Métodos Numéricos
Segundo cuatrimestre 2016
Taller 1
Image denoising:
Eliminando ruido en imágenes

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

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* cuánto 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 $u, \tilde{u} : \Omega \subset \mathbb{Z}^2 \to [0 \ldots 255]$ son las 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 y donde el valor 0 representa al color negro y el 255 al 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:

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

---

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

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

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

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 imagen.

Se debe:

1. Repasar la definición de factorización LU y matriz Simétrica Definida Positiva (SDP). ¿Bajo qué condiciones podemos garantizar que una matriz tiene factorización LU?
2. ¿Es cierto que si una matriz es inversible entonces tiene factorización LU?. Y si tengo una matriz que tiene factorización LU, ¿entonces es no singular? Demostrar o dar un contraejemplo.
3. Sean $A, B \in \mathbb{R}^{n\times n}$. Demostrar que A es definida positiva y B es no singular si y sólo si BAB^t es definida positiva.
4. Rellenar el archivo CheckCondLU.m para verificar que una matriz tiene factorización LU, usando para esto el ejercicio 7 de la práctica 2. Evaluar la rutina, por ejemplo de la siguiente forma:

   ```
   >> A = rand(5,5)
   >> CheckCondLU(A)
   ```

   Considerar distintas matrices, por ejemplo usando también la rutina `hilb`, y variar el tamaño de las mismas.
5. Rellenar el archivo LUFromBlocks.m para que tome una matriz A y devuelva las matrices L y U correspondientes a su factorización LU pero calculándola usando bloques. Hint: al pensar la matriz como bloques, considerar la posición izquierda superior con dimensión 1×1, o sea un número.
6. Sea $A \in \mathbb{R}^{n\times n}$ y sea $A^{(k)}$ la matriz que se obtiene a partir de A por el método de eliminación Gaussiana cuando las primeras k columnas ya han sido trianguladas.
   (a) Hallar la matriz M_k de tal forma que $M_k A^{k-1} = A^{(k)}$.
   (b) Probar que A es no singular si y sólo si $A^{(k)}$ es no singular.

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 Viernes 9 de Septiembre