

ARTICLE TYPE

Tarea 3: Filtros de Difusión Anisotrópica

Felipe Cubillos Arredondo

Procesamiento de Señales e Imágenes (1/2025)

Abstract

En el procesamiento de imágenes digitales, se busca reducir el ruido de una imagen. El paper de Perona-Malik propone un modelo de difusión basado en la ecuación de calor, en el presente informe se ahondará en su funcionamiento y su implementación en Python.

1. Introducción y Solución Propuesta

1.1 El Problema

Una de las metas principales del procesamiento de imágenes es la reducción del ruido en imágenes cómo podría ser el ruido sal y pimienta, este aparece en múltiples campos cómo lo son la medicina, diseño gráfico y reconocimiento biométrico. Uno de los filtros más utilizados es el **suavizado Gaussiano**, basado en la ya conocida distribución normal que sigue el siguiente comportamiento para imágenes en 2 dimensiones.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

El problema principal de este es la **pérdida de nitidez en la imagen**, perdiendo posibles detalles claves.

1.2 Solución

Cómo se menciona inicialmente, se verá la solución propuesta por Pietro Perona y Jitendra Malik en su paper *Scale-Space and Edge Detection Using Anisotropic Diffusion*, donde se emplea un enfoque basado en las imágenes cómo un **sistema anisotrópico**, osea cómo un material cuyas propiedades son no constantes para alguna dirección. Cómo tal, el modelo se basa en la siguiente ecuación

$$I_t = \text{div}(c(x, y, t)\nabla I) = \nabla c \cdot \nabla I + c(x, y, t)\Delta I$$

- I es el brillo de la imagen en escala de grises,
- ∇ es la derivada de primer orden de un vector,
- Δ es laderivada de segundo orden de un vector.
- div es el operador vectorial de la divergencia .
- $c(x, y, t) = g(\|\nabla I(x, y, t)\|)$ es el coeficiente de difusión, **este coeficiente es modelado por 3 funciones distintas:**

- Tipo 1: $g(|x|) = e^{\frac{-|x|^2}{K^2}}$
- Tipo 2: $g(|x|) = \frac{1}{1 + \frac{|x|^2}{K^2}}$
- Tipo 3: $g(|x|) = \frac{1}{\sqrt{1 + \frac{|x|^2}{K^2}}}$

En simples términos, **se realiza una difusión selectiva de manera iterativa**, determinado por nuestra función g el cual se ajusta localmente en cada punto de la imagen basándose en la magnitud del gradiente $\|\nabla I\|$, que mide qué tan brusco es el cambio de intensidad. Donde no hay bordes, el gradiente de la imagen es pequeño. En estos puntos, el algoritmo asigna un

coeficiente de difusión alto. Esto implica un suavizado intenso, eliminando el ruido en esa región específica. Donde hay un cambio abrupto de intensidad, el gradiente es grande. Aquí, el algoritmo asigna un coeficiente de difusión muy bajo. Esto detiene o reduce drásticamente el proceso de suavizado, lo que permite que los bordes se mantengan nítidos e incluso se realcen.

2. Experimentos Realizados y Resultados Logrados

2.1 Implementación

La implementación se hizo utilizando el lenguaje Python en un entorno de Jupyter Notebook para mejor visualización. Siguiendo las indicaciones del paper se implementan las 3 funciones de difusión y principalmente la función de difusión `diffusion_anisotropica`, los resultados son mostrados por medio de la función `mostrar_imagen`.

2.2 Resultados

El para cada una de las 4 imágenes propuestas se muestran con 4 recuadros distintos, una para la imagen original y las otras 3 para cada una de las funciones de difusión, se hizo el experimento con $iter = 40$, $K = 0.09$ y $\Delta t = 0.15$.

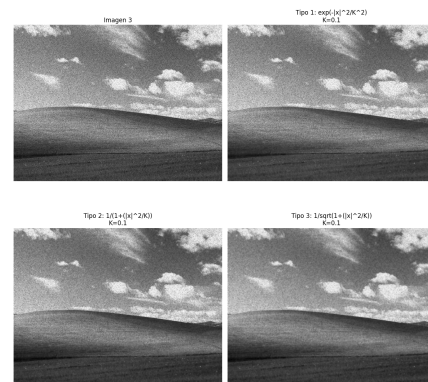


Figure 1. Resultado obtenido con imagen en escala de grises

3. Conclusiones

En conclusión, para imágenes que presentan ruido, especialmente de tipo Gaussiano o Sal-Pimienta, se nota una mejora significativa entre las imágenes de prueba para cada una de las 3 funciones de difusión, viendo además la difuminación de la imagen.