



Universidad Tecnológica de la Mixteca

**RESTAURACIÓN DE IMÁGENES CON EL
MÉTODO DE GRADIENTE CONJUGADO**

TESIS

PARA OBTENER EL TÍTULO PROFESIONAL DE:

INGENIERO EN COMPUTACIÓN

PRESENTA:

LIZ NICANDI CASTILLO JIMÉNEZ

ASESOR:

M.C. LUIS RENÉ MARCIAL CASTILLO

A mis padres (Alicia y Román)

Por haber sabido nutrirnos con un gran cariño y apoyo incondicional que ha permitido realizar nuestros sueños.

A mis hermanos (Miriam y Ato)

Porque aunque nos demos cuenta de que ya no traemos las rodillas raspadas, nos seguimos apoyando.

A Gerardo

Por ser el apoyo y proporcionar siempre el aliento que impulsa a continuar.

11583

Agradecimientos

Muchas gracias a quienes me apoyaron y asesoraron para que pudiera culminar este trabajo. A los M.C. Luis René, Marcela, Aroldo y José Antonio.

Muchas gracias también al equipo Yukunitzá que han sabido brindar una amistad incondicional y sincera.

A todas aquellas personas que de alguna manera directa o indirecta hicieron que esto fuera posible.

Índice General

U. T. M. 11583

1	Introducción	1
2	Conceptos Básicos	6
2.1	Error Absoluto y Error Relativo.	6
2.2	Normas Más Usadas.	7
2.3	Optimización .	8
2.4	Tipos de Mínimos	9
2.5	Gradiente y Hessiana.	10
2.6	Matriz Definida Positiva.	12
2.7	Funciones Cóncavas y Convexas.	12
2.8	Imágenes como Archivos.	13
3	Estimación Bayesiana	15
3.1	Introducción .	15
3.2	Teorema de Bayes	16
3.3	Estimador Máximo a Posteriori (MAP)	17
3.3.1	Funciones de Costo.	18
3.3.2	Definición del Estimador MAP	21
3.4	Pasos de la Estimación Bayesiana	21

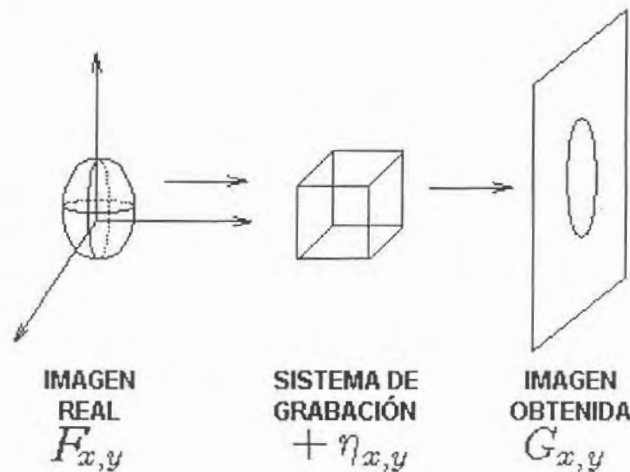
4	Campos Aleatorios Markovianos	22
4.1	Introducción	22
4.2	Etiquetado	23
4.3	Vecinos y Cliques.	25
4.4	Distribución de Gibbs	28
4.5	Energía Posterior de los CAM	29
4.6	Modelo de Resortes	30
5	Programación No Lineal Sin Restricciones	32
5.1	Introducción	32
5.2	Condiciones Necesarias y Suficientes	34
5.3	Convergencia de los Algoritmos de Optimización.	34
5.3.1	Estabilidad	35
5.3.2	Velocidad de Convergencia	35
5.4	Búsqueda Lineal	36
5.5	Dirección de Descenso	37
5.6	Problemas Mal Condicionados y Bien Condicionados	38
6	Métodos de Gradientes	39
6.1	Introducción	39
6.2	Direcciones de Descenso	40
6.3	Método de Descenso de Gradiente	41
6.3.1	Algoritmo	42
6.4	Método de Newton-Raphson	43
6.4.1	Algoritmo	43
6.5	Gradientes Conjugados	44

6.5.1 Búsqueda Lineal	46
7 Algoritmo Propuesto	47
7.1 Descenso de Gradiente	48
7.2 Métodos de Gradientes Conjugados	49
7.2.1 Cálculo del Valor β	50
7.2.2 Cálculo de β para PR ⁺	51
7.2.3 Cálculo de β Híbrido de FR y PR.	52
7.3 Funciones Comunes	53
7.3.1 Gradiente.	53
7.3.2 U(F)	54
7.3.3 Búsqueda Lineal.	55
8 Pruebas	57
9 Conclusiones	75
Bibliografía	77
Apéndice A. Manual de Usuario	79
A.1 Requerimientos del Programa	79
A.2 Pantalla Principal	80
A.3 Menú Principal	80
A.4 Menú Métodos de Restauración	82
A.5 Menú Configuración.	85
A.6 Salir	86

Capítulo 1

Introducción

Para poder obtener una imagen a partir de un sistema tridimensional, es necesario convertirla en una imagen 2D, esto se puede lograr a partir de un sistema de grabación (como una cámara). Debido a que los sistemas de grabación no son perfectos, estos introducirán en el momento de adquisición de la imagen alguna distorsión en la misma, también conocido como ruido.



Adquisición de una imagen

Como es lógico pensar, en imágenes sintéticas no se tiene este problema, pero aún así puede estar presente en ellas algún tipo de distorsión generada por ejemplo en la trans-

misión de datos. La supresión de ruido es muy importante desde que se emplean imágenes en campos de investigación como medicina, astronomía, industria, etc., en donde se requiere tener precisión en los datos adquiridos, que al encontrarse alterados, degeneran los resultados esperados. Éste es un tema fundamental dentro del procesamiento de imágenes.

El ruido puede estar presente de diferentes formas, de acuerdo al agente que lo genere. Entre las formas más comunes se encuentran: el ruido uniforme; el ruido de Sal y Pimienta; el ruido multiplicativo; y el ruido Gaussiano, que por su naturaleza, es el más difícil de eliminar.

Una representación matemática de éste fenómeno se puede hacer de la siguiente forma:

$$G_{x,y} = F_{x,y} + \eta_{x,y}, \quad (1.1)$$

donde:

$G_{x,y}$, es la imagen observada

$F_{x,y}$, es la imagen original

$\eta_{x,y}$, es el ruido

Para poder eliminar el ruido (o una parte de él), es necesario un filtro que permita hacerlo. Este trabajo de tesis está enfocado precisamente a las imágenes que presenten por alguna razón distorsión en sus datos.

El principal objetivo de esta tesis es tratar de suprimir la mayor cantidad de ruido posible de las imágenes. Para lograrlo, se aplican diversos tipos de filtros de Gradientes Conjugados, con lo que se espera obtener una imagen que se aproxime más a $F_{x,y}$, de forma que se logre obtener el mínimo de $\eta_{x,y}$, es decir $G_{x,y} \approx F_{x,y}$. Se seleccionó para este fin un método de optimización de Gradientes Conjugados que será aplicado a Procesamiento de Imágenes, este tipo de filtros tienen además la característica de que no degenera los rasgos significativos de la imagen (como bordes) después de múltiples iteraciones. Es

importante remarcar que el filtrado de ruido no tiene una aplicación en específico, ya que existen una gran diversidad de áreas en las que se requiere trabajar con imágenes, por lo tanto se puede considerar como una herramienta de trabajo multidisciplinaria.

El problema a lo largo del trabajo de tesis se planteará como un problema de minimización clásico:

$$\min f(x), \quad (1.2)$$

donde f es suave y su gradiente g es factible.

Para el método de interés de gradientes conjugados, las iteraciones que permiten hallar la solución, son de la forma:

$$d_k = \begin{cases} -g_k & \text{para } k = 1, \\ -g_k + \beta_k d_{k-1} & \text{para } k \geq 2, \end{cases} \quad (1.3)$$

$$x_{k+1} = x_k + \alpha_k d_k, \quad (1.4)$$

donde g_k es el gradiente de la función, β_k es un escalar, y α_k es el tamaño de paso obtenido mediante una búsqueda unidimensional.

Para realizar las pruebas y elección del mejor algoritmo en la restauración de imágenes, se emplean las más conocidas fórmulas para β_k : Fletcher-Reeves [8],[6], Polak-Ribiere [8],[6], Hestenes-Stiefel [8],[6], además la de Polak-Ribiere positivo [8].

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}. \quad (1.5)$$

$$\beta_k^{PR} = \frac{\langle g_k, g_k - g_{k-1} \rangle}{\|g_{k-1}\|^2}. \quad (1.6)$$

$$\beta_k^{HS} = \frac{\langle g_k, g_k - g_{k-1} \rangle}{\langle d_{k-1}, g_k - g_{k-1} \rangle}. \quad (1.7)$$

$$\beta_k^{PR+} = \max \{ \beta_k^{PR}, 0 \}. \quad (1.8)$$

Se instrumentó un método Híbrido [8] que combina las fórmulas de Fletcher-Reeves y la de Polak-Ribiere tomando la mejor parte de ambas, de forma que:

$$\beta_k = \begin{cases} -\beta_k^{FR} & \text{si } \beta_k^{PR} < -\beta_k^{FR} \\ \beta_k^{PR} & \text{si } |\beta_k^{PR}| \leq \beta_k^{FR} \\ \beta_k^{FR} & \text{si } \beta_k^{PR} \leq \beta_k^{FR} \end{cases} \quad (1.9)$$

Además se hacen comparaciones con algunos filtros más sencillos para la eliminación de ruido: media, mediana y descenso de gradiente.

A las imágenes de prueba se les añadió para efectos prácticos ruido gaussiano, el cual fue generado por el programa Caliman, una calculadora de imágenes desarrollada por el Centro de Investigación en Matemáticas [11]. En las pruebas se reportan resultados sobre imágenes reales e imágenes sintéticas.

El trabajo de tesis se encuentra organizado de la siguiente forma: En el *Capítulo 2* se mencionan algunos Conceptos Básicos que permitirán sustentar las bases del trabajo, estos conceptos permitirán entender los capítulos subsecuentes. En el *Capítulo 3* se menciona la Estimación Bayesiana, concepto mediante el cual se le da a la imagen un tratamiento desde una perspectiva probabilística, incorporando los elementos del conocimiento a priori y verosimilitud, con los cuales se obtiene el conocimiento a posteriori. En el *Capítulo 4* se hace mención a los Campos Aleatorios Markovianos, a través de estos se puede proporcionar un comportamiento al sistema a través de sus cliques. En el *Capítulo 5* se abordan algunos tópicos de la Programación No Lineal Sin Restricciones, campo dentro de la Programación Matemática para la optimización. En el *Capítulo 6* se habla acerca de los Métodos de Gradientes, se mencionan algunas generalidades, así como los métodos de Gradientes Conjugados a comparar. En el *Capítulo 7* se plasma el Algoritmo Propuesto para la restauración de imágenes, aquí se mencionan los algoritmos que fueron instrumentados para su posterior comparación en el capítulo de Pruebas. En el *Capítulo 8* se muestran algunas Pruebas sobre el desempeño de los algoritmos de gra-

dientes conjugados, comparándolos entre sí, además de realizar comparaciones con otros métodos para eliminar ruido en imágenes, estos son: Descenso de Gradiente, Media y Mediana. En el *Capítulo 9* se muestran las Conclusiones de este trabajo así como el trabajo a futuro. Posteriormente se encuentra la *Bibliografía* sirvió como base y sustento en la realización de este trabajo. Finalmente se encuentra el Apéndice, en donde se detalla el manual de usuario del programa *RESGRAD* que fue instrumentado en el presente trabajo de tesis.

Capítulo 2

Conceptos Básicos

Para una mejor comprensión de algunos términos que serán utilizados a través del desarrollo de esta tesis se incluye este capítulo, en él se mencionan algunos conceptos de manera general, tal como: Errores (Absoluto y Relativo), Normas Más Usadas, Optimización, algunos Tipos de Mínimos, Gradiente y Hessiana, e Imágenes como Archivos.

2.1 Error Absoluto y Error Relativo

En el momento de realizar alguna medición o cálculo se pueden permitir algunos tipos de errores, generados de diversas formas: por calibración en los aparatos de medición, condiciones no apropiadas, fórmulas o modelos aproximados, sensibilidad de medición del aparato, errores debido a la apreciación limitados por el observador (como oído o vista), cambio en condiciones del entorno, etc.

Ya que en muchos cálculos sólo nos podemos acercar a un valor, se da pie a cuantificar la aproximación deseada, es decir, qué tanto podemos alejarnos o acercarnos a un valor verdadero.

Se define el error absoluto como la diferencia existente entre el valor exacto y el valor obtenido experimentalmente. Para el caso de la restauración de imágenes no se puede saber el valor exacto, por lo tanto, no se puede tampoco conocer el error absoluto tal como está definido anteriormente. Entonces, lo que se espera conocer es una estimación de la incertidumbre que se asocia a un resultado obtenido, a esta incertidumbre también se le conoce como error absoluto. Matemáticamente se puede expresar de la siguiente forma:

$$\text{Valor Obtenido} = m \pm E, \quad (2.1)$$

donde:

E representa el error absoluto que se añade a la medición.

El signo \pm indica que el error puede producirse por exceso o defecto.

m es la medición real sin error.

De lo anterior se puede ver que la magnitud media se encuentra en el intervalo $(m - E, m + E)$. Dado que no representa lo mismo un error de 0.1 con respecto a 1 que a 100, se define entonces el error relativo, el cual se expresa como:

$$\frac{E}{m}, \quad (2.2)$$

esto indica que el error relativo es aquél que permite relacionar el error absoluto con el valor de la magnitud medida.

2.2 Normas Más Usadas

Para medir el grado de aproximación, es necesario medir el 'tamaño' del error E . Una manera de hacer esta medición es con el concepto de norma de un vector, para lo cual se definen las siguientes normas vectoriales:

$$1. \|x\|_{\infty} = \max_{1 \leq i \leq n} |x_i|, \quad \text{donde } x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

A lo anterior se le conoce como *norma infinita* o *norma máxima*, que no es más que el valor máximo (en valor absoluto) de los términos del vector x .

$$2. \|x\|_1 = \sum_{i=1}^n |x_i| = |x_1| + |x_2| + \dots + |x_n|.$$

A lo que se conoce como *norma uno* o *norma aditiva*, que es la suma de todos los valores absolutos de los términos del vector x .

$$3. \|x\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}.$$

Conocida también como *norma dos* o *norma euclídeana*, que representa a la raíz cuadrada de la suma de todos los términos del vector x elevados al cuadrado.

2.3 Optimización

La optimización se puede considerar como la forma en que se puede responder a una pregunta ¿qué es lo mejor?, en el caso cuando la respuesta a determinados problemas puede ser expresada como un valor numérico. Los problemas que se enmarcan en este tipo son aplicados a diversas áreas científicas, tales como Química o Física, además de campos como Ingeniería, Arquitectura, Economía y Administración.

En la optimización se busca formular un problema como la combinación de los parámetros que permitan obtener el mejor resultado de una cantidad dada, sometiéndolo a restricciones. El valor que se desea optimizar es la *función objetivo*, los parámetros que pueden ser cambiados en la búsqueda del óptimo son las llamadas variables de decisión y las *restricciones* permiten acotar el problema.

Un problema de optimización se expresa de manera general de la siguiente forma:

Minimizar $f(x)$

$$x \in \mathbb{R}^n$$

sujeto a

$$c_i(x) = 0, \quad i = 1, 2, \dots, m' \quad (2.3)$$

$$c_i(x) \geq 0, \quad i = m' + 1, \dots, m$$

donde:

$f(x)$, es la función objetivo.

x , es un vector columna de n variables independientes.

$c_i(x)$, es el conjunto de restricciones, expresadas como términos de igualdad o desigualdad.

$f(x)$ y $c_i(x)$ en conjunto son las funciones problema.

2.4 Tipos de Mínimos

Cuando hablamos de la solución óptima de un problema, es necesario ver cuáles son los tipos de resultados que se están obteniendo, por lo que se deben tener bien en claro los conceptos de: *Mínimo Local* y *Mínimo Global*, la Figura 2-1 ilustra los conceptos anteriores.

Definición 2.1 *Mínimo Global*: Un punto x^* se dice ser un mínimo global si cumple que $f(x^*) \leq f(x)$ para todo x .

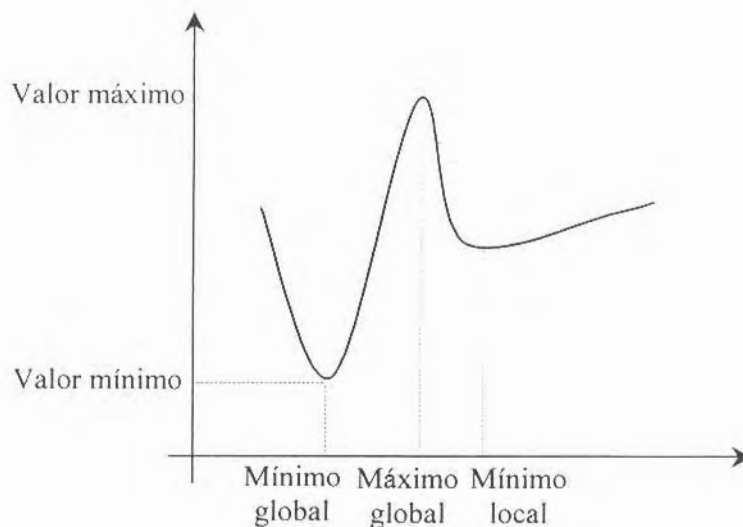


Figura 2-1: En la figura se muestran: el mínimo global, mínimo local y el máximo global para una función.

Definición 2.2 *Mínimo Local*: Un punto x^* es un mínimo local si existe una Vecindad¹ N de x^* tal que $f(x^*) \leq f(x)$, para todo $x \in N$.

Definición 2.3 *Mínimo Local Estricto*: Un punto x^* es un mínimo local estricto si existe una vecindad N de x^* tal que $f(x^*) < f(x)$ para todo $x \in N$ con $x \neq x^*$.

2.5 Gradiente y Hessiana

Dada una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$, se define el vector gradiente² como:

¹Una vecindad de x^* es un conjunto abierto que contiene a x^* .

²El vector gradiente se puede denotar como $\nabla f(x)$ o g y g_k es equivalente a $\nabla f(x_k)$. En el desarrollo de la tesis se usan indistintamente estas notaciones.

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_i} \right) = \begin{pmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{pmatrix}.$$

Por ejemplo, si se tiene $f(x) = x_1^2 x_2 + e^{x_1} x_2$

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 2x_1 x_2 + e^{x_1} x_2 \\ x_1^2 + e^{x_1} \end{bmatrix}.$$

El gradiente caracteriza el comportamiento de un campo escalar en el entorno de un punto, es decir, proporciona el sentido de crecimiento más rápido de una función en ese punto.

Definición 2.4 La matriz Hessiana de una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ se define como:

$$\nabla^2 f(x) = \left(\frac{\partial f}{\partial x_i x_j} (x) \right) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 f(x)}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} & \dots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{pmatrix}.$$

Por ejemplo, si se tiene $f(x) = x_1^2 x_2 + e^{x_1} x_2$, su matriz Hessiana es la siguiente:

$$\nabla^2 f(x) = \begin{pmatrix} 2x_2 + e^{x_1} x_2 & 2x_1 + e^{x_1} \\ 2x_1 + e^{x_1} & 0 \end{pmatrix}.$$

2.6 Matriz Definida Positiva

Una matriz A es *definida positiva* si cumple con cualquiera de las siguientes condiciones:

- $x^T Ax > 0$ para todos los vectores x distintos de cero.
- Todos los valores propios de A satisfacen $\lambda_i > 0$.
- Todas las submatrices A_k tienen determinantes positivos.
- Todos los pivotes (sin intercambio de filas) satisfacen $d_i > 0$.

2.7 Funciones Cóncavas y Convexas

Una función es convexa cuando los segmentos que unen cualquier par de puntos en su gráfica, quedan siempre por encima de la gráfica. Si la función es estrictamente convexa, los segmentos no pueden tocar a la gráfica, salvo en los puntos extremos.

Definición 2.5 Una función $f(x)$ definida sobre un conjunto convexo S es convexa cuando $\forall x, y \in S$ y $\forall \lambda \in [0, 1]$ si cumple:

$$f((1 - \lambda)x + \lambda y) \leq (1 - \lambda)f(x) + \lambda f(y). \quad (2.4)$$

Cuando en la condición anterior el signo de la desigualdad es $<$ y $\lambda \in (0, 1)$, la función se dice que es estrictamente convexa.

Para las funciones cóncavas y estrictamente cóncavas, se aplica la misma definición que para las funciones convexas, cambiando el signo de la desigualdad de la ecuación (2.4). Para lo cual, una función $f(x)$ es cóncava si y sólo si su opuesta $-f(x)$ es convexa.

2.8 Imágenes como Archivos

El objeto de trabajo serán precisamente las imágenes, estas pueden estar en blanco y negro, en escala de gris o en color. Por simplicidad, el desarrollo de la tesis se hizo sobre imágenes en escala de grises, esto debido a que de esta forma es mucho más fácil transportarlo a imágenes a color. Para poder entender esto, se mencionarán a continuación algunas generalidades de las imágenes a color.

Debido a la estructura del ojo humano, los colores se ven en combinación de los llamados colores primarios (o básicos): Rojo, Verde y Azul, en este trabajo se enunciará por su notación inglesa RGB (Red, Green, Blue). Si se realiza la combinación de los colores primarios se pueden obtener los colores secundarios, tal como amarillo (rojo + verde), cian (verde + azul), magenta (rojo + azul).

Es necesario diferenciar entre los colores primarios de luz y los colores primarios de pigmentación. Un color primario de pigmentación se define como uno que absorbe un color primario de luz y transmite los otros dos.

La característica que generalmente se emplea para poder distinguir los colores son el brillo, el color (matiz) y la saturación. El brillo es asociado a la intensidad, el matiz es asociado con la longitud de onda dominante en la mezcla de ondas luminosas y la saturación se asocia a la cantidad de luz blanca mezclada con el matiz.

El modelo de color RGB se basa en un sistema de coordenadas cartesiano (ver figura 2-2), el subespacio de interés es el cubo, por conveniencia se supone que todos los colores han sido normalizados de forma que el cubo es unitario (los valores están en el rango $[0, 1]$).

De lo ya mencionado anteriormente, se puede inferir que una imagen a color está formada por tres imágenes, cada una en un plano distinto (rojo, verde y azul). La cantidad de colores que soporte esta imagen, dependerá directamente del número de bits

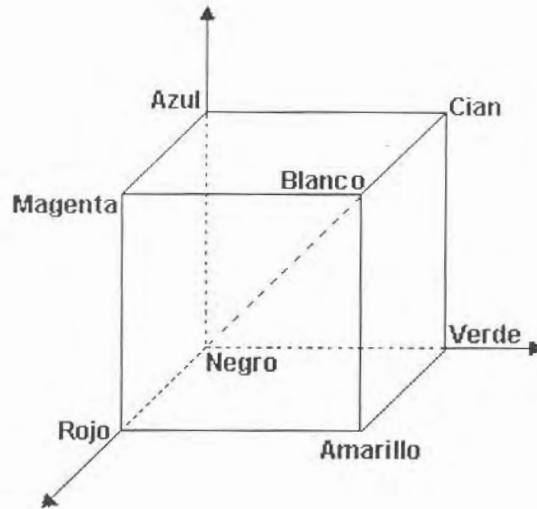


Figura 2-2: Colores representados en un Cubo RGB, los puntos que forman la diagonal principal contendrán los niveles de gris, negro en (0,0,0) y blanco en (1,1,1).

empleados para cada plano $num_colores = (2^{bits_x_plano})^3$, es decir : una imagen de 1 bit por plano permitirá almacenar una imagen de 8 colores, una imagen de 8 bits por plano permitirá almacenar 16777216 colores.

Para las imágenes en escala de grises se emplea un solo plano, que indica únicamente la cantidad de iluminación del pixel, 0 se asocia al negro y 1 se asocia al blanco.

Se puede ver que trabajar con imágenes a color, es equivale a trabajar con tres imágenes en escala de grises, una asociada a cada plano (RGB). Por lo que para el trabajo de tesis se realiza el procesado con imágenes en escala de grises, se almacena en 8 bits por pixel con formato BMP³.

³Para una mejor referencia de los formatos de imágenes se puede consultar el libro *Graphics File Formats*, David C. Kay John R. Levine, Windcrest / Mc Graw Hill 1992. o la página web: <http://www.dcs.ed.ac.uk/home/mxr/gfx/2d-lo.html>

Capítulo 3

Estimación Bayesiana

3.1 Introducción

Al obtener una imagen de cualquier tipo, normalmente se generan datos por una función desconocida. Los datos son usualmente ruido que se añade a la imagen original. El problema es estimar esta función y realizarlo de manera exacta es un problema intratable, pero existen métodos para lograr una aproximación y minimizar la cantidad de ruido razonablemente. Una de las formas para lograrlo es a través de la estimación Bayesiana, la cual permite el tratamiento de datos ante la incertidumbre.

El manejo de probabilidades tiene como ventaja, mantener siempre todas las posibles estimaciones candidatas, además de ofrecer como mejor estimación la que tenga mayor probabilidad o la estimación esperada.

Por medio de la estimación bayesiana se puede fusionar la información de una imagen inicial, que se expresa por medio de una distribución de probabilidad *a priori* (o distribución inicial); con las observaciones estadísticas, que permitirán producir una distribución *a posteriori* (o distribución final). Mediante esta fusión se tiene una buena base para poder extraer conclusiones y tomar decisiones acerca del ruido que se desea eliminar.

3.2 Teorema de Bayes

Para poder aplicar el Teorema de Bayes es necesario especificar la distribución inicial, así como probabilidad en términos de grados de creencia que terminarán siendo ya sea personales o subjetivos pero que estarán condicionados a la información de la imagen con la que se dispone. El Teorema de Bayes se puede denotar como una Hipótesis H , con evidencia E :

$$p(H|E) = p(H) \times \frac{p(E|H)}{p(E)}. \quad (3.1)$$

El término del lado izquierdo, $p(H|E)$ es llamado la probabilidad “*a posteriori*”, y es la probabilidad de la hipótesis H después de considerar el efecto que tiene en ella la evidencia E .

El término $p(H)$ es la principal probabilidad de que ocurra H , dicho de otro modo, creer en H antes de que la evidencia E sea considerada.

El término $p(E|H)$ es la llamada verosimilitud, y está dada por la probabilidad de la evidencia E , asumiendo que la hipótesis H sea verdadera.

El último término $1/p(E)$ es independiente de H y puede ser estimado como una constante de normalización que hace que todas las probabilidades sumen 1.

Se debe notar que todas estas probabilidades son condicionales, es decir, especifican el grado de creencia en las proposiciones bajo determinadas suposiciones. El Teorema de Bayes es consecuencia de la regla de productos de Probabilidad. La regla de productos da la probabilidad de la conjunción lógica “*y*” de dos enunciados A y B , escrito como A, B , es decir:

$$p(A, B) = p(A|B) \times p(B) = p(B|A) \times p(A). \quad (3.2)$$

La Regla de Bayes está derivada del reordenamiento de los términos de la igualdad (3.2). Usando la regla de productos directamente, se puede extender la regla de Bayes a

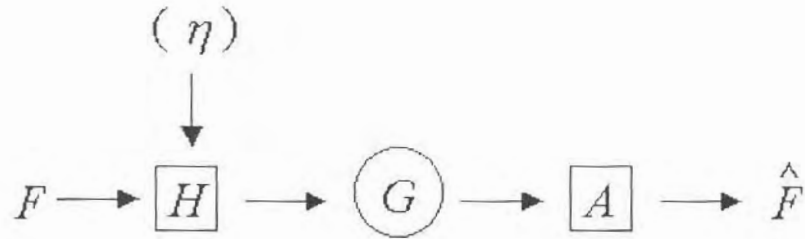


Figura 3-1: F representa la imagen original; H el dispositivo de adquisición; η el ruido introducido; G la imagen obtenida; A el filtro para eliminar el ruido, \hat{F} la imagen restaurada

múltiples actualizaciones secuenciales, quedando de la siguiente forma:

$$\begin{aligned}
 p(H|E_1, E_2, E_3) &= p(H) \times \frac{p(E_1, E_2, E_3|H)}{p(E_1, E_2, E_3)} \\
 &= \frac{p(H) \times p(E_1|H) \times p(E_2|E_1, H) \times p(E_3|E_2, E_1, H)}{p(E_1) \times p(E_2|E_1) \times p(E_3|E_2, E_1)}.
 \end{aligned}$$

De las ecuaciones anteriores se puede ver que cada nueva evidencia E_n es factorizada entre el número de cálculos a realizar.

3.3 Estimador Máximo a Posteriori (MAP)

Hasta este momento se ve que se pueden encontrar una gran infinidad de valores para aproximaciones a la función objetivo, dado que se cuenta con muchos grados de libertad. Para obtener una única solución, se necesita introducir el conocimiento a priori de la imagen F , es decir, se deben poner restricciones en A (representado en la figura 3-1).

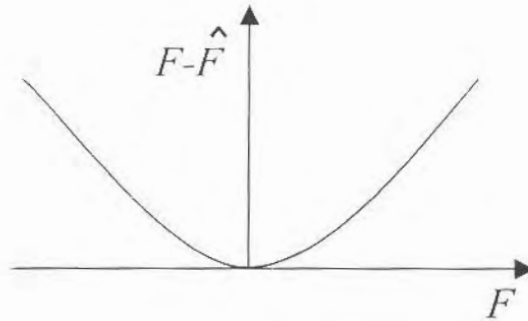


Figura 3-2: Función de costo para la media

3.3.1 Funciones de Costo

Para poder hablar de un resultado óptimo, se debe ponderar de acuerdo a una función de costo, algunas de estas funciones son:

$$C(F, \hat{F}) = (F - \hat{F})^2. \quad (3.3)$$

$$C(F, \hat{F}) = |F - \hat{F}|. \quad (3.4)$$

$$C(F, \hat{F}) = 1 - \delta(F - \hat{F}). \quad (3.5)$$

A la función de costo (3.3) se le conoce como Media (ver Figura 3-2), la cual indica: Si se obtiene el resultado óptimo, el costo es cero; si se está cerca del valor óptimo, el costo es poco; pero si el resultado está lejos de ser el óptimo, la función de costo toma un valor alto.

El valor esperado con respecto a una función de costo se expresa de la siguiente forma:

$$Q(\hat{F}) = E[C(F, \hat{F})]. \quad (3.6)$$

De acuerdo a la ecuación (3.6). se obtiene para la función de costo de la media (3.3)

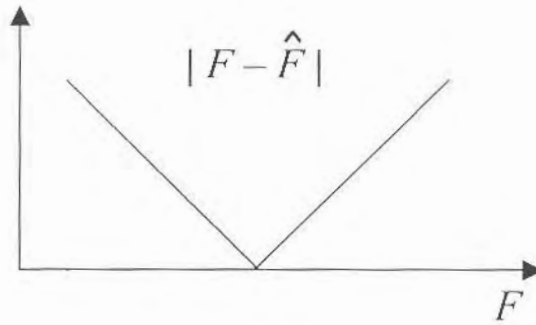


Figura 3-3: Función de costo para la mediana

lo siguiente:

$$\begin{aligned}
 Q(\hat{F}) &= E\left[(F - \hat{F})^2\right] \\
 &= \int_{-\infty}^{\infty} (F - \hat{F})^2 P_{F|G}(F) dF \\
 &= \int_{-\infty}^{\infty} F^2 P_{F|G}(F) dF - 2\hat{F} \int_{-\infty}^{\infty} F P_{F|G}(F) dF + \hat{F}^2 \int_{-\infty}^{\infty} P_{F|G}(F) dF \\
 &= K - 2\hat{F}\bar{F} + \hat{F}^2
 \end{aligned}$$

Por lo tanto, $Q(\hat{F}) = K - 2\hat{F}\bar{F} + \hat{F}^2$

Minimizando, $\frac{\partial Q(\hat{F})}{\partial \hat{F}} = 0 - 2\bar{F} + 2\hat{F} = 0$

$$\hat{F} = \bar{F}$$

A la función de costo (3.4) se le conoce como Mediana (ver figura 3-3), la cual pesa proporcionalmente los errores, es decir: si se obtiene el óptimo, la función de costo toma el valor de cero; a medida que se está más lejos del óptimo, el valor de la función de costo será mayor.

A la función de costo (3.5), se le conoce como Moda (ver figura 3-4). esta indica lo

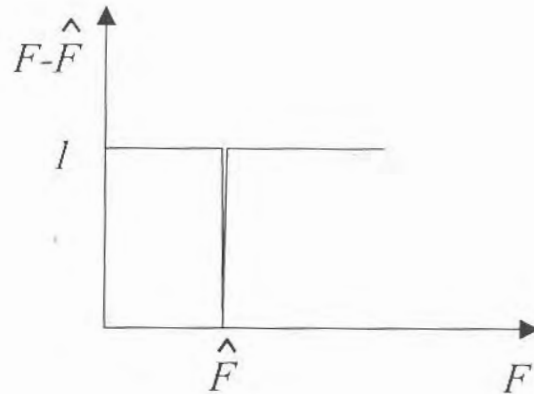


Figura 3-4: Función de costo de la moda

siguiente: Si se obtiene el valor óptimo, el valor de la función de costo será cero, de lo contrario, la función de costo toma el valor de 1.

Para la función de costo de la moda, se puede obtener¹:

$$\begin{aligned}
 Q(\hat{F}) &= E[1 - \delta(F - \hat{F})] & (3.7) \\
 &= \int_{-\infty}^{\infty} (1 - \delta(F - \hat{F})) P_{F|G}(F) dF \\
 &= \int_{-\infty}^{\infty} P_{F|G}(F) dF - \int_{-\infty}^{\infty} \delta(F - \hat{F}) P_{F|G}(F) dF
 \end{aligned}$$

$$\text{Finalmente, } Q(\hat{F}) = 1 - P_{F|G}(\hat{F}). \quad (3.8)$$

Para encontrar el óptimo se debe minimizar la función de costo (3.8). Ya que se ha definido la función de costo, se puede buscar los estimadores óptimos.

¹ $\delta(x) = \begin{cases} \infty & x = 0 \\ 0 & x \neq 0 \end{cases}$ y además $\int_{-\infty}^{\infty} \delta(x) dx = 1$.

3.3.2 Definición del Estimador MAP

Para definir el estimador MAP, se introduce la expresión:

$$\hat{F}^* = \min Q(\hat{F}) = \min \left(1 - P_{F|G}(\hat{F}) \right). \quad (3.9)$$

La ecuación (3.9) se puede expresar también como:

$$\hat{F}^* = \max P_{F|G}(\hat{F}). \quad (3.10)$$

A la ecuación (3.10) se le conoce precisamente como el estimador Máximo a Posteriori, que es muy empleado en Procesamiento de Imágenes, con este estimador se obtienen los parámetros más probables, y estos son aquellos que maximizan la probabilidad a posteriori, para este trabajo de tesis se aplicará dicho estimador.

El estimador MAP tiene como características que necesita realizar una búsqueda muy exhaustiva cuando el número de parámetros es muy pequeño.

3.4 Pasos para la Estimación Bayesiana

Los pasos para poder realizar correctamente una estimación bayesiana son:

1. Definir la función de verosimilitud $P_{G|F}(F) = P_{\eta}(G - H_F)$.
2. Definir la distribución a priori $P_F(F)$.
3. Resumir la información por medio de la Regla de Bayes para obtener la distribución a posteriori.
4. Definir la Función de costo $C(F, \hat{F})$.
5. Encontrar el Estimador Bayesiano óptimo.

Capítulo 4

Campos Aleatorios Markovianos

4.1 Introducción

Dado que la probabilidad a posteriori se deriva usualmente de un modelo a priori, así como de un modelo de verosimilitud, es fundamental para un problema representar en forma de restricciones la probabilidad a priori, esto puede ser posible mediante los Campos Aleatorios Markovianos (CAM).

Los CAM son una extensión del concepto proceso de Markov, el cual proporciona una herramienta para el análisis entre los fenómenos físicos de sus dependencias, ya sean espaciales o contextuales.

Entre algunas de las razones que hacen interesante el modelado a través de los CAM se encuentran:

- Hacen más fácil cuantificar el rendimiento de un algoritmo para el análisis de la imagen.
- Pueden ser usados para incorporar diversa información contextual a priori o restricciones.

- Los algoritmos basados en CAM tienden a ser locales, y tienden por sus características a ser implementados en hardware en paralelo de forma natural.

La solución óptima de un problema está definida por el estimador de probabilidad MAP. Los pasos sugeridos para realizar un modelado mediante los Campos Aleatorios Markovianos son:

1. Plantear el problema como un problema de etiquetado, en el que la configuración de cada etiqueta representa una solución.
2. Representar el problema como de etiquetado bayesiano, en el que la solución óptima se definida por la configuración de las etiquetas MAP.
3. Usar la distribución de Gibbs para caracterizar la distribución a priori de las configuraciones del etiquetado.
4. Definir la densidad de la probabilidad de los datos basados en un modelo de observaciones asumido.
5. Emplear las reglas bayesianas para derivar la distribución a posteriori de la configuración de etiquetas, midiendo el costo de una solución.

4.2 Etiquetado

La solución de un problema puede ser visto como un conjunto de etiquetas que son asignadas a los pixeles de la imagen. Existe un problema de etiquetado, el cual se puede especificar en términos de un conjunto de estados y un conjunto de etiquetas.

Sea d un conjunto discreto de m estados

$$d = \{1, \dots, m\}. \quad (4.1)$$

El orden de los estados no es importante, su relación es determinada por un sistema de vecindades. Se define D como un conjunto de etiquetas. Etiquetar no es más que clasificar, asignando una etiqueta de D para cada estado en d .

Un conjunto de estados puede ser categorizado en términos de su homogeneidad y un conjunto de etiquetas en términos de su continuidad.

Una etiqueta puede ser continua o discreta. En el caso continuo, el valor de la etiqueta asignado a i está dentro de un intervalo real, es decir:

$$f_i \in D = [x_j, x_h], \quad (4.2)$$

para este caso, existen un número infinito de etiquetas. Para el caso discreto, una etiqueta asume un valor discreto, es decir:

$$f_i \in D = \{1, \dots, M\}. \quad (4.3)$$

Sea $F = \{F_1, \dots, F_m\}$ una familia de variables aleatorias definidas en d , donde cada variable aleatoria F_i toma valores en D . Una intersección de eventos $\{F_1 = f_1, \dots, F_m = f_m\}$, abreviado como $F = f$, es una ejecución de F donde $f = \{f_1, \dots, f_m\}$ es llamado una configuración de F . Una configuración puede representar una imagen. Para el conjunto de todas las configuraciones es:

$$S = D^m = \underbrace{D \times D \times \dots \times D}_{m \text{ veces}}. \quad (4.4)$$

El espacio de soluciones posibles puede ser idéntico a S o si son impuestas restricciones adicionales, un subconjunto de S . Una configuración de f puede ser interpretada como una de las dos siguientes formas: Como un mapeo $f : d \rightarrow D$; o como un etiquetado $\{f_1, \dots, f_m\}$ de los estados.

Los Campos Aleatorios Markovianos son un campo de la teoría de probabilidad, los cuales proveen una herramienta para analizar espacial o contextualmente la dependencia de los fenómenos físicos.

En términos de homogeneidad y continuidad, se pueden clasificar los etiquetados en cuatro niveles:

Nivel 1. Estados homogéneos con etiquetas continuas.

Nivel 2. Estados homogéneos con etiquetas discretas.

Nivel 3. Estados no homogéneos con etiquetas continuas.

Nivel 4. Estados no homogéneos con etiquetas discretas.

La restauración de imágenes en escala de grises, pertenece al nivel 1. Un conjunto d de estados corresponden a los pixeles en la imagen, y un conjunto D de etiquetas es un intervalo real. La restauración no es más que estimar la imagen real de una imagen corrompida por ruido o degradada.

4.3 Vecinos y Cliques

Definición 4.1 Se define un *Sistema de Vecinos* para d como:

$$\mathcal{N} = \{\mathcal{N}_i | \forall i \in d\}, \quad (4.5)$$

donde \mathcal{N} es la colección de vecinos de i , que debe cumplir:

1. $i \notin \mathcal{N}_i$.
2. $i \in \mathcal{N}_j \iff j \in \mathcal{N}_i$.

Para una latice (rejilla) regular d , se define al conjunto de vecinos de i como el conjunto de estados próximos dentro del radio r .

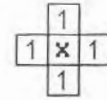
$$\mathcal{N} = \{j \in d / [dist(pixel_j, pixel_i)]^2 \leq r, j \neq i\}, \quad (4.6)$$

donde:

$dist(pixel_j, pixel_i)$, es la distancia euclideana entre el $pixel_i$ y el $pixel_j$.

r , es un número entero.

Para un sistema de vecinos de primer orden (4-vecinos)



, x tiene cuatro

vecinos y su distancia euclideana es 1.

Para un sistema de vecinos de segundo orden (8-vecinos)



, x tiene ocho

vecinos y su máxima distancia euclideana es 2.

El conjunto de estados de una rejilla regular $d = \{(i, j) \leq i, j \leq n\}$ corresponde a los pixeles de una imagen de dos dimensiones de tamaño $n \times n$.

Definición 4.2 Un clique c para (d, \mathcal{N}) es un subconjunto de d , tal que c consiste de un solo estado $c = \{i\}$, o un par de estados vecinos $c = \{i, j\}$, o tres estados vecinos $c = \{i, j, k\}$, y así sucesivamente.

Se puede denotar una colección de cliques por C_1, C_2, \dots , y la colección de todos los cliques para (d, \mathcal{N}) es $C = C_1 \cup C_2 \cup \dots$.

Por ejemplo, si se toma un sistema de vecinos de primer orden, ¿cuántos cliques diferentes existen?. La figura 4-1 muestra los diferentes cliques.

Para un sistema de vecinos de segundo orden, ¿cuántos cliques diferentes existen?. En la figura 4-2 se pueden observar.

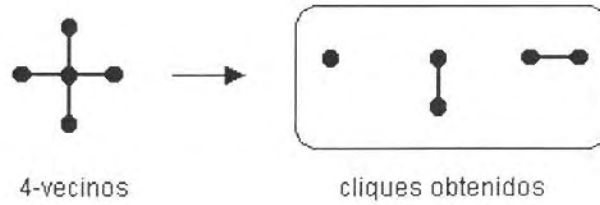


Figura 4-1: Cliques obtenidos a partir de un sistema de 4-vecinos

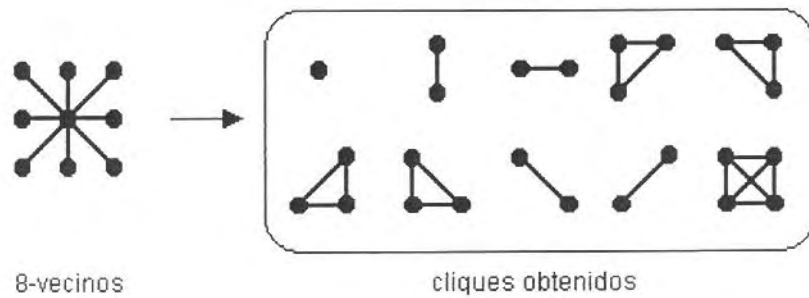


Figura 4-2: Cliques obtenidos a partir de un sistema de 8-vecinos

Definición 4.3 Una familia F de variables aleatorias, se dice que son un Campo Aleatorio Markoviano en d con respecto a \mathcal{N} si y sólo si se satisfacen las dos condiciones siguientes:

1. $P(F = f) > 0, \forall f \in S$.
2. $P(F_i = f_i | F_j = f_j, j \in d, j \neq i) = P(F_i = f_i | F_j = f_j, j \in \mathcal{N}_i)$.

Se dice que la probabilidad de un evento es condicionado por todos los sucesos restantes, es equivalente a aquella condicionada a los sucesos vecinos de i .

La probabilidad $P(F = f)$ para cualquier campo aleatorio está determinado únicamente por sus probabilidades condicionales locales.

4.4 Distribución de Gibbs

Teorema 4.1 F es un CAM en d con respecto a \mathcal{N} si y sólo si la probabilidad de distribución $P(F = f)$ en las configuraciones es una distribución de Gibbs¹ con respecto a \mathcal{N} .

Se puede definir una distribución de Gibbs de las configuraciones de f con respecto a \mathcal{N} de la siguiente forma:

$$P(f) = Z^{-1} \times e^{-\frac{1}{T}U(f)}, \quad (4.7)$$

donde:

Z , es una constante de normalización.

T , es un parametro de control global llamado temperatura.

¹De acuerdo al Teorema de Hammersley-Clifford

$U(f)$, es la función de energía.

La función de energía tiene la siguiente forma:

$$U(f) = \sum_{c \in C} V_c(f) = \sum_{\{i\} \in C_1} V_1(f_i) + \sum_{\{i,j\} \in C_2} V_2(f_i, f_j) + \dots, \quad (4.8)$$

donde:

$V_c(f)$ representa a todos los cliques posibles C , también se le conoce como *clique potencial*.

“...” denota los posibles términos de orden superior.

El valor de $V_c(f)$ dependerá de las configuraciones locales del clique c . El valor práctico del teorema es que provee una forma simple de especificar la probabilidad a priori $P(F = f)$ con las configuraciones, especificando el *clique potencial* $V_c(f)$ para todo $c \in C$.

4.5 Energía Posterior de los CAM

La función de verosimilitud a partir de (4.7) puede ser expresada de la siguiente forma:

$$P(F = f) = Z^{-1} \times e^{-U(f)}, \quad (4.9)$$

donde $U(f)$ es llamada la energía de verosimilitud. Entonces la probabilidad posterior es una distribución de Gibbs

$$P(F = f | g) = Z_E^{-1} \times e^{-E(f)}, \quad (4.10)$$

con energía posterior

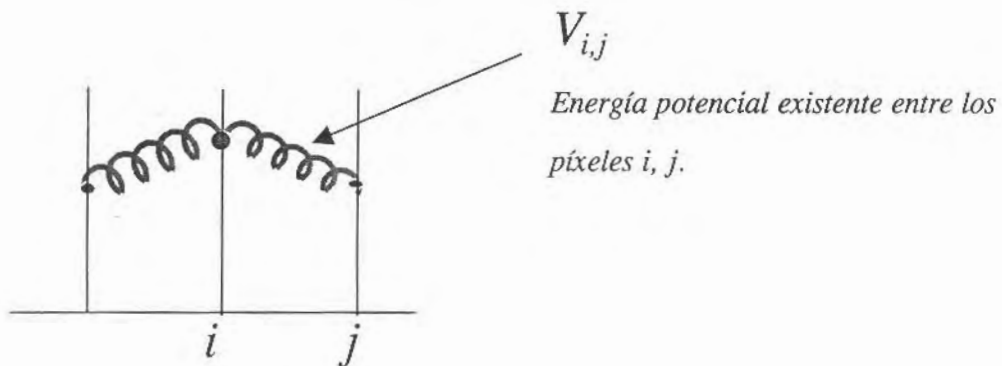
$$E(f) = U(f | g) = U(f) / T + U(g | f). \quad (4.11)$$

La solución del estimador MAP es equivalente a encontrar:

$$f^* = \arg \min_{f \in S} U(f | g). \quad (4.12)$$

4.6 Modelo de Resortes

Después de haber planteado los conceptos anteriores, se ve que el problema se puede acoplar a un modelo de resortes, en donde la energía potencial para un píxel i, j es $V_{i,j}$.



Sistema acoplado al modelo de resortes

La función $U(f)$ ecuación (4.8) se puede reescribir de la siguiente forma:

$$U(f) = \lambda \sum_{i,j} (f_i - f_j)^2, \quad (4.13)$$

entonces, la ecuación (4.7) se puede expresar como:

$$P(f) = \frac{1}{Z} e^{-\lambda \sum_{i,j} (f_i - f_j)^2} \quad (4.14)$$

De acuerdo al teorema de Bayes, la función de probabilidad a posteriori quedaría de la siguiente forma:

$$P(f | g) = \frac{1}{Z} e^{-\left(\sum (g_i - f_i)^2 + \lambda \sum_{i,j} (f_i - f_j)^2\right)} \quad (4.15)$$

lo que es equivalente a obtener:

$$\min U(f) = \frac{1}{2} \left(\sum (g_i - f_i)^2 + \lambda \sum_{i,j} (f_i - f_j)^2 \right). \quad (4.16)$$

Capítulo 5

Programación No Lineal Sin Restricciones

5.1 Introducción

La optimización para problemas no lineales es importante cuando se requiere la caracterización y localización de máximos y mínimos en funciones no lineales. En el Capítulo 2. Conceptos Básicos, se mencionaron brevemente algunos conceptos fundamentales dentro de la programación no lineal sin restricciones, tales como Optimización, Gradiente, Hessiana y Tipos de Mínimos, dichos conceptos serán retomados en este capítulo.

Dentro del campo de la teoría de optimización, se encuentra la programación matemática, la cual a su vez se divide en:

$$\text{Programación Matemática} \left\{ \begin{array}{l} \text{Programación Lineal} \\ \text{Programación Entera} \\ \text{Programación no lineal} \left\{ \begin{array}{l} \text{Con restricciones} \\ \text{Sin restricciones} \end{array} \right. \end{array} \right.$$

De acuerdo a la forma general de un problema de optimización (2.3), el problema principal de la programación matemática consistirá en elegir aquellos valores de x que

proporcionen ya sea el mayor o el menor valor de la función objetivo. La optimización es frecuentemente asociada con el costo, el cual es representado por un modelo matemático de un sistema bajo consideración.

Existen diversos algoritmos de optimización, de acuerdo al problema que se esté tratando hay métodos más apropiados para su solución. Es muy importante por lo tanto, reconocer las características de un problema para identificar una técnica apropiada de solución. Los problemas de Optimización son clasificados de acuerdo a las características matemáticas de la función objetivo, las restricciones y las variables de control.

Dentro de un problema de optimización, quizá la característica más relevante sea la naturaleza de la función objetivo. Una función es lineal si la relación entre la función $f(x)$ y las variables de control son de la siguiente forma:

$$f(x) = b^T x + c, \quad (5.1)$$

donde:

b es un vector de valores constantes,

c es una constante,

Una función es cuadrática si

$$f(x) = x^T A x + b^T x + c. \quad (5.2)$$

En el capítulo 1 se mencionaron los conceptos de mínimo local, mínimo global y mínimo local estricto, estos conceptos son necesarios para conocer las condiciones de optimalidad.

En la mayoría de los casos se tienen funciones objetivo cuyos valores se incrementan localmente en todas direcciones, esa es la definición de un mínimo local estricto; sin embargo, existen algunos puntos en donde los valores de la función permanecen localmente

en la misma dirección, pero se incrementan en algunos otros puntos, esto es a lo que se le conoce un mínimo local débil. En general, existen funciones en las cuales están presentes ambos tipos de mínimos, a los que se les nombra simplemente mínimo local.

Para que un punto x^* sea un mínimo local fuerte (estricto) de $f(x)$, se debe satisfacer que la matriz Hessiana sea definida positiva en x^* .

5.2 Condiciones Necesarias y Suficientes

Es importante tener condiciones las cuales sean declaradas de manera algebraica para poder comprobar si un punto dado es mínimo o no.

Condición necesaria de primer orden.

Si x^* es un mínimo local y f es continuamente diferenciable en una vecindad abierta de x^* , entonces: $\nabla f(x^*) = 0$.

Condición necesaria de segundo orden.

Si x^* es un mínimo local de $f(x)$, entonces $\nabla^2 f(x^*)$ es semidefinida positiva.

Condición suficiente de segundo orden.

Si $\nabla^2 f(x^*)$ es definida positiva, entonces x^* es un mínimo local.

5.3 Convergencia de los Algoritmos de Optimización

Existen dos factores que son fundamentales en el desarrollo de los algoritmos de optimización: estabilidad y velocidad de convergencia.

5.3.1 Estabilidad

Un algoritmo que provee la convergencia a un mínimo (en algún punto estacionario) sin tener en cuenta el punto inicial y no demora en puntos extraños, se dice que es estable o que presenta una convergencia global. En la optimización no lineal, la estabilidad es casi siempre asociada con esquemas iterativos, los cuales garantizan una reducción suficiente en el valor de la función por cada iteración, a menos que el mínimo haya sido localizado con un grado fijo de aproximación.

5.3.2 Velocidad de Convergencia

Si un método es estable, es importante entonces, tener en cuenta la velocidad de convergencia hacia una solución, dado que puede ser muy lento el rendimiento del método, reduciendo así su valor práctico, además de que un algoritmo debe poder llegar a la vecindad de un mínimo en tan pocas iteraciones como sea posible.

Al aplicar un algoritmo basado en un método iterativo, se está realizando una sucesión s_n de aproximaciones de la solución s buscada. Para poder medir la velocidad de convergencia, se define el *orden de convergencia* como el mayor número q tal que existe el límite definido por:

$$\lim_{n \rightarrow \infty} \frac{|s_{n+1} - s|}{|s_n - s|^q} \neq 0,$$

no es necesario que q exista, si existe, puede no ser un número entero.

En el caso cuando $q = 1$ se le llama *convergencia lineal*.

En el caso cuando $q = 2$ se le llama *convergencia cuadrática*.

Finalmente, si $q > 1$ se llama *convergencia superlineal*.

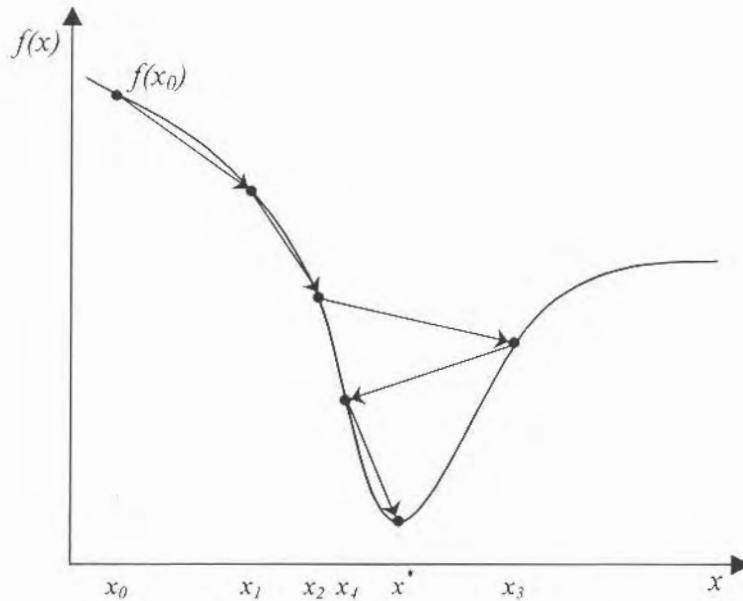


Figura 5-1: Proceso de búsqueda del mínimo.

5.4 Búsqueda Lineal

Cuando se cuenta con un método iterativo x_k es conocido, es necesario realizar una búsqueda a lo largo de un vector d_k para poder obtener un nuevo punto x_{k+1} , el cual puede ser hallado por $x_{k+1} = x_k + \alpha d_k$. Para poder calcular el tamaño de paso se puede realizar una búsqueda lineal exacta o una búsqueda lineal inexacta.

Con la búsqueda lineal lo que se pretende es encontrar una serie de soluciones prueba que conduzcan hacia una solución óptima en cada iteración, a través de las iteraciones se realiza una búsqueda que culmina en la identificación de una nueva solución prueba mejorada (ver figura 5-1).

La estructura fundamental para resolver problemas de minimización por medio de técnicas iterativas es simple. Se elige un punto inicial, una dirección de movimiento se define de acuerdo a algún algoritmo, se realiza una búsqueda lineal para determinar el

siguiente paso. El proceso se repite para un nuevo punto y el algoritmo continúa hasta que se encuentra un mínimo. Se puede expresar este procedimiento a través del siguiente algoritmo:

- Proponer un punto inicial x_0 .
- Para $k = 0, 1, 2, \dots$ hasta obtener la convergencia
 1. Probar la convergencia de x_k .
 2. Calcular una dirección de búsqueda d_k .
 3. Determinar un apropiado tamaño de paso¹ α_k .
 4. Establecer $x_{k+1} = x_k + \alpha_k d_k$.

5.5 Dirección de Descenso

Es razonable escoger un vector de búsqueda d_k que tenga una dirección de descenso, esto es, que de a la función una dirección en la que se reduzca. Una dirección de descenso d está definida como una a través de la cual la derivada direccional es negativa:

$$g(x)^T d < 0, \quad (5.3)$$

de donde se puede escribir la siguiente aproximación:

$$f(x + \alpha d) - f(x) \approx \alpha g(x)^T d,$$

lo anterior garantiza que un valor menor para la función puede ser encontrado por medio de d para α suficientemente pequeño.

¹ $\alpha_k = \min_{\alpha} f(x_k + \alpha d_k)$ para una búsqueda lineal exacta.

$\alpha_k \approx \min_{\alpha} f(x_k + \alpha d_k)$ para una búsqueda lineal inexacta.

Diversos métodos se distinguen por la elección de su dirección de búsqueda. A continuación se listan diversos métodos para determinar la dirección de descenso d_k :

- Descenso de Gradiente. $d_k = -\nabla f(x_k)$.
- Newton. $d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$.
- Cuasinewton. $d_k = -B_k^{-1} \nabla f(x_k)$.
- Gradientes Conjugados. $d_k = -\nabla f(x_k) + \beta_k d_{k-1}$.

5.6 Problemas Mal Condicionados y Bien Condicionados

Se puede clasificar a un problema como *mal condicionado* cuando un cambio “pequeño” en los datos produce un cambio “grande” en la solución del sistema, es decir, la solución de un sistema perturbado es “muy diferente” de la solución del sistema original.

Un problema se dice *bien condicionado* cuando cambios “pequeños” en los datos producen a su vez cambios “pequeños” en la solución.

Cabe aclarar que el hecho de que un problema esté mal o bien condicionado dependerá del problema mismo y no del algoritmo que se emplee para resolverlo.

Capítulo 6

Métodos de Gradientes

6.1 Introducción

Los métodos para la minimización de una función objetivo arbitraria $f(x)$ de n variables independientes sin restricciones forman el principal campo de estudio de la optimización no lineal.

Generalmente para los métodos iterativos, al inicio de la k -ésima iteración se denota a x_k como el estimado del mínimo actual, en donde se realiza el cálculo del vector de búsqueda d_k , a partir del cual se obtiene un nuevo estimado x_{k+1} , tal como y se expresa a continuación:

$$x_{k+1} = x_k + \alpha_k d_k, \quad (6.1)$$

donde α_k se obtiene a partir de una búsqueda lineal (ya mencionada en el capítulo anterior).

Las técnicas del gradiente permiten optimizar funciones continuas que son dos veces diferenciables. La idea general para este método es generar puntos sucesivos comenzando en un punto inicial dado, en la dirección de más rápido aumento de la función.

El proceso (6.1) involucra entonces una búsqueda a lo largo del vector d_k a partir del punto actual x_k , produciendo métodos iterativos estables, ya que se requiere un menor número de elecciones de α_k , entonces:

$$f(x_k + 1) < f(x_k), \quad (6.2)$$

a menos que el problema haya sido resuelto.

No todos los métodos garantizan que el d_k seleccionado satisfaga la ecuación (6.2). Algunos métodos pueden generar una secuencia de puntos que satisface (6.2) pero los cuales convergen a un punto no estacionario. Tales métodos no son muy usados para un propósito general pero es posible encontrar algunas modificaciones que superan la dificultad.

Debido a que muchos algoritmos usan la primera y algunos la segunda derivada de la función objetivo para calcular d_k , las derivadas pueden ser calculadas analíticamente o por algún tipo de aproximación. A éste tipo de métodos es a los que se les denomina *Métodos de Gradiente*.

6.2 Direcciones de Descenso

La idea central en el desarrollo de los métodos de gradiente es la dirección de descenso, esto es, un vector d_k para el cual la ecuación (6.2) se satisface con un valor suficientemente pequeño y positivo de α_k , como una dirección da la pendiente de x_k , se puede caracterizar d_k algebraicamente considerando las series de expansión de Taylor¹ para $f(x_{k+1})$ en términos de $f(x_k)$ y $\nabla f(x_k)$ cuando $\alpha_k \rightarrow 0$:

$$f(x_k + 1) = f(x_k + \alpha_k d_k) \simeq f(x_k) + \alpha_k \nabla f(x_k)^T d_k, \quad (6.3)$$

¹ Expansión de la serie de Taylor: $F(x + \Delta x) = F(x) + \Delta x^T g(x) + \frac{1}{2} \Delta x^T G(x) \Delta x + \dots$

con $\alpha_k > 0$ se tiene que:

$$\nabla f(x_k)^T d_k < 0, \quad (6.4)$$

esta relación se puede usar para definir la dirección de descenso d_k en el punto x_k . De forma más general, se puede decir que si el método genera direcciones de descenso (a menos que $\nabla f(x_k) = 0$), converge en un punto estacionario con una conveniente búsqueda lineal.

Un punto estacionario puede ser un punto silla, del cual es evidentemente posible hacer un movimiento a un punto con menor valor en la función. Si la segunda derivada de la función objetivo es factible, o alguna aproximación de ella, una de las mejores formas de moverse de un punto silla es a lo largo de la dirección de la curvatura negativa. La definición algebraica de una curvatura negativa es la siguiente:

$$d_k^T \nabla^2 f(x_k) d_k < 0. \quad (6.5)$$

6.3 Método de Descenso de Gradiente

La terminación de los métodos de gradiente se efectúa en el punto donde el vector gradiente se anula, la optimalidad no puede verificarse a menos que se conozca a priori si $f(x)$ es cóncava o convexa. La ecuación (6.3) se puede escribir como:

$$f(x_{k+1}) - f(x_k) = \alpha_k \|\nabla f(x_k)\| \|d_k\| \cos \theta, \quad (6.6)$$

donde θ puede ser interpretado geoméricamente como el ángulo entre $\nabla f(x_k)$ y d_k . Si se permite variar θ , dejando $\alpha_k \|\nabla f(x_k)\|$ y $\|d_k\|$ constantes, entonces el lado derecho de la ecuación (6.6) es más negativa cuando $\theta = \pi$. Esto para un valor lo suficientemente pequeño para α_k , la reducción más grande en los valores de la función se obtienen en la

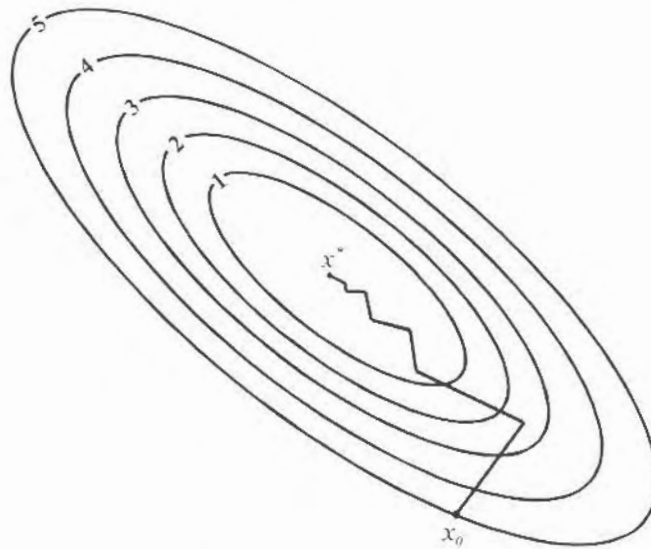


Figura 6-1: Proceso de búsqueda del óptimo por medio del método de Descenso de Gradiente.

dirección:

$$d_k = -\nabla f(x_k). \quad (6.7)$$

La dirección del gradiente negativo satisface el criterio de descenso (6.4) si $\nabla f(x_k) \neq 0$, es a lo que se le llama dirección de descenso en máxima escalada, y se emplea en la iteración (6.1) para dar un incremento al método de Descenso de Gradiente. La figura 6-1 ilustra gráficamente el progreso del método de máximo descenso.

6.3.1 Algoritmo

El procedimiento para poder hallar el valor para $f(x)$ óptimo se puede ver por medio del siguiente algoritmo:

1. Datos de entrada: x_0, ε (ε es un valor de tolerancia).

2. Para $k = 0, 1, \dots$ repetir

- $d_k = -\nabla f(x_k)$.
- calcular α_k .
- $x_{k+1} = x_k + \alpha_k d_k$.

3. Finalizar hasta que $\|\nabla f(x_{k+1})\| < \varepsilon$.

Este algoritmo fue programado en el trabajo de tesis, con el fin de hacer comparaciones contra algunos métodos de Gradientes Conjugados.

6.4 Método de Newton - Raphson

El método de Newton Raphson es un procedimiento iterativo que permite resolver ecuaciones simultáneas no lineales, éste método forma parte de los conocidos métodos de gradiente. Éste método está diseñado para obtener una terminación cuadrática con una matriz Hessiana definida positiva en una simple iteración, lo que representa una desventaja, ya que en la práctica, existen pocos problemas que tengan una matriz Hessiana definida positiva.

6.4.1 Algoritmo

1. Datos de entrada: x_0 , ε (ε es un valor de tolerancia).

2. Para $k = 0, 1, \dots$ repetir

- Resolver: $\nabla^2 f(x_k) d_k = -\nabla f(x_k)$
- $x_{k+1} = x_k + d_k$.

3. Finalizar hasta que $\|\nabla f(x_{k+1})\| < \varepsilon$.

Este método no fue programado, ya que el cálculo de la matriz Hessiana implica un gran costo computacional.

6.5 Gradientes Conjugados

Los métodos de Gradientes Conjugados son una alternativa a los métodos de Newton cuando la segunda derivada no es posible calcularla analíticamente, porque la aproximación de la matriz Hessiana por diferencias finitas requiere un considerable número de evaluaciones de gradiente. Este es un serio problema sólo si n es muy grande o el vector gradiente es computacionalmente caro. Estos métodos fueron propuestos inicialmente por Hestenes y Stiefel (1952) con el propósito de resolver sistemas de ecuaciones lineales, la aplicación en la optimización fue hecha por Fletcher y Reeves (1964). Éste tipo de métodos se caracterizan por ser menos robustos que los métodos de Cuasi-Newton, pero presentan la gran ventaja que el almacenamiento requerido para sus operaciones es muy modesto.

Para el método de gradientes conjugados, la primera iteración es la misma que en el método de Descenso de Gradiente. Como se ha mencionado anteriormente, el problema es minimizar una función f de n variables, donde f es suave y su gradiente se puede obtener fácilmente. Las iteraciones son de la forma

$$d_k = \begin{cases} -g_k & \text{para } k = 1, \\ -g_k + \beta_k d_{k-1} & \text{para } k \geq 2, \end{cases} \quad (6.8)$$

$$x_{k+1} = x_k + \alpha_k d_k, \quad (6.9)$$

donde β_k es un escalar, y α_k es el tamaño de paso obtenido, mediante una búsqueda unidimensional.

En el desarrollo de la tesis se emplearon para el cálculo de β_k las fórmulas de Fletcher-Reeves (FR) [8],[6], Polak-Ribiere (PR) [8],[6], Hestenes-Stiefel (HS) [8],[6] y Polak-Ribiere positivo (PR+) [8]:

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}. \quad (6.10)$$

$$\beta_k^{PR} = \frac{\langle g_k, g_k - g_{k-1} \rangle}{\|g_{k-1}\|^2}. \quad (6.11)$$

$$\beta_k^{HS} = \frac{\langle g_k, g_k - g_{k-1} \rangle}{\langle d_{k-1}, g_k - g_{k-1} \rangle}. \quad (6.12)$$

$$\beta_k^{PR+} = \max \{ \beta_k^{PR}, 0 \}. \quad (6.13)$$

donde $\langle \cdot, \cdot \rangle$, es el producto escalar usual y $\|\cdot\|$, es la norma euclídeana.

Según estudios realizados, el rendimiento numérico del método de Fletcher-Reeves es algo impredecible: es a veces tan eficiente como los métodos de Polak-Ribiere y Hestenes-Stiefel, pero frecuentemente es mucho más lento. Según Powell, bajo algunas circunstancias, el método de Fletcher-Reeves con búsqueda lineal exacta producirá muy pequeños desplazamientos, y normalmente no se recupera, a menos que se realice un reinicio a través de la dirección del gradiente.

Los métodos de Hestenes-Stiefel y Polak-Ribiere parecen desempeñarse muy similarmente en la práctica, y han sido preferidos sobre el método de Fletcher-Reeves. No obstante, Powell [12] fue capaz de mostrar que el método de Polak-Ribiere con búsqueda lineal exacta puede ciclarse indefinidamente sin aproximarse al punto solución. El mismo resultado se aplica al método de Hestenes-Stiefel, ya que los dos métodos son idénticos cuando $\langle g_k, d_{k-1} \rangle = 0$, lo cual se mantiene cuando la búsqueda lineal es exacta.

El método de Polak-Ribiere Positivo (PR+), es interesante debido al reinicio que emplea, como se puede notar de (6.8) y (6.13), cuando $\beta_k^{PR} < 0$, se reduce al método de Descenso de Gradiente.

Se implementó también un método Híbrido [8] que combina las fórmulas de Fletcher-Reeves y la de Polak-Ribiere tomando la mejor parte de ambas, de forma que:

$$\beta_k = \begin{cases} -\beta_k^{FR} & \text{si } \beta_k^{PR} < -\beta_k^{FR}, \\ \beta_k^{PR} & \text{si } |\beta_k^{PR}| \leq \beta_k^{FR}, \\ \beta_k^{FR} & \text{si } \beta_k^{PR} \leq \beta_k^{FR}. \end{cases} \quad (6.14)$$

con esta estrategia se evita una de las principales desventajas del método de FR que es el retardo en alcanzar la solución, ya que los pasos se vuelven muy pequeños y este comportamiento puede continuar por un gran número de iteraciones, a menos que exista un reinicio, como es este el caso.

6.5.1 Búsqueda Lineal

Para la instrumentación de los algoritmos antes mencionados, se empleó la búsqueda lineal con las condiciones de Armijo, tal proceso se puede resumir a continuación:

1. Elegir $\bar{\alpha} > 0, p, c \in (0, 1)$.
2. Asignar $\alpha = \bar{\alpha}$.
3. Repetir hasta que $f(x_k + \alpha d_k) \leq f(x_k) + c\alpha \nabla f(x_k)^T d_k$

$$\alpha = p\bar{\alpha}.$$

$$\bar{\alpha} = \alpha$$

4. $\alpha_k = \alpha$.

Capítulo 7

Algoritmo Propuesto

Una vez que se han reunido los elementos para poder desarrollar un modelo que permita realizar la restauración de imágenes (Figura 7-1), se plasman los algoritmos que fueron implementados para poder realizar las comparaciones pertinentes entre ellos.

De acuerdo a lo mencionado anteriormente, la función que se desea minimizar es:

$$U(f) = \frac{1}{2} \left[\sum_i (f_i - g_i)^2 + \lambda \sum_{i,j} (f_i - f_j)^2 \right], \quad (7.1)$$

la cual, para un sistema de 4-vecinos se puede representar como:

$$U(f) = \frac{1}{2} \min \sum_{i,j=1}^m [(f_{ij} - g_{ij})^2 + \lambda [(f_{ij} - f_{ij-1})^2 + (f_{ij} - f_{i-1j})^2 + (f_{ij} - f_{ij+1})^2 + (f_{ij} - f_{i+1j})^2]], \quad (7.2)$$

para este caso, g_{ij} son las observaciones.

Por otro lado, de la ecuación (7.1) se obtiene su derivada e igualándola a cero queda como:

$$f_i (1 + \lambda |N_i|) - g_i - \lambda \sum_{f_j \in N_i} f_j = 0, \quad (7.3)$$

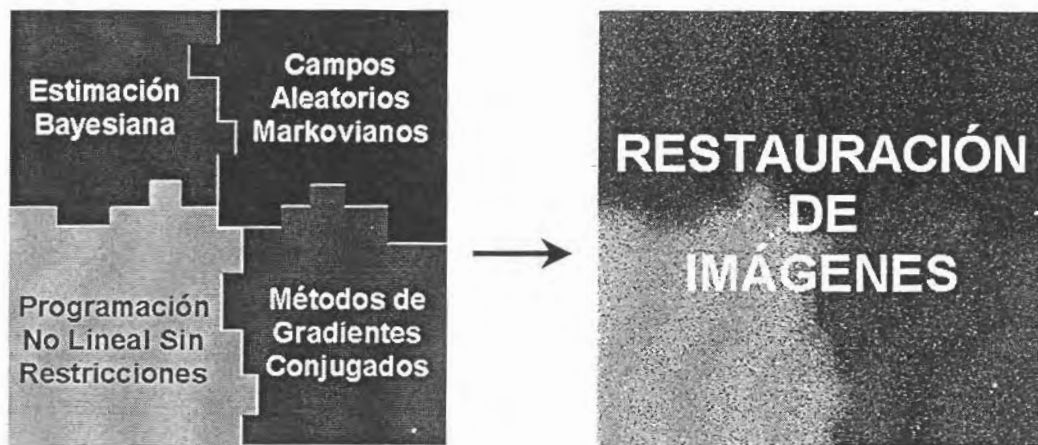


Figura 7-1: Elementos de la Restauración de Imágenes

donde:

N_i son los vecinos de i .

$|N_i|$ representa el número de vecinos de i .

g_{ij} son las observaciones iniciales.

Para un sistema de 4-vecinos, que será la vecindad empleada para éste trabajo de tesis, la ecuación (7.3) se puede escribir como:

$$\frac{\partial U(f)}{\partial f_{ij}} = f_{ij}(1 + 4\lambda) - g_{ij} - \lambda(f_{ij-1} + f_{i+1j} + f_{i-1j} + f_{ij+1}). \quad (7.4)$$

en base a esto, se pueden plantear los algoritmos que permitirán realizar la restauración de imágenes.

7.1 Descenso de Gradiente

1. ImOr \leftarrow Observaciones.
2. $F_k = \text{ImOR}$, $\alpha = 0.1$, $it=0$.

3. $G_k = \text{Gradiente}(F_k)$.
4. Para... (norma > 0.001 and it < 100)
 - $d_k = -G_k$.
 - $F_{k+1} = F_k + \alpha d_k$.
 - $G_k = \text{Gradiente}(F_k)$.
 - it = it + 1, norma = $\|G_k\|$.

7.2 Métodos de Gradientes Conjugados

Los métodos de Gradientes Conjugados siguen el algoritmo listado a continuación, variando únicamente en la forma de obtener la β , se muestra el procedimiento y posteriormente se detalla la forma en que se obtiene la β .

1. ImOr \leftarrow Observaciones.
2. $F_k = \text{ImOR}$, $\varepsilon \leftarrow$ Leer valor para tolerancia.
3. $G_k = \text{Gradiente}(F_k)$.
4. $d_k = -G_k$.
5. $UF_k = U(F_k)$.
6. Para it = 0 hasta ($|UF_k - UF_{k-1}| > \varepsilon$ and it < 100)
 - $\alpha = \text{BusqLin}(F_k, d_k, G_k, \text{ImOr})$.
 - $F_k = F_k + \alpha d_k$.
 - $G_k = \text{Gradiente}(F_k)$.

- $\beta = \text{Beta de Método de Gradientes Conjugados.}$
- $d_k = -G_k + \beta d_k.$
- $UF_k = U(F_k).$
- $it = it + 1.$

7.2.1 Cálculo del Valor β

A continuación se lista el procedimiento general para calcular el valor β de Fletcher-Reeves [8],[6], Polak-Ribiere [8],[6] y Hestenes-Stiefel [8],[6]. Para calcular la β de Polak-Ribiere positiva (PR⁺) [8], existe una variación a este procedimiento, de igual forma para el cálculo del β Híbrido [8]; tales variaciones serán descritas posteriormente.

1. $alto \leftarrow$ alto de la imagen.
2. $ancho \leftarrow$ ancho de la imagen.
3. Para $j = 0$ mientras $j < alto$

- Para $i = 0$ mientras $i < ancho$

// El valor de Numerador y Denominador será definido más

// adelante acorde a cada método.

Calcular Numerador.

Calcular Denominador.

$i = i + 1.$

- $j = j + 1.$

4. $\beta = \text{Numerador/Denominador.}$

β para Fletcher-Reeves

$$\text{Numerador} = (G_k[i, j])^2.$$

$$\text{Denominador} = (G_{k-1}[i, j])^2.$$

β para Polak-Ribiere

$$\text{Numerador} = G_k[i, j] (G_k[i, j] - G_{k-1}[i, j]).$$

$$\text{Denominador} = (G_{k-1}[i, j])^2.$$

β para Hestenes-Stiefel

$$\text{Numerador} = G_k[i, j] (G_k[i, j] - G_{k-1}[i, j]).$$

$$\text{Denominador} = d_k[i, j] (G_k[i, j] - G_{k-1}[i, j]).$$

7.2.2 Cálculo de β para PR^+

1. *alto* \leftarrow alto de la imagen.
2. *ancho* \leftarrow ancho de la imagen.
3. Para $j = 0$ mientras $j < \text{alto}$

- Para $i = 0$ mientras $i < \text{ancho}$

$$\text{Numerador} = G_k[i, j] (G_k[i, j] - G_{k-1}[i, j]).$$

$$\text{Denominador} = (G_{k-1}[i, j])^2.$$

$$i = i + 1.$$

- $j = j + 1.$

4. Si $(\text{Numerador}/\text{Denominador}) < 0$

entonces $\beta = 0$

sino $\beta = \text{Numerador}/\text{Denominador}$.

7.2.3 Cálculo de β Híbrido de FR y PR

1. $\text{alto} \leftarrow$ alto de la imagen.

2. $\text{ancho} \leftarrow$ ancho de la imagen.

3. Para $j = 0$ mientras $j < \text{ancho}$

• Para $i = 0$ mientras $i < \text{ancho}$

$$\text{Elem1} = (G_k[i, j])^2.$$

$$\text{Elem2} = (G_{k-1}[i, j])^2.$$

$$\text{Elem3} = G_k[i, j] (G_k[i, j] - G_{k-1}[i, j]).$$

$$i = i + 1.$$

• $j = j + 1$.

4. $\beta^{PR} = \text{Elem1} / \text{Elem2}$.

5. $\beta^{PR} = \text{Elem3} / \text{Elem2}$.

6. Si $\beta^{PR} < -\beta^{FR}$

entonces $\beta = -\beta^{FR}$.

7. Si $|\beta^{PR}| \leq \beta^{FR}$

entonces $\beta = \beta^{PR}$.

8. Si $\beta^{PR} \leq \beta^{FR}$

entonces $\beta = \beta^{FR}$.

7.3 Funciones Comunes

7.3.1 Gradiente

// El valor del gradiente se calcula de acuerdo a la ecuación (7.4).

1. $\lambda \leftarrow$ Leer valor deseado para λ .
 2. ancho \leftarrow Ancho de la imagen.
 3. alto \leftarrow Alto de la imagen.
 4. $F \leftarrow$ Imagen en la iteración actual.
 5. $Obs \leftarrow$ Imagen observada.
 6. *Calcular gradiente del pixel de la esquina superior izquierda*
 7. Para $i = 1$ hasta $i < \text{ancho} - 1$
 - *Calcular gradiente de los pixeles del primer renglón sin esquinas*
 - $i = i + 1$.
 8. *Calcular gradiente del pixel de la esquina superior derecha*
- //Renglones intermedios de la imagen
9. Para $j = 1$ hasta $j < \text{alto} - 1$

- *Calcular gradiente de los pixeles de la columna izquierda*

- Para $i = 1$ hasta $i < \text{ancho} - 1$

Calcular gradiente de los pixeles de la imagen sin líneas de borde.

$i = i + 1.$

- *Calcular gradiente de los pixeles de la ultima columna (derecha)*

- $j = j + 1.$

10. *Calcular gradiente del pixel de la esquina inferior izquierda*

11. Para $i = 1$ hasta $i < \text{ancho} - 1$

- *Calcular gradiente del último renglón, sin esquinas*

- $i = i + 1.$

12. *Calcular gradiente del pixel de la esquina inferior derecha*

7.3.2 U(F)

// El valor de $U(f)$ está dado por la ecuación (7.1).

1. $\lambda \leftarrow$ Leer valor deseado para λ .
2. $\text{ancho} \leftarrow$ Ancho de la imagen.
3. $\text{alto} \leftarrow$ Alto de la imagen.
4. $F \leftarrow$ Imagen en la iteración actual.
5. $\text{Obs} \leftarrow$ Imagen observada.
6. *Calcular $U(f)$ para el pixel esquina superior izquierda*

7. Para $i = 1$ hasta $i < \text{ancho} - 1$

- Calcular $U(f)$ para los píxeles del primer renglón sin esquinas.
- $i = i + 1$.

Calcular $U(f)$ para el píxel de la esquina superior derecha

//Renglones intermedios de la imagen

8. Para $j = 1$ hasta $j < \text{alto} - 1$

- Calcular $U(f)$ para los píxeles de la columna izquierda
- Para $j = 1$ hasta $j < \text{alto} - 1$

Calcular $U(f)$ para los píxeles de la imagen sin líneas de borde

$j = j + 1$.

- Calcular $U(f)$ para los píxeles de la última columna (derecha)

9. Calcular $U(f)$ para el píxel de la esquina inferior izquierda

10. Para $i = 1$ hasta $i < \text{ancho} - 1$

- Calcular $U(f)$ para los píxeles del último renglón, sin esquinas
- $i = i + 1$.

11. Calcular $U(f)$ para el píxel de la esquina inferior derecha

7.3.3 Búsqueda Lineal

1. $\lambda \leftarrow$ Leer valor deseado para λ .
2. $\text{ancho} \leftarrow$ Ancho de la imagen.

3. $\text{alto} \leftarrow$ Alto de la imagen.
4. $F_k \leftarrow$ Leer valores de imagen en la iteración actual.
5. $d_k \leftarrow$ Leer dirección de descenso en la iteración actual.
6. $G_k \leftarrow$ Leer gradiente en la iteración actual.
7. $Obs \leftarrow$ Leer imagen observada (original).
//Pixel esquina superior izquierda
8. $\bar{\alpha} = 0.5$.
9. $p = 0.5$.
10. mientras $U(F_k + \bar{\alpha}d_k) > U(F_k) + 0.5\bar{\alpha}G_kd_k$
entonces $\bar{\alpha} = p\bar{\alpha}$.
11. $\alpha_k = \bar{\alpha}$.

Capítulo 8

Pruebas

Una vez que se instrumentaron los algoritmos descritos en el capítulo anterior, es necesario realizar las pruebas pertinentes para poder decidir cuál de estos algoritmos se comporta mejor ante la restauración de imágenes. Para ello, es necesario escoger un valor de λ apropiado, este valor es elegido “a prueba y error”, esto debido a que el cálculo de un valor óptimo se vuelve un problema intratable en la optimización. En este caso, después de seleccionar diversos valores, se puede observar que:

- Valores de λ menores que 0.3 no afectan significativamente los resultados, dejando la imagen ruidosa casi sin modificaciones.
- Valores mayores a 1 para λ suelen distorsionar la imagen, el aplanamiento de la misma es significativo, hay pérdida de bordes.

Por esto, después de realizar varias pruebas en diversas imágenes, el valor seleccionado para λ fue de 0.65, con este valor, el desempeño en la mayoría de los algoritmos fue óptimo. El valor del ε es de 0.001, valores más pequeños no efectúan cambios visibles en la imagen.

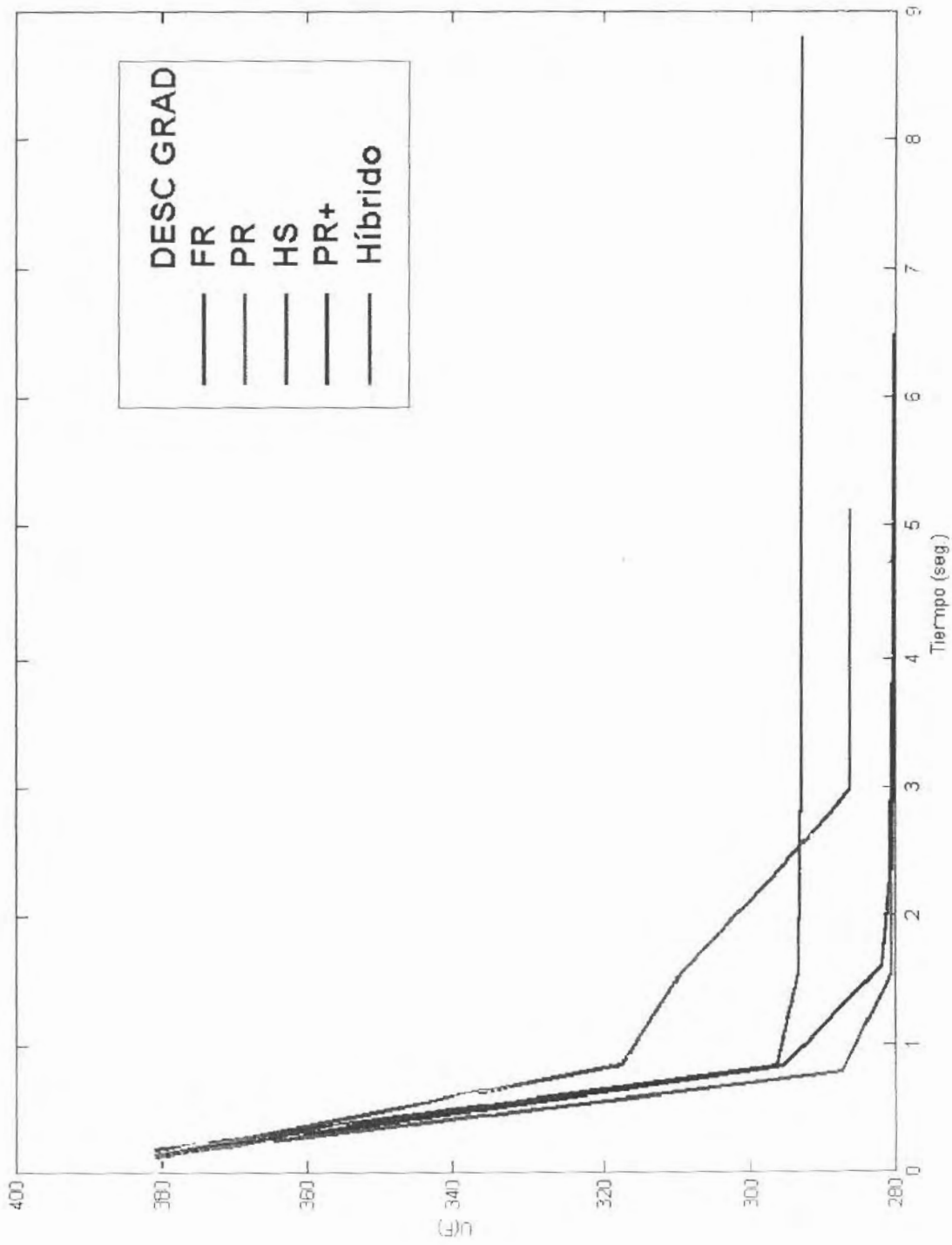
Las pruebas consisten en medir cuál de los algoritmos tiene mayor rapidez de convergencia, es decir, se mide el valor de $U(F)$ contra el tiempo empleado en la restauración de imágenes reales y sintéticas.

Para realizar las pruebas se empleó una computadora Pentium III 450Mhz con sistema operativo Windows 98.

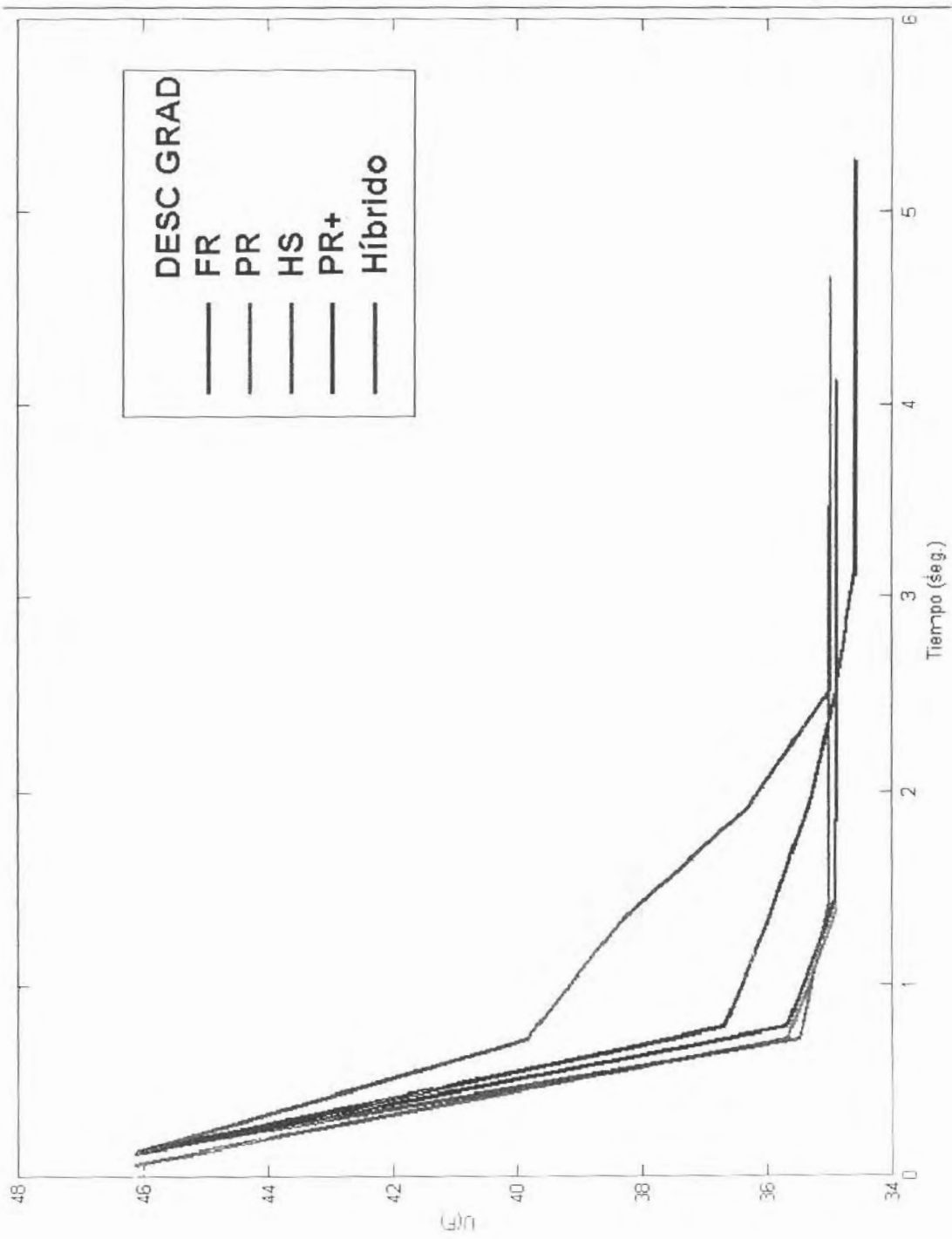
Las imágenes de prueba tienen dimensiones de 128×128 pixeles con formato bmp en escala de grises. Las imágenes ruidosas fueron generadas con el programa Caliman [11], a estas se les añadió ruido gaussiano con varianza 1.

Cabe recalcar que el tiempo empleado en la restauración depende del equipo en el que se efectúen las pruebas. Pero las gráficas que se muestran a continuación dan una idea muy clara del comportamiento de cada algoritmo, ya que éste será el mismo independientemente del equipo empleado.

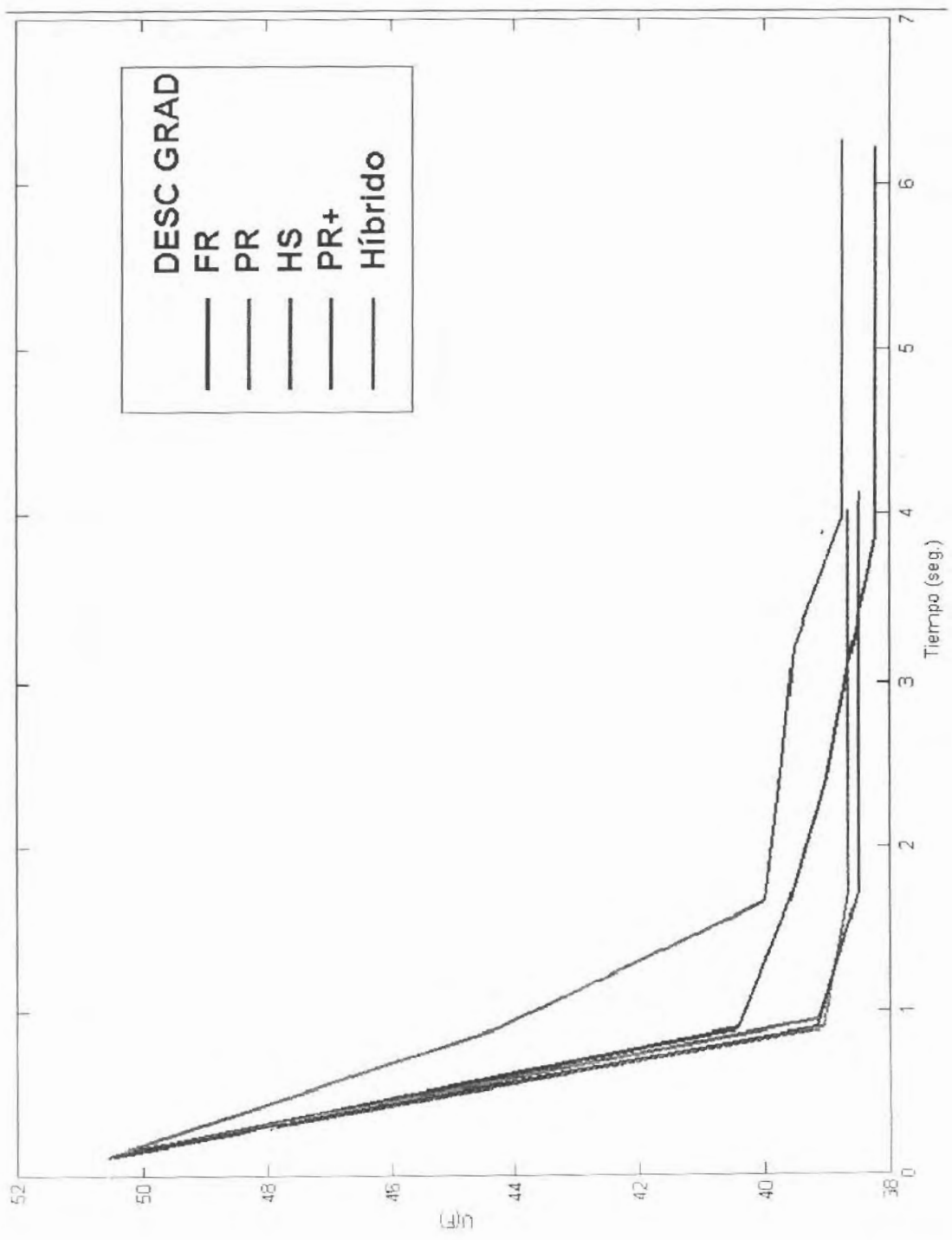
Posteriormente a las gráficas, se encuentran los resultados visuales de las cuatro imágenes de prueba. En las imágenes mostradas, se pueden comparar los resultados de los algoritmos entre sí, también contra algunos filtros de operadores de ventana (Media y Mediana).



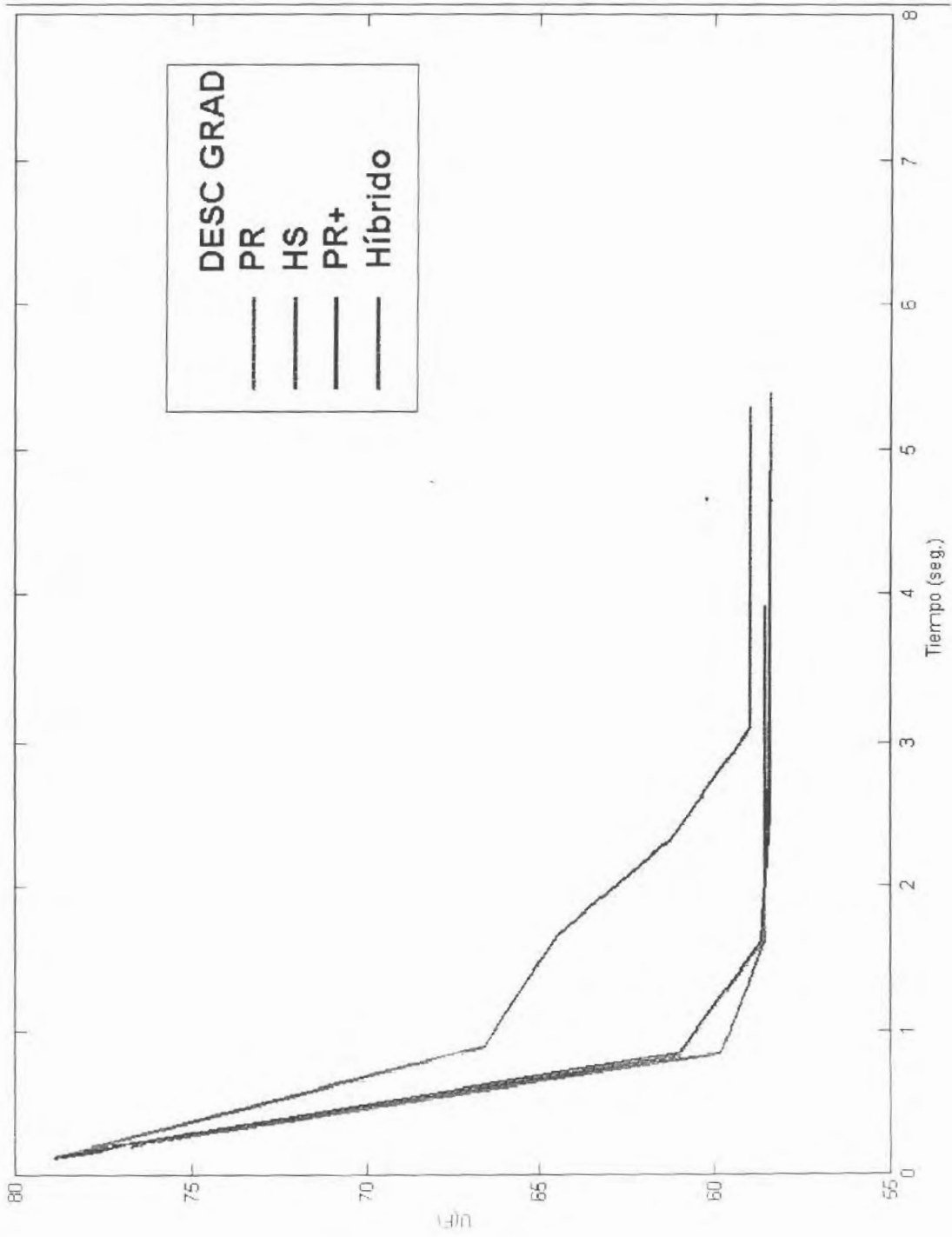
Gráfica 8.1: Aquí se muestra cómo se comportaron los diversos algoritmos para la restauración de la imagen de una muestra sanguínea. Cabe destacar que la gráfica del método de FR fue truncada, esto debido a que el tiempo que empleó fue mayor a los 10 segundos.



Gráfica 8.2: En esta gráfica se muestra el comportamiento de los diferentes algoritmos. Todos muestran una velocidad de convergencia aceptable para la imagen de la mano con el cascanueces.



Gráfica 8-3: Gráfica que muestra los resultados para los distintos métodos en la restauración de la imagen sintética de las franjas, el comportamiento de los algoritmos fue muy similar.



Gráfica 8-4: En la restauración de la imagen sintética de la estrella, el comportamiento de los algoritmos PR, PR+ y HS fue muy similar. No se incluye la gráfica de FR, ya que mostró un retardo que iba más allá de 1 minuto en la restauración de esta imagen, además de que los resultados visuales fueron muy pobres.

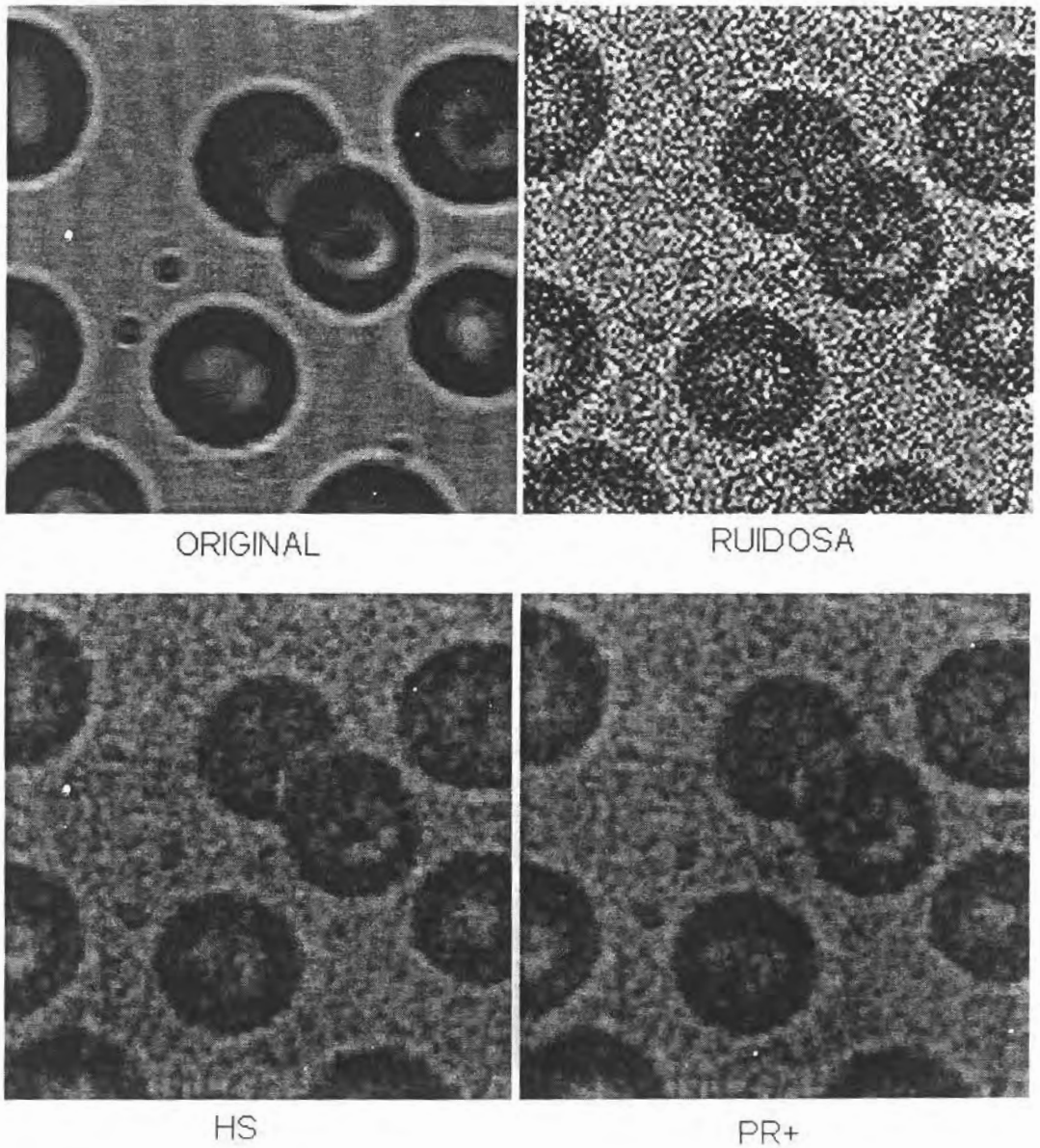
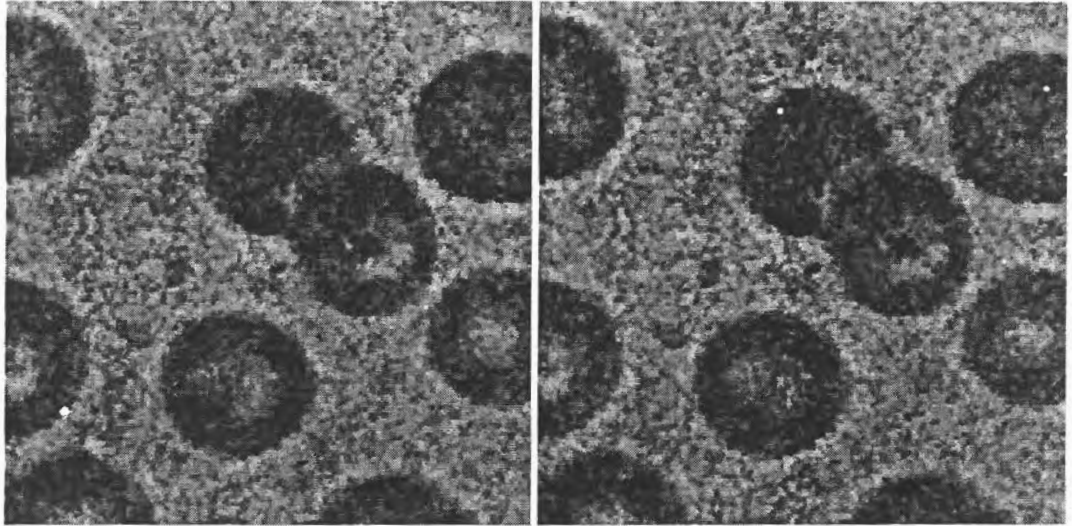
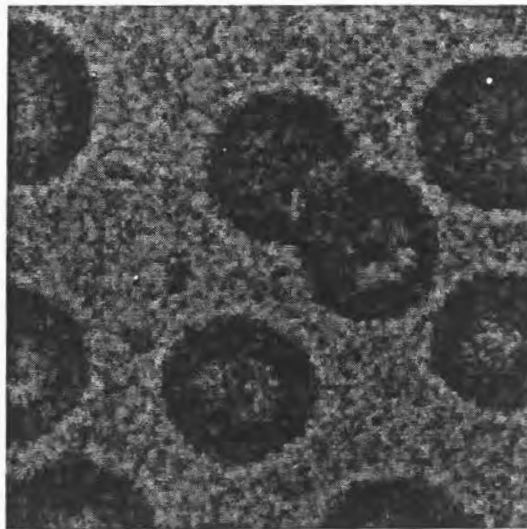


Figura 8-1: Imagen de una muestra sanguínea, mostrando las restauraciones de los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.



HIBRIDO

DESCENSO DE GRADIENTE



FR

Figura 8-2: Como se puede observar, los resultados obtenidos por los métodos Híbrido, Descenso de Gradiente y Fletcher-Reeves son muy similares, lo que es importante analizar en este caso es su velocidad de convergencia (Gráfica 8-1).

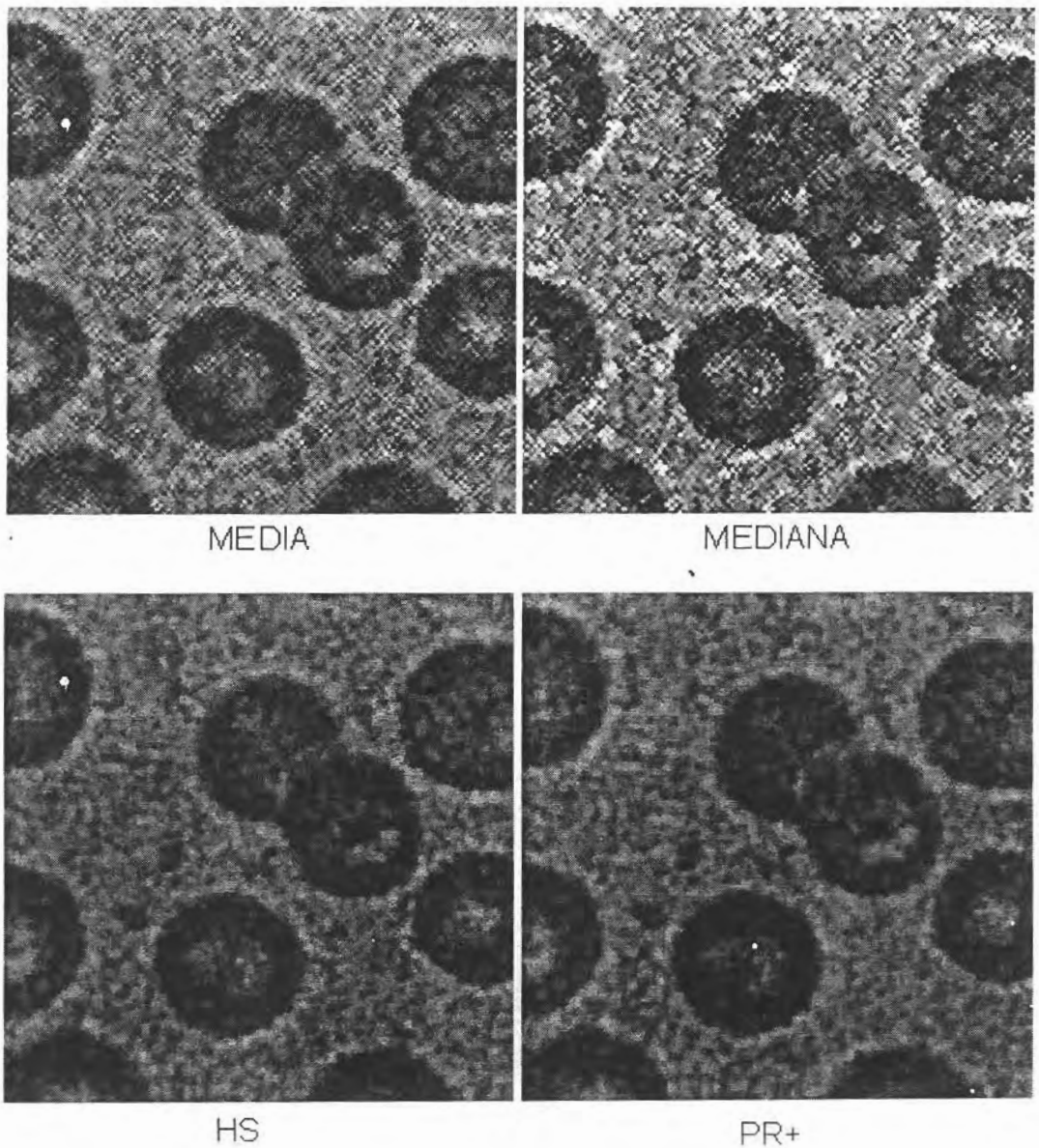


Figura 8-3: imagen de la muestra sanguínea, mostrando las restauraciones de los filtros de media y mediana, comparando los resultados con los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.

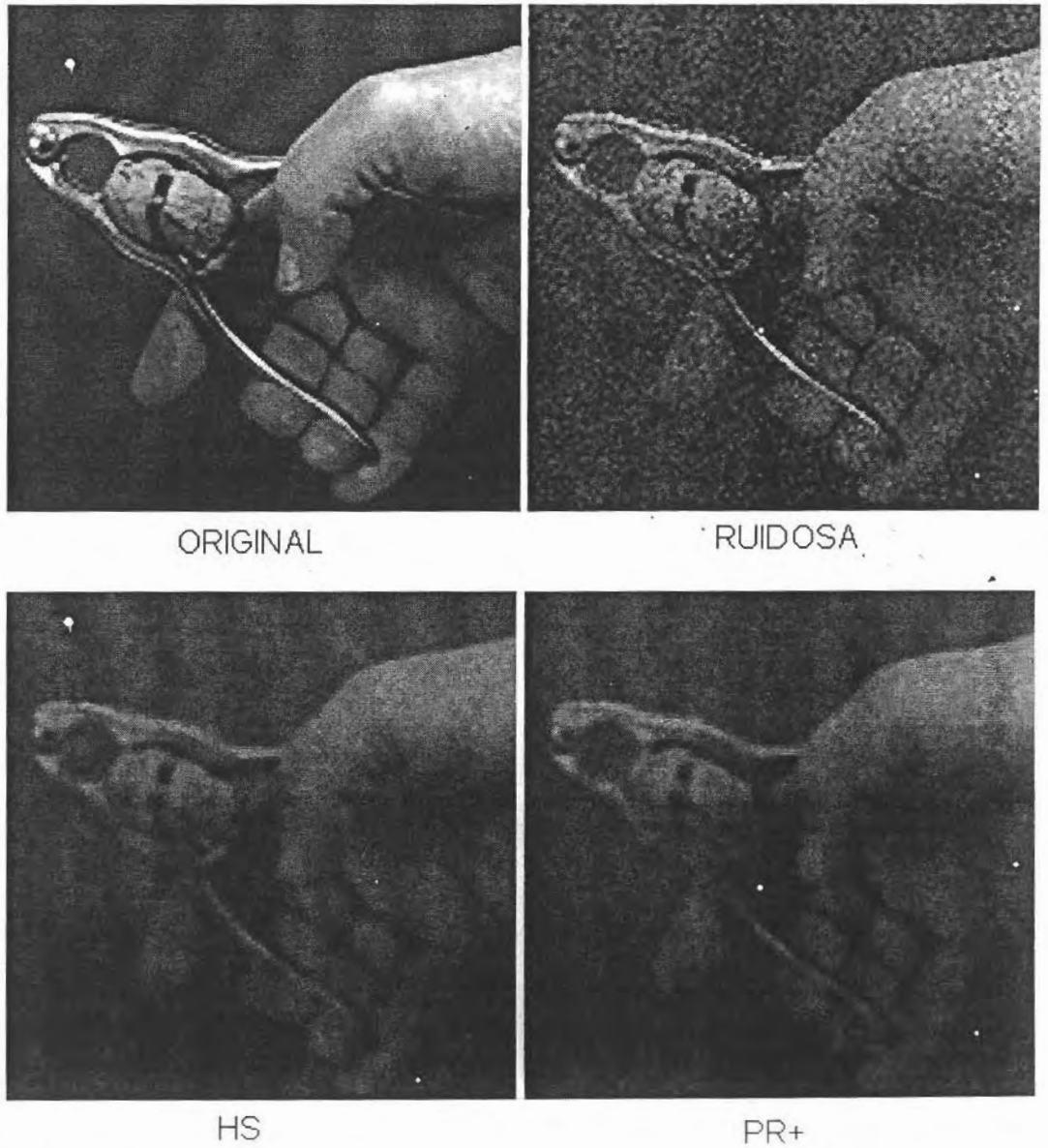


Figura 8-4: Imagen de una mano sujetando un cascavetes, mostrando las restauraciones de los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.



HIBRIDO

DESCENSO DE GRADIENTE



FR

Figura 8-5: Los resultados visuales que se obtienen para la imagen de la mano con el cascanueces con los métodos Híbrido, Descenso de Gradiente y Fletcher-Reeves son muy similares, ver su velocidad de convergencia (Gráfica 8-2).

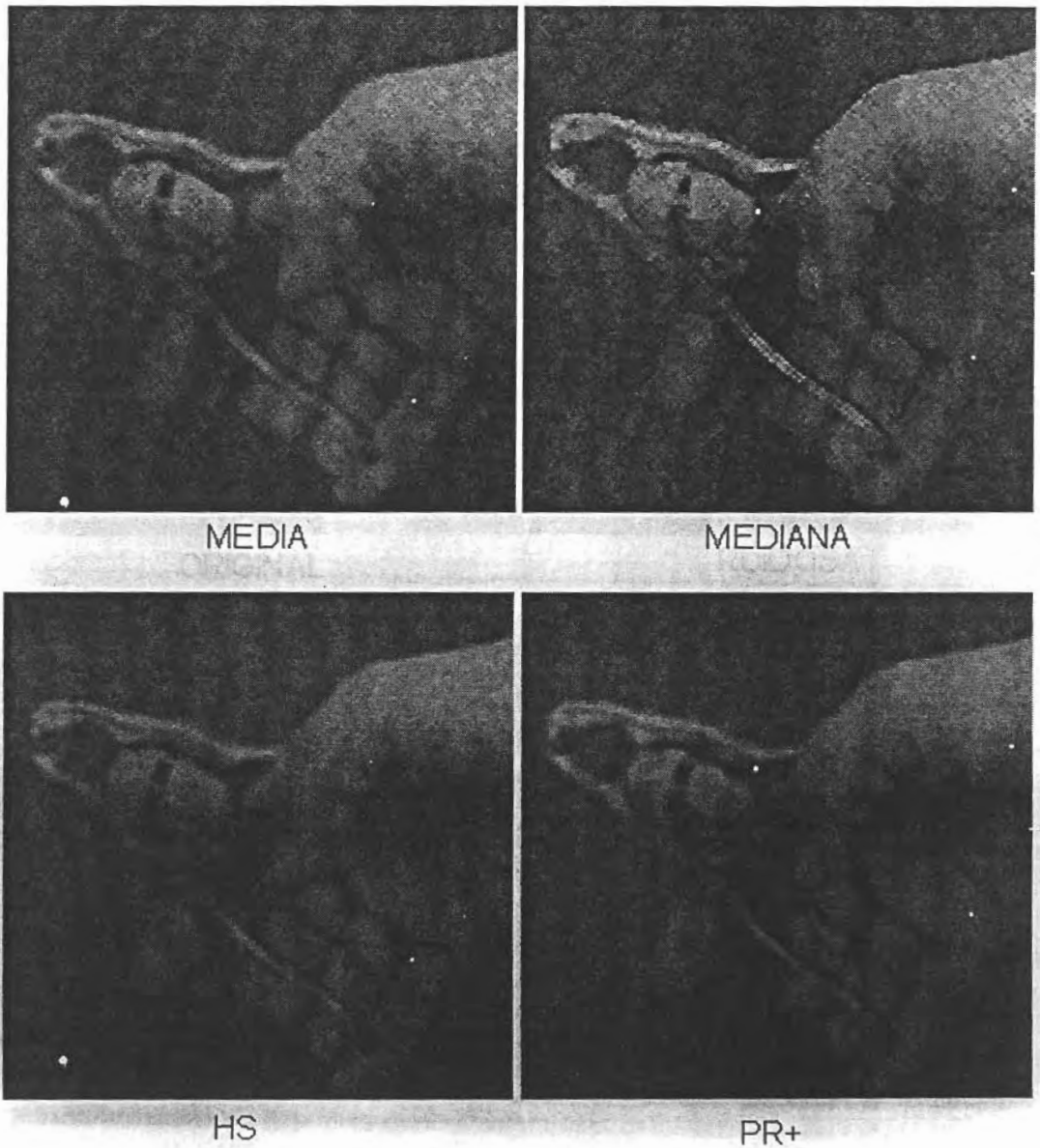


Figura 8-6: Imagen de la mano sujetando un cascanueces, muestra las restauraciones de los filtros de media y mediana, comparando los resultados con los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.

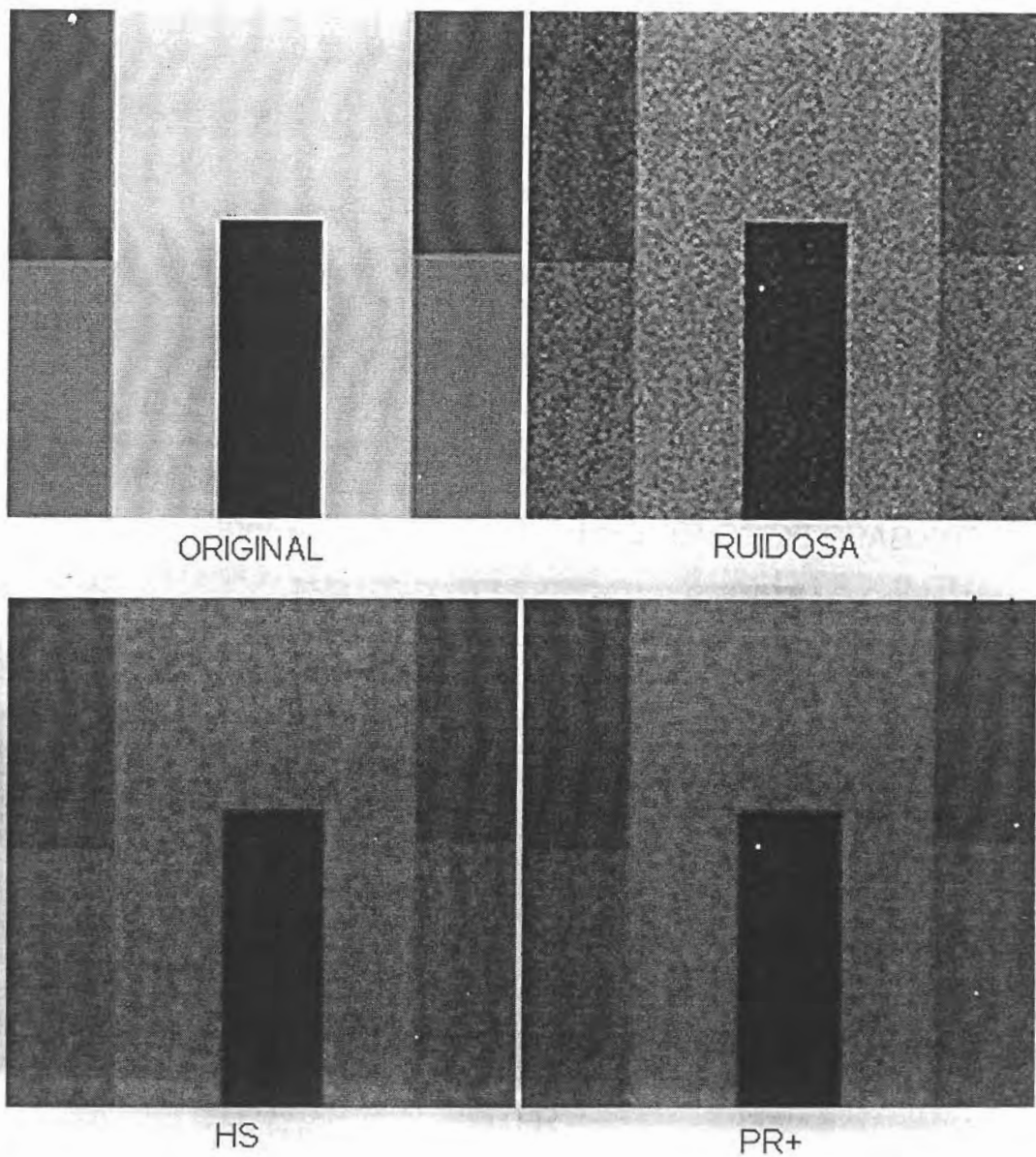


Figura 8-7: Imagen sintética de franjas en tonos de gris, muestra las restauraciones de los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.

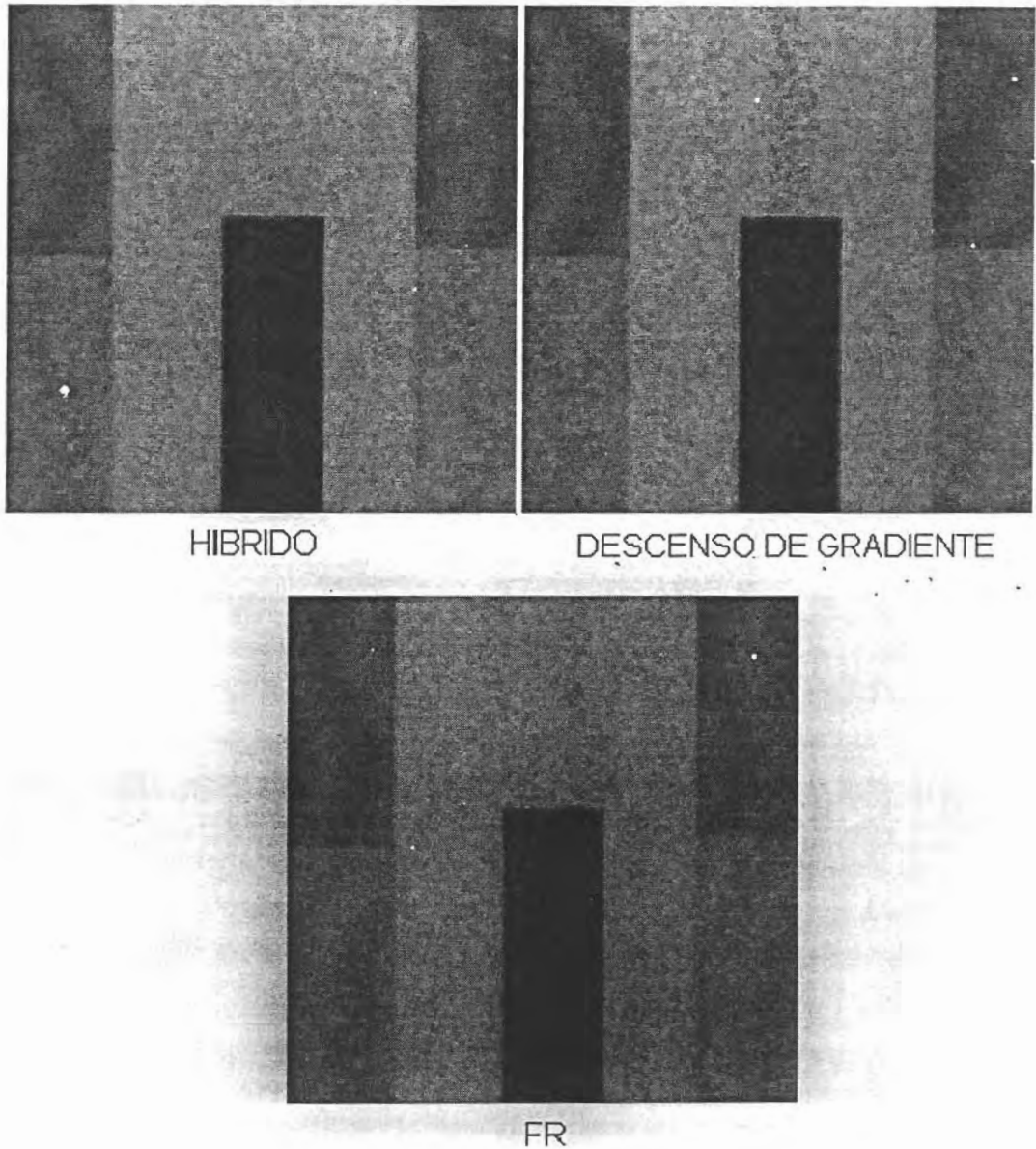


Figura 8-8: Resultados de la restauración para la imagen sintética de las franjas en escala de grises por los métodos Híbrido, Descenso de Gradiente y Fletcher-Reeves, estos son muy similares, ver su velocidad de convergencia (Gráfica 8-3).

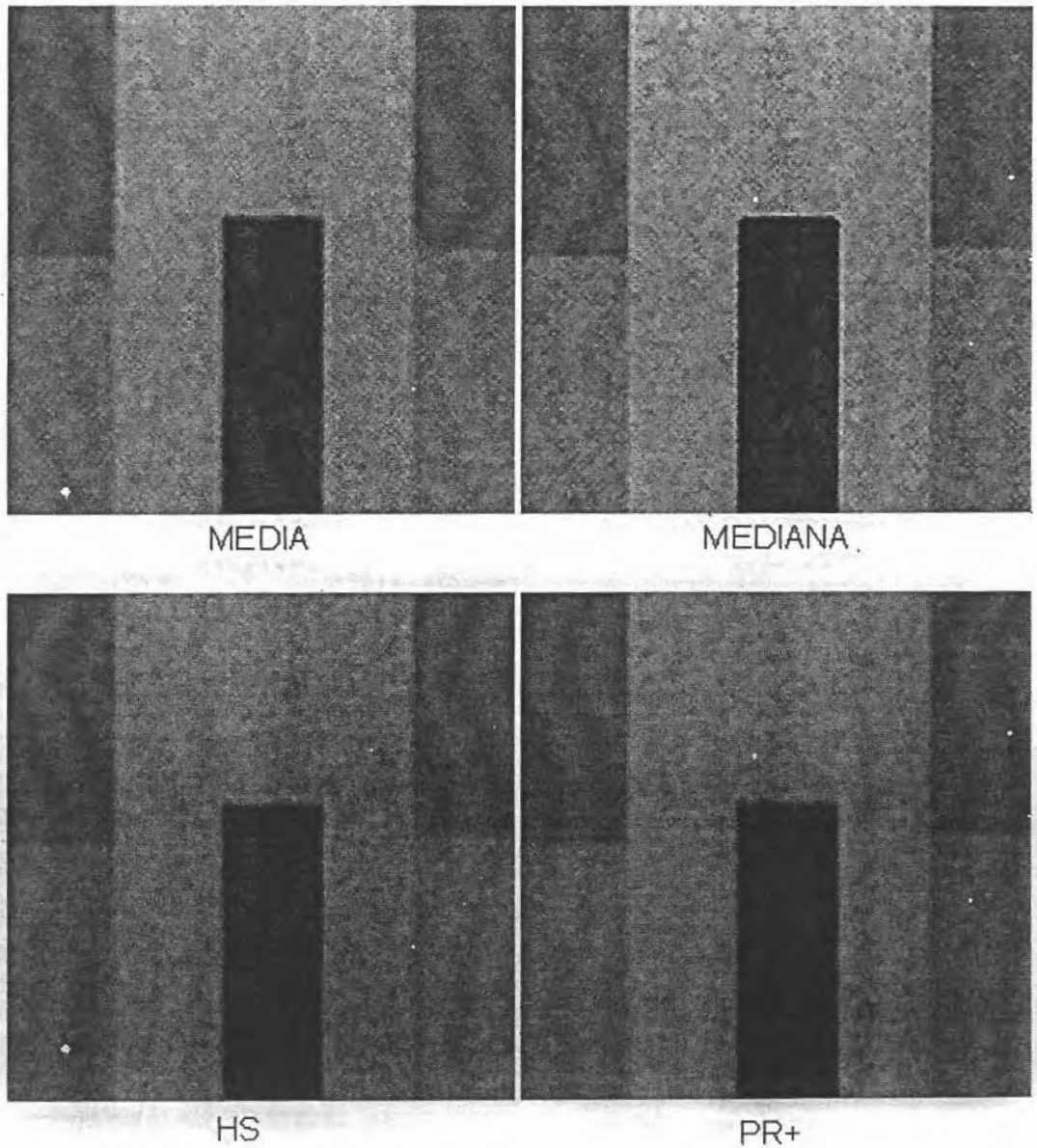


Figura 8-9: Imagen sintética con franjas, muestra las restauraciones de los filtros de media y mediana comparando los resultados con los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.

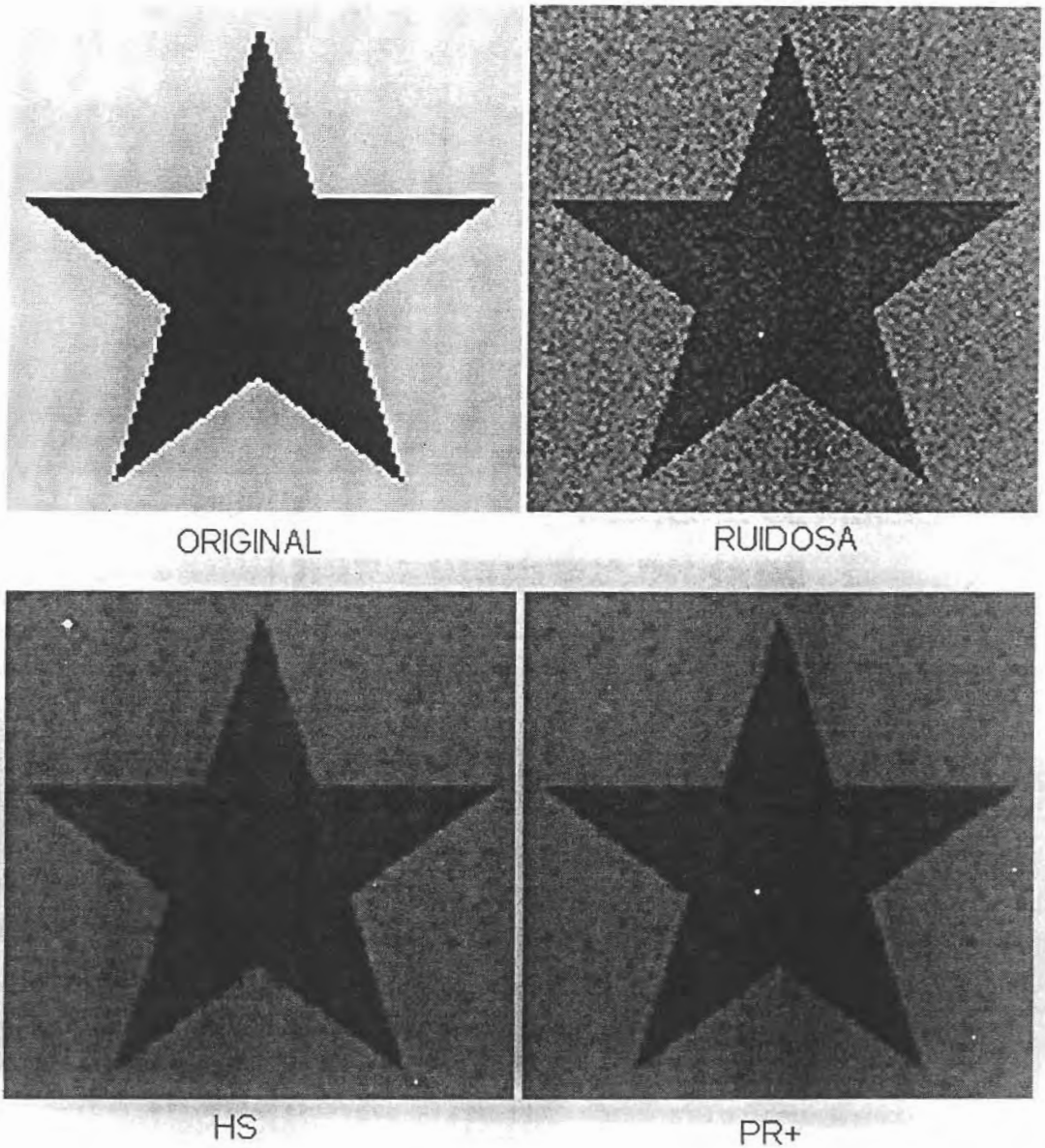
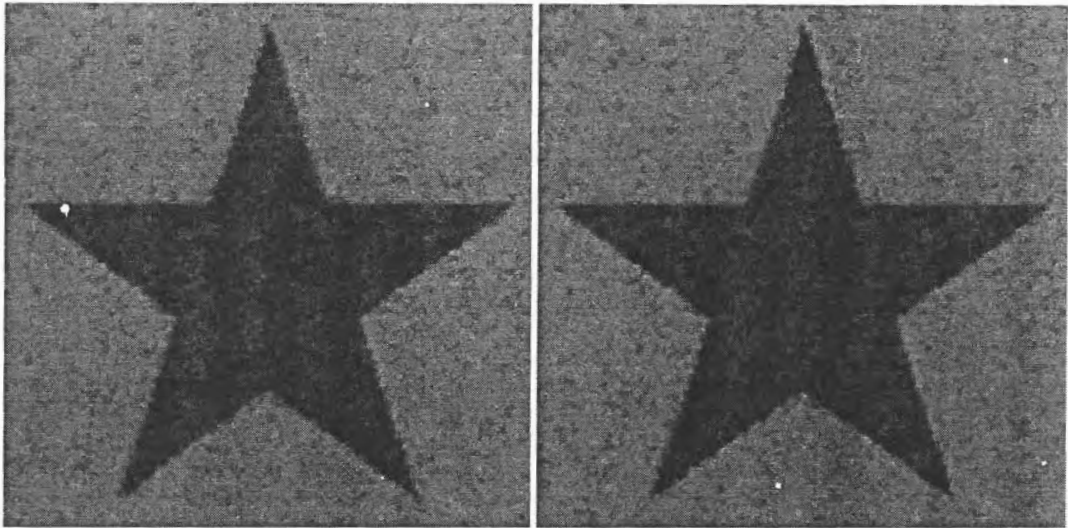


Figura 8-10: Imagen sintética de una estrella mostrando las restauraciones de los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.



HIBRIDO

DESCENSO DE GRADIENTE



FR

Figura 8-11: Resultados de la restauración de la imagen de la estrella por los métodos Híbrido, Descenso de Gradiente y FR. Como se puede ver en la Gráfica 8-4 y en la restauración del método de FR, la restauración es muy pobre. Los métodos Híbrido y de Descenso de Gradiente se comportaron aceptablemente.

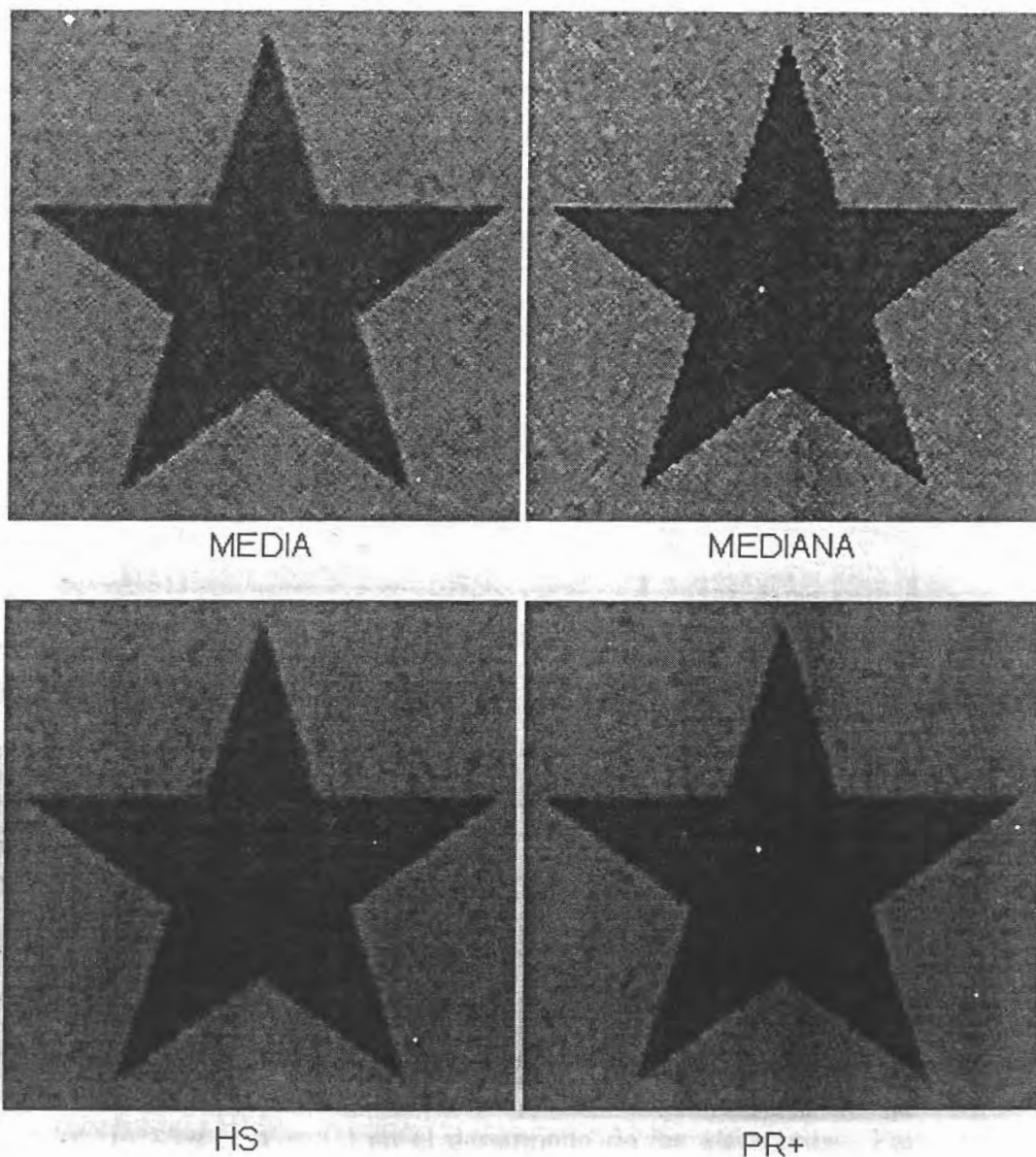


Figura 8-12: Imagen sintética de una estrella muestra las restauraciones obtenidas por los filtros de media y mediana comparando los resultados con los métodos de Polak-Ribiere positivo y Hestenes-Stiefel.

Capítulo 9

Conclusiones

En base a las pruebas presentadas para los algoritmos evaluados se puede concluir lo siguiente:

- La diferencia entre el desempeño de los diversos algoritmos evaluados no fue demasiado marcada. Los filtros de Media y Mediana fueron instrumentados de manera ilustrativa, pero se puede observar claramente para la gran mayoría de las imágenes un desempeño muy por debajo de los algoritmos aquí evaluados.
- De los algoritmos que consiguieron un mejor desempeño fueron Polak-Ribiere positivo y Hestenes-Stiefel, los resultados muestran que son de los algoritmos que para cualquier imagen alojan de los mejores resultados. Estos algoritmos muestran en general una mayor velocidad de convergencia.
- El valor de λ es primordial en el desempeño de los algoritmos. Para el método de descenso de gradiente, valores por arriba de 1 degeneran la imagen, lejos de restaurarla. En algunos otros algoritmos, este valor por arriba de 1 presenta un aplanamiento muy notorio en las imágenes.
- El algoritmo de Polak-Ribiere presentó también muy buenos resultados en general, pero se desempeñó mejor su sucesor, el algoritmo de Polak-Ribiere positivo.

- De acuerdo a las pruebas de rapidez de convergencia y estabilidad realizadas, el algoritmo de Hestenes-Stiefel cuenta con uno de los mejores tiempos y mejor minimización de la función $U(F)$, además de presentar resultados visuales bastante aceptables.
- En general, el algoritmo de Hestenes-Stiefel es el que funcionó mejor para todos los casos con respecto al resto de los algoritmos, aunque su desempeño no se puede catalogar como muy sobresaliente sobre los demás.

Trabajo Futuro:

El presente trabajo de tesis, aunque cumplió con las expectativas planteadas inicialmente, tiene por delante muchos puntos a explotar, tales como:

- Tratamiento de imágenes a color, que como ya se mencionó anteriormente, solamente consiste en realizar el mismo tratamiento a cada uno de los canales de la imagen.
- La instrumentación en algún sistema real es un trabajo muy promisorio también, el cual requiere de pruebas muy exhaustivas.
- Al programa realizado se le pueden añadir muchas más opciones de configuración y almacenamiento de los resultados obtenidos.

Bibliografía

- [1] S. Z. Li, Markov Random Field Models in Computer Vision, School of Electrical and Electronic Engineering Nanyang Technological University, (1994).
- [2] L.E. Scales, Introduction to Non-Linear Optimization, Department of Computer Science, University of Liverpool, (1984).
- [3] Mathematical Optimization (e-book), <http://csep1.phy.ornl.gov/mo/mo.html>, Copyright (C) 1991, 1992, 1993, 1994, 1995 by the Computational Science Education Project.
- [4] Enciclopedia Británica, <http://www.eb.com>.
- [5] Bernd Jähne, Digital Image Processing, Concepts, Algorithms and Scientific Applications, Springer-Verlag, 1991.
- [6] Jorge Nocedal, Theory of Algorithms for Unconstrained Optimization, Northwestern University, 1991.
- [7] G. H. Liu, L. L. Jing, L. X. Han y D. Han, A Class of Nomonotone Conjugate Gradient Methods for Unconstrained Optimization, Journal of Optimization Theory and Applications, Vol. 101, No.1, pp.127-140, 1999.
- [8] Jean Charles Gilbert and Jorge Nocedal, Global Convergence Properties of Conjugate Gradient Methods for Optimization, SIAM J. Optimization, Vol.2 No.1. pp.21-42; 1992.

- [9] David Heckerman, A Tutorial on Learning with Bayesian Networks, International Society for Bayesian Analysis, <http://www.bayesian.org/>.
- [10] M. Arioli, T.F.Chan, I.S.Duff, N.I.M.Gould and J.K.Reid, Computing a Search Direction for Large-Scale Linearly-Constrained Nonlinear Optimization Calculations, <http://www.numerical.rl.ac.uk/reports/reports.html>
- [11] Calculadora de imágenes Versión 0.1, CIMAT Departamento de Ciencias de la Computación, http://www.cimat.mx/proy_comp/caliman/caliman.html.
- [12] Powell, Non convex minimization calculations and conjugate gradient method, in Lecture Notes in Mathematics 1066, Springer-Verlag, Berlín, 1984, pp. 122-141.

Apéndice A

Manual de Usuario

En este apartado se realiza la descripción de las funciones implementadas en el programa de apoyo para este trabajo de tesis *RESGRAD*. Los puntos básicos para poder realizar la restauración de imágenes son muy sencillos, y se pueden resumir en:

- Abrir imagen ruidosa.
- Aplicar filtro.

A.1 Requerimientos del Programa

Para que el programa RESGRAD tenga un óptimo desempeño, el equipo en el que se instale deberá contar con al menos las siguientes características:

- Computadora Pentium MMX 200Mhz.
- 64Mb en memoria RAM.
- 10Mb de espacio disponible en disco duro.

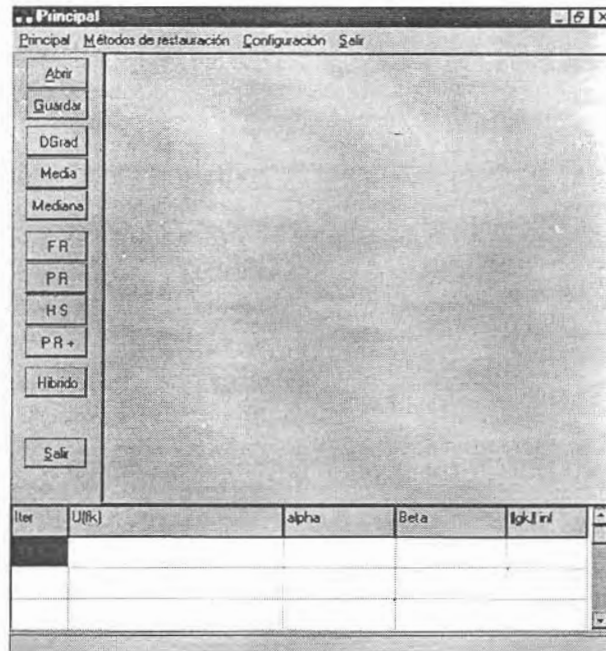


Figura A-1: Pantalla Principal

A.2 Pantalla Principal

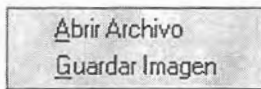
La pantalla principal (figura A-1) se puede ver que está compuesta de tres partes:

- En la parte superior la sección del menú principal con las opciones Principal, Métodos de restauración, Configuración y Salir.
- En la parte izquierda, una sección de botones rápidos, estos botones pueden ser empleados sin tener que acceder al menú principal.
- En la parte inferior, una sección de visualización de resultados.

A.3 Menú Principal

El menú principal tiene las opciones Abrir Archivo y Guardar Imagen.

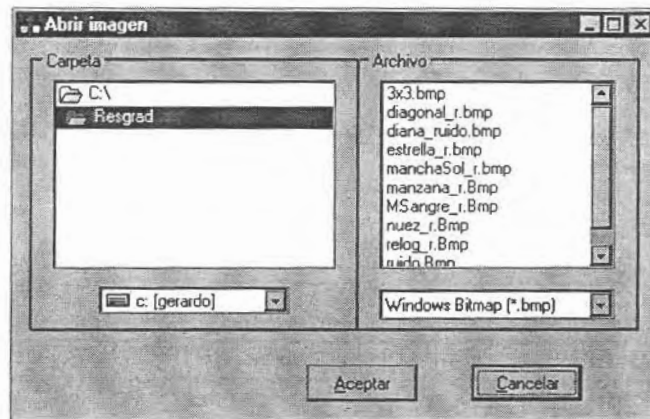
Para abrir una imagen se puede realizar mediante la opción del menú principal, con la que se muestra el siguiente menú:



Con la opción abrir, se despliega una pantalla con la cual se podrá seleccionar la imagen a abrir, esta opción se puede acceder también por medio del botón de acceso rápido **Abrir**

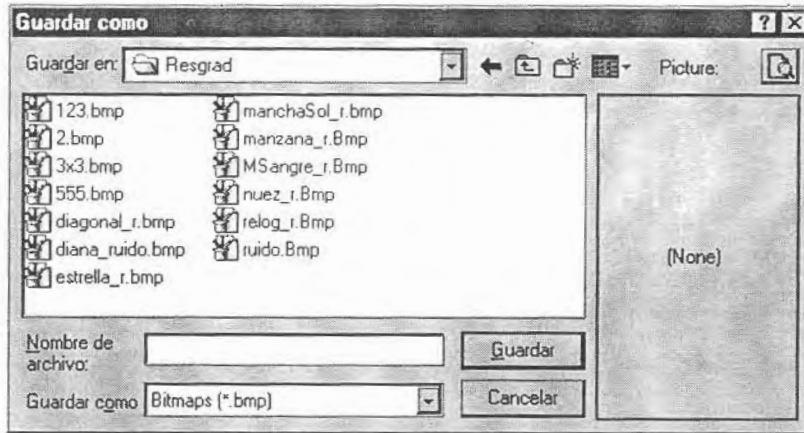


La pantalla para abrir una imagen es la siguiente:



En el lado izquierdo aparece el subdirectorio de trabajo y en la parte derecha las imágenes con formato bmp en dicho subdirectorio, para abrir una imagen sólo es necesario dar click sobre una de ellas y pulsar el botón aceptar.

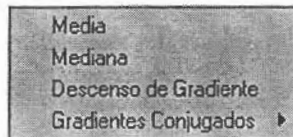
Para guardar una imagen también se puede realizar por medio del menú principal o también por su botón de acceso rápido **Guardar**. Después de pulsar cualquiera de las dos opciones, aparecerá la pantalla siguiente:



En la pantalla anterior sólo es necesario introducir el nombre del archivo con formato bmp a guardar y la ruta para dicho archivo, presionando finalmente el botón guardar.

A.4 Menú Métodos de Restauración

El menú Métodos de Restauración contiene los filtros que se pueden aplicar a una imagen para realizar su restauración. Al presionar esta opción, se despliegan las siguientes opciones:



Los filtros se encuentran agrupados por tipo, primeramente el filtro de Media, después el de Mediana, Descenso de Gradiente y finalmente los de Gradientes Conjugados, este último tiene a su vez el siguiente submenú:



Los filtros descritos anteriormente pueden ser accesados también por medio de los botones de acceso rápido localizados en la parte izquierda de la pantalla principal:

Media

Aplica el filtro de media a la imagen.

Mediana

Aplica el filtro de mediana a la imagen.

DGrad

Aplica el filtro de Descenso de Gradiente a la Imagen

Para Gradientes Conjugados se puede aplicar el filtro para cada algoritmo con su botón correspondiente:

FR	PR+
PR	Hibrido
HS	

FR para Fletcher-Reeves.

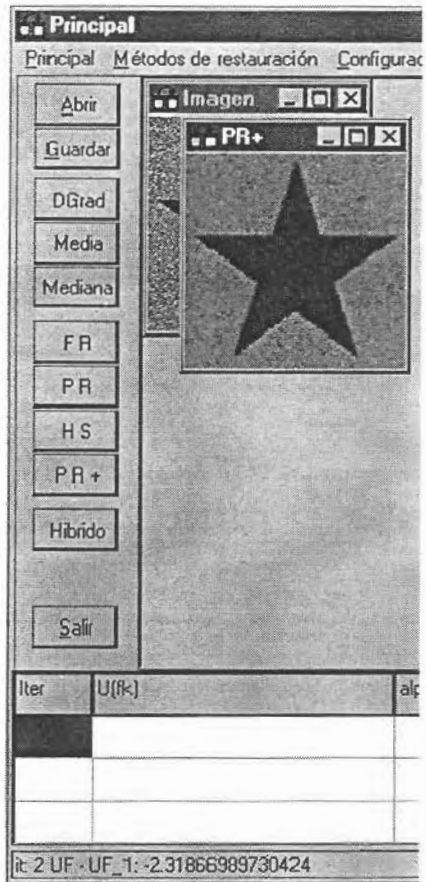
PR para Polak-Ribiere.

HS para Hestenes-Stiefel.

PR+ para Polak-Ribiere positivo.

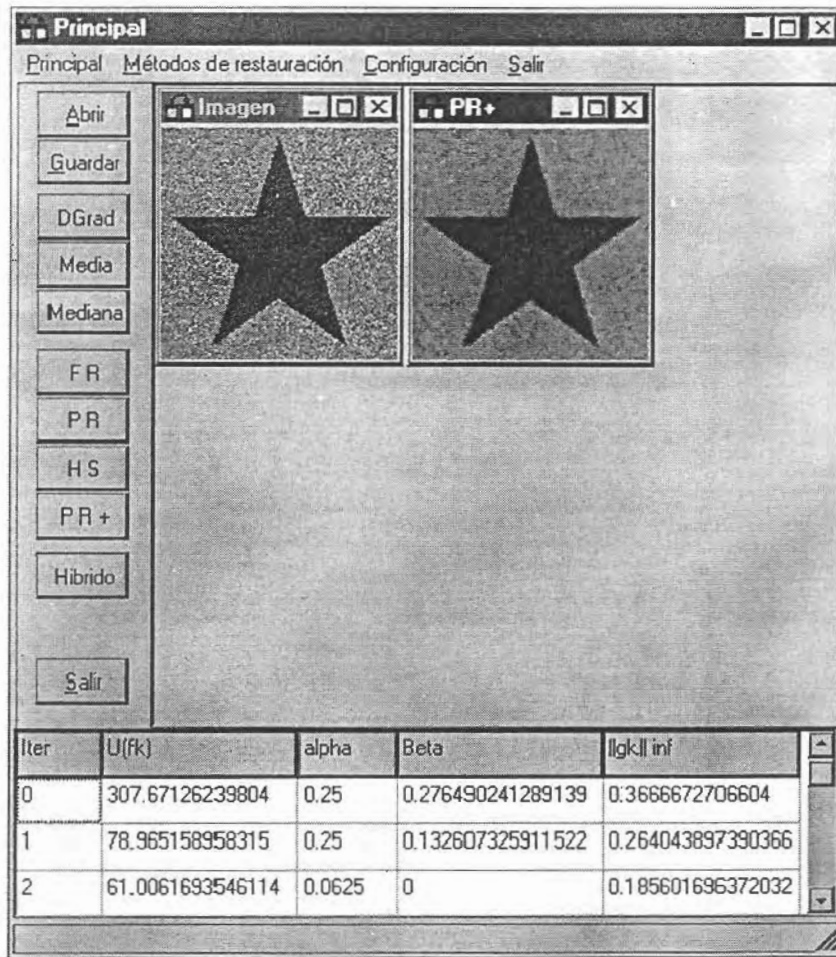
Hibrido para Híbrido de FR y PR.

El procedimiento que hay que seguir es, tener una imagen abierta, una vez seleccionada esta imagen, presionar ya sea la opción en el Menú Métodos de restauración o por medio de los botones de acceso rápido. Un ejemplo de cómo se vería una imagen en proceso de restauración es la siguiente:



Mientras se está restaurando una imagen, en la parte inferior se muestra el número de iteración y el valor de $U(F_k) - U(F_{k-1})$ mostrado como UF-UF_1.

Una vez restaurada la imagen, se puede mover para comparar los resultados, tal como lo muestra la siguiente figura.



A.5 Menú Configuración

Este menú únicamente contiene una opción *Visualizar Resultados*.

Visualizar Resultados

al dar click sobre esta opción, se puede ocultar o mostrar la sección de visualización de resultados:

Iter	U(fk)	alpha	Beta	gk inf

En esta sección aparecen los valores de la última restauración realizada:

Iter, Número de iteración.

U(fk), Valor para $U(F_k)$.

alpha, Valor calculado de α en cada iteración.

beta, Valor calculado de β en cada iteración.

||gk|| inf, Valor para $\|\nabla(F_k)\|_\infty$.

A.6 Salir

Con esta opción sale del programa, puede pulsarse también el botón de acceso rápido

