



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Sobre el modelo de difusión anisotrópica de Perona-Malik

Diana Angélica Pulido López

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Matemáticas
Bogotá D.C, Colombia
2018

Sobre el modelo de difusión anisotrópica de Perona-Malik

Diana Angélica Pulido López

Tesis presentada como requisito parcial para optar al título de:
Magíster en Ciencias Matemática Aplicada

Director:
Ph.D., John Jaime Rodríguez Vega

Línea de Investigación:
Procesamiento de imágenes

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de matemáticas
Bogotá D.C., Colombia
2018

A mi hermano Andrés, la persona que siempre
confía en mi.

Agradecimientos

Inicialmente, quisiera agradecer a mi director de tesis, el profesor John Jaime Rodríguez por su apoyo y orientación en el desarrollo de este trabajo, en especial por darme la oportunidad de conocer más a fondo el maravilloso mundo del procesamiento de imágenes. Adicionalmente, quiero expresar mi reconocimiento a todas las personas que leyeron este trabajo en sus versiones previas y que con sus aportes permitieron la creación de este escrito.

De manera muy especial deseo agradecer a mi pareja, Carlos Isaac Zainea por su paciente colaboración, sus grandes aportes y sus constantes motivaciones.

Resumen

La ecuación de Perona-Malik para procesamiento de imágenes abre un campo de investigación sobre las necesidades que debe satisfacer un modelo con el fin de mejorar los resultados de procesamiento. El objetivo principal de este documento es estudiar en detalle las motivaciones de estos autores para proponer su modelo, especialmente las características del coeficiente de difusión, para así, proponer otros coeficientes y evaluar sus resultados experimentales. Adicionalmente, estudiamos la teoría del análisis multiescala que demuestra la necesidad de usar ecuaciones diferenciales parciales en el procesamiento de imágenes y propone un modelo basado en la curvatura, el cual será discretizado e implementado en Matlab, para comparar la calidad de las imágenes obtenidas con los diferentes coeficientes estudiados. Se estudiará la combinación de ambos modelos y se exhibirán mejores resultados que los obtenidos por el modelo de Perona-Malik usando estas combinaciones.

Palabras clave: Procesamiento de imágenes, Filtros de preservación de bordes, Difusión Anisotrópica, Ecuación de Perona-Malik, Curvatura, coeficiente de difusión.

Abstract

The Perona-Malik equation for image processing begins a investigation area about needs that a model must satisfy in order to improve the processing results. Main objective of this document is to study author's motivations to consider their model, specially the diffusion coefficient characteristics, and with it, to propose other coefficients and evaluate their experimental results. Moreover, we study multiscale analysis theory which shows the need to use partial differential equations in image processing and proposes a model based on curvature. This model will be discretized and implemented in Matlab, to compare the quality of the images obtained with the different coefficients studied. The combination of both models will be studied and better results will be shown than those obtained by the Perona-Malik model using these combinations.

Keywords: Image processing, Edge-preserving Filters, Anisotropic diffusion, Perona-Malik equation, curvature, diffusion coefficient.

Contenido

Agradecimientos	VII
Resumen	IX
Contenido	XI
Lista de Figuras	XIII
Introducción	1
1. Preliminares	5
1.1. Procesamiento de Imágenes	5
1.1.1. Representación matricial	5
1.1.2. Filtrado de imágenes	6
1.1.2.1. Filtros espaciales	7
1.1.2.2. La derivada en filtros de procesamiento	8
1.1.2.3. Filtros basados en ecuaciones diferenciales parciales	14
1.1.2.4. Difusión anisotrópica	16
1.1.2.5. Operadores morfológicos	19
1.2. Ruido	20
2. Análisis multiescala y el modelo de Perona-Malik	23
2.1. Introducción	23
2.1.1. Axiomas del análisis multiescala	24

2.1.1.1. Axiomas arquitectónicos	25
2.1.1.2. Principio de comparación	27
2.1.1.3. Axiomas morfológicos	27
2.1.2. Forma general de un operador de análisis multiescala	29
2.1.3. Análisis axiomático de los modelos clásicos	39
2.1.3.1. El modelo de difusión anisotrópica de Perona-Malik	42
2.1.4. Modelo basado en la curvatura	43
3. Modificación del modelo de Perona-Malik	47
3.1. Modelo de Perona-Malik y su versión discreta	47
3.1.1. Implementación en Matlab	48
3.2. Coeficientes de difusión	54
3.3. Modelo basado en la curvatura	62
3.4. Métricas de comparación	66
3.4.1. Error cuadrático medio (MSE)	66
3.4.2. Relación pico señal / ruido (pSNR)	67
3.4.3. Índice de similitud estructural (SSIM)	67
3.4.4. Evaluación de calidad	68
3.5. Combinación de filtros	70
3.6. Análisis ante diferentes niveles de ruido	73
Introducción	5
Conclusiones	81
Bibliografía	83

Lista de Figuras

1.1. Dominio de una imagen digital	6
1.2. Vecindades.	7
1.3. Filtros puntuales.	7
1.4. Filtros espaciales	8
1.5. Representación de la primera y segunda derivada	10
1.6. Filtro Laplaciano.	11
1.7. Imagen agudizada con filtro Laplaciano.	11
1.8. Filtro Laplaciano escalado.	12
1.9. Imagen gradiente.	13
1.10. Imagen procesada con filtro isotrópico.	16
1.11. Coeficiente de Lorentz.	18
1.12. Adición de ruido.	21
3.1. Procesamiento con la función de Lorentz $c_1(x)$	51
3.2. Procesamiento con la función de Leclerc $c_2(x)$	52
3.3. Procesamiento de Perona-Malik con ruido Gaussiano	52
3.4. Procesamiento de perona-Malik con ruido Speckle	53
3.5. Coeficiente de Petrou c_3 y su flujo.	54
3.6. Procesamiento con coeficiente de Petrou $c_3(x)$	55
3.7. Coeficiente propuesto c_4 y su flujo.	56
3.8. Procesamiento con coeficiente $c_4(x)$	57
3.9. Coeficiente propuesto c_5 y su flujo.	58

3.10. Procesamiento con coeficiente $c_5(x)$	59
3.11. Coeficiente propuesto c_6 y su flujo.	59
3.12. Procesamiento con coeficiente $c_6(x)$	60
3.13. Coeficiente $c_7(x)$	61
3.14. Procesamiento con coeficiente $c_7(x)$	62
3.15. Procesamiento con el modelo de la curvatura con 10 iteraciones.	65
3.16. Mapas de calidad de la imagen.	70
3.17. Comparación filtro de mediana y su combinación con c_4	70
3.18. Comparación filtros c_4 , c_8 y gaussiano.	71
3.19. Imágen procesada con combinación c_3 y c_8	72
3.20. Imagen de prueba original: El camarógrafo	73
3.21. Procesamiento del camarógrafo.	74
3.22. Imagen de prueba original: Europa	75
3.23. Procesamiento de Europa.	77
3.24. Imagen de prueba original: Cielo	77
3.25. Procesamiento de Cielo.	78

Introducción

El procesamiento digital de imágenes trabaja en la adquisición, transmisión, procesamiento y representación de las imágenes. Esta disciplina es actualmente un amplio campo de investigación cuyo desarrollo está ligado a otras áreas del conocimiento como las matemáticas, la computación y el estudio de la percepción psicovisual del ser humano; Además ha mostrado importantes resultados en otras áreas como medicina, astronomía, geología, telecomunicaciones, control de procesos industriales y el entretenimiento.

Los primeros intentos de manipulación, almacenamiento y transmisión de imágenes se dieron en 1920, con el desarrollo de un sistema de transmisión de fotografías a través de un cable transatlántico usando códigos telegráficos. Este sistema permitía una codificación de la imagen en 5 niveles de gris, que fue mejorado en 1929 a 15 niveles y produjo un cambio importante en cuestiones económicas, puesto que redujo la transmisión de imágenes para periódicos de dos semanas a quince minutos.

En 1964 el procesamiento digital de imágenes tuvo un renacimiento gracias a la NASA. El programa *Ranger 7* que transmitía cerca de la luna logró enviar a la tierra algunas imágenes de la superficie lunar, que fueron convertidas de su forma analógica a digital, haciéndose visiblemente necesario el uso de computadoras para el procesamiento de tales imágenes y lograr la corrección de algunas distorsiones producidas por la cámara de televisión. En esta oportunidad, se logró eliminar la distorsión geométrica, la no linealidad fotométrica y el ruido oscilatorio [15].

El procesamiento de imágenes se divide en cuatro áreas de estudio: *Adquisición*: procesos de obtención de imágenes; *Realce o procesado*: técnicas de reducción de ruido y mejoramiento de detalles; *Segmentación*: División de una imagen en varias partes de interés y *Extracción de características*: Detección y localización de objetos geométricos en las imágenes.

Este trabajo se desarrolla dentro de las técnicas de realce, es decir las técnicas que se usan para mejorar la apariencia visual de las imágenes, recuperar o restaurar las imágenes degradadas. Los objetivos del realce de imágenes son bastante amplios, sin embargo los más comunes son la reducción del ruido de fondo en la adquisición, el ajuste de intensidad, el ajuste del contraste y el realce de bordes.

Una imagen puede ser definida como una función de dos variables $f(x, y)$ donde x y y son llamadas las coordenadas espaciales y f la intensidad o nivel de gris de la imagen en punto de coordenadas (x, y) . La imagen es llamada digital si el dominio de f tiene un número finito de elementos (x, y) , los cuales son llamados píxeles.

Las técnicas usadas para el procesamiento de imágenes son llamadas “filtros” y se dividen en dos ramas:

1. **Filtros puntuales:** técnicas donde el valor de la imagen realizada en un pixel depende unicamente del valor de su pixel en la imagen original. Dentro de los filtros puntuales se encuentra el negativo de una imagen, la variación de contraste, la extracción de campos de bits, el procesado del sistemático del histograma, la sustracción y el promedio de imágenes.
2. **Filtros espaciales:** Técnicas en las cuales el valor de un pixel en la imagen realizada depende del valor de su pixel pero también de los pixeles vecinos a éste en la imagen original. Un caso particular de estas técnicas son las que no trabajan directamente con el dominio espacial de la imagen original sino que lo hacen con su dominio de frecuencia, su dominio logarítmico u otro dominio que se considere adecuado. Dentro de esta categoría encontramos los filtros de paso bajo y de paso alto, el filtrado adaptativo, el filtro de mediana y los filtros de realce de bordes, entre los cuales nos interesa distinguir los llamados *filtros de difusión*.

En el procesamiento de imágenes se han usado de manera muy común los llamados filtros lineales (ya sean puntuales o espaciales), gracias a la economía computacional que ellos conllevan, sin embargo se ha arriesgado la calidad de los resultados obtenidos pues los procesos que se llevan a cabo no diferencian la zona de la imagen que se está procesando y por lo tanto regiones homogéneas y bordes de la imágenes son sometidos a la misma transformación.

Desde el año de 1983 [5], se han dado propuestas para solucionar este inconveniente mediante el uso de ecuaciones diferenciales parciales en el procesamiento de imágenes, abriendo paso a los llamados filtros de difusión.

Los filtros basados en la difusión son aquellos que para limpiar ruido, suavizar una imagen y realzar los bordes de una imagen consideran un flujo entre pixeles adyacentes, cuyo objetivo es que el proceso evolucione dependiendo del flujo para que las zonas similares se vuelvan más homogéneas y las zonas donde hay cambios (bordes) se realcen. Si la difusión se produce por igual en todas las direcciones, los filtros son llamados de *difusión isotrópica*. Si por el contrario, la difusión varía según la dirección, los filtros son llamados de *difusión anisotrópica*.

Perona-Malik [4] propusieron en 1990 un modelo de difusión que aunque puede clasificarse como una técnica isotrópica, ha sido considerada la base de la difusión anisotrópica pues establece que la transformación de una imagen debe depender de una función (coeficiente de difusión) que aumente la difusión cerca de los bordes y la detenga en las regiones interiores. Dicho argumento, que no es considerado por otros filtros de difusión isotrópica, abre un campo de investigación sobre las necesidades que se deben satisfacer en un modelo de procesamiento desde el punto de vista matemático, físico y de la teoría de la percepción para obtener resultados de más alta calidad. Ejemplos de esto pueden encontrarse fácilmente en [9], [11], [12].

El objetivo principal de este documento es estudiar en detalle las motivaciones de estos autores para proponer su modelo, especialmente las características y el efecto que tiene el

coeficiente de difusión. Perona-Malik propusieron en [4] dos coeficientes: la función de Lorentz y la función de Leclerc, pero en este trabajo analizaremos otros coeficientes existentes en la literatura [20] y propondremos algunos más, con el fin de comparar y evaluar los resultados experimentalmente.

Es también nuestra intención adentrarnos en algunas teorías que han surgido a raíz de este modelo y que muestran la necesidad de usar ecuaciones diferenciales parciales en el procesamiento de imágenes [10], [13], [14], [19], para determinar si es posible mejorar la calidad de una imagen que ha sido procesada por el método original de Perona-Malik.

Teniendo en cuenta lo anterior, la tesis estará dividida en tres partes. Un primer capítulo en el que se describirán algunos preliminares del procesamiento de imágenes, iniciando desde su representación matricial, algunos filtros de procesamiento básicos y el uso de la derivada como herramienta de transformación que dan paso a los filtros de difusión isotrópica y anisotrópica.

Luego, un capítulo en el que se enuncian los axiomas y propiedades del análisis multiescala, teoría que permite demostrar la necesidad de usar ecuaciones diferenciales parciales de segundo orden que dependen únicamente de la imagen original y sus derivadas. Analizaremos a la luz de esta teoría las condiciones que satisface el modelo de Perona-Malik y por último estudiaremos un modelo basado en la curvatura, que ha sido creado en [17] para satisfacer la totalidad de los axiomas propuestos en esta teoría.

Finalmente, un capítulo en donde teniendo en cuenta que el coeficiente de difusión $c(x)$, debe ser una función infinitamente diferenciable, monótona decreciente y cuyo flujo $xc(x)$ debe poseer un máximo en los reales positivos, consideraremos el coeficiente dado por Petrou en [20]:

$$c_3(x) = \begin{cases} 0,67 \left[1 - \left(\frac{x}{\sqrt{5}K} \right)^2 \right]^2, & \text{si } x \leq K\sqrt{5} \\ 0, & \text{en otros casos,} \end{cases}$$

y propondremos tres coeficientes más que satisfacen estas condiciones:

$$c_4(x) = \begin{cases} \frac{1}{K^4} \left(\frac{x}{\sqrt{K}} - K \right)^2 \left(\frac{x}{\sqrt{K}} + K^2 \right), & \text{si } x \leq K\sqrt{K} \\ 0, & \text{en otros casos.} \end{cases}$$

$$c_5(x) = \begin{cases} 0,5 \cos \left(\frac{\pi x}{K} \right) + 0,5, & \text{si } x \leq K \\ 0, & \text{en otros casos.} \end{cases}$$

$$c_6(x) = -\frac{1}{\pi} \arctan \left(\frac{x}{2K} - K \right) + 0,5$$

Por medio de una implementación existente del modelo en los repositorios de Mathworks, evaluaremos experimentalmente el comportamiento del modelo de Perona-Malik haciendo uso de los nuevos coeficientes de difusión ante diferentes tipos de ruido y los resultados de combinar en cada iteración el modelo con algunos filtros espaciales. También analizaremos los resultados de usar un coeficiente que no es derivable en todo su dominio.

$$c_7(x) = \begin{cases} -\frac{x}{K} + 1, & \text{si } x \leq K\sqrt{K} \\ 0, & \text{en otros casos.} \end{cases}$$

Por último, el modelo dado desde la teoría del análisis multiescala será discretizado siguiendo los pasos dados en [5] e implementado en Matlab, para comparar la calidad de las imágenes obtenidas con este modelo y los obtenidos con los diferentes coeficientes estudiados. Se estudiará la posibilidad de combinar ambos modelos en cada iteración y se determinará si es posible mejorar los resultados obtenidos con el modelo de Perona-Malik por medio de estas combinaciones.

Capítulo 1

Preliminares

En este capítulo presentamos algunas nociones matemáticas y de procesamiento de imágenes necesarias para la comprensión de la teoría de la difusión y específicamente del modelo de Perona-Malik (PM). Puede encontrarse información más detallada en [15].

1.1. Procesamiento de Imágenes

El término imagen monocromática o imagen simplemente, se refiere a una función de intensidad de luz bidimensional $f(x, y)$, donde x e y indican las coordenadas espaciales y el valor de f en cualquier punto (x, y) es llamado la luminosidad (o nivel de gris) de la imagen en dicho punto. Una imagen puede ser continua con respecto a los ejes coordenados x y y y también en su amplitud. Convertir una imagen en una imagen digital, requiere digitalizar tanto las coordenadas como la amplitud de la misma. La digitalización de las coordenadas es llamada muestreo y la digitalización de la amplitud el llamada cuantización. Cuando los valores de x , y y $f(x, y)$ son valores discretos y finitos, tenemos entonces una imagen digital. Los elementos de estos arreglos digitales son llamados elementos de imagen o píxeles.

1.1.1. Representación matricial

Podemos suponer el dominio de la función f como un subconjunto del plano, que tiene N filas y M columnas. Así el dominio de la imagen digital puede representarse en el plano de la siguiente forma

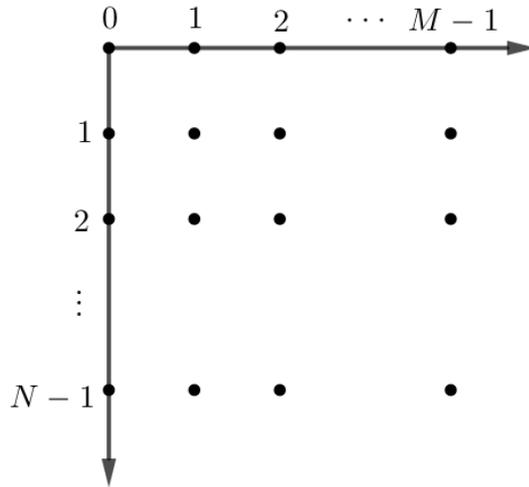


Figura 1.1. Dominio de una imagen digital

y la imagen puede entonces representarse por la matriz

$$f(x, y) = \begin{pmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \cdots & f(M-1, N-1) \end{pmatrix}. \quad (1.1)$$

El proceso de muestreo puede entenderse como una partición de un subconjunto del plano xy por medio de una cuadrícula, donde las coordenadas del centro de cada cuadro están en $\mathbb{Z} \times \mathbb{Z}$ y la cuantización como una función f que asigna a cada punto (x, y) un número que indica el nivel de gris, generalmente el número de niveles de grises es una potencia de 2.

1.1.2. Filtrado de imágenes

Dentro del grupo de transformaciones que se pueden aplicar a una imagen, nos interesan especialmente las transformaciones que se aplican sobre su dominio espacial, es decir que transforman el valor de cada pixel de la imagen dependiendo de los valores de los pixeles cercanos. Podemos expresarlo como

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

donde $f(x, y)$ es la imagen original y $g(x, y)$ es la imagen transformada.

Para definir los vecinos que intervienen en la transformación de un pixel (x, y) generalmente se usa un cuadrado o un rectángulo centrado en (x, y) . Este conjunto de pixeles vecinos es llamado *kernel* o *vecindad*.

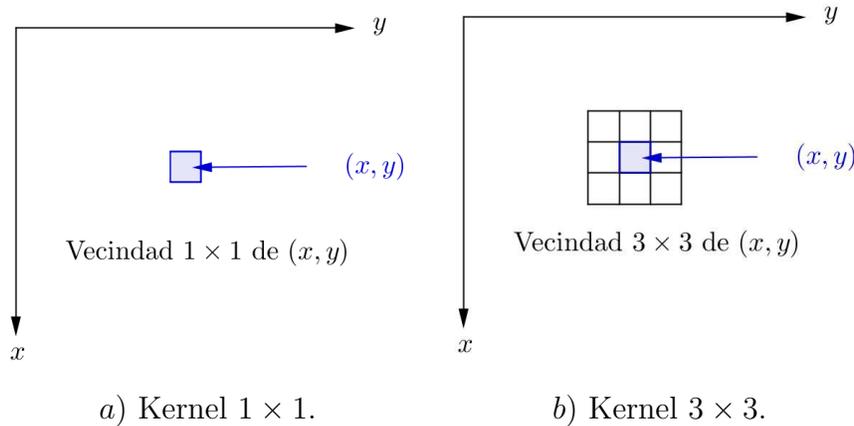


Figura 1.2. Vecindades.

Algunas transformaciones sobre el pixel (x, y) depende únicamente del valor $f(x, y)$, es decir que tienen como kernel cuadrados de tamaño 1×1 ; estas son llamadas transformaciones de intensidad y entre ellas podemos encontrar el negativo de la imagen, las funciones de contraste o la ecualización del histograma.

Ejemplificamos a continuación el uso de filtros puntuales haciendo uso de la imagen de Lena, tomada de [23]. En adelante mostramos los resultados de aplicación de diferentes filtros con esta imagen, que es una de las más usadas en documentos de procesamiento de imágenes.



a) Imagen original.



b) Negativo.



c) Función de contraste
 $\gamma = 0,3$

Figura 1.3. Filtros puntuales.

1.1.2.1. Filtros espaciales

Un filtro espacial consiste de una vecindad del pixel a transformar (centro) y una operación predefinida que se aplica a los pixeles de la vecindad, cuyo resultado es la imagen del del centro $g(x, y)$. Si esta operación se aplica a cada uno de los pixeles de la imagen, obtenemos una imagen *procesada o filtrada*. Cuando la operación desarrollada es lineal el filtro es conocido como filtro espacial lineal, en caso contrario es llamado un filtro no lineal.

En los filtros espaciales, el valor de la transformación del pixel (x, y) se obtiene mediante la suma de los productos de los coeficientes del filtro y los pixeles de la imagen abarcados por el filtro:

$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots \\ + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1)$$

Esto implica que podemos entender el filtro como una matriz del mismo tamaño de la vecindad que contiene los coeficientes que multiplicarán a cada vecino. Esta matriz es llamada *matriz de convolución* y sus coeficientes dependen de lo que se espera que el filtro haga, por ejemplo, en el *filtro promedio*, deseamos que el filtro transforme el valor de cada pixel por el promedio de los valores iniciales de su vecindad, entonces la matriz de convolución adecuada es

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

En algunas ocasiones, estos coeficientes están dados por una función $h(a, b)$ que depende de la posición de cada pixel vecino. Una función muy usada está dada por

$$h(a, b) = e^{-\frac{a^2+b^2}{2\sigma^2}}$$

Donde σ es la desviación estándar de la imagen, a y b indican la posición del vecino con respecto al centro del kernel. Este es conocido como filtro Gaussiano y otorga coeficientes cada vez más pequeños a los vecinos con mayores distancias del centro.



a) Imagen distorsionada.



b) Filtro promedio.



c) Filtro Gaussiano.

Figura 1.4. Filtros espaciales

El principal problema de estos filtros es la conservación de los bordes de la imagen, pues la operación usa todos los pixeles vecinos sin importar si ellos pertenecen al interior de una misma región o si por el contrario pertenecen a regiones distintas.

1.1.2.2. La derivada en filtros de procesamiento

Si llamamos *rampas* al conjunto de pixeles donde intensidad decrece con respecto a alguno de sus vecinos y *pasos* al conjunto de pixeles en los cuales la intensidad crece, podemos hacer uso de la derivada para estudiar el comportamiento de la imagen en estas regiones y la manera de caracterizar los bordes de cada una de ellas.

Representamos la primera derivada con la siguiente diferencia,

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x).$$

La derivada satisface las siguientes condiciones:

- $\frac{\partial f}{\partial x} = 0$ en áreas de intensidad constante.
- $\frac{\partial f}{\partial x} \neq 0$ al inicio de cada rampa o paso de intensidad.
- $\frac{\partial f}{\partial x} \neq 0$ a lo largo de las rampas o pasos pero no en su punto final.

De manera similar para la segunda derivada

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1) + f(x - 1) - 2f(x).$$

se tiene que:

- $\frac{\partial^2 f}{\partial x^2} = 0$ en áreas de intensidad constante.
- $\frac{\partial^2 f}{\partial x^2} \neq 0$ al inicio y al final de las rampas o pasos de intensidad, tomando en estos dos puntos valores de distinto signo.
- $\frac{\partial^2 f}{\partial x^2} = 0$ a lo largo de las rampas intensidad constante.

Los bordes o fronteras en las imágenes digitales se presentan en pixeles en los cuales la primera derivada toma valores suficientemente grandes, mientras que la segunda derivada produce en los bordes dos pixeles con valores distantes que están separados por ceros del resto de la imagen. Por esta razón se concluye que la segunda derivada proporciona mejor los detalles de una imagen y por esto es más usada en los filtros de transformación espacial.

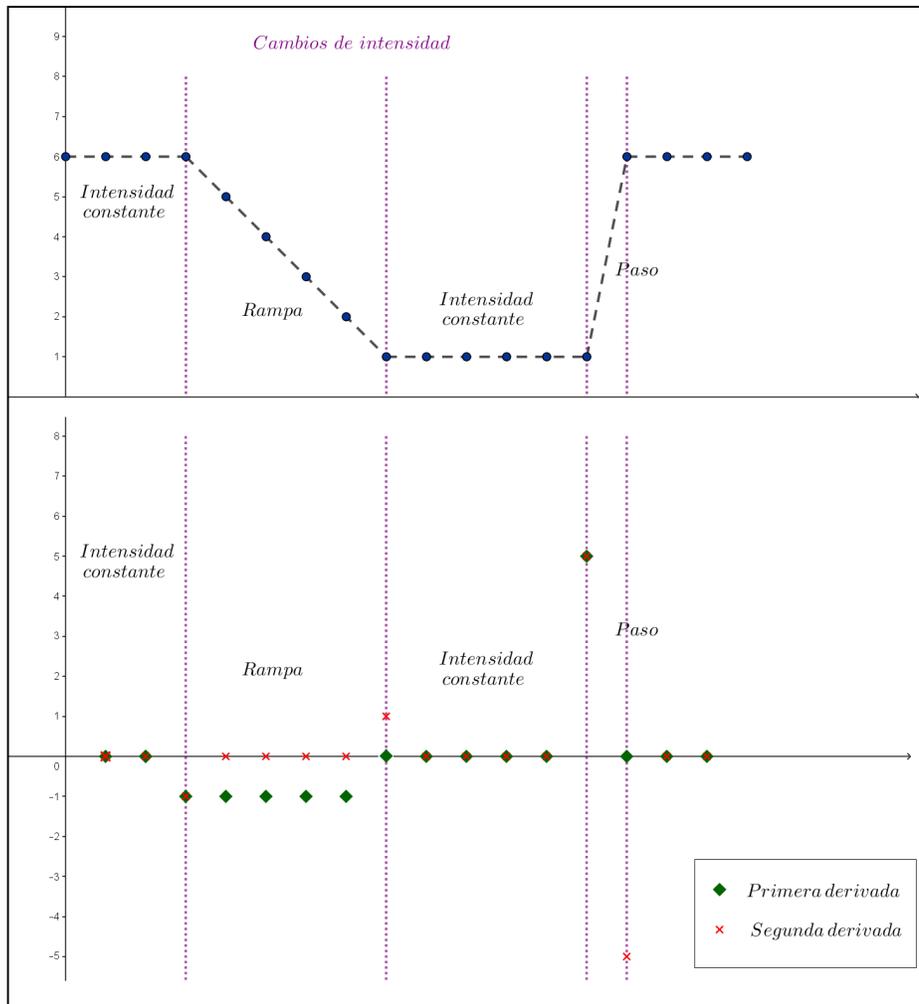


Figura 1.5. Representación de la primera y segunda derivada

Uno de los principales filtros que hace uso de la segunda derivada es el operador Laplaciano, el cual se define como

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}.$$

Este operador es considerado un filtro lineal, ya que las derivadas de cualquier orden son operadores lineales y además es el mayor representante de los filtros de difusión isotrópica en donde el filtro es aplicado sin importar la dirección, por lo cual son filtros invariantes bajo rotaciones.

Las derivadas parciales de una imagen $f(x, y)$ están definidas de manera discreta, por las expresiones

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y)$$

por lo cual el Laplaciano queda expresado en la siguiente forma

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Matricialmente lo representa la matriz

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Es posible hacer uso de las derivadas en dirección diagonal, por lo cual el operador se generaliza a la forma

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Es importante tener en cuenta que el filtro Laplaciano es usado para resaltar los bordes, por lo que al ser aplicado sobre una imagen se obtiene un resultado de niveles oscuros en la cual las líneas grises o blancas resaltan los bordes más prominentes de la imagen original.



a) Imagen original.



b) Laplaciano.



c) Laplaciano con direcciones diagonales.

Figura 1.6. Filtro Laplaciano.

Si nuestro objetivo es transformar la imagen original mediante este filtro, debemos realizar la operación

$$g(x, y) = f(x, y) - \nabla^2 f(x, y)$$

para obtener así una imagen con fronteras más marcadas o bordes más agudos.



a) Imagen original.



b) Imagen agudizada con filtro Laplaciano.



c) Imagen agudizada con filtro Laplaciano que incluye las direcciones diagonales.

Figura 1.7. Imagen agudizada con filtro Laplaciano.

Como en el Laplaciano pueden presentarse valores negativos, estos son enviados a cero, por lo cual muchas zonas de esta imagen son totalmente negras. Es por esto que en algunas ocasiones se usa un coeficiente para escalar los valores del Laplaciano y que tengan un rango mayor o igual que cero.



a) Imagen original.



b) Filtro Laplaciano escalado.



c) Imagen agudizada con filtro Laplaciano escalado.

Figura 1.8. Filtro Laplaciano escalado.

La primera derivada también es usada para en el desarrollo de filtros de procesamiento de imágenes. Para una imagen $f(x, y)$, se define el gradiente en el punto (x, y) como

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

y la magnitud del gradiente como

$$\|\nabla f\| = \sqrt{g_x^2 + g_y^2}$$

Recordemos que el gradiente y su magnitud tiene una importante propiedad geométrica: el vector ∇f indica la dirección con mayor tasa de cambio de f sobre el punto (x, y) , mientras que su magnitud, $\|\nabla f\|$, es el valor de esa máxima tasa de cambio.

Es posible considerar la imagen $M(x, y)$ como una imagen del mismo tamaño de $f(x, y)$ en donde se alojan todos los valores de $\|\nabla f\|$ en cada uno de los píxeles (x, y) . Esta imagen es conocida como *imagen gradiente*.

Aunque las derivadas son lineales, estos filtros son no lineales por el cálculo de la magnitud. En ocasiones la magnitud es aproximada como $\|\nabla f\| = |g_x| + |g_y|$ para reducir el costo computacional.

Para el caso discreto, podemos representar una vecindad 3×3 del pixel z_5 por la matriz

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}.$$

En este caso podemos decir que $g_x = z_8 - z_5$ y $g_y = z_6 - z_5$ si usamos la definición de derivada dada anteriormente, sin embargo en la literatura han sido dadas otras definiciones. Por ejemplo Roberts en 1975 propuso el uso de derivadas cruzadas de la forma $g_x = z_9 - z_5$ y $g_y = z_8 - z_6$. En términos matriciales podemos representarlas como

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Dado que los operadores de tamaño par tienen algunos problemas de implementación por no tener un centro simétrico, se proponen algunas versiones de la derivada para vecindades de tamaño 3×3 con centro en z_5

$$g_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

Las cuales son representadas por las matrices de convolución

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & -1 \end{bmatrix}$$

Estos filtros son conocidos como el operador de Sobel y el objetivo de los coeficientes dobles es resaltar el valor central de la vecindad. Es importante mencionar que tanto en estas definiciones de la derivadas parciales como en las anteriores, la suma de los coeficientes del filtro es cero, garantizando que la derivadas sea nula en las regiones constantes.



a) Imagen gradiente con derivadas usuales.



b) Imagen gradiente con filtro de Roberts.



c) Imagen gradiente con filtro de sobel.

Figura 1.9. Imagen gradiente.

1.1.2.3. Filtros basados en ecuaciones diferenciales parciales

La mayor dificultad en el procesamiento de imágenes hasta la década de los 70, era la aplicación de transformaciones a todas las zonas de una imagen sin importar si la zona era o no homogénea. El problema radicaba en el deterioro de los bordes de la imagen. Para erradicarlo, se realizaron algunas propuestas que haciendo uso de ecuaciones diferenciales parciales, permitían reconocer las características del área a procesar y dependiendo de ellas la acción que el filtro debía realizar en ciertas zonas.

El primer modelo presentado en esta dirección, fue formalizado por Witkin [1] en 1983. En este modelo, la imagen original $I(x, y, 0)$ se convoluciona con un filtro Gaussiano de varianza t , para obtener la imagen resultante:

$$I(x, y, t) = I(x, y, 0) * G(x, y; \sigma_t) \quad (1.2)$$

Para diferentes valores de t , se obtiene un conjunto de imágenes a distintas resoluciones o con diferentes niveles de suavizado, lo cual define un *espacio de escala* para la imagen.

En 1984, el trabajo de Witkin fue retomado y mejorado por Koenderink [2], quien observó que las imágenes obtenidas de un sólo parámetro podían ser vistas como la solución de la ecuación de difusión del calor:

$$\frac{\partial I(x, y, t)}{\partial t} = \Delta I(x, y, t) = \frac{\partial^2 I(x, y, t)}{\partial x^2} + \frac{\partial^2 I(x, y, t)}{\partial y^2} \quad (1.3)$$

con la condición inicial $I(x, y, 0)$ la imagen original.

Para entender la motivación de este modelo, se debe tener en cuenta que los procesos de difusión están estrechamente relacionados con las leyes de conservación: En ciertos sistemas donde la concentración o la temperatura, I , varía dependiendo de la región, se producirá un flujo de las regiones con mayor concentración hacia las regiones con concentración más baja hasta que el sistema encuentre un estado de equilibrio.

Suponiendo que la ley de conservación de I se satisfaga en un dominio Ω se produce una variación de la cantidad de la sustancia I , por la presencia de un flujo \mathbf{j} a través del dominio, lo cual puede representarse mediante la ecuación

$$\underbrace{\frac{\partial}{\partial t} \int_{\Omega} I dV}_{\text{Variación de } I \text{ en } \Omega} + \underbrace{\int_{\partial\Omega} \mathbf{j} \cdot \mathbf{n} dS}_{\text{flujo a través de } \Omega} = 0 \quad (1.4)$$

donde \mathbf{n} representa el vector normal a $\partial\Omega$.

Si el dominio es fijo con respecto al tiempo, usamos el teorema de la divergencia para obtener

$$\int_{\Omega} I_t dV + \int_{\Omega} \nabla \cdot \mathbf{j} dV = \int_{\Omega} I_t + \nabla \cdot \mathbf{j} dV = 0 \quad (1.5)$$

para regiones arbitrariamente pequeñas, esto implica que

$$I_t + \nabla \cdot \mathbf{j} = 0 \quad (1.6)$$

Suponiendo que el proceso de difusión satisface la primera ley de Fick (el flujo es proporcional y en el sentido opuesto al gradiente de la concentración), la cuál puede ser enunciada como

$$\mathbf{j} = -c \nabla I \quad (1.7)$$

siendo c el coeficiente de difusión o coeficiente de conductividad. Podemos reescribir la ecuación (1.6)

$$I_t = \nabla \cdot (c \nabla I) \quad (1.8)$$

Si $c = 1$, tenemos entonces que

$$I_t = \nabla \cdot \nabla I = \Delta I \quad (1.9)$$

Ecuación conocida como la ecuación diferencial del calor. Ahora bien, en la ecuación (1.8), cuanto más grande sea el valor de c el flujo de la sustancia es mayor y por tanto el proceso de difusión es más rápido.

Este proceso se encuentra estrechamente relacionado con el procesamiento de imágenes si consideramos los niveles de gris como la concentración o la temperatura del sistema. En este caso, los niveles de gris pasan por un proceso de difusión dictado por la ecuación del calor que avanza con el tiempo, así ya no tenemos una imagen $I(x, y)$ sino una familia de imágenes que cambia con el tiempo $I(x, y, t)$ donde $I(x, y, 0) = I(x, y)$.

Como las imágenes tienen un dominio acotado Ω , suponemos que el flujo es nulo sobre el contorno $\partial\Omega$, lo que impone al flujo condiciones de contorno homogéneas de tipo Neumann

$$\frac{\partial I}{\partial n} = \nabla I \cdot \mathbf{n} = 0$$

que brindan una estabilización local de la intensidad de la imagen en la dirección perpendicular a $\partial\Omega$. Esto aísla el dominio de la imagen de \mathbb{R}^2 pues el flujo puede variar a través de la frontera pero permanece constante al atravesar $\partial\Omega$ de manera perpendicular.

Lo cual nos lleva a plantear el problema de la difusión en la forma

$$\begin{cases} I_t = \Delta I \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.10)$$

La solución de esta ecuación se obtiene mediante la convolución de I_0 con un filtro Gaussiano cuya desviación estándar depende de t , (Ver [6], [14]). A medida que t aumenta la difusión es mayor, por lo cual el filtro tiene una desviación estándar mayor.

Sin embargo, el operador Laplaciano es isótropo, lo cual implica que el valor de $\Delta I = I_{xx} + I_{yy}$ es el mismo que el valor obtenido tras una rotación del plano en la que los nuevos ejes está alineados con los vectores de una base ortonormal (ξ, η) , así

$$\Delta I = I_{\xi\xi} + I_{\eta\eta}$$

lo cual hace que la difusión mediante la ecuación del calor no tenga direcciones preferentes, sino que sea igualmente aplicada en todas las direcciones del plano. Este flujo es llamado *difusión isotrópica* pero debido a la propiedad en mención, es un filtro en el que se obtendrá una imagen totalmente destruida.



a) Imagen original.



b) Filtro isotrópico.

Figura 1.10. Imágen procesada con filtro isotrópico.

1.1.2.4. Difusión anisotrópica

En la propuesta de Koenderink [2], se menciona que es necesario encontrar un planteamiento matemático que esté estrechamente relacionado con la teoría de la percepción visual, en donde existe una escala de resolución, en la cual una imagen inicial es procesada sin añadir nuevos detalles en cada paso y propone la ecuación del calor como opción responder la estas necesidades. Hummel [3], en 1986 sugiere que la ecuación del calor no es la única ecuación diferencial parcial que satisface las condiciones expuestas por Koenderink y que además lo que se requiere para obtener un buen resultado en el realce, es que el flujo con el que se implemente el filtro satisfaga el principio del máximo.

Desde allí, muchas propuestas se ha presentado, algunas sustituyendo el flujo del calor por una PDE no lineal que no difunda el flujo de manera uniforme aunque no se satisfaga el principio del máximo. Estos filtros son llamados de difusión anisotrópica.

Los principales exponentes de la difusión anisotrópica son Pietro Perona y Jitendra Malik, quienes en 1990 presentaron un modelo que presenta un suavizado multiescala preservando bordes basados en tres criterios:

1. **Causalidad:** Disminuir la resolución de una imagen no pueden generar nuevos detalles en la imagen.

2. **Localización inmediata:** En cada resolución los límites de las regiones deben ser distinguidos y deben coincidir con los originales.
3. **Suavizado según la zona:** El suavizado debe ser mayor dentro de una región que en sus fronteras.

El modelo está basado en la segunda ley de Fick (1.8), con un coeficiente de conductividad variable $c(x, y, t)$ que controle la tasa de difusión en cada punto de Ω , así, el problema puede ser reescrito a la forma

$$\begin{cases} I_t = \operatorname{div}(c(x, y, t)\nabla I) \\ I(x, y, 0) = I_0(x, y) \end{cases} \quad (1.11)$$

Donde $\operatorname{div}(c(x, y, t)) = c\Delta I + \nabla c \cdot \nabla I$, puesto que

$$\begin{aligned} \operatorname{div}(c \cdot \nabla I) &= \operatorname{div}(c \cdot (I_x, I_y)) \\ &= \operatorname{div}(cI_x, cI_y) \\ &= cI_{xx} + c_x I_x + cI_{yy} + c_y I_y \\ &= c(I_{xx} + I_{yy}) + (c_x, c_y) \cdot (I_x, I_y) \\ &= c\Delta I + \nabla c \cdot \nabla I \end{aligned}$$

El coeficiente de difusión c , dependerá de la norma del gradiente en cada punto. Para los gradientes con normas pequeñas, los cuales se presentan en las zonas homogéneas, se espera que el coeficiente de difusión sea alto para realizar un proceso de filtrado muy fuerte, logrando así un efecto de suavizado. Mientras que en los bordes de la imagen, donde se encuentra una norma del gradiente grande, se espera asignar un coeficiente de difusión pequeño, de tal manera que el proceso de difusión sea muy lento, casi nulo, y con ello lograr que se protejan los bordes.

Teniendo en cuenta estas características el modelo se puede formular de la siguiente forma:

$$\begin{cases} I_t = \nabla(c(|\nabla I|^2)\nabla I) & \text{en } \Omega \times (0, \infty) \\ \frac{\partial I}{\partial n} = 0 & \text{en } \partial\Omega \times (0, \infty) \\ I(x, 0) = I_0(x) & \text{en } \Omega \end{cases} \quad (1.12)$$

Donde Ω denota el dominio de la imagen. En el caso del procesamiento de imágenes Ω es un subconjunto acotado de \mathbb{R}^n con frontera de clase C^1 .

Al buscar soluciones para este problema, se asumen las siguientes características:

1. Las derivadas de I_0 se anulan en $\partial\Omega$.

2. Se restringe el coeficiente de difusión c a una función monótona decreciente entre 1 a 0 cuando $|\nabla I|^2$ varía de 0 a ∞ .
3. c es una función infinitamente derivable.
4. Tomando $s = |\nabla I|$, la función $\Phi(s) = sc(s^2)$ tiene un máximo en \mathbb{R}^+ . Esta función Φ es llamada la función de flujo.

La primera sugerencia de Perona-Malik para la elección de estas funciones fue:

La función de Lorentz:

$$c(s^2) = \frac{1}{1 + \frac{s^2}{K^2}} \quad (1.13)$$

y la función de Leclerc

$$c(s^2) = e^{-\frac{s^2}{2K^2}} \quad (1.14)$$

Para una constante positiva K , y con la función de flujo definida como

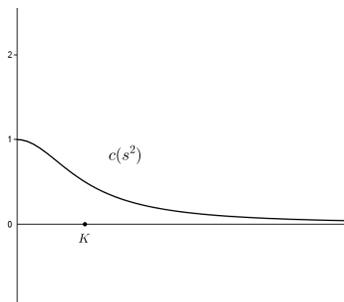
$$\Phi(s) = s \cdot c(s^2) \quad (1.15)$$

En el caso de la función de Lorentz (1.13) se tiene:

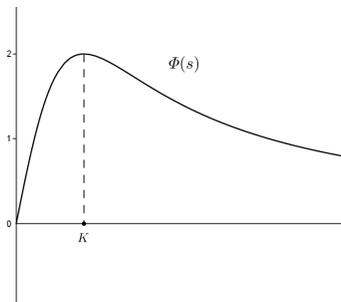
$$\Phi(s) = s \cdot \frac{1}{1 + \frac{s^2}{K^2}} = \frac{sK^2}{K^2 + s^2} \quad (1.16)$$

$$\Phi'(s) = \frac{K^2(K^2 - s^2)}{(K^2 + s^2)^2}$$

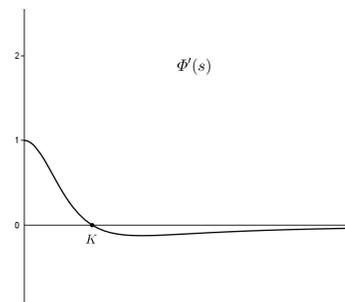
En donde $\Phi'(s) < 0$ para $|s| > K$, $\Phi'(s) > 0$ para $|s| < K$ y Φ posee un máximo cuando $s = K$.



a) Función de Lorentz.



b) Flujo.



c) Derivada del flujo.

Figura 1.11. Coeficiente de Lorentz.

En el caso de la función de Leclerc (1.14), tenemos que:

$$\begin{aligned}\Phi(s) &= s \cdot e^{-\frac{s^2}{2K^2}} \\ \Phi'(s) &= \frac{K^2 - s^2}{K^2} e^{-\frac{s^2}{2K^2}}\end{aligned}\tag{1.17}$$

Lo que nos lleva nuevamente a $\Phi'(s) < 0$ para $|s| > K$, $\Phi'(s) > 0$ para $|s| < K$ y que Φ posee un máximo cuando $s = K$.

Hasta el momento el modelo de Perona-Malik puede verse como un modelo de difusión isotrópica, sin embargo en el caso de dos dimensiones, el problema puede ser reescrito de la siguiente manera [7]:

$$I_t = \Phi'(|\nabla I|)I_{\eta\eta} + c(|\nabla I|^2)I_{\xi\xi}\tag{1.18}$$

donde las coordenadas η y ξ denotan direcciones paralelas y perpendiculares a ∇I respectivamente. Esto indica que la difusión varía directamente en la frontera de I y de manera inversa en la dirección del gradiente, lo cual incluye el modelo de Perona-Malik como un filtro de difusión anisotrópica.

El aporte de Perona-Malik en el procesamiento de imágenes ha orientado esta disciplina hacia la comprensión de las propiedades matemáticas del modelo de difusión anisotrópica relacionadas con su buen planteamiento y estabilidad, además de su extensión y modificación para obtener implementaciones computacionales más rápidas o para ciertas aplicaciones específicas [5].

1.1.2.5. Operadores morfológicos

A continuación enunciaremos algunas propiedades que son bien conocidas en la morfología matemática. Ellas establecen que el análisis de imágenes debe ser invariante bajo fluctuaciones de luz y bajo cambios de posición, orientación y escala de las curvas planas.

Los operadores principales en esta línea de procesamiento son la erosión, la dilatación, la apertura y el cierre. Su objetivo principal es caracterizar algunas propiedades de la imagen como tamaño, forma, convexidad, conectividad y distancia, para poder estudiar los cambios que se producen en ella a partir de traslaciones, rotaciones o reflexiones.

1.2. Ruido

En los procesos de adquisición, transformación o compresión de imágenes existe la posibilidad de que la imagen original se distorsione, estas perturbaciones en general reciben el nombre de *ruido*. De manera intencional, es posible añadir ruido a una imagen dada por medio con el objetivo de medir la efectividad de un filtro de procesamiento. El ruido añadido a una imagen se puede clasificar en tres tipos: aditivo, impulsivo y multiplicativo de acuerdo a la técnica usada para tal inclusión.

El ruido aditivo consiste en adicionar la la imagen original I una imagen de perturbación μ que modifica el valor de algunos pixeles.

$$g(x, y) = I(x, y) + \mu(x, y)$$

Generalmente, la función adicionada μ es producida por la curva Gaussiana y $g(x, y)$ representa la imagen con ruido añadido.

El ruido impulsivo implica cambiar en la imagen el valor algunos pixeles por valores muy distintos a los pixeles vecinos. Se puede modelar mediante la ecuación

$$g(x, y) = (1 - p) I(x, y) + p \mu(x, y)$$

donde $\mu(x, y)$ representa el ruido impulsivo y p toma valores en el conjunto $\{0, 1\}$. El ruido de este tipo más conocido es llamado sal y pimienta.

Por último, el ruido multiplicativo realiza cambios dependiendo de los valores iniciales de I , en este caso se modela mediante

$$g(x, y) = I(x, y) + I(x, y) \mu(x, y)$$

Visualmente, este ruido aporta a la imagen un aspecto granulado y el más común es llamado speckle.

Para la implementación en Matlab de estos tipos de ruido, el programa cuenta con la herramienta “imnoise”. Específicamente las funciones para los tres tipos de ruido que usaremos en adelante son: `imnoise(I, 'gaussian', m, v)` para adicionar ruido Gaussiano de media m y varianza v , `imnoise(I, 'salt&pepper', d)` para adicionar ruido sal y pimienta con un porcentaje d de pixeles afectados y `imnoise(I, 'speckle', v)` para ruido multiplicativo con varianza v .



a) Imagen original.



b) Ruido sal y pimienta.



c) Imagen con ruido Gaussiano



d) Imagen con ruido speckle.

Figura 1.12. Adición de ruido

Capítulo 2

Análisis multiescala y el modelo de Perona-Malik

Uno de los mayores retos del procesamiento de imágenes radica en la comprensión de las propiedades de una imagen y cómo los filtros aplicados a ella deben respetar y resaltar tales propiedades, en otras palabras, los procedimientos para pasar de una imagen original a una versión suavizada de ella que debe contener la información más significativa de la imagen original.

Para determinar las condiciones que debe satisfacer un filtro de procesamiento de imágenes estudiaremos en este capítulo la propuesta de Álvarez, Guichard, Lions y Morel en [17] y [18], en donde una serie de axiomas o propiedades, que bajo algunos criterios físicos y de la teoría de la percepción, un operador debe satisfacer para lograr los objetivos de suavizado y conservación de bordes deseados en el tratamiento de imágenes.

Su teoría, llamada análisis multiescala busca la axiomatización de los filtros de procesamiento y la definición explícita del tipo de ecuaciones se deben usar se para satisfacer estos axiomas. Nuestro objetivo, es justificar a la luz de esta teoría el uso de ecuaciones diferenciales parciales de segundo orden, analizar las propiedades del modelo propuesto por Perona-Malik y estudiar una mejora al modelo propuesta por estos autores.

2.1. Introducción

El análisis multiescala es llamado comúnmente teoría del procesamiento de imágenes y tiene como objetivo estudiar cómo los niveles de gris fluyen dentro de la imagen sin dar tanta información local sobre la imagen. En esta teoría, la imagen original es transformada en nuevas imágenes cuyo parámetro principal es la escala t , que mide el grado de suavizado que se aplica a la imagen.

Se define entonces, el análisis multiescala de una imagen como una familia de transformaciones $(T_t)_{t \geq 0}$ que al ser aplicada a la imagen original $f(x)$ produce una sucesión de imágenes

$u(x, t) = (T_t f)(x)$. Rigurosamente hablando $T_t f$ es una versión semi-local de f donde una vecindad de tamaño t alrededor de x ha sido escaneada para determinar el valor de $(T_t f)(x)$.

El objetivo principal de esta teoría es considerar una serie de axiomas que debe satisfacer el análisis multiescala T_t y deducir de ellos una fórmula explícita para realizar este tratamiento. Específicamente se estudiará bajo qué hipótesis el análisis multiescala T_t produce una sucesión de imágenes $u(x, t) = (T_t)(f)$ que son soluciones de una ecuación diferencial de segundo orden

$$\frac{\partial u}{\partial t} = F(D^2 u, Du, t) \text{ con } u(x, 0) = f(x).$$

La axiomatización de esta teoría formulada en [17] permite el análisis de los modelos propuestos por Witkin [1] y Koenderink [2] basados en la ecuación del calor $\frac{\partial u}{\partial t} = \Delta u$,

el modelo de Perona-Malik [4] de difusión anisotrópica $\frac{\partial u}{\partial t} = \text{div}(c(|\nabla u|)\nabla u)$ mencionados anteriormente. Pero también permite el análisis de otros que no trataremos en este texto como

la noción de curvatura media $\frac{\partial u}{\partial t} = \text{div} \left(\frac{Du}{|Du|} \right) |Du|$ y las transformaciones en morfología

matemática (dilatación y erosión), dadas por $\frac{\partial u}{\partial t} = \pm |Du|$. Además, propone un modelo de análisis morfológico multiescala cuya ecuación asociada es

$$\frac{\partial u}{\partial t} = \beta(t \text{ curv}(u)) |\nabla u|,$$

donde β es una función real no decreciente, $\text{curv}(u)$ es la curvatura de la curva de nivel de u que pasa a través de x .

Esta teoría demuestra que existe un único análisis multiescala que satisface todas las propiedades de invarianza clásicas y invarianza bajo proyecciones. Además posee además propiedades que no tenían los modelos anteriores: permite analizar formas planas en una forma independiente de su orientación o su localización en espacio tridimensional. El modelo está dado por la ecuación

$$\frac{\partial u}{\partial t} = |\nabla u| (t \cdot \text{curv}(u))^{1/3}.$$

2.1.1. Axiomas del análisis multiescala

Consideramos un análisis multiescala a una familia T_t parametrizada por $t \geq 0$ de operadores (posiblemente no lineales) sobre funciones definidas sobre \mathbb{R}^n . En la práctica $n = 2$ o $n = 3$. Se asume que los elementos de esta familia están bien definidos sobre C_b^∞ , es decir,

el espacio de funciones acotadas con derivadas acotadas de cualquier orden, y que $(T_t f)(x)$ es una función continua acotada sobre $\mathbb{R}^n \times (0, \infty)$ cuando $f \in C_b^\infty$.

En la práctica, una imagen no está necesariamente definida sobre un espacio euclidiano completo, pero sin esta hipótesis la mayoría de las propiedades de invarianza serán difíciles de tratar de manera rigurosa. Además no hay pérdida de generalidad en este hecho pues una imagen rectangular puede ser extendida a una imagen cuyo dominio sea el espacio completo (por ejemplo por reflexiones) y los modelos en ecuaciones diferenciales parciales son compatibles con tales extensiones.

Dentro de la teoría del análisis multiescala se distinguen tres tipos de axiomas: los axiomas arquitectónicos, el principio del máximo o principio de comparación y los axiomas morfológicos. Los primeros tratan sobre las propiedades en la construcción del análisis multiescala, el segundo sobre las características de suavizado del modelo y los terceros sobre las propiedades de conservación de bordes. A continuación los enunciamos según lo expuesto en [17].

2.1.1.1. Axiomas arquitectónicos

Para iniciar los axiomas de arquitectura se propone un principio básico: la propiedad de causalidad o también llamada arquitectura piramidal. Este axioma implica cualquier transformación a una escala mayor puede ser calculada de la imagen original o de cualquier transformación previa, evidentemente, la transformación más fina que puede aplicarse a una imagen f es la identidad. Una versión fuerte de este axioma es la propiedad de recursividad o propiedad de semigrupo

Axioma 2.1. Recursividad

$$T_0(f) = f, \quad T_s \circ T_t(f) = T_{s+t}(f) \text{ sobre } \mathbb{R}^n,$$

para todo $s, t \geq 0$ y $f \in C_b^\infty$.

Si la recursividad se satisface, el proceso visual se reduce a un bucle único para las escalas discretas, es decir, T_t puede ser deducido de la n -ésima iteración de $T_{t/n}$.

Ahora bien, en la teoría de la percepción, el análisis de percepción de una imagen consiste en una sucesión de filtros los cuales son aplicados secuencialmente. Como nuevas imágenes están llegando a la retina, el proceso de análisis de imágenes es pensado como un flujo de imágenes a través de diferentes filtros, cada uno asociado a una escala t . Por lo tanto, si se incluye una familia de operadores de transición $T_{s,t}$ indexada por $0 \leq s, t \leq \infty$ en la cual $T_t = T_{t,0}$, se puede tener una versión más débil de la causalidad:

Axioma 2.2. Causalidad

$$T_{t+s} = T_{t+s,s} \circ T_s \text{ para todo } 0 \leq s, t \leq \infty$$

Para retornar a la recursividad, es necesario asumir que $T_{t+s,t} = T_{t,0}$ y si discretizamos tal proceso, se produce una serie de operadores $S_n = T_{(n+1)h,nh}$ aplicados sucesivamente. Además, observamos que si $t = nh$, entonces $T_t = T_{nh}$ puede ser deducido de la n -ésima iteración de T_h .

El siguiente axioma establece la independencia del análisis multiescala de la elección de h , en él se asume la existencia de un generador infinitesimal A para el semigrupo T_t , definido de la siguiente manera

Axioma 2.3. Existencia de un generador

$$\frac{T_h f - f}{h} \rightarrow A[f] \text{ cuando } h \rightarrow 0^+$$

o en términos de los operadores de transición

$$\frac{T_{t+h} f - f}{h} \rightarrow A_t[f]$$

Para justificar la existencia de este generador es necesario introducir el siguiente axioma:

Axioma 2.4. Regularidad

$$\|T_t(f + gh) - (T_t f + hg)\|_\infty \leq Cht$$

para todo h, t en $[0, 1]$, $f, g \in C_b^\infty$ y donde C depende de f y de g .

Este último axioma establece una hipótesis natural de continuidad de T_t y es además una justificación fuerte para la existencia de un generador infinitesimal para el análisis multiescala.

Para dar carácter local al generador infinitesimal y al análisis multiescala se requiere la siguiente propiedad, llamada propiedad de localidad:

Axioma 2.5. Localidad

$$[T_t(f) - T_t(g)](x) = o(t) \text{ cuando } t \rightarrow 0^+$$

para todo $f, g \in C_b^\infty$ talque $D^\alpha f(x) = D^\alpha g(x)$ para todo $|\alpha| \geq 0$ y para todo x .

La notación $[T_t(f) - T_t(g)](x) = o(t)$ cuando $t \rightarrow 0^+$, significa que

$$\lim_{t \rightarrow 0^+} \frac{T_t(f) - T_t(g)}{t} = 0.$$

Rigurosamente hablando, este axioma significa que el valor de $T_t(f)$ para t pequeño, sobre cualquier punto x , es determinado por el comportamiento cerca de x .

2.1.1.2. Principio de comparación

El principio de comparación en una propiedad necesaria para preservar bordes, también conocida como principio del máximo, implica que en las transformaciones realizadas no se deben crear nuevas características a la imagen, únicamente se realiza un suavizado de la imagen original. Así, si una imagen g es más brillante que otra f , este orden es preservado.

Axioma 2.6. Principio de comparación

$$T_t(f) \leq T_t(g) \text{ sobre } \mathbb{R}^n \text{ para todo } t \geq 0 \text{ y toda } f, g \in C_b^\infty \text{ talque } f \leq g.$$

En los filtros lineales definidos por $T_t f = f * F_t$ este axioma es equivalente a la desigualdad $F_t \geq 0$. Por tanto, es la generalización en el caso no lineal de un kernel de suavizado no negativo.

Es importante tener en cuenta que el principio de comparación es natural para suavizar una imagen en escala de grises, pero si otra clase de transformación es buscada, por ejemplo una función de bordes o de profundidad, el principio del máximo no es totalmente válido, pues puede ocurrir que $g \geq f$ para una g constante mientras que f puede ser una imagen que contenga muchos bordes.

2.1.1.3. Axiomas morfológicos

A continuación enunciaremos algunos principios conocidos en la morfología matemática. Ellos establecen que el análisis de imágenes debe ser invariante bajo fluctuaciones de luz y bajo cambios de posición, orientación y escala de las curvas planas.

El primer axioma de esta clase, indica que la transformación debe ser independiente de la escala de grises.

Axioma 2.7. Invarianza bajo cambios en la escala de gris

$$T_t(0) = 0, \quad T_t(f + C) = T_t(f) + C \text{ sobre } \mathbb{R}^n$$

para todo $t \geq 0$, y $f \in C_b^\infty$ y cualquier constante C .

Esta propiedad implica que la interpretación de una imagen no depende de su tiempo de exposición, la imagen puede estar en una escala de grises más oscura o una llena de luz, pero los objetos en ella deben ser identificados igualmente. Una versión más fuerte de la invarianza bajo cambios en la escala de grises es la siguiente

Axioma 2.8. Invarianza bajo la escala de grises

$$T_t(h(f)) = h(T_t(f))$$

para todo $t \geq 0$, donde h es cualquier función real no decreciente.

La función h en este axioma implica una nueva disposición de la escala de niveles de grises de una imagen. Se asume que h es no decreciente por el hecho que si dos puntos tienen el mismo brillo en la imagen original, esta relación debe ser preservada por cualquier transformación. Observe que el axioma (2.7) es un caso particular del axioma (2.8).

Se muestran ahora una serie de axiomas que indican que el análisis multiescala debe ser invariante ante los cambios de posición de observador. Los primeros que enunciamos son las propiedades de invarianza bajo traslaciones y rotaciones, lo cual significa que que todos los puntos del espacio son equivalentes y que las características de la imagen no dependen de su ubicación.

Axioma 2.9. Invarianza bajo traslaciones

$$T_t(\tau_h \cdot f) = \tau \cdot (T_t(f))$$

para todo $h \in \mathbb{R}^n$, $t \geq 0$, $f \in C_b^\infty$, donde $(T_t(f))(x) = f(x + h)$.

Axioma 2.10. Invarianza isométrica

$$T_t(R \cdot f) = R \cdot T_t(f)$$

para todo $f, t \geq 0$ y para todas las transformaciones R definidas por $(R \cdot f)(x) = f(Rx)$ donde R es una transformación ortogonal de \mathbb{R}^n .

Ahora bien, el análisis multiescala también debe ser independiente del tamaño del objeto analizado, es importante tener en cuenta que en el mundo real la distancia implica que los objetos se visualicen en diferentes escalas, pero la forma o características de ellos no dependen de la distancia. Sea $D_\lambda f(x) = f(\lambda x)$, la invarianza bajo cambios en la escala puede ser definida como:

Axioma 2.11. Invarianza de escala

Para cualquier λ y cualquier t , existe t' tal que $D_\lambda T_{t'} = T_t D_\lambda$

Finalmente, se enuncia un axioma que implica la invarianza isométrica, la invarianza de escala y también la invarianza bajo cualquier proyección de una curva. La combinación de estas transformaciones produce una transformación lineal A del plano. Para este conjunto de transformaciones $Af(x) = f(Ax)$ se establece la siguiente propiedad.

Axioma 2.12. Invarianza de proyección

Para cualquier A y t , existe t' tal que $AT_{t'} = T_t A$

2.1.2. Forma general de un operador de análisis multiescala

En esta sección se desea mostrar la forma que debe tener un operador de análisis multiescala dependiendo de los axiomas o propiedades que cumpla. Iniciamos recordando la definición de la norma infinito, para así poder establecer nuevas y más fuertes versiones de algunos axiomas.

Sea

$$\|f\|_\infty = \sup_{x \in \mathbb{R}^n} |f(x)|$$

Como $f \leq g + \|f - g\|_\infty$, el principio de comparación (2.6) y la invarianza de escala (2.11) implican que

$$\|T_t(f) - T_t(g)\|_\infty \leq \|f - g\|_\infty \quad \text{para todo } t \geq 0, f, g \in C_b^\infty. \quad (2.1)$$

Sea $X = BUC(\mathbb{R}^n)$ el espacio de funciones acotadas uniformemente continuas sobre \mathbb{R}^n . La propiedad (2.1) indica que T_t puede ser extendida por continuidad como una función de X en $C_b(\mathbb{R}^n)$. Ahora bien, la invarianza bajo traslaciones (2.9) y la propiedad (2.1) muestran que T_t es una función que envía X sobre si mismo.

Observamos también que por densidad, la propiedad de traslación (2.9), el principio de comparación (2.6), la invarianza bajo los cambios en los niveles de gris (2.7) y (2.1) todavía se tienen para $f, g \in X$. Es posible así, formular la recursividad (2.1) para funciones $f \in X$ de la siguiente manera.

Axioma 2.13. Recursividad sobre X

$$T_0(f) = f, \quad T_s \circ T_t(f) = T_{s+t}(f) \text{ sobre } \mathbb{R}^n,$$

para todo $s, t \geq 0$ y $f \in X$.

Con el objetivo de asegurar la existencia de un generador infinitesimal (2.3), es necesario reescribir la regularidad en la siguiente forma:

Axioma 2.14. Regularidad 2

$$\|T_t(f + gh) - (T_t(f) + hg)\|_\infty \leq Cht$$

para todo $h, t \in [0, 1]$ y toda $f, g \in Q$, donde $C \geq 0$ es una constante dependiente sólo sobre Q , $Q \subset C_b^\infty$ está definido por

$$Q = \{f \in C_b^\infty, \forall n \geq 0, \|D^\alpha f\|_\infty \leq C_n \text{ para todo } |\alpha| = 0\}$$

y C_n es una sucesión creciente de constantes no negativas.

Si definimos $\delta_t(f) = \frac{T_t f - f}{t}$, podemos reescribir el enunciado anterior sobre la regularidad en la forma

$$\|\delta_t(f + hg) - \delta_t(f)\|_\infty \leq Ch$$

para todo $h, t \in [0, 1]$. Esta igualdad brinda estabilidad a los diferentes cocientes que aproximan el generador infinitesimal. Sin embargo si el objetivo es deducir la existencia de este generador que depende del tiempo, entonces es necesario imponer la regularidad a la familia de operadores de transición $T_{t+s,t}$. Así en el axioma de regularidad, el tiempo inicial $s = 0$ es reemplazado por uno arbitrario $s \geq 0$ y la regularidad toma la siguiente forma.

Axioma 2.15. Regularidad para los operadores de transición

$$\|T_{t+s,t}(f + gh) - T_{t+s,t}(f) + gh\|_\infty \leq Chs$$

Por último, es necesario adicionar un axioma que brinde estabilidad a las diferencias $(T_{t+s,t}(f) - T_{s,0}(f))/s$.

Axioma 2.16. Regularidad temporal

$$\|T_{t+s,t}(f + gh) - T_{s,0}(f)\|_\infty \leq Csn(t),$$

donde $n(t)$ es una función positiva que tiende a 0 cuando t tiende a 0, uniformemente en $s \in (0, 1)$ y donde $f \in Q$.

El siguiente teorema establece que esta hipótesis de estabilidad es suficiente para asegurar la existencia de un generador para el análisis multiescala.

Teorema 2.1. [Parte I] Si el análisis multiescala T_t satisface los axiomas de:

1. Invarianza de traslación.
2. Principio de comparación.
3. Invarianza bajo cambios en el nivel de gris.
4. Recursividad.
5. Regularidad.

Entonces T_t también satisface la propiedad del generador. Además la convergencia del generador es uniforme para $f \in Q$ y $D^\alpha f_n \rightarrow D^\alpha f$ uniformemente sobre \mathbb{R}^n par todo $|\alpha| > 0$.

[Parte II] Si el análisis multiescala T_t satisface los axiomas de:

1. Invarianza de traslación.
2. Principio de comparación.
3. Invarianza bajo cambios en el nivel de gris.
4. Causalidad.
5. Regularidad para lo operadores de transición.
6. Regularidad temporal.

Entonces la conclusión de la parte [I] también es válida con la siguiente adaptación para el generador

$$\|T_t(f) - f\|_\infty \leq Ct \text{ para todo } t \in [0, 1], f \in Q.$$

Observación. Si T_t es un operador lineal, la propiedad de regularidad (2.14) se reduce a

$$\|T_t(f) - f\|_\infty \leq Ct \text{ para todo } t \in [0, 1], f \in Q,$$

lo cual indica que para funciones suaves las órbitas son Lipschitz continuas en $t = 0$.

Demostración. La demostración de este teorema se divide en cuatro pasos. Antes de iniciar, se realizarán algunas convenciones en la notación. Sea

$$\delta_{t,s}(f) = \frac{T_t(f) - T_s(f)}{t - s}$$

y

$$\delta_t(f) = \delta_{t,0}(f) = \frac{T_t(f) - f}{t}.$$

Esta notación nos permite escribir la propiedad de regularidad (2.14) en la forma

$$\|\delta_t(f + hg) - \delta_t(f)\|_\infty \leq Ch \text{ para toda } t, h \in [0, 1] \text{ y } f, g \in Q. \quad (2.2)$$

[Paso 1.] En este paso probamos que $\delta_t(f)$ es uniformemente acotada.

Por la propiedad de invarianza de bajo cambios en la escala de grises tenemos que $T_t(0) = 0$. Usando la regularidad

$$\|T_t(f + gh) - (T_t(f) + hg)\|_\infty \leq Cht$$

Para $h = 1$, $f = 0$ y $g = f$, tenemos

$$\|T_t(0 + f) - (T_t(0) + f)\|_\infty \leq C_0t$$

Lo que es equivalente a

$$\|T_t(f) - f\|_\infty \leq C_0t, \quad (2.3)$$

para alguna constante C_0 que dependa de sólo de los límites de las derivadas de f en C_b^∞ . Así

$$\delta_t = \left\| \frac{T_t(f) - f}{t} \right\|_\infty \leq C_0,$$

y nos lleva a afirmar que $\delta_t(f)$ es uniformemente acotada.

[Paso 2.] En este paso demostraremos que $\delta_t(f)$ es Lipschitz uniformemente continua para $t \in (0, 1]$ y $f \in Q$.

Sea $z \in \mathbb{R}^n$ tal que $|z| = 1$. Usando la propiedad de invarianza bajo traslaciones (2.9) tenemos que

$$\tau_{hz}\{\delta_t(f)\} = \delta_t(\tau_{hz} \cdot f).$$

Tomando $\tau_{hz}(f) = f + hg_h$ para algún $g_h \in C_b^\infty$ cuyas derivadas sólo dependen de los límites de las derivadas de f y aplicando la propiedad (2.2) tenemos que

$$\|\delta_t(\tau_{hz}f) - \delta_t(f)\|_\infty \leq Ch,$$

por lo tanto

$$\begin{aligned} \delta_t(f) \text{ es Lipschitz, uniformemente continua sobre } \mathbb{R}^n \text{ para } t \in (0, 1] \\ \text{y uniformemente continua para } f \in Q. \end{aligned} \quad (2.4)$$

[Paso 3.] En este paso, se probará una propiedad de compacidad para $\delta_t(g)$ cuando t tiende a 0.

Iniciamos escribiendo

$$\delta_s = \delta_s(f) * \rho_\epsilon \{ \delta_s(f) * \rho_\epsilon - \delta_s(f) \}$$

donde $\rho_\epsilon = \epsilon^{-n} \rho(f/\epsilon)$, $\epsilon \geq 0$, $\int_{\mathbb{R}^n} \rho(y) dy = 1$, $\rho \in C_0^\infty(\mathbb{R}^n)$.

Usando la propiedad (2.4) se obtiene que

$$\|\delta_s(f) * \rho_\epsilon - \delta_s(f)\|_\infty \leq C_1 \epsilon \text{ para todo } \epsilon \geq 0, \quad (2.5)$$

para algún $C_1 \geq 0$ que dependa únicamente de los límites de las derivadas de f .

Teniendo en cuenta las propiedades (2.5), la propiedad (2.1) y la recursividad (2.13), podemos concluir que

$$\|T_{t+s}(f) - T_t(f + s\delta_s(f) * \rho_\epsilon)\|_\infty \leq C_1 \epsilon s, \text{ para todo } \epsilon \geq 0. \quad (2.6)$$

Ahora bien, si tenemos en cuenta que $\delta_s(f) * \rho_\epsilon \in C_b^\infty$, la propiedad de regularidad (2.14) implica que

$$\|T_t(f + s\delta_s(f) * \rho_\epsilon) - T_t(f) - s\delta_s(f) * \rho_\epsilon\|_\infty \leq C_\epsilon s t, \quad (2.7)$$

para algún $C_\epsilon \geq 0$ que depende sólo de $\epsilon > 0$ y los límites de las derivadas de f .

Finalmente, si se consideran las propiedades (2.5), (2.6) y (2.7) se deduce que

$$\|\delta_t(f) - \delta_g(f)\|_\infty \leq 2C_1 \epsilon + C_\epsilon t$$

lo que implica que

$$\|\delta_{t+s}(f) - \delta_s(f)\|_\infty \leq m(t) \text{ para todo } t, s \in [0, 1/2] \quad (2.8)$$

para alguna función m continua, no decreciente, no negativa sobre $[0, 1/2]$ tal que $m(0) = 0$ y que dependa sólo de las derivadas de f . Específicamente $m(t) = \inf_{\epsilon \in (0, 1]} (2C_1\epsilon + C_\epsilon t)$. Lo que demuestra, que toda la sucesión de operadores converge al mismo límite.

[Paso 4.] Gracias a la estimación de Cauchy obtenida en el paso anterior, en este cuarto y último paso, de dará una estimación de Cauchy para δ_s en $0 < h \leq t \leq 1/2$.

Denotando $N = [t/h]$, se observa que $\delta_t(f) = (Nt/h)\delta_{Nh}(f) + (r/t)\delta_{Nh+r, Nh}(f)$ y usando la propiedad (2.8) se deduce que

$$\left\| \delta_t(f) - \frac{Nh}{t}\delta_{Nh}(f) - \frac{r}{t}\delta_r(f) \right\|_\infty \leq \frac{r}{t}m(Nh). \quad (2.9)$$

Escribiendo $\delta_{Nh}(f) = ((N-1)h/Nh)\delta_{(N-1)h}(f) + (h/Nh)\delta_{Nh, (N-1)h}(f)$ y usando la propiedad (2.8) con $t = (N-1)h$, $s = h$, se deduce que

$$\left\| \delta_{Nh}(f) - \frac{N-1}{N}\delta_{(N-1)h}(f) - \frac{1}{N}\delta_h(f) \right\|_\infty \leq \frac{1}{N}m((N-1)h). \quad (2.10)$$

Combinando (2.10) y (2.9), se obtiene que

$$\left\| \delta_t(f) - (N-1)\frac{h}{t}\delta_{(N-1)h}(f) - \frac{h}{t}\delta_h(f) - \frac{t}{t}\delta_r(f) \right\|_\infty \leq \frac{r}{t}m(Nh) + \frac{h}{t}m((N-1)h). \quad (2.11)$$

Reiterando $N-1$ veces el argumento que lleva a (2.10), se obtiene que

$$\left\| \delta_t(f) - \frac{Nh}{t}\delta_h(f) - \frac{r}{t}\delta_r(f) \right\|_\infty \leq \frac{r}{t}m(Nh) + \frac{h}{t}m(jh). \quad (2.12)$$

En particular, teniendo en cuenta (2.3) y que m es una función no decreciente, se deduce que

$$\|\delta_t(f) - \delta_h(f)\|_\infty \leq \left(\frac{r}{t} + (N-1)\frac{h}{t} \right) m(t) + 2\frac{r}{t}C_0.$$

Esto implica que si $t = Nh + r$, se tiene que

$$\|\delta_t(f) - \delta_h(f)\|_\infty \leq 2\frac{C_0 r}{t} + m(t). \quad (2.13)$$

A partir de esto, (2.3) y (2.4), se selecciona $h_n \rightarrow 0$ tal que $\delta_h(f)$ converja uniformemente sobre un conjunto compacto a una función Lipschitz acotada sobre \mathbb{R}^n , la cual denotamos por $A[f]$, por tanto podemos decir que

$$\|\delta_t(f) - \delta_h(f)\|_\infty \leq m(t), \quad (2.14)$$

probando así la primera parte del teorema.

Para la prueba de la segunda parte, se consideran los operadores $T_{t,s}$ y se asume que $s = 0$. El único paso que se debe modificar es el tercer paso, pues es el único paso donde se usa la recursividad. En este caso, usamos la regularidad temporal (2.16) y obtenemos que para todo $t, s \in [0, 1/2]$

$$\|\delta_{t+s,t}(f) - \delta_s(f)\|_\infty \leq m(t)$$

para la función m continua, no negativa, no decreciente sobre $[0, 1/2]$ talque $m(0) = 0$, que depende únicamente de las derivadas de f , dada por

$$m(t) = \inf_{\epsilon \in (0,1]} (2C_1\epsilon + C_\epsilon t) + C_n(t)$$

□

Para obtener y demostrar cuál es la forma precisa de un operador multiescala, es necesario reescribir la propiedad de localidad (2.5) bajo ciertas hipótesis. Iniciamos suponiendo que el operador de análisis multiescala satisface la propiedad de causalidad.

Axioma 2.17. Localidad bajo la hipótesis de causalidad

$$\{T_{t,s}(f) - T_{t,s}(g)\}(x) = o(t-s) \text{ cuando } t-s \rightarrow 0^+$$

para toda $f, g \in C_b^\infty$ talque $D^\alpha f(x) = D^\alpha g(x)$ para todo x y para todo $|\alpha| \geq 0$.

Continuamos suponiendo que el análisis multiescala admite un generador infinitesimal. En este caso, la propiedad de localidad puede ser escrita en la siguiente forma.

Axioma 2.18. Localidad bajo la hipótesis de un generador

$$A[f](x) = A[g](x)$$

para toda $f, g \in C_b^\infty$ talque $D^\alpha f(x) = D^\alpha g(x)$ para todo x y para todo $|\alpha| \geq 0$.

Lo que significa que el valor de $T_t(f)$, para un t pequeño en cualquier punto x , es determinado por el comportamiento de f cerca de x .

El interés con el siguiente teorema es estudiar, bajo qué condiciones existe una función continua F definida sobre $S^n \times \mathbb{R}^n \times \mathbb{R}$ (donde S^n denota el espacio de matrices simétricas de tamaño $n \times n$) la cual satisface la ecuación

$$F(A, p, t) \geq F(B, p, t) \text{ para todo } p \in \mathbb{R}^n, A, B \in S^n \text{ con } A \geq B. \quad (2.15)$$

donde $A \geq B$ significa que $A - B \geq 0$ y cumple la condición

$$A_t[f] = F(D^2f, Df, t) \text{ para toda } f \in C_b^\infty. \quad (2.16)$$

Para finalizar, se relaciona el semigrupo T_t a la solución de la ecuación no lineal parabólica de segundo orden

$$\frac{\partial u}{\partial t} - F(D^2u, Du, t) = 0 \text{ en } \mathbb{R}^n \times [0, \infty), \quad u(0, x) = u_0(x) \text{ en } \mathbb{R}^n. \quad (2.17)$$

La existencia y unicidad de la solución en (2.17) es conocida como una solución débil de la ecuación (2.17). Se dividirá el resultado en tres partes, enunciadas en el siguiente teorema.

Teorema 2.2. [I] Si T_t satisface las propiedades de

- Invarianza bajo traslaciones.
- El principio de comparación.
- La invarianza bajo cambios en la escala de grises.
- Recursividad.
- Regularidad.
- Localidad.

Entonces, existe una función continua F sobre $S^n \times \mathbb{R}^n$ satisfaciendo (2.15) talque $(T_t f - f)/t \rightarrow F(D^2f, Df)$ y además (2.16) también se satisface.

Además, $(T_t f - f)/t \rightarrow F(D^2f, Df)$ es uniformemente continua sobre \mathbb{R}^n cuando $t \rightarrow 0^+$, uniformemente sobre \mathbb{R}^n , uniformemente para f en C_b^∞ con un módulo uniforme de continuidad para sus segundas derivadas.

[II] Sea T_t un análisis multiescala donde los operadores $T_{t,s}$ satisfacen

- Invarianza bajo traslaciones.
- El principio de comparación.
- La invarianza bajo cambios en la escala de grises.
- Regularidad temporal.
- Regularidad para los operadores de transición.
- Localidad bajo la hipótesis de causalidad.
- Causalidad.

Entonces existe una función continua $F(A, p, t)$ que satisface (2.15) y (2.16) para todo $t \geq 0$.

[III] Para cualquier u_0 in X , $u(x, t) = T_t(u_0)(x)$ es la única solución viscosa de $\frac{\partial u}{\partial t} = F(D^2u, Du, t)$ en $BUC_x(\mathbb{R}^n \times [0, \infty))$.

Observación. Notamos $BUC_x(\mathbb{R}^n \times [0, \infty))$ el espacio de funciones continuas y acotadas sobre $\mathbb{R}^n \times [0, \infty)$ que son uniformemente continuas en x , uniformemente en t .

Observación. Es posible mostrar que cualquier función continua F satisfaciendo (2.15) produce un único semigrupo viscoso T_t el cual satisface la invarianza de traslación, el generador, y la Localidad.

Demostración Sea $f, g \in C_b^\infty$ satisfaciendo que

$$\begin{aligned} f(0) = g(0) = 0, \quad Df(0) = Dg(0) = p \in \mathbb{R}^n, \\ D^2f(0) = D^2g(0) = A \in S^n. \end{aligned} \tag{2.18}$$

La prueba inicia probando que $A[f](0) = A[g](0)$. Para esto, se define $z \in C_b^\infty$ tal que $z \geq 0$, $z(x) = |x|^2$ cerca de 0 y $f^\epsilon = f + \epsilon z$. Donde, $f^\epsilon \geq g$ para $|x| \leq c\epsilon$ con $c > 0$.

Consideramos el conjunto $e_\epsilon = w(x/\epsilon)$ donde $w \in C_b^\infty$, $0 \leq w \leq 1$ sobre \mathbb{R}^n , $w(x) = 1$ si $|x| \leq c/2$, $w(x) = 0$ si $|x| > c$ y finalmente introducimos $f^\epsilon = w_\epsilon f^\epsilon + (1 - w_\epsilon)g$. Esta función f^ϵ cumple dos propiedades: la primera todas sus derivadas en 0 son iguales a las derivadas de f^ϵ y la segunda, $f^\epsilon \geq g$ sobre \mathbb{R}^n , usando el principio de comparación, tenemos que $T_t(f^\epsilon) \geq T_t(g)$ sobre \mathbb{R}^n para todo $t \geq 0$.

Ahora bien, como $f^\epsilon(0) = f(0) = g(0) = 0$, se deduce que

$$A[f^\epsilon](0) = A[g](0)$$

pero, dado que las derivadas en 0 son iguales a las derivadas de f^ϵ y hipótesis de localidad, se obtiene que $A[f^\epsilon](0) = A[f^\epsilon](0)$.

Usando la primera parte del teorema (2.1) se deduce que $A[f^\epsilon](0) \rightarrow A[f](0)$, y por simetría

$$A[f](0) = A[g](0).$$

Para demostrar que F satisface (2.15) se usa un argumento similar, considerado $A \geq B$ y ajustando

$$f(x) = [(p, x) + \frac{1}{2}(Ax, x)]w(x)$$

$$g(x) = [(p, x) + \frac{1}{2}(Bx, x)]w(x)$$

Además, $f \geq g$ sobre \mathbb{R}^n mientras que $f(0) = g(0)$ y por lo tanto

$$\begin{aligned} F(A, p) = A[f](0) &= \lim_{t \rightarrow 0^+} \frac{T_t(f)(0) - f(0)}{t} \\ &\geq \lim_{t \rightarrow 0^+} \frac{T_t(g)(0) - g(0)}{t} = A[g](0) = F(B, p) \end{aligned}$$

Usando la última conclusión del teorema (2.1), llamamos $A[f_n] \rightarrow A[f]$ cuando $n \rightarrow \infty$ uniformemente si $f_n, f \in Q$ y $D^\alpha f_n \rightarrow D^\alpha f$ uniformemente sobre \mathbb{R}^n para todo $|\alpha| \geq 0$, también deducimos que F es continua sobre $S^n \times \mathbb{R}^n$.

Un vez esto este probado, observamos que la invarianza bajo traslaciones y la invarianza bajo el cambio en la escala de niveles de gris implican que

$$A[\tau_h \cdot f] = \tau_h(A[f]), A[f + C] = A[f] \text{ para todo } f \in C_b^\infty, h \in \mathbb{R}^n, C \in \mathbb{R}. \quad (2.19)$$

y por lo tanto

$$A[f] = F(D^2 f, Df) \quad (2.20)$$

para alguna función F sobre $S^n \times \mathbb{R}^n$.

Para probar la segunda parte del teorema, se usa la parte II del teorema (2.1) y la continuidad de $F(D^2u, Du, t)$ con respecto a t se obtiene gracias a la regularidad temporal.

Para la tercera y última parte, con el objetivo de probar que $u(t, x) = T_t(u_0)(x)$ es una solución viscosa de (2.17) es suficiente probar que u es una solución viscosa de (2.17). Sea $\phi \in C_b^\infty(\mathbb{R}^n \times [0, T])$ para cualquier $T < \infty$ y sea (x_0, t_0) en $\mathbb{R}^n \times [0, \infty)$ un punto máximo global de $u - \phi$. Es necesario mostrar que

$$\frac{\partial \phi}{\partial t}(x_0, t_0) - F(D^2\phi(x_0, t_0), D\phi(x_0, t_0)) \leq 0 \quad (2.21)$$

Asumiendo que $u(x_0, t_0) = \phi(x_0, t_0)$, $u \leq \phi$ sobre $\mathbb{R}^n \times [0, \infty)$. Ahora, asumiendo que ϕ es de la forma

$$\phi(t, x) = f(x) + g(t) \quad (2.22)$$

donde $g(t_0) = 0$, $f(x_0) = u(x_0, t_0)$ y $f \in C_b^\infty(\mathbb{R}^n)$, $g \in C^\infty([0, \infty))$. Entonces para $h > 0$, consideramos $\phi(x_0, t_0) = u(x_0, t_0) = T_h(u(t_0 - h))(x_0)$ donde establecemos $u(t) = u(x, t)$ para todas las funciones $u \in \mathbb{R}^n \times [0, \infty)$. En esta desigualdad, usamos la propiedad de recursividad, luego el principio de comparación y la invarianza bajo cambios en la escala de grises para obtener que

$$T_h(u(t_0 - h)) \leq T_h(\phi(t_0 - h)) \leq T_h(f) + g(t_0 - h)$$

por lo tanto,

$$\frac{1}{h}[g(t_0) - g(t_0 - h)] + \frac{1}{h}[f - T_h f] \leq 0$$

y cuando $h \rightarrow 0$, usando el generador del teorema (2.1), tenemos que

$$g'(t_0) - F(D^2f(x_0), Df(x_0)) \leq 0$$

y 2.21 es probado gracias a que $\frac{\partial \phi}{\partial t}(t_0, x_0) = g'(t_0)$ y $D^\alpha \phi(t_0, x_0) = D^\alpha f(x_0)$ para $|\alpha| = 1, 2$. \square

2.1.3. Análisis axiomático de los modelos clásicos

Gracias al resultado obtenido en el teorema (2.2) se puede deducir que la ecuación del calor $\frac{\partial u}{\partial t} = \Delta u$ es el único modelo multiescala lineal que satisface la invarianza isométrica, debido a su dependencia única de las derivadas de u . Además, este teorema también nos muestra que para los modelos en procesamiento de imágenes, la linealidad y la invarianza bajo la escala de grises son propiedades compatibles.

En el primer modelo de difusión dado por Koenderink se observó que la convolución de la señal con un filtro Gaussiano en cada escala es equivalente a la solución de la ecuación del calor con la imagen original como dato inicial. Si este dato es denotado con u_0 , el espacio de escala asociado a u_0 consiste en la solución del sistema

$$\frac{\partial u}{\partial t} = \Delta u(x, y, t), \quad u(x, y, 0) = u_0(x, y)$$

La solución de esta ecuación para un dato inicial con norma acotada es $u(t, x, y) = G_t * u_0$ donde G es la función de Gauss

$$G_t(x, y) = (4\pi)^{-1} t^{-1} e^{-(x^2+y^2)/4t}$$

Uno de los principales objetivos de la teoría de la difusión es la detección de bordes. Inicialmente se creía, según Mirr [5], que (x, y) es un punto frontera para la escala t si $\Delta u(x, y)$ cambia de signo y $|Du(t, x, y)|$ es grande. Sin embargo, esta condición introduce que un límite a priori deber ser definido. Desafortunadamente, en las escalas bajas del espacio de escala esta propiedad no se cumple. De otro lado, si consideramos un filtro con varianza pequeña todos los bordes mantienen su posición correcta, pero se debe tener en cuenta que los bordes principales de la imagen se pueden confundir por el ruido o la textura de la misma.

El espacio de escala propuesto por Witkin [1], propone identificar los bordes en una escala baja y luego mantenerlos por medio de una escala decreciente. Este método puede mostrar teóricamente la ubicación exacta de todos los bordes principales pero su implementación es bastante complicada desde el punto de vista computacional pues debe involucrar todos los bordes a través de la escala y los múltiples límites involucrados en la detección de bordes en cada valor de la escala.

En el siguiente teorema se enuncian las propiedades que debe satisfacer el espacio de escala para satisfacer la ecuación planteada por Witkin.

Teorema 2.3. Sea T_t un análisis multiescala que satisface las propiedades de

- Causalidad.
- Regularidad para los operadores de transición.
- Invarianza bajo traslaciones.
- Invarianza bajos cambios de la escala de grises.
- Regularidad temporal.
- Principio de comparación.

Y además si los operadores de transición $T_{t,s}$ son lineales y satisfacen la invarianza isométrica entonces $u(x, t) = (T_t u_0)(x)$ es la solución de la ecuación del calor

$$\frac{\partial u}{\partial t} - c\Delta u = 0$$

en $\mathbb{R}^n \times [0, \infty)$, $u(x, 0) = u_0(x)$ en \mathbb{R}^n , donde c es una constante positiva.

Demostración. Iniciamos con el caso independiente de la escala. Dado que

$$F(D^2u, Du) = \lim_{t \rightarrow 0} \frac{T_t u - u}{t},$$

F es lineal en u y por lo tanto satisface la ecuación

$$F(rD^2y + sD^2v, rDu + sDv) = rF(D^2u, Du) + sF(D^2v, Dv)$$

para cualquier $r, s \in \mathbb{R}$ y cualquier función $u, v \in X$ en cualquier punto x . Como los valores de Du, Dv, D^2u, D^2v son arbitrarios y pueden ser independientemente tomados para ser 0, obtenemos para cualquier par de vectores p y p' y matrices simétricas A y A' que

$$F(rA + sA', rp + sp') = rF(A, p) + sF(A', p'), \quad F(A, p) = F(A, 0) + F(0, p).$$

Así $F(A, p) = F'(p) + F''(A)$ y por lo tanto F' y F'' son lineales.

Gracias a la invarianza isométrica, obtenemos que $F(R^T AR, R^T p) = F(A, p)$ para cualquier isometría R de \mathbb{R}^n . Tomando $A = 0$ se obtiene que $F(Rp) = F(p)$ y así $F'(Rp) = F'(p)$ para cualquier isometría R .

Como F' es lineal, $F'(p) = 0$ para cualquier p . Así $F(A, p) = F''(A)$. Usando nuevamente la invarianza isométrica, se tiene que $F''(R^T AR) = F''(A)$ para cualquier isometría R y cualquier matriz simétrica A . como cada matriz simétrica puede ser diagonalizada por una bases ortonormal y cada par de bases ortonormales puede ser intercambiado por una isometría, vemos que F'' sólo depende de los valores propios $\lambda_1, \lambda_2, \dots, \lambda_n$ de A . Así $F''(A) = F''(\lambda_1, \lambda_2, \dots, \lambda_n)$ debe ser invariante bajo permutaciones de los valores propios de A . Con ello, F depende únicamente de la función simétrica de los auto valores.

Ahora, la única función lineal simétrica es la suma, por lo tanto, $F''(A)$ sólo depende linealmente de la traza de A y por lo tanto $F''(A) = c \text{Tr}(A)$ para alguna constante c . Con esto concluimos que $F(D^2u, Du) = c\Delta u$. Como F debe ser creciente en A , la constante c es no negativa. Esto completa la prueba en el caso de una arquitectura recursiva. En el caso de una arquitectura causal, podemos aplicar exactamente el mismo razonamiento pero se obtiene que $F(D^2u, Du, t) = c(t)\Delta u$ para alguna función continua no negativa $c(t)$, lo cual nos lleva nuevamente la ecuación del calor.

2.1.3.1. El modelo de difusión anisotrópica de Perona-Malik

El modelo propuesto por Perona-Malik [4], presenta una importante mejora del análisis multiescala lineal clásico por su detección de bordes más precisa. El modelo fue planteado por medio de ecuaciones diferenciales parciales y los autores establecieron el principio del máximo como requerimiento básico en el procesamiento de imágenes.

La idea de Perona-Malik es introducir en el modelo una parte del proceso de detección de bordes dentro del mismo filtro, permitiendo una interacción entre las escalas iniciales del algoritmo. La propuesta es en esencia cambiar la ecuación del calor por la ecuación no lineal

$$\frac{\partial u}{\partial t} = \operatorname{div}(g(|Du|)Du), \quad u(0) = u_0 \quad (2.23)$$

En esta ecuación, g es función suave no creciente con $g(0) = 1$, $g(s)$ tendiendo a 0 cuando s tiende a ∞ . La motivación de este proceso de suavizado es condicionar la difusión:

- Si $Du(x, y)$ es grande, entonces la difusión es pequeña y por lo tanto la localización exacta de los bordes es alcanzada.
- Si $Du(x, y)$ es pequeña, entonces la difusión se incrementa más alrededor de (x, y) . Así la elección de g corresponde a un umbral que se compara con el límite de $|Du|$.

Como este límite introduce un dispositivo no lineal es natural usarlo antes en el método, en el proceso de suavizado mismo. Los resultados experimentales obtenidos por Perona-Malik son visualmente impresionantes y muestran que una función de reconocimiento de bordes permite que los mismos permanezcan más estables a través de las escalas.

Sin embargo, la difusión anisotrópica de Perona-Malik no es propiamente una difusión en dirección ortogonal al gradiente por lo cual crea difusión en la dirección de este vector. En efecto,

$$\operatorname{div}(g(|Du|)Du) = g(Du)\Delta u + g'(|Du|)|Du|^{-1}D^2u(Du, Du).$$

Si se considera un término de difusión en la dirección ortogonal a Du

$$u_{\xi\xi} = (u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2)/|Du|^2 = D^2u(Du^\perp/|Du^\perp|, Du^\perp/|Du^\perp|),$$

y un término de difusión en la dirección de $|Du|$

$$u_{\eta\eta} = (u_{xx}u_x^2 - 2u_{xy}u_xu_y + u_{yy}u_y^2)/|Du|^2 = D^2u(Du/|Du|, Du/|Du|),$$

La difusión anisotrópica queda reescrita como

$$\operatorname{div}(g(|Du|)Du) = g(Du)u_{\xi\xi} + G'(|Du|)u_{\eta\eta}, \quad \text{donde } G(s) = sg(s).$$

Dado que g es una función positiva, el primer término de la difusión siempre es una difusión a lo largo de los bordes. Mientras que el coeficiente de difusión en la dirección perpendicular a los bordes puede ser:

- Positivo si $G'(|Du|) > 0$. En este caso, la ecuación se comporta localmente como una difusión en ambas direcciones.
- Cero si $G'(|Du|) = 0$. Lo que implica que la difusión se da exactamente en la dirección ortogonal al gradiente.
- Negativo si $G'(|Du|) < 0$. Esto corresponde a una inversión local de la ecuación del calor, en este caso hay una combinación de suavidad y choque en la ecuación.

Este modelo ha sido desarrollado por Osher y Rudin [19].

En cuanto a las propiedades del modelo, se inicia con la buena colocación, so $G'(t) > 0$ para algún t , entonces las imágenes indagan por una ecuación del calor inversa, en la cual no hay solución única ni estabilidad. Si $G'(t) \geq 0$, entonces la ecuación tiene dos buenas propiedades: unicidad en la solución y cumplimiento del principio de comparación.

En cuanto a los axiomas que se satisfacen, se pueden establecer los axiomas de arquitectura y las propiedades elementales de invarianza. La invarianza de escala es cierta si y sólo si g es una función potencia, pero la invarianza bajo la escala de grises nunca se tiene. En efecto, la ecuación puede ser reescrita como

$$\frac{\partial u}{\partial t} = F(D^2u, Du) \text{ con } F(A, p) = g'(|p|)|p|^{-1}A(p, p) + g(|p|)\text{trace}(A).$$

La invarianza bajo la escala de grises implica que $F(tA, tp) = tF(A, p)$. Escogiendo A como una matriz simétrica talque $Ap = 0$ y $\text{tr}(A) \neq 0$, se obtiene que $g(t|p|) = g(p)$ para todo $t > 0$, así g es una función constante y el modelo es la ecuación del calor, el cual no es invariante bajo la escala de grises.

2.1.4. Modelo basado en la curvatura

Álvarez, lions y Morel proponen en [17], un modelo con una ecuación diferencia no lineal parabólica, cuya idea surge del modelo de Perona-Malik,

$$\frac{\partial u}{\partial t} = |Du| \operatorname{div} \left(\frac{Du}{|Du|} \right), \quad u(x, y, 0) = u_0(x, y)$$

donde $u_0(x, y)$ es el nivel de gris de la imagen que sera procesado, $u(x, y, t)$ es su versión suavizada dependiendo del parámetro de la escala t .

Para el planteamiento del nuevo modelo, los términos de la ecuación son interpretados de la siguiente manera:

El término $|Du| \operatorname{div}(Du/|Du|) = \Delta u - D^2u(Du, Du)/|Du|^2$ representa un término de difusión degenerativa, el cual difunde a u únicamente en dirección ortogonal a su gradiente Du . El objetivo de esta difusión degenerativa es hacer que u se suavice en ambos lados del borde con un suavizado minimal dentro del borde mismo. Esto significa que para tal teoría las fronteras de las curvas de nivel de la imagen están representando los bordes de la misma, por lo que la ecuación responde a los requerimientos del modelo Perona-Malik para una difusión anisotrópica pero no presenta más que una velocidad adaptativa de la difusión.

En la expresión $|Du| \operatorname{div}(Du/|Du|) = \Delta u - D^2u(Du, Du)/|Du|^2$, observamos que el primer término, el Laplaciano, es el mismo que en la teoría del análisis de escala mientras que el segundo es una inhibición de la difusión en la dirección del gradiente.

Si usamos ξ para denotar las coordenadas asociadas con la dirección ortogonal al Du , la ecuación anterior se puede formular con respecto a esta nueva coordenada

$$\frac{\partial u}{\partial t} = u_{\xi\xi},$$

donde, por supuesto, ξ depende de Du . Explícitamente,

$$\frac{\partial u}{\partial t} = \frac{1}{u_x^2 + u_y^2} (u_y^2 u_{xx} - 2u_x u_y u_{xy} + u_x^2 u_{yy}). \quad (2.24)$$

De esta ecuación se puede interpretar que las curvas de nivel de la solución se mueven en la dirección normal, a una velocidad proporcional a su propia curvatura, (en dos dimensiones). Esta propiedad implica que el nuevo análisis multiescala satisface el principal axioma de morfología: la invarianza en la escala de grises.

Si el análisis multiescala satisface los axiomas de invarianza en la escala de nivel de gris y si la dimensión $n = 2$, F puede ser reescrito como

$$\frac{\partial u}{\partial t} = |Du| G \left(\operatorname{curv} u, \frac{Du}{|Du|}, t \right)$$

Donde G es no decreciente con respecto al primer término y

$$\operatorname{curv} (u)(x) = \operatorname{div} \left(\frac{Du}{|Du|} \right) = |Du|^{-1} \left(\Delta u - D^2u \left(\frac{Du}{|Du|}, \frac{Du}{|Du|} \right) \right).$$

Este término puede ser interpretado como la curvatura de la curva de nivel $\{y, u(y) = u(x)\}$ pasando a través de x . Así, se obtiene una descripción general del operador multiescala morfológico.

Los únicos operadores ya conocidos en esta clase son los operadores de morfología los cuales corresponden al caso donde G no depende de la curvatura de u y tienen una forma especial $\frac{\partial u}{\partial t} = \pm \|Du\|_B$. Si se adiciona la propiedad de invarianza isométrica, entonces la ecuación que resulta para dos dimensiones es

$$\frac{\partial u}{\partial t} = |Du| G(\text{curv } u, t).$$

Si además se asume la invarianza de escala, entonces la ecuación toma la forma

$$\frac{\partial u}{\partial t} = |Du| \beta(t \text{ curv } u).$$

donde β es una función continua no decreciente. Por último, se impone la invarianza bajo proyecciones la ecuación se resultante es

$$\frac{\partial u}{\partial t} = |Du|(t \text{ curv } u)^{1/3}.$$

Capítulo 3

Modificación del modelo de Perona-Malik

Este capítulo se dedicará a presentar algunas variaciones experimentales en el modelo plantado por Perona-Malik y la comparación de los resultados obtenidos con el modelo original.

Para iniciar, presentaremos la formulación discreta del problema y su implementación en Matlab.

3.1. Modelo de Perona-Malik y su versión discreta

El modelo propuesto por Perona-Malik está dado por la ecuación

$$\begin{cases} \frac{\partial I(x, y, t)}{\partial t} = \operatorname{div} [c(\|\nabla I\|) \nabla I] \\ I(x, y, 0) = I_0(x, y) \end{cases}$$

donde $\|\nabla I\|$ es la magnitud del gradiente y $c(\|\nabla I\|)$ es una función de detección de bordes, llamada coeficiente de difusión. Este coeficiente se elige para satisfacer la condición $c(x) \rightarrow 0$ cuando $x \rightarrow \infty$.

Este modelo puede verse de manera discreta por medio de la ecuación

$$I_s^{t+1} = I_s^t + \frac{\lambda}{|\eta_s|} \sum_{p \in \eta_s} c(\nabla I_{s,p}) \nabla I_{s,p}, \quad (3.1)$$

donde I_s^t representa el valor de la iteración t de la imagen en el pixel s . La constante $\lambda \in \mathbb{R}^+$ es el escalar que determina la tasa de difusión, η_s representa la vecindad espacial

del pixel s que se usará en la transformación y $|\eta_s|$ es el número de vecinos en tal vecindad. $\nabla = I_{s,p}$ representa el gradiente de la imagen en el punto p y en la dirección de s a p , específicamente puede verse como

$$\nabla = I_{s,p} = I_p - I_s^t$$

3.1.1. Implementación en Matlab

Para la implementación de código en Matlab, hemos usado como referencia la implementación del modelo de Perona-Malik propuesta por Daniel Simoes Lopes en [22], realizando los cambios necesarios para que el coeficiente de difusión pueda ser cambiado por los propuestos en este trabajo. Los pasos relevantes para esta implementación son:

1. Convertir la imagen de entrada en una imagen de doble precisión.
2. Usar esta imagen como la condición inicial del programa.
3. Definir la distancia del centro a cada uno de los puntos de la vecindad.
4. Definir las matrices de convolución que se usarán para calcular el gradiente de la imagen en cada una de las direcciones deseadas.

Ahora, en cada iteración (cada valor del espacio de escala) tenemos los siguientes pasos:

1. Calculamos $\nabla I_{s,p}$ en cada una de las direcciones establecidas.
2. Calculamos $c(\nabla I_{s,p})$ en cada dirección.
3. Calculamos I_s^{t+1}

En este código, usamos un kernel 8 vecinos del pixel (Matriz 3×3) en las direcciones norte (N), sur (S), este (E), oeste (W) y las direcciones diagonales noreste (NE), sureste (SE), noroeste (NW), suroeste (SW). Por lo tanto el valor de $|\eta_s| = 8$ y el valor de λ es el recíproco del cuadrado de la distancia a cada vecino. Los argumentos de entrada de la función serán la imagen a transformar, el número de iteraciones, el valor de kappa (que limita el valor del gradiente que controla la difusión) y por último la función que hará el papel de coeficiente de difusión.

```

1 function diff_im = Perona(im, num_iter, kappa, CD)
2
3
4 % Paso 1 general. Convertir im en una imagen de doble precisión.
5 im = double(im);
6
7 % Paso 2 general. Introducir la condicion inicial de la ecuación.
8 diff_im = im;
9

```

```

10 %Paso 3 general. Distancia del centro a cada uno de los puntos de la vecindad
11     .
12 dx = 1;
13 dy = 1;
14 dd = sqrt(2);
15
16 %Paso 4 general. Matrices de convolución.
17 hN = [0 1 0; 0 -1 0; 0 0 0];
18 hS = [0 0 0; 0 -1 0; 0 1 0];
19 hE = [0 0 0; 0 -1 1; 0 0 0];
20 hW = [0 0 0; 1 -1 0; 0 0 0];
21 hNE = [0 0 1; 0 -1 0; 0 0 0];
22 hSE = [0 0 0; 0 -1 0; 0 0 1];
23 hSW = [0 0 0; 0 -1 0; 1 0 0];
24 hNW = [1 0 0; 0 -1 0; 0 0 0];
25
26 % Difusión
27 for t = 1:num_iter
28
29     %Paso 1. Calculo de los gradientes direccionales
30
31     nablaN = imfilter(diff_im ,hN, 'conv');
32     nablaS = imfilter(diff_im ,hS, 'conv');
33     nablaW = imfilter(diff_im ,hW, 'conv');
34     nablaE = imfilter(diff_im ,hE, 'conv');
35     nablaNE = imfilter(diff_im ,hNE, 'conv');
36     nablaSE = imfilter(diff_im ,hSE, 'conv');
37     nablaSW = imfilter(diff_im ,hSW, 'conv');
38     nablaNW = imfilter(diff_im ,hNW, 'conv');
39
40     %Paso 2. Evalaución de la función de difusión en cada dirección.
41
42     cN = CD(nablaN ,kappa);
43     cS = CD(nablaS ,kappa);
44     cW = CD(nablaW ,kappa);
45     cE = CD(nablaE ,kappa);
46     cNE = CD(nablaNE ,kappa);
47     cSE = CD(nablaSE ,kappa);
48     cSW = CD(nablaSW ,kappa);
49     cNW = CD(nablaNW ,kappa);
50
51
52     %Paso 3. Transformación de la imagen
53
54     diff_im = diff_im + ...
55     1/8*(...
56     (1/(dy^2))*cN.*nablaN + (1/(dy^2))*cS.*nablaS + ...
57     (1/(dx^2))*cW.*nablaW + (1/(dx^2))*cE.*nablaE + ...
58     (1/(dd^2))*cNE.*nablaNE + (1/(dd^2))*cSE.*nablaSE + ...
59     (1/(dd^2))*cSW.*nablaSW + (1/(dd^2))*cNW.*nablaNW );
60
61     fprintf('\rIteration %d\n',t);
62 end

```

A continuación mostramos el trabajo de este filtro con diferentes tipos de ruido y con los dos coeficientes de difusión propuestos por Perona-Malik. Es importante mencionar que los límites del gradiente dados por K en el modelo, se han calculado con la técnica propuesta por ellos inicialmente. Esta consisten en calcular usar el estimador de ruido dado por Canny, el cual consiste en calcular un histograma de los valores absolutos de la imagen gradiente y estimar K como el 90 % de la suma acumulada de este histograma.

Iniciamos con el coeficiente dado por la función de Lorentz

$$c_1(x) = \frac{1}{1 + \left(\frac{x}{K}\right)^2}$$

Donde K representa el límite de difusión, el valor donde el flujo $\Phi(x)$ presenta el valor máximo de difusión.

En esta primera imagen se muestra el proceso de difusión con 10 iteraciones pero con diferentes valores de K .



a) Imagen original.



b) Imagen con ruido sal y pimienta al 5 %.



c) Transformación con $K = 40$



d) Transformación con $K = 80$.

e) Transformación con $K = 120$ f) Transformación con $K = 160$.**Figura 3.1.** Procesamiento con la función de Lorentz $c_1(x)$

Veamos ahora el proceso con el coeficiente dado por la función de Leclerc

$$c_2(x) = e^{-\frac{x}{2K^2}}$$

Donde K representa el límite de difusión, el valor donde el flujo $\Phi(x)$ presenta el valor máximo de difusión. En esta primera imagen se muestra el proceso de difusión con 10 iteraciones pero con diferentes valores de K .



a) Imagen original.



b) Imagen con ruido sal y pimienta al 5%.

c) Transformación con $K = 40$.d) Transformación con $K = 80$.

e) Transformación con $K = 120$.f) Transformación con $K = 160$.**Figura 3.2.** Procesamiento con la función de Leclerc $c_2(x)$.

Observemos ahora el procesado añadiendo a la imagen original ruido Gaussiano.



a) Imagen original.



b) Imagen con ruido Gaussiano al 5%.

c) Transformación con Lorentz $K = 40$.d) Transformación con Leclerc $K = 40$ e) Transformación con Lorentz $K = 80$.f) Transformación con Leclerc $K = 80$.**Figura 3.3.** Procesamiento de Perona-Malik con ruido Gaussiano

Observe como en este caso, se requiere un K menor que en ambas funciones para obtener mejores resultados en el procesamiento de la imagen. Esto se debe a que el ruido sal y pimienta proporciona a la imagen pixeles blancos y negros de manera aleatoria. En tales puntos el gradiente puede llegar a ser muy grande y por lo tanto se requiere un límite de difusión mayor.

Observemos por último, el resultado del procesamiento con ruido Speckle.



a) Imagen original.



b) Imagen con ruido Speckle.



c) Transformación con Lorentz $K = 40$



d) Transformación con Leclerc $K = 40$.



e) Transformación con Lorentz $K = 80$



f) Transformación con Leclerc $K = 80$.

Figura 3.4. Procesamiento de perona-Malik con ruido Speckle

3.2. Coeficientes de difusión

Los coeficientes de difusión, son funciones que controlan el flujo existente entre los píxeles de una imagen, por lo cual en la búsqueda de preservar las fronteras de la misma, buscamos coeficientes que permitan un máximo flujo para magnitudes del gradiente muy cercanas pero un flujo mínimo cuando estos valores no lo son. Teóricamente, se requieren funciones c con las siguientes características

1. Infinitamente diferenciable.
2. Función monótona decreciente de 1 a 0 cuando $|\nabla I|^2$ varía de 0 a ∞ .
3. Tomando $x = |\nabla I|$, la función de flujo, definida por $\Phi(x) = xc(x^2)$ tiene un máximo en \mathbb{R}^+ .

Iniciamos estudiando el comportamiento de un coeficiente propuesto por Tsitsios-Petrou [20] dado por la siguiente función

$$c_3(x) = \begin{cases} 0,67 \left[1 - \left(\frac{x}{\sqrt{5}K} \right)^2 \right]^2, & \text{si } x \leq K\sqrt{5} \\ 0, & \text{en otros casos.} \end{cases} \quad (3.2)$$

Esta función, cumple las siguientes condiciones

1. Infinitamente diferenciable.
2. Es monótona decreciente de 0.67 a 0 cuando x varía de 0 a $\sqrt{5}K$ y es constante en en intervalo $(\sqrt{5}K, \infty)$.
3. La función de flujo $xc_3(x)$ tiene un máximo cuando $x = K$.

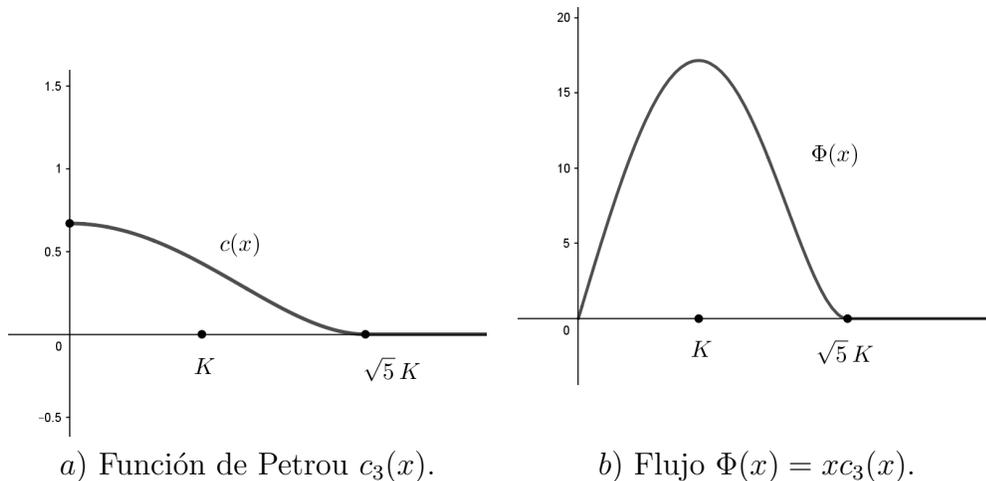


Figura 3.5. Coeficiente de Petrou c_3 y su flujo.

A diferencia de los coeficientes de Perona-Malik, la función propuesta por Petrou termina completamente la difusión cuando $x = \sqrt{5}K$, lo que indica que no se realizará algún tipo de difusión para las magnitudes del gradiente que superen este valor.

A continuación presentamos imágenes con 5% de ruido y sus correspondientes imágenes procesadas después de 10 iteraciones haciendo uso del coeficiente de Petrou.



a) Imagen con sal y pimienta.



b) Imagen procesada $K = 90\sqrt{5}$.



c) Imagen con ruido gaussiano.



d) Imagen procesada $K = 40\sqrt{5}$.



e) Imagen con ruido speckle.



f) Imagen procesada $K = 40\sqrt{5}$.

Figura 3.6. Procesamiento con coeficiente de Petrou $c_3(x)$.

Proponemos ahora un primer coeficiente, dado por la expresión

$$c_4(x) = \begin{cases} \frac{1}{K^4} \left(\frac{x}{\sqrt{K}} - K \right)^2 \left(\frac{x}{\sqrt{K}} + K^2 \right), & \text{si } x \leq K\sqrt{K} \\ 0, & \text{en otros casos.} \end{cases} \quad (3.3)$$

Esta función, cumple las siguientes condiciones

1. Infinitamente diferenciable.
2. Es monótona decreciente de 1 a 0 cuando x varía de 0 a $K\sqrt{K}$ y es constante en en intervalo $(K\sqrt{K}, \infty)$.
3. La función de flujo $xc(x)$ tiene un máximo cuando $x = 1/11(-3K^{5/2} + 8K^{3/2} - 3K^{3/2}(K + 1))$.

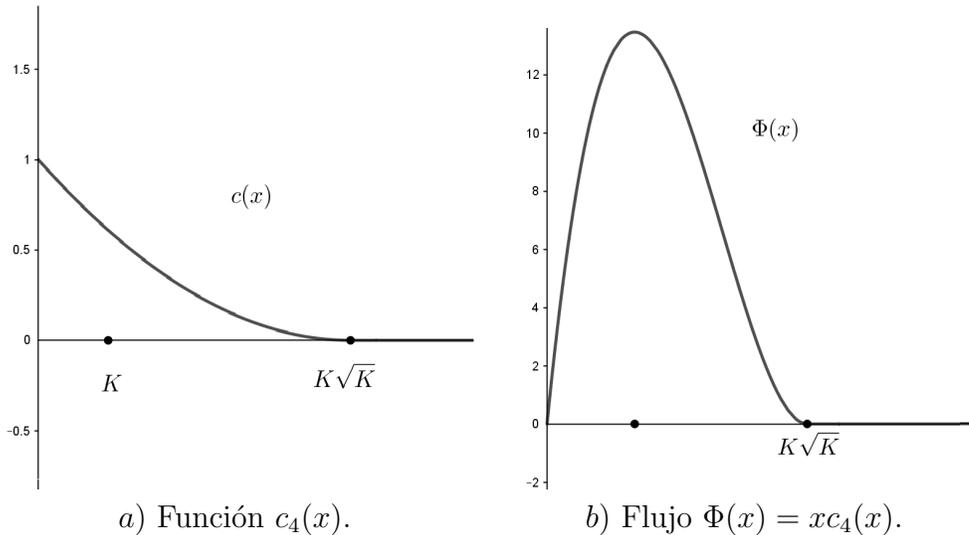


Figura 3.7. Coeficiente propuesto c_4 y su flujo.

Esta función termina completamente la difusión cuando $x = \sqrt{K}K$, lo que indica que no se realizará algún tipo de difusión para las magnitudes del gradiente que superen este valor.

A continuación presentamos imágenes con 5 % de ruido y sus correspondientes imágenes procesadas después de 10 iteraciones haciendo uso del coeficiente $c_4(x)$.



a) Imagen con sal y pimienta.

b) Imagen procesada $K = 160$.

c) Imagen con ruido gaussiano.

d) Imagen procesada $K = 100$.

e) Imagen con ruido speckle.

f) Imagen procesada $K = 80$.**Figura 3.8.** Procesamiento con coeficiente $c_4(x)$.

Hemos buscado ahora, un coeficiente basado en una función trigonométrica. Llamaremos a este el coeficiente c_5 .

$$c_5(x) = \begin{cases} 0,5 \cos\left(\frac{\pi x}{K}\right) + 0,5, & \text{si } x \leq K \\ 0, & \text{en otros casos.} \end{cases} \quad (3.4)$$

Esta función, cumple las siguientes condiciones

1. Infinitamente diferenciable.
2. Es monótona decreciente de 1 a 0 cuando x varía de 0 a K y es constante en el intervalo (K, ∞) .

3. La función de flujo $xc(x)$ tiene un máximo cuando $x \in \mathbb{R}^+$.

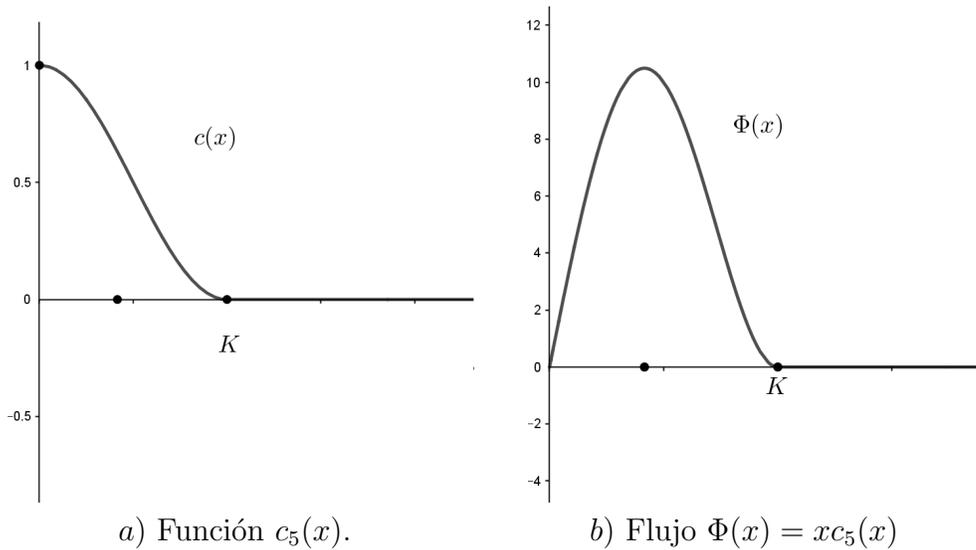


Figura 3.9. Coeficiente propuesto c_5 y su flujo.

Esta función termina completamente la difusión cuando $x = K$, lo que indica que no se realizará algún tipo de difusión para las magnitudes del gradiente que superen este valor.

A continuación presentamos imágenes con 5% de ruido y sus correspondientes imágenes procesadas después de 10 iteraciones haciendo uso del coeficiente $c_5(x)$.



a) Imagen con sal y pimienta.



b) Imagen procesada $K = 300$.



c) Imagen con ruido gaussiano.



d) Imagen procesada $K = 120$.



e) Imagen con ruido speckle.

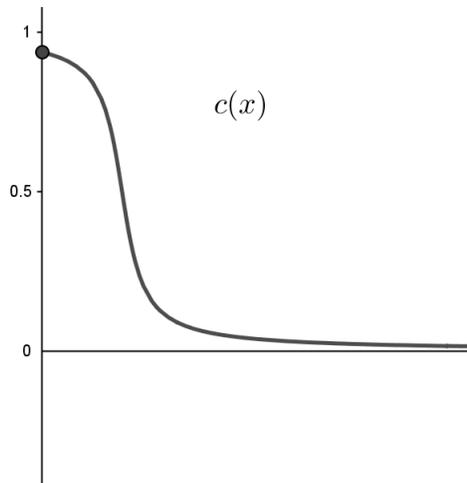
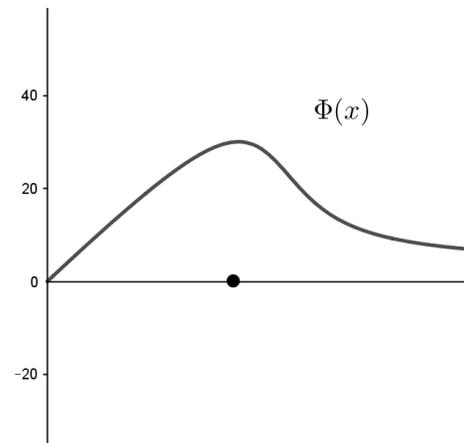
f) Imagen procesada $K = 140$.**Figura 3.10.** Procesamiento con coeficiente $c_5(x)$.

Proponemos ahora el coeficiente c_6 , que hace uso de la función inversa de tangente.

$$c_6(x) = -\frac{1}{\pi} \arctan\left(\frac{x}{2K} - K\right) + 0,5 \quad (3.5)$$

Esta función, cumple las siguientes condiciones

1. Infinitamente diferenciable.
2. Es monótona decreciente de 1 a 0 cuando x varía de 0 a ∞ .
3. La función de flujo $xc(x)$ tiene un máximo cuando $x \in \mathbb{R}^+$.

a) Función $c_6(x)$.b) Flujo $\Phi(x) = xc_6(x)$.**Figura 3.11.** Coeficiente propuesto c_6 y su flujo.

A diferencia de los coeficientes anteriores, esta función no termina completamente la difusión para algún valor de x , lo que indica que actúa bajo las mismas condiciones de las

propuestas por Perona-Malik.

A continuación presentamos imágenes con 5 % de ruido y sus correspondientes imágenes procesadas después de 10 iteraciones haciendo uso de este coeficiente de difusión.



a) Imagen con sal y pimienta.



b) Imagen procesada $K = 60$.



c) Imagen con ruido gaussiano.



d) Imagen procesada $K = 40$.



e) Imagen con ruido speckle



f) Imagen procesada $K = 20$.

Figura 3.12. Procesamiento con coeficiente $c_6(x)$.

Por último, queremos mostrar el comportamiento del procesamiento si trabajamos con un coeficiente no diferenciable, para eso, usaremos una función $c_7(x)$ definida a trozos, y cuya difusión depende de una función lineal.

$$c_7(x) = \begin{cases} -\frac{x}{K} + 1, & \text{si } x \leq K\sqrt{K} \\ 0, & \text{en otros casos.} \end{cases} \quad (3.6)$$

Esta función, cumple las siguientes condiciones

1. Es monótona decreciente de 1 a 0 cuando x varía de 0 a K y es constante en el intervalo (K, ∞) .
2. La función de flujo $xc(x)$ tiene un máximo cuando $x = K/2$.

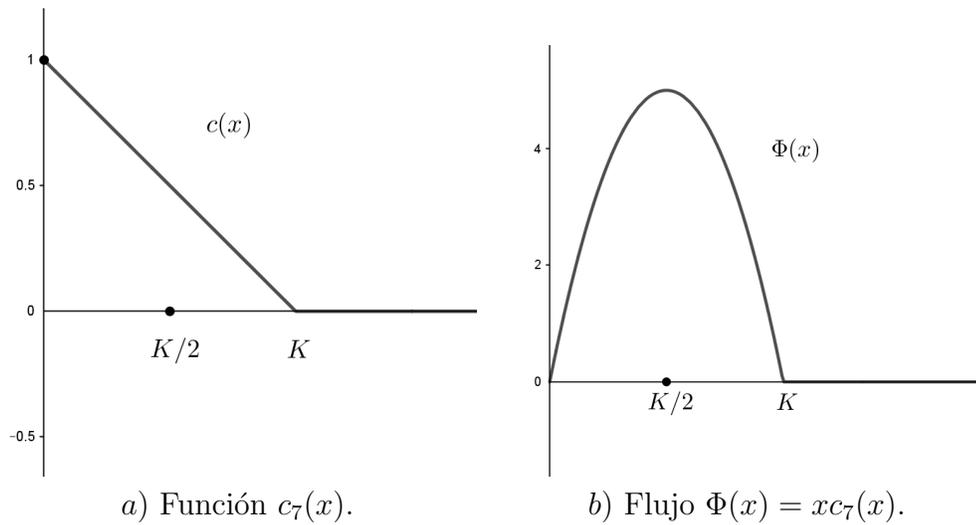
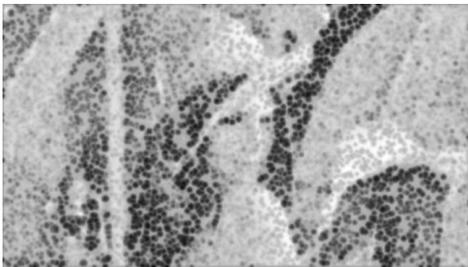


Figura 3.13. Coeficiente $c_7(x)$.

A continuación presentamos imágenes con 5% de ruido y sus correspondientes imágenes procesadas después de 10 iteraciones haciendo uso de este coeficiente de difusión.



a) Imagen con ruido sal y pimienta procesada con $k = 100$.



b) Imagen con ruido sal y pimienta procesada con $k = 200$.



c) Imagen con ruido gaussiano procesada con $k = 100$.



d) Imagen con ruido gaussiano procesada con $k = 200$.



e) Imagen con ruido speckle procesada con $k = 100$.



f) Imagen con ruido speckle procesada con $k = 200$.

Figura 3.14. Procesamiento con coeficiente $c_7(x)$.

3.3. Modelo basado en la curvatura

Como se mencionó en el capítulo 2, Álvarez [17] propone modelo de análisis multiescala basado en la curvatura, que teóricamente satisface más principios que el modelo de difusión de Perona-Malik. A continuación mostramos la discretización del problema basados en [5], la implementación del método en Matlab y sus resultados experimentales.

Para la discretización del problema tenemos que

$$I^{n+1}(i, j) = I^n(i, j) + \Delta t I_t^n(i, j)$$

Donde

$$I_t^n(i, j) = \left[\overrightarrow{dL}^n(i, j) \cdot \overrightarrow{u}^n(i, j) \right] |\nabla I^n(i, j)|$$

En esta expresión, debemos recordar que el Laplaciano en el pixel (i, j) está dado por

$$L^n(i, j) = I_{xx}^n(i, j) + I_{yy}^n(i, j)$$

con esto en mente, tenemos que

$$\vec{dL}(i, j) = [L^n(i+1, j) - L^n(i-1, j), L^n(i, j+1) - L^n(i, j-1)]$$

El vector $\vec{u}^h(i, j)$, es ortogonal al gradiente, por tanto depende de las primeras derivadas y está dado por

$$\vec{u}^h(i, j) = \frac{[-I_y^n(i, j), I_x^n(i, j)]}{|\nabla I|}$$

Para calcular el cambio de L en la dirección de $\vec{u}^h(i, j)$, debemos encontrar la coordenada en proyección de $\vec{dL}^h(i, j)$ sobre $\vec{u}^h(i, j)$,

$$\beta^n(i, j) = \vec{dL}^h(i, j) \cdot \vec{u}^h(i, j)$$

y así, la magnitud del gradiente se obtendrá por medio de la expresión

$$|\nabla I^n(i, j)| = \begin{cases} \sqrt{(I_{bxm}^n)^2 + (I_{fxM}^n)^2 + (I_{bym}^n)^2 + (I_{fxM}^n)^2}, & \text{cuando } \beta^n > 0, \\ \sqrt{(I_{bxM}^n)^2 + (I_{fxm}^n)^2 + (I_{byM}^n)^2 + (I_{fxm}^n)^2}, & \text{cuando } \beta^n < 0, \end{cases}$$

Donde bx y fx denotan las derivadas hacia atrás y hacia adelante de I , análogamente con by y fy . m y M el mínimo o el máximo, respectivamente, entre la derivada y cero.

Proponemos a continuación una implementación en Matlab del método recién descrito:

```

1
2 function diff_im = curv2D(im, num_iter, delta_t)
3 % Paso 1 general. Convertir im en una imagen de doble precisión.
4 im = double(im);
5
6 % Paso 2 general. Introducir la condicion inicial de la ecuación.
7 diff_im = im;
8
9 % Paso 3 general. Matrices de convolución y diferencias finitas.
10 L = fspecial('laplacian', 0);
11 dL1 = [0 0 -1 0 ; 0 -1 4 -1 ; 0 0 0 0 ; 0 1 -4 1 ; 0 0 1 0 ];
12 dL2 = [0 0 0 0 0; 0 -1 0 1 0; -1 1 0 -4 1; 0 -1 0 1 0];
13 bx = [0 1 ; -1 0];
14 fx = [1 0; 0 -1];
15 by = [0 1; 0 -1];
16 fy = [1 0; -1 0];
17
18
19 % Proceso de Difusión
20 for t = 1:num_iter
21
22 % Cálculo de la imagen gradiente, su magnitud y su normal
23 [Gx, Gy] = imgradientxy(im);
24 [Gmag, -] = imgradient(Gx, Gy);

```

```

25 n=Gmag;
26 u(:, :, 1)=-Gy./n;
27 u(:, :, 2)=Gx./n;
28
29 % Cálculo de L
30
31 dLx=imfilter(diff_im ,dL1, 'conv');
32 dLy=imfilter(diff_im ,dL2, 'conv');
33 v(:, :, 1)=dLx;
34 v(:, :, 2)=dLy;
35
36 % Calculo de beta
37
38 w=u.*v;
39 beta=w(:, :, 1)+w(:, :, 2);
40
41 % Derivadas de la imagen en distintas direcciones
42
43 Ibx=imfilter(diff_im ,bx, 'conv');
44 Ifx=imfilter(diff_im ,fx, 'conv');
45 Iby=imfilter(diff_im ,by, 'conv');
46 Ify=imfilter(diff_im ,fy, 'conv');
47
48 Ibxm=(Ibx-abs(Ibx))./2;
49 IfxM=(Ifx+abs(Ifx))./2;
50 Ibym=(Iby-abs(Iby))./2;
51 IfyM=(Ify+abs(Ify))./2;
52
53 IbxM=(Ibx+abs(Ibx))./2;
54 Ifxm=(Ifx-abs(Ifx))./2;
55 IbyM=(Iby+abs(Iby))./2;
56 Ifym=(Ify-abs(Ify))./2;
57
58 % Construcción de la imagen gradiente dependiente del valor de beta
59 [M,N] = size(beta);
60
61 gradI=zeros(M,N);
62 for i=1:M
63 for j=1:N
64 if beta(i, j)>0;
65 gradI(i, j)=sqrt((Ibxm(i, j)).^2+(IfxM(i, j)).^2+(Ibym(i, j)).^2+(IfyM(i, j)).^2);
66 else
67 if isnan(beta(i, j))== 1
68 gradI(i, j)=0;
69 beta(i, j)=0;
70 else gradI(i, j)=sqrt((IbxM(i, j)).^2+(Ifxm(i, j)).^2+(IbyM(i, j)).^2+(Ifym(i, j)).^2);
71 end
72 end
73 end
74 end
75
76 % Imagen iterada
77

```

```

78 I_t=beta.*gradI;
79
80 diff_im = diff_im + (delta_t.*I_t);
81
82 fprintf('\rIteration %d\n',t);
83 end

```

A continuación mostramos los resultados obtenidos con este modelo:



a) Imagen con sal y pimienta.

b) Imagen procesada.



c) Imagen con ruido gaussiano.

d) Imagen procesada.



e) Imagen con ruido speckle.

f) Imagen procesada.

Figura 3.15. Procesamiento con el modelo de la curvatura con 10 iteraciones.

Es fácil observar que este modelo preserva muy bien las fronteras de la imagen, pero el proceso de suavizado en las regiones internas no supera los resultados obtenidos mediante el proceso de Perona-Malik.

Para analizar con más formalidad los resultados obtenidos en las anteriores procesados, veremos a continuación algunas métricas existentes para evaluación de la calidad de imágenes.

3.4. Métricas de comparación

En la cotidianidad, la calidad de la imagen puede verse afectada debido a las distorsiones durante la adquisición o el procesamiento de la misma, por ejemplo ruido, borrosidad o zumbido. Para evaluar la calidad de una imagen existen dos clases de métodos: subjetivos y objetivos. En general estos métodos reciben el nombre de métricas de comparación o evaluación de calidad.

Los métodos subjetivos, requieren personas que bajo algunos criterios y escalas preestablecidas teniendo en cuenta el sistema visual humano, compartan su opinión sobre un material predeterminado. Aunque estas técnicas son ideales para establecer la calidad ante la vista humana, presentan inconvenientes pues las evaluaciones están influenciadas por las condiciones físicas y emocionales de los observadores.

Los métodos objetivos intentan establecer la calidad de una imagen que ha sido procesada por un canal de transmisión, un algoritmo de filtrado o de compresión. En este caso, se hace uso de modelos matemáticos que están basados en medidas estadísticas del error, de coeficientes de correlación o en algunos casos que están basadas en propiedades de la percepción visual como la sensibilidad del ojo a las frecuencias espaciales, a la luz, a la detección de ruido o la presencia de señales introducidas en el proceso. Para ver más información sobre el sistema de percepción visual ver [15].

En esta sección estamos interesados en evaluar la calidad de las imágenes transformadas con el modelo de Perona-Malik y con los cambios propuestos anteriormente. Es por esto, que mencionamos brevemente las características, modelos matemáticos e implementación en Matlab de las métricas de comparación que usaremos. Si se desea conocer más información sobre las técnicas de evaluación en calidad de imágenes, ver [21].

3.4.1. Error cuadrático medio (MSE)

Esta métrica de comparación no se encuentra basada en el sistema de visión humano, sino que hace uso del error cuadrático medio entre una imagen de referencia I_r y una imagen procesada I_p , mediante la siguiente ecuación:

$$\text{MSE} = \frac{1}{mn} \sum_{(i,j)}^{(m,n)} [I_r(i, j) - I_p(i, j)]^2$$

Donde $m \times n$ denota el tamaño de I_r e I_p .

Este es uno de los métodos más usados en las aplicaciones de procesamiento de imágenes tales como eliminación de ruido, reconstrucción, clasificación, restauración y diseño de filtros, gracias a su simplicidad en el cálculo o su independencia del parámetro, sin embargo tiene

grandes desventajas pues al no estar correlacionado con el sistema visual no establece si las distorsiones afectan fuertemente la calidad de la imagen o si son claramente visibles a ojo humano, así si el valor de MSE es cero o pequeño, entendemos nula o mínima distorsión, pero en general dos imágenes con el mismo MSE pueden tener calidades visuales totalmente diferentes.

En Matlab, puede calcularse por medio de la función `immse(X,Y)` donde X y Y son las imágenes del mismo tamaño y de la misma clase.

3.4.2. Relación pico señal / ruido (pSNR)

Esta técnica se deriva del error cuadrático medio e indica la relación (en dB) entre una imagen de referencia y una imagen procesada. Está definido como

$$\text{pSNR} = 10 \log_{10} \left(\frac{(\text{Máx } I)^2}{\text{MSE}} \right)$$

Donde Máx I representa el valor máximo de que puede tomar el pixel (por ejemplo 255 para imágenes de 8 bits).

En la interpretación del índice pSNR, entendemos que entre más alto es este valor, la imagen procesada es más cercana a la imagen original y por tanto podría suponerse una mejor calidad, sin embargo esto no está relacionado con el sistema visual humano por lo que posee las mismas desventajas de la métrica MSE. En la literatura es una de las técnicas más usadas, en situaciones con muy poco ruido, por su simplicidad e independencia de condiciones de medición.

En Matlab, puede calcularse por medio de la función `psnr(X, Y)` donde X es la imagen transformada y Y es la imagen original de referencia, estas deben ser del mismo tamaño y de la misma clase que se desean comparar.

3.4.3. Índice de similitud estructural (SSIM)

Esta métrica tiene en cuenta algunas propiedades del sistema visual humano como la luminosidad, el contraste y la estructura de la información en una imagen. Para el cálculo de este índice es necesario tener en cuenta las intensidades medias de la imagen original I_r y de de la imagen procesada I_p , dados por μ_r y μ_p , las desviaciones estándar σ_r y σ_p y la correlación cruzada de las imágenes σ_{rp} .

En este caso la similaridad de las imágenes con respecto a la luminosidad $l(I_r, I_p)$, la similaridad en el contraste $c(I_r, I_p)$ y la similaridad en la estructura local $s(I_r, I_p)$ están definidas como

$$l(I_r, I_p) = \frac{2\mu_r\mu_p + C_1}{\mu_r^2 + \mu_p^2 + C_1}, \quad c(I_r, I_p) = \frac{2\sigma_r\sigma_p + C_2}{\sigma_r^2 + \sigma_p^2 + C_2}$$

$$s(I_r, I_p) = \frac{\sigma_{rp} + C_3}{\sigma_r^2 \sigma_p^2 + C_3},$$

donde C_1 , C_2 y C_3 son constantes que positivas que estabilizan estos términos para evitar problemas de inestabilidad numérica.

Finalmente, el índice de similaridad estructural está dado por

$$\text{SSIM} = l(I_r, I_p)c(I_r, I_p)s(I_r, I_p)$$

Para la evaluación de este resultado, se debe tener en cuenta que este índice está acotado por 1, donde la igualdad sólo es alcanza cuando las imágenes son iguales. Dentro de las ventajas de este índice se encuentran su cercanía con el sistema visual y la posibilidad de obtener un mapa puntual de la calidad de la imagen, mostrando explícitamente las regiones en las que el proceso de filtrado es más efectivo. Como desventaja se encuentra la complejidad de su cálculo.

En Matlab, puede calcularse por medio de la función `ssim(X, Y)` donde X es la imagen transformada y Y es la imagen original de referencia, estas deben ser del mismo tamaño y de la misma clase. Si se desea obtener el mapa puntual de calidad dado por este índice la función `[ssimval, ssimmap] = ssim(X,Y)` calcula el valor global del SSIM (`ssimval`) y una imagen con los valores del índice en cada pixel (`ssimmap`).

3.4.4. Evaluación de calidad

A continuación presentamos los resultados de comparación de estos tres métodos con los coeficientes de difusión presentados anteriormente y con el método de la curvatura. Iniciamos con los resultados obtenidos con la métrica **MSE**.

Tabla 3.1. Comparaciones con la métrica **MSE**.

$C(x)$ /Ruido	Sal y Pimienta	Gaussiano	Speckle
c_1	169,4146	278,3174	130,5931
c_2	175,2633	286,8644	110,3393
c_3	151,3243	258,2342	116,5925
c_4	198,8442	276,2827	240,4817
c_5	179,2604	255,6230	110,6923
c_6	188,4951	319,6169	171,7044
c_7	362,4504	253,8286	277,5383
c_8	948,9420	808,4814	550,0787

Según esta métrica, el mejor resultado depende del tipo de ruido que tenga la imagen. Para el ruido sal y pimienta, se encuentra un mejor resultado con el procesamiento realizado con el coeficiente de difusión c_3 , seguido por la función de Lorentz c_1 . Para la imagen con ruido gaussiano, los mejores resultados se encuentran con c_7 y c_5 , mientras que para el ruido Speckle la mejor imagen procesada está lograda con el coeficiente de Leclerc c_2 seguida del coeficiente c_5 .

Veamos ahora los resultados obtenidos con la métrica **PSNR**.

Tabla 3.2. Comparaciones con la métrica **PSNR**.

$C(x)$ /Ruido	Sal y Pimienta	Gaussiano	Speckle
c_1	-22,2895	-24,4454	-21,1592
c_2	-22,4369	-24,1973	-20,4273
c_3	-21,7991	-24,1201	-20,6667
c_4	-22,9851	-24,4123	-23,8108
c_5	-22,5348	-24,0760	-20,4412
c_6	-22,7530	-25,0474	-22,3478
c_7	-25,5925	-24,0454	-24,4332
c_8	-29,7724	-29,0767	-27,4042

Nuevamente tenemos que resultado depende del tipo de ruido que tenga la imagen. Para el ruido sal y pimienta, se encuentra un mejor resultado con c_3 , seguido por la función de Lorentz c_1 . Para la imagen con ruido Gaussiano, los mejores resultados se encuentran con c_7 y c_5 , mientras que para el ruido Speckle la mejor imagen procesada está lograda con el coeficiente de Leclerc c_2 seguida del coeficiente c_2 .

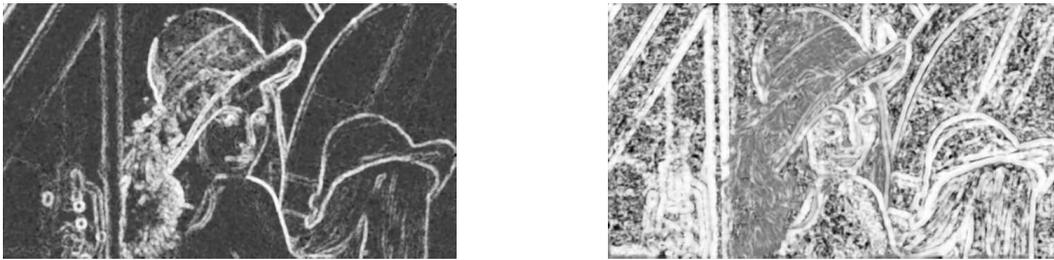
Por último, analicemos los resultados con el índice de similitud estructural **SSIM**.

Tabla 3.3. Comparaciones con la métrica **SSIM**.

$C(x)$ /Ruido	Sal y Pimienta	Gaussiano	Speckle
c_1	0,3594	0,3922	0,4077
c_2	0,3587	0,4006	0,4155
c_3	0,3816	0,4046	0,4261
c_4	0,3867	0,3812	0,3797
c_5	0,3600	0,4076	0,4158
c_6	0,3653	0,3823	0,3991
c_7	0,3139	0,3771	0,3758
c_8	0,2231	0,1708	0,2324

Este índice muestra que para el ruido sal y pimienta, se encuentra un mejor resultado con c_4 , seguido por la función c_3 . Para la imagen con ruido Gaussiano, los mejores resultados se encuentran con c_5 y c_3 , mientras que para el ruido Speckle la mejor imagen procesada está lograda con el coeficiente c_3 seguida por el coeficiente c_5 .

Es importante observar como las tres métricas muestran que el modelo basado en curvatura tiene los peores resultados en cuanto a suavizado de la imagen, sin embargo, los mapas de calidad obtenidos con el índice **SSIM**, nos permiten evidenciar que el trabajo de este filtro es muy apropiado en la conservación de bordes.



a) Imagen con ruido gaussiano filtrada con c_8 . b) Imagen con ruido gaussiano filtrada con c_5 .

Figura 3.16. Mapas de calidad de la imagen.

3.5. Combinación de filtros

Finalmente, estudiamos la idea de combinar en cada iteración dos filtros de difusión. Iniciamos usando imágenes con ruido sal y pimienta. Dado que los mejores coeficientes de difusión están dados por las funciones c_4 , c_3 y c_1 , indagaremos experimentalmente los resultados de combinar estos coeficientes con el filtro de mediana, con un filtro gaussiano de desviación estandar igual al ruido impuesto y por último con el método basado en la curvatura.



a) Imagen con ruido sal y pimienta. b) Imagen con filtro de mediana. c) Imagen procesada con mediana y c_4 .

Figura 3.17. Comparación filtro de mediana y su combinación con c_4 .

Los índices **SSIM** obtenidos en este caso son

Tabla 3.4. Comparaciones con la métrica **SSIM**.

Filtro	SSIM
Mediana	0.8061
Mediana + c_4	0.7248
Mediana + c_3	0.7695
Mediana + c_1	0.7543

Por lo que podemos concluir que el filtro de mediana es suficiente para la eliminación del ruido sal y pimienta, añadir un proceso de filtrado más presenta peores resultados.

Veamos ahora los resultados obtenidos combinando el coeficiente c_4 con un filtro Gaussiano en cada iteración y con el modelo basado en la curvatura.



a) Imagen con ruido sal y pimienta.



b) Imagen con filtro c_4 .



c) Imagen procesada con filtro gaussiano y c_4 .



d) Imagen procesada con c_4 y c_8 .

Figura 3.18. Comparación filtros c_4 , c_8 y gaussiano.

Tabla 3.5. Comparaciones con la métrica **SSIM**.

Filtro	Gaussiano	c_4	c_8
Sin combinar	0.2211	0.3822	0.2231
c_4	0.3667	-	0.5067

Observe como en este caso, la combinación de los filtros de difusión anisotrópica con coeficiente c_4 y el modelo basado en la curvatura producen un mejor resultado que cualquiera de los obtenidos anteriormente (salvo por el filtro de mediana). La función c_4 proporciona una mejor difusión en las regiones internas, mientras que c_8 preserva los bordes de la imagen.

Indagamos ahora los resultados que se obtienen de la combinación de filtros c_3 , c_5 y c_7 con los filtros de mediana, Gaussiano y el modelo basado en la curvatura, aplicados sobre una imagen con ruido Gaussiano.

Tabla 3.6. Comparaciones con la métrica **SSIM**.

Filtro	Sin combinar	Mediana	Gaussiano	c_8
c_3	0.4046	0.3789	0.4046	0.5517
c_4	0.4046	0.3614	0.3886	0.3800
c_7	0.3771	0.3523		0.3851



a) Imagen con ruido sal y pimienta.

b) Imagen procesada con c_3 y c_8 .

Figura 3.19. Imágen procesada con combinación c_3 y c_8 .

Para terminar, hacemos uso de imágenes con ruido Speckle y combinamos los filtros dados por los coeficientes de difusión c_2 , c_3 y c_5 con un filtro de mediana, un Gaussiano y el de curvatura.

Tabla 3.7. Comparaciones con la métrica **SSIM**.

Filtro	Sin combinar	Mediana	Gaussiano	c_8
c_2	0.4155	0.3835	0.4285	0.3798
c_3	0.4261	0.3705	0.4340	0.3713
c_5	0.4158	0.3644	0.3054	0.3648

El índice SSIM nos indica, que en este caso cualquier combinación de filtros daña el proceso logrado por los coeficientes de difusión logrados.

3.6. Análisis ante diferentes niveles de ruido

Dado que nuestro mejor resultado ha sido obtenido con una combinación de filtros, a continuación enfrentaremos nuestro modelo a diferentes niveles de ruido y diferentes imágenes de prueba.

Iniciamos con imágenes impregnadas de ruido sal y pimienta al 20 %, 40 % y 60 %. Comparamos visualmente con el modelo de Perona-Malik haciendo uso del coeficiente de c_4 (por obtener el índice SSIM más alto con este tipo de ruido) y nuestra combinación del coeficiente c_4 con el modelo de curvatura. Usaremos inicialmente la imagen del camarógrafo tomada de [23], la cual es bastante usada en libros y artículos de procesamiento de imágenes.



Figura 3.20. Imagen de prueba original: El camarógrafo



Imagen con ruido del 20 %.



Imagen procesada con c_4 .



Imagen procesada con c_4 y c_8 .



Imagen con ruido del 40 %.



Imagen procesada con c_4 .



Imagen procesada con c_4 y c_8 .



Imagen con ruido del 60 %.



Imagen procesada con c_4 .



Imagen procesada con c_4 y c_8 .

Figura 3.21. Procesamiento del camarografo.

Tabla 3.8. Comparaciones con la métrica **SSIM** de procesado con c_4 y combinación de filtros.

Ruido	c_4	$c_4 + c_8$
20 %	0.1023	0.4875
40 %	0.1314	0.3414
60 %	0.0746	0.1853

Veamos ahora los resultados tomando como imagen original la fotografía llamada Europa, que ha sido tomada por autor.

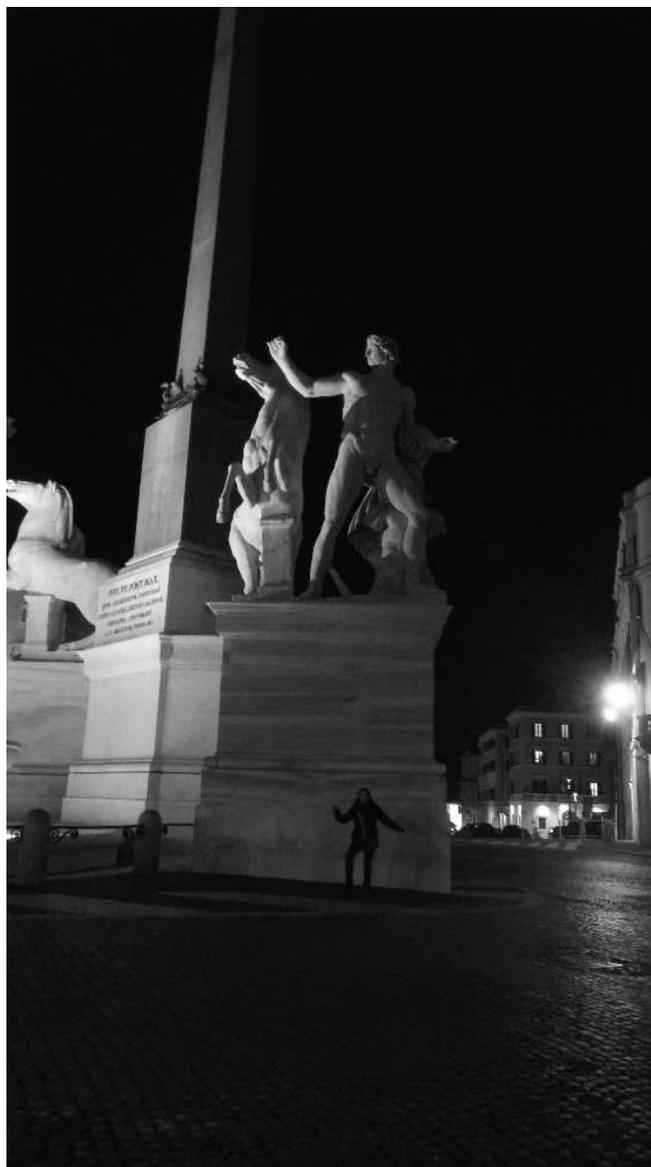


Figura 3.22. Imagen de prueba original: Europa



a) Imagen con ruido al 20 %.



b) Imagen procesada con c_4 .



c) Imagen procesada con c_4 y c_8 .



d) Imagen con ruido al 40 %.



e) Imagen procesada con c_4 .



f) Imagen procesada con c_4 y c_8 .



g) Imagen con ruido al 60 %.



h) Imagen procesada con c_4 .



1) Imagen procesada con c_4 y c_8 .

Figura 3.23. Procesamiento de Europa.

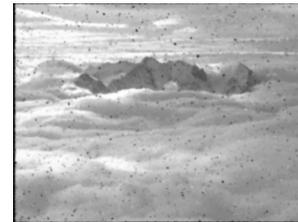
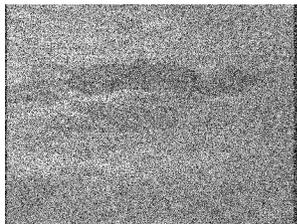
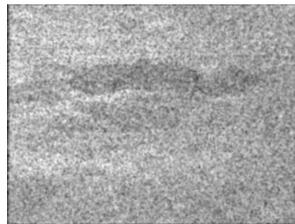
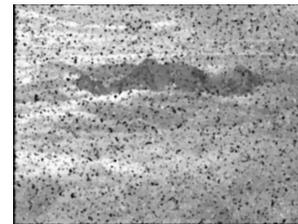
Nuevamente, los resultados obtenidos con la combinación de filtros superan por un 23 % las transformaciones realizadas con el coeficiente c_4 . Es importante observar, que entre más alto sea el porcentaje de ruido aplicado a la imagen original, ambos métodos disminuyen su nivel de eficiencia.

Tabla 3.9. Comparaciones con la métrica **SSIM** de procesado con c_4 y combinación de filtros.

Ruido	c_4	$c_4 + c_8$
20 %	0.0983	0.4034
40 %	0.1133	0.3570
60 %	0.0501	0.1928

Por último, veamos los resultados en una imagen original con altos índices de luminosidad muy usada en la literatura de procesamiento y tomada de [23]. En este caso realizaremos la comparación con el modelo de Perona-Malik haciendo uso del coeficiente de Lorentz c_1 e igual que en los ejemplos anteriores con porcentajes de ruido del 20 %, 40 % y 60 %.



Figura 3.24. Imagen de prueba original: Cielo*a)* Imagen con ruido al 20 %.*b)* Imagen procesada con c_1 .*c)* Imagen procesada con c_4 y c_8 .*d)* Imagen con ruido al 40 %.*e)* Imagen procesada con c_1 .*f)* Imagen procesada con c_4 y c_8 .*g)* Imagen con ruido al 60 %.*h)* Imagen procesada con c_1 .*i)* Imagen procesada con c_4 y c_8 .**Figura 3.25.** Procesamiento de Cielo.

En este caso, los resultados obtenidos con la combinación de filtros superan por un 29 % las transformaciones realizadas con el coeficiente c_1 . Es importante observar, que entre más alto sea el porcentaje de ruido aplicado a la imagen original, ambos métodos disminuyen su nivel de eficiencia, sin embargo con un porcentaje de ruido del 20 % en este caso el filtro propuesto coincide con la imagen original en un 46 % más, lo cual es un resultado importante.

Tabla 3.10. Comparaciones con la métrica **SSIM** de procesado con c_1 y combinación de filtros.

Ruido	c_1	$c_4 + c_8$
20 %	0.2232	0.6853
40 %	0.1532	0.4968
60 %	0.0920	0.1639

Conclusiones

El modelo de difusión anisotrópica presentado por Perona-Malik, es uno de los modelos más importantes en el procesamiento de imágenes por su capacidad de suavizar imágenes preservando los bordes de las mismas. El estudio de sus propiedades matemáticas nos ha llevado a formalizar la necesidad de usar ecuaciones diferenciales parciales de segundo orden para satisfacer condiciones y necesidades que nacen desde otras ramas del conocimiento como la física y la teoría de la percepción.

Con el desarrollo de este trabajo hemos mostrado la importancia del coeficiente de difusión, pues de ellos depende la calidad de la imagen obtenida, sin embargo también hemos evidenciado la importancia de indagar por funciones distintas a las propuestas inicialmente por Perona-Malik, que puedan proporcionarnos mejores resultados experimentales. Además, la combinación de métodos nos ha mostrado resultados más efectivos que los dados por el modelo original.

En el desarrollo teórico nos encontramos con el modelo dado por Álvarez [17], que aunque inicialmente estaba basado en el modelo de Perona-Malik, nos mostró la importancia de incluir en los filtros de suavizado la curvatura de la imagen. Aunque el modelo inicialmente no presenta las mejores propiedades en el suavizado de regiones internas, si muestra un gran potencial en la preservación de fronteras. Por lo cual, hemos hecho uso del mismo para proponer la combinación de los dos modelos y lograr resultados visiblemente mejorados que están apoyados en las métricas de calidad de imagen.

El camino a recorrer en este sentido es aún bastante prometedor, pues las técnicas de procesamiento se han venido desarrollado por medio de muchas otras herramientas, por ejemplo, estadísticas o de morfología, que no han sido tratadas en este trabajo, pero que pueden ser estudiadas en la búsqueda de nuevos algoritmos que proporcionen mejores resultados.

Bibliografía

- [1] Witkin, A. (1983). *Scale-space filtering* . Int'l Joint Conf. Artificial Intelligence, 1019-1021.
- [2] Koenderink, J. (1984). *The structure of images*. Biological Cybernetics, 50, 363-370.
- [3] Hummel, R. A. (1986). *Representations based on zero-crossings in scale-space*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, New York, 204-209.
- [4] Perona, P. & Malik, J. (1990). *Scale-space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell. 12, 629-639.
- [5] Sapiro, G. (2001). *Geometric Partial Differential Equations and Image Analysis*, Cambridge University Press, New York.
- [6] Strzelecki, P. (2006). *Una breve introducción a las ecuaciones en derivadas parciales*, Universidad de Varsovia, Polonia.
- [7] Ghisi, M. & Gobbino, M. (2011). *An example of global classical solution for the Perona-Malik equation*, Interfaces Free Bound, 13, 105- 125.
- [8] Hollig, K. (1983). *Existence of Infinitely Many Solutions for a Forward Backward Heat Equation*, Transactions of the American Mathematical Society, 278, 1, 299-316.
- [9] Kitchenassamy.S. (1997). *The Perona-Malik Paradox*, SIAM Journal on Applied Mathematics, 57, 5, 1328-1342.
- [10] Kawohl,B. & Kutev,N. (1998). *Maximum and comparison principles for onedimensional anisotropic diffusion*, Mathematische Annalen, 311, 107-123.
- [11] Rosatia,M. & Schiaffinob, A. (2006) *Some remarks about Perona-Malik equation*, Non-linear Analysis, 65, 1-11.
- [12] Ghisi, M. & Gobbino, M. (2009) *A class of local classical solutions for the onedimensional Perona-Malik equation*, Journal: Transaction of the American Mathematical Society 361, 6429-6446.

-
- [13] Catté, F., Lions, P., Morel, J., Coll, T. (1992). *Image Selective Smoothing and Edge Detection by Nonlinear Diffusion*, SIAM Journal on Numerical Analysis, 29, 1, 182-193.
- [14] Barenblatt, G. I., Bertsch, M., Dal Passo, R., Ughi, M. (1993). *A degenerate pseudoparabolic regularization of a nonlinear forward-backward heat equation arising in the theory of heat and mass exchange in stably stratified turbulent shear flow*, SIAM Journal on Mathematical Analysis, 24, 1414-1439.
- [15] Gonzalez, R., & Woods, R. (2002). *Digital Image Processing*, Prentice Hall, New York.
- [16] Gonzalez, R., Woods, R. & Eddins, S. (2004). *Digital Image Processing Using MATLAB*, Pearson Prentice Hall, Upper Saddle River.
- [17] Álvarez, L., Guichard, F., Lions, P.L., Morel, J.M. (1993). *Axioms and Fundamental Equations of Image Processing*. Arch. Ration. Mech. Anal. 123, 199-257.
- [18] Álvarez, L., & Morel, J.M. (1994). *Morphological Approach to Multiscale Analysis. From principles to equations* Kluwer Academic Publishers. 6, 3, 10, 978-1007.
- [19] Osher, S., Rudin, L., (1990). *Feature oriented image enhancement using shock filter*. SIAM Journal on Numerical Analysis, 27, 919-940.
- [20] Petrou, M., & Tsitsios, C. (2012). *On the choice of the parameters for anisotropic diffusion in image processing*. Pattern Recognition. 46, 5, 1-9.
- [21] Opozda, S., & Sochan, A. (2014). *The survey of subjective and objective methods for quality assessment of 2D and 3D images*. Theoretical and Applied Informatics. 26, 1.2, 39-67.
- [22] Lopes, D. (2007). *Anisotropic Diffusion (Perona & Malik)*. <https://la.mathworks.com/matlabcentral/fileexchange/14995-anisotropic-diffusion-perona-malik>.
- [23] Gonzalez, R., & Woods, R. (2018). *ImageProcessingPlace.com*. https://www.imageprocessingplace.com/root_files_V3/image_databases.htm