.19: Máscaras Laplacianas: (a) sin términos diagonales (b) con términos diagonales

0	-1	0
-1	C+4	-1
0	-1	0

(a)

-1	-1	-1
-1	C+8	-1
-1	-1	-1

(b)

Figura 4.20: Máscaras laplacianas para alto impulso: (a) sin términos diagonales (b) con términos diagonales

Tabla 4.6: Parámetros del algoritmo de Máscara de Desenfoque

Parámetro	Rango de Valores
Tipo de máscara	Laplaciana, Gaussiana
Tamaño de la máscara	3,7,9,11,...
Constante de alto impulso	1-10
Tipo de vecindario (Para la máscara Laplaciana)	Vecindario de 4 u 8
Varianza (Para la máscara Gaussiana)	0-100

Capítulo 5

Análisis de Resultados

En éste capítulo se presentan los resultados que se obtuvieron a partir de la evaluación de los algoritmos de reducción de ruido, mejoramiento de contraste y mejoramiento de nitidez. El eje de ésta evaluación es el *Frente de Pareto*, que fue discutido en el Capítulo 3. Además, se mostrarán imágenes mejoradas mediante los algoritmos para ilustrar los resultados expresados en el Frente de Pareto. Para efectos de ésta evaluación se empleó una computadora con una procesador Intel de doble núcleo y con 3GB de memoria RAM. En la Figura 5.1 se muestra un ejemplo de una imagen de gel de electroforesis empleada para evaluar los algoritmos.

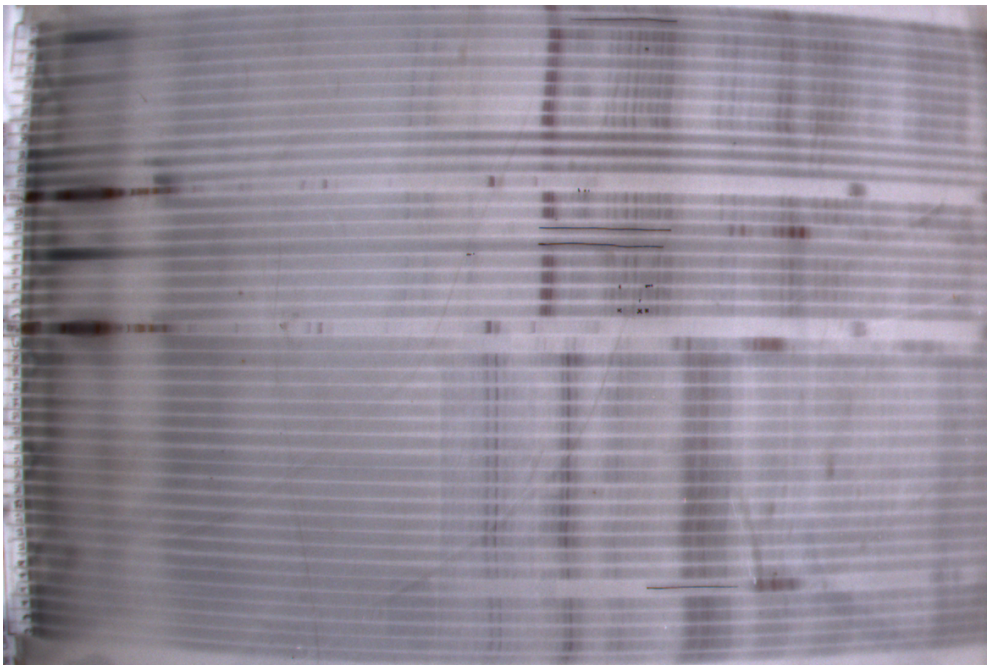


Figura 5.1: Imagen de gel de electroforesis empleada para la evaluación de los algoritmos.

5.1 Reducción de ruido

Los algoritmos de reducción de ruido que se compararán son: el *reductor de ruido gaussiano*, el *filtro de mediana*, el *reductor de ruido SUSAN* y *medias no locales*. Para evaluarlos, se le aplicó ruido gaussiano y blanco con desviaciones estándar de: $\sigma = 0,01$, $\sigma = 0,05$ y $\sigma = 0,1$ a una imagen de gel de electroforesis con una resolución de 354×256 y con un rango de niveles de gris entre $0,065$ y $0,684$, ésto con el objetivo de estudiar el comportamiento de los métodos ante diferentes niveles de ruido. Adicionalmente, se evaluó el desempeño de los algoritmos ante el ruido impulsional con una cobertura de 10% de la imagen.

Para ilustrar el efecto de reducción de ruido de los diferentes algoritmos, se presentarán imágenes con ruido y su correspondiente imagen mejorada con cada algoritmo, esto con el fin de realizar una comparación visual entre las imágenes procesadas por cada algoritmo. Adicionalmente, se mostrará la imagen del ruido de método que consiste en sustraer la imagen filtrada $\widehat{\mathcal{I}}$ de la imagen original $\mathcal{N}$. La imagen resultante de ésta resta debe ser idealmente solo el ruido eliminado por el algoritmo. Sin embargo, si los algoritmos de reducción de ruido eliminan características y detalles geométricos de la imagen original, en la imagen del ruido de método se observará dicho efecto. Lo mostrado cualitativamente por el ruido de método se expresa cuantitativamente en una medida que se le llamó *ruido de método escalar*. Así, las medidas de aptitud para obtener el Frente de Pareto de los algoritmos de ruido son: el *error cuadrático medio inverso* (f_{ecmi}), que mide cuanto reduce el ruido un algoritmo y el *ruido de método escalar* (f_{rme}), que mide cuanta distorsión introducen los algoritmos a las imágenes procesadas.

Ruido Gaussiano con $\sigma = 0,01$

Los Frentes de Pareto obtenidos para los algoritmos de reducción de ruido a partir de una imagen de gel de electroforesis con un ruido gaussiano y blanco con $\sigma = 0,01$ se muestran en la Figura 5.2. De ésta se observa que el Frente de Pareto del algoritmo de medias no locales dominó a los frentes de los demás algoritmos desde un error cuadrático medio de 10000 hasta aproximadamente 27000. Sin embargo, de un error cuadrático medio inverso de 27000 hasta más de 30000 dominó el reductor de ruido gaussiano. Por otra parte, el Frente de Pareto del reductor de ruido SUSAN abarcó un rango de ruido de método escalar de aproximadamente 8250 hasta 10000. En éste sentido el desempeño con respecto a la distorsión del reductor de ruido SUSAN es comparable con el reductor de ruido gaussiano y el algoritmo medias no locales, debido a que su Frente de Pareto abarca un rango ruido de escalar similar, a pesar de ello posee algunos puntos inferiores a los mínimos alcanzados por éstos algoritmos. Sin embargo, lo mismo no sucedió con

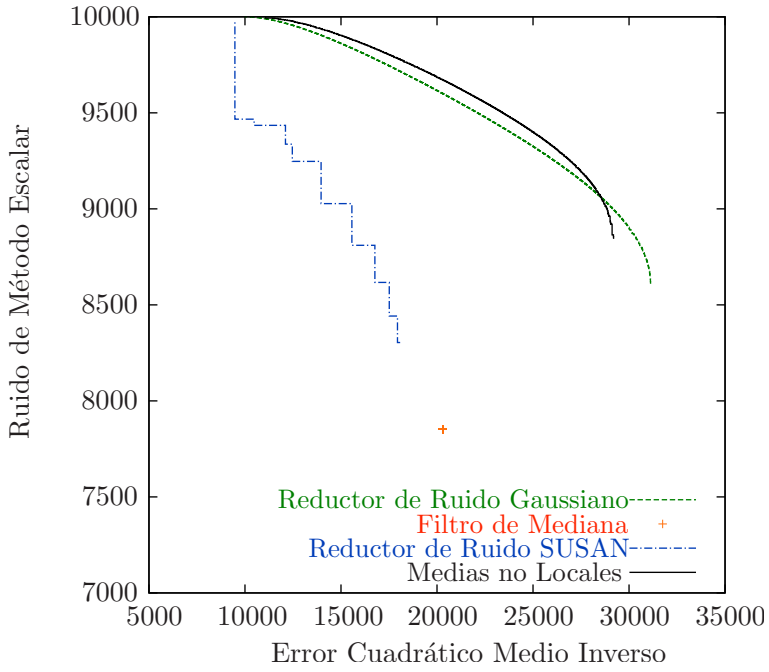


Figura 5.2: Frentes de Pareto de los algoritmos de reducción de ruido con $\sigma = 0,01$. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Ruido de Método Escalar* (f_{rme}).

respecto a la reducción del error cuadrático medio. Finalmente, el Frente de Pareto del filtro de mediana está conformado por un solo punto, que en general está dominado por los Frentes de los demás algoritmos. A pesar de ésto el filtro de mediana logró disminuir el error cuadrático medio más que cualquier punto del frente del reductor de ruido SUSAN. El hecho de que el Frente del filtro de mediana sea un único punto, se debe a que éste algoritmo cuenta con un solo parámetro que es el tamaño de la ventana, que además cuenta pocos posibles valores. Por lo tanto, es de esperar que el Frente del filtro de mediana se caracterice por pocos puntos.

En la Figura 5.3(a) se muestra segmento de una imagen de un gel con un ruido gaussiano y blanco con $\sigma = 0,01$, que es un nivel de ruido que visualmente casi no se percibe. Para comparar el efecto de reducción de ruido de cada uno de los algoritmos sobre la imagen se eligió un punto aproximadamente constante de ruido de método escalar en 9500, con excepción del frente del filtro de mediana que posee un solo punto en aproximadamente $f_{rme} = 7900$. Al comparar las imágenes procesadas por los diferentes algoritmos de reducción de ruido de las Figuras 5.3(b) a la 5.3(e), se observa que la imagen con menor ruido es la procesada por el algoritmo de medias no locales, mostrada en la Figura 5.3(e), con un valor de $f_{ecmi} = 24709$. Por otra parte, se encontró que el algoritmo que le tomó más tiempo procesar la imagen fue al algoritmo de medias no locales, razón por la que es preferible utilizar el reductor de ruido gaussiano debido a que es más rápido y con resultados similares. En general, los parámetros empleados por los diferentes algoritmos

fueron los mínimos, lo que es congruente con el bajo nivel de ruido aplicado. Así, por ejemplo el reductor de ruido gaussiano empleó una máscara de 3×3 píxeles, que es el mínimo valor, con una baja varianza de $0,26$. En el caso del filtro de mediana y del reductor de ruido SUSAN, se empleó igualmente el tamaño de máscara mínimo, 3×3 píxeles. Finalmente, en el algoritmo de medias no locales se empleó una ventana de similitud de 3×3 píxeles, una ventana de búsqueda de 5×5 píxeles y un grado de filtrado de $0,001$, que son valores relativamente bajos.

Tabla 5.1: Medidas de aptitud de las imágenes de la Figura 5.3

Algoritmo	Error Cuadrático Medio Inverso (f_{ecmi})	Método de Ruido Escalar (f_{rme})	Tiempo de Ejecución (s)
Reductor de Ruido Gaussiano	23868	9421	$0,01$
Filtro de Mediana	20793	7899	$0,02$
Reductor de Ruido Susan	12242	9351	$0,01$
Medias no Locales	24709	9360	$0,20$

Tabla 5.2: Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.3

Algoritmo	Parámetro	Valor
Reductor de Ruido Gaussiano	Tamaño de la máscara	3
	Varianza	0,27
	Tipo de frontera	Constante
Filtro de Mediana	Tamaño de la máscara	3
	Tipo de frontera	Constante
Reductor de Ruido SUSAN	Tamaño de la máscara	3
	Umbral	4
	Factor de forma	4
	Tipo de frontera	Cero
Medias no Locales	Grado de filtrado (h)	0.0005
	Tamaño de la ventana de similud	3
	Tamaño de la ventana de búsqueda	5
	Varianza de la máscara Gaussiana	0,75
	Tipo de frontera	Espejo

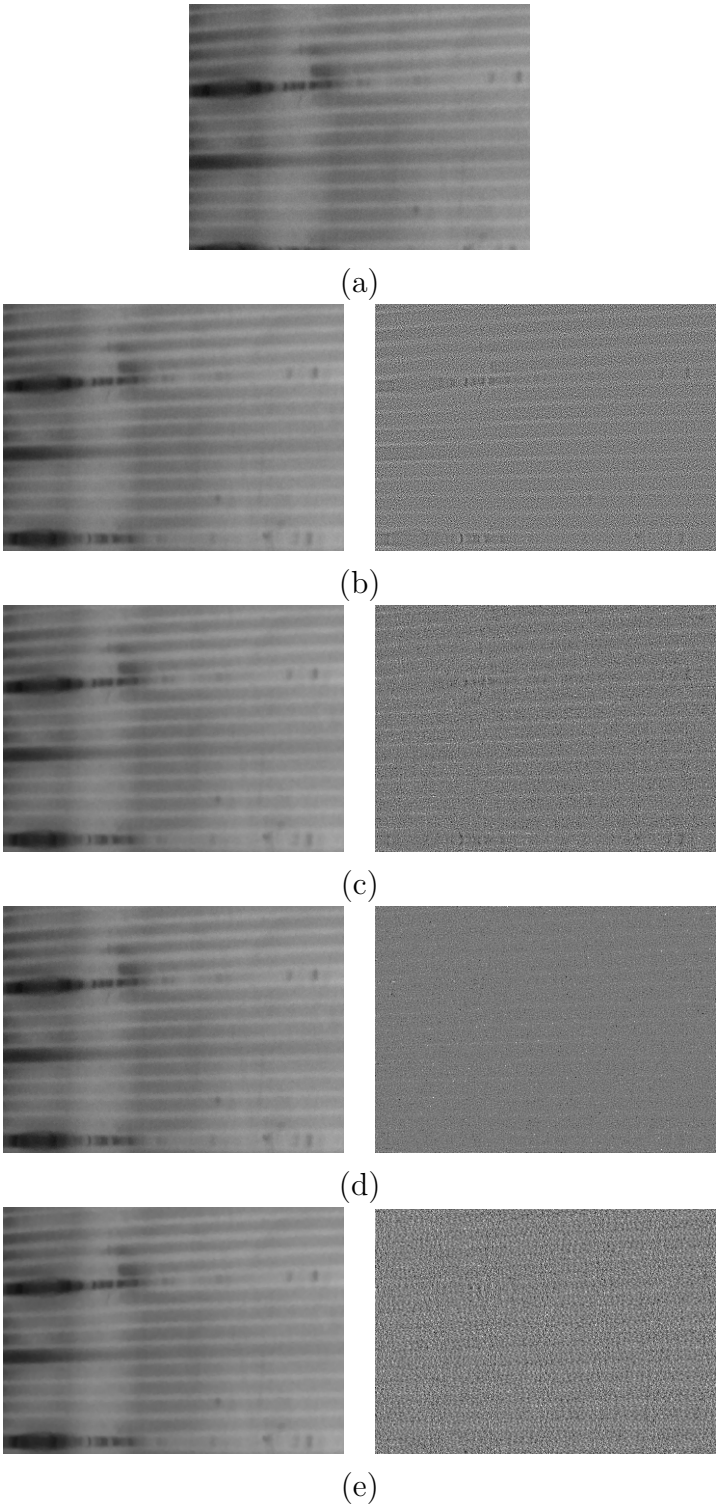


Figura 5.3: Imágenes procesadas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original $\mathcal{N}$ con un ruido gaussiano con $\sigma = 0,01$, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales.

Ruido Gaussiano con $\sigma = 0,05$

Los frentes de Pareto obtenidos para los algoritmos de reducción de ruido a partir de una imagen de gel de electroforesis con un ruido gaussiano con $\sigma = 0.05$ se muestran en la Figura 5.4. Al comparar éstos Frentes con los obtenidos con un ruido con $\sigma = 0.01$, mostrados en la Figura 5.2, se observa cómo el rango de valores de la medida de aptitud del ruido de método escalar se mantiene, ya que, ésta mide la distorsión de las imágenes procesadas por los algoritmos y es independiente de la cantidad de ruido. Por otra parte, el rango de la medida de aptitud del error cuadrático medio inverso disminuyó, que es acorde con el aumento del ruido.

Al comparar los Frentes de Pareto de la Figura 5.4, se observa que el frente del algoritmo de medias no locales domina claramente el resto de algoritmos. Además, tanto el frente del algoritmo de medias no locales como el reductor de ruido gaussiano presentan la misma tendencia cóncava. Por otra parte, el reductor de ruido SUSAN se caracteriza al igual que en el caso del ruido con desviación estándar de $0,01$, por abarcar un rango similar de ruido de método escalar similar a los algoritmos de medias no locales y reductor de ruido gaussiano, pero ésto con un alto error cuadrático medio. En contraposición al reductor de ruido SUSAN, el frente del filtro de mediana está por debajo con respecto al ruido de método escalar, pero con un menor error cuadrático medio en todos los puntos de su frente.

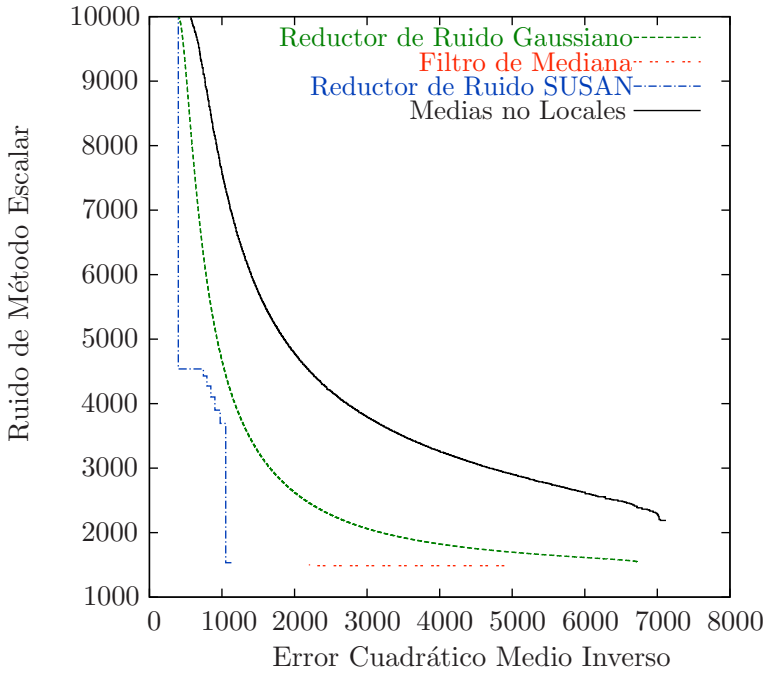


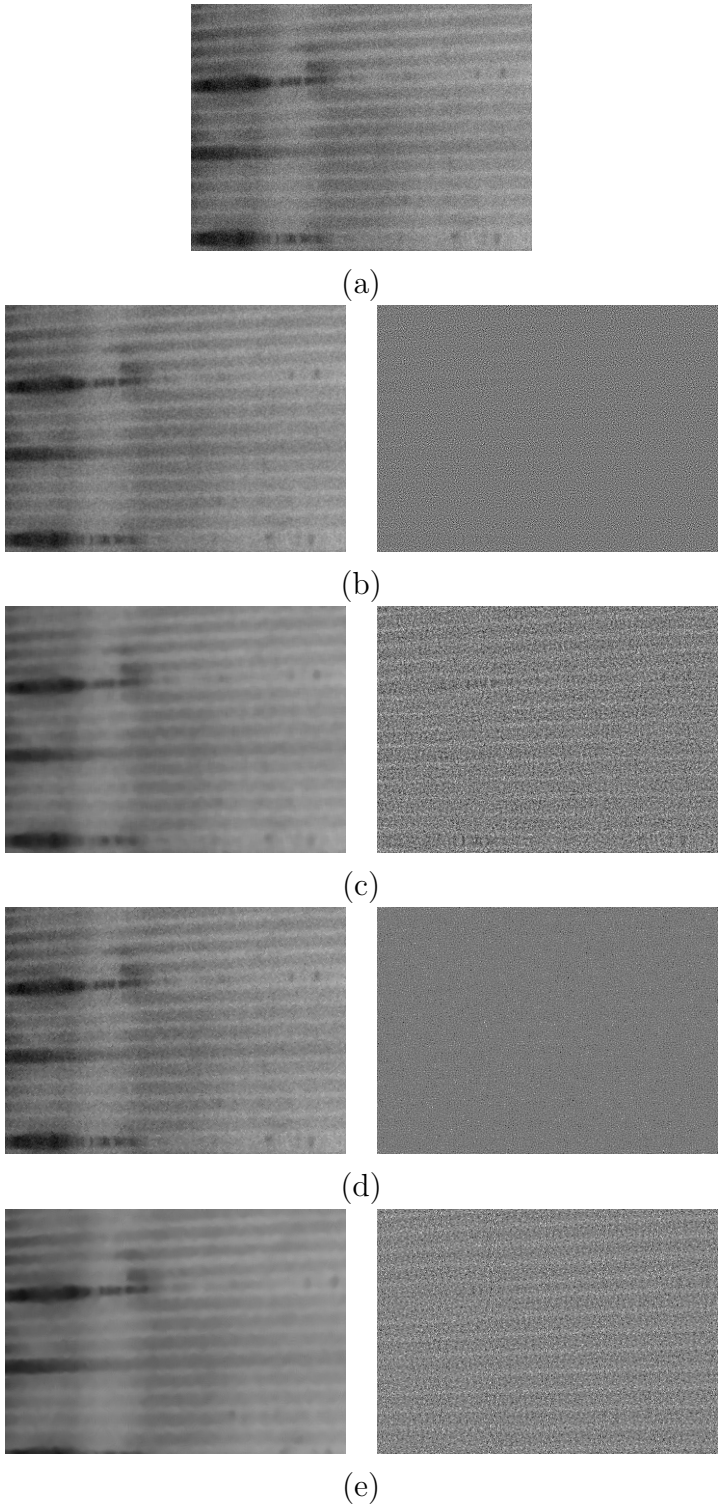
Figura 5.4: Frente de Pareto de los algoritmos de reducción de ruido con $\sigma = 0,05$. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Método de Ruido Escalar* (f_{mre}).

En la Figura 5.5(a) se muestra un segmento de una imagen de un gel con un ruido gaussiano con $\sigma = 0,05$. Para observar las reducción de ruido en las imágenes por parte de los algoritmos se comparan los algoritmos con valor de ruido de método escalar con un valor de aproximadamente 3500, a excepción del filtro de mediana que presenta un único valor de ruido de método escalar de aproximadamente 1500. De la comparación de las Figuras 5.5(b) a la 5.5(e), se encontró que para un valor constante de ruido de método escalar constante el algoritmo de medias no locales redujo en mayor cantidad el ruido. En la Tabla 5.4 se muestran los parámetros empleados por los diferentes algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.5. Además, en la Tabla 5.3 se resumen las medidas de aptitud obtenidas con las imágenes de la Figura 5.5.

Por otra parte, con respecto a los parámetros empleados por los diferentes algoritmos para obtener los Frentes de Pareto de la Figura 5.4 se encontró, que en general aumentaron sus valores respecto al caso del ruido con $\sigma = 0,01$. Ésto responde precisamente al aumento de la desviación estándar de ruido. Así por ejemplo, el reductor de ruido gaussiano aumentó el tamaño de las máscaras en un rango entre 7×7 y 17×17, y la desviación estándar aumentó a un rango entre 1 y 4. En el caso del filtro de mediana el tamaño de la máscara aumentó a un rango entre 3×3 y 9×9. En el caso del reductor de ruido SUSAN el factor de forma se encontró en un rango entre 4 y 10, mientras que el umbral abarcó el rango de 1 y 15. Finalmente, en el algoritmo de medias no locales el tamaño de la máscara de similitud aumentó a un rango entre 5×5 y 9×9, la ventana de búsqueda aumentó a un rango entre 9×9 y 11×11, la varianza se encontró en un rango entre $0,1$ y 10, y el grado de filtrado aumentó en un orden de magnitud, de tal manera que los valores se encontraron en un rango comprendido entre $0,001$ y $0,003$.

Tabla 5.3: Medidas de aptitud de las imágenes de la Figura 5.5

Algoritmo	Error Cuadrático Medio Inverso (f_{ecmi})	Método de Ruido Escalar (f_{rme})	Tiempo de Ejecución (s)
Reductor de Ruido Gaussiano	1428	3436	$0,01$
Filtro de Mediana	4890	1498	$0,12$
Reductor de Ruido Susan	1044	3751	$0,01$
Medias no Locales	3418	3544	$1,3$



(a)

(b)

(c)

(d)

(e)

Figura 5.5: Imágenes filtradas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original $\mathcal{N}$ con un ruido gaussiano con $\sigma = 0,05$, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales.

Tabla 5.4: Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.5

Algoritmo	Parámetro	Valor
Reductor de Ruido Gaussiano	Tamaño de la máscara	9
	Varianza	$0,33$
	Tipo de frontera	Constante
Filtro de Mediana	Tamaño de la máscara	7
	Tipo de frontera	Constante
Reductor de Ruido SUSAN	Tamaño de la máscara	3
	Umbral	15
	Factor de forma	2
	Tipo de frontera	Cero
Medias no Locales	Grado de filtrado (h)	$0,0036$
	Tamaño de la ventana de similud	7
	Tamaño de la ventana de búsqueda	9
	Varianza de la máscara Gaussiana	$0,39$
	Tipo de frontera	Constante

Ruido Gaussiano con $\sigma = 0,10$

Los Frentes de Pareto obtenidos para los algoritmos de reducción de ruido a partir de una imagen de gel de electroforesis con un ruido gaussiano con $\sigma = 0,10$ se muestran en la Figura 5.6. En éste caso se mantiene la misma tendencia de los Frentes de Pareto obtenido con un ruido con $\sigma = 0,05$ mostrados en la Figura 5.4, donde los valores del ruido de método escalar se mantienen en el mismo rango de 0 a 10000, que confirma que ésta medida cuantifica la distorsión introducida a las imágenes procesadas por los algoritmos independiente del nivel de ruido aplicado. Por otro lado, como es de esperar al aumentar el nivel de ruido disminuye el rango de valores del error cuadrático medio inverso.

Con respecto al comportamiento de los algoritmos se encontró que éste es similar al caso del ruido con $\sigma = 0,05$, donde por un lado, el frente de Pareto del algoritmo de medias no locales es el que dominó los frentes del resto de algoritmos, y por otro, el frente del reductor de ruido SUSAN abarcó un rango de ruido de método de 500 a 10000 pero con una disminución del error cuadrático medio en baja medida, en contraposición al filtro de mediana, que a pesar de presentar un ruido de método de aproximadamente 500,

disminuyó el error cuadrático medio en mayor medida que el reductor de ruido SUSAN.

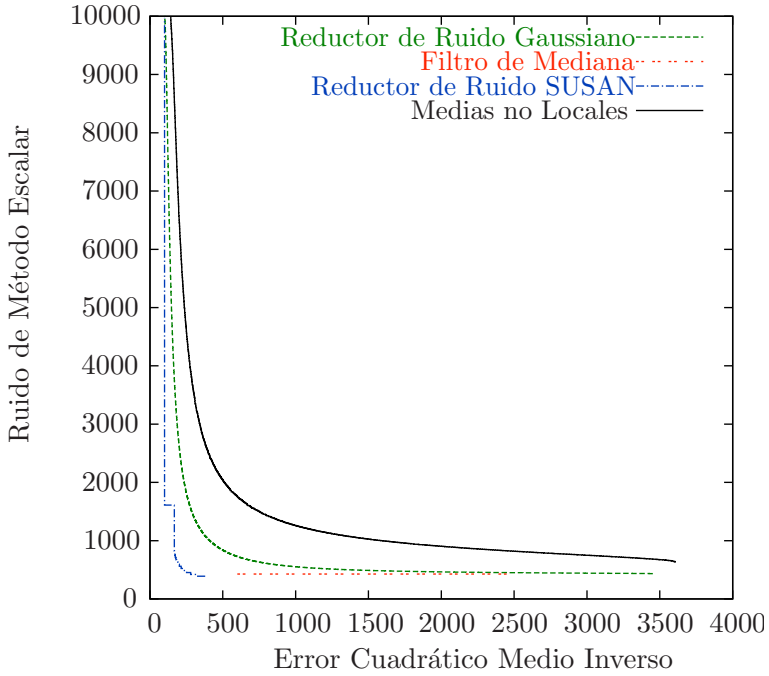


Figura 5.6: Frente de Pareto de los algoritmos de reducción de ruido con $\sigma = 0.10$. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Método de Ruido Escalar* (f_{mre}).

En la Figura 5.7(a) se muestra un segmento de una imagen de un gel con un ruido gaussiano con $\sigma = 0,10$. Para efectos de comparar las imágenes procesadas por los algoritmos se eligieron puntos en los diferentes frentes de Pareto con un ruido de método de aproximadamente 1000. Al comparar las imágenes de las Figuras 5.7(b) a la 5.7(e), se observa que la que redujo en mayor medida el ruido y que introdujo distorsión en menor medida fue el algoritmo de medias no locales. En la Tabla 5.6 se muestran los parámetros empleados por los diferentes algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.7. Además, en la Tabla 5.5 se resumen las medidas de aptitud obtenidas con las imágenes de la Figura 5.7.

Los parámetros empleados por los algoritmos para obtener los Frentes de Pareto de la Figura 5.6, son congruentes con el nivel de ruido aplicado en éste caso. Así, el reductor de ruido gaussiano empleó máscaras de tamaños en el rango de 9×9 y 17×17, con una varianza entre el rango de 1 a 6. Por su parte el filtro de mediana empleó máscaras de 9×9 a 13×13. El reductor de ruido SUSAN utilizó factores de forma de 2 a 16 y umbrales de 0 a 3. Finalmente, el algoritmo de medias no locales empleó ventanas de similitud de 7×7 a 9×9, ventanas de búsqueda de 9×9 a 17×17, y el grado de filtrado se encontró entre $0,05$ y $0,15$.

Tabla 5.5: Medidas de aptitud de las imágenes de la Figura 5.7

Algoritmo	Error Cuadrático Medio Inverso (f_{ecmi})	Método de Ruido Escalar (f_{rme})	Tiempo de Ejecución (s)
Reductor de Ruido Gaussiano	391	1064	0.001
Filtro de Mediana	2314	432	0.18
Reductor de Ruido Susan	160	840	0.01
Medias no Locales	730	919	0.9

Tabla 5.6: Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.7

Algoritmo	Parámetro	Valor
Reductor de Ruido Gaussiano	Tamaño de la máscara	9
	Varianza	$0,35$
	Tipo de frontera	Constante
Filtro de Mediana	Tamaño de la máscara	9
	Tipo de frontera	Constante
Reductor de Ruido SUSAN	Tamaño de la máscara	3
	Umbral	16
	Factor de forma	12
	Tipo de frontera	Cero
Medias no Locales	Grado de filtrado (h)	$0,044$
	Tamaño de la ventana de similud	9
	Tamaño de la ventana de búsqueda	9
	Varianza de la máscara Gaussiana	$0,42$
	Tipo de frontera	Espejo

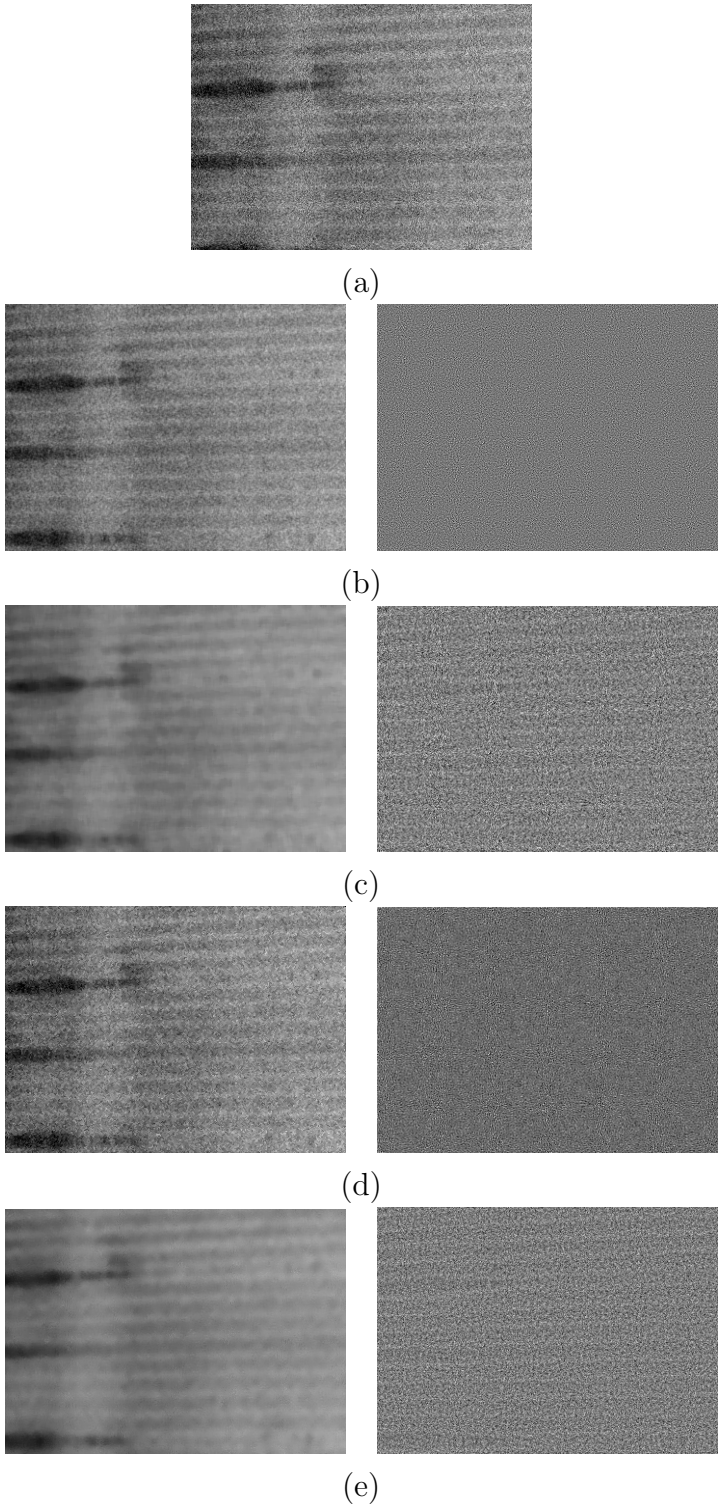


Figura 5.7: Imágenes filtradas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original $\mathcal{N}$ con un ruido gaussiano con $\sigma = 0,10$, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales.

Ruido Impulso

Para la obtención del Frente de Pareto se le aplicó un ruido impulsional con un 10% de cobertura de la imagen. En la Figura 5.8 se muestra los Frentes de los algoritmos obtenidos en éste caso. De ésta se observa que a diferencia de los resultados obtenidos con el ruido gaussiano y blanco, el frente de Pareto del filtro de mediana dominó al resto de los algoritmos respecto a ambas medidas de aptitud.

De lo anterior se concluye que por un lado los algoritmos idóneos para el ruido gaussiano y blanco son el algoritmo de medias no locales seguido del reductor de ruido gaussino, mientras que por otro lado, los algoritmos idóneos para el ruido impulsional son el filtro de mediana seguido del reductor de ruido SUSAN. Particularmente, para las imágenes de geles de electroforesis se encontró que el algoritmo de medias no locales es el idóneo.

En la Figura 5.9(a) se muestra una imagen con un ruido impulsional que cubre el 10% de la imagen. Para comparar las imágenes con ruido impulsional procesadas se eligió el punto en el frente de cada algoritmo con el menor error cuadrático medio (mayor valor del error cuadrático medio inverso). Al comparar las imágenes de las Figuras 5.9(b) a 5.9(e), se observa como efectivamente el filtro de mediana fue el que mejor se desempeñó eliminando por completo el ruido impulsional con un valor de error cuadrático medio

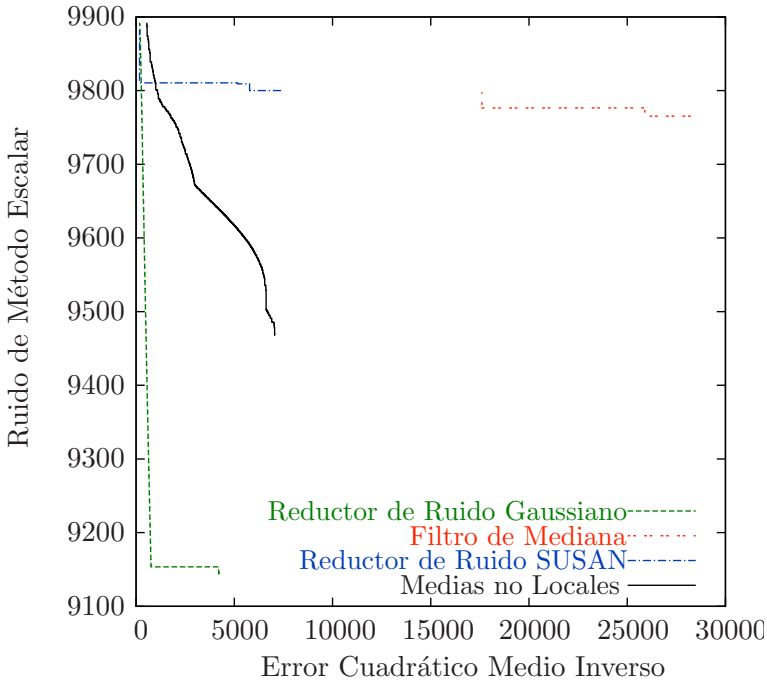


Figura 5.8: Frente de Pareto de los algoritmos de reducción de ruido aplicados a imágenes con ruido impulsional. *Error Cuadrático Medio Inverso* (f_{ecmi}) versus *Método de Ruido Escalar* (f_{rme}).

inverso de $f_{ecmi} = 28236$, con una mínima distorsión en la imagen procesada, con un valor de ruido de método escalar de $f_{rme} = 9745$. En la Tabla 5.8 se muestran los parámetros empleados por los diferentes algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.9. Además, en la Tabla 5.7 se resumen las medidas de aptitud obtenidas con las imágenes de la Figura 5.9.

Para obtener el Frente de Pareto de reductor de ruido gaussiano de la Figura 5.8 empleó un rango de tamaños de máscaras de 9×9 a 17×17, y varianza entre $0,5$ y $4,5$. En el caso del filtro de mediana el único tamaño empleado fue 3×3, con todos los tipos de frontera de la imagen posibles: cero, espejo, periódica, constante. Por su parte el reductor de ruido SUSAN, utilizó valores de umbral entre 0 y 2, valores de factor de forma entre 2 y 16. Finalmente, el algoritmo de medias no locales empleó valores de tamaño de la ventana de similitud entre 3×3 y 9×9, los tamaños de la ventana de búsqueda oscilaron entre 5×5 y 7×7, el valor del grado de filtrado se encontró entre $0,002$ y $0,007$, y la varianza entre $0,03$ y 3.

Tabla 5.7: Medidas de aptitud de las imágenes de la Figura 5.9

Algoritmo	Error Cuadrático Medio Inverso (f_{ecmi})	Método de Ruido Escalar (f_{rme})
Reductor de Ruido Gaussiano	4220	9145
Filtro de Mediana	28236	9795
Reductor de Ruido Susan	7383	9780
Medias no Locales	7050	9463

Tabla 5.8: Parámetros empleados en los algoritmos de reducción de ruido para obtener las imágenes de la Figura 5.9

Algoritmo	Parámetro	Valor
Reductor de Ruido Gaussiano	Tamaño de la máscara	9
	Varianza	$4,55$
	Tipo de frontera	Constante
Filtro de Mediana	Tamaño de la máscara	3
	Tipo de frontera	Constante
Reductor de Ruido SUSAN	Tamaño de la máscara	3
	Umbral	1
	Factor de forma	12
	Tipo de frontera	Cero
Medias no Locales	Grado de filtrado (h)	$0,0073$
	Tamaño de la ventana de similud	5
	Tamaño de la ventana de búsqueda	7
	Varianza de la máscara Gaussiana	$1,55$
	Tipo de frontera	Constante

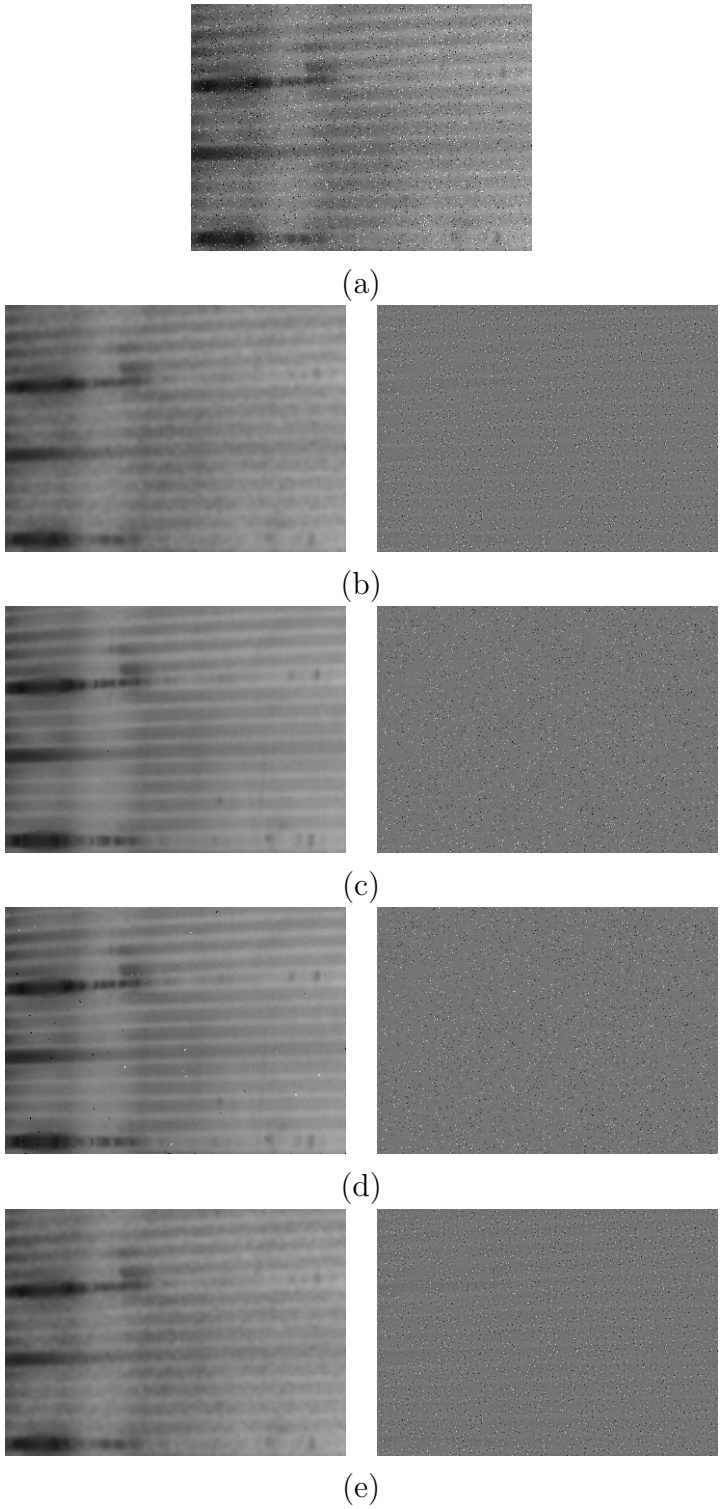


Figura 5.9: Imágenes filtradas por los algoritmos de reducción de ruido y su correspondiente imagen de ruido de método ($\mathcal{N} - \widehat{\mathcal{I}}$): (a) imagen original con un ruido impulsional que cubre el 10% de la imagen, (b) reductor de ruido Gaussiano, (c) filtro de mediana, (d) reductor de ruido SUSAN, (e) medias no locales.

5.2 Mejoramiento de contraste

Para efectos de evaluar los algoritmos de mejoramiento de contraste se emplearon tres imágenes de gel de electroforesis: una con iluminación normal con una resolución de 573×409 con un rango de niveles de gris de $0,065$ a $0,719$, una clara con una resolución de 556×398 con un rango de niveles de gris entre $0,58$ y 1 y una oscura con una resolución de 535×398 y con niveles de gris entre $0,1$ y $0,32$. Éstas fueron previamente procesadas con el algoritmo de medias no locales, con el objetivo de evitar que con la mejora de contraste además de los detalles de la imagen, se enfatizara el ruido.

Las medidas de aptitud empleadas para evaluar los algoritmos de mejoramiento de contraste son: *mejoramiento de contraste promedio* (f_{mcp}), que mide el grado de aumento de contraste de la imagen mejorada respecto a la imagen original, y *entropía* (f_e), que mide el contenido de información de una imagen, así el contraste de una imagen es proporcional a la entropía de ésta. En la Figura 5.10 se muestran los Frentes de Paretos de los algoritmos: *ecualización de histograma* y *mejoramiento de gradiente*. Éstos Frentes se obtuvieron mediante el conjunto de tres imágenes discutido anteriormente. Al comparar éstos Frentes es evidente la dominancia con respecto a ambas medidas de aptitud del algoritmo de mejoramiento de contraste sobre la ecualización de histograma.

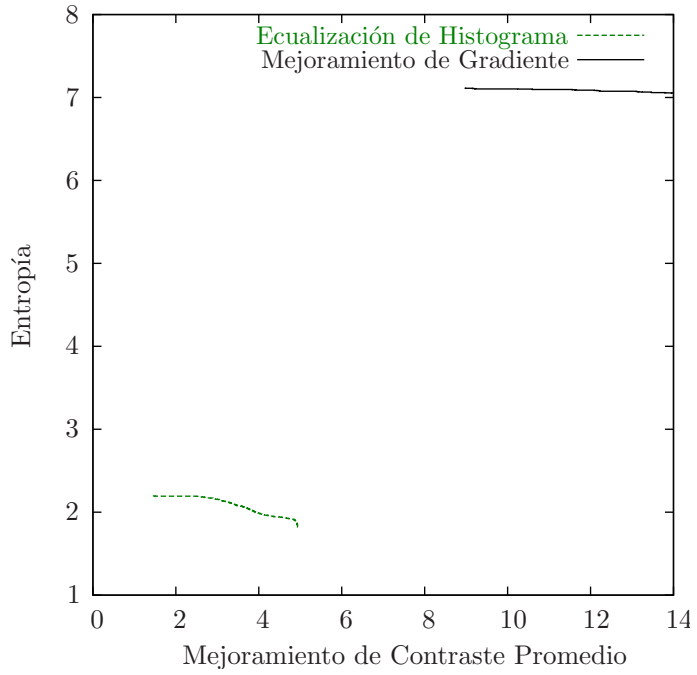


Figura 5.10: Frentes de pareto de los algoritmos de mejoramiento de contraste. *Mejoramiento de Contraste Promedio* (f_{mcp}) versus *Entropía* (f_e).

En la Figura 5.11(a) se muestra una imagen de gel de electroforesis con bajo contraste y su correspondiente histograma. Como se discutió en la Sección 4.2.1 una imagen con bajo contraste se caracteriza por tener las componentes de su histograma en un rango reducido de la escala de gris. En éste sentido, tal y como se muestra en el histograma de Figura 5.11(a), las componentes de éste se concentran en el centro de la escala de gris. Para efectos de comparación de las imágenes procesadas por los algoritmos se eligieron las parametrizaciones de los algoritmos que permitieron que se obtuviera un mayor mejoramiento de contraste promedio.

En la Figura 5.11(b) se muestran una imagen de gel de electroforesis, mejorada en contraste mediante la ecualización de histograma y su correspondiente histograma. La ecualización de histograma transforma el histograma de la imagen original en uno uniforme, tal y como se muestra el histograma de la Figura 5.11(b). Lo anterior tiene el inconveniente de que el valor medio del brillo de la imagen original cambia a un valor medio en la mitad del rango dinámico de los niveles de gris, independientemente del valor original. Además, la ecualización de histograma al ser implementada globalmente procesa la imagen como un todo, por lo que, el resultado del contraste local de ciertos sectores de la imagen no es el deseado, como es el caso del extremo izquierdo de la imagen de gel de electroforesis de la Figura 5.11(b), que se observa oscuro. En la Tabla 5.10 se muestran los parámetros empleados para la ecualización de histograma, donde el rango de valores de entrada y salida se eligió de 0 a 1, debido a que por un lado la imagen de entrada a pesar de que concentra sus componentes en el centro del histograma, posee algunos pocos valores en los extremos del rango, de tal manera que el rango de 0 a 1 los abarque por completo. Además por otro lado, el rango deseado en la imagen procesada es de 0 a 1 para que abarque el rango dinámico completo y así se logre un mayor contraste, siendo ésto congruente con los parámetros obtenidos a partir del Frente de Pareto. Por otro lado, el número de celdas empleadas por la ecualización de histograma fue 256.

Por otra parte, en la Figura 5.11(c) se muestran una imagen mejorada en contraste mediante el algoritmo de mejoramiento de gradiente y su correspondiente histograma. El histograma de imagen mejorada como se aprecia en la Figura 5.11(c) abarca todo el rango dinámico de niveles de gris. Lo anterior aunado al hecho de que el mejoramiento de gradiente opera localmente, permite que la imagen procesada presente más detalles, a diferencia de la imagen procesada mediante la ecualización de histograma. En la Tabla 5.10 se muestra que el parámetro δ fue 1.5. Además, debido a que las componentes del histograma de la imagen original con bajo contraste se concentran en la mitad del rango dinámico un $\delta = 1.5$ es suficiente para que el algoritmo logre mejorar la imagen, de tal manera que el histograma de la imagen procesada abarque el rango dinámico completo de niveles de gris, como se muestra en la Figura 5.11(c).

Finalmente, se puede concluir que una imagen con alto contraste se caractiza por poseer

componentes del histograma que abarque el rango dinámico completo de niveles de gris. Es así como se espera que las imágenes procesadas por los algoritmos de mejoramiento de contraste posean un histograma que contenga componentes en todo el rango dinámico. La diferencia entre los algoritmos consiste en cómo se logre lo anterior, así la ecualización de histograma opera directamente sobre el histograma, que como se comentó anteriormente tiene sus desventajas, y por otra parte el algoritmo de mejoramiento de gradiente sobre los gradientes de la imagen. En la Tabla 5.9 se muestran las medidas de aptitud obtenidas con las imágenes de la Figura 5.11.

Tabla 5.9: Medidas de aptitud de las imágenes de la Figura 5.11

Algoritmo	Mejoramiento de Contraste Promedio (f_{mcp})	Entropía (f_e)
Ecualización de Histograma	1.87	6.05
Mejoramiento de Gradiente	8.30	12.59

Tabla 5.10: Parámetros empleados en los algoritmos de mejoramiento de contraste para obtener las imágenes de la Figura 5.11

Algoritmo	Parámetro	Valor
Ecualización de Histograma	Valor menor que es ecualizado desde la entrada	0
	Valor mayor que es ecualizado desde la entrada	1
	Valor menor que es ecualizado hacia la salida	0
	Valor mayor que es ecualizado hacia la entrada	1
	Número de celdas empleadas para la ecualización	256
Mejoramiento de Gradiente	Cantidad de mejoramiento de contraste (δ)	1.5

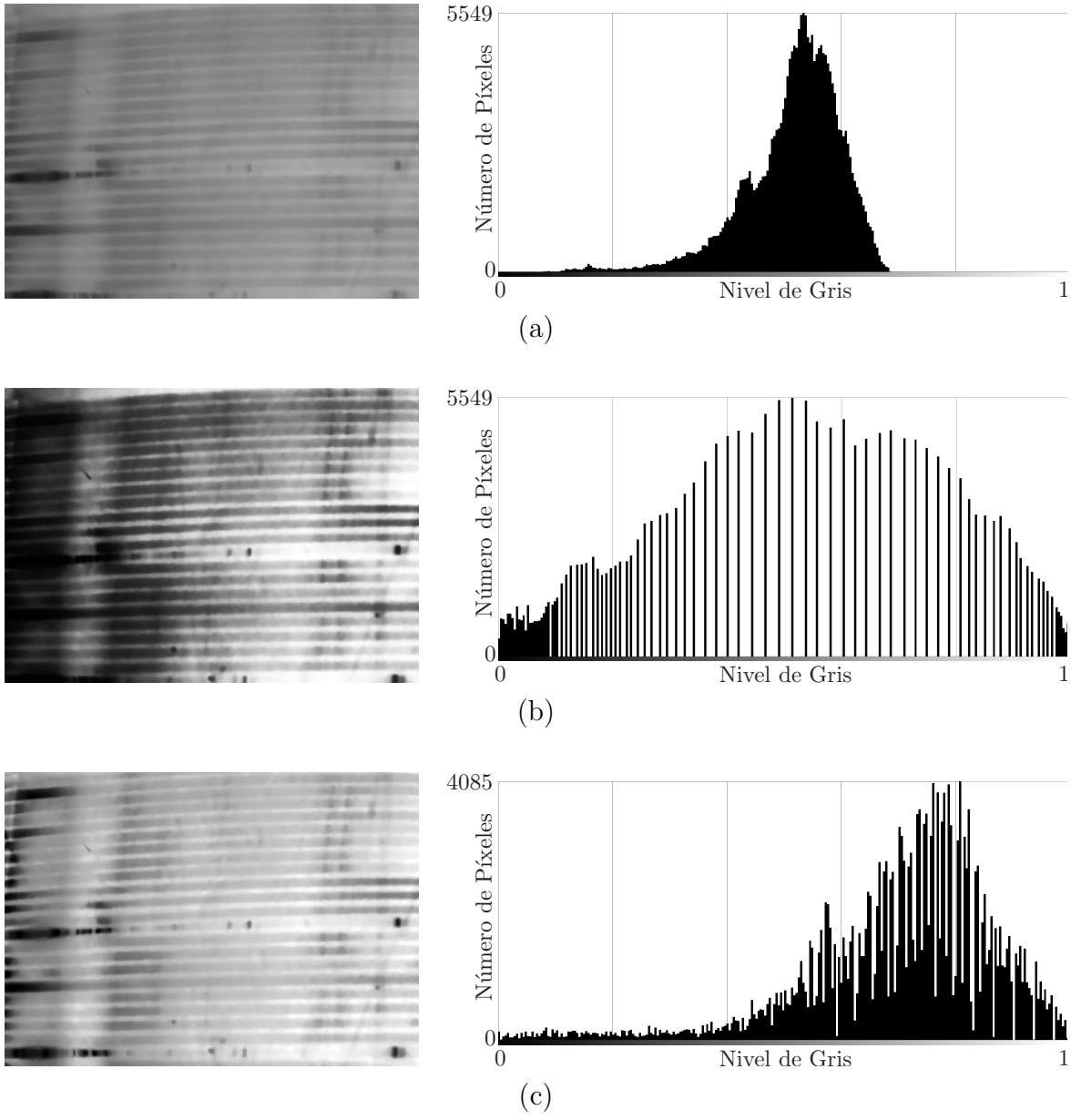


Figura 5.11: Imágenes procesadas por los algoritmos de mejoramiento de contraste y su correspondiente histograma: (a) imagen de gel de electroforesis con bajo contraste, (b) ecualizador de histograma, (c) mejoramiento de gradiente.

En la Figura 5.12(a) se muestran una imagen de gel de electroforesis clara con bajo contraste y su correspondiente histograma. Como se aprecia en el histograma de ésta imagen los componentes se concentran en la parte superior del rango dinámico, por lo que imagen luce clara. Por otra parte, en la Figura 5.12(b) se muestran tanto la imagen procesada por la ecualización de histograma, como su correspondiente histograma. En éste caso la imagen se caracteriza por poseer ciertos sectores de la imagen en extremo oscuros, por lo que se pierde el detalle de ciertas bandas y carriles. Por otro lado, debido a que los componentes del histograma de la imagen original se encuentran agrupadas en el extremo superior del rango dinámico, se empleó como rango de la imagen de entrada de $0,5$ a 1, como se muestra en la Tabla 5.12, sin embargo, se empleó el rango de 0 a 1 para la imagen procesada. Por otra parte, en la Figura 5.12(c) se muestra la imagen procesada por el algoritmo de mejoramiento de gradiente y su correspondiente histograma. En éste caso la imagen presenta mayor detalle en las bandas y carriles, lo que es congruente con el hecho de que el valor de la entropía obtenido con éste algoritmo fue $f_e = 11.6$, que es más del doble del obtenido con la ecualización de histograma $f_e = 5.4$. En éste caso el algoritmo empleó un δ más grande de 3.5 para abarcar el rango dinámico completo, debido a que los componentes del histograma se ubican totalmente en el extremo superior del rango.

Tabla 5.11: Medidas de aptitud de las imágenes de la Figura 5.12

Algoritmo	Mejoramiento de Contraste Promedio (f_{mcp})	Entropía (f_e)
Ecualización de Histograma	2.21	5.4
Mejoramiento de Gradiente	2.28	11.60

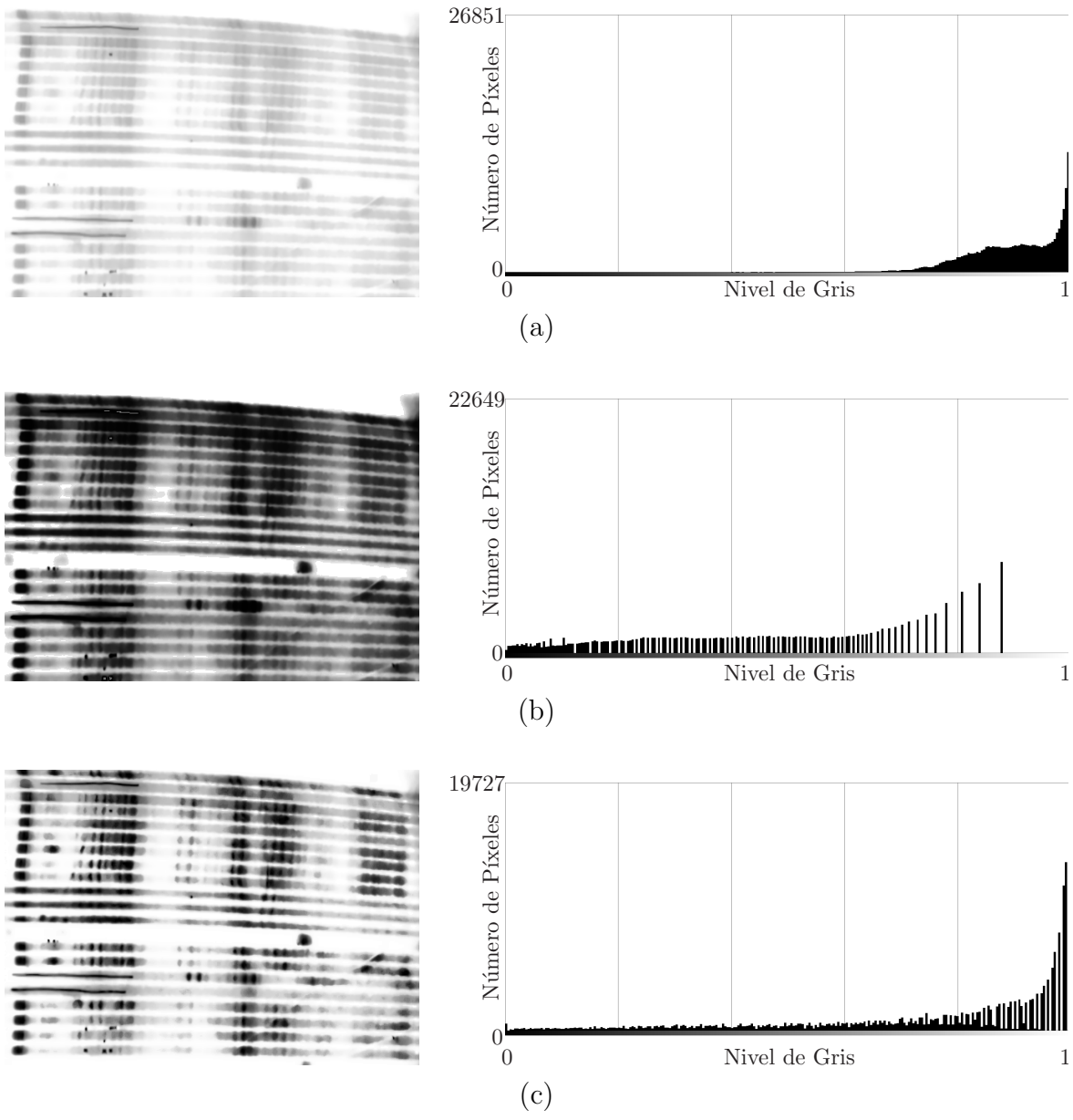


Figura 5.12: Imágenes procesadas por los algoritmos de mejoramiento de contraste y su correspondiente histograma: (a) imagen de gel de electroforesis clara con bajo contraste, (b) ecualizador de histograma, (c) mejoramiento de gradiente.

Tabla 5.12: Parámetros empleados en los algoritmos de mejoramiento de contraste para obtener las imágenes de la Figura 5.12

Algoritmo	Parámetro	Valor
Ecualización de Histograma	Valor menor que es ecualizado desde la entrada	0.5
	Valor mayor que es ecualizado desde la entrada	1
	Valor menor que es ecualizado hacia la salida	0
	Valor mayor que es ecualizado hacia la entrada	1
	Número de celdas empleadas para la ecualización	256
Mejoramiento de Gradiente	Cantidad de mejoramiento de contraste (δ)	3.5

En la Figura 5.13(a) se muestran una imagen de gel de electroforesis oscura con bajo contraste y su correspondiente histograma. Éste último se caracteriza por tener sus componentes concentrados en el extremo inferior del rango dinámico lo que le da una apariencia oscura a la imagen. En la Figura 5.13(b) se muestra la imagen procesada por el algoritmo de ecualización de histograma, donde se observa que nuevamente ésta posee ciertos sectores oscuros que hacen perder detalle a las bandas y carriles de la imagen de electroforesis. Como se muestra en la Tabla 5.14, se empleó como rango de entrada de 0 a 0.5, debido a que los componentes de la imagen oscura se encuentran en extremo inferior del rango dinámico. Además, en éste caso se empleó nuevamente un rango dinámico de salida de 0 a 1, con 256 celdas para la ecualización. Por otra parte, en la Figura 5.13(c) se muestra una imagen de gel de electroforesis procesada por el algoritmo de mejoramiento de gradiente, que presenta mayor detalle que la procesada por la ecualización de histograma. En éste caso como se muestra en la Tabla 5.13 se empleó $\delta = 5.5$, con el objetivo de que la imagen procesada abarcara todo el rango dinámico de valores.

Finalmente, al comparar los resultados expresados en los frentes de Pareto de la ecualización de histograma y el mejoramiento de gradiente, se concluye que éste último es el idóneo para las imágenes de geles de electroforesis

Tabla 5.13: Medidas de aptitud de las imágenes de la Figura 5.13

Algoritmo	Mejoramiento de Contraste Promedio (f_{mcp})	Entropía (f_e)
Ecualización de Histograma	5.05	4.98
Mejoramiento de Gradiente	5.76	11.4

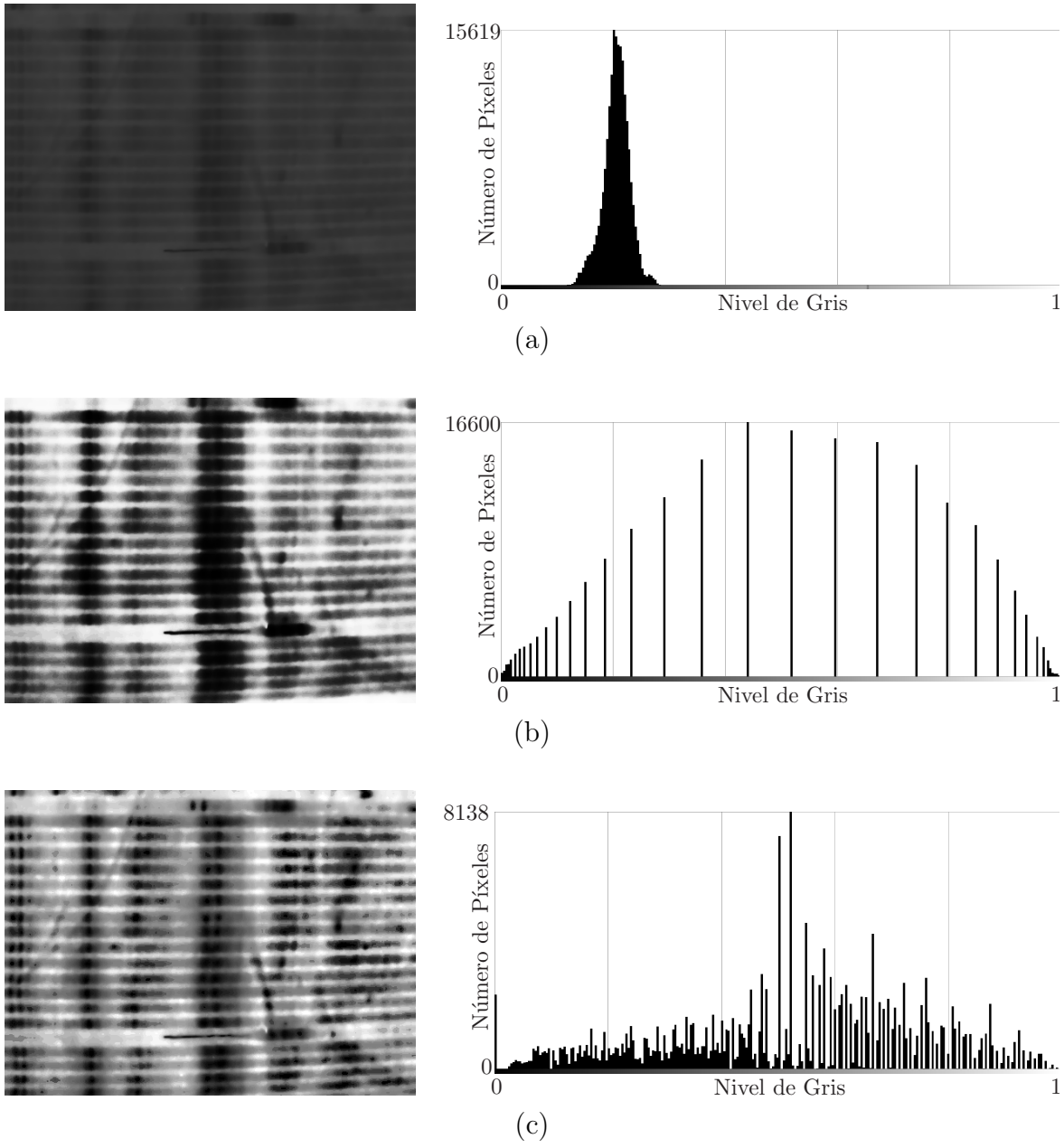


Figura 5.13: Imágenes procesadas por los algoritmos de mejoramiento de contraste y su correspondiente histograma: (a) imagen de gel de electroforesis oscura con bajo contraste, (b) ecualizador de histograma, (c) mejoramiento de gradiente.

Tabla 5.14: Parámetros empleados en los algoritmos de mejoramiento de contraste para obtener las imágenes de la Figura 5.13

Algoritmo	Parámetro	Valor
Ecualización de Histograma	Valor menor que es ecualizado desde la entrada	0
	Valor mayor que es ecualizado desde la entrada	0.5
	Valor menor que es ecualizado hacia la salida	0
	Valor mayor que es ecualizado hacia la entrada	1
	Número de celdas empleadas para la ecualización	256
Mejoramiento de Gradiente	Cantidad de mejoramiento de contraste (δ)	5.5

5.3 Mejoramiento de nitidez

Para evaluar el algoritmo de mejoramiento de nitidez *enmascarado de desenfoque*, se empleó una imagen de gel electroforesis que al momento de adquirirla se le aplicó un leve desenfoque en la cámara. Antes de utilizar ésta imagen para obtener el Frente de Pareto se le aplicó el algoritmo *medias no locales* para efectos de eliminar ruido proveniente del proceso de adquisición de la imagen, debido a que el enmascarado de desenfoque enfatiza no solo los bordes sino también lo hace con el ruido. Ésto se realiza para que el análsis se concentre en la mejora en nitidez y que el ruido no interfiera en ello.

El algoritmo de enmascarado de desenfoque se evaluó en dos etapas: en la primera de ellas se empleó la máscara laplaciana y en la segunda la máscara gaussiana, ésto con el objetivo de comparar las dos variantes del algoritmo. Como se discutió en la Sección 3.3.3 las medidas para evaluar el mejoramiento de nitidez en una imagen son: *difuminación* (f_d) que mide la atenuación introducida por el algoritmo en las regiones planas, y *nitidez* (f_n) que mide la amplificación introducida por el algoritmo en los bordes. En la Figura 5.14 se muestran los Frentes de Pareto que se obtuvieron, donde se aprecia la clara superioridad de la máscara laplaciana sobre la gaussiana, principalmente respecto a f_n. La naturaleza derivativa de la máscara laplaciana hace que ésta presente un mejor rendimiento en la mejora de nitidez en los bordes.

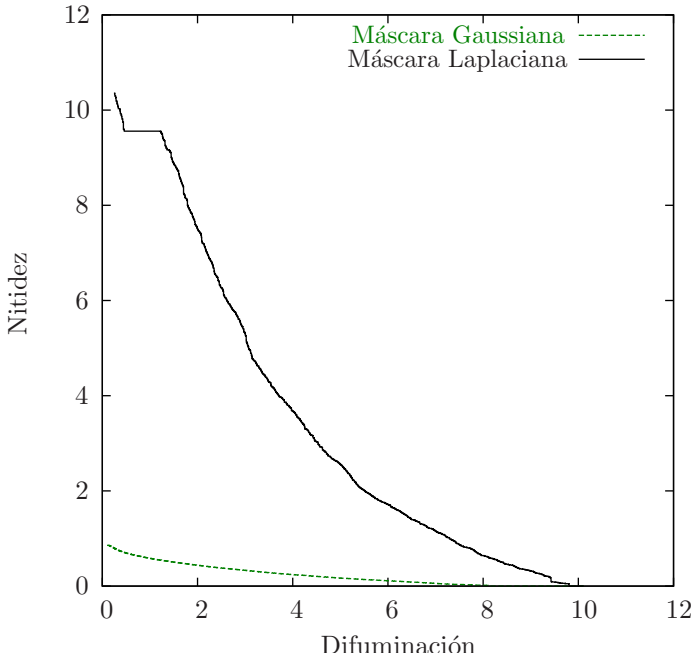


Figura 5.14: Frente de Pareto de las dos variantes de *enmascarado de desenfoque*. *Difuminación* (f_d) versus *Nitidez* (f_n).

Para ilustrar lo expresado en los Frentes de Pareto de la Figura 5.14, se eligió un punto con un valor constante de difuminación de aproximadamente 2, tanto en el frente de Pareto de la máscara gaussiana, como laplaciana. En cada uno se emplearon los parámetros indicados por el frente de Pareto para lograr esos puntos. En la Figura 5.15(a) se muestra un segmento de una imagen de gel de electroforesis, con baja nitidez. En la Figura 5.15(b) la imagen con nitidez mejorada mediante el algoritmo de enmascarado de desenfoque con una máscara gaussiana, en donde se observa que tanto los carriles (líneas horizontales) como las bandas (pequeñas columnas verticales oscuras en cada carril) presentan una mayor definición. Un aspecto relevante que se observa en la Figura 5.15(b) es que la imagen además de haber mejorado su nitidez, también mejora su contraste; ésto gracias a la constante de alto impulso con un valor $C > 1$. Los valores de las medidas de aptitud en éste caso son $f_d = 2.03$ y $f_n = 0.43$, y ésto es congruente con el diagrama de dispersión que se muestra en el extremo derecho de la Figura 5.15(b), donde el rango de las magnitudes de los gradientes aumentados de los píxeles se encontró de 0 a $0,35$. Por otra parte, en la Figura 5.15(c) se muestra una imagen con nitidez mejorada mediante el algoritmo de enmascarado de desenfoque con una máscara laplaciana, en donde se observa que los carriles y las bandas presentan una mayor definición que en el caso de la máscara gaussiana. Los valores de las medidas de aptitud en éste caso son $f_d = 1.99$ y $f_n = 8.51$, lo que indica como se comprueba en el diagrama de dispersión del extremo derecho de la Figura 5.15(c), que el rango de las magnitudes de los gradientes aumentados de los píxeles

Tabla 5.15: Parámetros empleados en el algoritmo de enmascarado de desenfoque para obtener las imágenes de la Figura 5.15

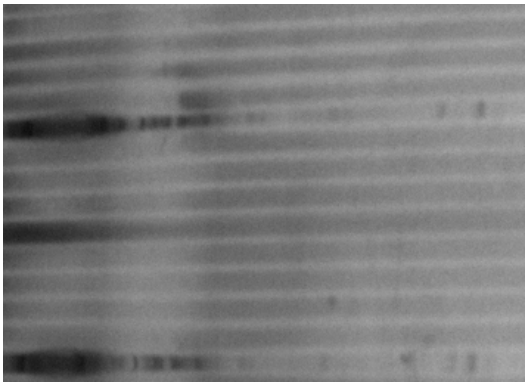
Tipo de Máscara	Parámetro	Valor
Gaussiana	Tamaño de la máscara	5
	Constante de alto impulso	2.1
	Varianza	8.0
	Tipo de frontera	Constante
Laplaciana	Tamaño de la máscara	3
	Constante de alto impulso	1.2
	Tipo de vecindario	Vecindario de 8
	Tipo de frontera	Constante

Tabla 5.16: Medidas de aptitud de las imágenes de la Figura 5.15

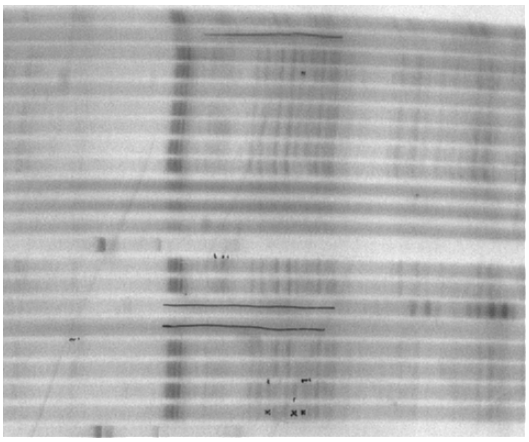
Tipo de Máscara	Difuminación (f_d)	Nitidez (f_n)
Gaussiana	0.18	0.83
Laplaciana	1.99	8.51

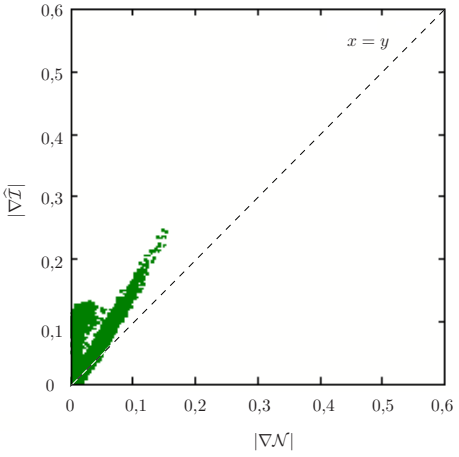
es de 0 a $0,47$.

Al comparar los frentes de Pareto de la Figura 5.14, se observa que el algoritmo de enmascaramiento de desenfoque con máscara laplaciana domina al de máscara gaussiana respecto a ambas medidas de aptitud. Por otra parte, se demostró que el diagrama de dispersión de las magnitudes de los gradientes de la imagen original versus la imagen filtrada, es una herramienta útil para la comparación de algoritmos de mejoramiento de nitidez. Así, el aumento de nitidez no solo se expresa en la cantidad de píxeles que aumentaron su gradiente, sino también en la magnitud de dicho aumento. En éste sentido al comparar el diagrama de dispersión de la máscara gaussiana en la Figura 5.15(b) con el de la máscara laplaciana en la Figura 5.15(c), se observó que éste ultimo presentó un aumento en mayor medida. En la Tabla 5.15 se muestra los parámetros empleados por las dos variantes del algoritmo enmascarado de desenfoque para lograr las imágenes de la Figura 5.15. Además, en la Tabla 5.16 se resume los valores de las medidas de aptitud correspondientes a las imágenes de gel de electroforesis de la Figura 5.15.

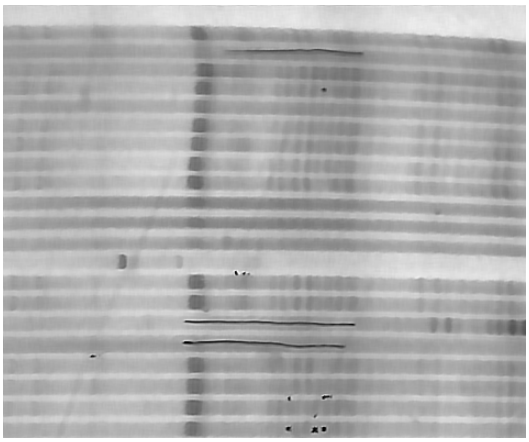


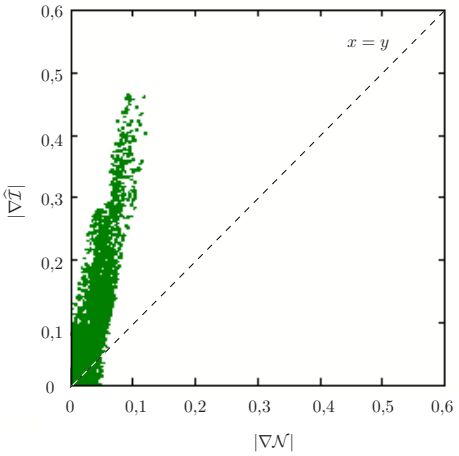
(a)





(b)





(c)

Figura 5.15: Imágenes filtradas mediante el algoritmo enmascarado de desenfoque y su correspondiente diagrama de dispersión de la magnitud del gradiente de la imagen original $|\mathcal{N}|$ versus la magnitud del gradiente de la imagen filtrada $|\widehat{\mathcal{I}}|$: (a) imagen original con baja nitidez $\mathcal{N}$, (b) imagen procesada con máscara gaussiana, (c) imagen procesada con máscara laplaciana.

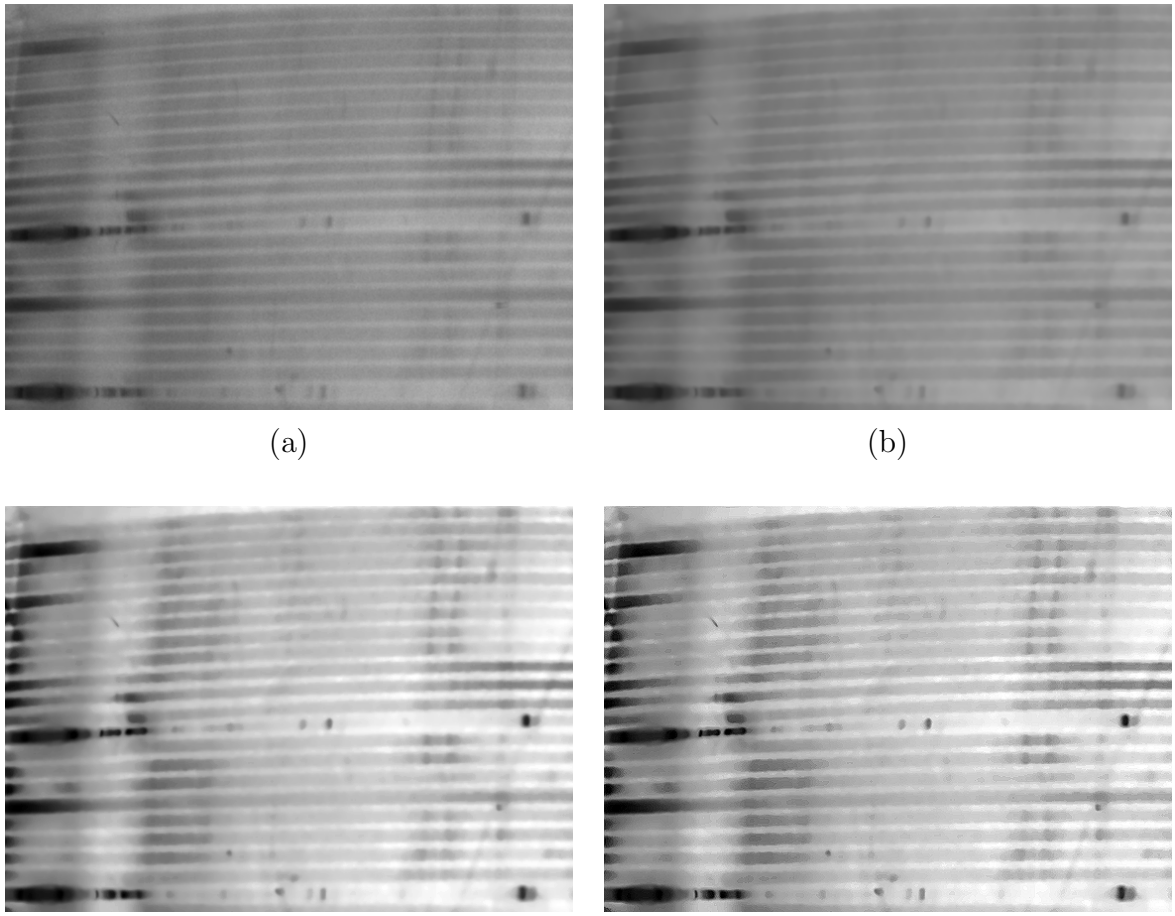
5.4 Combinación de algoritmos de mejoramiento de imágenes

En las secciones anteriores se analizó individualmente el efecto de los algoritmos de reducción de ruido, mejoramiento de contraste y nitidez. Sin embargo, como se expresó en el Capítulo 1 el objetivo principal de éste trabajo es mejorar simultáneamente éstos tres parámetros en las imágenes de gel de electroforesis. De tal manera, que en ésta sección se presentan los resultados que se obtuvieron de la combinación de los algoritmos que presentaron el mejor rendimiento en cada caso. Así, el mejor algoritmo de reducción de ruido es *medias no locales*, el de mejoramiento de contraste es el algoritmo *mejoramiento de gradiente* y para el mejoramiento de nitidez es el algoritmo *enmascarado de desenfoque* con una máscara *laplaciana*.

La secuencia que se empleó se muestra en la Figura 5.16, donde la primera etapa es la reducción de ruido, esto con el objetivo de que ni el algoritmo de mejoramiento de contraste, ni el de mejoramiento de nitidez enfaticen el ruido. La segunda etapa es el mejoramiento de contraste y finalmente, la tercera etapa es el mejoramiento de nitidez, que se coloca al final para eliminar tanto el desenfoque de la imagen original, como cualquiera introducido por los algoritmos de reducción de ruido y mejoramiento de contraste.

Figura 5.16: Secuencia de algoritmos para el mejoramiento de las imágenes de geles de electroforesis

En la Figura 5.17(a) se muestra un segmento de una imagen de gel de electroforesis, que presenta ruido gaussiano y blanco, bajo contraste y nitidez. En la Figura 5.17(b) se muestra la imagen con reducción de ruido mediante el algoritmo de medias no locales para evitar que la etapa de mejoramiento de contraste lo enfatice. En la Figura 5.17(c) se muestra la imagen mejorada en contraste mediante el algoritmo de mejoramiento de gradiente. Finalmente, en la Figura 5.17(d) se muestra la imagen final con mejora en nitidez, que está lista para pasar a la etapa de compensación de distorsiones geométricas que se desarrollará en otro trabajo. Los parámetros empleados por los algoritmos para obtener las imágenes de la Figura 5.17 se muestran en la Tabla 5.17.



(a) (b)

(c) (d)

Figura 5.17: Secuencia de mejoramiento de imágenes: (a) imagen de gel de electroforesis original, (b) imagen con disminución de ruido, (c) imagen mejorada en contraste, (d) imagen mejorada en nitidez.

Tabla 5.17: Parámetros empleados por los algoritmos para obtener las imágenes de la Figura 5.17

Etapa	Algoritmo	Parámetro	Valor
Reducción de Ruido	Medias no Locales	Tamaño de la máscara de similitud	5
		Tamaño de la máscara de búsqueda	9
		Grado de filtrado (h)	0.001
		Varianza de la máscara gaussiana	2.6
		Tipo de Frontera	Espejo
Mejoramiento de Contraste	Mejoramiento de Gradiente	Mejoramiento logrado (δ)	1.5
Mejoramiento de Nitidez	Enmascaramiento de Desenfoque	Tipo de máscara	Laplaciana
		Tamaño de la máscara	3
		Tipo de vecindario	De 4
		Constante de alto impulso (C)	1
		Tipo de Frontera	Constante

Capítulo 6

Conclusiones y Recomendaciones

El ser humano en la mayoría de los casos es quién al final juzga la calidad de una imagen. Sin embargo, ésta evaluación subjetiva es insuficiente para analizar el redimiento de algoritmos de mejora de calidad de imágenes, principalmente en aquellas aplicaciones en donde la imágenes son analizadas automáticamente por un sistema, como es el caso del análisis por computadora de imágenes de gel de electroforesis. En éste sentido la principal contribución de éste trabajo a parte de la implementación de algoritmos de mejoramiento de imágenes, es la evaluación objetiva de algoritmos mediante la obtención de Frentes de Pareto. En éste capítulo se sintetizan los principales resultados y aportes de éste trabajo.

6.1 Conclusiones

Éste trabajo se concentró en la mejora de imágenes respecto a tres aspectos: reducción de ruido, mejora de contraste y mejora de nitidez. El interés particularmente se ha concentrado en las imágenes de geles de electroforesis, empleadas en la caracterización molecular de organismos. Éstas imágenes se caracterizan por presentar mayormente ruido gaussiano y blanco, además presentan bajo contraste y baja nitidez. En éste trabajo se empezó por analizar diferentes enfoques propuestos en la literatura de mejoramiento de imágenes para la reducción de ruido, mejoramiento de contraste y nitidez. La evaluación consistió en determinar primero medidas de aptitud que evaluaran a los algoritmos de reducción de ruido, mejoramiento de contraste y nitidez, y segundo evaluar objetivamente los algoritmos con éstas medidas de aptitud, para encontrar los Frentes de Pareto de los diferentes algoritmos.

En primera instancia, se evaluaron los algoritmos de reducción de ruido, empleando un conjunto de imágenes a las que se les aplicó ruido gaussiano y blanco con desviaciones

estándar de 0,01, 0,05 y 0,10. Las medidas de aptitud definidas para evaluar los algoritmos fueron: *error cuadrático medio inverso* (f_{ecmi}) que cuantifica qué tanto disminuyeron el ruido los algoritmos y *el ruido de método escalar* (f_{rme}) qué cuantifica que tanta distorsión introducen los algoritmos a las imágenes procesadas. A partir de ésta evaluación se encontró que el Frente de Pareto del algoritmo llamado *medias no locales*, dominó a los frentes de pareto del resto de algoritmos, debido a que éste fue el que redujo en mayor medida el ruido gaussiano y blanco, con muy poca distorsión. El *reductor de ruido gaussiano* presentó un redimiento cercano al algoritmo medias no locales para el caso del ruido gaussiano y blanco con una desviación estándar de 0,01. Además, se encontró que para el caso del ruido con una desviación estándar de 0,01 es preferible emplear el reductor de ruido gaussiano, debido a que en comparación con el algoritmo de medias no locales es menos costoso computacionalmente. Por otra parte, el *filtro de mediana* presentó un buen redimiento en cuanto a disminución del error cuadrático medio, pero con un bajo redimiento en cuanto a la distorsión de las imágenes procesadas. En contraposición, el *reductor de ruido SUSAN*, presentó un buen redimiento en cuanto a la introducción de distorsión de las imágenes procesadas pero un bajo rendimiento en relación con la disminución del error cuadrático medio.

Otro tipo de ruido que suele presentarse en el proceso de adquisión es el ruido impulsional. En éste sentido, los algoritmos fueron evaluados también con un conjunto de imágenes a las cuales se le aplicó un ruido impulsional con una convertura de 10% de la imagen. A diferencia de los resultados obtenidos con el ruido gaussiano y blanco, acá el Frente de Pareto del Filtro de Mediana dominó al resto de algoritmos con respecto a ambas medidas de aptitud, seguido del reductor de ruido SUSAN. El filtro de mediana logró eliminar por completo el ruido impulsional con una distorsión mínima. Así, de acuerdo con los resultados obtenidos se determinó que el algoritmo de medias no locales es el idóneo para la reducción de ruido en éste tipo de imágenes.

Por otro lado, en un segundo momento se evaluaron los algoritmos de mejoramiento de contraste a patir de un conjunto de tres imágenes: una con bajo contraste, una oscura con bajo contraste y una clara con bajo contraste. Para evaluar los algoritmos de mejoramiento de contraste se emplearon las siguientes medidas de aptitud: *mejoramiento de contraste promedio* (f_{mcp}), que mide la cantidad de aumento de contraste que presentan las imágenes procesadas respecto a su versión original, y *entropía* (f_e) que mide la cantidad de información contenida en una imagen, siendo el contraste proporcional a la entropía. De ésta evaluación se encontró que el Frente de Pareto del algoritmo de *mejoramiento de gradiente* dominó, debido a que éste opera localmente a diferencia de la *ecualización de histograma* que opera globalmente. El hecho de que la ecualización de histograma opere globalmente tiene como consecuencia que el contraste en algunos sectores de la imagen no es el adecuado, por ejemplo en ciertas regiones de la imagen el contraste puede ser más oscuro o más claro de lo que debería. Así, de acuerdo con los resultados obtenidos

se encontró que el algoritmo de mejoramiento de gradiente es el idóneo para el aumento del contraste de las imágenes de geles de electroforesis.

Finalmente, se evaluaron dos variantes del algoritmo llamado *enmascaramiento de desenfoque*, mediante una imagen que al momento de ser adquirida se le aplicó un desenfoque. En éste caso las medidas de aptitud fueron: *difuminación* (f_d), que es el factor de atenuación de las regiones planas, y *nitidez* (f_n), que es el factor de amplificación de bordes y otros detalles de la imagen. A partir de los resultados se encontró que al emplear una máscara laplaciana para el desenfoque, el gradiente de los píxeles y por ende la nitidez de la imagen aumentaron en mayor medida que con la máscara gaussiana. Así, se determinó que el algoritmo de enmascaramiento de desenfoque con máscara laplaciana, es el idóneo para aumentar la nitidez en imágenes de gel de electroforesis.

Para lograr el objetivo de mejorar la calidad de las imágenes de electroforesis se debe aplicar simultáneamente los algoritmos de reducción de ruido, mejoramiento de contraste y mejoramiento de definición, sin embargo, se debe determinar la secuencia adecuada en que se aplican las mejoras. En sentido se encontró que ésta secuencia es: 1) reducción de ruido, 2) mejoramiento de contraste y 3) mejoramiento de nitidez. La reducción de ruido se aplica primero con el objetivo de que ni el algoritmo de mejoramiento de contraste ni el de mejoramiento de nitidez enfaticen el ruido, posteriormente se aplica el mejoramiento de contraste y la etapa de mejoramiento de nitidez se colocó al final de la imagen que puede estar difuminada, ya sea por el proceso de adquisición, o bien por los algoritmos de reducción de ruido y de mejoramiento de contraste.

Las contribuciones más importantes de éste trabajo se resumen de la siguiente manera:

- Análisis de diferentes enfoques propuestos en el área de mejoramiento de imágenes para reducción de ruido, mejoramiento de contraste y mejoramiento de nitidez (Sección 2.2).
- Implementación de un algoritmo de reducción de ruido con una mínima distorsión en las imágenes procesadas (Sección 4.1.5).
- Implementación de un algoritmo de mejoramiento de contraste local en imágenes digitalizadas (Sección 4.2.2).
- Implementación de un algoritmo de mejoramiento de nitidez en imágenes digitalizadas (Sección 4.3.1).
- Definición de diferentes medidas de aptitud para cuantificar el rendimiento de los algoritmos de reducción de ruido, mejoramiento de contraste y mejoramiento de nitidez (Sección 3.3).

- Evaluación multiobjetivo mediante el Frente de Pareto de los algoritmos aplicados a imágenes de geles de electroforesis. Ésto con el objetivo de encontrar el conjunto de parametrizaciones óptimas para la aplicación de interés por un lado, y para efectuar comparaciones entre diferentes algoritmos por otro. (Capítulos 3 y 5).

Así las cosas, en el presente trabajo se implementó algoritmos que disminuyeran el ruido, mejoraran el contraste y la nitidez en una imagen, además se determinaron medidas de aptitud para los diferentes algoritmos, para finalmente evaluar los algoritmos objetivamente mediante el Frente de Pareto. Todo lo anterior orientado al mejoramiento de imágenes de geles de electroforesis.

6.2 Recomendaciones

El tiempo de ejecución es un aspecto relevante en aplicaciones que se ejecutan en tiempo real. En el caso particular del algoritmo de medias no locales como se presentó en la Sección 4.1.5, existen dos enfoques para implementar eficientemente éste algoritmo: multiescala y mediante bloques. En éste trabajo se implementó el enfoque de bloques, sin embargo, se debe implementar el enfoque multiescala y compararlo con el de bloques. Esto se debe a que, a pesar del rendimiento presentado por éste algoritmo en cuanto a la reducción de ruido con poca distorsión, es costoso computacionalmente, por lo que se deben buscar alternativas que incrementen la velocidad. Por otra parte, como se describió en la Sección 2.2.3 los enfoques propuestos para mejoramiento de nitidez son escasos, de tal manera que se debe ampliar la búsqueda a otros enfoques o bien proponer nuevos.

Finalmente, en éste trabajo los algoritmos de mejoramiento de imágenes se aplicaron a imágenes de gel de electroforesis. Sin embargo, el uso de éstos algoritmos se puede ampliar a otro tipo de aplicaciones, siguiendo el mismo proceso de evaluación multiobjetivo empleado para encontrar las parametrizaciones óptimas respectivas.

Bibliografía

[1] P. Alvarado. *Segmentation of color images for ineractive 3D object retrieval.* PhD thesis, RWTH-Aachen University, 2004.

[2] P. Alvarado. *Notas de Clase. Modelos de Sistemas.* ITCR, primera edición, 2007.

[3] P. Alvarado, J. Castro, A. Salazar, O. Murillo, F. F. Rojas, and J. Peraza. *Análisis automatizado de patrones de ADN para la caracterización molecular. Propuesta de Proyecto de Investigación.* VIE. Instituto Tecnológico de Costa Rica, 2006.

[4] P. Alvarado and P. Dörfler. *LTI Image Processing Library. Developers Guide*, 2004.

[5] P. Alvarado, A. Salazar, O. Murillo, F. F. Rojas, and J. Peraza. *Análisis de imágenes de geles de electroforesis para la caracterización molecular de organismos. Propuesta de Proyecto de Investigación.* VIE. Instituto Tecnológico de Costa Rica, 2007.

[6] R. G. Baraniuk. Optimal tree aproximation with wavelets. In *Proceedings SPIE Tech. Conf. Wavelet Aplications Signal Processing VII*, volume 3813, pages 196–207, 1999.

[7] M. Beauchemin, Ko B Fung, and X. Geng. A method based on local variance for quality assessment of multiresolution image fusion. 2002.

[8] A. Buades. *Image and film denoise by non-local means.* PhD thesis, Les Illes Balears University, 2006.

[9] A. Buades, B. Coll, and J.M. Morel. On image denoising methods. Technical Report 2004-15, CMLA, 2004.

[10] A. Buades, B. Coll, and J.M. Morel. Image and movie denoising by non-local means. *International Journal of Computer Vision*, 2005.

[11] A. Buades, B. Coll, and J.M. Morel. A non-local algorithm for image denoising. *In Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2005.

[12] A. Buades, B. Coll, and J.M. Morel. A review of image denoising algorithm, with a new one. *Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

[13] R. W. Buccigrossi and E. P. Simoncelli. Image compression via joint statistical characterization in the wavelet domain. *IEEE Image Process*, 8(12):1688–1701, 1999.

[14] H. A. Chipman, E. D. Kolaczyk, and R. E. McCulloch. Adaptative bayesian wavelet shrinkage. *Amer. Stat. Assoc.*, 92(440):1413–1421, 1997.

[15] J. Dijik, D. Ridder, P. W. Verbeek, J. VWalraven, I. T. Young, R. Duin, and L. J. Vliet. A new measure for the effect of sharpening and smoothing filter on images. In B. K. Ersboll and P. Johansen, editors, *Proc. 11th Scadinavian Conference on Image Analysis*, SC1A'99, pages 213–220, 1999.

[16] D. L. Donoho and M. E. Raimondo. A fast wavelet algorithm for image deblurring. In R. May and A. J. Roberts, editors, *Proceedings of the 12th Computational Techniques and Applications Conference*, volume IV of *ANZIAM*, pages 29–46, 2005.

[17] David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.

[18] David L. Donoho and Iain M. Ideal spatial adaption via wavelet shrinkage. *Biometrike*, 81:425–455, 1994.

[19] M. R. Everingham, H. Muller, and B. T. Thomas. Evaluating image segmentation algorithms using the pareto front. In G. Sparr A. Heyden and P. Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision*, volume IV of *LNCS 2353*, pages 34–48, 2002.

[20] A. Picariello G. Boccignone. Multiscale contrast enhancement of medical images. *Proceedings of ICASSP*, 1997.

[21] R. Gonzalez and R. Woods. *Digital Image Processing*. Second edition, Prentice Hall, 2002.

[22] M. Hanmandlu, D. Jha, and R. Sharma. Color image enhancement by fuzzy intensification. In *Proceedings of International Conference on Pattern Recognition*, 2000.

[23] M. Hanmandlu, D. Jha, and R. Sharma. Localizated contrast enhancement of color images using clustering. In *Proceedings of IEEE International Conference on Information Technology*, Coding and Computing (ITCC), 2001.

[24] N. Y. Hassan and N. Aakamatsu. Contrast enhancemente technique of dark blurred image. *Int'l Journal of Computer Science and Netwotk Security*, 6(2A), 2006.

[25] I. Jafar and H. Ying. A new method for image contrast enhancement based on automatic specification of local histograms. *Int'l Journal of Computer Science and Netwotk Security*, 7(7), 2007.

[26] G. James. *Matemáticas avanzadas para ingeniería*. Prentice Hall, segunda edición, 2002.

[27] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and F. C. Harris. Survey of image denoising techniques. In *Proceedings of GSPx 2004*, 2004.

[28] S. Mukhopadhyay and B. Chanda. Hue preserving color image enhancement using multiscale morphology. *Indian Conference on Computer Vision, Graphics and Image Processing*, 2002.

[29] A. Oppenheim, A. Willsky, and S. H. Nawab. *Matemáticas avanzadas para ingeniería*. Prentice Hall, segunda edición, 1998.

[30] J. K. Romberg, H. Choi, and R. G. Baraniuk. Bayesian tree-structured image modeling usion wavelet-domain hidden markov models. *IEEE Image Process*, 10(7):1056–1068, 2001.

[31] M. Shyu and J. Leou. A geneticle algorithm approach to color image enhancement. *Pattern Recognition*, 31:871–880, 1998.

[32] E. P. Simoncelli and E. H. Adelson. Noise removal via bayesian wavelet coring. In *Third Int'l Conf on Image Proc*, volume I, pages 379–382, 1996.

[33] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. Technical Report TR95SMS1c, UK Secretary of State for Defence, 1995.

[34] J. L. Stark, F. Murtagh, E. J. Candes, and D. L. Donoho. Gray color image contrast enhancement by curvelet transform. *IEEE Transactions on Image Processing*, 12, 2003.

[35] K. Subr, A. Majumder, and S. Irani. Greedy algorithm for local contrast enhancement of images. In *Proceedings of the 13th International Conference on Image Analysis and Processing*, 2005.

[36] K. Subr, A. Majumder, and S. Irani. Contrast enhancement of images using human contrast sensitivity. In *Proceedings of the Symposium on Applied Perception in Graphics and Visualization*, 2006.

[37] A. Toet. A hierarchical morphological image descomposition. *Pattern recognition letters*, 11:267–274, 1990.

[38] A. Toet. Multiscale colo image enhancement. *Pattern recognition letters*, 13:167–174, 1992.

[39] R. L. D. Valois and K. K. D. Valois. *Spatial Vision*. Oxford University Press, 1900.

[40] K. V. Velde. Multiscale colo image enhancement. In *Proceedings on International Conference on Image Processing*, volume 3, pages 584–587, 1999.

[41] Y. J. Zhang. A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

Apéndice A

LTI-Lib

La LTI-Lib es una colección de algoritmos y estructuras de datos comunmente empleadas en aplicaciones de procesamiento digital de imágenes y visión por computadora, desarrollada en un principio en la cátedra de Informática Técnica (en alemán *Lehrstuhl fuer Techische Informatik (LTI)*) de la Universidad RWTH de Aachen en Alemania. Está escrita en C++ con el objeto de permitir la programación orientada a objetos y obtener un código eficiente. Su principal objetivo es proveer una biblioteca independiente, lo más cercana al estándar ANSI C++. La intensión es que trabaje en diferentes sistemas operativos con especial atención en Linux/GCC y MS Windows/Visual C++.

Historia

En los últimos años, varios grupos de investigación de LTI han trabajado en diferentes aplicaciones en el campo de visión por computadora. Antes de 1999, todos los proyectos que involucraban algoritmos de procesamiento de imágenes desarrollaban su propio software y aplicaciones independientemente. Lo anterior implica los siguientes problemas.

1. El trabajo se desperdiciaba innecesariamente en duplicación de código.

2. Reutilizar código era imposible debido a la falta de estandarización de las interfaces.

Éstas razones forzaron el diseño de una librería interna, que fuera utilizada por todos los grupos del LTI. La decisión de desarrollar una nueva librería se basa en los siguientes factores:

Primero, se necesitaba una biblioteca de C++ que siguiera consistentemente el concepto de orientación a objetos. ésto tenía como objeto simplificar el desarrollo de software, el mejoramiento y la mantenibilidad.

Segundo, la librería debería proveer no solo algoritmos para procesamiento de imágenes, pero también para tareas matemáticas y estadísticas, permitiendo un intercambio flexible de datos entre todos lo módulos. Muchas de las librerías existentes no cumplían con éstos requisitos.

Tercero, otras de las bibliotecas ya no se mantenían más, y algunas no tenían documentación del todo. Las bibliotecas comerciales estaban bien documentadas y tenían buenas herramientas, pero debido al concepto de "código cerrado" de éstas, era imposible mejorarlas o cambiar los algoritmos existentes sin reimplementarlos. Éste último punto es inaceptable para propósitos de investigación.

Cuarto, herramientas poderosas y generalizadas usadas en investigación pueden ser usadas para buscar soluciones de problemas relativamente pequeños, pero son usualmente muy lentas para aplicaciones más complejas o para prototipos complejos que involucran el proceso completo de adquisión y procesamiento. Usualmente también son requeridas aplicaciones en tiempo real o que involucren grandes cantidades de datos.

La creación de una nueva biblioteca en el LTI no fue un proyecto que inició desde cero, debido a la existencia de código previo y suficiente experiencia de todos los proyectos de investigación. En un inicio, una interface fue especificada y muchos de los algoritmos más importantes se adaptaron a ella. La LTI-Lib ha crecido a más de 750 clases (más de 350000 líneas de código), cubriendo procesamiento de imágenes, matemáticas, estadística, redes neuronales, interfaces con hardware, entre otros. El proyecto de la LTI-Lib-2 inició como una evolución natural de los previos conceptos a mediados del 2003. En ésta versión conceptos modernos pueden ser empleados debido a que las restricciones impuestas por el compilador de Microsoft Visual C++ 6.0 fueron removidas en las nuevas versiones de los compiladores MS. También, cambios adicionales de diseño se hicieron basados en la experiencia obtenida de la LTI-Lib-1.

El uso de una sola biblioteca ahorra tiempo de desarrollo, que sería requerido en la reimplementación de algoritmos usados comunmente. Éste tiempo puede ser invertido ahora en el desarrollo de nuevas soluciones o bien optimizar las existentes. En este sentido no solo el grupo de desarrolladores se verá beneficiado con las mejoras sino también los usuarios de la biblioteca. Su uso incrementa también la legibilidad de código, debido al hecho de que todas las tareas simples o complejas, siguen especificaciones generales conocidas, así el desarrollo y mantenimiento futuro será más fácil.

Arquitectura de la LTI-Lib

La LTI-Lib es fácil de usar debido a la especificación consistente de una interfaz de programación para todas las clases. La preservación de ésta consistencia es parcialmente lograda mediante el uso del PERL-script **ltiGenerator**. Basado en unos pocos datos rudimentarios dados por el programador éste script construye algunas plantillas, que contienen todas la definiciones estándar. Después de ésto solo la funcionalidad necesita ser implementada.

- **Funciones objeto**

Muchos de los algoritmos requieren parámetros, que son valores definidos por el usuario que modifican su comportamiento. Todos los algoritmos en la LTI-Lib están encapsulados en las llamadas *clases-objetos funcionales* (en inglés *functor-classes*). Ellas siempre contienen una clase llamada parámetros (*parameters*), que puede ser declara explícitamente o bien herada de la clase madre.

Esto significa, que cuando se utiliza la LTI-Lib, no se llama a algunas funciones o métodos con un muchos argumentos confusos, algunos datos de entrada, otros datos de salida, y adicionalmente una larga lista de parámetros. Los parámetros por defecto usualmente se almacenan en la clase función objeto, y todos los métodos que proveen la funcionalidad del algoritmo experan del usuario solo los datos de entrada y los objetos de salida en donde los resultados esperados se escriben. También se pueden cambiar los parámetros usados para definir la funcionalidad de la función objeto de acuerdo a sus necesidades.

Los parámetros de una función objeto deben ser distinguidos de su estado, que consiste de todos los atributos de la clase que son computados durante la ejecución del algoritmo, pero no directamente dados o requeridos por el usuario. Éstos conceptos se muestran en la figura A.1.

Hay muchas razones para tener una clase independiente para los parámetros. Se pueden crear múltiples instancias con diferentes conjuntos de valores que puede cambiar la funcionalidad de una función objeto con un simple llamado a la función **setParameters()**. También es posible cargar y salvar parámetros en un archivo, o bien puede darlos a una interfaz de usuario en donde se pueda elegir entre múltiples valores.

Los parámetros contienen valores directamente especificados por el usuario y ellos no deberían ser modificados por la función objeto. Si el programador piensa que el algoritmo debe cambiar un parámetro, Ésto es un signo de que éste parámetro debería ser copiado en algún lugar al inicio del algoritmo, como varible local o variable de estado.

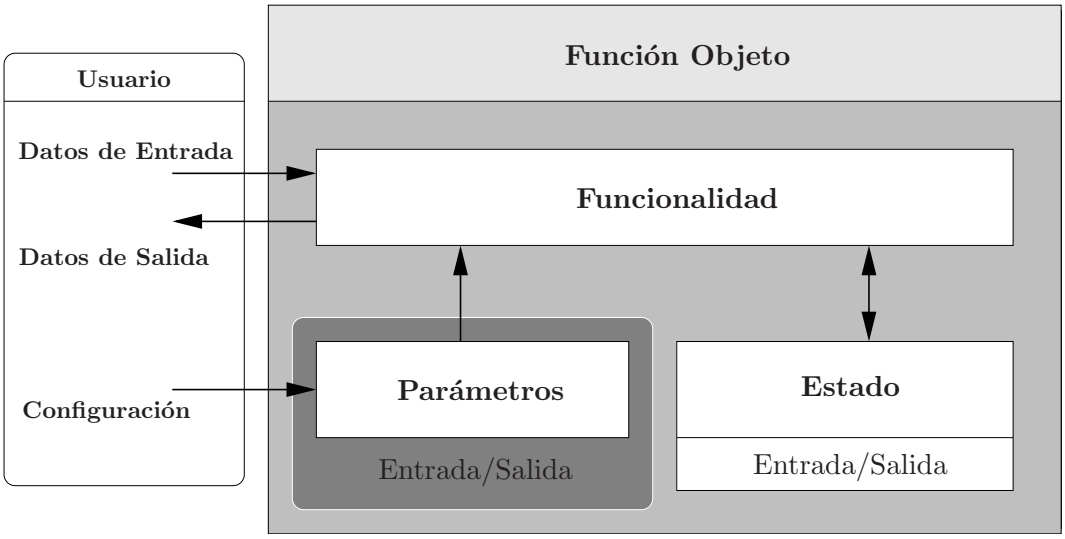


Figura A.1: Arquitectura de una función objeto. El usuario puede cambiar su comportamiento mediante los parámetros. La función objeto también tiene un estado, que enventualmente al igual que los parámetros pueden ser guardados[4]

La funcionalidad la función objeto es siempre accesada por los métodos *apply*. Ellos esperan datos de entrada (usualmente referencias a objetos como matrices o imágenes), y referencias a objetos de salida (referencia a contenedores donde el resultado será escrito). Todas las funciones objeto deben proveer una interfaz basada en objetos de parámetros y métodos *apply*, con el objetivo de brindar una vía uniforme de acceso a la funcionalidad de la función objeto.

Apéndice B

Referencia de Clases

En éste apéndice se presentan la referencias de las clases implementadas en la LTI-Lib para éste trabajo. Para reducción de ruido: *Medias no Locales*; para mejoramiento de contraste: *Mejoramiento de Gradiente*; finalmente, para mejoramiento de nitidez: *Máscara de desenfoque*. La referencia contiene el detalle de las funciones miembro y de los datos miembro de cada clase. Para más detalles del la LTI-Lib refiérase al Apéndice A.

B.1 Medias no Locales

La clase lti::nonLocalMeansDenoising contiene el algoritmo de medias no locales descrito en la Sección 4.1.5 y se incluye mediante:

```
#include <ltiNonLocalMeansDenoising.h>
```

El diagrama de herencia para lti::nonLocalMeansDenoising: se muestra en la Figura B.1.

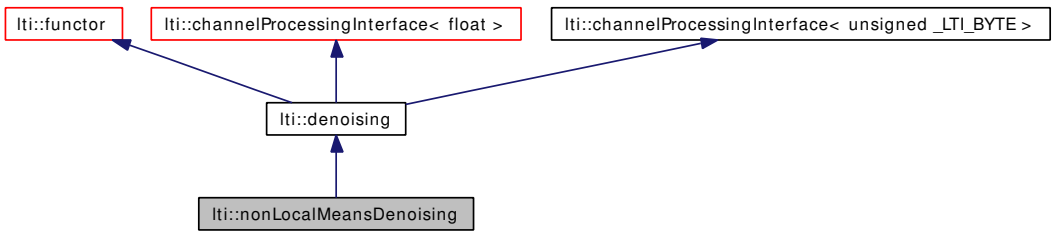


Figura B.1: Diagrama de herencia de la clase lti::nonLocalMeansDenoising

El diagrama de colaboración para la clase lti::nonLocalMeansDenoising se muestra en la Figura B.2.

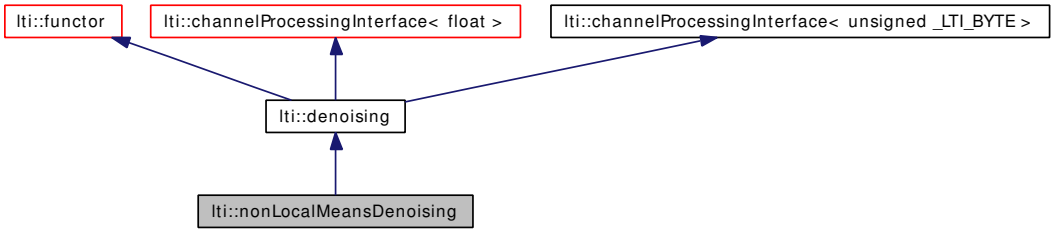


Figura B.2: Diagrama de colaboración de la clase lti::nonLocalMeansDenoising

B.1.1 Descripción Detallada

Ésta clase implementa el algoritmo descrito en el artículo "A non-local algorithm for image denoising" que fue presentado por Antoni Buades, Bartomeu Coll and Jean-Michel Morel. Éste es un algoritmo para reducción de ruido con poca o nula distorsión en el resultado.

Dada una imagen discreta y ruidosa $\mathcal{N} = \mathcal{N}(p_m)|p_m \in \mathcal{N}$, el valor estimado $NL[\mathcal{N}](p_m)$, para un píxel es calculado como el promedio ponderado de todos los píxeles de la imagen,

$$NL[\mathcal{N}](p_m) = \sum_{p_n \in \mathcal{N}} w(p_m, p_n)\mathcal{N}(p_n)$$

donde la familia de pesos $w(p_m, p_n)_{p_n}$ dependen de la similitud entre los píxeles p_m y p_n , y satisface las condiciones usuales $0 \leq w(p_m, p_n) \leq 1$ and $\sum_{p_n} w(p_m, p_n) = 1$.

La similitud entre dos píxeles p_m y p_n depende de la similitud de la intensidad de los niveles de gris de los vectores $\mathcal{N}(\mathcal{V}p_m)$ y $\mathcal{N}(\mathcal{V}p_n)$, donde $\mathcal{V}_k$ denota un vecindario cuadrado de tamaño fijo y centrado en el píxel p_{m_k}. Ésta similitud es medida como una función de decrecimiento de la distancia euclideana ponderada.

Los píxeles con un vecindario con un nivel de gris similar a $\mathcal{N}(\mathcal{V}_{p_m})$, tienen una ponderación mayor en el promedio. Éstos pesos se definen como

$$w(p_m, p_n) = \frac{1}{Z(p_m)} e^{-\frac{||\mathcal{N}(\mathcal{V}_{p_m}) - \mathcal{N}(\mathcal{V}_{p_n})||}{h^2}}$$

donde $Z(p_m)$ es la constante de normalización,

$$Z(p_m) = \sum_{p_n} e^{-\frac{||\mathcal{N}(\mathcal{V}_{p_m}) - \mathcal{N}(\mathcal{V}_{p_n})||}{h^2}}$$

y el parámetro h actúa como el grado de filtrado. Éste controla la caída de la función exponencial y así mismo la caída de los pesos como un función de distancias euclideanas. Para la descripción completa de éste algoritmo refiérase a la Sección 4.1.5.

Funciones Miembro Públicas

- nonLocalMeansDenoising ()
- nonLocalMeansDenoising (const parameters &par)
- nonLocalMeansDenoising (const nonLocalMeansDenoising &other)
- virtual ∼nonLocalMeansDenoising ()
- bool apply (channel &srcdest) const
- bool apply (channel8 &srcdest) const
- bool apply (const channel &src, channel &dest) const
- bool apply (const channel8 &src, channel8 &dest) const
- nonLocalMeansDenoising & copy (const nonLocalMeansDenoising &other)
- nonLocalMeansDenoising & operator= (const nonLocalMeansDenoising &other)
- virtual const std::string & name () const
- virtual nonLocalMeansDenoising ∗ clone () const
- virtual nonLocalMeansDenoising ∗ newInstance () const
- const parameters & getParameters () const

Clases

- class parameters

  Parámetros para la clase nonLocalMeansDenoising.

B.1.2 Documentación para el Constructor y el Destructor

lti::nonLocalMeansDenoising::nonLocalMeansDenoising ()

Constructor por defecto.

lti::nonLocalMeansDenoising::nonLocalMeansDenoising (const parameters & *par*)

Construye una función objeto usando los parámetros dados.

lti::nonLocalMeansDenoising::nonLocalMeansDenoising (const nonLocalMeansDenoising & *other*)

Constructor de copia.

Parameters:

other el objeto a ser copiado.

virtual lti::nonLocalMeansDenoising::~nonLocalMeansDenoising () [virtual]

Destructor.

B.1.3 Documentación de las Funciones Miembro

bool lti::nonLocalMeansDenoising::apply (channel & *srcdest*) const [virtual]

Opera en los argumentos dados.

Parámetros:

srcdest channel con los datos de origen. El resultado se guardará aquí también.

Retorna:

returna **true** si es exitoso, y **false** de lo contrario.

Reimplementado de lti::denoising.

bool lti::nonLocalMeansDenoising::apply (channel8 & *srcdest*) const [virtual]

Opera en los argumentos dados.

Parámetros:

srcdest channel8 con los datos de origen. El resultado se guardará aquí también.

Retorna:

true si es exitoso, y **false** de lo contrario.

Reimplementado de lti::denoising.

bool lti::nonLocalMeansDenoising::apply (const channel & *src*,channel & *dest) const `[virtual]`

Opera en una copia de los argumentos dados.

Parámetros:

src channel con los datos de origen.

dest channel donde se guardará el resultado.

Retorna:

true si es exitoso, y **false** de lo contrario.

Implementa lti::denoising.

bool lti::nonLocalMeansDenoising::apply (const channel8 & *src*, channel8 & *dest*) const `[virtual]`

Opera en una copia de los argumentos dados.

Parámetros:

src channel8 con los datos de origen.

dest channel8 donde se guardará el resultado.

Retorna:

true si es exitoso, y **false** de lo contrario.

Implementa lti::denoising.

nonLocalMeansDenoising & lti::nonLocalMeansDenoising::copy (const nonLocalMeansDenoising & *other*)

Copia datos de otra ("other") función objeto.

Parámetros:

other la función objeto a ser copiada.

Retorna:

una referencia a ésta función objeto.

nonLocalMeansDenoising& lti::nonLocalMeansDenoising::operator= (const nonLocalMeansDenoising & *other*)

Alias para el miembro de la copia.

Parámetros:

other la función objeto a ser copiada

Retorna:

una referencia a ésta función objeto.

virtual const std::string& lti::nonLocalMeansDenoising::name () const `[virtual]`

Retorna el nombre completo de la clase.

Reimplementado de lti::denoising.

virtual nonLocalMeansDenoising* lti::nonLocalMeansDenoising::clone () const `[virtual]`

Retorna un puntero a un clon de ésta función objeto.

Implementa lti::denoising.

virtual nonLocalMeansDenoising* lti::nonLocalMeansDenoising::newInstance () const `[virtual]`

Retorna un puntero a una nueva instancia de ésta función objeto.

Implementa lti::denoising.

const parameters& lti::nonLocalMeansDenoising::getParameters () const

Retorna los parámetros usados.

Reimplementado de lti::denoising.

B.1.4 Documentación de los Datos Miembro

Los siguientes son los datos miembros de la clase lti::nonLocalMeansDenoising::parameters.

int lti::nonLocalMeansDenoising::parameters::similarityWindowSize

Tamaño de la ventana de similitud.

Valor por defecto: 5x5

int lti::nonLocalMeansDenoising::parameters::searchWindowSize

Tamaño de la ventana de búsqueda, donde el algoritmo se aplicará. Un tamaño más grande mejora el resultado de la reducción de ruido pero es más lento. El algoritmo completo en teoría emplea la imagen completa, pero el costo computacional no justifica el mejoramiento en el resultado.

Valor por defecto: 11x11

double lti::nonLocalMeansDenoising::parameters::variance

Ésta la varianza empleada por la máscara gaussiana.

Valor por defecto: 5.0

float lti::nonLocalMeansDenoising::parameters::filterDegree

Éste parámetro mide el grado de filtrado de la imagen obtenida. Cuando se sabe que la desviación estándar del ruido σ el valor del grado de filtrado dependerá de ello. Un buen valor está entre $10 * \sigma^2$ and $15 * \sigma^2$. Valores muy grandes pueden causar imágenes difuminadas.

Valor por defecto: 0.006f

B.2 Mejoramiento de Gradiente

La clase lti::galConstrastEnhancement contiene el algoritmo de mejoramiento de gradiente descrito en la Sección 4.2.2 y se incluye mediante:

```
#include <ltiGALContrastEnhancement.h>
```

El diagrama de herencia para lti::galContrastEnhancement: se muestra en la Figura B.3.

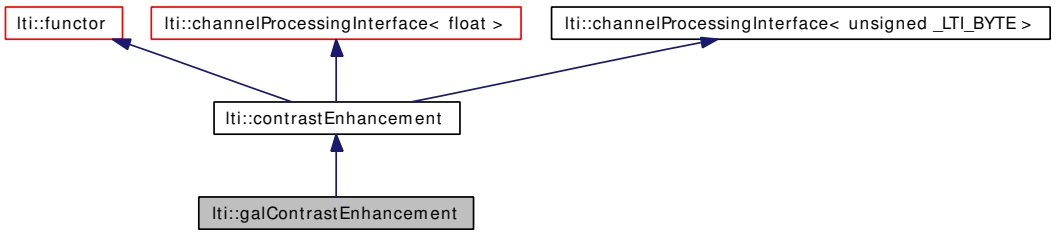


Figura B.3: Diagrama de herencia de la clase lti::galContrastEnhancement

El diagrama de colaboración para la clase lti::galContrastEnhancement se muestra en la Figura B.4.

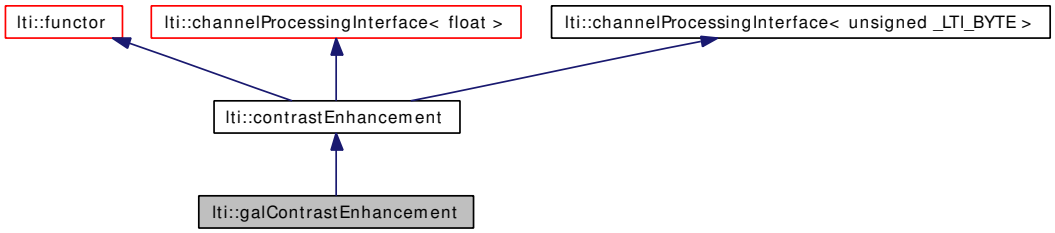


Figura B.4: Diagrama de colaboración de la clase lti::galContrastEnhancement

B.2.1 Descripción Detallada

Ésta clase implementa el algoritmo descrito en el artículo "Greedy Algorithm for Local Contrast Enhacement" and "Contrast Enhacement of Images using Human contrast Sensivity" presentado por Kartic Subr, Aditi Majumder and Sandy Irani.

El nombre de ésta clase galContrastEnhancement se deriva de las siglas en inglés "Greedy Algorithm for Local Contrast Enhacement, que significan Algoritmo Greedy para el Mejoramiento Local de Contraste.

Éste algoritmo es un método de mejoramiento local de contraste, cuyo el objetivo es mejorarar los gradientes locales, que están directamente relacionados con el contraste local

de una imagen. El mejoramiento de contraste es propuesto como un problema de optimización que maximiza el contraste local de una imagen. La función objetivo a maximizar es

$$f(\mathcal{N}) = \frac{1}{4|\mathcal{N}|} \sum_{p_m \in (\mathcal{N}, \widehat{\mathcal{I}})} \sum_{p_n \in N_4(p_m)} \frac{\widehat{\mathcal{I}}(p_m) - \widehat{\mathcal{I}}(p_n)}{\mathcal{N}(p_m) - \mathcal{N}(p_n)}$$

sujeto a las siguientes condiciones,

$$1 \leq \frac{\widehat{\mathcal{I}}(p_m) - \widehat{\mathcal{I}}(p_n)}{\mathcal{N}(p_m) - \mathcal{N}(p_n)} \leq (1 + \delta)$$

$$L \leq \widehat{\mathcal{I}}(p_m) \leq U$$

donde las funciones escalares $\mathcal{N}$ y $\widehat{\mathcal{I}}$ representan los niveles de gris en el píxel p_m de la imagen de entrada and salida, respectively, $\mathcal{N}$ es la imagen de entrada que supone con bajo contraste, $|\mathcal{N}|$ denota la cardinalidad de $\mathcal{N}$, $\mathcal{V}_4(p_m)$ denota el conjunto de cuatro vecinos de p_m. L y U son los límites inferior y superior de los valores de los niveles de gris, y $\delta > 0$ es un solo parámetro que controla la cantidad de mejoramiento logrado. La primera condición asegura que un mejoramiento limitado de los gradientes. La segunda condición asegura que la satura no tendrá valores de intensidad saturados.

Funciones Miembro Públicas

- galContrastEnhancement ()
- galContrastEnhancement (const parameters &par)
- galContrastEnhancement (const float delta)
- galContrastEnhancement (const galContrastEnhancement &other)
- virtual ∼galContrastEnhancement ()
- bool apply (channel &srcdest) const
- bool apply (channel8 &srcdest) const
- bool apply (const channel &src, channel &dest) const
- bool apply (const channel8 &src, channel8 &dest) const
- galContrastEnhancement & copy (const galContrastEnhancement &other)
- galContrastEnhancement & operator= (const galContrastEnhancement &other)
- virtual const std::string & name () const
- virtual galContrastEnhancement ∗ clone () const
- virtual galContrastEnhancement ∗ newInstance () const
- const parameters & getParameters () const

Clases

- class parameters

  Parámetros para la clase galContrastEnhancemet.

B.2.2 Documentación del Constructor y Destructor

lti::galContrastEnhancement::galContrastEnhancement ()

Constructor por defecto.

lti::galContrastEnhancement::galContrastEnhancement (const parameters & *par*)

Construye una función objeto de mejoramiento de contraste con los parámetros dados.

lti::galContrastEnhancement::galContrastEnhancement (const float *delta*)

Construye una función objeto de mejoramiento de contraste usando el δ dado.

Parámetros:

delta cantidad de contraste logrado.

lti::galContrastEnhancement::galContrastEnhancement (const galContrastEnhancement & *other*)

Constructor de copia.

Parámetros:

other objeto a ser copiado

virtual lti::galContrastEnhancement::~galContrastEnhancement () [virtual]

Destructor.

B.2.3 Documentación de las Funciones Miembro

bool lti::galContrastEnhancement::apply (channel & *srcdest*) const `[virtual]`

Opera en los argumentos dados.

Parámetros:

srcdest channel con los datos de origen. El resultado se guardará aquí también.

Retorna:

true si fue exitoso, **false** de lo contrario.

Implementa lti::contrastEnhancement.

bool lti::galContrastEnhancement::apply (channel8 & *srcdest*) const `[virtual]`

Opera en los argumentos dados.

Parámetros:

srcdest channel8 con los datos de origen. El resultado se guardará aquí también.

Retorna:

true si fue exitoso, **false** de lo contrario.

Implementa lti::contrastEnhancement.

bool lti::galContrastEnhancement::apply (const channel & *src*, channel & *dest*) const `[virtual]`

Opera en un copia de los argumentos dados.

Parámetros:

src channel con los datos de origen.

dest channel donde se guardará el resultado.

Retorna:

true si es exitoso, **false** de lo contrario.

Implementa lti::contrastEnhancement.

bool lti::galContrastEnhancement::apply (const channel8 & *src*, channel8 & *dest*) const `[virtual]`

Opera en un copia de los argumentos dados.

Parámetros:

src channel8 con los datos de origen.

dest channel8 donde se guardará el resultado.

Retorna:

true si es exitoso, **false** de lo contrario.

Implementa lti::contrastEnhancement.

galContrastEnhancement& lti::galContrastEnhancement::copy (const galContrastEnhancement & *other*)

Copia los datos de otra ("other") función objeto.

Parámetros:

other la función objeto a ser copiada

Retorna:

una referencia a ésta función objeto

galContrastEnhancement& lti::galContrastEnhancement::operator= (const galContrastEnhancement & *other*)

Alias para el miembro de la copia.

Parámetros:

other la función objeto a ser copiada

Retorna:

una referencia a ésta función objeto

virtual const std::string& lti::galContrastEnhancement::name () const `[virtual]`

Retorna el nombre completo de la clase función objeto.

Reimplementado de lti::contrastEnhancement.

virtual galContrastEnhancement* lti::galContrastEnhancement::clone () const `[virtual]`

Retorna un puntero a un clon de ésta función objeto.

Implementa lti::contrastEnhancement.

virtual galContrastEnhancement* lti::galContrastEnhancement::newInstance () const `[virtual]`

Retorna un puntero a una nueva instancia de ésta función objeto.

Implementa lti::contrastEnhancement.

const parameters& lti::galContrastEnhancement::getParameters () const

Retorna los parámetros usados.

B.2.4 Documentación de los Datos Miembro

Los siguientes son los datos miembros de la clase lti::galContrastEnhancement::parameters.

float lti::galContrastEnhancement::parameters::delta

Delta δ: cantidad de mejoramiento que se pretende lograr.

Valor por defecto: 0.5.

B.3 Máscaras para Mejoramiento de Nitidez

La clase lti::sharpeningKernels contiene dos máscara empleadas para el mejoramiento de nitidez y se incluye con:

```
#include <ltiSharpeningKernels.h>
```

La clase lti::unsharpMasking emplea las máscaras de ésta clase para mejorar la nitidez de imágenes. El diagrama de herencia para lti::sharpeningKernels: se muestra en la Figura B.5.

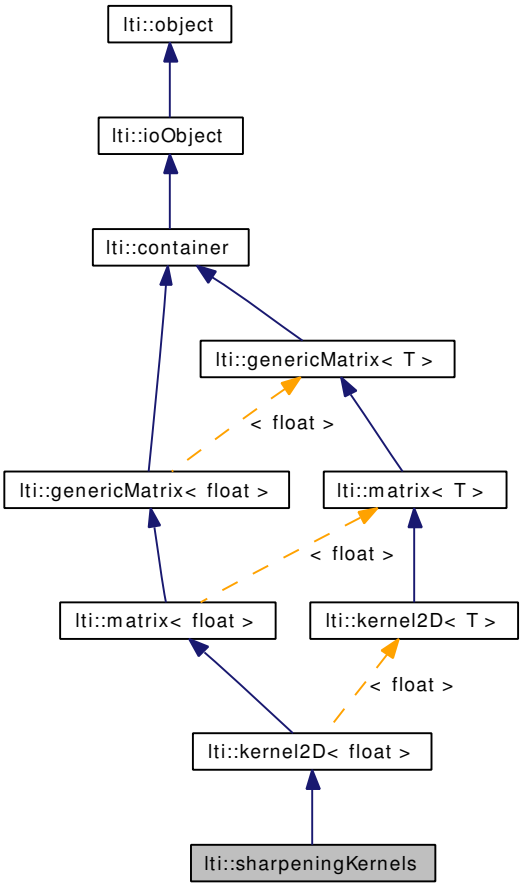


Figura B.5: Diagrama de herencia de la clase lti::sharpeningKernels

El diagrama de colaboración para la clase lti::sharpeningKernels se muestra en la Figura B.6.

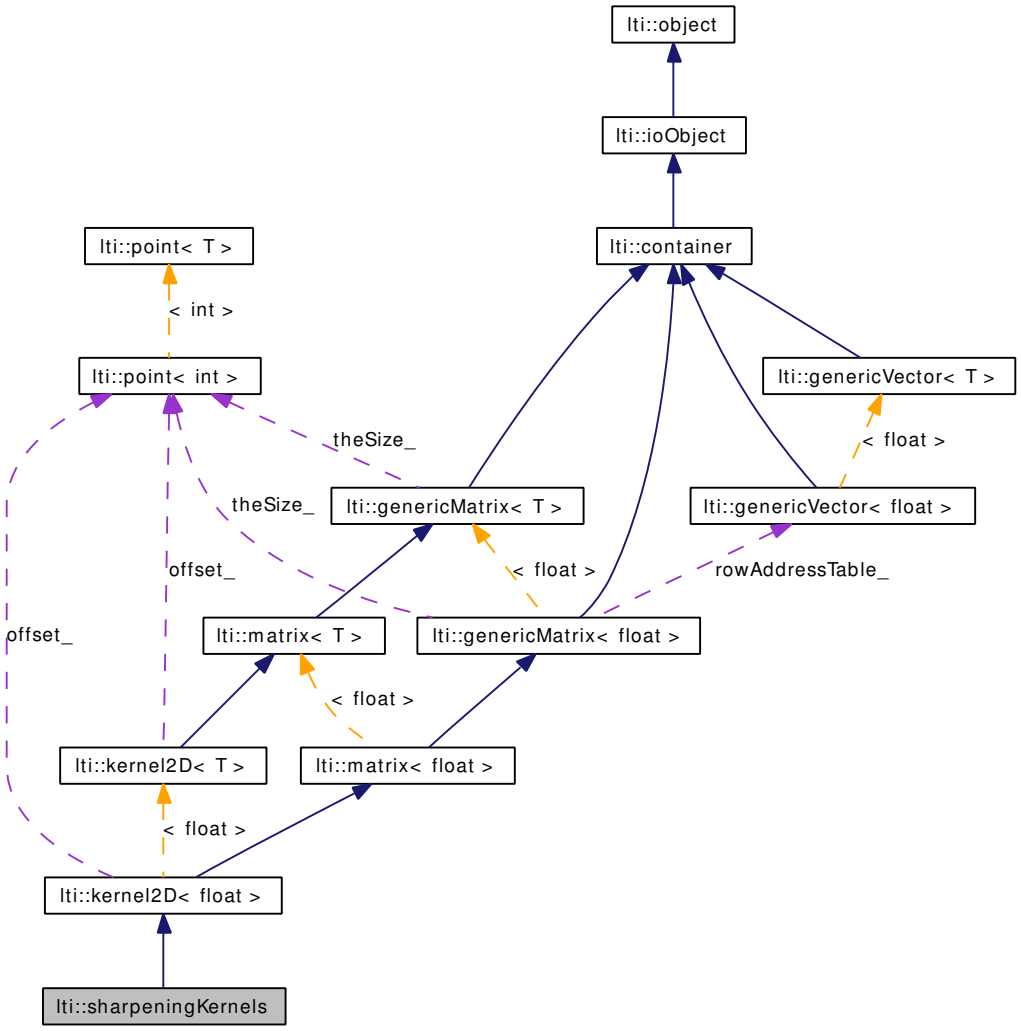


Figura B.6: Diagrama de colaboración de la clase lti::sharpeningKernels

B.3.1 Descripción detallada

Ésta clase contiene máscaras (en inglés kernels, masks, filters) para el mejoramiento de la nitidez en imágenes, basadas en el algoritmo *máscara de desenfoque y filtrado de alto impulso*.

Ésta es una técnica común para mejorar la nitidez de imágenes, que consiste en sustraer un versión difuminada de la imagen de la imagen en sí misma. Ésto se expresa de la siguiente manera

$$\widehat{\mathcal{I}}(i,j) = \mathcal{N}(i,j) - \mathcal{N}_d(i,j)$$

donde $\widehat{\mathcal{I}}(i,j)$ denota la imagen que se le mejoró la nitidez mediante la técnica de máscara de desenfoque, y $\mathcal{N}_d(i,j)$ es una versión difuminda de $\mathcal{N}(i,j)$.

Una generalización de la máscara de desenfoque se llama *filtrado de alto impulso*, y se define como

$$\widehat{\mathcal{I}}(i,j) = C\mathcal{N}(i,j) - \mathcal{N}_d(i,j)$$

donde $C \geq 1$ es una constante. Reordenando obtenemos,

$$\widehat{\mathcal{I}}(i,j) = \mathcal{N}(i,j) * C - \mathcal{N}(i,j) * \mathcal{M}(m,n)$$
$$\widehat{\mathcal{I}}(i,j) = \mathcal{N}(i,j) * [C\delta(i,j) - \mathcal{M}(m,n)]$$

Ésto significa que para obtener una imagen con mayor nitidez se necesita convolucionar una imagen con $[C\delta(i,j) - \mathcal{M}(m,n)]$, donde $\mathcal{M}$ es una máscara gaussiana o laplaciana.

Ésta clase implementa la máscara definida por $[C\delta(i,j) - \mathcal{M}(m,n)]$. Los parámetros definen el tamaño, la constante C, el tipo de máscara, ya sea, laplaciana o gaussiana, el tipo de vecidario, ya sea de 8 o 4 vecinos para el caso de la máscara laplaciana y la varianza para el caso de la máscara gaussiana.

Éste es un ejemplo de una máscara laplaciana de 3×3, con un vecindario de 4 vecinos:

```
 0    -1    0
-1    A+4  -1
 0    -1    0
```

Éste es un ejemplo de una máscara laplaciana de 3×3, con un vecindario de 8 vecinos:

```
-1    -1   -1
-1    A+8  -1
-1    -1   -1
```

Para convolucionar una máscara laplaciana con un canal siga éste ejemplo:

```
// el canal a ser filtrado:
lti::channel data;

// ... inicialice el canal aquí ...
```

```
// máscara laplaciana de 3x3, constante de alto impulso
// and un vecindario de 4 vecinos
lti::sharpening kernel(3,1.1,sharpeningKernels::Laplacian,
sharpeningKernels::FourNeighbor);

// operador de convolución
lti::convolution filter(kernel);

// filtrar el canal para mejorar su nitidez
filter.apply(data);
```

Para convolucionar una máscara gaussiana con un canal siga éste ejemplo:

```
// el canal a ser filtrado:
lti::channel data;

// ... inicialice el canal aquí ...

// máscara gaussiana de 3x3, constante de alto impulso
// and una varianza de 5.0
lti::sharpening kernel(3,1.1,sharpeningKernels::Gaussian,5.0);

// operador de convolución
lti::convolution filter(kernel);

// filtrar el canal para mejorar su nitidez
filter.apply(data);
```

Nota: la convolución con éstas máscaras puede dar valores debajo de cero o sobre el limite superior del tipo (>1 en floats o >255 en ubytes). Así, que debe limitarlos.

Tipos Públicos

- enum eKernelType {Laplacian,Gaussian}
- enum eNeighborType {FourNeighbor,EightNeighbor}

Funciones Miembro Públicas

- sharpeningKernels (const int KernelSize=3, const float C=1.0f, eKernelType kernelType=Laplacian, eNeighborType neighborType=FourNeighbor, const double &variance=1.3)

- sharpeningKernels (const int KernelSize, const float A, eKernelType kernelType, const double &variance)
- void generate (const int kernelSize, const float C, eKernelType kernelType, eNeighborType neighborType, const double &variance)

B.3.2 Documentación de Enumeraciones Miembro

enum lti::sharpeningKernels::eKernelType

Tipo de máscara: Laplaciana o Gaussiana.

Enumerador:

[Laplacian] Máscara Laplaciana. [Gaussiana] Máscara Gaussiana.

enum lti::sharpeningKernels::eNeighborType

Tipo de vecindario: 4 u 8 vecinos.

Enumerador:

[FourNeighbor] Máscara de 4 vecinos. [EightNeighbor] Máscara de 8 vecinos.

B.3.3 Documentación del Constructor y Destructor

lti::sharpeningKernels::sharpeningKernels (const int *KernelSize* = `3`, const float *C* = `1.0f`, eKernelType *kernelType* = `Laplacian`, eNeighborType *neighborType* = `FourNeighbor`, const double & *variance* = `1.3`)

Constructor.

Inicializa la mácara con los valores especificador. Si un valor inválido es dado, una máscara vacía se creará.

Parámetros:

size tamaño de la máscara en una dimensión.

C constante de alto impacto.

kernelType tipo de máscara (Laplaciana or Gaussiana).

neighborType tipo de vecindario de la máscara. Ésto solo es válido para la máscara Laplaciana.

variance varianza de la máscara. Ésto solo es válido para la máscara Gaussiana.

lti::sharpeningKernels::sharpening (const int *KernelSize*, const float *C*,*kernelType*, const double & *variance*)

Constructor.

Inicializa la mácara con los valores especificador. Si un valor inválido es dado, una máscara vacía se creará.

Parámetros:

size tamaño de la máscara en una dimensión.

C constante de alto impacto.

kernelType tipo de máscara (Laplaciana or Gaussiana).

variance varianza de la máscara. Ésto es válido solo para la máscara gaussiana.

B.3.4 Documentación de las Funciones Miembro

void lti::sharpeningKernels::generate (const int *kernelSize*, const float *C*, eKernelType *kernelType*, eNeighborType *neighborType*, const double & *variance*)

Inicializa la máscara con los valores especificados. Si un valor inválido es dado, una máscara vacía se creará.

Parámetros:

size tamaño de la máscara en una dimensión.

C constante de alto impacto.

kernelType tipo de máscara (Laplaciana or Gaussiana).

variance varianza de la máscara. Ésto es válido solo para la máscara gaussiana.

B.4 Máscara de Desenfoque

La clase lti::unsharpMasking contiene el algoritmo descrito en la Sección 4.3.1 y se incluye con:

```
#include <ltiUnsharpMasking.h>
```

El diagrama de herencia para lti::sharpeningKernels: se muestra en la Figura B.7.

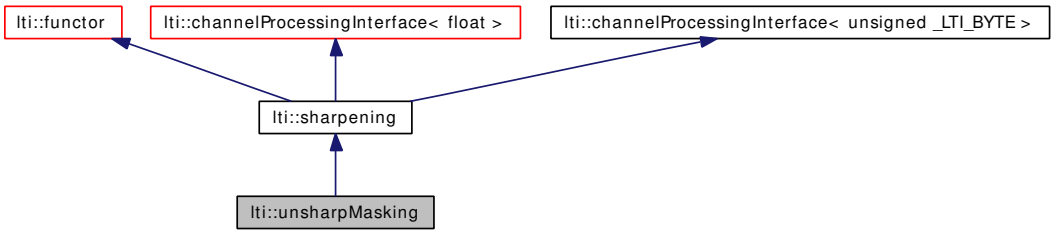


Figura B.7: Diagrama de herencia de la clase lti::unsharpMasking

El diagrama de colaboración para la clase lti::unsharpMasking se muestra en la Figura B.8.

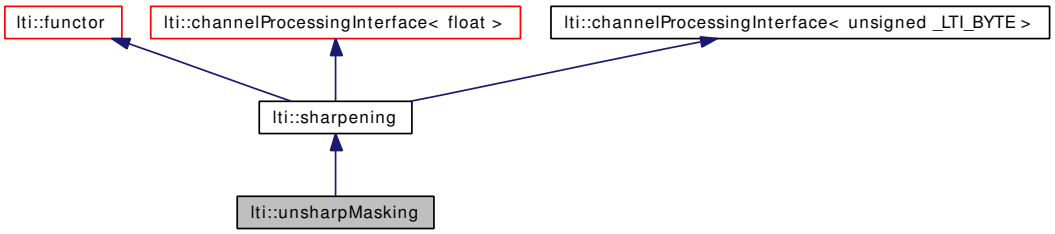


Figura B.8: Diagrama de colaboración de la clase lti::unsharpMasking

B.4.1 Descripción detallada

Ésta es una técnica común para mejorar la nitidez de imágenes, que consiste en sustraer un versión difuminada de la imagen de la imagen en sí misma. Ésto se expresa de la siguiente manera

$$\widehat{\mathcal{I}}(i,j) = \mathcal{N}(i,j) - \mathcal{N}_d(i,j)$$

donde $\widehat{\mathcal{I}}(i,j)$ denota la imagen que se le mejoró la nitidez mediante la técnica de máscara de desenfoque, y $\mathcal{N}_d(i,j)$ es una versión difuminda de $\mathcal{N}(i,j)$.

Una generalización de la máscara de desenfoque se llama *filtrado de alto impulso*, y se define como

$$\widehat{\mathcal{I}}(i,j) = C\mathcal{N}(i,j) - \mathcal{N}_d(i,j)$$

donde $C \geq 1$ es una constante. Reordenando obtenemos,

$$\widehat{\mathcal{I}}(i,j) = \mathcal{N}(i,j) * C - \mathcal{N}(i,j) * \mathcal{M}(m,n)$$
$$\widehat{\mathcal{I}}(i,j) = \mathcal{N}(i,j) * [C\delta(i,j) - \mathcal{M}(m,n)]$$

Ésto significa que para obtener una imagen con mayor nitidez se necesita convolucionar una imagen con $[C\delta(i,j) - \mathcal{M}(m,n)]$, donde $\mathcal{M}$ es una máscara gaussiana o laplaciana.

Ésta clase implementa la máscara definida por $[C\delta(i,j) - \mathcal{M}(m,n)]$. Los parámetros definen el tamaño, la constante C, el tipo de máscara, ya sea, laplaciana o gaussiana, el tipo de vecidario, ya sea de 8 o 4 vecinos para el caso de la máscara laplaciana y la varianza para el caso de la máscara gaussiana. Vea la documentación de la clase lti::sharpeningKernels para más información acerca de las máscaras.

Los parámetros definen el tamaño, la constante de alto impulso C, el tipo de máscara (Laplaciana o Gaussiana) y el tipo de vecindario (4 u 8 vecinos).

Funciones Miembro Públicas

- unsharpMasking ()
- unsharpMasking (const parameters&par)
- unsharpMasking (const unsharpMasking &other)
- virtual ∼unsharpMasking ()
- bool apply (channel8 &srcdest) const
- bool apply (channel &srcdest) const
- bool apply (const channel8 &src, channel8 &dest) const
- bool apply (const channel &src, channel &dest) const
- unsharpMasking & copy (const unsharpMasking &other)
- unsharpMasking & operator= (const unsharpMasking &other)
- virtual const std::string & name () const
- virtual unsharpMasking ∗ clone () const
- virtual unsharpMasking ∗ newInstance () const
- const parameters & getParameters () const

Clases

- parameters

  Parámetros para la clase unsharpMasking.

B.4.2 Documentación del Constructor y Destructor

lti::unsharpMasking::unsharpMasking ()

Constructor por defecto.

lti::unsharpMasking::unsharpMasking (const parameters & *par*)

Contruye una función objeto de mejoramiento de nitidez con los parámetros dados.

lti::unsharpMasking::unsharpMasking (const unsharpMasking & *other*)

Constructor de copia..

Parámetros:

other objeto a ser copiado

virtual lti::unsharpMasking::~unsharpMasking () [virtual]

Destructor.

B.4.3 Documentación de las Funciones Miembro

bool lti::unsharpMasking::apply (channel8 & *srcdest*) const [virtual]

Opera en los argumentos dados.

Parámetros:

srcdest channel8 con los datos de origen. El resultado se guardará aquí también.

Retorna:

true si fue exitoso, **false** de lo contrario.

Implementa lti::sharpening.

bool lti::unsharpMasking::apply (channel & *srcdest*) const `[virtual]`

Opera en los argumentos dados.

Parámetros:

srcdest channel con los datos de origen. El resultado se guardará aquí también.

Retorna:

true si fue exitoso, **false** de lo contrario.

Implementa lti::sharpening.

bool lti::unsharpMasking::apply (const channel8 & *src*, channel8 & *dest*) const `[virtual]`

Opera en una copia de los argumentos dados.

Parámetros:

src channel8 con los datos de origen.

dest channel8 donde se guardará el resultado.

Retorna:

true si fue exitoso, **false** de lo contrario.

Implementa lti::sharpening.

bool lti::unsharpMasking::apply (const channel & *src*, channel & *dest*) const `[virtual]`

Opera en una copia de los argumentos dados.

Parámetros:

src channel con los datos de origen.

dest channel donde se guardará el resultado.

Retorna:

true si fue exitoso, **false** de lo contrario.

Implementa lti::sharpening.

unsharpMasking& lti::unsharpMasking::copy (const unsharpMasking & *other*)

Copia datos de otra ("other") función objeto de mejoramiento de nitidez.

Parámetros:

other la función objeto a ser copiada

Returns:

una refencia a ésta función objeto

unsharpMasking& lti::unsharpMasking::operator= (const unsharpMasking & *other*)

Alias para el miembro de la copia.

Parámetro:

other función objeto a ser copiada

Retorna:

una refencia a ésta función objeto

virtual const std::string& lti::unsharpMasking::name () const [virtual]

Retorna el nombre completo de la clase.

Reimplementado de lti::sharpening.

virtual unsharpMasking* lti::unsharpMasking::clone () const `[virtual]`

Retorna un puntero a un clon de ésta función objeto.

Implementada de lti::sharpening.

virtual unsharpMasking* lti::unsharpMasking::newInstance () const `[virtual]`

Retorna un puntero a una nueva instancia de ésta función objeto.

Implementa lti::sharpening.

const parameters& lti::unsharpMasking::getParameters () const

Retorna los parámetros usados.

Reimplementado de lti::sharpening.

B.4.4 Documentación de los Datos Miembro

Los siguientes son los datos miembros de la clase lti::unsharpMasking::parameters.

int lti::unsharpMasking::parameters::kernelSize

Tamaño de la máscara en una dimensión.

Valor por defecto: 3x3.

float lti::unsharpMasking::parameters::C

Constante de alto impacto.

Valor por defecto: 1.

sharpeningKernels::eKernelType lti::unsharpMasking::parameters::kernelType

Tipo de máscara para la máscara de desenfoque (Laplaciana o Gaussiana).

Default value: sharpeningKernels::Laplacian.

sharpeningKernels::eNeighborType lti::unsharpMasking::parameters::neighborType

Éste es el tipo de vecindario considerado en el caso de la máscara laplaciana.

Valor por defecto: sharpeningKernels::FourNeighbor.

double lti::unsharpMasking::parameters::variance

Varianza para la máscara gaussiana.

Valor por defecto: 5.0.

Índice alfabético

adquisición, xiii
AFLP, 1
algoritmos, 39
    genéticos, 6
    para la reducción de ruido, 39
    para mejoramiento de contraste, 53
análisis
    de imagenes de geles, 1, 3

bandas, 2
biblioteca, 6
binarización de una imagen, 2

caracterización molecular de organismos, 1
carga eléctrica, 1
conclusiones, 103
contraste, xiii, 4, 5
convolución
    con una máscara, 11

diagrama
    de dispersión, 36
Difuminación, 38
distorsión
    geométrica, 2
distorsiones geométricas de las bandas, 4
dominio
    de la frecuencia, 13
    espacial, 9

ecualización de histograma, 54
electroforesis, xiii, 1
elemento de imagen, 7, 8
entropía, 36
error cuadrático medio inverso, 34
etapa
    de pre-procesamiento, 4
etapa de preprocesamiento, 4

filtrado
    en el dominio de la frecuencia, 13
    lineal, 10
    no lineal, 11
frente
    de pareto, 31
frente de pareato
    de los algoritmos de reducción de ruido, 73
frente de pareto, xiii, 6
fuerza electromotriz, 1
función
    de aptitud, xiii
funciones
    de aptitud, 31, 33
    de aptitud de contraste, 35
    de aptitud de nitidez, 36
    de aptitud de ruido, 34
    de densidad de probabilidad, 40
    de distribución acumulada, 60

histograma, 54

imágenes
    de geles, 3, 4
    digitales, 5

digitalizadas, 6
procesadas por segundo, 34
imagen
de gel, 2
de gel de electroforesis, 1
digital, 8
digitalizada, 4

LTI-Lib, 6
lti::galContrastEnhancement, 123
lti::nonLocalMeansDenoising, 115
lti::sharpeningKernels, 130
lti::unsharpMasking, 136

máscara, 10
de desenfoque, 136
máscaras
para mejoramiento de nitidez, 130
medias no locales, 48, 115
mejoramiento
de contraste, 4, 6
de contraste promedio, 36
de gradiente, 61
de imágenes, 9
de nitidez, 6
mejoramiento de gradiente, 123
mejorar
contraste, 6
nitidez, 6
modelos de ruido, 39

nitidez, xiv, 4, 5, 38

píxel, 7, 8
parámetros
de ecualización de histograma, 60
del filtro de mediana, 44
del reductor de ruido gaussiano, 43
del reductor de ruido SUSAN, 48
medias no locales, 53
PDF, 40
polímero, 1
principio
de SUSAN, 45
procesamiento
digital de imágenes, xiv, 1, 6
proceso
de binarización, 2

recomendaciones, 106
reducción
de ruido, 4, 6
reducir
ruido, 6
reductor de ruido
gaussiano, 42
SUSAN, 45
referencia de clases, 115
resultado
mejoramiento de nitidez, 95
resultados
en la reducción de ruido, 72
en mejoramiento de contraste, 87
Ruido, xiv
ruido, 4, 5
de método, 34
de método escalar, 35
gaussiano y blanco, 40
impulsional, 41

sistema
de procesamiento digital de imágenes, 4, 7

tamaño molecular, 1

vecindario
de un píxel, 8
visión por computadora, 6
voraz, 61