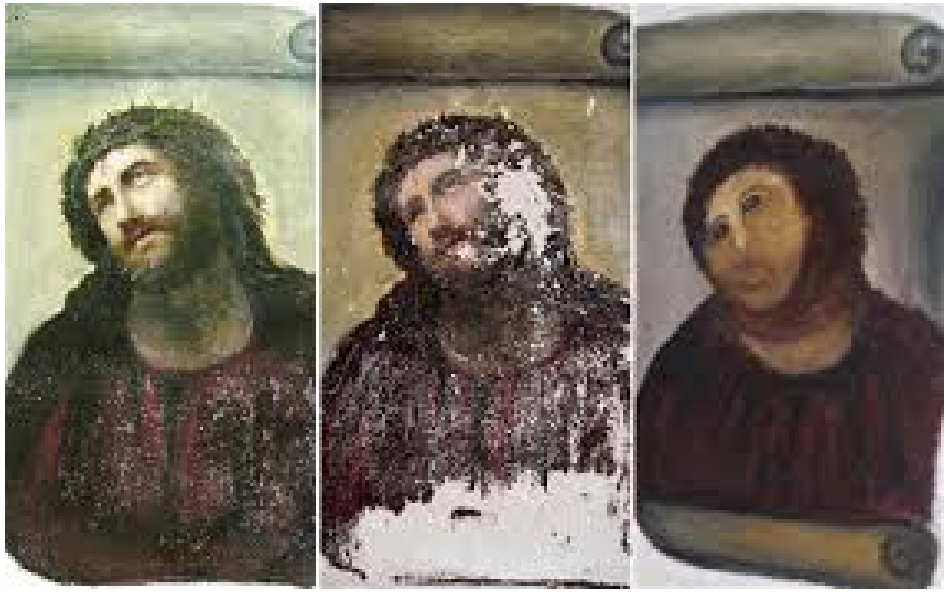
Métodos Numéricos

Image denoising:

Eliminando ruido en imágenes



El problema

- Dada una **imagen ruidosa**, remover el ruido de la misma para obtener una imagen similar a la original.

Pará, pará, pará...

- ¿Qué es una imagen?
- ¿Qué significa que sea ruidosa?
- ¿Cómo puedo extraer ruido de una imagen si no conozco la original?
- ¿Qué significa que la imagen sea similar a la original?

¿Qué es una imagen?

- Para nosotros es simplemente una matriz de píxeles con valores de una escala de grises.
- La leemos en MATLAB con **imread**
- La procesamos como cualquier matriz
- La mostramos con funciones como **imshow**, **image**, **imagesc**
- Es decir que cualquier matriz podemos verla como una imagen. (ejemplo: **imagesc(rand(16))**, arte, arte, arte . . .)

¿Qué es el ruido?

- Para nosotros quiere decir que los valores de una imagen difieren del valor real.
- Existen varios tipos de ruidos (aditivo, multiplicativo, etc.).
- En MATLAB los simulamos **a mano** o con la función **imnoise**.

Un ejemplo real: nivel de exposición ISO

Shhh...

¿Cómo hacemos para sacar el ruido de una imagen?

- Existen muchos métodos.
- Varios de ellos dependen de conocer la distribución de ruido particular de la imagen a procesar.
- Filtro de mediana.
- Filtro pasabajos.
- Filtros basados en DFT y DCT.

Nuestro método

Consultamos a un muy trabajador experto del dominio, y nos dijo:

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

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

donde:

$u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ describe la imagen filtrada

$\tilde{u} : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ describe la imagen a filtrar (con ruido)

Minimizando...

Y mientras seguía trabajando, el experto enseguida agregó:

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

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

Otra vez sopa

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:

$$\lambda u_{i,j} - \left(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}\right) = \lambda \tilde{u}_{i,j}$$

donde ahora $u, \tilde{u} : \Omega \subset \mathbb{Z}^2 \to [0 \ldots 255]$ 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.

¿Quedó mejor?

Una vez que filtramos la imagen, ¿cómo medimos que tan similar quedo a la original?.

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:

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

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

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. Sea $A \in \mathbb{R}^{n \times n}$. Determinar *Verdadero* o *Falso* y, según corresponda, demostrar o dar un contraejemplo:
   - AA^t es una matriz simétrica.
   - Si A es no singular, entonces A^tA es SDP.

Enunciado

4. Rellenar el archivo CHECKCONDLU.m para verificar que una matriz tiene factorización LU, usando para esto el ejercicio 7 de la práctica 2. De forma análoga, rellenar el archivo CHECKCONDCHOL.M para verificar que una matriz es SDP (sugerencia: utilizar la funcion `det` de MATLAB). Evaluar ambas rutinas, por ejemplo de la siguiente forma:

```
>> A = rand(5,5)
>> CheckCondLU(A)
>> CheckCondChol(A'*A)
```

Considerar distintas matrices, por ejemplo usando también la rutina `hilb`, y variar el tamaño de las mismas.

Enunciado

5. Rellenar el archivo CHOLFROMLU.m para que tome una matriz A y devuelva la matriz L correspondiente a su factorización de Cholesky. En este caso se comienza teniendo la factorización LU de A y se quiere generar la factorización de Cholesky. Se debe incluir código para chequear la factorización con respecto al comando *chol* de MATLAB.
6. Rellenar el archivo CHOLFROMBLOCKS.m para que tome una matriz A y devuelva la matriz L correspondiente a su factorización de Cholesky. En este caso se comienza teniendo la factorización de Cholesky de la submatriz principal de A que omite la última fila y la última columna, y se quiere generar la factorización de Cholesky. Se debe incluir código para chequear la factorización con respecto al comando *chol* de MATLAB.