

# Aplicación de la Descomposición en Valores Singulares en la Compresión de Imágenes Digitales

Luciano Savoie, María Mercedes Gaitán, Ernesto Klimovsky

Grupo de Investigación en Enseñanza de la Matemática en Carreras de Ingeniería (GIEMCI). Proyecto de investigación:

La trascendencia del álgebra lineal y sus aplicaciones en el ciclo superior de ingeniería

Facultad Regional Paraná, Universidad Tecnológica Nacional

Almafuerte 1033, 3100, Paraná, Entre Ríos, Argentina

savoieluciano@gmail.com, mgaitan@frp.utn.edu.ar, erklimo@gmail.com

**Resumen.** En este trabajo abordamos el estudio de un tema de Álgebra Lineal como es la Descomposición en Valores Singulares y su aplicación directa a una tecnología que nos rodea diariamente como son las imágenes digitales. Además de introducir los conceptos matemáticos necesarios, exponemos aquellos relacionados al tratamiento de imágenes digitales. Realizamos simulaciones mediante software y mostramos un ejemplo práctico con el objetivo de facilitar la asimilación de los conceptos.

La modelización matemática de las imágenes digitales como matrices las convierte también en una excelente herramienta docente para motivar al estudiante en la profundización del estudio de Álgebra Lineal.

**Palabras Clave:** Descomposición en valores singulares, Compresión, Procesamiento de imágenes.

## 1 Introducción

La descomposición de una matriz en factores, es decir, la posibilidad de expresarla como producto de dos o más matrices, tal es el caso de la descomposición LU y la QR, tiene una gran utilidad en variadas aplicaciones.

En este trabajo utilizamos una de las descomposiciones más importantes que puede aplicarse a cualquier matriz  $A$  de tamaño  $m \times n$ , la Descomposición en Valores Singulares (DVS).

La DVS se utiliza en una amplia variedad de áreas del procesamiento de imágenes. Aquí trabajaremos una de las aplicaciones más conocidas de esta descomposición matricial, que es la compresión de imágenes digitales.

Es importante resaltar que el método de compresión JPEG, basado en la Transformada Discreta del Coseno (DCT) no proporciona un nivel de compresión tan elevado como la DVS; en efecto, el método presentado para la DVS aproxima cada imagen por su aproximación “óptima”. En cambio, la DCT transforma todas las imágenes de la misma manera, no es una transformación optimizada a cada imagen particular, como la DVS. Sin embargo, la ventaja de la DCT es su facilidad de aplicación mediante algoritmos rápidos, junto con una eficiencia más que aceptable; por ello, el método de compresión JPEG se ha extendido mundialmente, aunque no sea “óptimo”. [1]

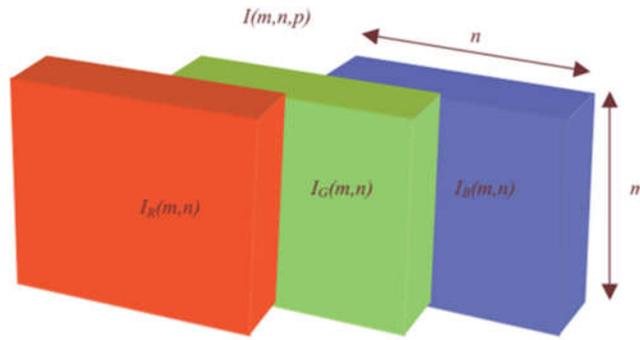
## 2 Marco teórico

### 2.1 Imágenes digitales

Una imagen puede ser definida como una función bidimensional  $f(x,y)$ , donde  $x$  e  $y$  son las coordenadas en el plano y  $f$  es la amplitud de cualquier par de coordenadas  $(x,y)$ , llamada intensidad de la imagen en el punto.

El término nivel de gris se usa para referirse a la intensidad monocromática de las imágenes. Por su parte las imágenes a color están formadas por una combinación individual de imágenes 2-D. En el sistema de color RGB, un color consiste en tres componentes individuales de una imagen que son Red, Green y Blue (rojo, verde y azul respectivamente). Por esta razón, muchas técnicas de desarrollo para imágenes monocromáticas pueden ser extendidas para el procesamiento de imágenes a color.

Cuando  $(x, y)$  y los valores de la amplitud de la función  $f$  son cantidades discretas finitas, a dicha imagen se le llama imagen digital. Una imagen digital está compuesta de un número finito de elementos llamados píxeles, donde cada uno tiene una localidad y un valor establecido. [2]



**Fig. 1.** Esquema de una imagen RGB  $I(m,n,p)$  con sus tres componentes individuales. Aquí  $m$  y  $n$  tienen el mismo significado que para el caso de las imágenes monocromáticas, mientras que  $p$  representa el plano, que para RGB puede ser 1 para el rojo, 2 para el verde y 3 para el azul. [3]

## 2.2 Manejo de imágenes digitales mediante software

Ya sea que utilicemos MatLab o una alternativa en software libre como SciLab, la estructura básica de datos es el arreglo, el cual se puede definir como un conjunto ordenado de números reales o complejos. En el caso de las imágenes, éstas pueden ser representadas por matrices formadas por conjuntos ordenados de valores reales que, a su vez, representan la intensidad de color o de niveles de grises.

Estos programas almacenan las imágenes como arreglos bidimensionales (matrices) en los cuales cada elemento de la matriz corresponde a la intensidad de un píxel de la imagen. Por ejemplo, una imagen de 200 filas por 300 columnas se almacena como una matriz de  $200 \times 300$ . Algunas imágenes, como las imágenes RGB nombradas anteriormente, requieren de un arreglo tridimensional, donde el primer plano representa la intensidad de rojo en los píxeles, el segundo plano, la intensidad de verde de los píxeles, y el tercer plano, la intensidad de azul de los mismos.

Esta convención sustituye el trabajo con imágenes por una representación matricial. Por ejemplo, se puede seleccionar un solo píxel de una imagen-matriz (estos términos serán análogos a partir de ahora para las definiciones que siguen) de la forma  $I(2,15)$ , con lo cual el software regresará el valor del píxel localizado en el renglón 2, columna 15 de  $I$ . [2]

## 2.3 ¿Qué es la Descomposición en Valores Singulares?

En esta parte del trabajo, damos las definiciones y teoremas matemáticos necesarios para comprender los temas a desarrollar. Para más detalles y demostraciones recomendamos ver las referencias [4] y [5].

Dada una matriz  $A$  real de dimensiones  $m \times n$ , existe una matriz  $U$  ortogonal de orden  $m$ , una matriz diagonal  $S$  de dimensiones  $m \times n$  y una matriz  $V$  ortogonal de orden  $n$  de forma que:

$$A = USV^t \quad (1)$$

siendo  $V^t$  la matriz transpuesta de  $V$ .

Los elementos no nulos de la diagonal de  $S$  se denominan valores singulares de  $A$  y son números no negativos, pues son la raíz cuadrada de los autovalores no nulos de la matriz  $A^t A$  o de  $A A^t$ , ya que ambos coinciden (salvo el autovalor nulo, si es que lo tienen). Las columnas de  $U$  y  $V$  se denominan vectores singulares por la izquierda y por la derecha respectivamente. Esta factorización de  $A$  es lo que llamamos *Descomposición en Valores Singulares de  $A$* .

Una propiedad muy interesante de la DVS de una matriz  $A$  es que el número de valores singulares no nulos de ésta coincide con el rango de dicha matriz.

Supongamos que una matriz  $A$  es de rango  $r$  y que los valores singulares no nulos están ordenados en orden decreciente:

$$s_1 \geq s_2 \geq \dots \geq s_r \geq 0 \quad (2)$$

Se puede comprobar que:

$$A = \sum_{j=1}^r s_j u_j v_j^t = s_1 u_1 v_1^t + s_2 u_2 v_2^t + \dots + s_r u_r v_r^t \quad (3)$$

donde  $s_j$  es un escalar (el valor singular correspondiente), mientras que  $u_j$  es la columna  $j$ -ésima de la matriz  $U$  y  $v_j$  es la columna  $j$ -ésima de la matriz  $V$ .

La identidad (3) expresa  $A$  como suma de  $r$  matrices de rango 1. A cada matriz  $s_j u_j v_j^t$  se le llama  $j$ -ésimo modo de  $A$ , así  $s_1 u_1 v_1^t$  es el primer modo de  $A$ ,  $s_2 u_2 v_2^t$  es el segundo modo, etc.

Ejemplo: Sea  $A$  una matriz  $3 \times 2$  con los siguientes valores:

$$A = \begin{bmatrix} 2 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} \quad (4)$$

A partir de (1) expresamos su DVS como:

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (5)$$

Los valores singulares son  $s_1=3$  y  $s_2=2$ . Y de (3) podemos expresar:

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} = 3 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = s_1 u_1 v_1^t + s_2 u_2 v_2^t \quad (6)$$

$$A = A_1 + A_2 \quad (7)$$

$$\begin{bmatrix} 2 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (8)$$

De donde se observa que  $\text{Rango}(A) = 2$ ,  $\text{Rango}(A_1) = 1$  y  $\text{Rango}(A_2) = 1$ .

## 2.4 La DVS y el teorema de aproximación

Un resultado fundamental para las aplicaciones de la DVS es el siguiente *Teorema de Aproximación* cuya demostración puede verse en [5]; supongamos que a partir del cálculo de la DVS tenemos  $A$  (de rango  $r$ ) expresada según (3), entonces la mejor aproximación de  $A$  en la norma espectral (o norma 2) mediante matrices de su mismo tamaño y rango  $k < r$  es la matriz:

$$A_k = \sum_{j=1}^k s_j u_j v_j^t = s_1 u_1 v_1^t + s_2 u_2 v_2^t + \dots + s_k u_k v_k^t \quad (9)$$

que se obtiene tomando solamente los  $k$  valores singulares más grandes. La norma espectral de una matriz  $A$  viene dada por:

$$\|A\|_2 = s_1 \quad (10)$$

Es decir, es igual al mayor de sus valores singulares. En relación con la mejor aproximación mediante matrices de rango  $k < r$  se tiene entonces:

$$\|A - A_k\|_2 = s_{k+1} \quad (11)$$

Este resultado nos dice que la mejor aproximación a la matriz  $A$  mediante matrices de rango  $k$  es la matriz dada por  $A_k$ , y que el error cometido con dicha aproximación vale  $s_{k+1}$ , el valor singular  $k$ -ésimo+1.

Si consideramos el ejemplo realizado en la sección anterior, la norma 2 de  $A$  es:

$$\|A\|_2 = s_1 = 3 \quad (12)$$

Con lo cual verificamos (10). Si ahora tomamos la aproximación  $A_k$  con rango  $k = 1$  (recordar que  $k < r$ ), tenemos:

$$A_1 = \begin{bmatrix} 0 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix} \quad (13)$$

El error cometido con esta aproximación es:

$$\|A - A_1\|_2 = 2 = s_2 \quad (14)$$

Resultado que comprueba lo obtenido en (11).

**Algoritmo 1.** Para la implementación computacional se puede utilizar el paquete informático MatLab. Una vez definida la matriz  $A$ , la norma 2 puede ser calculada mediante la función *norm*. Para el caso de la DVS, la función *svd* calcula los valores singulares de  $A$  y las matrices  $U$  y  $V$  que cumplen la condición (1).

```
norm(A, 2)
[U, S, V] = svd(A)
```

### 3 Aplicación a la compresión de imágenes digitales

Hasta aquí, hemos expuesto de una manera resumida la descripción matemática de la descomposición en valores singulares. De las aplicaciones que existen, una es en la compresión de imágenes digitales. Esto permite que ocupen menos espacio en nuestro disco duro, se transmitan de manera más eficiente y viajen más rápido por internet. [6]

Por una cuestión práctica y para facilitar la comprensión de los temas, a continuación mostramos ejemplos desarrollados sobre imágenes monocromáticas de 8 bits. En ellas, el valor de cada píxel se corresponde a un determinado valor en una escala de grises de  $2^8$ , es decir, 256 tonalidades distintas. La misma está comprendida en un rango de números enteros de 0 a 255 en el cual cada valor está asociado a distintas graduaciones de gris, desde el negro cuyo valor es el 0, hasta el blanco cuyo valor es 255. Todos los conceptos desarrollados pueden hacerse extensivos a imágenes color, recordando que éstas son tratadas como arreglos tridimensionales, como se mencionó anteriormente. [2]

Un ejemplo de imagen monocromática y su correspondiente representación matricial se presenta en la Fig. 2.

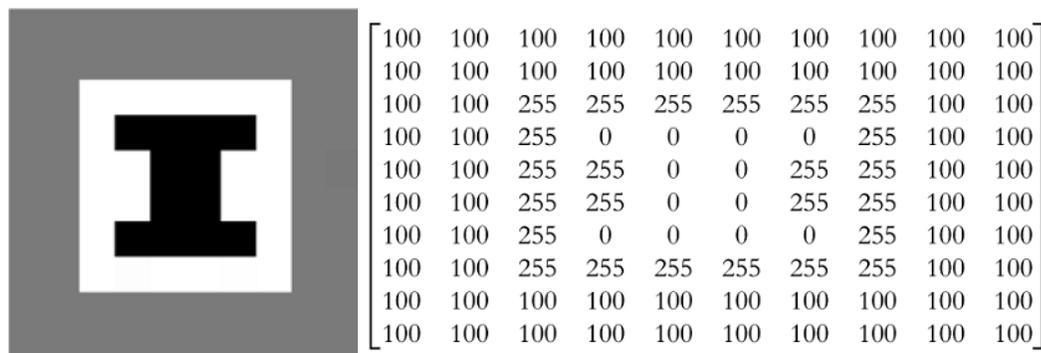


Fig. 2. Representación de una imagen en escala de grises mediante una matriz

#### 3.1 Implementación

El objetivo ahora es poder desarrollar un algoritmo que permita comprimir la imagen sin perder una cantidad considerable de resolución. Es decir, tratar de conseguir la misma imagen, quizá con unos tonos de grises menores, pero con muchos menos datos.

Dada una matriz  $A$  de tamaño  $m \times n$ , la transferencia completa de dicha matriz requiere la transmisión de  $m \cdot n$  valores. El teorema de aproximación obtenido en (9) permite escribir la matriz de rango  $k$  que mejor aproxima a  $A$  de la siguiente manera:

$$A_k = \sum_{j=1}^k s_j u_j v_j^t \quad (15)$$

Incrementar en una unidad el rango de la matriz con la que aproximamos a la matriz original supone la transmisión de  $(m+n+1)$  valores adicionales: los  $m$  elementos de  $u_j$ , los  $n$  elementos de  $v_j^t$  y 1 de  $s_j$ . Así, si en vez de transmitir  $A$ , transmitimos la información necesaria para generar la aproximación  $A_k$ , es suficiente transmitir  $k(m+n+1)$  datos en lugar de los  $m \cdot n$  que forman la matriz original, lo que demuestra la reducción conseguida en el número de datos (obsérvese que  $A_k$  sigue siendo una matriz de dimensiones  $m \times n$ ).

El procedimiento tendrá interés solo si se puede tomar:

$$k \ll \frac{mn}{m+n+1} \quad (16)$$

En caso contrario, además de tener que transmitir más datos que los necesarios para formar la imagen original, habríamos tenido que calcular la DVS de la matriz, lo que implica un aumento en el número de operaciones necesarias para la transferencia de datos.

Resultará ilustrativo realizar un ejemplo práctico, en este caso, con una imagen de tamaño  $480 \times 640$  (Marcie.bmp) que es la fotografía de una mujer. Comenzamos cargando una matriz de tamaño  $480 \times 640$  correspondiente a la imagen y luego determinando su rango.



Fig. 3. Fotografía (Marcie.bmp) utilizada como ejemplo

**Algoritmo 2.** Las funciones *imread* e *imshow* permiten la lectura y la visualización de la imagen respectivamente. Para el caso del rango, prestar atención a que no aparece el “;” luego de la instrucción *rank* para que aparezca el resultado en pantalla.

```
A = imread('Marcie.bmp');
imshow(A);
A = im2double(A);
rank(A)
```

Comprobamos que el rango de  $A$  es el máximo posible, es decir  $r = 480$ . Luego, realizamos la aproximación  $A_k$  de (9) para un valor determinado de  $k$ , y vemos la correspondiente imagen aproximada, ejecutando el **Algoritmo3**.

**Algoritmo 3.** Para un valor determinado de  $k$  (en este caso  $k=5$ ), obtenemos y visualizamos la aproximación correspondiente.

```
[U,S,V] = svd(A);
k = 5; % definir k
Ak = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
imshow(Ak);
```

### 3.2 Ensayos

Para comprobar la aplicabilidad de la DVS en la compresión de imágenes ejecutamos el **Algoritmo 3** para distintos valores de  $k$ . Además, para cada uno de los casos calculamos el error relativo (usando la norma 2) [7]:

$$\text{Error relativo} = \frac{s_{k+1}}{s_1} \quad (17)$$

Y también el factor de compresión, que representa el cociente entre el espacio requerido para almacenar la imagen aproximada y el necesario para almacenar la imagen original [7]:

$$\text{Factor de compresión} = \frac{k(480+640+1)}{480 \cdot 640} \quad (18)$$

**Algoritmo 4.** Se realizan unos pequeños cambios al **Algoritmo 3**, agregándole la posibilidad de determinar el error relativo y el factor de compresión para un valor de  $k$  determinado. Nuevamente se omite el signo “;” en dos instrucciones para que aparezca el resultado en pantalla.

```
[U,S,V] = svd(A);
k = 5; % definir k
Ak = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
imshow(Ak);

%Error Relativo
E = A - Ak;
e = norm(E,2) / norm(A,2)

%Factor de compresión
fc = (k*(480+640+1)) / (480*640)
```

Los resultados correspondientes a tres casos distintos son los siguientes:

- Aproximación de rango  $k = 10$  ( $A_{10}$ ): Para este primer caso el factor de compresión resulta ser de 0.0365, es decir que la imagen aproximada  $A_{10}$  ocupa menos del 4% de la imagen original  $A$ . El error relativo  $e_{11}/e_1$  es 0.0507, y vemos como tomando solamente los primeros 10 valores singulares de los 480 no es posible dilucidar con claridad de que se trata de la fotografía, como puede observarse en la **Fig. 4**.



**Fig. 4.** Aproximación de rango  $k = 10$

- Aproximación de rango  $k = 50$  ( $A_{50}$ ): En este caso el factor de compresión resulta de 0.1825, es decir que la imagen aproximada ocupa poco más del 18% de la imagen original. Como observamos en la **Fig. 5**, es posible distinguir perfectamente que se trata de la fotografía de una mujer. Esta mejora se ve reflejada también en la disminución del error relativo  $e_{51}/e_1$ , que ahora vale 0.0114. Sin embargo, el nivel de detalle de la fotografía es bajo, algo que se ve, por ejemplo, en los rasgos faciales y el pelo.



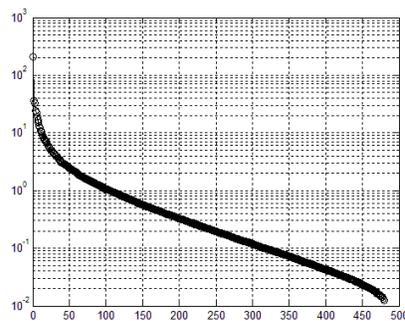
**Fig. 5.** Aproximación de rango  $k = 50$

- Aproximación de rango  $k = 100$  ( $A_{100}$ ): En este último caso el factor de compresión es de 0.3649 (aproximadamente 36,5%) y obtenemos una calidad casi idéntica a la conseguida en la imagen original (**Fig. 6**). El error relativo es ahora bastante más pequeño que en los dos casos anteriores  $e_{101}/e_1$  (0.050).



**Fig. 6.** Aproximación de rango  $k = 100$

Si hay valores singulares muchos mayores que otros, es evidente que éstos son más determinantes en la formación de  $A$  que los valores singulares pequeños. Si además de calcularlos, los graficamos usando una escala logarítmica (debido a la posibilidad de que haya valores singulares muy grandes), podremos apreciar de forma aproximada la preponderancia de los valores singulares más grandes frente al resto de ellos. [8]



**Fig. 7.** Representación gráfica de los valores singulares

Observando la gráfica de la **Fig. 7**, correspondiente a la **Fig. 3**, se aprecia que aproximadamente los 100 primeros valores singulares dominan frente al resto de ellos, es decir, que son más determinantes en la formación de la matriz  $A$ . Por lo que aproximando  $A$  por la suma de sus primeros modos, las imágenes de estas matrices deberán asemejarse a las de  $A$ , a medida que aumenta el número de modos que tomamos en consideración. El ejemplo desarrollado nos demuestra claramente que con pocos modos conseguimos resultados muy buenos.

## 4 Conclusiones y trabajos a futuro

A lo largo del trabajo hemos visto la utilización de la Descomposición en Valores Singulares en una aplicación concreta, como es la compresión de imágenes digitales. Pudimos observar los cambios en las imágenes a medida que aumentamos el rango  $k$  de la aproximación, y llegamos a la conclusión de que lo primordial para recuperar la información de nuestra imagen original son sólo unos pocos valores singulares (con sus respectivos vectores  $u_i$  y  $v_i$ ): los mayores de todos los que posee la matriz.

Además, esta aplicación de la teoría no solo ayuda al estudiante a entender mejor el concepto de la Descomposición en Valores Singulares, sino que adicionalmente motiva su aprendizaje al observar su utilidad en un ejemplo de la vida real.

Como trabajo a futuro se plantea la posibilidad de realizar una aplicación de la Descomposición en Valores Singulares en un campo de la estadística, que es el análisis de componentes principales (PCA). Esto es muy utilizado en el reconocimiento de rostros, en una técnica conocida como *Eigenfaces*.

## Referencias

1. Domínguez Jiménez, M.E: Matrices: un modelo para las fotografías digitales. *Modelling in Science Education and Learning*, Volumen 4, No. 13, pp.169–179 (2011)
2. Savoie, L.: Algebra matricial y su aplicación en el procesamiento digital de imágenes. *Actas de las Jornadas de Jóvenes Investigadores Tecnológicos 2016*. (En proceso de edición)
3. Cuevas Jimenez E.; Zaldivar Navarro D.; Rojas R.: *Computer Vision using MatLAB and the Toolbox of Image Processing*. Freie Universitat Berlin, pp. 27 (2005)
4. Strang, G.: *Algebra lineal y sus aplicaciones*. Thomson, pp.331-338 (2007)
5. Golub, G.H.; Van Loan C.F: *Matrix computations*. The Johns Hopkins University Press, pp.70-73 (1996)
6. Rojas Matas, A.; Cano Rojas, A.: Trabajando con imágenes digitales en clases de Matemáticas. *La Gaceta de la Real Sociedad Matemática Española*, Volumen 13, No. 2, pp. 317-336 (2010)
7. Fernández de las Heras, J.J.M.: La descomposición en valores singulares (SVD) y algunas de sus aplicaciones. *La Gaceta de la Real Sociedad Matemática Española*, Volumen 8.3, pp. 795-810 (2005)
8. Zaballa, I.: *Valores singulares. ¿Qué son? ¿Para qué sirven?*. Departamento de Matemática Aplicada y EIO, Universidad del País Vasco.