



# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL CENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA



## IMPLEMENTACIÓN DE UN ALGORITMO PARA EL CÁLCULO ESTIMADO DEL MOVIMIENTO LINEAL EN IMÁGENES Y SU APLICACIÓN EN LA RESTAURACIÓN DE IMÁGENES MOVIDAS.

José Luis Masache Narváez <sup>(1)</sup>, Ernesto Javier Roldán Monge <sup>(2)</sup>  
Facultad de Ingeniería en Electricidad y Computación (FIEC)  
Escuela Superior Politécnica del Litoral (ESPOL)  
Campus Gustavo Galindo, Km 30.5 Vía Perimetral  
Apartado 09-01-5863. Guayaquil, Ecuador  
jmasache@fiec.espol.edu.ec <sup>(1)</sup>, eroldan@fiec.espol.edu.ec <sup>(2)</sup>  
Msc. Patricia Chávez  
pchavez@fiec.espol.edu.ec

### Resumen

*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. 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.*

**Palabras Claves:** *Distorsión Lineal*

### Abstract

*In the digital signal processing is very common to find problems such as noise (in the case of audio or video signals) or distortion (in the case of images). There is a special case called linear distortion, which is basically the distortion caused by the relative motion of the object either photographed or the camera itself. Our main goal during development of this project is the restoration of all or part of an image that has been previously linearly distorted, because it is virtually impossible to obtain a complete restoration will consider a satisfactory outcome in which to recognize the object or face image, which obviously was the reason that the picture was taken.*

## 1. Introducción

La distorsión lineal que se produce en una imagen se debe a que el objeto se mueve de su posición original, con lo cual aparece una especie de sombra alrededor de los bordes del objeto.

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.

## 2. 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

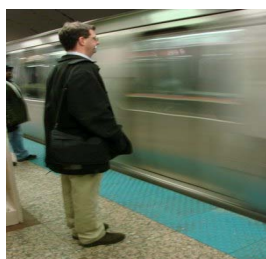
L: Imagen sin degradar

n : Ruido aditivo

f: PSF

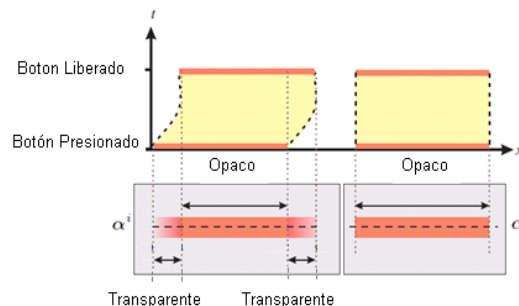
$\otimes$ : Convolución

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 Function) 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 2.1** 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 2.2** tiempo vs espacio (apreciación del fenómeno de Transparencia) [1].

En la figura 1.2 se muestra un objeto sólido 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. 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  $\alpha_i$  es un coeficiente que representa el grado de opacidad del frente de la imagen para el i-esimo pixel.



**Figura 2.3** 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  $\alpha^i$ , donde  $0 \leq \alpha^i \leq 1$ . El objeto opaco sin distorsión, oculta

completamente el fondo de la imagen, produciendo valores binarios de alfa  $\alpha^0$ , 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].

**2.1. 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 una buena aproximación para el ancho del filtro de distorsión lineal [1].

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

$$f_j = \begin{cases} \alpha_{j+n}^t - \alpha_{j+n-1}^t & -n < j \leq n \\ \alpha_0^t & j = -n \end{cases}$$

### 3. Implementación

Para el desarrollo de nuestro proyecto vamos a utilizar el software matemático MATLAB 2007<sup>a</sup> (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.

#### 3.1 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:

$$\text{PSF}=\text{fspecial}(\text{'motion'},\text{len},\text{theta})$$

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.

$$g=\text{imfilter}(f,\text{PSF},\text{filtering\_mode})$$

En “filtering\_mode” se usa “conv” para realizar el filtrado usando convolución.



Figura 3.1 Imagen sin distorsión.



Figura 3.2 Imagen distorsionada.

#### 3.2 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 3.3** Trazos



**Figura 3.4** Mapa de transparencia resultante.

### 3.3 Estimación del Tamaño de la PSF.

Basados en la proposición mencionada en el marco teórico se realizó 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 3.5** Acercamiento del mapa de transparencia resultante.

### 3.4 Cálculo de la PSF.

Basados en (5) se desarrolló 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.

### 3.5 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 realizó 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.

#### 3.5.1 Función deconvwnr

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

$$fr = \text{deconvwnr}(g, \text{PSF})$$

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 [4].



**Figura 3.6** Imágenes restauradas usando Filtro Wiener.

#### 3.5.2 Función deconvlucy

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

$fr = \text{deconvlucy}(g, \text{PSF}, \text{NUMIT}, \text{DAMPAR}, \text{WEIGHT})$

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



**Figura 3.7** Imágenes restauradas usando el Algoritmo L-R.

### 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:

$[fr, \text{PSFe}] = \text{deconvblind}(g, \text{INITPSF}, \text{NUMIT})$

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 usó este parámetro con un valor de 30.



**Figura 3.8** Imágenes restauradas usando Deconvolución sin referencia.

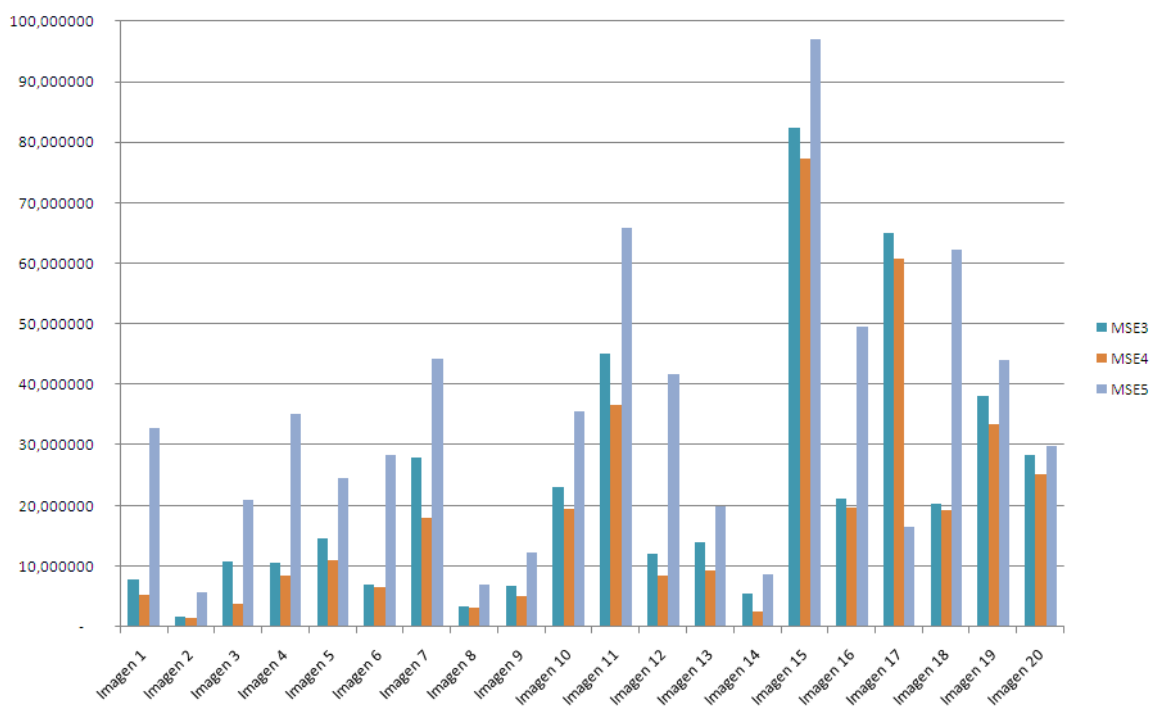
## 4. Análisis de Resultados.

Para comprobar que tan eficiente es el algoritmo propuesto se realizaron pruebas con 20 imágenes distintas, en donde se varió el número de píxeles 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.



**Figura 4.1** MSE de las imágenes restauradas por los 3 métodos mencionados con respecto a la imagen original.

Donde:

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.

## 5. 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  $\alpha^i$  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  $\alpha$  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 funciona 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.

## 6. 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.

## 6. Referencias

- [1] Jianya Jia. Single Image Deblurring 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. Publications, 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](http://www.wikipedia.org/wiki/MATLAB)