

Eliminando ruido en imágenes

Métodos Numéricos

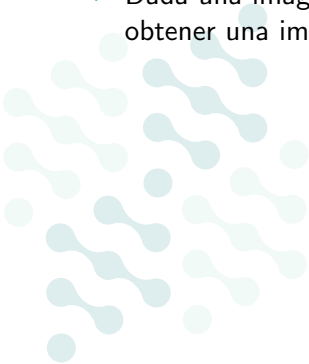
Departamento de Computación, FCEyN, Universidad de Buenos Aires.

2 de Septiembre de 2016



El problema

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



Oye, espacio cerebritito...

- ▶ ¿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?

Oye, espacio cerebritito...

- ▶ ¿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?

Oye, espacio cerebritito...

- ▶ ¿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?

Oye, espacio cerebritito...

- ▶ ¿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 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 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 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 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, puntual).
- ▶ En MATLAB los simulamos *a mano* o con la función *imnoise*.

¿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, puntual).
- ▶ En MATLAB los simulamos *a mano* o con la funcion *imnoise*.

¿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, puntual).
- ▶ En MATLAB los simulamos *a mano* o con la funcion *imnoise*.

Shhh, extrayendo el ruido...

¿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 pasabajos.
- ▶ Filtros basados en DFT y DCT.
- ▶ Filtro de mediana.

Shhh, extrayendo el ruido...

¿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 pasa-bajos.
- ▶ Filtros basados en DFT y DCT.
- ▶ Filtro de mediana.

Shhh, extrayendo el ruido...

¿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 pasabajos.
- ▶ Filtros basados en DFT y DCT.
- ▶ Filtro de mediana.

Shhh, extrayendo el ruido...

¿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 pasabajos.
- ▶ Filtros basados en DFT y DCT.
- ▶ Filtro de mediana.

Shhh, extrayendo el ruido...

¿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 pasabajos.
- ▶ Filtros basados en DFT y DCT.
- ▶ Filtro de mediana.

Nuestro método

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 $u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ describe la imagen filtrada y $\tilde{u} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ 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.

Minimizando...

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


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

Discretizando...

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:

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

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

¿Quedó mejor?

Una vez que filtramos la imagen, ¿cómo medimos qué tan similar quedó 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:

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

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).

¿Quedó mejor?

Una vez que filtramos la imagen, ¿cómo medimos qué tan similar quedó 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:

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

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).

Pasemos al enunciado

