ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Informe de Materia de Graduación

"Implementación de un algoritmo para el cálculo estimado del movimiento lineal en imágenes y su aplicación en la restauración de imágenes movidas"

Previa a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

José Luis Masache Narváez

Ernesto Javier Roldán Monge

GUAYAQUIL – ECUADOR
AÑO
2010

A G R A D E C I M I E N T O S

En primer lugar, agradezco a Dios por su presencia e inspiración desde el inicio hasta el final de este proyecto, a mi Papá y a mi Mamá por ser modelos de ética y responsabilidad, además de recibir su incondicional apoyo durante toda mi carrera.

A los Ingenieros que fueron clave en mi formación profesional. A la Ing. Patricia Chávez Burbano, por su consideración, asistencia, y paciencia durante la totalidad del proceso de la materia de graduación.

José Luis Masache.

Agradezco a mis padres Joffre y Leonor quienes con su sabiduría me han guiado por el camino correcto, gracias por su confianza y apoyo incondicional durante toda mi carrera. A mi tía Betty por sus consejos y apoyo durante toda mi vida estudiantil.

A la Ing. Patricia Chávez, por su paciencia y constancia para la culminación del presente trabajo investigativo.

Ernesto Roldán Monge.

D E D I C A T O R I A S

A Dios,

a la Madre Dolorosa.

A mi familia,

y a mis amigos.

José Luis Masache.

A mis Padres,

a mi tía,

a mis hermanos

y amigos.

Ernesto Roldán Monge.

TRIBUNAL DE SUSTENTACIÓN

Ing. Juan Carlos Avilés

DELEGADO DECANO FIEC

Ing. Patricia Chávez

PROFESORA MATERIA DE GRADUACION

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este Proyecto de Grado, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

José Masache Narváez.

Ernesto Roldán Monge.

RESUMEN

Nuestro objetivo principal durante el desarrollo de este proyecto es la restauración total o parcial de una imagen que previamente ha sido distorsionada linealmente, debido a que es prácticamente imposible obtener una restauración completa consideraremos un resultado como satisfactorio en el cual se pueda reconocer al objeto o frente de imagen, que obviamente fue la razón por la que se tomó la fotografía.

Utilizaremos métodos propuestos por Matlab, pero nuestro reto será encontrar una PSF (Point Spread Funtion) aceptable con la cual podamos hacer el proceso de deconvolución y así restaurar la imagen afectada.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En el procesamiento digital de señales es muy común encontrarnos con problemas como el ruido (en el caso de señales de audio o video) o la distorsión (en el caso de imágenes). Existe un caso particular denominado distorsión lineal, que es básicamente la distorsión provocada por el movimiento relativo ya sea del objeto a fotografiarse o de la cámara en sí. Es necesario entender que una fotografía no representa un instante sino un periodo de tiempo, restringido por las características tecnológicas de la cámara.

Se han propuesto varios algoritmos que restauran imágenes afectadas por distorsión lineal que se basan en diferentes métodos, este proyecto se enfoca en el método de restauración propuesto en [1], en el cual se analiza como la degradación de la imagen y la transparencia están relacionadas, además se prueba que el filtro de distorsión lineal puede ser determinado por la transparencia existente en los bordes del objeto afectado.

CAPÍTULO I

FUNDAMENTOS TEÓRICOS

El fenómeno de distorsión lineal es usualmente modelado como un proceso de la siguiente manera:

$$I = L \otimes f + n \quad (1)$$

En donde:

I : Imagen degradada

L : Imagen sin degradar

n : Ruido aditivo

f : PSF

$\otimes$: Convolución

Basado en este modelo, uno de los principales problemas al tratar de restaurar la imagen degradada es la estimación del filtro de distorsión lineal o PSF (Point Spread Funtion) ya que la calidad de la imagen restaurada dependerá directamente del grado de aproximación de la PSF estimada con respecto a la PSF original.

Figura 1.1 Típico caso de distorsión lineal.

A lo que se denomina Transparencia es justamente la distorsión producida en los bordes del objeto durante el mínimo tiempo que transcurre al presionar y soltar el botón de la cámara.

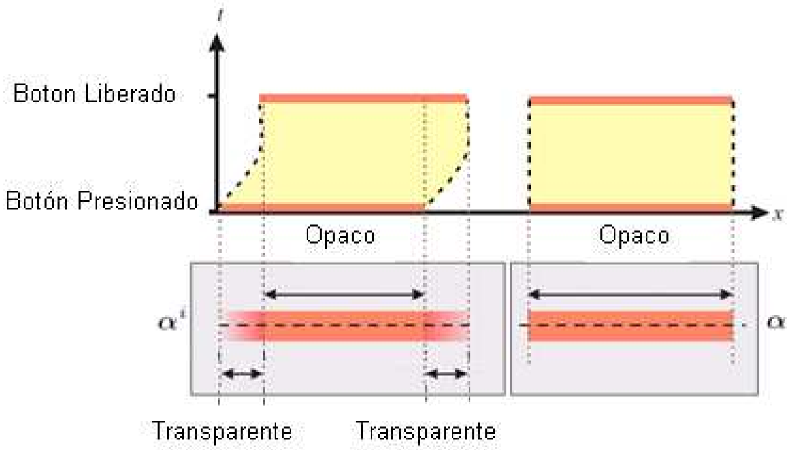


Figura 1.2 tiempo vs espacio (apreciación del fenómeno de Transparencia) [1].

En la figura 1.2 se muestra un objeto solido y opaco (la barra roja), el cual se mueve horizontalmente. El eje horizontal es la dirección de movimiento y el eje vertical es el tiempo, en el cual se muestran dos instantes: el tiempo en el que se presiona el botón de la cámara fotográfica y el tiempo en el cual el botón se suelta.

En la figura 1.2, del lado izquierdo, el objeto opaco se mueve durante la captura de la imagen, se muestra la trayectoria de este por medio de una línea punteada. Se puede notar que durante la captura de la imagen los bordes de la barra se mezclan con el fondo de la imagen, y es justo en este punto en el cual se produce la transparencia [1].

La imagen a analizar I, se asume como una combinación del frente F y el fondo B de la imagen. El color del i-esimo pixel, es asumido como una combinación lineal de los correspondientes colores del frente y el fondo de la imagen [2],

$$I_i = \alpha_i F_i + (1 - \alpha_i) B_i, \quad (2)$$

donde α_i es un coeficiente que representa el grado de opacidad del frente de la imagen para el i-esimo pixel.

Figura 1.3 Imagen como combinación de frente (F) y fondo (B).

Para el análisis de distorsión se va a denotar los valores alfa de la imagen distorsionada como α^i, donde $0 \leq \alpha^i \leq 1.$ El objeto opaco sin distorsión, oculta completamente el fondo de la imagen, produciendo valores binarios de alfa α^0, donde $\alpha^0 \in \{0,1\}.$

La estructura de los valores alfa en la porción de la imagen distorsionada es una mezcla del fondo ($\alpha^0 = 0$) y el frente ($\alpha^0 = 1$) de la imagen. Dicho de otra forma, la Transparencia está determinada por la PSF y el borde del objeto no distorsionado [1].

Desde el punto de vista de la Transparencia tenemos que:

$$\alpha^i = \alpha^0 \otimes f \text{ (3)}$$

Donde f es el filtro de distorsión y que es usualmente denotado como un vector de (2n+1) por 1 en el mismo sentido de la distorsión.

$$f = [f_{-n}, \dots, f_0, \dots, f_n]^T \text{ (4)}$$

Donde f_{-n} y f_n son diferentes de cero.

Desde este punto vamos a asumir que existe un solo objeto distorsionado en el centro de la imagen de entrada y cuyo ancho es mayor que el tamaño del filtro de distorsión lineal 2n+1 [1].

1.1 **Lema:** Se define a α^i y α^0 como los valores alfa en una línea a lo largo de la dirección de distorsión respectivamente de la siguiente manera [1]:

$$\alpha^i = [\alpha^i{}_0, \alpha^i{}_1, \dots, \alpha^i{}_{m-1}]^T$$

$$\alpha^0 = [\alpha^0{}_0, \alpha^0{}_1, \dots, \alpha^0{}_{m-1}]^T$$

Donde $\alpha^i{}_0$ y $\alpha^i{}_{m-1}$ son el primer y último alpha diferentes de cero a lo largo de la dirección de distorsión y m es el número de pixeles en α^i.

Entonces α^0 puede ser expresado como:

$$\alpha^0 = [0, \dots, 0, 1, \dots, 1, 0, \dots, 0]^T$$

Tanto los primeros como los últimos valores de alfa son próximos a cero debido a que son los que sufrieron la distorsión.

1.2 Proposición: Denotar el pixel que se encuentra más a la derecha con valor de alfa $\alpha_i \neq 0$ y el pixel que se encuentra más a la derecha con $\alpha_i = 1$, en el mapa de transparencia distorsionado como (x_{r1}, y_{r1}) y (x_{r2}, y_{r2}) respectivamente.

Entonces $x_{r1} - x_{r2} + 1$ es un límite superior para el ancho del filtro de distorsión lineal [1].

1.3 Teorema: En una imagen distorsionada por distorsión lineal en una dimensión, cada elemento del filtro de distorsión lineal puede ser explícitamente expresado como [1]:

$$f_j = \begin{cases} \alpha^i_{j+n} - \alpha^i_{j+n-1} & -n < j \leq n \\ \alpha^i_0 & j = -n \end{cases} \quad (5)$$

dada la definición de α^i y α^0 en el Lema.

1.4 Filtro Wiener

El Filtro Wiener (cuyo nombre es debido a N. Wiener quien propuso este método en 1942) es uno de los más recientes y mejores métodos que aproxima la restauración lineal de imágenes. Un filtro Wiener busca y estima $\hat{f}$ que minimiza la función estadística de error:

$$e^2 = E\left\{\left(f - \hat{f}\right)^2\right\} \quad (6)$$

Donde E es el valor esperado y f es la imagen no degradada. La solución a esta expresión en el dominio de la frecuencia es:

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{S_n(u,v)}{S_f(u,v)}}\right] G(u,v) \quad (7)$$

Donde:

$H(u,v)$= Función de Degradación.

$|H(u,v)|^2 = H^*(u,v)H(u,v).$

$H^*(u,v)$= El complejo conjugado de $H(u,v).$

$S_n(u,v)=|N(u,v)|^2$ = La densidad espectral de potencia del ruido.

$S_f(u,v)=|F(u,v)|^2$ = La densidad espectral de potencia de la imagen no degradada.

Este cociente $\frac{S_n(u,v)}{S_f(u,v)}$ es denominado Relación Señal a Ruido [4].

1.5 Restauración Iterativa No-Lineal usando el algoritmo Lucy-Richardson

Durante las dos décadas anteriores, técnicas iterativas no-lineales han ganado aceptación como herramientas de restauración que a menudo no brindan un resultado que los obtenidos con métodos lineales. Las principales desventajas de los métodos no-lineales son que su comportamiento no es siempre predecible y generalmente requieren recursos computacionales considerables. La primera desventaja a menudo pierde importancia basado en que los métodos no-lineales son superiores a los métodos lineales en un amplia gama de aplicaciones. La segunda desventaja pierde importancia debido al gran desarrollo computacional en la última década. El método no-lineal que analizaremos es una técnica desarrollada por Richardson (1972) y por Lucy (1974), trabajando independientemente.

El algoritmo L-R surge de una formulación de máxima probabilidad en la cual la imagen es modelada estadísticamente aproximándola a una Poisson. Maximizando la función de probabilidad del modelo, obtenemos como resultado una ecuación que se cumple cuando la siguiente iteración converge:

$$\hat{f}_{k+1}(x,y) = \hat{f}_k(x,y)\left[h(-x,-y) * \frac{g(x,y)}{h(x,y)*\hat{f}_k(x,y)}\right] \quad (8)$$

Como ya lo hemos dicho, "*" indica convolución, $\hat{f}$ es el estimado de la imagen no degradada, y $g(x,y)$ como $h(x,y)$ son la imagen degradada y la imagen de entrada respectivamente. La naturaleza iterativa del algoritmo es evidente. Su naturaleza no-lineal surge de la división por $\hat{f}$ en el lado derecho de la ecuación [4].

1.6 Algoritmos de deconvolución sin referencia

Los métodos de restauración de imágenes que no están basados en un conocimiento específico de la PSF son llamados algoritmos de deconvolución sin referencia.

Un enfoque que ha recibido especial atención en las últimas dos décadas es el basado en la estimación de máxima probabilidad (MLE por sus siglas en ingles), una estrategia de optimización usada para obtener estimaciones de cantidades corrompidas por ruido aleatorio. En resumen, una forma de interpretar la MLE es pensar en la imagen como cantidades aleatorias que tienen una gran probabilidad de ser producidas por una familia de otras posibles cantidades aleatorias. La función de probabilidad es expresada entonces en función de la imagen degradada, la imagen original y la función de degradación de la imagen, y el problema entonces es hallar el máximo de la función de probabilidad. En la deconvolución sin referencia el problema de optimización es resolver de una manera iterativa con limitaciones especificas y asumiendo convergencia, la función de degradación especifica y la imagen original que resulta en un máximo, están son entonces la imagen restaurada y la PSF [4].

CAPÍTULO II

IMPLEMENTACIÓN

2.1 Software Utilizado

Para el desarrollo de nuestro proyecto vamos a utilizar el software matemático MATLAB 2007ª (Matrix Laboratory), que por su variado contenido de funciones especializadas en la manipulación de matrices nos será de mucha utilidad debido a que una imagen dentro de MATLAB es simplemente una matriz.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets). [5]

2.2 Distorsión de la imagen.

Nuestro análisis comienza con una imagen original que será distorsionada al convolucionarla con la PSF, la cual es modelada con la función de Matlab llamada `fspecial`:

```
PSF=fspecial('motion',len,theta)
```

Esta función retorna una PSF que aproxima el efecto del movimiento lineal de una cámara en “len” pixeles. El parámetro “theta” esta en grados, medido con respecto al eje horizontal positivo en contra de las manecillas del reloj. El valor por defecto de “len” es 9 y el valor por defecto de “theta” es 0, los cuales corresponden al movimiento de 9 pixeles en la dirección horizontal [4]. En nuestro caso el número de pixeles varia, y el ángulo de distorsión se fija en cero grados para simular el movimiento horizontal.

Además también se usa la función `imfilter` para crear la imagen degradada con la PSF creada mediante la función descrita anteriormente.

```
g=imfilter(f,PSF,filtering_mode)
```

En “`filtering_mode`” se usa “`conv`” para realizar el filtrado usando convolución.

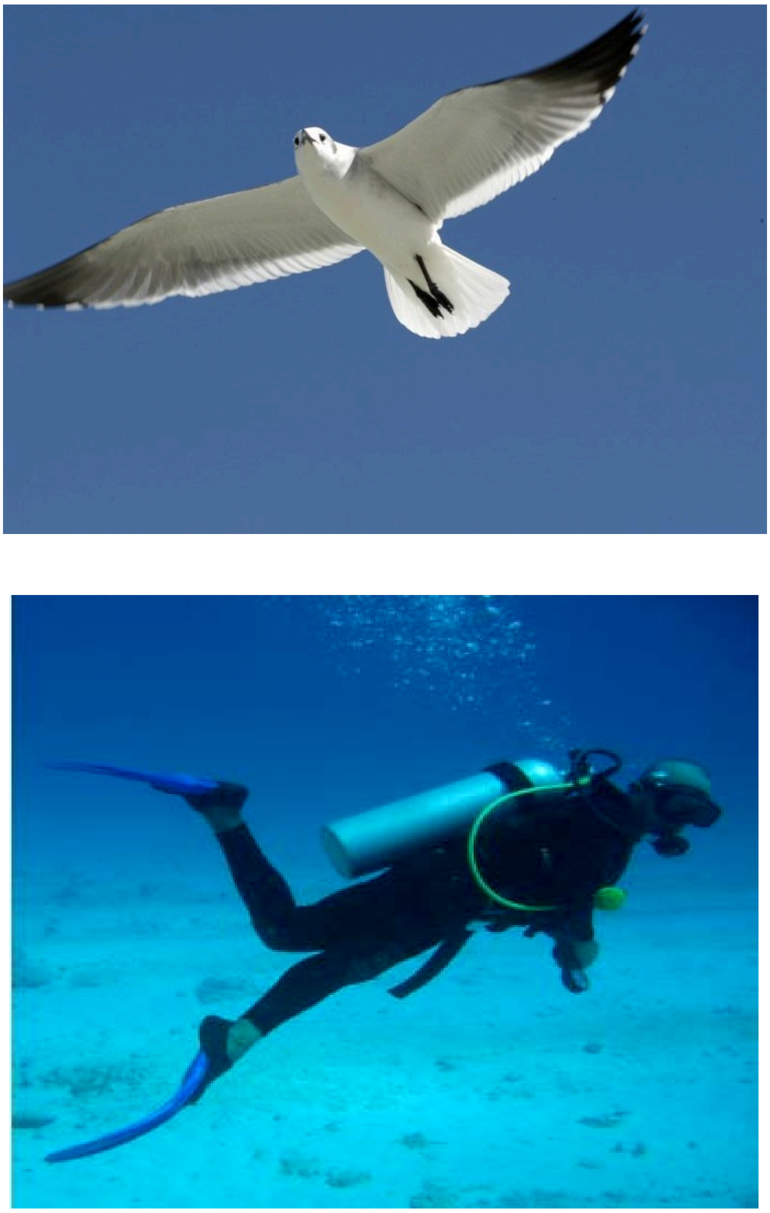


Figura 2.1 Imágenes sin distorsión.

Figura 2.2 Imágenes distorsionadas.

2.3 Proceso de obtención de la Transparencia

Para determinar la transparencia de una imagen existen varios métodos y algoritmos, para este proyecto se utilizo el algoritmo encontrado en [3], el cual se basa en el método propuesto en [2]. Este algoritmo se escogió debido a que se obtienen resultados satisfactorios con poca intervención del usuario.

El algoritmo mencionado se encarga de diferenciar claramente el fondo del frente de la imagen, dejando el fondo de la imagen en color negro y el frente de color blanco. Los bordes de la imagen en los cuales se evidencia la distorsión quedan como en escala de grises ya que son una mezcla de frente y fondo de la imagen.

La calidad de la transparencia resultante es directamente proporcional a la exactitud con la que definamos los trazos, los cuales deben ser ingresados por el usuario y son guías para que el programa pueda diferenciar el fondo y el frente de la imagen.

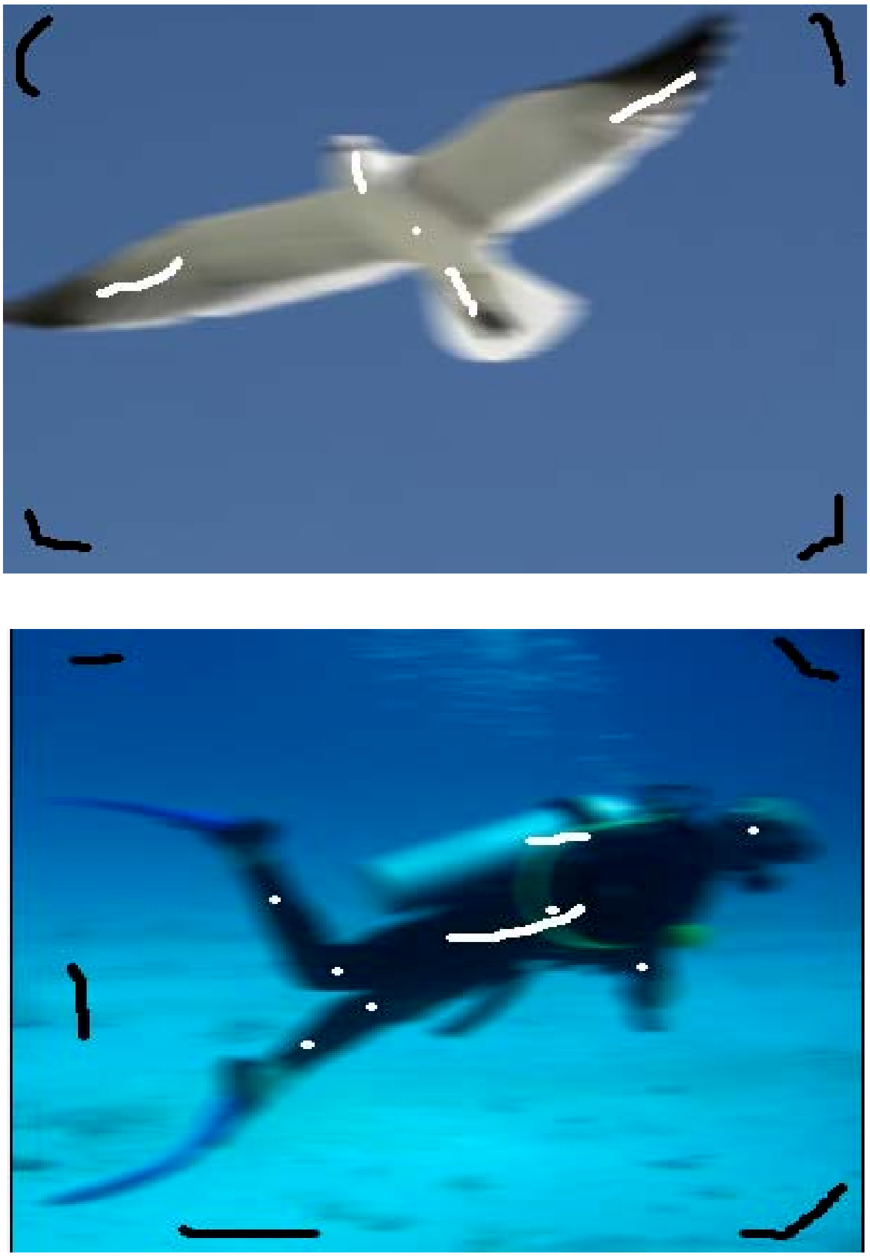


Figura 2.3 Trazos

Figura 2.4 Mapa de transparencia resultante.

2.4 Estimación del Tamaño de la PSF.

Basados en la proposición mencionada en el marco teórico se realizo una función en MATLAB que estima el tamaño del filtro de distorsión, el cual en otros métodos tiene que ser estimado por el usuario.

La función recibe como entradas el mapa de transparencia obtenido previamente y los valores de los límites para los cuales se considera que la porción de la imagen es fondo o frente.

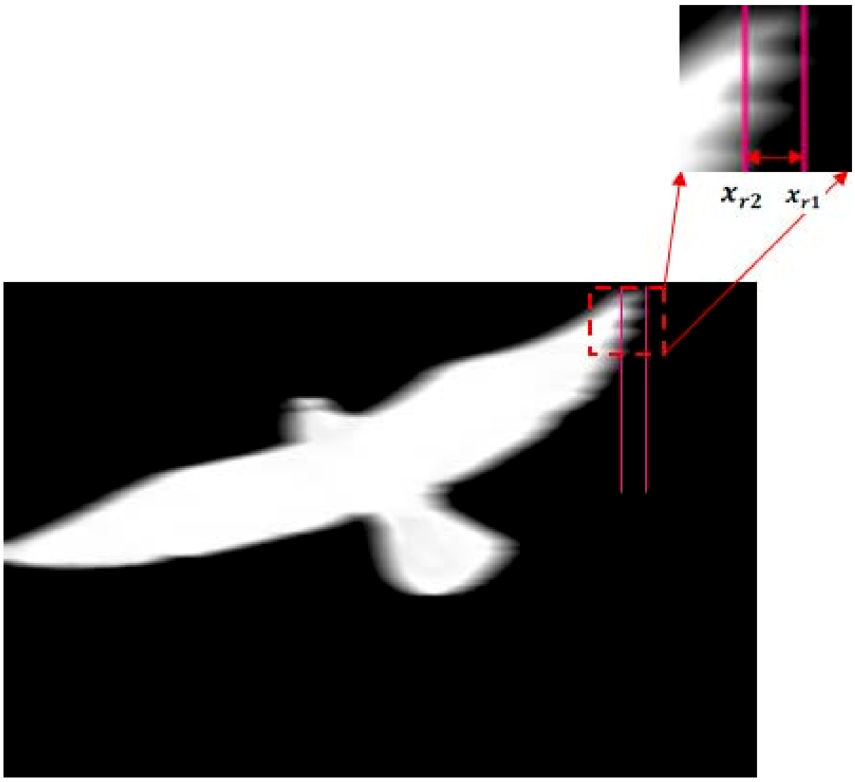


Figura 2.5 Acercamiento del mapa de transparencia resultante.

2.5 Cálculo de la PSF.

Basados en (5) se desarrollo una función que calcula la PSF que será utilizada para restaurar la imagen.

La función calcula del mapa de transparencia determinado anteriormente una PSF de cada línea y luego saca un promedio de todas las PSFs calculadas, esta PSF promedio es el resultado de la función, y la cual se utilizara posteriormente para realizar la restauración.

2.6 Restauración de la Imagen.

Este es básicamente el proceso de deconvolución en el cual utilizando funciones propias de MATLAB restauramos la imagen distorsionada.

Para un mejor análisis se lo realizo por tres métodos diferentes, dos de los cuales utilizan la PSF calculada por nosotros y el método restante utiliza una PSF calculada por el propio MATLAB para así hacer una comparación y sacar nuestras propias conclusiones de cuál nos brinda un mejor resultado.

Los métodos utilizados son el Filtro Wiener, el Algoritmo Lucy - Richardson y la restauración usando Deconvolución sin referencia, esta última es la única de las tres que utilizará una PSF calculada directamente por MATLAB, las dos primeras necesitan como parámetro de la función la PSF promedio calculada anteriormente.

2.6.1 Función deconvwnr

El filtro Wiener es implementado en MATLAB usando la función `deconvwnr`.

```
fr= deconvwnr (g, PSF)
```

En donde, g denota la imagen degradada y fr es la imagen restaurada. En esta sintaxis se asume que la imagen degradada está libre de ruido y es la que se usa en este proyecto debido a que nuestro principal interés es analizar la distorsión lineal [4].

2.6.2 Función deconvlucy

El algoritmo Lucy – Richardson es implementado en MATLAB usando la función deconvlucy, la cual tiene la sintaxis básica:

```
fr= deconvlucy (g, PSF, NUMIT, DAMPAR, WEIGHT)
```

donde fr es la imagen restaurada, g la imagen degradada, PSF la función de distorsión lineal, NUMIT es el numero de iteraciones (10 por defecto) [4]. Para este proyecto se uso el parámetro NUMIT con un valor de 30 ya que es el que dio mejores resultados en las pruebas realizadas.

2.6.3 Función deconvblind

En MATLAB el algoritmo de deconvolución sin referencia se implementa mediante la función `deconvblind` el cual tiene la sintaxis básica:

```
[fr, PSFe]=deconvblind (g, INITPSF, NUMIT)
```

Donde g es la imagen degradada, INITPSF es una estimación inicial de la PSF, PSFe es la estimación final de la PSF para esta función, y fr es la imagen restaurada usando la PSF estimada [4], NUMIT es el número de iteraciones de la función, se uso este parámetro con un valor de 30.

Figura 2.6 Imágenes distorsionadas en MATLAB.

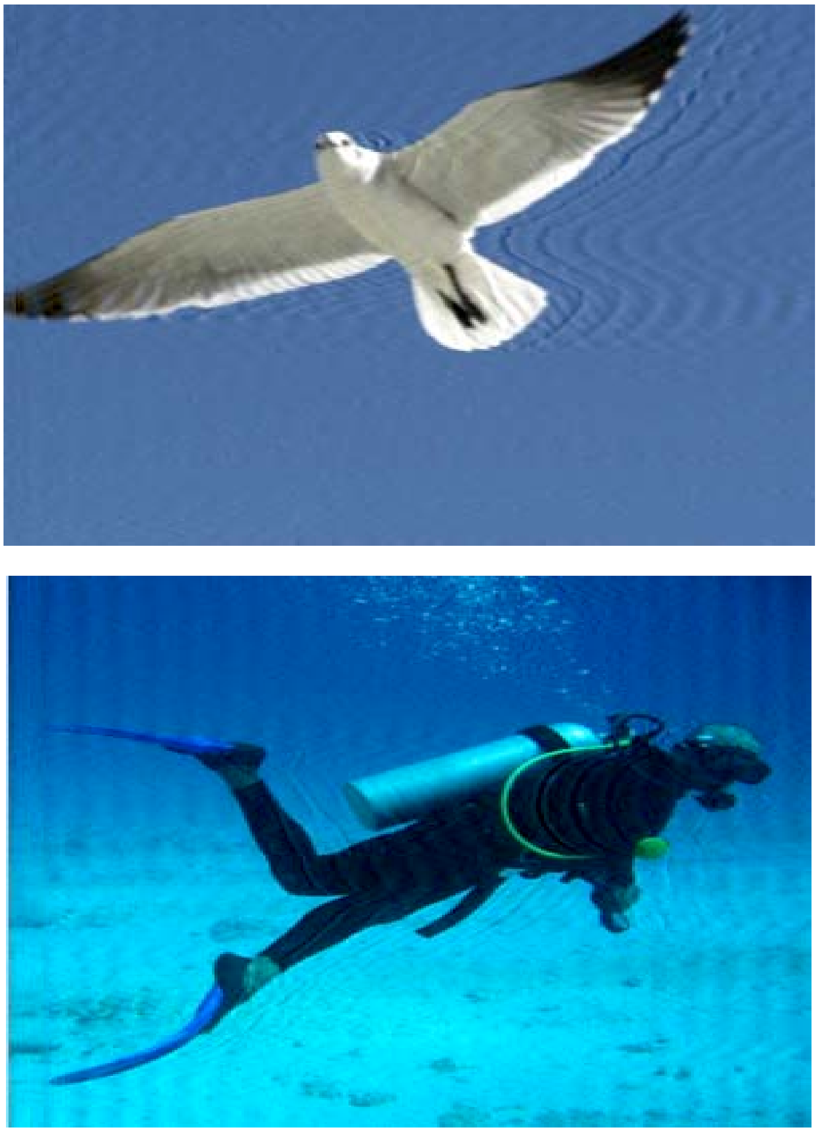


Figura 2.7 Imágenes restauradas usando Filtro Wiener.

Figura 2.8 Imágenes restauradas usando el Algoritmo L-R.

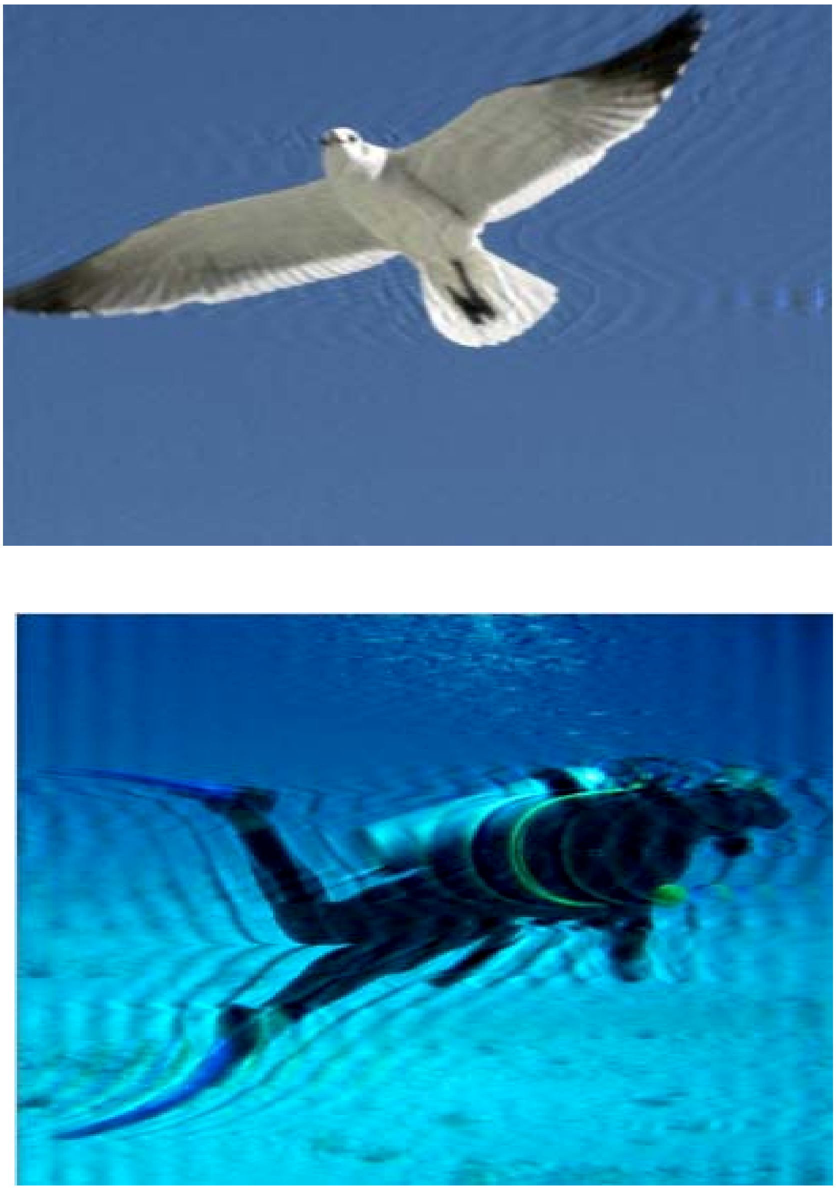


Figura 2.9 Imágenes restauradas usando Deconvolución sin referencia.

CAPÍTULO III

ANÁLISIS DE RESULTADOS

Para comprobar que tan eficiente es el algoritmo propuesto se realizo pruebas con 20 imágenes distintas, en donde se vario el numero de pixeles movidos desde 9 hasta 37.

Al apreciar las imágenes resultantes podemos observar que tenemos una restauración aceptable en todos los casos y podemos asegurar que en la mayor parte del total de imágenes restauradas al utilizar la PSF calculada por nosotros tenemos un mejor resultado.

Pero para no dejar limitado nuestro análisis a una apreciación visual se decidió hacer uso de un método estadístico llamado Error Cuadrático Medio o MSE (por sus siglas en inglés Mean Square Error).

A continuación se presenta una tabla demostrativa en la cual están detallados todos los Errores Cuadráticos Medios de la PSF calculada por nosotros y por Matlab, asimismo el MSE de las imágenes restauradas por los tres métodos mencionados previamente.

Tabla I: MSE de cada imagen restaurada.

	N usada	N estimada	MSE1	MSE2	MSE3	MSE4	MSE5
Imagen 1	9	8	0.000564	0.000194	7.88	5.26	32.81
Imagen 2	11	11	0.000120	0.000000	1.69	1.48	5.77
Imagen 3	11	10	0.000112	0.000093	10.85	3.76	20.90
Imagen 4	13	16	0.000087	0.000305	10.67	8.56	35.08
Imagen 5	15	17	0.000209	0.000059	14.67	11.08	24.49
Imagen 6	15	14	0.000174	0.000023	7.08	6.46	28.37
Imagen 7	17	19	0.000207	0.000042	27.87	18.73	44.33
Imagen 8	18	18	0.000384	0.000042	3.47	3.21	6.87
Imagen 9	19	18	0.000062	0.000010	6.70	5.1	12.17
Imagen 10	21	20	0.000052	0.000006	23.06	19.51	35.53
Imagen 11	21	26	0.000084	0.000078	45.19	36.69	65.80
Imagen 12	23	20	0.000046	0.000043	12.07	8.35	41.77
Imagen 13	23	24	0.000172	0.000004	14.02	9.26	19.85
Imagen 14	23	23	0.000066	0.000000	5.48	2.50	8.77
Imagen 15	25	37	0.000109	0.000172	82.41	77.23	96.89
Imagen 16	25	30	0.000330	0.000042	21.11	19.59	49.43
Imagen 17	27	28	0.000042	0.000002	65.02	60.82	16.53
Imagen 18	31	26	0.000093	0.000039	20.25	19.22	62.15
Imagen 19	35	35	0.000041	0.000000	38.12	33.43	43.95
Imagen 20	37	38	0.000021	0.000001	28.29	25.19	29.85

N usada= Tamaño de la PSF definida originalmente.

N estimada= Tamaño de la PSF calculada.

MSE1= MSE de la PSF estimada con respecto a la original.

MSE2= MSE de la PSF estimada con deconvolución sin referencia.

MSE3= MSE de la imagen restaurada con filtro wiener.

MSE4= MSE de la imagen restaurada con algoritmo L-R.

MSE5= MSE de la imagen restaurada con deconvolución sin referencia.

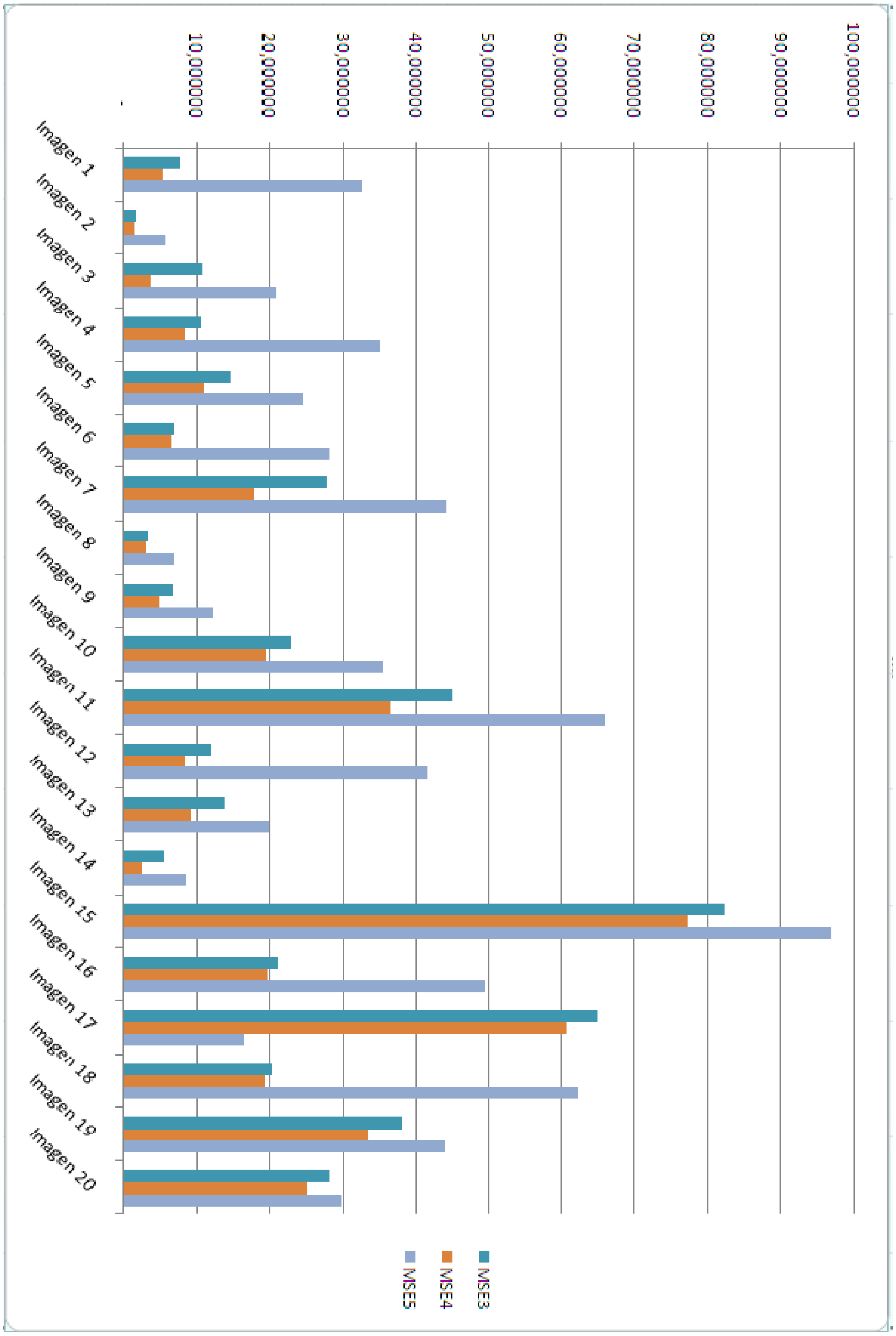


Figura 3.1 MSE de las imágenes restauradas por los 3 métodos mencionados con respecto a la imagen original.

Conclusiones

1. El software desarrollado es de utilidad para la restauración de imágenes en las cuales la finalidad de la restauración es recuperar en su mayor parte al objeto o frente de imagen que es lo que se ha logrado restaurar de una manera considerable, al punto de poder reconocer ya sea un rostro, un cuerpo o cualquiera que sea el objeto fotografiado.

2. La estimación del tamaño de la PSF está condicionada a que se haya hecho una buena determinación del mapa de transparencia.

3. La calidad de la imagen resultante está determinada por el correcto cálculo de la PSF debido a que la restauración de la imagen es básicamente una deconvolución de la imagen distorsionada con la PSF.

4. Debido a que para el cálculo del tamaño de la PSF debemos definir un rango de valores α^i para los cuales consideremos que el pixel ya esta distorsionado (recordando que $\alpha = 0$ es el fondo, $\alpha = 1$ es el objeto o frente de imagen y valores de

α entre 0 y 1 serían para un pixel distorsionado) un rango aceptable para este efecto es $0.05 \leq \alpha^i \leq 0.95$, dicho rango fue encontrado experimentalmente y funciono para imágenes para las cuales se definió un tamaño para la PSF menor a 25.

5. Según las encuestas realizadas un 53% de la muestra aleatoria de encuestados piensa que la restauración hecha por el algoritmo Lucy-Richardson es la mejor, un 38% piensa que lo es el Filtro Wiener y solo un 9% cree que usando Deconvolución sin referencia la imagen se regenera mejor, lo cual nos es muy satisfactorio debido a que los métodos que tienen mayor acogida son aquellos que utilizan la PSF calculada por nosotros.

Recomendaciones

1. Para el uso correcto del software implementado recomendamos hacer correctamente los trazos de diferenciación del fondo o background y el objeto o frente de imagen puesto que de esto depende mucho el resultado de la Transparencia y esta a su vez es de suma importancia para el resultado de la imagen restaurada.

2. El tamaño recomendable para las imágenes a procesar es de máximo 400 por 300, el procesamiento de imágenes más grandes usando MATLAB resulta imposible debido a las limitaciones de memoria. Si se quiere realizar el procesamiento de imágenes de un tamaño mayor al recomendable se debe reducir el tamaño de la imagen, esto se puede realizar mediante el programa MATLAB.

ANEXOS

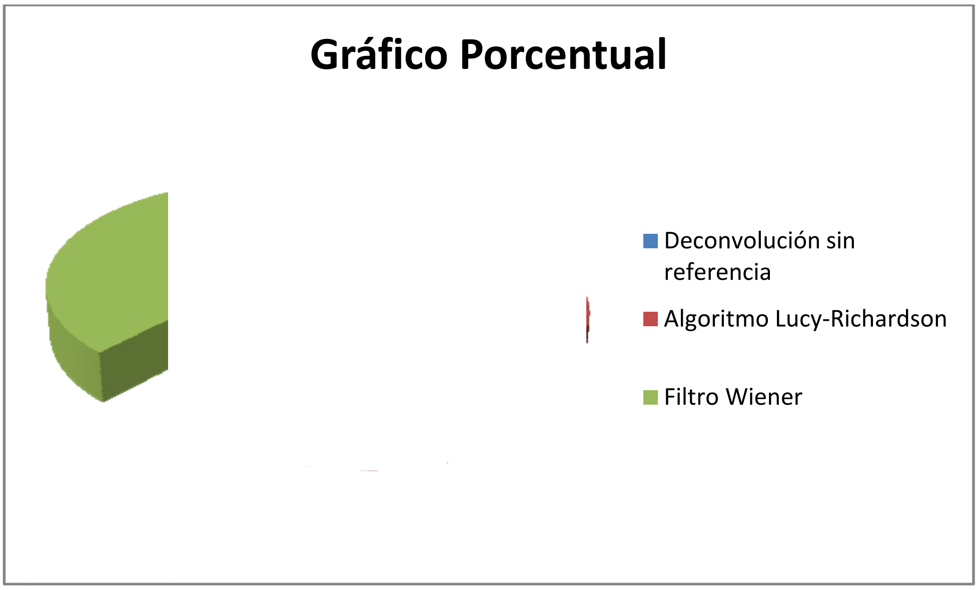
Anexo A
Tabla II Encuesta

Numero de Encuestados	Qué método de restauración cree usted que brinda un mejor resultado?		
	Filtro Wiener	Algoritmo Lucy - Richardson	Deconvolución sin referencia
1		X	
2	X		
3		X	
4	X		
5	X		
6		X	
7	X		
8		X	
9		X	
10		X	
11	X		
12			X
13	X		
14		X	
15	X		
16		X	
17	X		
18		X	
19		X	
20	X		
21	X		
22	X		
23		X	
24		X	
25		X	
26			X
27		X	
28	X		
29		X	
30			X
31		X	
32	X		
33		X	
34			X
35		X	
36	X		
37		X	
38	X		
39		X	
40		X	

Resultados:

Tabla III Porcentajes de la encuesta.

Tabla de Resultados	
Método	**Porcentajes**
Filtro Wiener	38%
Algoritmo Lucy-Richardson	53%
Deconvolución sin referencia	9%



Anexo B

Funciones creadas para el proyecto

function [n] = hallar_n (alpha, lim_sup, lim_inf);

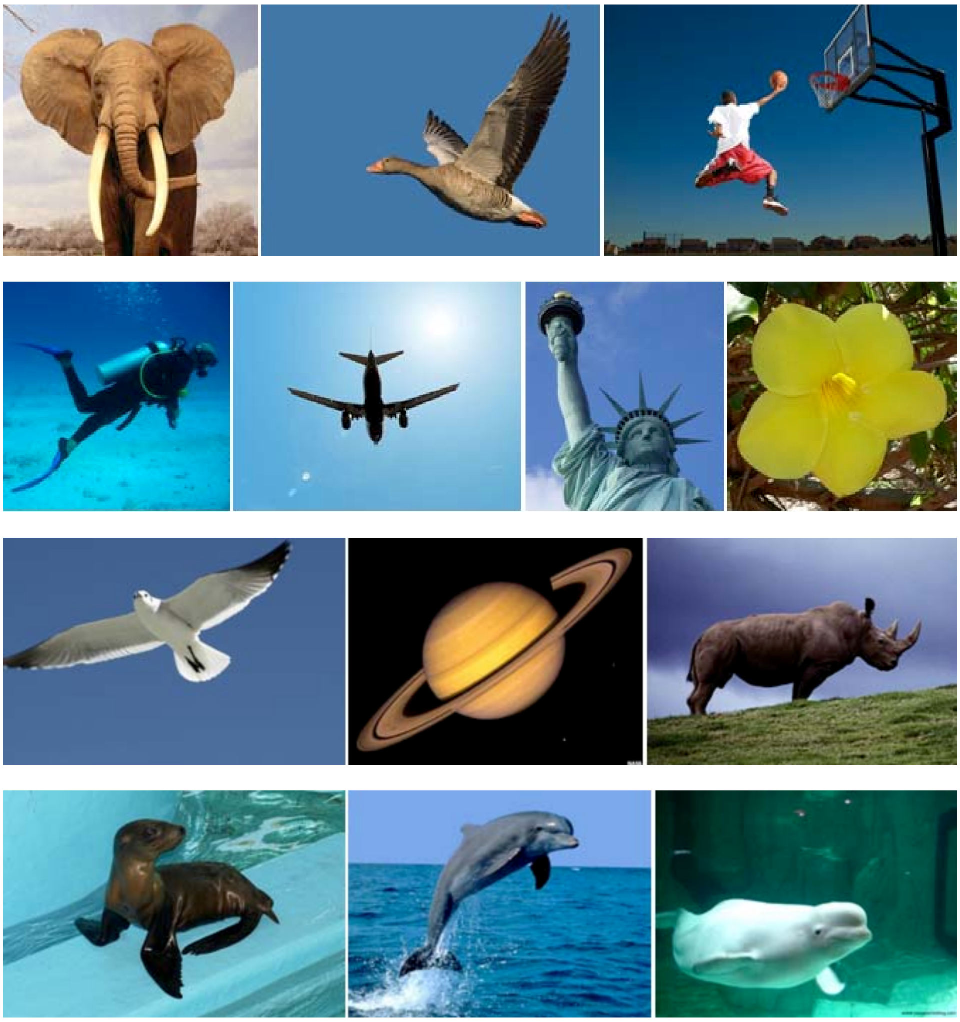
Esta función estima el número de pixeles que se han movido en la imagen. El parámetro “alpha” es el mapa de transparencia de la imagen el cual debe haber sido determinado previamente. El parámetro “lim_sup” es el valor considerado como frente de la imagen. El parámetro “lim_inf” es el valor considerado como fondo de la imagen. La variable de salida “n”, es el valor estimado del número de pixeles movidos en la imagen, y se determina mediante la proposición mencionada en el marco teórico.

function [PSFprom] = estimar_psf (n, alpha, lim_inf);

Esta función estima la PSF que ha causado la distorsión de la imagen. El parámetro “n” es el número de pixeles que se encuentran movidos en la imagen. El parámetro “alpha” es el mapa de transparencia de la imagen el cual debe haber sido determinado previamente. El parámetro “lim_inf” es el valor considerado como fondo de la imagen. La variable de salida “PSFprom” es un promedio de las PSFs que se han calculado para cada fila de la matriz de transparencia. Cada PSF es calculada en base a (5).

Anexo C

Imágenes utilizadas en el proyecto







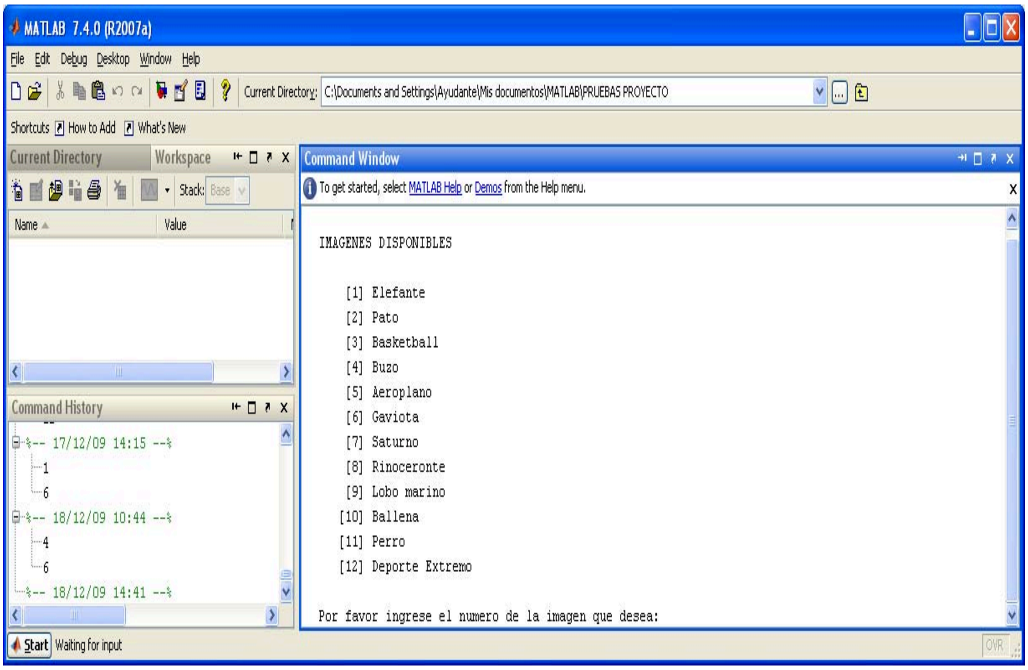


ANEXO D

Manual de Usuario

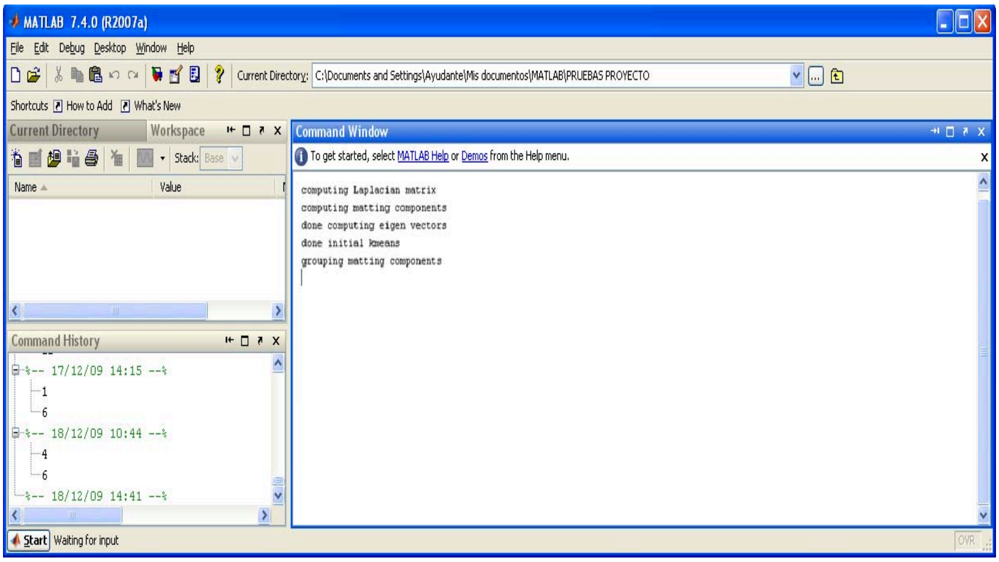
Abrir el archivo “demostracion.m”

Al correr el programa aparecerá el siguiente menú para que el usuario escoja la imagen a la cual se le aplicara la distorsión y restauración.

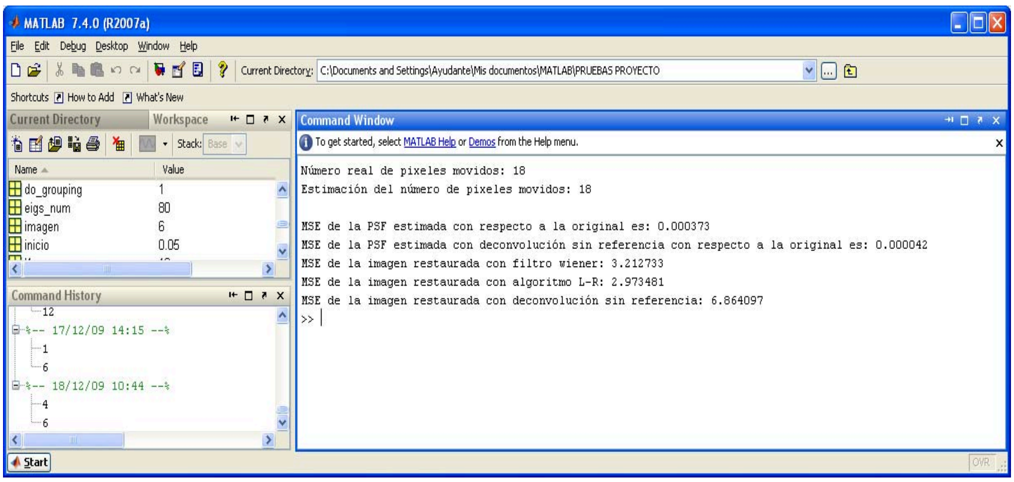


Una vez escogida la imagen aparecerán secuencialmente la imagen original, la imagen distorsionada y los trazos que se realizaron para la obtención de la matriz de transparencia, luego el programa procede a calcular la matriz de transparencia lo cual puede demorar unos cuantos minutos.

Durante el cálculo de la matriz de transparencia aparecerán en la pantalla sucesivamente los siguientes mensajes.



Después con base a esta matriz se estima el número de pixeles movidos en la imagen y se calcula la PSF con la cual se procede a restaurar la imagen. Al final el programa muestra el número real de pixeles movidos y la estimación hecha, así como los MSE.



REFERENCIAS

[1] Jianya Jia. *Single Image Debluring Using Transparency.* Department of Computer Science and Engineering. The Chinese University of Hong Kong, 2007.

[2] A. Levin, D. Lischinski, and Y. Weiss. *A Closed Form Solution to Natural Image Matting*. In CVPR, 2006.

[3] Anat Levin. Pubications, Codes and Images
http://www.wisdom.weizmann.ac.il/~levina/matting.tar.gz

[4] R. C. Gonzalez, R. E. Woods y S. L. Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall, 2004.

[5] Wikipedia. MATLAB, www.wikipedia.org/wiki/MATLAB

Imagenes.:

Figura 1.1:

http://1.bp.blogspot.com/_tGRbgaFgWh4/R0jBBRg1hPI/AAAAAAAABjQ/HbpGpmp4Y0Y/s400/motion_blur.jpg

Figura 1.3:

http://www.wisdom.weizmann.ac.il/~levina/papers/Matting-Levin-Lischinski-Weiss-06.ppt

Imagen 1:

http://bloggarte.com/wp-content/uploads/2007/10/elefante.jpg

Imagen 2:

http://sobrefotos.com/wp-content/uploads/2008/10/aves5.jpg

Imagen 3:

http://img.timeinc.net/time/photoessays/2007/basketball/basketball_01.jpg

Imagen 4:

http://s3.amazonaws.com/lcp/el-rincon-de-la-expresion/myfiles/buzo.jpg

Imagen 5:

Proporcionada por la Ing. Patricia Chávez

Imagen 6:

http://www.aereoo.com/wp-content/uploads/2009/09/estatua-de-la-libertad.jpg

Imagen 7:

Proporcionada por la Ing. Patricia Chávez

Imagen 8:

http://humano.ya.com/paquicano-1/_headerimages/gaviota.jpg

Imagen 9:

http://yami178.files.wordpress.com/2009/10/saturno.jpg

Imagen 10:

http://www.elpais.com/recorte/20080808elpepusoc_4/XLCO/Ies/20080808elpepusoc_4.jpg

Imagen 11:

http://www.notihuatulcopuertoescondido.com/wp-content/uploads/2008/03/gustavo-zambrano.jpg

Imagen 12:

http://www.vootar.com/imgs/elementos/1252424735_Delfin.jpg

Imagen 13:

http://www.pasaporteblog.com/wp-content/uploads/2007/12/ballena-beluga-oceanografico-valencia.jpg

Imagen 14:

http://www.answersingenesis.org/assets/images/articles/am/v2/n3/sea-turtle.jpg

Imagen 15:

http://www.gatosyperros.net/wp-content/uploads/2009/07/labrador.jpg

Imagen 16:

http://studyabroad.tcu.edu/userfiles/image/contest%202007/The%20Jump.jpg

Imagen 17:

http://www.masmoto.net/blog/wp-content/uploads/2008/04/carlos-checa-paracaidas-valencia.jpg

Imagen 18:

http://www.fotos-animales.com/imagenes/fondos-pantalla-aves.jpg

Imagen 19:

http://www.nicewallpapers.info/pics/animals/tiger/tiger_7.jpg

Imagen 20:

http://1.bp.blogspot.com/_5NR3cUZKz2s/Sc_h3nndW2I/AAAAAAAAAGk/W5psuKjINeI/s400/2300-2024%257EDeportes-extremos-snowboard-Posters.jpg