



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

**Algoritmos para la restauración de
imágenes digitales en colores**

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

Ing. Julio César Castillo García

DIRECTORES DE TESIS

Dr. Edgardo Manuel Felipe Riverón

Dr. Jesús Alberto Martínez Castro



Ciudad de México
Diciembre de 2015



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 02 del mes de diciembre de 2015 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

“Algoritmos para la restauración de imágenes digitales en colores”

Presentada por el alumno:

CASTILLO

GARCÍA

JULIO CÉSAR

Apellido paterno

Apellido materno

Nombre(s)

Con registro:

B	1	3	0	0	7	0
----------	----------	----------	----------	----------	----------	----------

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

Dr. Edgardo Manuel Felipe Riverón

Dr. Jesús Alberto Martínez Castro

Dr. Sergio Suárez Guerra

Dr. Oleksiy Pogrebnyak

Dr. Grigori Sidorov

Dr. Víctor Hugo Ponce Ponce

PRESIDENTE DEL COLEGIO DE PROFESORES

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN

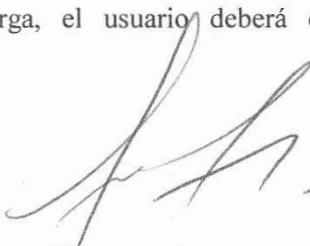


INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México, D.F. el día 03 del mes diciembre del año 2015, el (la) que suscribe Julio César Castillo García alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B130070, adscrito al Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Doctor Edgardo Manuel Felipe Riverón y Doctor Jesús Alberto Martínez Castro y cede los derechos del trabajo intitulado Algoritmos para la restauración de imágenes digitales en colores, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección juliocesar.castillo@gmail.com, edgardo@cic.ipn.mx o macj@cic.ipn.mx. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.


Julio César Castillo García

Nombre y firma

Resumen

Desde los primeros días del arte y la fotografía, la restauración de obras ha sido hecha manualmente por artistas profesionales, ya sea retocando o repintando. Imitando el desempeño de profesionales con técnicas digitales semiautomáticas, la restauración de imágenes es un área activa de investigación. El repintado como una aplicación para la restauración de imágenes ha presentado un importante avance, debido al éxito de los numerosos enfoques utilizados para resolver el problema.

La metodología se divide en tres componentes principales: el primero actúa como entrada por parte del usuario para indicar la zona objetivo del algoritmo y que permite generar la máscara de repintado. El segundo componente es el de cálculo de las prioridades para establecer un orden para la restauración, de forma que las zonas con mayor detalle se procesen antes. El tercer componente se encarga de sintetizar la textura de la imagen y copiar la información más adecuada de acuerdo a los valores de intensidad de los píxeles en la imagen.

Finalmente, se presentan una serie de comparaciones para evaluar los resultados obtenidos a partir de la implementación de los algoritmos propuestos para la resolución del problema.

Abstract

From the early days of art and photography, the restoration of images has been done manually by professional artists through the usage of inpainting and retouching. Imitating their performance with semiautomatic digital techniques, image restoration is an active area of research. Inpainting as an image restoration application has found an important progress due to the success of the numerous approaches proposed to solve this problematic.

The proposed methodology is divided in three main stages: the first stage acts as a user input to select the target zone of the algorithm and allows generating the inpainting mask. The second stage calculates the inpainting priorities to set an order for restoration, so that the most textured-zones are processed first. The third stage synthesizes the image texture and copies the most suitable information to fill in the missing region according to image pixel intensity values.

Finally, comparisons to evaluate the results obtained through the algorithms proposed to solve the problematic are presented.

Agradecimientos

Primero quiero agradecer a mi familia, especialmente a mis padres, a mi hermana Ana y a mi hermano Rubén, por su apoyo incondicional, motivación y todo lo que he aprendido a su lado.

A mis amigos, con los que coincidí, y a los que están y se quedaron compartiendo esta travesía que llamamos vida. Me quedo siempre con algo de sus experiencias.

A mis asesores, el Dr. Edgardo Manuel Felipe Riverón y el Dr. Jesús Alberto Martínez Castro, por el asesoramiento y por cada uno de sus consejos.

Al Instituto Politécnico Nacional y al Centro de Investigación en Computación por las facilidades otorgadas durante el desarrollo del presente trabajo. Finalmente al Consejo Nacional de Ciencia y Tecnología (Conacyt) y a la SIP-IPN, a través de los proyectos BEIFI 20141381 y 20150616 por el apoyo económico para que este trabajo haya concluido satisfactoriamente.

Contenido

Resumen	IV
<i>Abstract</i>	V
Capítulo 1 Introducción.....	1
1.1 Planteamiento del problema.....	2
1.2 Justificación	2
1.3 Hipótesis	2
1.4 Objetivo general.....	3
1.5 Objetivos particulares	3
1.6 Consideraciones generales	3
1.7 Organización del documento	3
Capítulo 2 Estado del arte	5
2.1 Primeras técnicas de repintado.....	6
2.2 Repintado	7
2.3 Enfoques actuales	9
2.3.1 Bertalmio, Sapiro, y otros, 2000.....	9
2.3.2 Bertalmio, Bertozzi y Sapiro, 2001	10
2.3.3 Criminisi, Pérez y Toyama, 2004	11
2.3.4 Wu y Ruan, 2006.....	12
2.3.5 Guo, Ono y Sagayama, 2008	12
2.3.6 Xu y Sun, 2010.....	13
2.3.7 Dang, Larabi y Beghdadi, 2012.....	13
2.3.8 Le Meur, Ebdelli y Guillemot, 2013	14
2.3.9 Ghorai y Chanda, 2015.....	14
Capítulo 3 Marco teórico.....	16
3.1 Procesamiento digital de imágenes	17
3.1.1 Imagen digital	17
3.2 Morfología matemática.....	26
3.2.1 Elemento de estructura	26
3.2.2 Erosión.....	27
3.2.3 Dilatación	27

3.3	Color	27
3.3.1	Modelos de color	28
3.4	Restauración de imágenes.....	28
3.4.1	Un modelo del fenómeno de degradación de una imagen.....	28
3.4.2	Clasificación de las técnicas de restauración.....	29
Capítulo 4	Metodología.....	31
4.1	Metodología.....	32
4.1.1	Obtención de la máscara de repintado.....	33
4.1.2	Cálculo de prioridades para el orden de repintado	34
4.1.3	Búsqueda de la información para repintar.....	36
4.1.4	Actualización de la imagen.....	38
4.2	Iteraciones del algoritmo	39
4.3	Interfaz gráfica.....	40
4.3.1	Módulo de creación de la máscara de repintado.....	40
4.3.2	Módulo de restauración	41
Capítulo 5	Evaluación de los resultados	43
5.1	Pruebas con imágenes sintéticas	44
5.1.1	Imagen sintética_1	44
5.1.2	Imagen sintética_2.....	44
5.1.3	Imagen sintética_3.....	45
5.1.1	Imagen sintética_4.....	45
5.1.2	Imagen sintética_5.....	46
5.1.3	Imagen sintética_6.....	46
5.1.4	Resultados con imágenes sintéticas.....	47
5.1.5	El tamaño del parche en el resultado del algoritmo.....	48
5.1.6	El tamaño de la ventana de búsqueda.....	49
5.2	Pruebas con imágenes naturales	50
5.2.1	Base de datos <i>TUM</i>	50
5.2.2	Índice SSIM.....	60
5.2.3	Comparación de resultados.....	61
5.3	Resultados con la restauración de imágenes reales	62

5.3.1	Imagen con texto sobrepuesto	62
5.3.2	Imagen de Lincoln	63
5.3.3	Imagen endoscópica	64
5.3.4	Cebra.....	65
5.3.5	Lena	67
Capítulo 6	Conclusiones y trabajo futuro.....	69
6.1	Conclusiones	70
6.2	Recomendaciones	71
6.3	Trabajo futuro	71
Referencias	72

Lista de figuras

Figura 2.1 El rectángulo negro se puede percibir en ambas imágenes (Masnou y otros, 1998).	6
Figura 2.2 (a) Máscara de repintado, los pixeles en blanco indican las zonas objetivo (b) Imagen.	8
Figura 2.3 La región objetivo es Ω ; se elige primero el parche Ψ_p ubicado en el punto p para repintar, dado que se encuentra sobre la continuación de un borde.	8
Figura 2.4 La región objetivo es Ω , Ψ_p es el parche actual para repintar y $\Psi_{q'}$, $\Psi_{q''}$ son los parches candidatos.	9
Figura 2.5 El parche repintado toma la información de los pixeles del parche más similar.	9
Figura 2.6 (a) Imagen degradada (b) Imagen resultante.	10
Figura 2.7 (a) Imagen con texto superpuesto (b) Imagen resultante.	11
Figura 2.8 (a) Imagen con objetos superpuestos (b) Imagen resultante.	12
Figura 2.9 (a) Eliminación de una persona (b) Imagen resultante.	12
Figura 2.10 (a) Zona objetivo sobre la continuación de bordes (b) Imagen resultante.	13
Figura 2.11 (a) Imagen con avión (b) Imagen resultante.	13
Figura 2.12 (a) Zonas objetivo (b) Imagen resultante.	14
Figura 2.13 (a) Zona objetivo (b) Imagen resultante.	14
Figura 2.14 (a) Imagen con elefante (b) Imagen resultante.	15
Figura 3.1 Vecindad de 3×3 en el pixel (x, y) de la imagen en el dominio espacial.	19
Figura 3.2 Máscara de filtrado espacial de 3×3 pixeles.	22
Figura 3.3 Máscaras de Prewitt para calcular (a) La derivada en la dirección x . (b) La derivada en la dirección y .	22
Figura 3.4 Máscaras de Sobel para calcular. (a) La derivada en la dirección y . (b) La derivada en la dirección x .	23
Figura 3.5 (a) Imagen original. (b) Histograma de la imagen. (c) Resultado del umbralado con $T = 127$.	26
Figura 3.6 Longitudes de onda del espectro electromagnético visible.	27
Figura 3.7 Modelo de degradación de la imagen.	29
Figura 3.8 Diagrama del proceso de degradación y restauración de la imagen.	29
Figura 4.1 Diagrama de bloques general del algoritmo de repintado.	33
Figura 4.2 Terminología de las zonas principales en la imagen para el algoritmo de repintado.	33
Figura 4.3 Método de pelado de cebolla para el repintado (a) Región objetivo mostrada en blanco (b) Sentido en el que avanza el algoritmo (c) Se obtienen artefactos en el resultado final.	34
Figura 4.4 Notación de los elementos para el cálculo de prioridades.	36
Figura 4.5 Ejemplo de síntesis de textura. (a) Imagen de muestra (b) Imagen a sintetizar.	37
Figura 4.6 Ventana de búsqueda.	38
Figura 4.7 Actualización del parche de repintado.	39

Figura 4.8 Seudocódigo de la metodología.	40
Figura 4.9 Módulo de creación de la máscara de repintado.	41
Figura 4.10 Resultado del módulo de creación de la máscara de repintado (a) Imagen de repintado (b) Máscara de repintado.	41
Figura 4.11 Módulo de restauración.	42
Figura 5.1 Imagen sintética_1 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 9x9 píxeles.	44
Figura 5.2 Imagen sintética_2 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 9x9 píxeles.	45
Figura 5.3 Imagen sintética_3 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 9x9 píxeles.	45
Figura 5.4 Imagen sintética_4 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 5x5 píxeles.	46
Figura 5.5 Imagen sintética_5 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 5x5 píxeles.	46
Figura 5.6 Imagen sintética_6 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 5x5 píxeles.	47
Figura 5.7 Imagen sintética para la evaluación del tamaño de parche (a) Imagen original (b) Máscara de repintado (c) Imagen para repintar.	48
Figura 5.8 Error medio cuadrático al variar el tamaño de parche.	49
Figura 5.9 Variación del error cuadrático medio a medida que aumenta la ventana de búsqueda.	50
Figura 5.10 (a) Imagen Calle para repintar (b) Máscara de repintado.	51
Figura 5.11 Imagen Calle restaurada con parche de 3x3 píxeles.	51
Figura 5.12 Imagen Calle restaurada con parche de 5x5 píxeles.	52
Figura 5.13 Imagen Calle restaurada con parche de 9x9 píxeles.	52
Figura 5.14 (a) Imagen Edificio para repintar. (b) Máscara de repintado.	53
Figura 5.15 Imagen Edificio restaurada con parche de 3x3 píxeles.	53
Figura 5.16 Imagen Edificio restaurada con parche de 5x5 píxeles.	54
Figura 5.17 Imagen Edificio restaurada con parche de 9x9 píxeles.	54
Figura 5.18 (a) Imagen Estructura (b) Máscara de repintado.	55
Figura 5.19 Imagen Estructura restaurada con parche de 3x3 píxeles.	55
Figura 5.20 Imagen Estructura restaurada con parche de 5x5 píxeles.	56
Figura 5.21 Imagen Estructura restaurada con parche de 9x9 píxeles.	56
Figura 5.22 (a) Imagen Oficina (b) Máscara de repintado.	57
Figura 5.23 Imagen Oficina restaurada con parche de 3x3 píxeles.	57
Figura 5.24 Imagen Oficina restaurada con parche de 5x5 píxeles.	58
Figura 5.25 Imagen Oficina restaurada con parche de 5x5 píxeles.	58
Figura 5.26 (a) Imagen Tierra (b) Máscara de repintado.	59
Figura 5.27 Imagen Tierra restaurada con parche de 3x3 píxeles.	59
Figura 5.28 Imagen Tierra restaurada con parche de 5x5 píxeles.	60

Figura 5.29 Imagen Tierra restaurada con parche de 9x9 pixeles.....	60
Figura 5.30 Comparación de resultados con nuestro algoritmo empleando el índice SSIM.....	62
Figura 5.31 (a) Imagen con texto sobrepuesto (b) Máscara de repintado.	63
Figura 5.32 Imagen restaurada después de remover el texto con parches (a) 3x3 (b) 5x5 (c) 9x9 pixeles.....	63
Figura 5.33 (a) Imagen de Lincoln (b) Máscara de repintado.	64
Figura 5.34 Imagen de Lincoln restaurada con parches (a) 3x3 (b) 5x5 (c) 9x9 pixeles.	64
Figura 5.35 (a) Imagen endoscópica (b) Máscara de repintado.	65
Figura 5.36 Imagen endoscópica restaurada con parches (a) 3x3 (b) 5x5 (c) 9x9 pixeles.....	65
Figura 5.37 (a) Imagen de una cebra (b) Máscara de repintado.	66
Figura 5.38 Imagen de la cebra una vez restaurada con parches (a) 3x3 (b) 5x5 (c) 9x9 (d) 11x11 pixeles.....	66
Figura 5.39 (a) Imagen de Lena (b) Máscara de repintado.	67
Figura 5.40 Imagen de Lena restaurada con parche (a) 3x3 (b) 5x5 (c) 9x9.....	68

Lista de tablas

Tabla 5.1 Número de píxeles distintos entre la imagen original y la restaurada.....	47
Tabla.2 Porcentajes de acierto.....	47

Capítulo 1 Introducción

1.1 - Planteamiento del problema

1.2 - Justificación

1.3 - Hipótesis

1.4 - Objetivo general

1.5 - Objetivos particulares

1.6 - Consideraciones

1.7 - Organización del documento

Las degradaciones de todo tipo presentes en particular en fotografías antiguas o en imágenes debido a errores de transmisión, son procesos inevitables en muchos casos; por ello la necesidad de preservar los trabajos y restaurar aquellos que han sido deteriorados, es una tarea de extrema importancia en el procesamiento de imágenes digitales.

El presente trabajo describe la metodología desarrollada para la restauración de imágenes digitales mediante el uso de algoritmos de repintado (*inpainting*), los que por los resultados tan prometedores obtenidos en su aplicación en la restauración de imágenes, serán empleados como base.

1.1 Planteamiento del problema

El problema de la restauración de imágenes busca recuperar toda la información perdida de una imagen original que ha sido afectada por diversas formas de degradación (Guillemot & Le Meur, 2014). El origen y tipo de la degradación depende de la aplicación: texto sobrepuesto, ralladuras en una fotografía digital, imágenes tomadas de pinturas antiguas que se han cuarteado (Pei, Zeng y Chang, 2004), etc. Así mismo, en el problema de la restauración, la región a restaurar generalmente no es muy grande, por lo que el uso de una técnica de repintado basada en parches podría dar resultados satisfactorios.

1.2 Justificación

Dentro del análisis y del procesamiento digital de imágenes, un paso imprescindible previo a la segmentación de imágenes es el mejoramiento de la imagen, en el cual la restauración de imágenes ocupa un lugar preponderante.

En el caso de la restauración de imágenes de obras de arte deterioradas físicamente o por el color, el repintado está muy presente al tomar en cuenta el deterioro que sufren los cuadros o registros digitales de obras de arte, por lo que su aplicación para obtener una imagen aproximada a la original es muy importante como referencia.

De esta forma, el repintado en sus aplicaciones en la restauración de imágenes es un área de investigación actual, en la cual los autores buscan mejorar los resultados de la restauración, así como reducir los tiempos de procesamiento.

1.3 Hipótesis

Mediante el análisis de imágenes y algoritmos de repintado, se plantea que es posible efectuar la restauración de imágenes digitales en colores provenientes de diversas aplicaciones con resultados factibles de ser utilizados en la práctica.

1.4 Objetivo general

Crear algoritmos para la restauración de imágenes digitales en colores mediante técnicas de repintado de imágenes.

1.5 Objetivos particulares

- Generar la máscara de repintado a partir de la selección hecha por el usuario.
- Rellenar las zonas marcadas por la máscara mediante los algoritmos de restauración propuestos.
- Diseñar y desarrollar una interfaz gráfica que permita manipular fácil y confiablemente la metodología propuesta.
- Evaluar los resultados.

1.6 Consideraciones generales

Para el desarrollo de este trabajo se han tomado en cuenta las siguientes consideraciones generales:

- Las imágenes sobre las que se trabajará corresponden a paisajes, imágenes de escenas reales pero no con demasiado detalle como en el caso de los rostros humanos.
- No existe un conocimiento previo acerca de las características de la imagen, por lo cual la restauración se efectúa solo con la información existente en ella.
- El procesamiento no dependerá de la forma de la zona objetivo donde se ejecutará el repintado.

La razón de que en este trabajo se hayan establecido estas consideraciones, no ha sido otra que el interés de basar la validez de los resultados sólo en las características relevantes encontradas en la información existente en la imagen.

De llegar a resultados aceptables, con ello se puede considerar, indiscutiblemente, una mayor validez y generalidad de los mismos con la metodología desarrollada.

1.7 Organización del documento

La presente tesis consta de 6 capítulos.

En este *Capítulo 1* se presentó de manera general el planteamiento del problema, la justificación del trabajo, el objetivo general y los particulares, la hipótesis general y las consideraciones generales fundamentales que justifican el alcance del mismo. Además, la organización del documento.

El *Capítulo 2* está orientado hacia el estado del arte, creado a partir de las publicaciones halladas en revistas de impacto y en congresos internacionales reconocidos sobre la temática. El capítulo se ha dedicado al análisis de los métodos de repintado en la restauración de imágenes digitales.

El *Capítulo 3* describe los conceptos y definiciones básicos que se han utilizado para la realización del trabajo, mismos que se han incorporado en cuatro secciones: la primera sobre el procesamiento digital de imágenes, la segunda sobre la morfología matemática, la tercera sobre conceptos relacionados con el color y la cuarta sobre la restauración de imágenes digitales.

En el *Capítulo 4* se expone ampliamente el desarrollo del trabajo de investigación. Incluye una breve descripción de la interfaz gráfica para efectuar la restauración de imágenes mediante el repintado, así como una descripción detallada de los algoritmos empleados para la resolución del problema planteado en esta tesis.

En el *Capítulo 5* se evalúan los parámetros de análisis y se comparan los resultados obtenidos a partir de la restauración mediante ciertas métricas encontradas en publicaciones tomadas en cuenta en el estado del arte.

Finalmente, el *Capítulo 6* trata sobre las conclusiones, las recomendaciones y el trabajo futuro que a partir de lo logrado en el presente trabajo, permitirá hacer modificaciones y ampliaciones que mejorarán sin dudas los resultados alcanzados hasta el momento.

Capítulo 2 Estado del arte

2.1 - Primeras técnicas de repintado

2.2 - Repintado

2.3 - Enfoques actuales

En este capítulo se presentan algunos trabajos que sirvieron como punto de partida para la presente investigación y que permitieron llevar el trabajo hacia las tendencias actuales en el área de repintado. Cabe mencionar que los enfoques utilizados para resolver el problema de la restauración de imágenes mediante el repintado son muy variados, por lo que a su vez se presenta un panorama de las técnicas que se utilizan en la actualidad.

2.1 Primeras técnicas de repintado

El repintado de imágenes (del inglés *inpainting*) ha evolucionado como una importante aplicación para técnicas automáticas o semiautomáticas de restauración de imágenes. La necesidad de retocar imágenes de forma no obstructiva se extendió naturalmente de las pinturas antiguas a la fotografía y el cine. El propósito continuó siendo el mismo: revertir el deterioro así como agregar o eliminar elementos de la imagen (Bertalmio y otros, 2011).

En el dominio digital el problema de repintado apareció primero bajo el nombre de ocultamiento del error en telecomunicaciones (del inglés *error concealment*), donde la necesidad era rellenar bloques de datos o de imagen que se habían perdidos durante la transmisión de datos. Uno de los primeros trabajos dirigidos al repintado automático digital fue llamado desoclusión de la imagen (del inglés *image disclusion*), ya que trataba los huecos en la imagen como un objeto ocluido (con uno o varios objetos superpuestos) que tenía que ser eliminado rellenándolo; con esto, la imagen de abajo sería el resultado de la restauración (Masnou y otros, 1998).

En el proceso de la visión humana la desoclusión es muy común, pues los objetos en raras ocasiones son totalmente visibles al estar ocultos por otros objetos. Pero nuestra percepción es capaz bajo ciertas circunstancias de reconstruir el objeto entero interpolando la parte faltante. En el ejemplo ilustrado en la Figura 2.1, la visión humana percibe generalmente el mismo rectángulo en ambas imágenes, a pesar de que el rectángulo nunca está totalmente visible.

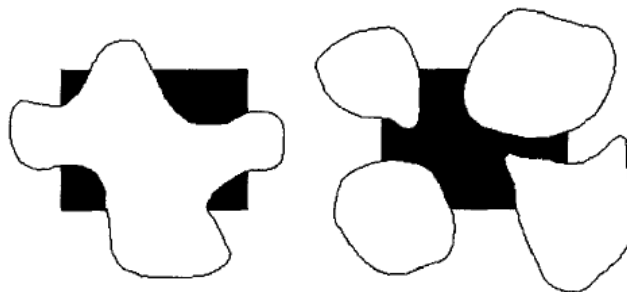


Figura 2.1 El rectángulo negro se puede percibir en ambas imágenes (Masnou y otros, 1998).

2.2 Repintado

El término repintado viene de la restauración de obras de arte, donde también es llamado retocado. En la restauración de obras de arte y de fotografías, el objetivo del repintado es revertir el deterioro de las obras (causado, por ejemplo, por rayones, grietas, manchas de moho, etc.) y consiste en modificar una imagen con arreglos y modificaciones indetectables al volverla a su estado original. Otra de sus aplicaciones consiste en la eliminación de objetos no deseados que están presentes en la imagen.

El término repintado en imágenes digitales fue introducido por Bertalmio y otros, 2000, y desde su aparición se ha convertido actualmente en un área activa de investigación (Bertalmio, Bertozzi & Sapiro, 2001).

El repintado ha sido estudiado en el campo del procesamiento digital de imágenes y en la visión por computadora, en la restauración (eliminación de rayaduras), en la edición de imágenes (eliminación de objetos o texto) y en la codificación y transmisión (recuperación de información faltante).

Para resolver el problema se han propuesto métodos asumiendo que los píxeles en las partes conocidas y desconocidas de la imagen comparten algunas propiedades estadísticas o estructuras geométricas (Guillemot & Le Meur, 2014).

Los enfoques para el repintado en imágenes digitales pueden dividirse en tres categorías: (a) Enfoque basado en ecuaciones diferenciales parciales (PDE del inglés *partial differential equation*) (b) Enfoque basado en ejemplares y (c) Enfoque basado en representación dispersa (del inglés *sparse representation*).

La primera categoría de los métodos, conocido como repintado basado en difusión, introduce prioridades por medio de modelos paramétricos o de ecuaciones diferenciales parciales para propagar o difuminar estructuras locales del exterior al interior del hueco. Estos métodos son naturalmente adecuados para completar líneas rectas, curvas y para rellenar regiones pequeñas. Sin embargo, en presencia de texturas en áreas grandes tienden a presentar un efecto borroso.

La segunda categoría se basa en el uso de la estadística de la imagen y prioridades de autosimilitud. La textura a ser sintetizada se aprende de regiones similares en una muestra de textura o de la parte conocida de la imagen. El aprendizaje se hace muestreando y copiando parches (llamados *exemplar* en inglés) tomados de la parte conocida de la imagen. Estos métodos son conocidos como técnicas basadas en ejemplares.

La última categoría surge con el uso de representaciones dispersas en la cual la imagen o el parche se consideran como dispersos sobre cierta base. De esta forma se asume que las partes conocidas y desconocidas comparten la misma representación dispersa. Los métodos basados

en ejemplares y en representaciones dispersas, son más adecuados para rellenar áreas grandes con textura. De esta forma, han emergido soluciones que combinan métodos dedicados a componentes estructurales (geométricos) y de textura.

Para este tipo de métodos híbridos se ha establecido un proceso general mediante el cual se efectúa el repintado, aunque cabe mencionar que algunos autores hacen más énfasis en una etapa u otra del proceso de acuerdo al enfoque elegido (Guillemot y otros, 2014) (Zhang y otros, 2012).

Selección de la zona a restaurar. El parámetro que indica el usuario es la máscara de repintado, como se muestra en la Figura 2.2, la cual marca manualmente la imagen objetivo (Bertalmio y otros, 2000).

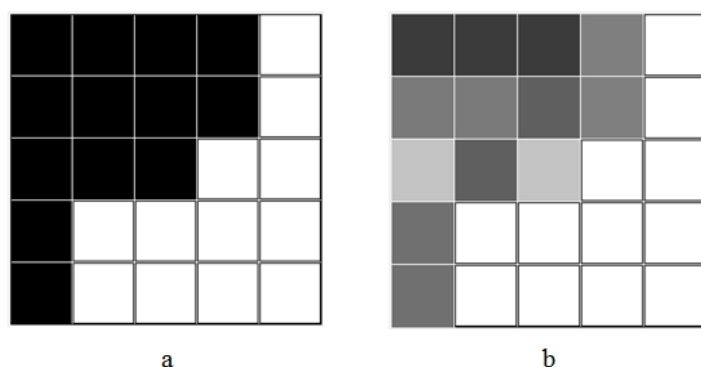


Figura 2.2 (a) Máscara de repintado, los pixeles en blanco indican las zonas objetivo (b) Imagen.

Asignación de prioridad para el orden del repintado. Se ha observado que las regiones faltantes en una imagen están compuestas por texturas y estructuras, por lo que es importante empezar recuperando estructuras (Criminisi y otros, 2004), lo cual ha dado lugar a definir un orden en el procesamiento de los parches para el repintado como se muestra en la Figura 2.3. La región objetivo Ω es la que se desea repintar.

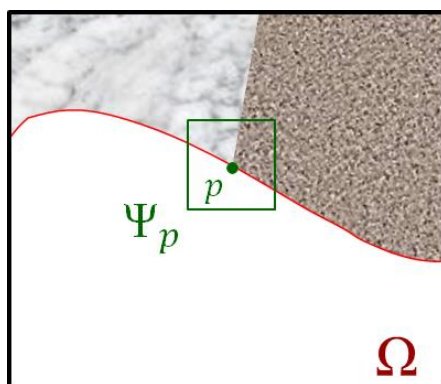


Figura 2.3 La región objetivo es Ω ; se elige primero el parche Ψ_p ubicado en el punto p para repintar, dado que se encuentra sobre la continuación de un borde.

Búsqueda del parche más similar. Una vez elegido el parche a repintar, es importante buscar el parche que de acuerdo a la información que está alrededor del pixel central sea más similar al parche objetivo, como se puede notar en la Figura 2.4.

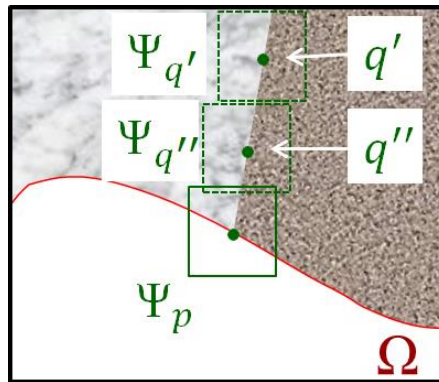


Figura 2.4 La región objetivo es Ω , Ψ_p es el parche actual para repintar y $\Psi_{q'}$, $\Psi_{q''}$ son los parches candidatos.

Copiado del parche más similar al parche objetivo. En la Figura 2.5 se muestra que una vez encontrado el parche más similar, se copian sus píxeles a los píxeles correspondientes desconocidos en el parche objetivo, de forma que sea rellenado o repintado con la nueva información.

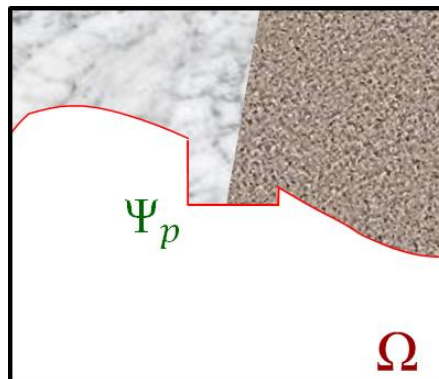


Figura 2.5 El parche repintado toma la información de los píxeles del parche más similar.

2.3 Enfoques actuales

Desde la aparición del término repintado han sido diversos los enfoques empleados para resolver la problemática. Se presenta a continuación un compendio de los trabajos más relevantes consultados para el desarrollo del presente trabajo de tesis.

2.3.1 Bertalmio, Sapiro, y otros, 2000

En el trabajo de Bertalmio y otros, 2000, se introduce el término de repintado automático en imágenes digitales. Se hace énfasis en la limitante de la intervención del usuario para especificar qué textura poner y dónde. De esta forma su algoritmo no hace suposiciones acerca

de la topología de la región para efectuar el repintado ni sobre la simplicidad de la imagen. Se basa en el uso del gradiente en la imagen para detectar las zonas de mayor cambio, con el fin de completar la información de los píxeles faltantes.

La mayor limitante es la incapacidad para reproducir áreas grandes con textura, de forma que si bien los resultados son imperceptibles para zonas pequeñas o delgadas, en presencia de regiones grandes con textura se obtiene el molesto efecto borroso, como se observa en la Figura 2.6.



Figura 2.6 (a) Imagen degradada (b) Imagen resultante.

2.3.2 Bertalmio, Bertozzi y Sapiro, 2001

En Bertalmio y otros, 2001, se muestra la importancia en general de la dinámica de fluidos computacional (CFD, del inglés *Computational fluid dynamics*), y en particular, del uso de la ecuación de Navier-Stokes para problemas de visión. Se propone una equivalencia entre la ecuación de flujo-vorticidad al “flujo” de píxeles en una imagen, además de un algoritmo para el cálculo numérico que mejora la velocidad en la obtención de los valores de píxeles para el repintado.

Se propone también la aplicación del algoritmo para obtener una imagen de superresolución, esto es, una imagen que se requiere tener en mayor tamaño mejorando los resultados que se obtienen mediante interpolación (cúbica o bicúbica).

De esta forma, se mejora la velocidad del algoritmo de repintado; sin embargo, las zonas de repintado deben ser pequeñas (Figura 2.7), pues la calidad del resultado disminuye a medida que dichas zonas se incrementan.



Figura 2.7 (a) Imagen con texto sobrepuesto (b) Imagen resultante.

2.3.3 Criminisi, Pérez y Toyama, 2004

Criminisi y sus colegas (Criminisi y otros, 2004) destacan que las técnicas de repintado para rellenar huecos en imágenes propagando estructuras lineales o isofotas (llamadas *isophotes* en inglés en la literatura del repintado) hacia la región objetivo usando difusión, resultan en la borrosidad de las regiones grandes. De esta forma introducen el término de *prioridad* en el orden del repintado, con el que dan prioridad a parches con estructuras lineales y a su vez hacen uso de la síntesis de texturas para el algoritmo (Efros y otros, 1999).

Para el cómputo de la prioridad del parche, se buscan parches que: (a) estén en la continuación de bordes fuertes y (b) estén rodeados de píxeles de alta confianza. De esta forma, dado un parche Ψ_p centrado en el punto p , se define su prioridad $P(p)$ como el producto de dos términos:

$$P(p) = C(p)D(p)$$

Donde $C(p)$ es el término de *confianza* y $D(p)$ el término de *datos*. La confianza está representada por el promedio de los valores de los píxeles existentes en el parche, el cual puede ser visto como una medida de la cantidad de información confiable alrededor del píxel p . El término de datos es una función de la fuerza de las isofotas que entran hacia la zona de repintado. Este término incrementa la prioridad de un parche cuando una isofota fluye hacia la zona de repintado.

Los resultados superan a los demás algoritmos que habían sido publicados hasta el momento en cuanto a regiones grandes de repintado con texturas menos sencillas, de tal forma que actualmente trabajos sobre *inpainting* parten del mismo principio. (Figura 2.8.)

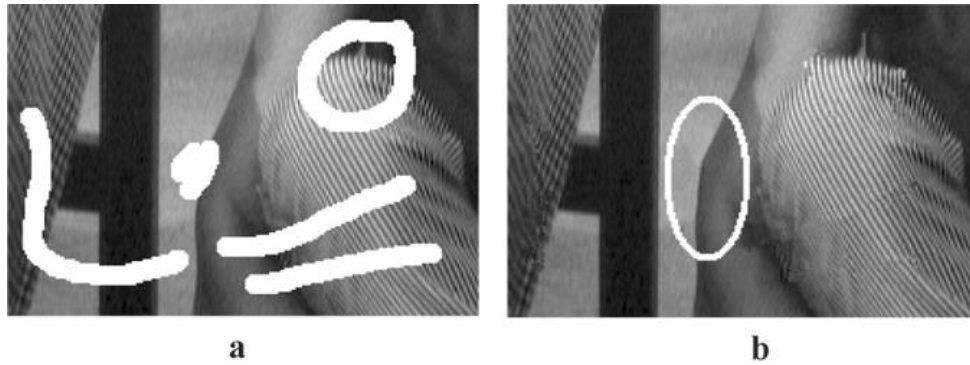


Figura 2.8 (a) Imagen con objetos sobrepuestos (b) Imagen resultante.

2.3.4 Wu y Ruan, 2006

A partir de la idea planteada por Criminisi, Wu y otros, 2006, proponen un nuevo modelo de repintado basado en ejemplares y en una combinación de isofotas (*CIEI* del inglés *Cross isophotes exemplar-based inpainting*), en el cual se mejora la prioridad en la selección del parche para repintar, tomando en cuenta la extensión del borde detectado de hacia dónde fluyen las isofotas. Este modelo preserva mejor las estructuras lineales en el repintado.

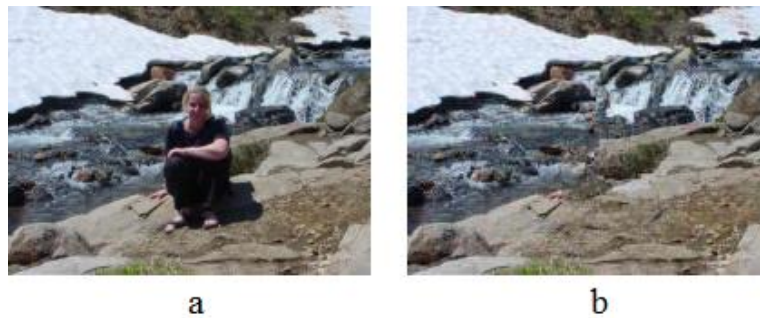


Figura 2.9 (a) Eliminación de una persona (b) Imagen resultante.

2.3.5 Guo, Ono y Sagayama, 2008

En el trabajo de Guo y otros, 2008, se usa la operación morfológica de erosión para establecer el orden para rellenar las regiones faltantes en la imagen, así como se usan características de estructura presentes en la imagen, teniendo mejores resultados en zonas grandes con textura.

Además, con base en las características de estructura y textura restauran estructuras ordinarias, tales como contornos lineales o curvas con pequeña curvatura, sin agregar el efecto borroso a la zona de restauración de la imagen resultante (Figura 2.10.)

Finalmente, gracias a la implementación basada en operaciones morfológicas, es posible procesar varios píxeles a medida que se erosiona la zona de repintado, lo cual acelera la velocidad del algoritmo.

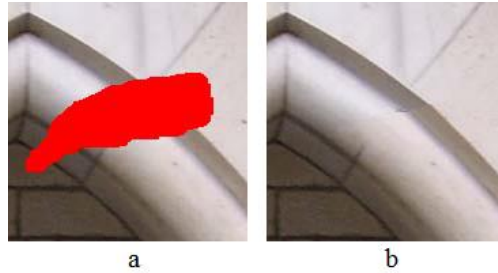


Figura 2.10 (a) Zona objetivo sobre la continuación de bordes (b) Imagen resultante.

2.3.6 Xu y Sun, 2010

Xu y sus colegas (Xu y otros, 2010) introducen dos conceptos: dispersión (del inglés *sparsity*) a nivel del parche para modelar la prioridad del parche y su representación.

Primero, se diseña la dispersión del parche para medir la confianza de un parche ubicado en la estructura de la imagen (borde o esquina) por la dispersión de similitudes no cero (del inglés *nonzero similarities*) a los parches vecinos. Al parche con mayor estructura dispersa le es asignada una prioridad mayor para el repintado posterior.

Segundo, se asume que el parche a ser rellenado puede representarse mediante la combinación lineal dispersa de parches candidatos bajo la restricción de una representación dispersa.

Sus resultados muestran que el algoritmo puede inferir mejor acerca de las estructuras y texturas de la región faltante, además de que produce resultados nítidos de repintado consistentes con las texturas que rodean a la zona objetivo (Figura 2.11.)



Figura 2.11 (a) Imagen con avión (b) Imagen resultante.

2.3.7 Dang, Larabi y Beghdadi, 2012

En Dang, Larabi y Beghdadi, 2012, se introduce un algoritmo que completa las imágenes a restaurar con el enfoque de representación de imágenes en multirresolución. Un parche de esta forma asegura que la textura y la estructura geométricas serán restauradas con parches de tamaño fijo en lugar de emplear varios parches de diferente tamaño o de tamaño dinámico, lo cual aumenta el tiempo de ejecución del algoritmo.

La prioridad es un problema importante a tomar en cuenta en los enfoques basados en ejemplares; por ello se propone una prioridad basada en ventanas que clasifica los parches de una forma más adecuada mejorando los resultados. Así mismo, la selección de un parche adaptativo ofrece mejores resultados cuando hay más de dos parches igualmente similares al parche origen (Figura 2.12).

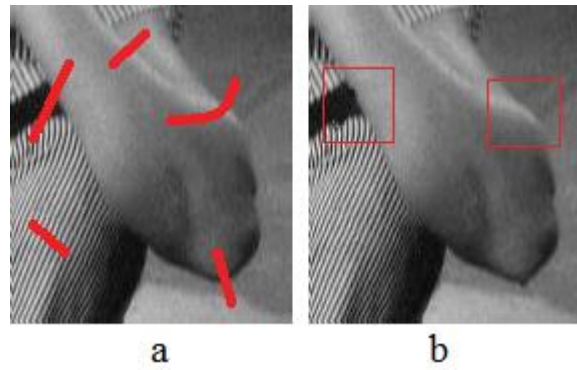


Figura 2.12 (a) Zonas objetivo (b) Imagen resultante.

2.3.8 Le Meur, Ebdelli y Guillemot, 2013

Continuando con el uso de la superresolución, en Le Meur y otros, 2013, la imagen es muestreada a una resolución menor (submuestreada) y se efectúan varios repintados. Las imágenes repintadas de baja resolución son combinadas minimizando globalmente un término de energía que modela la imagen. Una vez que se completa la combinación, se aplica un método de superresolución para recuperar los detalles a la resolución original.

Los resultados presentan mejoras a costa de un mayor tiempo de ejecución del algoritmo; sin embargo, los autores mencionan que si se plantea una implementación en paralelo del algoritmo, podrían encontrarse mejorías en los tiempos de ejecución. (Figura 2.13.)



Figura 2.13 (a) Zona objetivo (b) Imagen resultante.

2.3.9 Ghorai y Chanda, 2015

Finalmente, los autores Ghorai y Chanda, 2015, toman el enfoque del análisis de la imagen a través del procesamiento de lenguaje natural. Una vez que se elige el parche para repintar, la

búsqueda del parche más similar se estima usando análisis semántico latente probabilístico (*pLSA* del inglés *probabilistic latent semantic analysis*).

Para inferir acerca de los pixeles faltantes en el parche objetivo emplean la medida de similitud de contexto e histograma basada en *pLSA* entre el parche objetivo y los parches candidatos (más similares).

De esta forma, los resultados en imágenes reales sobre las que se efectúa el trabajo de *inpainting* son prácticamente imperceptibles. Sin embargo, el tiempo de procesamiento se incrementa por las búsquedas que se efectúan dentro de la imagen para encontrar los parches más similares, uno de los resultados se muestra en la Figura 2.14:



Figura 2.14 (a) Imagen con elefante (b) Imagen resultante.

Capítulo 3 Marco teórico

3.1 - Procesamiento digital de imágenes

3.2 - Morfología matemática

3.3 - Color

3.4 - Restauración de imágenes

El presente capítulo expone brevemente los conceptos y definiciones empleados en el desarrollo de este trabajo. Se ha dividido en cuatro secciones, la primera enfocada en los aspectos generales del procesamiento digital de imágenes, la segunda sección dedicada a la morfología matemática, la tercera sección tratará conceptos relativos al color en las imágenes digitales y finalmente, la cuarta sección versa sobre la restauración de imágenes digitales.

3.1 Procesamiento digital de imágenes

El campo del procesamiento digital de imágenes se refiere al procesamiento de imágenes mediante el uso de una computadora digital. Una imagen digital está compuesta por un número finito de elementos, cada uno con una ubicación y un valor particulares.

3.1.1 Imagen digital

Se define una imagen digital como una función de dos dimensiones $f(x, y)$ donde x y y son coordenadas espaciales finitas y discretas; la amplitud de f en cualquier par de coordenadas (x, y) es llamada intensidad o nivel de gris de la imagen en ese punto. Se puede considerar como una retícula de puntos, elementos de imagen o píxeles (del inglés *picture elements*), cada uno de los cuales tiene asignado un valor de tono o nivel de gris.

3.1.1.1 Imagen de niveles de gris

Formalmente, una imagen f es un mapeo de un conjunto D_f de $Z \times Z$ llamado dominio de definición de f hacia un conjunto limitado (también llamado cadena finita) de enteros no negativos Z :

$$f : D_f \subset Z \times Z \rightarrow \{0, 1, \dots, t_{max}\}$$

Donde t_{max} es el valor máximo del dato empleado para almacenar la imagen (por ejemplo, $2^n - 1$ para valores de pixel codificados con n bits). De esta forma, si se emplean 8 bits para codificar los valores de pixel en una imagen, se tiene $2^8 - 1 = 255$ como t_{max} .

Las imágenes de niveles de gris poseen la intensidad como único atributo, el cual se percibe como tonalidades que van del negro al blanco dentro de un rango denominado escala de grises. En general, para efectos prácticos, se suele considerar el valor 0 para el negro y t_{max} para el blanco.

Un caso particular de imagen de niveles de gris se presenta cuando la imagen sólo puede tener valores normalizados de 0 para el negro y 1 para el blanco. Este tipo de imágenes se denominan binarias.

3.1.1.2 Imagen multicanal

Una imagen multicanal consiste de un vector de imágenes de niveles de gris donde un vector de valores escalares es asociado a cada pixel de la imagen multicanal. Denotamos a una imagen multicanal con m canales por f_m y los valores de cada pixel p del dominio de f_m definen un vector de m dimensiones de la siguiente forma:

$$f_m(p) = (f_1(p), f_2(p), \dots, f_m(p))$$

Debido a esta representación, usualmente cada canal f_i de una imagen multicanal es procesado como una imagen de niveles de gris individual, esto es, independientemente de los otros canales (Soille, 2003).

Existen varios tipos de imágenes multicanal dependiendo del tipo de información que se colecte para cada pixel de la imagen. Por ejemplo, las imágenes en colores representadas en el modelo de color RGB, son imágenes multicanal que contienen tres canales, uno por cada color primario (del inglés *Red-Green-Blue* - Rojo-Verde-Azul).

3.1.1.3 Histograma

Para una imagen digital con niveles de intensidad en el rango $[0, t_{max} - 1]$, se denomina histograma a una función discreta $h(r_k) = n_k$, donde r_k es el k -ésimo valor de intensidad y n_k es el número de pixeles con intensidad r_k en la imagen.

Es una práctica común normalizar un histograma dividiendo cada uno de sus componentes por el número total de pixeles en la imagen denotado por el producto MN , donde M y N son las dimensiones en pixeles de las filas y columnas de la imagen. Así, el histograma normalizado está dado por $p(r_k) = n_k/MN$, para $k = 0, 1, 2, \dots, t_{max} - 1$. Podría decirse que $p(r_k)$ es un estimado de la probabilidad de ocurrencia del nivel de intensidad r_k en una imagen. La suma de los componentes de un histograma normalizado es igual a uno.

El histograma se emplea como base para numerosas técnicas de procesamiento de imágenes basadas en la estadística.

3.1.1.4 Vecindad de pixeles

Un pixel p en las coordenadas (x, y) tiene cuatro vecinos horizontales y verticales cuyas coordenadas están dadas por:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)$$

Este conjunto de pixeles se denomina vecinos 4-conectados de p . Si algún vecino queda ubicado fuera de la imagen digital, se suele considerar asignar un valor de 0 a la intensidad que correspondería al pixel (Gonzalez y otros, 2008).

Los vecinos en diagonal de p tienen las coordenadas:

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

Cuando se toman en cuenta tanto los vecinos en diagonal como los horizontales y verticales, se denominan vecinos 8-conectados de p .

Existen aplicaciones para las cuales se requiere considerar vecindades (conjuntos de pixeles cercanos a p) de diferentes tamaños y características, en cuyo caso no siempre serán utilizadas vecindades 4 u 8-conectadas.

3.1.1.5 Procesamiento en el dominio espacial

El término dominio espacial hace referencia al plano de la imagen digital, así como los métodos de procesamiento de imágenes que involucran la manipulación directa de los pixeles. Este tipo de procesamiento se representa en forma general de la siguiente forma:

$$g(x, y) = T[f(x, y)] \quad (3.1)$$

Donde $f(x, y)$ es la imagen de entrada, $g(x, y)$ la imagen resultante del procesamiento y T un operador en f definido sobre una vecindad del pixel (x, y) . Típicamente, la vecindad es rectangular con centro en (x, y) y de dimensiones mucho menores que el tamaño total de la imagen, como se muestra en la Figura 3.1.

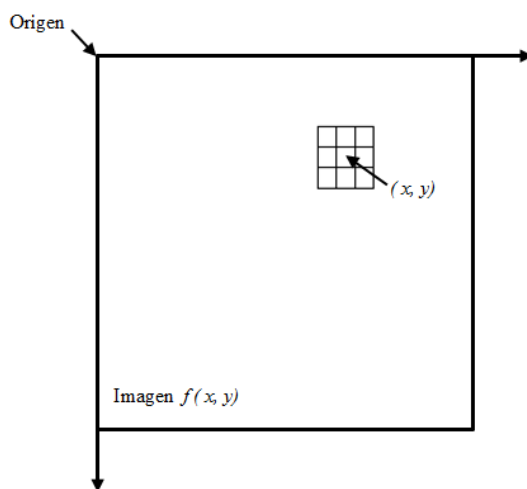


Figura 3.1 Vecindad de 3x3 en el pixel (x, y) de la imagen en el dominio espacial.

El origen de la vecindad es movido pixel a pixel y el operador T es aplicado a los pixeles en la vecindad para generar la salida en esa ubicación. Se llama a este procesamiento filtrado espacial, en el cual la vecindad (llamada *máscara espacial* o *ventana espacial*) junto con la operación predefinida, se denomina *filtro espacial*. El tipo de operación efectuada en la vecindad determina la naturaleza del proceso de filtrado.

Para el filtrado en el dominio espacial hay dos conceptos fundamentales conocidos como *correlación* y *convolución*. La *correlación* es el proceso de mover una máscara de filtrado sobre la imagen y calcular la suma de los productos en cada ubicación, del valor en cada posición de la máscara con el valor del pixel correspondiente de la imagen. El mecanismo de convolución es el mismo, excepto que el filtro es rotado primero 180° ; de esta forma para realizar una convolución, únicamente se rota la máscara y se efectúan las mismas operaciones que en la correlación.

La correlación de un filtro $w(x, y)$ de tamaño $m \times n$ con una imagen $f(x, y)$, denotada como $w(x, y) \star f(x, y)$, está dada por la ecuación (3.2) con $a = (m - 1)/2$ y $b = (n - 1)/2$, valores en los que por conveniencia de la notación se asume que m y n son enteros positivos impares. Esta ecuación es evaluada para todos los valores de las variables de desplazamiento x y y , de tal forma que todos los elementos de w visiten cada pixel en f .

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (3.2)$$

La convolución de $w(x, y)$ y $f(x, y)$, denotada como $w(x, y) * f(x, y)$, está dada por la ecuación (3.3).

$$w(x, y) * f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t) \quad (3.3)$$

El uso de una correlación o una convolución para efectuar el filtrado espacial es una cuestión de preferencia, ya que cualquiera de las dos puede hacer uso de la otra con una simple rotación del filtro. En la literatura los términos filtro de convolución o máscara de convolución se emplean para denominar al filtro espacial y no necesariamente implica que el filtro será utilizado para una convolución real; sin embargo, la convolución se emplea genéricamente para denotar cualquiera de las dos operaciones (Gonzalez y otros, 2008).

El mejoramiento de imágenes comprende un conjunto de procesos de manipulación de una imagen por medio del cual se obtiene una nueva imagen mejor que la original para una aplicación específica.

3.1.1.6 Segmentación de imágenes

La segmentación de imágenes permite extraer atributos de las mismas, con el fin de efectuar el análisis posterior de su contenido. La segmentación subdivide una imagen en los objetos o subregiones que la constituyen. El nivel de detalle de cada subdivisión depende del problema a resolver. Esto es, la segmentación debe detenerse cuando los objetos o regiones de interés en una aplicación particular han sido detectados.

Para efectos prácticos, los algoritmos de segmentación de imágenes empleados están basados en dos propiedades básicas relacionadas con los valores de intensidad de los píxeles que la constituyen: la discontinuidad y la similitud. En la primera categoría, con base en cambios abruptos de intensidades entre píxeles vecinos, se separan los objetos requeridos. Respecto a la similitud, se busca separar una imagen en regiones que son similares de acuerdo con una serie de criterios previamente establecidos.

3.1.1.7 Detección de bordes

La detección de bordes es el enfoque empleado más frecuentemente para segmentar imágenes, basado en cambios abruptos (localmente) de intensidad. Este tipo de cambios pueden ser detectados haciendo uso de la derivada, la cual para una función discreta se define en términos de diferencias.

Para aproximar la primera derivada, se requiere que se cumplan ciertas condiciones: que la derivada sea cero en zonas con intensidad constante, que no sea cero en un escalón o rampa de intensidades, y finalmente, que no sea cero en los puntos a lo largo de una rampa de intensidades.

Así mismo, para aproximar mediante la segunda derivada, se requiere que: sea cero en áreas de intensidad constante, no sea cero en el inicio y final de un escalón o rampa de intensidades y debe ser cero a lo largo de una rampa de intensidades.

Después de aplicar la primera derivada a una imagen se obtienen bordes gruesos y al aplicar la segunda derivada se obtienen bordes más delgados con realce de bordes finos.

La herramienta de elección para encontrar la fuerza del borde y dirección en la posición (x, y) de una imagen f , es el gradiente (∇f) definido en espacios continuos por la ecuación (3.4):

$$\nabla f \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (3.4)$$

De esta forma, obtener el gradiente de una imagen requiere el cálculo de las derivadas parciales $\frac{\partial f}{\partial x}$ y $\frac{\partial f}{\partial y}$ para cada pixel en la imagen. Debido a que se procesan valores discretos de intensidad, se requiere hallar una aproximación de cada una de las derivadas parciales.

Las derivadas pueden implementarse mediante el filtrado de la imagen con máscaras (cuyo menor tamaño sea 3 x 3) que tomen en cuenta la naturaleza de los valores en los lados opuestos del punto central y por tanto tengan mayor información con respecto a la dirección del borde; un ejemplo de dicha máscara se muestra en la Figura 3.2.

z1	z2	z3
z4	z5	z6
z7	z8	z9

Figura 3.2 Máscara de filtrado espacial de 3x3 pixeles.

La aproximación digital más simple de las derivadas parciales empleando máscaras, está dada por la convolución de la imagen con las dos máscaras que se muestran en la Figura 3.3 (tomar en cuenta la convención de la figura 3.1).

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Figura 3.3 Máscaras de Prewitt para calcular (a) La derivada en la dirección x. (b) La derivada en la dirección y.

De ellas se desprenden las ecuaciones(3.5) y (3.6) siguientes, respectivamente:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \quad (3.5)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \quad (3.6)$$

Una ligera variación de las dos ecuaciones anteriores es la ponderación de los coeficientes de las filas y columnas centrales de las máscaras con un peso de 2, como se muestran en la Figura 3.4.

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Figura 3.4 Máscaras de Sobel para calcular. (a) La derivada en la dirección y. (b) La derivada en la dirección x.

De ellas se desprenden las ecuaciones (3.7) y (3.8) siguientes, respectivamente:

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (3.7)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (3.8)$$

La suma de los coeficientes de las máscaras de Prewitt y Sobel da cero, por lo que en áreas de intensidad constante, como se espera del operador de derivada, el resultado es cero.

3.1.1.8 Umbralado

La técnica de umbralado consiste en separar imágenes directamente en regiones tomando en cuenta sólo los valores de intensidad, así como las propiedades de esos valores como un conjunto que forman bordes, componentes conexas, etc. De esta forma se aprovecha el hecho de que una imagen esté compuesta por objetos y fondo con valores de intensidad distintos (claros y oscuros) para seleccionar un umbral T que permita separarlos. Los pixeles que tengan un valor de intensidad menor o igual que T , pertenecen a una región distinta de aquella cuyos valores de intensidad sean mayores que T ; finalmente se obtiene una imagen binaria $g(x, y)$ definida por la ecuación (3.9).

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases} \quad (3.9)$$

Cuando T se aplica a toda una imagen, el proceso resultante de la ecuación (3.9) se denomina umbralado global.

El umbralado puede considerarse como un problema de decisión estadístico mediante el cual se requiere minimizar el error promedio al asignar píxeles a dos o más grupos o clases. De esta forma, el método de umbralado de Otsu (Otsu, 1979) busca maximizar la varianza entre clases y minimizar la varianza entre clases mediante una búsqueda exhaustiva. Los valores de intensidad de los píxeles en la misma clase deben ser semejantes, asimismo, deben ser distintos a los valores de los píxeles de las demás clases.

Parte del cálculo del histograma en el cual dada una imagen digital que contiene L valores de intensidad $\{0, 1, 2, \dots, L - 1\}$ y un total de $M \times N$ píxeles (M y N son el número de filas y columnas respectivamente). La probabilidad P_i de ocurrencia del valor de intensidad i está dada por la ecuación (3.10), donde n_i representa el número de píxeles que tienen el valor de intensidad i .

$$P_i = \frac{n_i}{MN} \quad (3.10)$$

Si seleccionamos un umbral T se obtiene una imagen con dos niveles, por lo cual los píxeles de la imagen son divididos en dos clases C_1 y C_2 , donde C_1 contiene los píxeles cuyo valor de intensidad está en el rango $[0, T]$ y C_2 contiene los píxeles con valores en el rango $[T+1, L-1]$. Mediante el empleo del umbral T , la probabilidad de que un píxel sea asignado a una clase está dada por la suma acumulativa de las probabilidades de ocurrencia de los valores de intensidad de cada clase, como se muestra en la ecuación (3.11).

$$P_1(T) = \sum_{i=0}^T P_i \quad (3.11)$$

Visto de otra forma, $P_1(T)$ es la probabilidad de que ocurra la clase C_1 . De forma similar, la probabilidad de que ocurra la clase C_2 se calcula con la ecuación (3.12).

$$P_2(T) = \sum_{i=T+1}^{L-1} P_i \quad (3.12)$$

De esta forma, el valor medio de intensidad de los píxeles asignados a la clase C_1 es:

$$m_1(T) = \frac{1}{P_1(T)} \sum_{i=0}^T iP_i \quad (3.13)$$

Para la clase C_2 se tiene:

$$m_2(T) = \frac{1}{P_2(T)} \sum_{i=T+1}^{L-1} iP_i \quad (3.14)$$

La media acumulativa (promedio de intensidades) hasta el nivel T está dada por la ecuación (3.15):

$$m(T) = \sum_{i=0}^T iP_i \quad (3.15)$$

El promedio de intensidades de toda la imagen (esto es, la media global) está dada por:

$$m_G = \sum_{i=0}^{L-1} iP_i = P_1 m_1 + P_2 m_2 \quad (3.16)$$

La varianza global (σ_G^2) o la varianza de la intensidad de todos los píxeles en la imagen es:

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 P_i \quad (3.17)$$

La varianza entre las clases (σ_B^2) se define según la ecuación (3.18) como:

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 \quad (3.18)$$

De esta forma se observa que entre más alejadas se encuentren las medias m_1 y m_2 entre sí, será más grande la varianza entre clases σ_B^2 , lo que indica que la varianza entre clases es una medida de *separabilidad* entre las clases. Así, para evaluar la calidad del umbral en el nivel T , empleamos la métrica siguiente:

$$\eta(T) = \frac{\sigma_B^2(T)}{\sigma_G^2} \quad (3.19)$$

Entonces, el objetivo es determinar el valor del umbral óptimo T^* que maximice la varianza entre clases $\sigma_B^2(T)$:

$$\sigma_B^2(T^*) = \max_{0 \leq T \leq L-1} \{\sigma_B^2(T)\} \quad (3.20)$$

En otras palabras, simplemente se requiere evaluar todos los valores enteros posibles de T para encontrar el valor del umbral óptimo T^* .

Una vez que el valor de T^* se ha obtenido, la imagen $f(x, y)$ es segmentada con base en la ecuación (3.9) que define el umbralado:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T^* \\ 0 & \text{si } f(x, y) \leq T^* \end{cases} \quad (3.21)$$

Un ejemplo de umbralado mediante el método de Otsu se muestra en la Figura 3.5 con el valor de umbral $T = 127$ que se encuentra a la mitad del histograma.

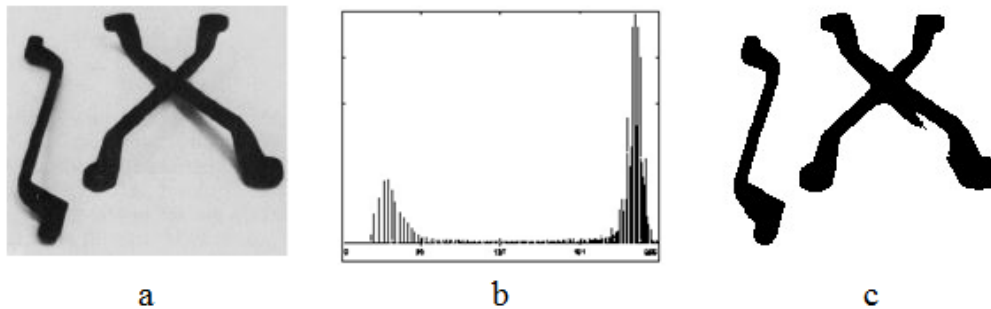


Figura 3.5 (a) Imagen original. (b) Histograma de la imagen. (c) Resultado del umbralado con $T = 127$.

3.2 Morfología matemática

La Morfología matemática puede definirse como una teoría para el análisis de estructuras espaciales. Se le llama morfología porque está dirigida al análisis de la forma de los objetos. Es matemática en el sentido de que el análisis se basa en la teoría de conjuntos, la geometría integral y el álgebra de retículas (*lattices*). Más allá de ser sólo una teoría, se trata de una poderosa técnica para el análisis de imágenes.

3.2.1 Elemento de estructura

El elemento de estructura (EE) es un conjunto pequeño de puntos con una estructura geométrica predefinida empleado para explorar la imagen de estudio. La forma y tamaño del EE debe ser adaptada a las propiedades geométricas de los objetos de la imagen procesada. Por ejemplo, EE lineales son aptos para la extracción de objetos lineales.

Las operaciones morfológicas fundamentales, la erosión y la dilatación, requieren de la definición de un origen o referencia en el EE, para posicionar cada vez el EE en un pixel dado. Los elementos de estructura se designan comúnmente con la letra B .

3.2.2 Erosión

La erosión de un conjunto X por un elemento de estructura B es denotado por $\varepsilon_B(X)$ y se define como la región de puntos $x|x \in X$ tales que B está incluido en X cuando su origen está situado en x :

$$\varepsilon_B(X) = \{x|B_x \subseteq X\} \quad (3.22)$$

Similarmente, la erosión de una imagen f por un elemento de estructura B es denotada por $\varepsilon_B(f)$. El resultado de la erosión en un pixel x es el valor mínimo en la ventana definida por el EE cuando su origen se encuentra en x .

3.2.3 Dilatación

La dilatación es la operación dual de la erosión. La dilatación de un conjunto X por un elemento de estructura B es denotada por $\delta_B(X)$ y es definida como la región de puntos $x|x \in X$ tales que B coincide con X cuando su origen corresponde con x :

$$\delta_B(X) = \{x|B_x \cap X \neq \emptyset\} \quad (3.23)$$

3.3 Color

El color es una sensación creada por la excitación de nuestro sistema de visión por la radiación electromagnética conocida como luz (Plataniotis y otros, 2000)(Gonzalez y otros, 2008). De forma más específica, el color es el resultado perceptual de la luz en la región visible del espectro electromagnético, el cual tiene longitudes de onda en la región de 400nm a 700nm, que incide sobre la retina del ojo humano. Esto se presenta en la Figura 3.6.

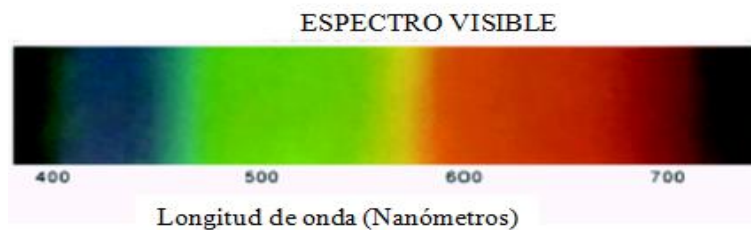


Figura 3.6 Longitudes de onda del espectro electromagnético visible.

La retina humana tiene tres tipos de células fotorreceptoras de color llamadas conos, los cuales responden a la radiación luminosa con curvas de respuesta espectral diferentes. Debido a que hay exactamente tres tipos de células foto receptoras, son necesarias y suficientes tres componentes numéricas para describir un color, considerando que se cuenta con funciones adecuadas para modelarlo. Por tanto, un color puede especificarse con un vector de tres

componentes. El conjunto de todos los colores forma un espacio de vectores llamado espacio de color o modelo de color. De esta forma, los tres componentes de un color pueden definirse de varias formas diferentes dando lugar a varios espacios de colores (Wyszecki y otros, 1982).

3.3.1 Modelos de color

En el dominio de las ciencias modernas, como la física, ingeniería, ciencias de la computación, psicología, etc., se distinguen cuatro familias de modelos de color:

- **Modelos de color colorimétricos.** Están basados en mediciones físicas del espectro de reflectancia. Generalmente se emplean tres filtros de color primario y un fotómetro como punto de partida para estos modelos.
- **Modelos de color físico-psicológicos.** Basados en la percepción humana del color. Tales modelos están basados en criterios de observación subjetiva con referencias comparativas, o son construidos mediante experimentación para obedecer a la percepción humana del color (por ejemplo, el modelo de intensidad, matiz y saturación).
- **Modelos de color fisiológicamente inspirados.** Están basados en la percepción de los tres colores primarios, con la ayuda de los tres tipos de conos presentes en la retina humana. El espacio de color Rojo-Verde-Azul (RGB) utilizado en el hardware de computadoras, es el mejor ejemplo conocido de un modelo de color fisiológicamente inspirado.
- **Modelos de colores opuestos.** Consisten en experimentos de percepción que emplean principalmente pares de colores primarios opuestos, tales como los pares de colores Amarillo-Azul y Rojo-Verde.

3.4 Restauración de imágenes

La restauración de imágenes es un proceso objetivo mediante el cual se busca recuperar una imagen que ha sido degradada, mediante algún conocimiento previo del fenómeno que degradó la imagen. De esta forma las técnicas de restauración están orientadas a modelar la función de degradación de la imagen y aplicar el proceso inverso para recuperar la imagen original.

3.4.1 Un modelo del fenómeno de degradación de una imagen

Modelamos el fenómeno de degradación de una imagen como una función que junto con ruido aditivo $\eta(x, y)$, opera sobre una imagen de entrada $f(x, y)$ para producir una imagen degradada $g(x, y)$. El esquema es presentado en la Figura 3.7.

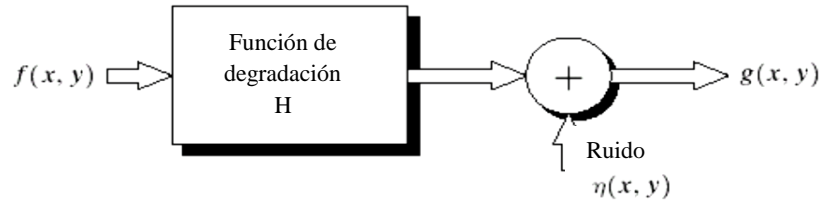


Figura 3.7 Modelo de degradación de la imagen.

El objetivo de la restauración es obtener una función $f'(x, y)$ de la imagen original. Así mismo, se busca que $f'(x, y)$ sea lo más parecida posible a $f(x, y)$. El proceso completo se presenta gráficamente en la Figura 3.8.

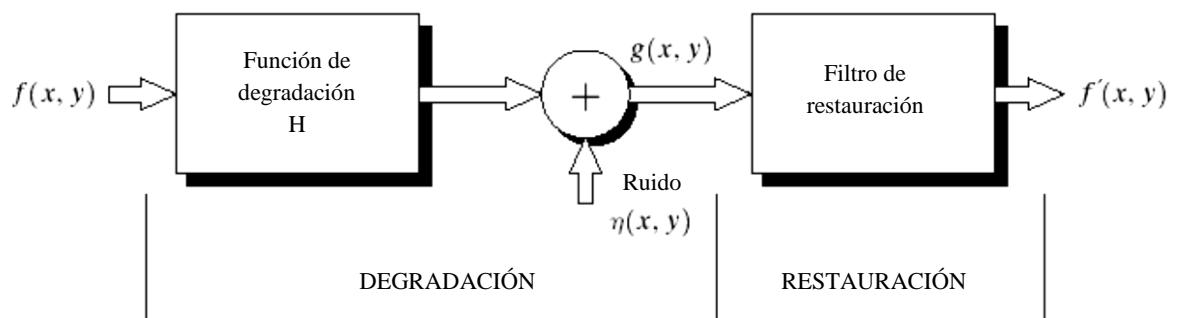


Figura 3.8 Diagrama del proceso de degradación y restauración de la imagen.

3.4.2 Clasificación de las técnicas de restauración

En general, podemos clasificar las técnicas de restauración de imágenes existentes de acuerdo al dominio sobre el cual se efectúa el procesamiento:

- Dominio espacial
 - Filtros de eliminación de ruido
 - Filtros de medias
 - Mediana
 - Media aritmética
 - Filtros de orden estadístico
 - Filtros adaptativos
 - Estimación de la función de degradación
 - Modelado
 - Experimentación
 - Interpolación
 - Repintado
 - Restauración de estructuras mediante ecuaciones diferenciales parciales

- Restauración de texturas basada en ejemplares
- Enfoques híbridos aplicados a la restauración.
- Dominio de la frecuencia
 - Filtros de la transformada de Fourier
 - Filtro selectivo de frecuencias
 - Filtro rechazo de banda
 - Filtro pasa banda
 - Filtros adaptativos
 - Estimación de la función de degradación

En nuestro trabajo nos referiremos fundamentalmente a la restauración por repintado debido a la actualidad de la problemática y las propuestas de solución.

Capítulo 4 Metodología

4.1 - Metodología

4.2 - Iteraciones del algoritmo

4.3 - Interfaz gráfica

En este capítulo se presenta el desarrollo de los algoritmos propuestos para la restauración de imágenes integrados por 4 componentes principales que se describen en la primera sección; en la segunda sección se hace notar la característica de las iteraciones en el algoritmo, y finalmente en la tercera sección se presenta la interfaz gráfica para uso de la metodología por parte del usuario.

4.1 Metodología

Debido a la relevancia del trabajo de Criminisi, Pérez y Toyama, 2004, en general el proceso de repintado se ha presentado en la literatura a través de cuatro etapas, mismas que se enlistan a continuación:

- Obtención de la máscara de repintado.
- Cálculo de prioridades para el orden de repintado.
- Búsqueda (o determinación) de la información para repintar.
- Actualización de la imagen.

El presente trabajo está basado principalmente en la metodología del trabajo de Criminisi et al., 2004. A partir del mismo, se efectuaron adecuaciones para obtener el algoritmo propuesto. El diagrama de flujo general del algoritmo se presenta en la Figura 4.1.

Inicialmente el usuario selecciona de forma manual la zona a restaurar, lo cual genera una máscara que funcionará para marcar la zona objetivo durante la ejecución del algoritmo. Posteriormente, se iterará hasta que no existan píxeles para repintar efectuando los siguientes pasos: (a) A partir del orden de repintado seleccionar el conjunto de píxeles que será procesado, (b) Efectuar la búsqueda de los píxeles más similares al conjunto de píxeles seleccionados que estén presentes en la imagen, (c) Actualizar los valores de intensidad del conjunto de píxeles seleccionados, a partir de los píxeles más similares encontrados.

Así mismo, nos basaremos en la terminología empleada en la literatura del repintado con fines de comparación.

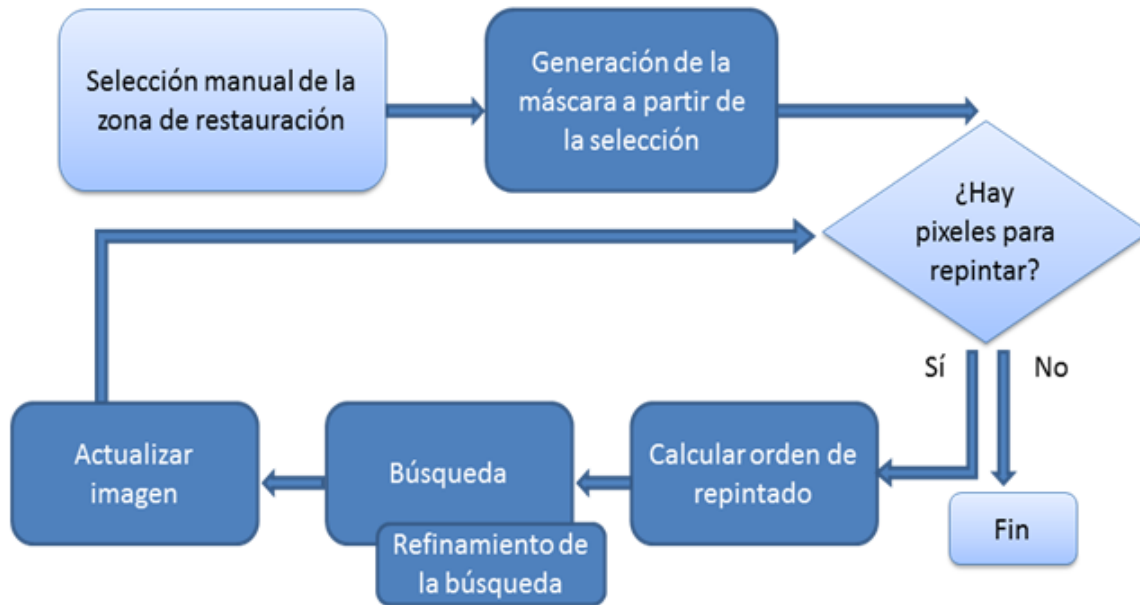


Figura 4.1 Diagrama de bloques general del algoritmo de repintado.

La región que será rellenada (zona objetivo) se denomina Ω . El contorno de la zona objetivo, el cual, evoluciona hacia adentro a medida que progresa el algoritmo se denomina: $\delta\Omega$ y la región origen, que permanece fija durante el progreso del algoritmo se denota como: Φ como se muestra en la Figura 4.2.

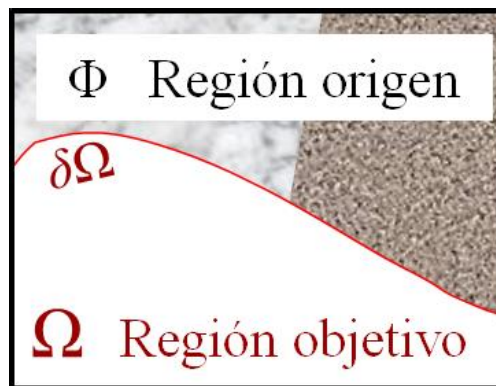


Figura 4.2 Terminología de las zonas principales en la imagen para el algoritmo de repintado.

4.1.1 Obtención de la máscara de repintado

Debido a que las aplicaciones del repintado varían desde la restauración de imágenes dañadas por grietas hasta la eliminación de objetos en la imagen, habitualmente la selección de la zona de repintado se efectúa manualmente (Bertalmio, Sapiro, y otros, 2000) (Criminisi, Pérez y Toyama, 2004), con la finalidad de permitir al usuario seleccionar cuidadosamente las zonas a procesar.

En la actualidad existen trabajos enfocados en la detección automática de grietas y su restauración mediante técnicas de repintado (Ruzic y otros, 2011). Sin embargo, la finalidad del presente trabajo es el diseño de algoritmos que no sólo estén orientados a una única aplicación.

Refiriéndonos a la Figura 2.2, la máscara puede ser vista como una imagen binaria, en la cual los pixeles blancos indican los pixeles que en la imagen deben ser restaurados. En cada paso del algoritmo, los pixeles marcados para ser repintados no serán tomados en cuenta para proporcionar información acerca de la imagen, pues se considera que la información de dichos pixeles es inexistente.

4.1.2 Cálculo de prioridades para el orden de repintado

De acuerdo con Criminisi, Pérez y Toyama, 2004, el orden de repintado es crítico, ya que una primera aproximación, como por ejemplo repintar mediante el pelado de cebolla (*onion peel* del inglés), que consiste en rodear la zona objetivo en un sentido (el de las manecillas del reloj, por ejemplo) hasta terminar el procesamiento, puede resultar en una restauración con defectos (Figura 4.3).

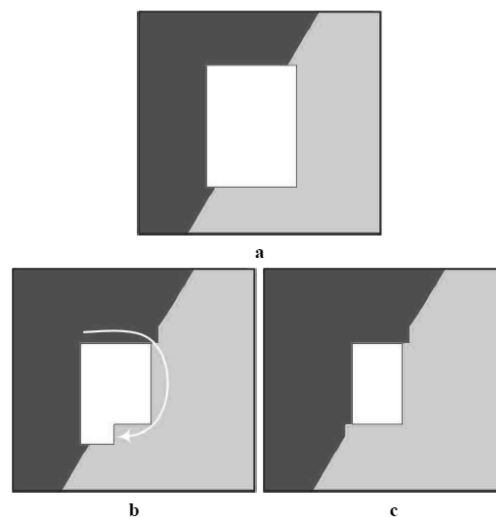


Figura 4.3 Método de pelado de cebolla para el repintado (a) Región objetivo mostrada en blanco (b) Sentido en el que avanza el algoritmo (c) Se obtienen artefactos en el resultado final.

De esta forma deben tomarse en cuenta ciertas características en la imagen para definir el orden del repintado. En este caso la información proporcionada por los bordes de la imagen indicaría hacia donde se debe hacer el repintado, tomando en cuenta que los pixeles que se encuentren sobre la continuación de un borde deberían tener una mayor prioridad (Masnou y Morel, 1998).

Así, a través de una vecindad de pixeles de tamaño específico se define un parche Ψ_p centrado en el pixel p , como una unidad de procesamiento para el algoritmo de repintado; con esto en

mente se buscan parches que (a) estén en la continuación de bordes fuertes y (b) estén rodeados de píxeles de alta confianza. Dado un parche Ψ_p centrado en el punto p , se define su prioridad $P(p)$ como el producto de dos términos:

$$P(p) = C(p)D(p)$$

Donde $C(p)$ es el término de *confianza* y $D(p)$ el término de *datos*. La confianza está representada por el promedio de los valores de los píxeles existentes en el parche, el cual puede ser visto como una medida de la cantidad de información confiable alrededor del píxel p .

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I - \Omega)} C(q)}{|\Psi_p|}$$

Donde $|\Psi_p|$ representa el tamaño del parche, I representa la imagen en su totalidad y Ω la zona objetivo. Este término nos indica el nivel de confianza que tenemos en el parche a partir del número de píxeles del parche que también existen en la imagen, el que irá disminuyendo a medida que avanza el algoritmo, puesto que en cada iteración se recalcula la confianza en los parches de repintado para actualizar las prioridades.

El término de datos es una función de la fuerza de las isofotas o píxeles con similar intensidad que entran hacia la zona de repintado. Este término incrementa la prioridad de un parche cuando una isofota fluye hacia la zona de repintado.

$$D(p) = \frac{|\nabla I_p^\perp \cdot n_p|}{\alpha}$$

Donde α es un factor de normalización, típicamente $\alpha = 255$ para imágenes de niveles de gris, n_p es un vector unitario ortogonal al frente de repintado $\delta\Omega$ en el punto p y finalmente \perp es el operador ortogonal. El efectuar un producto punto entre los dos vectores, permite proyectar al vector gradiente sobre la dirección hacia donde “avanzaría” un nuevo parche de repintado. La idea es que entre más representativo sea el borde, se tenga una mayor prioridad.

Finalmente en la Figura 4.4 se presenta la notación general para el cálculo de la prioridad. A partir del parche Ψ_p , n_p es el vector normal al contorno $\delta\Omega$ de la región objetivo Ω y ∇I_p^\perp es la isofota (dirección e intensidad) en el punto p . Toda la imagen se denota con I .

Es importante mencionar que para el cálculo del gradiente se emplea un filtro Sobel debido a que se requiere contar con las componentes en x y y del vector, ya que la simplicidad del filtro Sobel como aproximación al gradiente de la imagen, ha sido considerado como el más conveniente según se ha reportado en la literatura.

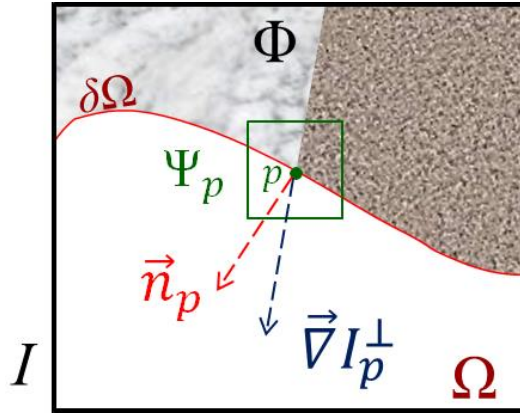


Figura 4.4 Notación de los elementos para el cálculo de prioridades.

Los autores Cheng y otros, 2005, descubrieron que el término de confianza definido por Criminisi decrece exponencialmente y propusieron una forma aditiva de prioridad, la cual utiliza un factor de regularización para controlar el comportamiento de la confianza, quedando ahora de la siguiente forma:

$$R_c(p) = (1 - \omega) C(p) + \omega, \quad 0 \leq \omega \leq 1,$$

El valor de $\omega = 0.7$ fue encontrado empíricamente, con el cual el comportamiento de la confianza se mantiene, pues el decaimiento es menos rápido. Finalmente, para mantener un balance entre el término de confianza y el término de datos, estos se definen de la siguiente forma:

$$P(p) = R_c(p) + D(p)$$

4.1.3 Búsqueda de la información para repintar

De acuerdo con Efros y Leung, 1999, la síntesis de textura busca a partir de una imagen pequeña como semilla inicial, crecerla hacia fuera para crear una imagen más grande, a un pixel a la vez. Se modela la imagen como un campo de Markov aleatorio, en el cual, a partir de un pixel y dados todos sus vecinos sintetizados hasta el momento, se estima la distribución condicional comparando toda la imagen de muestra y encontrando a los vecinos similares. Esto se ejemplifica en la Figura 4.5:

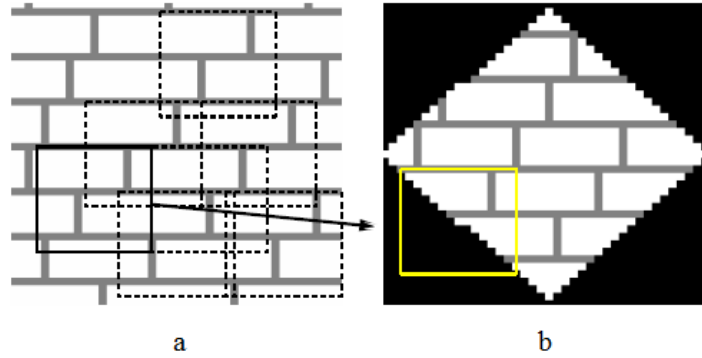


Figura 4.5 Ejemplo de síntesis de textura. (a) Imagen de muestra (b) Imagen a sintetizar.

De esta forma, dada una imagen de textura de muestra y una imagen siendo sintetizada un pixel a la vez, para sintetizar un pixel primero el algoritmo encuentra todos los vecinos en la imagen de muestra (cuadros punteados) que son similares a la vecindad del pixel (cuadro amarillo) y entonces elige una aleatoriamente y toma su centro para ser el nuevo pixel sintetizado.

Partiendo de esta idea, una vez que se ha seleccionado el parche que será rellenado con base en su prioridad, se efectúa la búsqueda del parche más similar; para ello se busca en toda la imagen la vecindad más similar (a través de obtener una distancia mínima) a los pixeles existentes en la imagen del parche objetivo como se muestra a continuación:

$$\Psi_{q'} = \arg \min_{\Psi_q \in \varphi} d(\Psi_{p'}, \Psi_q)$$

Donde $d = \text{SSD}$ o suma de diferencias cuadradas (*Sum of Square Differences* en inglés) calculada sobre cada parche, beneficiando a aquellos parches cuyo error en la diferencia de pixeles sea menor:

$$\sum_{r \in \Psi_p \cap \Omega^c} [\Psi_p(r) - \Psi_{q_j}(r)]^2$$

Sin embargo, no se considera el caso en que existan parches con distancia mínima igual, para lo cual en primera instancia se elegiría uno de los parches con distancia mínima de forma aleatoria o al último parche con distancia mínima encontrado. Por ello, en Lingni, Luat y de With, 2012, se sugiere el uso de una distancia hacia el pixel p , ya que se asume que es más probable que los parches más cercanos al parche que se analiza, contengan la información más confiable. La distancia inicial se muestra a continuación:

$$D(\Psi_p, \Psi_q) = |X_p - X_q| + |Y_p - Y_q|$$

No obstante, esta distancia no necesariamente refleja el parche más cercano del punto p , por lo que se propone emplear la distancia Euclidiana para los desempates, puesto que refleja de

forma más precisa la distancia entre píxeles, en este caso, entre parches con centro en p . Cabe mencionar que sólo se toman en cuenta los parches enteramente compuestos por píxeles dentro de la imagen.

El trabajo de Criminisi, al estar basado en síntesis de textura, considera toda la imagen para buscar el parche más similar; no obstante, se ha encontrado que a medida que la búsqueda se aleja del píxel de interés, es menos probable encontrar información confiable para repintar, por lo que basado en propuestas de algunos autores, se emplea una ventana de búsqueda que limita la búsqueda a un tamaño de vecindad definida sobre la que se buscará el parche más similar, como se muestra en la Figura 4.6, reduciendo así los tiempos de ejecución del algoritmo.

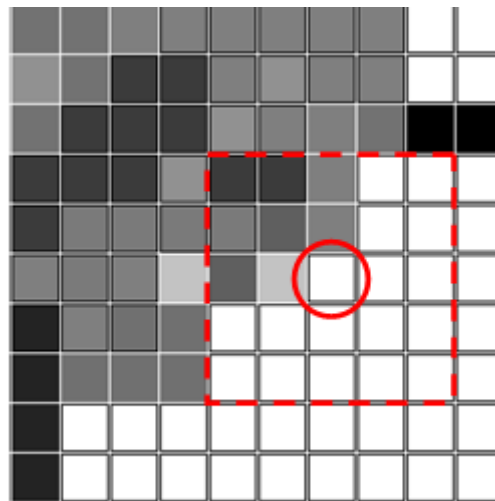


Figura 4.6 Ventana de búsqueda.

El tamaño adecuado para la ventana de búsqueda es aún una cuestión sin respuesta definitiva, por lo que algunos autores sugieren que sea de la mitad de las dimensiones de la imagen o de acuerdo a las dimensiones de la máscara de repintado (Anupam, Goyal y Diwakar, 2010). Sin embargo, para efectos de implementación de nuestro trabajo, se toma como valor por defecto una ventana de búsqueda de 29x29 píxeles por haberse obtenido mejores resultados que con ventanas de menor tamaño. La obtención del tamaño de la ventana de búsqueda mencionado se indica en el capítulo 5.

4.1.4 Actualización de la imagen

Finalmente, una vez encontrado el parche más similar, se procede a copiar desde el parche encontrado cada uno de los píxeles que no existen en la imagen dentro del parche. Se actualiza también la información relacionada con el término de datos para el cálculo de la prioridad, de forma que no sea necesario recalcular bordes en la imagen en cada iteración del algoritmo. La Figura 4.7 muestra el parche actualizado con la información del parche más similar encontrado.

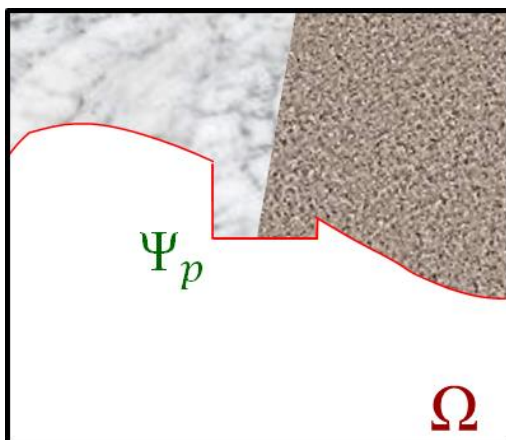


Figura 4.7 Actualización del parche de repintado.

4.2 Iteraciones del algoritmo

El algoritmo continúa mientras existan pixeles marcados en la máscara para repintado. En cada iteración se efectúa un cálculo de prioridades sobre los pixeles ubicados en el borde de repintado, es decir, en cada iteración se actualiza únicamente un parche, por lo que el número de iteraciones dependerá del tamaño del parche seleccionado. El proceso general de cada una de las iteraciones se describe en la Figura 4.8. La integración de un cálculo de prioridad normalizado con la delimitación de la ventana de búsqueda y el refinamiento de la búsqueda del parche más similar constituye la aportación más importante, ya que actualmente no se han empleado estas técnicas en conjunto hasta donde tenemos conocimiento. Dichas aportaciones se han subrayado dentro del pseudocódigo del algoritmo de repintado que se muestra en la Figura 4.8.

Algoritmo de Repintado	
Entrada:	Imagen una imagen para repintar Máscara una imagen binaria que indica la zona a repintar Tamaño_parche un entero positivo Tamaño_ventana_búsqueda un entero positivo
Salida:	Imagen_resultante la imagen repintada
Precondición:	Imagen , Máscara e Imagen_resultante tienen las mismas dimensiones.
1:	<code>Imagen_resultante = Imagen - Máscara</code>
2:	<code>gradiente_imagen = calcula_gradiente_sobel(Imagen)</code>
3:	mientras <code>conteo_pixeles_blanco(Máscara) > 0</code> :
4:	<code>vector_normal = calcula_vector_normal(Máscara)</code>
5:	para cada parche en <code>Máscara</code> :

```

6:          si está_en_limite_de_repintado(parche) :
7:          calcula prioridad(parche, gradiente imagen,
vector normal)
8:          si prioridad(parche) es máxima:
9:          parche_mayor_prioridad = parche
10:         para cada parche en una vecindad de tamaño
Tamaño ventana búsqueda
11:         si similitud(parche(parche_actual),parche) es
máxima :
12:         si hay_varios_parches_con_similitud_máxima():
13:         parche más similar =
elegir parche más cercano()
14:         otro:
15:         parche_más_similar = parche
16:         actualizar parche(parche_actual, parche más similar)

```

Figura 4.8 Seudocódigo de la metodología.

4.3 Interfaz gráfica

Como parte del sistema *Image processing system for teaching* desarrollado por el Dr. Edgardo Manuel Felipe Riverón, se desarrolló la metodología como módulos adicionales bajo las secciones de restauración de imágenes y la máscara de repintado.

4.3.1 Módulo de creación de la máscara de repintado

Permite cargar una imagen y marcar directamente sobre ésta la zona objetivo mediante un pincel, como en una aplicación tipo *Paint-Brush*. El tamaño del pincel y el color (únicamente para fines ilustrativos, pues el color no afecta en nada a la máscara) con el que se mostrará la máscara de repintado, se modifican con los medios que se muestran en la parte superior de la Figura 4.9.

Al pulsar el botón **Generar**, se obtienen dos imágenes, la primera con la marca de repintado únicamente para fines ilustrativos; la segunda se trata de una imagen binaria con la marca de repintado en píxeles blancos y el resto con píxeles negros, la cual servirá para alimentar al algoritmo para su posterior procesamiento como se observa en la Figura 4.10.

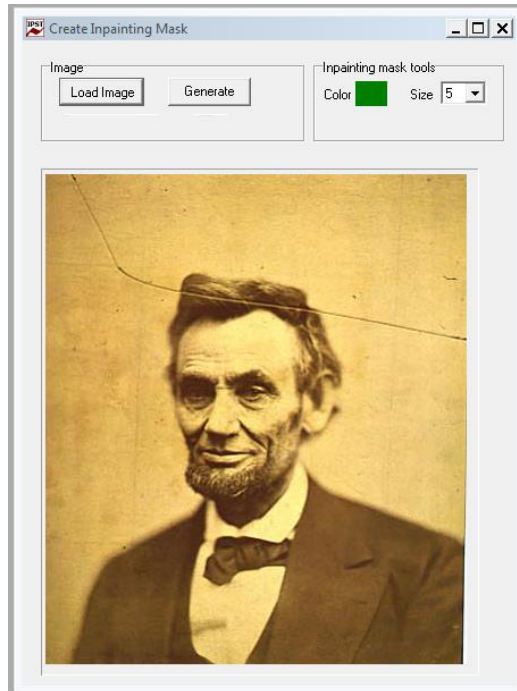


Figura 4.9 Módulo de creación de la máscara de repintado.

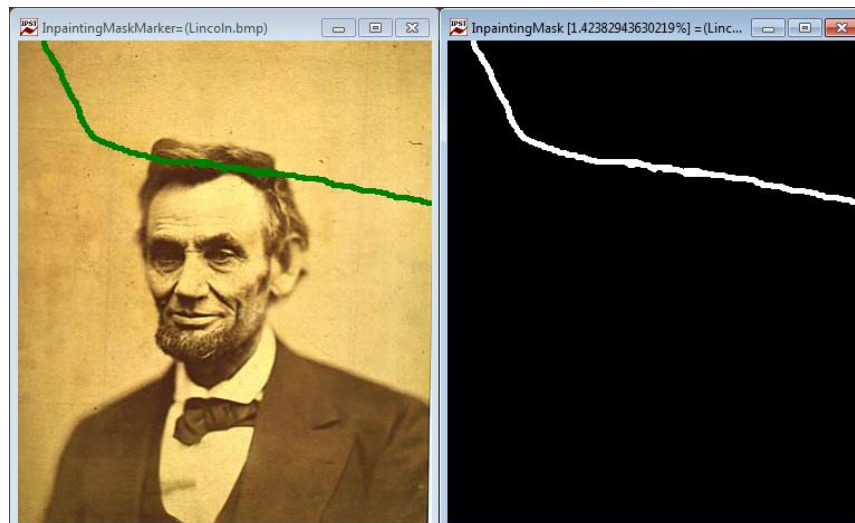


Figura 4.10 Resultado del módulo de creación de la máscara de repintado (a) Imagen de repintado (b) Máscara de repintado.

4.3.2 Módulo de restauración

El módulo de restauración incluye el establecimiento de los parámetros principales del trabajo, mediante los cuales se indica al algoritmo la imagen sobre la que se realizará la restauración y la máscara que indica la región objetivo. Así mismo se pueden especificar como parámetros el tamaño del parche y el tamaño de la ventana de búsqueda, los cuales varían obviamente los resultados brindados por el algoritmo. Esta pantalla se muestra en la Figura 4.11.

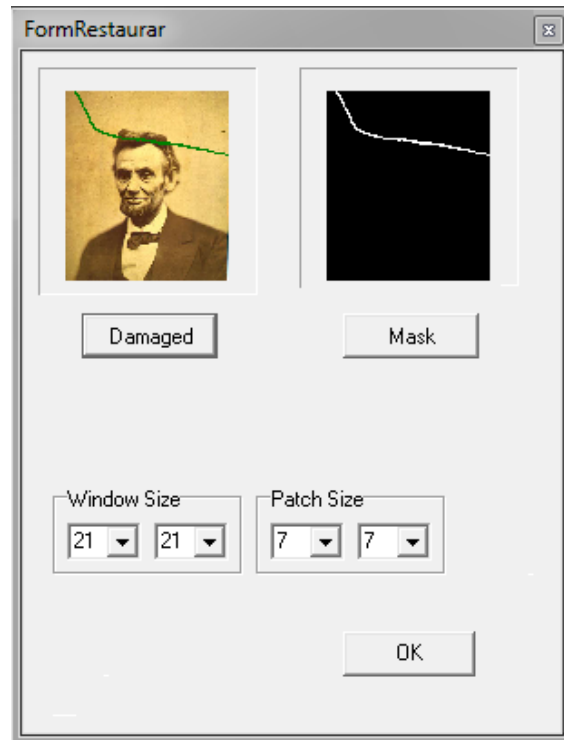


Figura 4.11 Módulo de restauración.

Una vez ingresadas las imágenes al algoritmo y modificados los parámetros si así se desea, comienza la ejecución del programa, el que al finalizar dará como resultado la imagen restaurada con la metodología propuesta.

Capítulo 5 Evaluación de los resultados

5.1 – Pruebas con imágenes sintéticas

5.2 – Pruebas con imágenes naturales

5.3 – Resultados con la restauración de imágenes reales

A continuación se evalúan los resultados mediante el uso de índices comúnmente empleados en la literatura para así poder comparar nuestros resultados con los de algunos trabajos actuales a considerarse dentro del estado del arte. Cabe mencionar que debido a que en la mayoría de los casos no se cuenta con la imagen original (la imagen antes de sufrir el proceso de degradación), en general se tiende a generar imágenes sintéticas para cuantificar de alguna forma la evaluación, de tal forma que se cuenta con una imagen inicial la cual se considera como la original. Así mismo, se empleó la base de datos TUM para efectos de comparación (Tiefenbacher y otros, 2015), aunque en algunos otros casos sólo se presenta el resultado del algoritmo, debido a que hay algunos trabajos de la literatura abierta que utilizan imágenes propias, lo que dificulta la comparación.

5.1 Pruebas con imágenes sintéticas

Para efectuar las pruebas se comenzará con una serie de imágenes sintéticas para evaluar la respuesta del algoritmo ante casos básicos. En la mayoría de los casos se evaluó con parches de 3x3, 5x5 y 9x9 píxeles.

5.1.1 Imagen sintética_1

La primera imagen sintética consiste en una imagen con líneas verticales de colores con grosores entre de 6 y 8 píxeles; la máscara de repintado está formada por una serie de líneas horizontales (cuatro) no completamente rectas. Esta zona a restaurar representa un 13.6% de daño en la imagen como se muestra en la Figura 5.1.

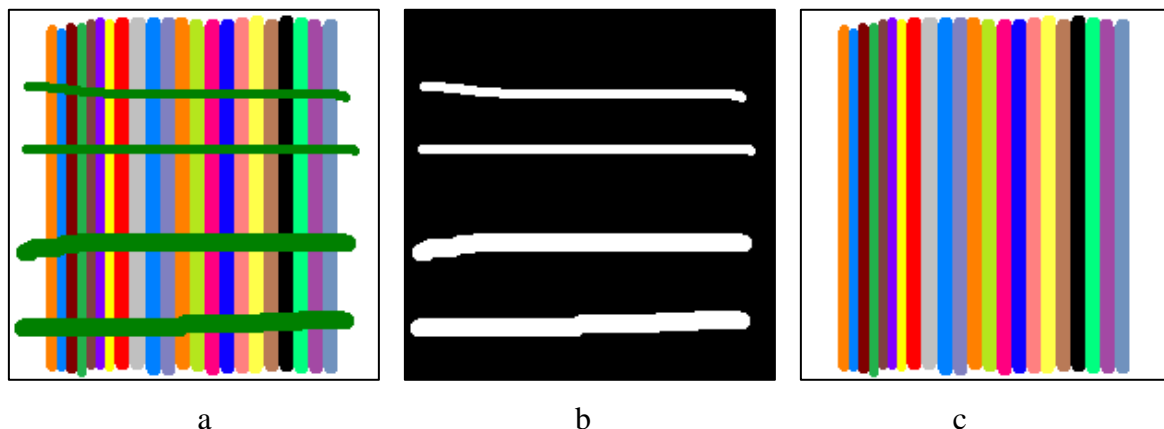


Figura 5.1 Imagen sintética_1 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 9x9 píxeles.

5.1.2 Imagen sintética_2

Se trata de las imágenes en la sección (5.1.1) rotadas 90° en el sentido de las manecillas del reloj; el porcentaje de la zona a repintar permanece igual. Figura 5.2.

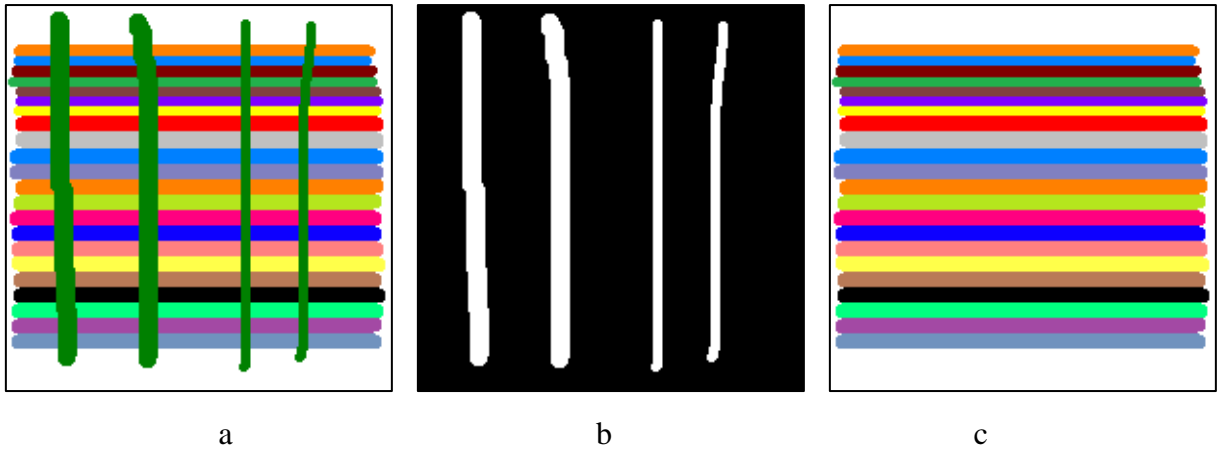


Figura 5.2 Imagen sintética_2 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 9x9 píxeles.

5.1.3 Imagen sintética_3

Se trata de las imágenes en la figura 5.1 rotadas 45° en el sentido de las manecillas del reloj, el porcentaje de la zona a repintar permanece igual. Figura 5.3:

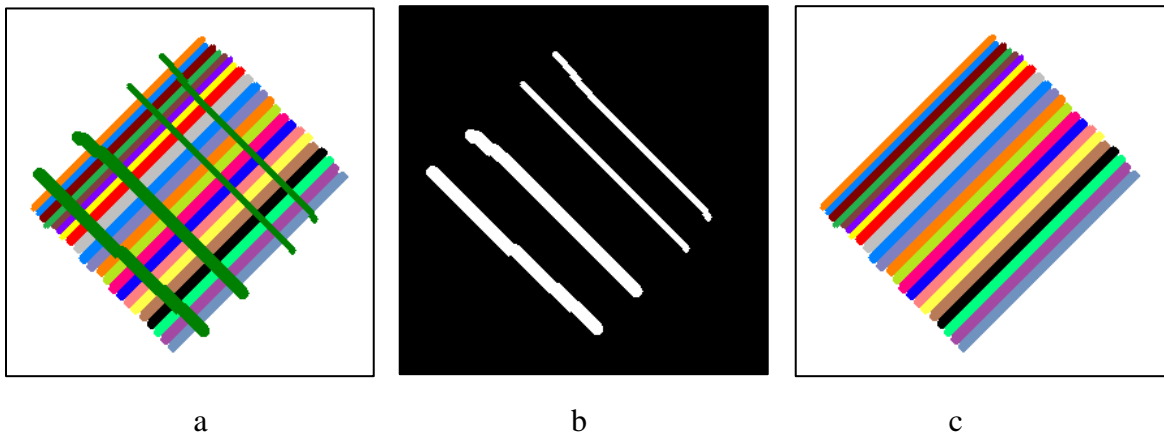


Figura 5.3 Imagen sintética_3 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 9x9 píxeles.

5.1.1 Imagen sintética_4

La imagen consiste en cuadros de colores de 30 píxeles de lado y una línea curvada forma la máscara de repintado, representando un 7.73% de daño en la imagen como se muestra en la Figura 5.4

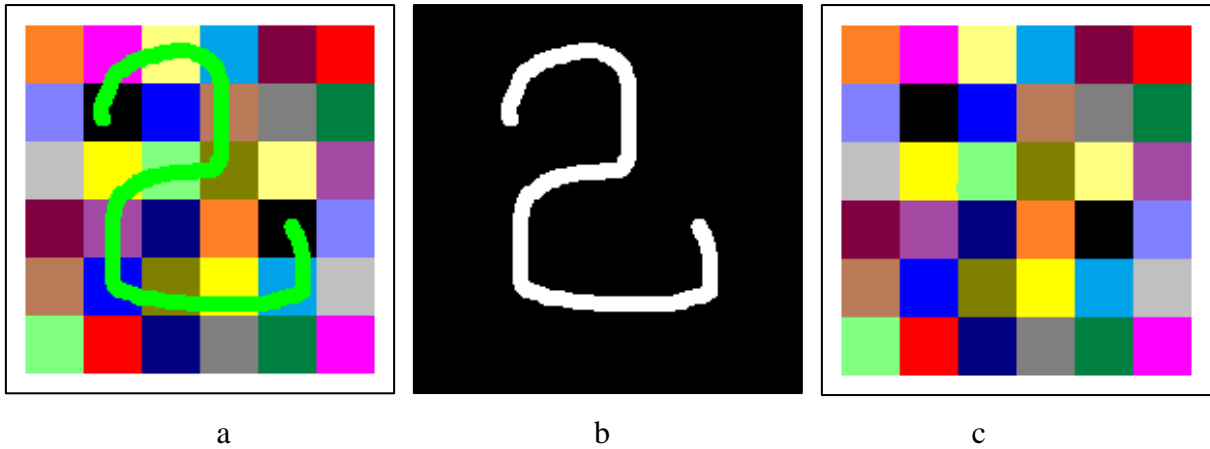


Figura 5.4 Imagen sintética_4 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 5x5 píxeles.

5.1.2 Imagen sintética_5

Se ha tomado una representación de la escala de grises para evaluar la capacidad del algoritmo para restaurar imágenes con texturas que varían de forma suave con niveles de gris. Una línea curva representa la máscara de repintado con 6.19% de daño en la imagen. Figura 5.5.

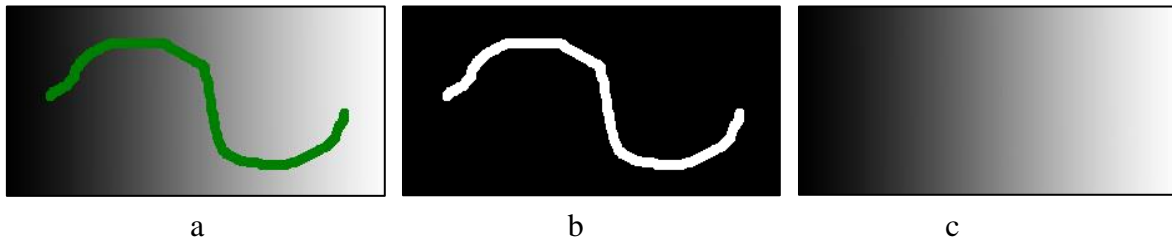


Figura 5.5 Imagen sintética_5 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 5x5 píxeles.

5.1.3 Imagen sintética_6

Para esta imagen sintética se ha tomado una representación de la escala de color para evaluar la capacidad del algoritmo para restaurar imágenes con texturas donde el color varía de forma suave. Una línea curva representa la máscara de repintado con 7.062% de daño en la imagen. Figura 5.6.

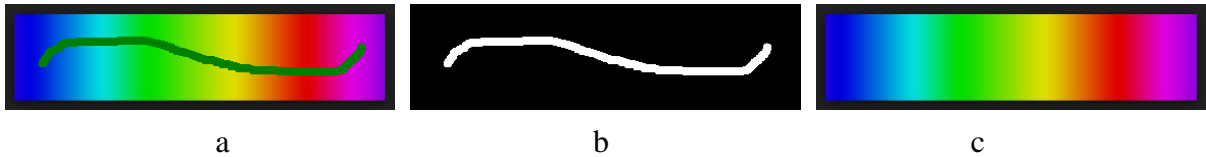


Figura 5.6 Imagen sintética_6 (a) Imagen para repintar (b) Máscara de repintado (c) Imagen restaurada con parches de 5x5 pixeles.

5.1.4 Resultados con imágenes sintéticas

Para evaluar los resultados con nuestros algoritmos de restauración por repintado con las imágenes sintéticas se empleó la función lógica XOR entre la imagen original y cada una de las imágenes resultantes con tamaños de parche de 3x3 pixeles, 5x5 pixeles y 9x9 pixeles; el resultado de ello fue siempre una imagen binaria con pixeles blancos producto de aquellos pixeles correspondientes de la imagen original y de la imagen resultante que no coincidieron. Entonces en esa imagen se cuentan los pixeles blancos y se saca la proporción con respecto a la totalidad de pixeles que tiene la imagen original. Es importante mencionar que esta operación se efectúa en los tres planos de colores rojo, verde y azul. Los resultados se muestran en las tablas 5.1 y 5.2.

Tabla 5.1 Número de pixeles distintos entre la imagen original y la restaurada.

Imagen	Pixeles con parche 3x3	Pixeles con parche 5x5	Pixeles con parche 9x9
Sintética 1	0	0	0
Sintética 2	0	0	0
Sintética 3	165	4	0
Sintética 4	6	158	456
Sintética 5	251	123	95
Sintética 6	0	0	186

Esta información se muestra en forma de porcentajes en la Tabla.2, para conocer el porcentaje de pixeles de la imagen restaurada que no coinciden con los de la imagen original.

Tabla.2 Porcentajes de acierto.

Imagen	Con parche 3x3	Con parche 5x5	Con parche 9x9
Sintética 1	100%	100%	100%
Sintética 2	100%	100%	100%
Sintética 3	99.792%	99.995%	100%
Sintética 4	99.985%	99.600%	98.900%
Sintética 5	99.400%	99.700%	99.800%
Sintética 6	100%	100%	99.500%

Promedio	99.863%	99.883%	99.367%
----------	---------	---------	---------

De las tablas se puede observar que a medida que se incrementa el tamaño del parche, se acarrean mayores errores durante el progreso del algoritmo, lo que obviamente se refleja en una disminución del porcentaje de precisión. No obstante, el porcentaje nunca es menor que 99% para las imágenes sintéticas presentadas.

5.1.5 El tamaño del parche en el resultado del algoritmo

Para verificar el tamaño adecuado del parche se efectuaron experimentos con una combinación de imagen sintética y una imagen real con estructura definida; en este caso la imagen sintética seleccionada fue la compuesta por cuadrados en colores y la imagen real fue una taza en niveles de gris sobrepuesta a ellos. La máscara de repintado en forma de una línea sinuosa cubre un 7.81% de la imagen completa.

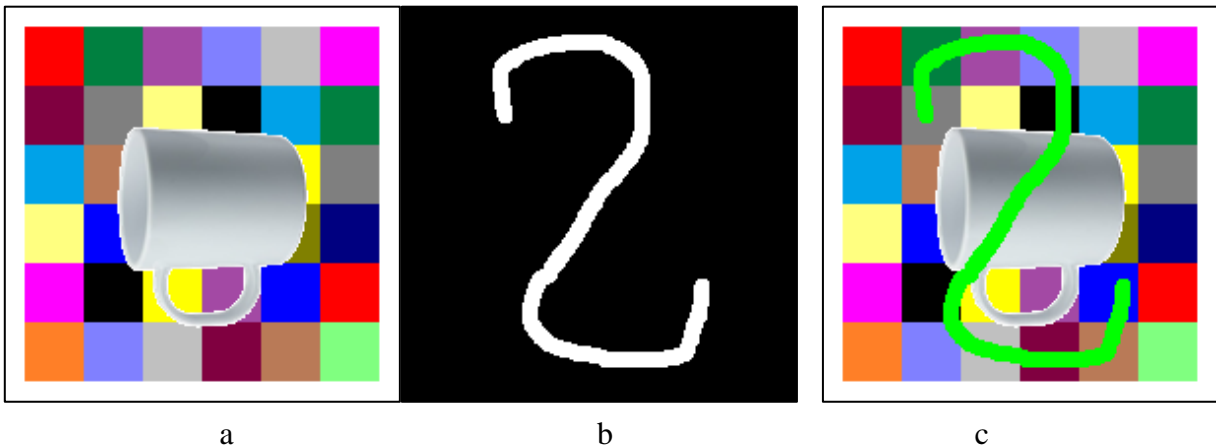


Figura 5.7 Imagen sintética para la evaluación del tamaño de parche (a) Imagen original (b) Máscara de repintado (c) Imagen para repintar.

Para efectos de comparación con algunos trabajos del estado del arte, se ha empleado el error cuadrático medio ECM (MSE en inglés de *Mean Squared Error*) para medir la diferencia promedio entre los píxeles obtenidos como resultado de la restauración y los píxeles de la imagen original. El ECM tiene las mismas unidades de medida que el cuadrado de la cantidad que se estima y se calcula mediante la siguiente expresión:

$$ECM = \frac{1}{n} \sum_{i=1}^n (\hat{I}_i - I_i)^2$$

Donde n es el número de píxeles comparados; para el caso de imágenes en color, se obtiene el número de píxeles de cada plano y se promedia entre el número total de planos en la imagen, en este caso 3, para el modelo de color RGB. \hat{I}_i representa el i -ésimo píxel en la imagen que se compara y I_i representa el i -ésimo píxel correspondiente en la imagen de referencia.

Se variaron los tamaños de parche en el algoritmo en incrementos de 2 en 2 para mantener la representación discreta del parche (tamaños de parche impares para mantener el centrado del punto p) y se obtuvo el error medio cuadrático al comparar la imagen restaurada y la original. La figura 5.8 muestra el comportamiento del error medio cuadrático al variar el tamaño del parche. El eje vertical indica el error medio cuadrático medido en píxeles cuadrados y el eje horizontal indica el tamaño de los parches cuadrados en píxeles.

Se puede observar que para tamaños de parche entre 3x3 y 9x9 píxeles se mantiene un error consistente; sin embargo, con parches de tamaño 11x11 píxeles en adelante, el error se incrementa considerablemente, lo cual indica que es más probable acarrear errores en la restauración con parches de mayor tamaño. De aquí se concluye igualmente que deben usarse tamaños pequeños de parche entre 3x3 y 9x9 píxeles. La ligera diferencia en dicho error se debe probablemente a las particularidades de la imagen que se restaura.

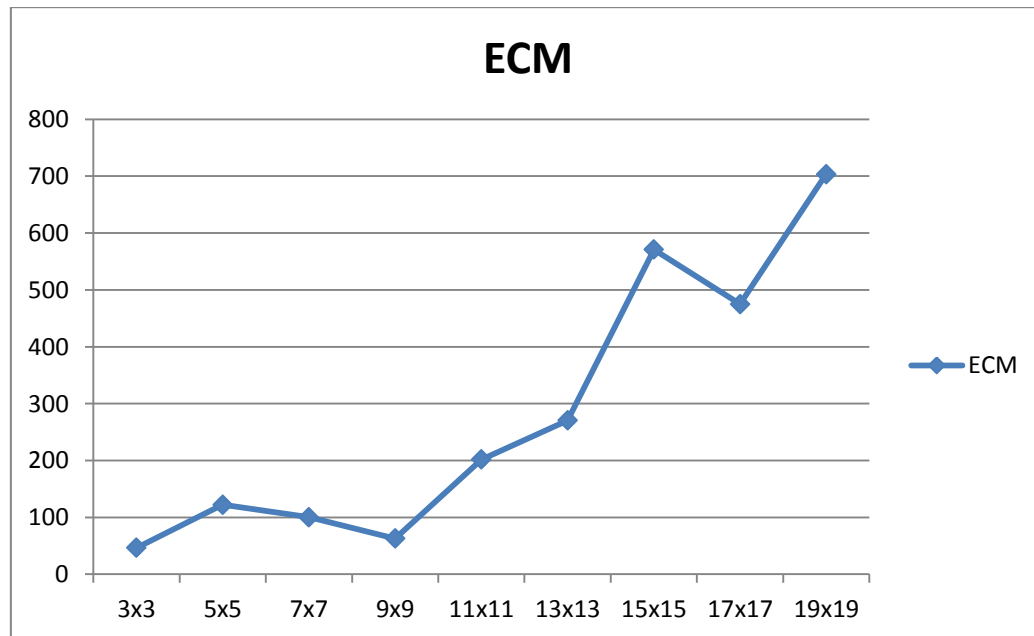


Figura 5.8 Error medio cuadrático al variar el tamaño de parche.

5.1.6 El tamaño de la ventana de búsqueda

Con el fin de reducir el tiempo de cómputo en la búsqueda del parche más similar, el parámetro ventana de búsqueda permite limitar el espacio de búsqueda. Se empleó la imagen en la Figura 5.7 para evaluar la variación del error cuadrático medio al ir aumentando el tamaño de la ventana de búsqueda, el cual se muestra en la Figura 5.9:

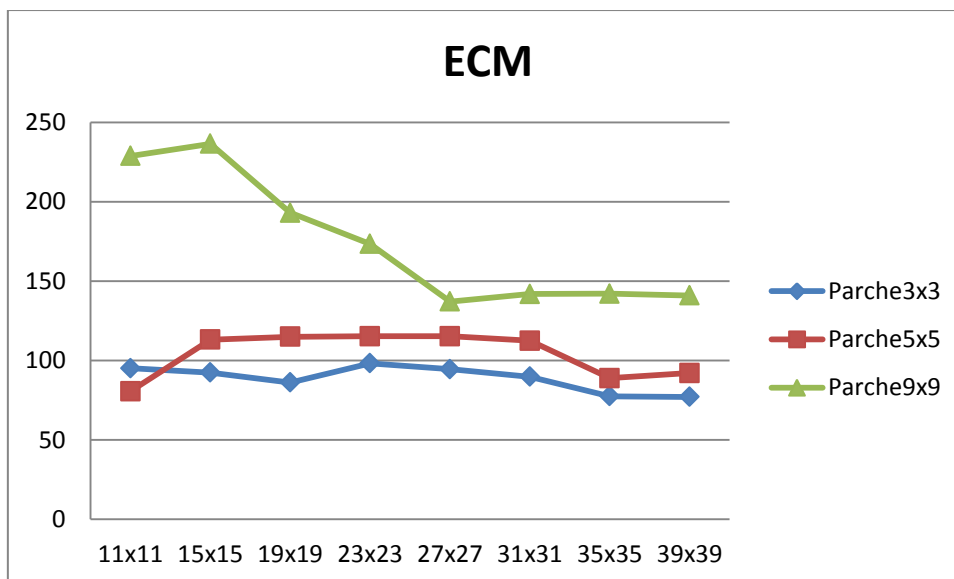


Figura 5.9 Variación del error cuadrático medio a medida que aumenta la ventana de búsqueda.

Cuando la ventana de búsqueda es muy pequeña, como en el caso de 11x11 píxeles, el error es grande para el parche de 9x9 en comparación con los parches de 3x3 y 5x5; sin embargo, a medida que aumenta el tamaño de la ventana de búsqueda, se puede notar que el error disminuye y se mantiene muy similar con ventanas de búsqueda entre 27x27 y 31x31. Posteriormente, aunque hay una ligera diferencia en el error, éste no es muy significativo. Por ello se decidió establecer por defecto un tamaño de ventana de búsqueda máximo de 29x29 píxeles; no obstante, su tamaño se puede cambiar de acuerdo a lo que exija alguna imagen específica.

5.2 Pruebas con imágenes naturales

5.2.1 Base de datos *TUM*

Para comparar los resultados con algunos algoritmos usados en trabajos considerados en el estado del arte, se hace uso de la base de datos *TUM - Image Inpainting Database*, de la Universidad de Munich (Tiefenbacher y otros, 2015). La base de datos cuenta con 17 imágenes de escenas reales de 640 x 480 píxeles; sin embargo, las imágenes tienen diversos tipos de daño, desde líneas hasta formas compactas de gran tamaño. Por ello, al enfocar el trabajo en la restauración de imágenes, cuando el daño no es tan extenso como en el caso de huecos o rayones, se seleccionaron las imágenes con un máximo de 2% de píxeles marcados para repintado.

Debido que las imágenes, en la mayoría de los casos, presentan muchos detalles como por ejemplo: ventanas, árboles, objetos conformados por otros objetos más pequeños, etc., se fijó una ventana de búsqueda de 39 x 39 píxeles al mejorar la calidad visual de los resultados al

permitir al algoritmo ampliar la vecindad de búsqueda. Así mismo, se han nombrado las imágenes empleadas de acuerdo a su contenido general para facilitar su identificación en la comparación y siguen el siguiente orden: calle, edificio, estructura, oficina y tierra.

La imagen Calle se muestra en la Figura 5.10 (a) con su respectiva máscara de repintado en la Figura 5.10 (b) con un 2% de píxeles marcados. En la Figura 5.11, Figura 5.12 y Figura 5.13 se presentan los resultados obtenidos con parches de 3x3, 5x5 y 9x9 píxeles, respectivamente.

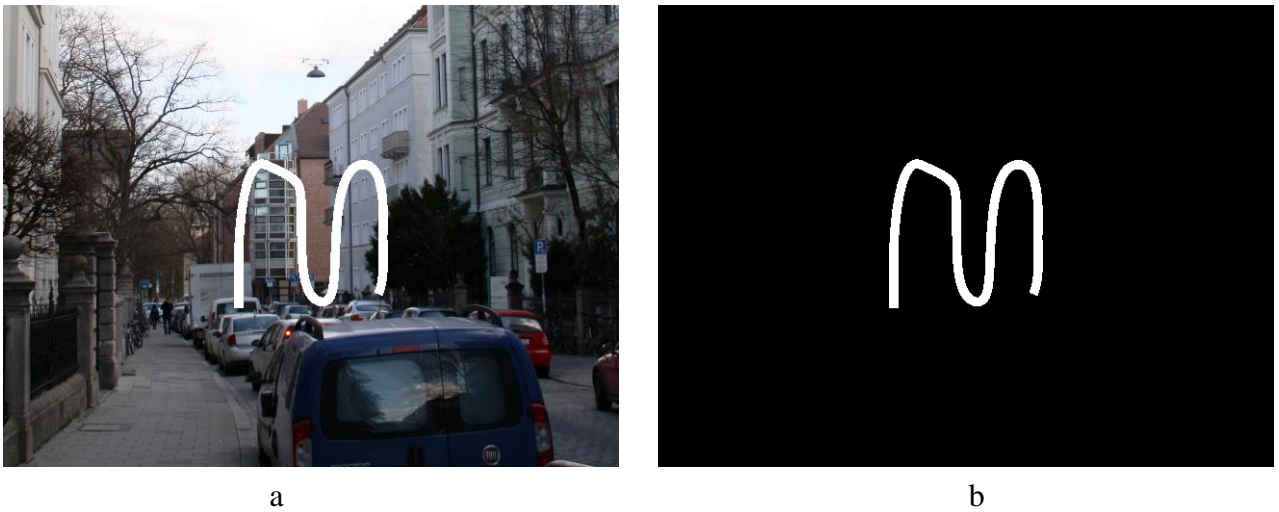


Figura 5.10 (a) Imagen Calle para repintar (b) Máscara de repintado.



Figura 5.11 Imagen Calle restaurada con parche de 3x3 píxeles.

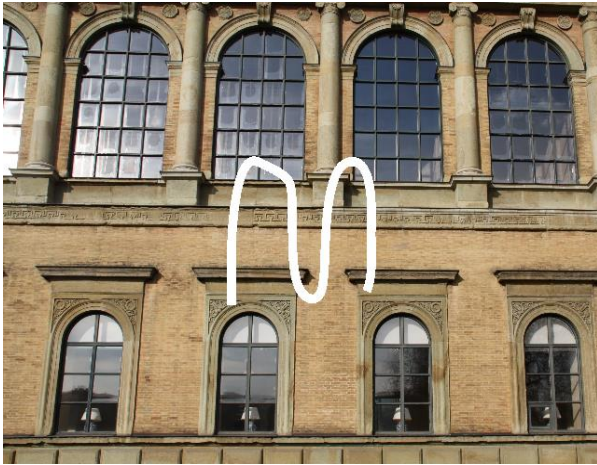


Figura 5.12 Imagen Calle restaurada con parche de 5x5 pixeles.

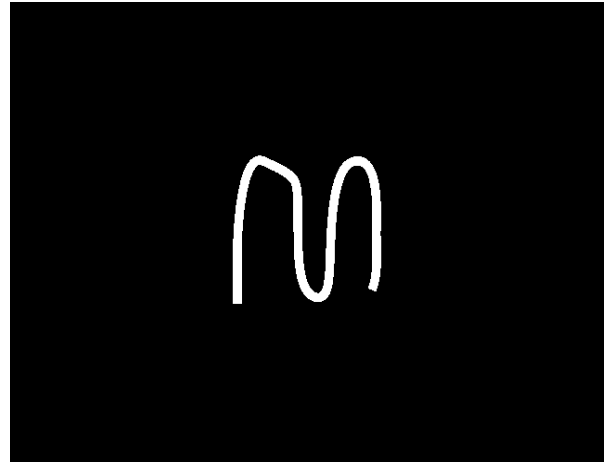


Figura 5.13 Imagen Calle restaurada con parche de 9x9 pixeles.

La siguiente imagen que se denomina Edificio se muestra en la Figura 5.14 (a) y su respectiva máscara de repintado en la Figura 5.14 (b) con un 2% de pixeles marcados. En la fFigura 5.15, Figura 5.16 y Figura 5.17 se presentan los resultados obtenidos con parches de 3x3, 5x5 y 9x9 pixeles, respectivamente.



a



b

Figura 5.14 (a) Imagen Edificio para repintar. (b) Máscara de repintado.



Figura 5.15 Imagen Edificio restaurada con parche de 3x3 pixeles.



Figura 5.16 Imagen Edificio restaurada con parche de 5x5 pixeles.

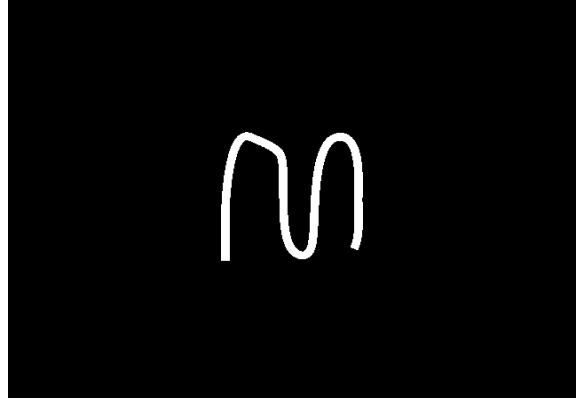


Figura 5.17 Imagen Edificio restaurada con parche de 9x9 pixeles.

La siguiente imagen denominada Estructura se muestra en la Figura 5.18 (a) y su respectiva máscara de repintado en la Figura 5.18 (b) con un 2% de pixeles marcados. En la Figura 5.19, Figura 5.20 y Figura 5.21 se presentan los resultados obtenidos con parches de 3x3, 5x5 y 9x9 pixeles, respectivamente.



a



b

Figura 5.18 (a) Imagen Estructura (b) Máscara de repintado.



Figura 5.19 Imagen Estructura restaurada con parche de 3x3 pixeles.



Figura 5.20 Imagen Estructura restaurada con parche de 5x5 pixeles.

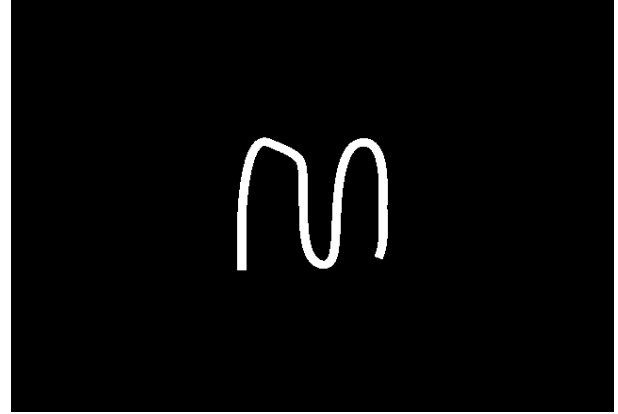


Figura 5.21 Imagen Estructura restaurada con parche de 9x9 pixeles.

La imagen Oficina se muestra en la Figura 5.22 (a) con su respectiva máscara de repintado en la Figura 5.22 (b) con un 2% de pixeles marcados. En la Figura 5.23, Figura 5.24 y Figura 5.25 se presentan los resultados obtenidos con parches de 3x3, 5x5 y 9x9 pixeles, respectivamente.



a



b

Figura 5.22 (a) Imagen Oficina (b) Máscara de repintado.

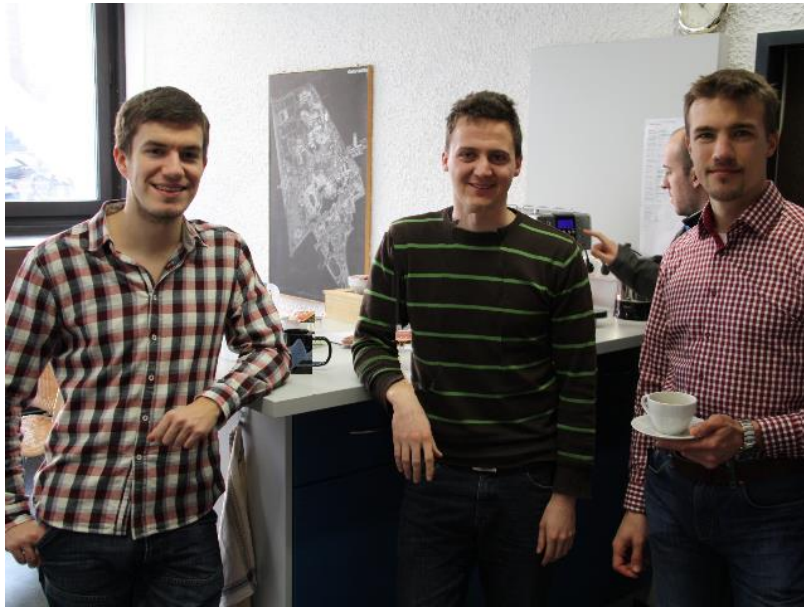


Figura 5.23 Imagen Oficina restaurada con parche de 3x3 pixeles.

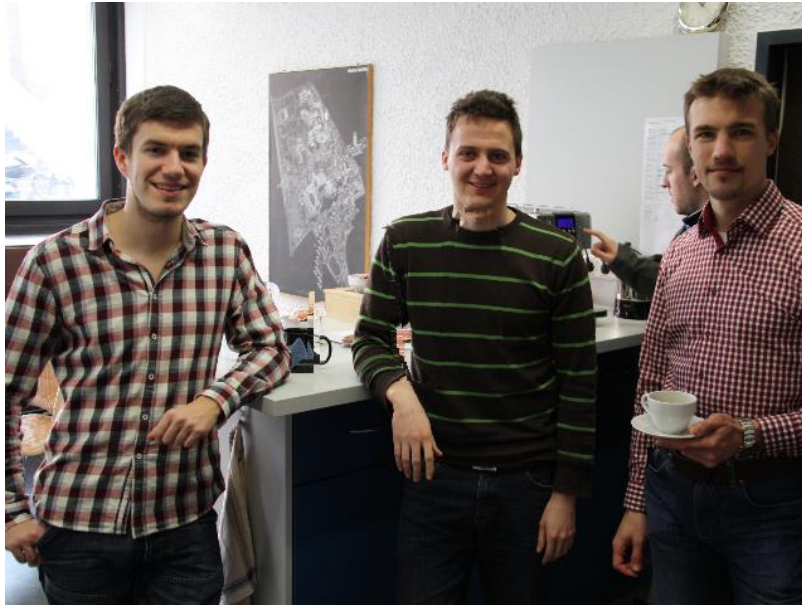


Figura 5.24 Imagen Oficina restaurada con parche de 5x5 pixeles.

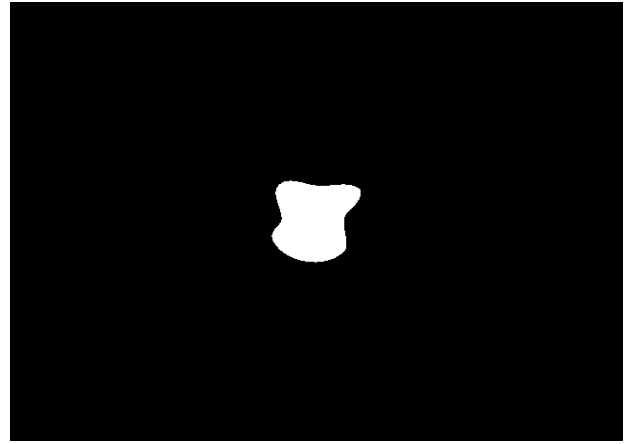


Figura 5.25 Imagen Oficina restaurada con parche de 5x5 pixeles.

Finalmente, la última imagen denominada Tierra se muestra en la Figura 5.26 (a) y su respectiva máscara de repintado en la Figura 5.26 (b) con un 2% de pixeles marcados. En la Figura 5.27, Figura 5.28 y Figura 5.29 se presentan los resultados obtenidos con parches de 3x3, 5x5 y 9x9 pixeles, respectivamente.



a



b

Figura 5.26 (a) Imagen Tierra (b) Máscara de repintado.



Figura 5.27 Imagen Tierra restaurada con parche de 3x3 pixeles.



Figura 5.28 Imagen Tierra restaurada con parche de 5x5 pixeles.



Figura 5.29 Imagen Tierra restaurada con parche de 9x9 pixeles.

5.2.2 Índice SSIM

Uno de los inconvenientes del error cuadrático medio para comparar los resultados de la restauración, es que sólo considera la diferencia promedio entre dos imágenes pixel a pixel. Es por ello que el índice SSIM (Zhou y otros, 2004) busca proveer un método objetivo para evaluar la calidad perceptual en la imagen, basado en el hecho de que la visión humana está altamente adaptada para extraer información estructural de una escena.

Se puede decir que las imágenes de escenas reales son altamente estructuradas, esto es, que sus píxeles presentan fuertes dependencias, especialmente cuando son espacialmente próximas y estas dependencias acarrearán información importante acerca de la estructura de los objetos en la escena natural. Primero se mide la luminancia de cada imagen, posteriormente se compara con el contraste en cada imagen lo cual posteriormente lleva a una comparación estructural con dicha información. Finalmente el índice combina cada uno de los resultados y se obtiene una medida de similitud.

Es uno de los índices más utilizados para evaluar los resultados del repintado cuando se dispone de la imagen de referencia. Varía desde 0 a 1; se obtiene el valor 1 cuando las dos imágenes (original y restaurada) son idénticas.

5.2.3 Comparación de resultados

Como se puede notar en la Figura 5.30, el algoritmo propuesto obtiene mejores resultados en la mayoría de los casos con las imágenes seleccionadas de la base de datos. Las imágenes con las que se obtienen mejores resultados son aquellas con texturas más definidas y sin gran cantidad de detalles, a diferencia del caso de rostros humanos, en el que se requiere conservar la simetría de los rasgos. Debido a que el índice varía entre 0 y 1, las diferencias entre cada resultado no son muy significativas, sin embargo, el algoritmo propuesto presenta mejores resultados en la mayoría de los casos.

Los resultados de todos los algoritmos tuvieron problemas con la imagen *Tierra*, debido a la gran cantidad de detalles formados por la tierra, piedras, trozos de madera, etc. que tiene la imagen.

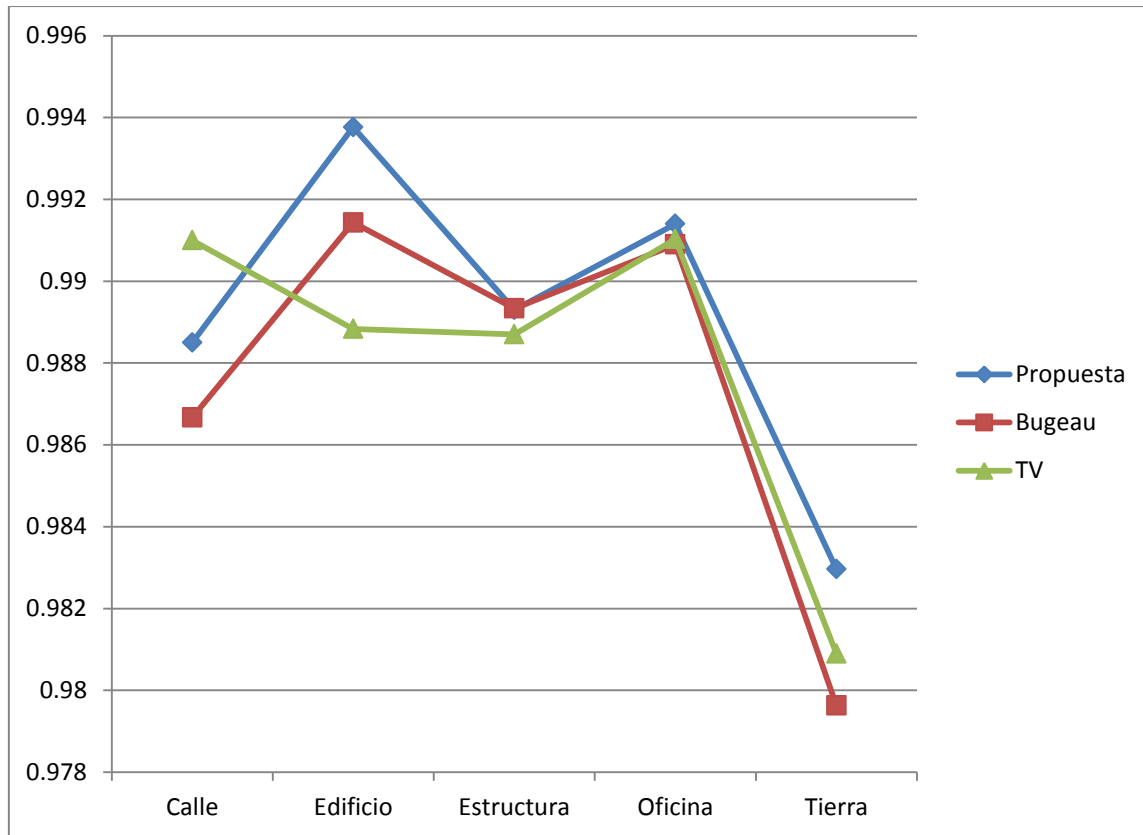


Figura 5.30 Comparación de resultados con nuestro algoritmo empleando el índice SSIM.

5.3 Resultados con la restauración de imágenes reales

Las siguientes imágenes presentan los resultados del algoritmo propuesto después de aplicarlo a imágenes reales que requerían ser restauradas. Al no contar con la imagen original, no es posible medir con índices cuantitativos las diferencias entre la original y la restaurada.

5.3.1 Imagen con texto sobrepuesto

La imagen contiene el texto *Hello* sobre una textura regular; la zona de repintado ocupa el 1.38% del total de la imagen como se observa en la Figura 5.31 (a).

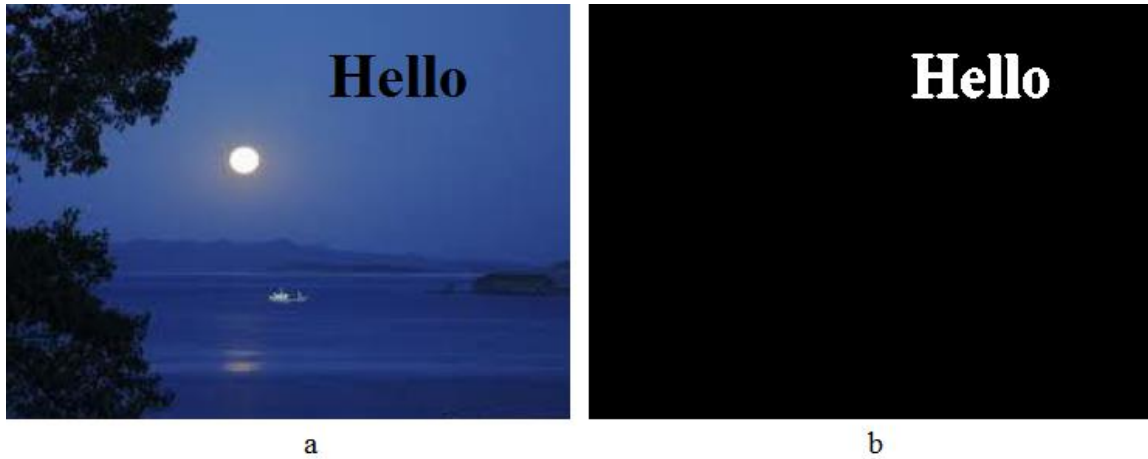


Figura 5.31 (a) Imagen con texto sobrepuesto (b) Máscara de repintado.

Después de la ejecución del algoritmo, el resultado de la restauración se puede observar en la Figura 5.32 (a), (b) y (c) para diferentes tamaños de parche. El fondo permanece uniforme y sin detalles que delaten que ha tenido lugar una restauración.



Figura 5.32 Imagen restaurada después de remover el texto con parches (a) 3x3 (b) 5x5 (c) 9x9 píxeles.

5.3.2 Imagen de Lincoln

La imagen presenta una grieta que atraviesa la imagen desde la zona lateral superior izquierda hacia la zona lateral central derecha. En total se tiene un 1.81% de daño después de enmascarar la grieta. Figura 5.33:

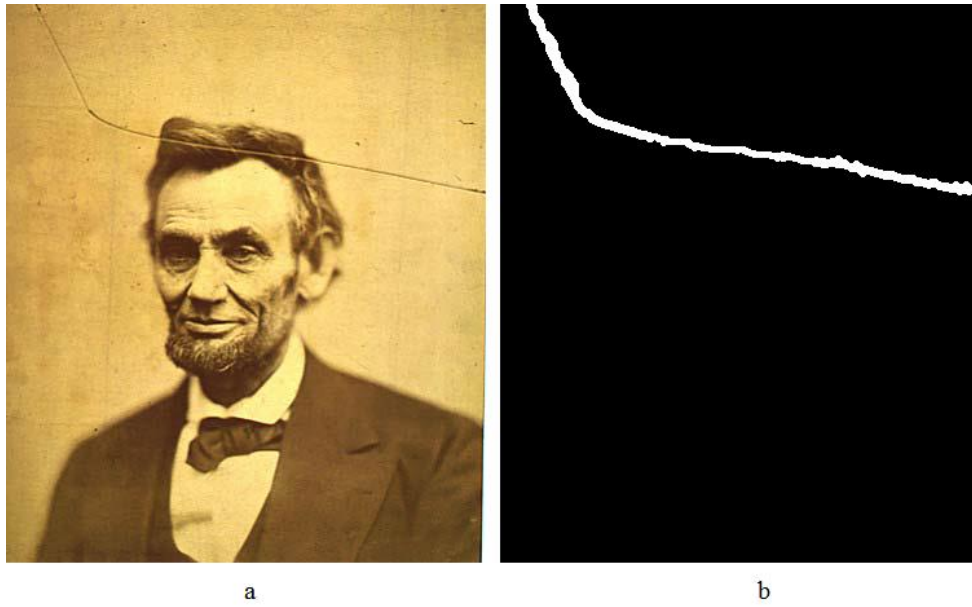


Figura 5.33 (a) Imagen de Lincoln (b) Máscara de repintado.

Después de la ejecución del algoritmo, el resultado de la restauración se puede observar en la Figura 5.34 (a), (b) y (c) para diferentes tamaños de parche. La imagen no muestra rastros evidentes de la restauración que tuvo lugar en la imagen.

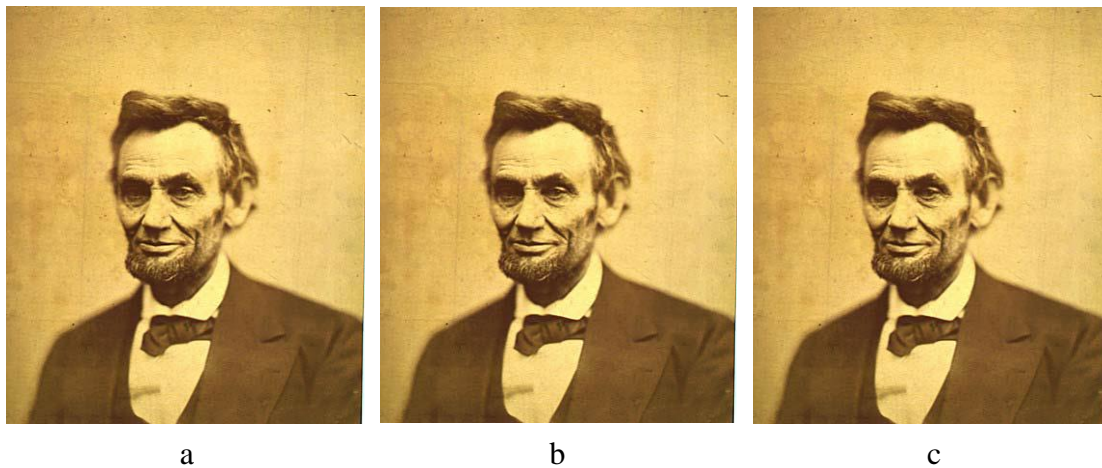


Figura 5.34 Imagen de Lincoln restaurada con parches (a) 3x3 (b) 5x5 (c) 9x9 píxeles.

5.3.3 Imagen endoscópica

La imagen presenta una serie de reflejos especulares causados por el medio de grabación. En total se tiene un 3.65% de daño después de marcar la imagen. Figura 5.35.

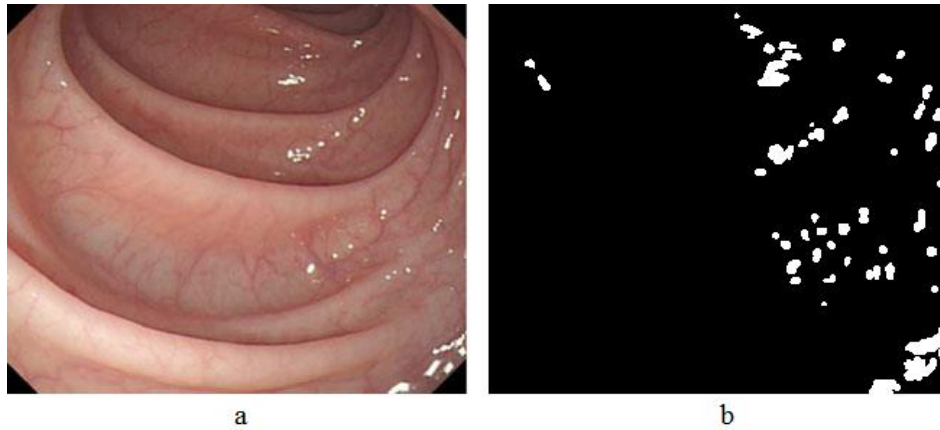


Figura 5.35 (a) Imagen endoscópica (b) Máscara de repintado.

El resultado de la restauración se puede observar en la Figura 5.36 (a), (b) y (c) para diferentes tamaños de parche. Los reflejos especulares son sustituidos por su información circundante en la imagen.

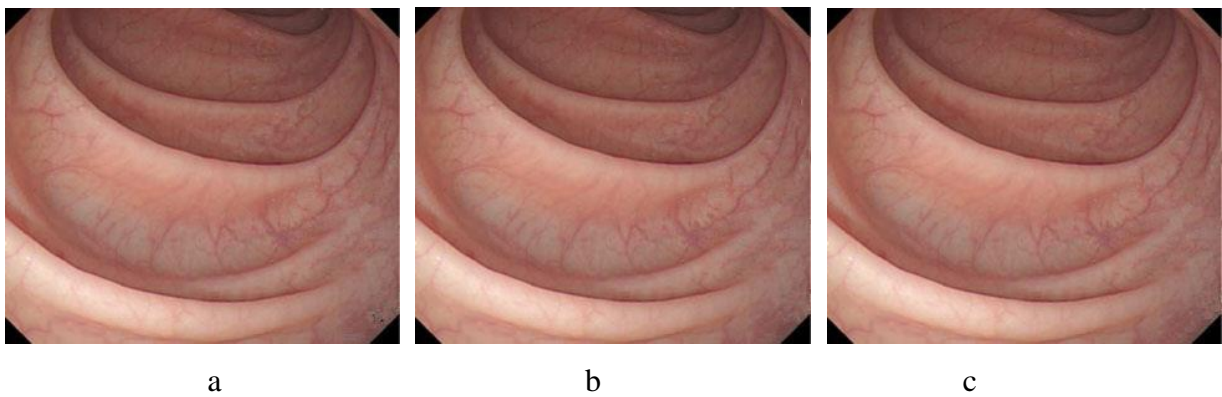


Figura 5.36 Imagen endoscópica restaurada con parches (a) 3x3 (b) 5x5 (c) 9x9 píxeles.

5.3.4 Cebra

La imagen presenta una cebra detrás de una reja marcada en color rojo. En total se tiene un 16.199% de daño, una vez que fue creada la zona de repintado. Figura 5.37.



Figura 5.37 (a) Imagen de una cebra (b) Máscara de repintado.

El resultado de la restauración se puede observar en la Figura 5.38 (a), (b), (c) y (d) para diferentes tamaños de parche. La restauración de los rasgos de la cebra presenta resultados visualmente aceptables aunque algunos detalles son perceptibles. A medida que se incrementa el porcentaje de daño en la imagen, la calidad de la restauración disminuye al contar con menos información para sintetizar.

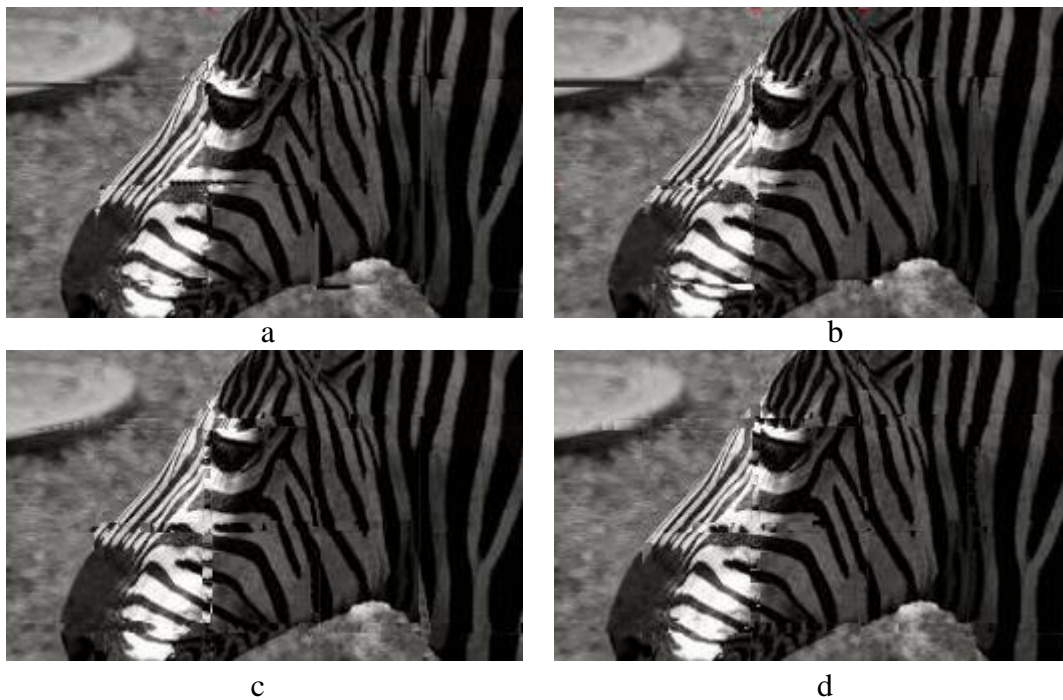


Figura 5.38 Imagen de la cebra una vez restaurada con parches (a) 3x3 (b) 5x5 (c) 9x9 (d) 11x11 píxeles.

Debido a la cantidad de detalle en la imagen de la cebra se incluye el resultado con un tamaño de parche de 11x11, esto para permitir al algoritmo sintetizar la mayor parte de las texturas presentes en la imagen. De esta forma, se obtienen mejores resultados visualmente al ser menos imperceptible la restauración efectuada sobre la imagen.

5.3.5 Lena

La imagen de Lena está marcada con 4 formas iguales repartidas en cada esquina; en total se tiene un 4.80% de daño de la imagen una vez que ha sido marcada. Figura 5.39.



Figura 5.39 (a) Imagen de Lena (b) Máscara de repintado.

El resultado de la restauración se puede observar en la Figura 5.40.





c

Figura 5.40 Imagen de Lena restaurada con parche (a) 3x3 (b) 5x5 (c) 9x9.

Visualmente el resultado con el parche de tamaño 9x9 provee mejores resultados, pues se redujo el número de píxeles con cambios bruscos de tonalidad en la parte superior derecha del sombrero de Lena. De esta forma se nota que el parche de tamaño 9x9 sintetiza adecuadamente la textura para el repintado.

Capítulo 6 Conclusiones y trabajo futuro

6.1 - Conclusiones

6.2 – Recomendaciones

6.3 - Trabajo futuro

En este capítulo se exponen las principales conclusiones del trabajo de tesis y se proponen algunos trabajos futuros que podrían desarrollarse como una continuación lógica del mismo con el fin de complementarlo o mejorarlo.

6.1 Conclusiones

En este trabajo se ha demostrado que la hipótesis planteada de que mediante técnicas de repintado es posible efectuar la restauración de imágenes digitales en colores, de forma semiautomática, sin exigirse que sean de un tipo específico.

Con respecto al cumplimiento de los objetivos particulares, podemos concluir lo siguiente:

- Generación de la máscara de repintado a partir de la selección hecha por el usuario: Se creó el módulo de creación de la máscara de repintado para marcar la zona a restaurar en la imagen.
- Rellenado de las zonas marcadas por la máscara mediante los algoritmos de restauración propuestos: Se aplicaron los algoritmos de repintado propuestos para efectuar la restauración de imágenes sintéticas y de escenas reales.
- Diseñar y desarrollar una interfaz gráfica que permita manipular fácil y confiablemente la metodología propuesta: Se diseñaron e integraron los módulos de máscara de repintado y de restauración dentro del sistema de procesamiento de imágenes administrado por el Dr. Edgardo Manuel Felipe Riverón.
- Evaluación de los resultados: Se evaluó el porcentaje de exactitud en la restauración para el caso de imágenes sintéticas y la calidad de la restauración mediante el índice SSIM como suele utilizarse en la literatura para el caso de imágenes de escenas reales.

El desarrollo de los módulos de restauración y generación de las máscaras de repintado nos permitió entregar dentro de una aplicación de enseñanza en el procesamiento digital de imágenes, el resultado de la presente investigación.

En promedio, el tiempo de ejecución del algoritmo para imágenes de tamaño 512 x 512 píxeles fue de uno a dos minutos cuando la imagen tiene un porcentaje de daño de entre 3% y 5%, que es menor comparándolo con algunos algoritmos del estado del arte, ya que pueden tardar 30 minutos o más a causa de lo exhaustivo de las búsquedas. Con el uso de una ventana de búsqueda de tamaño fijo se logró encontrar un equilibrio entre la calidad de la restauración y el tiempo de ejecución.

Debido a la cantidad de tiempo empleada en evaluar si los parámetros reportados en el estado del arte respondían adecuadamente a la configuración de los algoritmos propuestos, no nos fue posible proponer un mecanismo para que el tamaño de la ventana de búsqueda y del parche cambiara dinámicamente de acuerdo al progreso en cada iteración del algoritmo.

6.2 Recomendaciones

A partir de la experiencia obtenida durante el desarrollo de los algoritmos, se hacen las siguientes recomendaciones para su mejoramiento:

- Emplear adicionalmente a los bordes, para el cálculo de la prioridad de repintado, otras características de la imagen como el análisis de esquinas.
- Proponer nuevos modelos para el cálculo de la prioridad, de forma que no sólo dependan del gradiente de la imagen.
- Explorar el uso de técnicas de preprocesamiento para eliminar el ruido aditivo o algunos otros artefactos que puedan intervenir desfavorablemente en la búsqueda del parche más similar.
- Mejorar el algoritmo de búsqueda del parche más similar a través de otras técnicas de similitud, como por ejemplo, el análisis de la similitud de documentos en el procesamiento del lenguaje natural.

6.3 Trabajo futuro

Con el fin de mejorar los resultados del algoritmo, se proponen los siguientes trabajos futuros.

- Implementar las cuestiones expuestas en las recomendaciones (sección 6.2).
- Aplicar las optimizaciones necesarias al código fuente del sistema, de manera tal que se aprovechen las capacidades de cómputo actuales, o bien migrar el código a un lenguaje de programación más actual como *Python* que ya cuenta con bibliotecas de procesamiento de imágenes.
- Emplear técnicas de aprendizaje que permitan mejorar la síntesis de textura durante el repintado.
- Emplear técnicas de agrupamiento para agrupar los parches más similares y reducir los tiempos de búsqueda del parche más similar, al buscar sólo en el grupo de parches potencialmente más parecidos.
- Proponer algoritmos de análisis que de forma automática detecten grietas o algún tipo de degradación específica en la imagen y que posteriormente efectúen el repintado sin la intervención del usuario.

Referencias

- Anupam, Goyal, P., & Diwakar, S. (2010). Fast and enhanced algorithm for exemplar based image inpainting. *Fourth Pacific-Rim Symposium on Image and Video Technology*, 325 - 330.
- Bertalmio, M., Bertozzi, A., & Sapiro, G. (2001). Navier-stokes, fluid dynamics, and image and video inpainting. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1*, I-355 - I-362.
- Bertalmio, M., Caselles, V., Masnou, S., & Sapiro, G. (2011). Inpainting. En *Encyclopedia of Computer Vision*. Berlín: Springer.
- Bertalmio, M., Sapiro, G., Caselles, V., & Ballester, C. (2000). Image Inpainting. *Proceedings of SIGGRAPH 2000*, 417-424.
- Cheng, W., Hsieh, C., Lin, S., Wang, C., & Wu, J. (2005). Robust algorithm for exemplar-based image inpainting. *Processing of International Conference on Computer Graphics, Imaging and Visualization*, 64-69.
- Criminisi, A., Pérez, P., & Toyama, K. (2004). Region filling and object removal by exemplar-based inpainting. *IEEE Transactions in Image Processing*, 13(9), 1200-1212.
- Dang, T. T., Larabi, M., & Beghdadi, A. (2012). Multi-resolution patch and window-based priority for digital image inpainting problem. *3rd International Conference on Image Processing Theory, Tools and Applications*, 280-284.
- Efros, A., & Leung, T. (1999). Texture synthesis by non-parametric sampling. *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, 2, 1033-1038.
- Ghorai, M., & Chanda, B. (2015). An image inpainting method using pLSA-based search space estimation. *Machine Vision and Applications*, 26(1), 69-87.
- Gonzalez, R., & Woods, R. (2008). *Digital Image Processing* (3era ed.). Pearson Prentice Hall.
- Goswami, P., & Paunwala, C. (2014). Exemplar-based Image Inpainting Using ISEF for Priority Computation. *IEEE Int. Conf. on Circuits, Systems, Communication and Information Technology Applications*, 75-80.
- Guillemot, C., & Le Meur, O. (2014). Image inpainting: overview and recent advances. *IEEE signal processing magazine*, 127-144.

- Guo, H., Ono, N., & Sagayama, S. (2008). A structure-synthesis image inpainting algorithm based on morphological erosion operation. *CISP'08. Congress on Image and Signal Processing*, 3, 530-535.
- Le Meur, O., Ebdelli, M., & Guillemot, C. (2013). Hierarchical super-resolution-based inpainting. *IEEE Transactions on Image Processing*, 22(10), 3779-3790.
- Lingni, M., Luat, D., & de With, P. (2012). Fsats and improved exemplar-based inpainting techniques for natural images. *Proceedings of the 33rd WIC Symposium on information theory*, 13-19.
- Masnou, S., & Morel, J. (1998). Level lines based disocclusion. *International Conference on Image Processing, III*, 259-263.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1, 62-66.
- Pei, S.-C., Zeng, Y.-C., & Chang, C. (2004). Virtual restoration of ancient Chinese paintings using color contrast enhancement and lacuna texture synthesis. *IEEE Trans. Image Processing*, 13(3), 416-429.
- Plataniotis, K., & Venetsanopoulos, A. (2000). *Color Image Processing and Applications* (1a ed.). Springer.
- Ruzic, T., Cornelis, B., Platasa, L., Pizurica, A., Doms, A., Philips, W., y otros. (2011). Virtual restoration of the Ghent altarpiece using crack detection and inpainting. *Advanced Concepts for Intelligent Vision Systems*, 6915, 417-428.
- Soille, P. (2003). *Morphological Image Analysis - Principles and Applications* (2da ed.). Springer.
- Wu, J., & Ruan, Q. (2006). Object Removal By Cross Isophotes Exemplar-based Inpainting. *18th International Conference on Pattern Recognition*, 3, 810-813 .
- Wyszecki, G., & Stiles, W. (1982). *Color Science, Concepts and Methods, Quantitative Data and Formulas* (2a ed.). John Wiley.
- Xu, Z., & Sun, J. (2010). Image Inpainting by Patch Propagation Using Patch Sparsity. *IEEE Transactions on Image Processing*, 19(5), 1153-1165.
- Zhang, Q., & Lin, J. (2012). Exemplar-Based image inpainting using color distribution analysis. *Journal of Information Science and Engineering* 28, 641-654.