TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica

PROCESAMIENTO Y ANÁLISIS DE IMÁGENES DIGITALES

CE-5201

Tarea 1 - Parte 2

Estudiante	**Carné**
Marcelo Sánchez Solano	2016115728
Emanuel Esquivel López	2016133597
Luis López Salas	2015088115

Matriz de rango reducido

1. Introducción

El problema a resolver es realizar reconstrucción de imágenes satelitales, en si en la reconstrucción de una imagen tomada por algún satélite de la NASA o alguna otra agencia espacial, en particular se toma imagen de la NASA una imagen del planeta Saturno.

Para realizar este proceso se cuenta con una base de datos con 416 imágenes satelitales, con ruido y sin ruido.

Para la resolución del problema se utiliza las muestras tanto limpias como con ruido, así como también deposiciones en valores singulares.

2. Formulación matemática

Se utilizaron las siguientes formulas, **tomadas** del articulo científico [1]

$$P_s = C\widetilde{V}_s \left(\widetilde{V}_s\right)^T \tag{1}$$

Donde:

- C: Es la matriz entrante (Imágenes sin ruido base)
- $\widetilde{V}$: Valores propios derechos de la matriz B (Imágenes con ruido base)„ cuando viene V_s Son las primeras s columnas de V.
- s: Rango de B

$$\hat{Z} = P_r B^\dagger \tag{2}$$

Donde:

- $\hat{Z}$: El filtro que se desea aplicar.
- $B^\dagger$: Pseudo-inversa de la matriz B
- P_r: Es una matriz construida a partir de las primera r columnas de U, filas y columnas de S y columnas de V
- Ademas U, S, V provienen de svd(P_s).

Importante, si se elige un valor de r muy grande, es posible obtener un ruido inverso si r es muy grande.

Cosas importantes, B es el conjunto de imágenes con ruido, en forma de vector columna, C es el conjunto de imágenes con ruido.

Ademas P_r depende de la cantidad de imágenes a utilizar, va desde 1 a 416.

Una vez conocido eso, vamos a la explicación paso a paso.

3. Explicación del algoritmo

3.1. Pasos a seguir

Paso 1: Se carga la imagen en en python *oImg*, y se calculan sus dimensiones *n*,*m*.

Paso 2: Se crean las matrices *B* y *C*, que son *n*m*,K.

Paso 3: Se carga las imágenes con ruido en *B*, y las imágenes limpias en *C*.

Paso 4: Se calcula el svd de la matriz B -> *Ub*, *Sb*, *Vb* y el rango de *B* el cual es *s*.

Paso 5: Se calcula *Vs* que son las *s* primeras columnas de *Vb*

Paso 6: Se calcula *Ps* el cual es la multiplicación de *C*, *Vs*, *Vs'*

Paso 7: Se calcula el svd de la matriz Ps -> *U*, *S*, *V*.

Paso 8: Se crea la matriz de salida *nImg*

Paso 9: Se sacan las *r* columnas de la matriz *U* y *V*, y las primeras *r* filas y *r* columnas de *S*

Paso 10: Se calcula el valor de *Pr* como *Ur * Sr * Vr'*

Paso 11: Se calcula la matriz *Z* que es el filtro como *Pr*, pseudo-inversa(B)

Paso 12: Se calcula imgClean = reshape(*Z* * oImg)(*n*,*m*)

Paso 13: Se visualiza la imagen limpia.

3.2. Pseudo-codigo

Algorithm 1 Matriz de rango reducido para limpiar imágenes

Input: oImg: es la imagen la cual tiene ruido y quiere limpiarse
Output: nImg: Imagen nueva reconstruida o limpia

```
oImg ← limpiar.png
m, n ← size(oImg) # dimensiones de la matriz
total ← total de imagenes
B ← matiz[m*n, total], B = matiz[m*n, total]
r ← [1,40,120,220,300,380,416]
for i = 1 : total; i + +; do
    v_ limpio ← sat_original i .png
    v_ ruido ← sat_ruido i .png
    C[:, i] ← v_ limpio
    B[:, i] ← v_ ruido
s ← rango(B)
Ub, Sb, Vb ← svd(B)
Vs ← Vb[:, : s] # primeras s columnas de Vb
Ps ← C * Vs * Vs'
U, S, V ← svd(Ps)
nImg = matriz[n,m]
for x = 1 : len(r); x + +; do
    Ur ← U[:, : r[x]] # primeras r columnas de U
    Sr ← S[: r[x], : r[x]] # primeras r columnas y r filas de de S
    Vr ← V[:, : r[x]] # primeras r columnas de V
    Pr ← Ur * Sr * Vr'
    Bt ← pinv(B) # Pseudo-inversa de B
    Z ← Pr * Bt
    clean ← Z * tovect(oImg) # Vector columna de oImg
    nImg ← toimg(clean) # La imagen se reorganiza a una matriz m*n
    Se visualiza oImg, nImg # nImg cambia con cada iteracion, distintos r
Fin
```

4. Análisis de resultados

En esta sección veremos los resultados obtenidos por el algoritmo, en las siguientes imágenes.

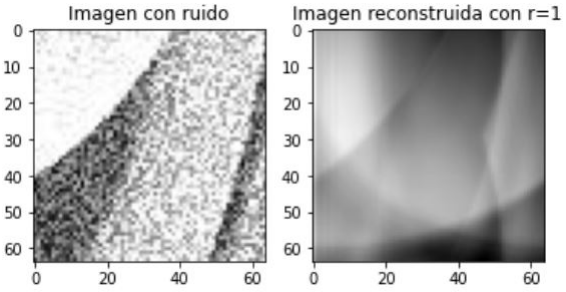

Figura 1: Comparación de reconstrucción con $r = 1$

Como se puede ver en esta comparacion, utilizando solo $r = 1$ mas bien empeora la imagen, ya que tomando solo 1, es imposible la reconstrucción.

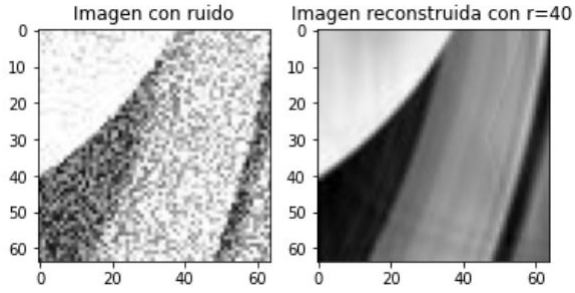

Figura 2: Comparación de reconstrucción con $r = 40$

Ya para una seleccion de $r = 40$ se ve casi perfecto la imagen de la reconstrucción.

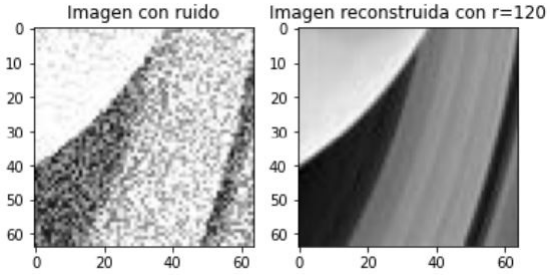

Figura 3: Comparación de reconstrucción con $r = 120$

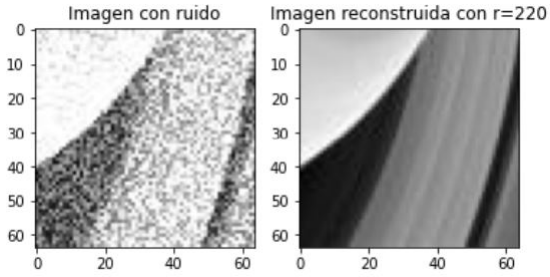

Figura 4: Comparación de reconstrucción con $r = 220$

Como se ve a partir de el valor de 120 y superior el cambio no es apreciable, prácticamente no cambia la imagen de la reconstrucción.

5. Referencias

[1] J. Chung, M. Chung. Computing Optimal Low-Rank Matrix Approximations for Image Processing