Laboratorio de Métodos Numéricos - Segundo cuatrimestre 2013
Eliminando ruido en imágenes - Lunes 7 de Octubre

Introducción

Un problema típico que se encuentra al trabajar con imágenes digitales es la existencia de "ruido" en las mismas. En pocas palabras, podemos decir que el ruido ocurre cuando el valor de uno o más píxeles de la imagen, no se corresponden con la realidad. La mayoría de las veces, esto se debe a la calidad del equipo electrónico utilizado para tomar las fotografías, o bien a posibles perturbaciones introducidas al momento de transmitir la información. Un caso muy común de imágenes con ruido son las fotografías satelitales.

Una forma de corregir (o reducir) este fenómeno en las imágenes es mediante la aplicación de filtros, con el objetivo de suavizar las mismas para obtener resultados más cercanos a la realidad. Hoy en día, existen muchas técnicas de filtrado de imágenes, muchas de ellas están basadas en modelos matemáticos que en general se resuelven mediante métodos numéricos.

Se puede pensar el problema de filtrar una imagen con ruido como la minimización del siguiente funcional:

$$\Pi = \int_\Omega \frac{\lambda}{2} |u - \tilde{u}|^2 + \frac{1}{2} \|\nabla u\|^2 d\Omega, \tag{1}$$

donde $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ describe la imagen filtrada y $\tilde{u} : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ la imagen a filtrar (con ruido). De esta manera, el primer término *pesa* cuanto ruido tiene $\tilde{u}$ y el segundo *pesa* la suavidad de la imagen obtenida. La constante λ controla la importancia relativa de los dos términos.

La minimización del funcional de la ecuación (1) da lugar a la siguiente ecuación diferencial:

$$\lambda (u - \tilde{u}) - \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = 0. \tag{2}$$

La solución de la ecuación (2) que representa la imagen filtrada se puede aproximar de manera discreta utilizando el método de diferencias finitas, lo cual conduce al siguiente sistema de ecuaciones:

$$\lambda u_{i,j} - (u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}) = \lambda \tilde{u_{i,j}} \tag{3}$$

donde ahora $u, \tilde{u} : \Omega \subset \mathbb{Z}^2 \to [0 \dots 255]$ son la versiones discretas de la imagen filtrada y la imagen original, respectivamente. Viendo la imagen u como una matriz, i, j son los índices de fila y columna de cada elemento (píxel) de la matriz, donde el 0 es representado por el color negro y el 255 por el blanco[1].

Mediciones

Una forma de medir la calidad visual de las imágenes filtradas, es a través del PSNR (*Peak Signal-to-Noise Ratio*). EL PSNR es una métrica "perceptual" (acorde a lo que perciben los humanos) y nos da una forma de medir la calidad de una imagen perturbada, siempre y cuando se cuente con la imagen original. Cuanto mayor es el PSNR mayor es la calidad de la imagen. La unidad de medida es el decibel (db) y se considera que una diferencia de 0.5 db ya es notada por la vista humana. El PSNR se define como:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_u^2}{ECM} \right)$$

donde MAX_u define el rango máximo de la imagen (para nuestro caso sería 255) y ECM es el *error cuadrático medio*, definido como:

$$\frac{1}{N} \sum_{i,j} (u^0_{i,j} - u_{i,j})^2$$

[1] Este modelo de filtrado de imágenes se puede extender a imágenes color RGB, repitiendo el proceso descripto para cada componente de color.

donde N es la cantidad de píxeles de la imagen, u^0 es la imagen original y u es la imagen perturbada (o en nuestro caso, la imagen recuperada).

Enunciado

El objetivo principal de este taller es implementar un programa para eliminar (o reducir) el ruido en imágenes digitales. Para ello, el programa deberá tomar como entrada una imagen (supuestamente con ruido) y resolver la ecuación (2) por el método de diferencias finitas (resolviendo el sistema de ecuaciones dado por las ecuaciones (3)). Finalmente, el programa deberá devolver la versión filtrada de la imágen.

Se debe:

- Enunciar dos propiedades para corroborar si el sistema a resolver puede ser resuelto mediante Jacobi o Gauss-Seidel.
- Rellenar *jacobi.m* para que dicha función tome una matriz A, dos vectores b y x_0 y resuelva el sistema $Ax = b$ mediante el método de Jacobi, utilizando a x_0 como vector inicial. Las iteraciones se deben resolver no matricialmente, sino despejando las sucesivas soluciones obtenidas.
- Rellenar *gausssei.m* para que dicha función tome una matriz A, dos vectores b y x_0 y resuelva el sistema $Ax = b$ mediante el método de Gauss-Seidel, utilizando a x_0 como vector inicial. Las iteraciones deben ser matriciales y no se debe despejar las sucesivas soluciones obtenidas.
- Rellenar *householder.m* para que dicha función tome una matriz A, un vector b y resuelva el sistema $Ax = b$ mediante la factorización QR , calculada con transformaciones de Householder. La complejidad temporal del método debe pertenecer a $\mathcal{O}(n^3)$.
- Rellenar *varfilt.m* (llamado desde *test.m*) para filtrar la imagen, resolviendo el sistema pertinente mediante los 3 métodos anteriormente mencionados. Incluir código para chequear la similitud de las diferentes soluciones.
- Analizar los PSNR obtenidos para discutir los resultados.

Evaluación:

- Coloquio con los docentes durante la clase
- En caso de no asistir a clase, se debe entregar la resolución del taller por escrito hasta el Lunes 21 de Octubre