



INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA DE INGENIERÍA EN COMPUTADORES

PROCESAMIENTO Y ANÁLISIS DE IMÁGENES DIGITALES

Reconstrucción de Imágenes Satelitales

Estudiantes:

Arturo CÓRDOBA V.
Fabián GONZÁLEZ A.
Erick CARBALLO P.
Sebastián MORA R.

Profesor:

PhD. Juan Pablo SOTO

25 de mayo de 2020

1. Reconstrucción de Imágenes Satelitales

Este documento presenta la solución al problema donde se requiere eliminar el ruido de la imagen *limpiar.jpg*, esto implementando el Teorema 3.1 explicado en el artículo científico *Computing Optimal Low-Rank Matrix Approximations for Image Processing*, el cual reconstruye imágenes utilizando el concepto de la descomposición en valores singulares y utilizando una matriz de rango reducido, esto se logra utilizando la base de datos de los archivos *original.zip* y *ruido.zip*.

Para la solución al problema, se cuentan con dos matrices B y C . Donde C es la matriz con cada una de las imágenes vectorizadas del archivo *original.zip* que contiene 416 imágenes, de tamaño 64×64 , del planeta Saturno tomadas por la NASA. La matriz B contiene cada una de las imágenes anteriores, pero con la diferencia que están alteradas con un ruido aditivo de tipo Gaussiano.

En primer lugar se debe realizar la descomposición de valores singulares (SVD) de la matriz $B \in \mathbb{R}^{n \times n}$ que se realiza de la siguiente manera:

$$B = U \Sigma V^T$$

Donde Σ es la matriz diagonal que contiene los valores singulares, $\sigma_1 \geq \dots \geq \sigma_n > 0$, y las matrices ortogonales $U = [u_1, u_2, \dots, u_n]$ y $V = [v_1, v_2, \dots, v_n]$ contienen los valores singulares de izquierda y derecha u_i y v_i , $i = 1, \dots, n$, respectivamente. Utilizando un valor entero positivo $r \leq \text{rank}(A)$, se define como:

$$B_r = U_r \Sigma_r V_r^T$$

Donde $U_r = [u_1, u_2, \dots, u_r] \in \mathbb{R}^{n \times r}$, $V_r = [v_1, v_2, \dots, v_r] \in \mathbb{R}^{n \times r}$ y

$$\Sigma_r = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_r \end{bmatrix} \in \mathbb{R}^{r \times r}$$

Entonces la matriz $B_r^t = V_r \Sigma_r^{-1} U_r^T$, se define como la matriz pseudo-inversa de B_r . de vuelta al problema, si V_r es la matriz de vectores singulares derechos de B y se define como $P = C V_s (V_s)^T$ donde $s = \text{rank}(B)$, entonces:

$$Z = P_r B^t$$

Donde para realizar la reconstrucción de la *limpiar.jpg*, se debe realizar la SVD a P para obtener P_r y utilizar un valor de rango reducido r de la matriz Z , utilizando los valores $r = 1, 40, 80, 120, 180, 220, 260, 300, 340, 380, 400, 416$, que viene a solucionar el problema de minimización planteado en el artículo:

$$Z = \arg \min_{\text{rank}(Z) \leq r} \frac{1}{K} \|ZB - C\|_F^2$$

Donde da como resultado del filtrado para la eliminación del ruido de la imagen *limpiar.jpg*. Para mostrar dicho resultado se tiene la imagen *limpiar.jpg* y su respectivo resultado final después de haber ejecutado el algoritmo desarrollado con base al Teorema 3.1 del artículo científico:

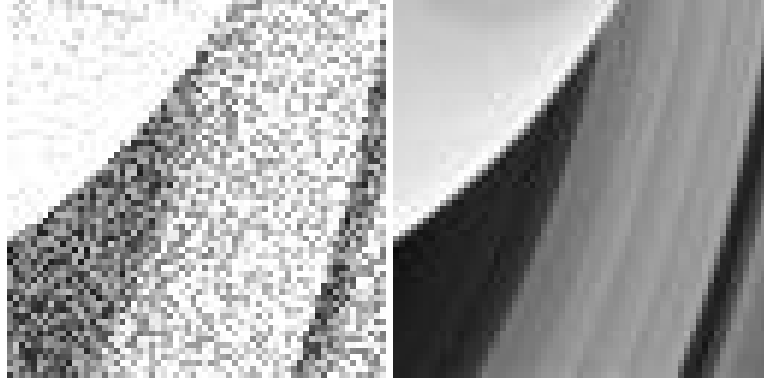


Figura 1: Imagen izquierda corresponde a limpiar.jpg, derecha la imagen filtrada

2. Pseudocódigo

En el algoritmo 1 se muestra el pseudocódigo del Teorema 3.1. Como entradas se reciben la matriz B , la matriz C y el vector columna A . En B se almacenan las imágenes vectorizadas con ruido, en C se almacenan las imágenes vectorizadas sin ruido, A es la imagen vectorizada que se desea filtrar. Como salida se obtiene un vector columna X , que corresponde a la imagen filtrada.

Algorithm 1 Teorema 3.1

Entradas: $B, C \in \mathbb{R}^{m \times n}$, $A \in \mathbb{R}^{(m \cdot n) \times 1}$

Salida: $X \in \mathbb{R}^{(m \cdot n) \times 1}$

- 1: $B = U \cdot \Sigma \cdot V^T$ Descomposición SVD de B
 - 2: $s = \text{rank}(B)$
 - 3: $P = C \cdot V_s \cdot V_s^T$
 - 4: $P = U \cdot \Sigma \cdot V^T$ Descomposición SVD de P
 - 5: $P_r = U_r \cdot \Sigma_r \cdot V_r^T$
 - 6: $Z = P_r \cdot B^+$
 - 7: **return** $X = Z * A$
-

3. Resultados obtenidos

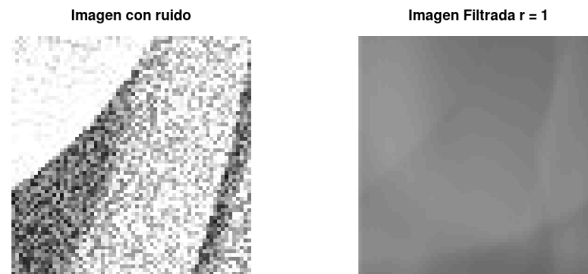


Figura 2: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 1$

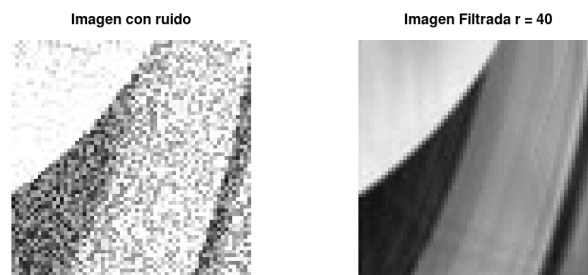


Figura 3: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 40$

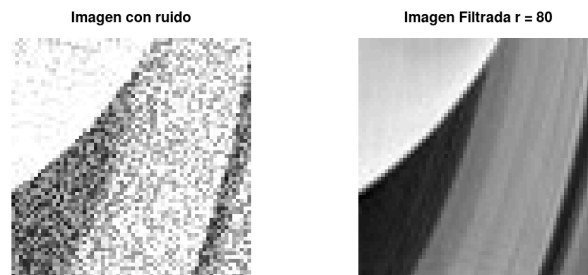


Figura 4: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 80$

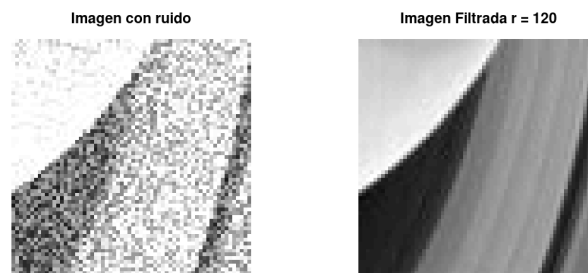


Figura 5: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 120$

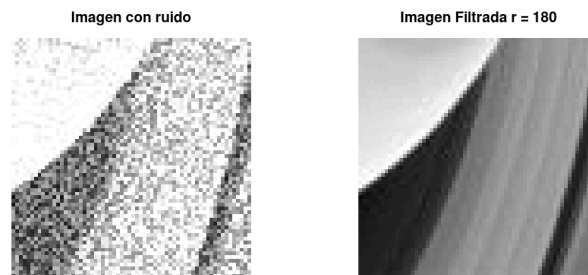


Figura 6: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 180$

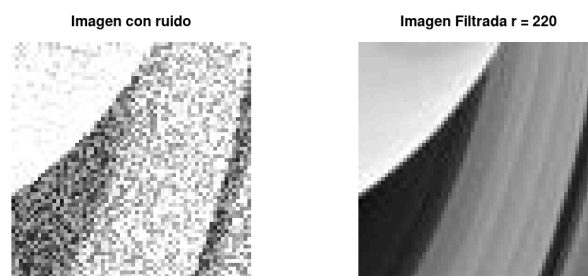


Figura 7: Comparación Imagen con Ruido vs. Imagen Filtrada $r220$

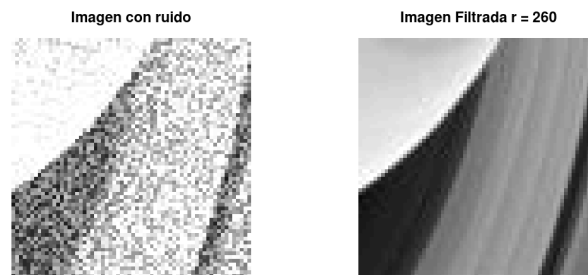


Figura 8: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 260$

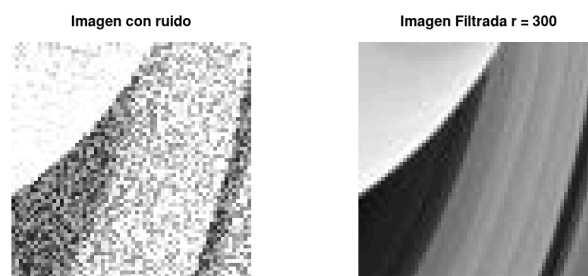


Figura 9: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 300$

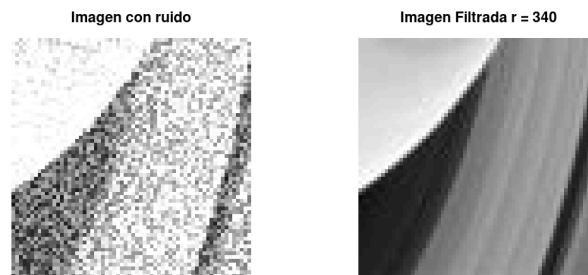


Figura 10: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 340$

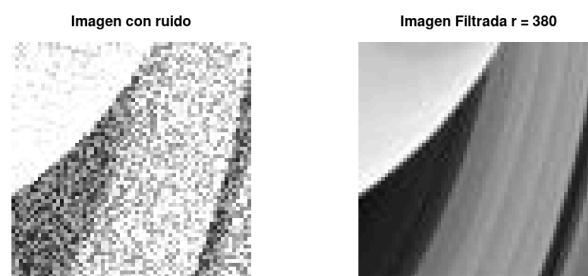


Figura 11: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 380$

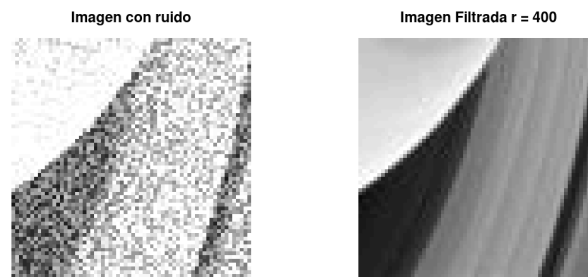


Figura 12: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 400$

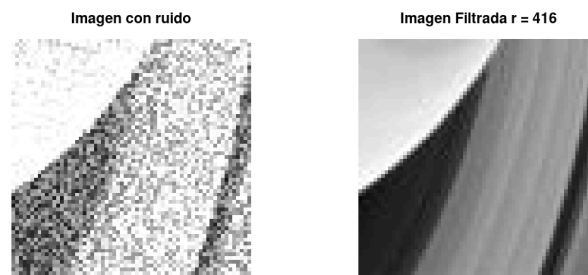


Figura 13: Comparación Imagen con Ruido vs. Imagen Filtrada $r = 416$

4. Referencias Bibliográficas

[1] J. Chung and M. Chung, "Computing Optimal Low-Rank Matrix Approximations for Image Processing", Blacksburg, 2013.