

Revista Eletrônica
Paulista de Matemática

ISSN 2316-9664
v. 22, n. 1, jul. 2022
Iniciação Científica

GlauCIA Maria Bressan

Câmpus Cornélio Procópio
Universidade Tecnológica Federal
do Paraná
glauciabressan@utfpr.edu.br

Jennifer Fernanda dos Santos

Câmpus Cornélio Procópio
Universidade Tecnológica Federal
do Paraná
jenniferfernanda151099@gmail.com

André Luis Machado Martinez

Câmpus Cornélio Procópio
Universidade Tecnológica Federal
do Paraná
martinez@utfpr.edu.br

Aplicação da Decomposição em Valores Singulares para compressão de imagens

Application of the Singular Value Decomposition to Image Compression

Resumo

Este trabalho tem por objetivo apresentar a Decomposição em Valores Singulares e desenvolver uma aplicação computacional para compressão e reconstrução de imagens digitais, as quais podem ser armazenadas e transmitidas utilizando-se menos informações que a imagem original. Para isso, é considerado o sistema de cores RGB. A imagem é descrita por meio de três matrizes, sendo uma para cada canal dessas cores, e a decomposição é aplicada para cada matriz. No caso de imagens em escalas de cinza, apenas uma matriz é necessária. Em seguida, o Teorema de Eckart-Young (1937) é aplicado para a reconstrução da imagem. Programas computacionais utilizando Matlab são apresentados, juntamente com exemplos numéricos e aplicações em imagens, as quais são compactas e exibidas de forma colorida e em escalas e cinza, necessitando de uma quantidade muito menor de valores para serem armazenados.

Palavras-chave: Teorema de Eckart-Young. Aplicação computacional. Decomposição de matrizes. Compressão de imagens. Sistema RGB

Abstract

The objective of this paper is to present the Singular Value Decomposition and develop a computational application for compression and reconstruction of digital images, which can be stored and transmitted using less information than the original image. For this, the RGB color system is considered. The image is described using three matrices, one for each channel of these colors, and decomposition is applied for each matrix. For grayscale images, only one matrix is required. Then, the Eckart-Young Theorem (1937) is applied to image reconstruction. Computational programs using Matlab are presented, with numerical examples and applications in images, which are compact and displayed in color and grayscale, requiring a much smaller amount of values to be stored.

Keywords: Eckart-Young Theorem. Computational application. Matrices decomposition. Image compression. RGB System.



1 Introdução

A Decomposição em Valores Singulares (SVD, do inglês *Singular Value Decomposition*), é um método numérico pertencente ao campo de estudo da Álgebra Linear, utilizado para fatorar matrizes retangulares. É um dos resultados mais importantes na Álgebra Linear, tanto computacional quanto teórico. A SVD é a base dos métodos mais precisos para a resolução de problemas de mínimos quadrados, para a determinação do posto de matrizes, do espaço-imagem e do espaço nulo de matrizes, e da solução de vários problemas envolvendo normas euclidianas (STEWART, 1993).

A Decomposição em Valores Singulares foi enunciada de forma independente pelos matemáticos Eugênio Beltrami em 1873 e Camille Jordan em 1874. Porém, esses estudiosos mostraram sua existência apenas para matrizes quadradas reais. Em 1936, Carl Eckart e Gale Young generalizaram e provaram a existência da SVD para matrizes retangulares reais e complexas, formalizando a aproximação de uma determinada matriz por uma outra de menor posto (ECKART; YOUNG, 1936). A partir daí, o método sofreu diversas modificações, entre estas, em 1965, Gene Howard Golub e William Kahan propuseram uma forma alternativa utilizando reflexões de Householder. Em 1970, foi publicado por Gene Howard Golub e Christian Reinsch o método amplamente utilizado até hoje (STEWART, 1993). No trabalho de Golub, Hoffman e Stewart (1987), foi proposto uma melhor aproximação de posto inferior, no qual um determinado conjunto de colunas da matriz permanece fixo. Algumas aplicações desta generalização são apresentadas.

A teoria e as aplicações da SVD aparecem em vários problemas práticos e reais, como por exemplo: o uso da SVD no telescópio Hubble, para diminuir o tamanho de imagens e enviar para a Terra (GOLUB; VAN LOAN, 1996); construção de biplots sobre inserção de aprendizes no mercado de trabalho (DEMMEL, 1997). As aplicações desta decomposição são vastas e muito utilizadas em muitos algoritmos, devido ao grande número de informações que podem ser obtidas (MOUNIKA; LAKSHMI; ALEKYA, 2015). Em Sadek (2012), é apresentado um *survey* sobre a SVD e suas aplicações no processamento de imagens. A SVD apresenta muitas aplicações em diferentes áreas e ainda há muito a se explorar neste método (STEWART, 1993).

A partir do estudo da SVD, este trabalho apresenta o desenvolvimento de uma aplicação computacional para compressão e reconstrução de imagens digitais de forma que a transmissão seja rápida, barata e sem perda de informações importantes. A compressão de imagens digitais consiste em considerar uma imagem em sua forma matricial, aplicar a SVD para determinar uma aproximação da imagem por vetores e, por fim, reconstruir a imagem aproximada a partir desses vetores. Para isso, foram desenvolvidas implementações computacionais utilizando o Matlab ¹, visto que esse *software* proporciona maior facilidade em trabalhar com matrizes que possuem muitos elementos.

Desta forma, este trabalho apresenta como contribuições: (i) a implementação computacional da função de decomposição em valores singulares e o retorno das matrizes e das informações desse processo, evitando-se o uso da função ‘*svd(A)*,’ presente no *software* Matlab, cujo código não é possível de ser acessado ou alterado. No trabalho de Mounika, Lakshmi e Alekya (2015), por exemplo, a decomposição é feita por meio da função ‘*svd(A)*,’ da qual não é possível extrair informações sobre o processo de decomposição; (ii) o detalhamento, não facilmente encontrado na literatura, de todos os passos da SVD, da compressão e da reconstrução da imagem, envolvendo a teoria e a aplicação. Portanto, este trabalho contribui para a literatura como uma referência que pode ser utilizada tanto pelo seu detalhamento teórico quanto pela aplicação dos algoritmos para compressão de imagens.

O restante deste artigo está dividido da seguinte forma. A Seção 2 apresenta os principais

¹<https://www.mathworks.com>

resultados teóricos relacionadas à compressão de imagens com uso da decomposição SVD. A Seção 3 discute como é o código de cores do padrão RGB. A Seção 4 detalha como foi programado, com o auxílio do *software* MatLab, algoritmos baseados nos resultados apresentados na Seção 2 para a reconstrução de imagens e ilustra os resultados com exemplos. Por fim, a Seção 5 apresenta a conclusão deste trabalho.

2 Decomposição em Valores Singulares

O objetivo da Decomposição em Valores Singulares é fatorar uma matriz retangular A como o produto entre matrizes ortonormais e uma matriz diagonal.

Os conceitos apresentados nesta seção são fundamentados em: Anton e Rorres (2012), Leon (2018) e Steinbruch e Winterle (1995).

2.1 Teorema principal

Teorema 1 (*Decomposição em Valores Singulares*): Toda matriz retangular A de ordem $n \times m$ e posto p pode ser decomposta em:

$$A = U\Lambda V^T$$

em que $U_{n \times p}$ e $V_{m \times p}$ são ortonormais nas colunas, ou seja, $U^T U = U U^T = I$ e $V^T V = V V^T = I$ com $\Lambda_{p \times p} = \text{diag}(\sqrt{\lambda_i})$, tal que $\lambda_i > 0$. Sendo $\lambda_1, \lambda_2, \dots, \lambda_p$ os autovalores não nulos das matrizes $A^T A$ ou AA^T , $U_{n \times p}$ e $V_{m \times p}$ matrizes de p autovetores ortonormais por coluna, respectivamente das matrizes $A^T A$ ou AA^T .

Demonstração:

Seja por suposição a matriz A decomposta em

$$A = U\Lambda V^T.$$

Considerando $U_{n \times n}$ e $V_{m \times m}$ ortonormais, ou seja, $U^T U = U U^T = I$ e $V^T V = V V^T = I$. Sendo a matriz diagonal $\Lambda_{n \times m} = \text{diag}(\sqrt{\lambda_i})$, com $\lambda_i \geq 0$, $i = 1, 2, \dots, m$. Por conveniência e sem perda de generalidade assume-se que $n \geq m$. Constrói-se matrizes ortogonais U e V com acréscimo de $n - r$ e $m - r$ vetores, respectivamente, às colunas de U e V . Assim, uma nova representação para Λ é dada por:

$$\Lambda = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}.$$

A matriz $A^T A$ possui os autovalores $\lambda_1, \lambda_2, \dots, \lambda_m$. A matriz $A^T A$ é simétrica, pois $(A^T A)^T = A^T A$. Como por suposição $A = U\Lambda_m V^T$, então:

$$A^T = V\Lambda_m U^T U\Lambda_m V^T = V\Lambda_m^2 V^T.$$

Como V é ortogonal, segue que:

$$V^T A^T A V = V^T V \Lambda_m^2 V^T V$$

$$V^T A^T A V = \Lambda_m^2.$$

Portanto,

$$V^T A^T A V = \Lambda_m^2 = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_m \end{bmatrix},$$

em que V é a matriz dos autovetores de $A^T A$ em suas colunas.

Assumindo uma situação restritiva, em que apenas r dos λ_i 's sejam não nulos, então a matriz $\Lambda_{n \times m}$ será dada por:

$$\Lambda = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 & \dots & 0 \\ 0 & \sqrt{\lambda_2} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\lambda_r} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} \Lambda_r & 0 \\ 0 & 0 \end{bmatrix}.$$

Essencialmente, Λ_r e Λ_m são idênticas nas primeiras r linhas e colunas. Ocorre que a matriz Λ_m possui $m - r$ linhas e $m - r$ colunas de zeros a mais que a matriz Λ_r . Pode-se ver que $\Lambda^T \Lambda = \Lambda_m^2$, sendo:

$$\Lambda_m^2 = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_r & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix}.$$

Pode-se verificar que $(V^T A^T) A V = \Lambda^T \Lambda$, ou seja, $(A V)^T A V = \Lambda^T \Lambda$. Fazendo $W = A V$, tem-se $W^T W = \Lambda^T \Lambda$, sendo $W = [w_1, w_2, \dots, w_r, \dots, w_m]$. Como $W^T W = \Lambda^T \Lambda$, conclui-se que:

$$w_i^T w_i = \begin{cases} \lambda_i & \text{se } i \leq r \\ 0 & \text{se } i > r \end{cases}$$

e $w_i^T w_j = 0$ se $i \neq j$ com $j = 1, 2, \dots, m$.

Se $w_i = 0$ para $i > r$, as primeiras colunas de W são linearmente independentes. Logo, conclui-se que $r \leq n$. Agora definindo

$$u_i = \frac{1}{\sqrt{\lambda_i}} w_i = \frac{1}{\sqrt{\lambda_i}} A v_i, \quad i = 1, 2, \dots, r,$$

então tomando $u_{r+1}, u_{r+2}, \dots, u_m$, se $r < n$, ortogonais entre si e aos demais u_i 's tais que a matriz $n \times n$:

$$U = [u_1, u_2, \dots, u_n],$$

seja ortogonal. Assim, pela definição de u_i , vê-se que a expressão $u_i \sqrt{\lambda_i} = A v_i$ pode ser expressa matricialmente por $U \Lambda = A V$. Como V é ortogonal, tem-se que $V^{-1} = V^T$, o que resulta em:

$$A V = U \Lambda$$

o que implica que

$$A = U \Lambda V^T.$$

Considerando apenas as r primeiras linhas de U , as r primeiras linhas e colunas de Λ e as r primeiras colunas de V , o resultado ainda continua válido. O que finaliza a demonstração.

Observação: Os elementos diagonais da matriz Λ são denominados valores singulares da matriz A . Usualmente os valores singulares são listados em ordem decrescente. Assim, a matriz diagonal Λ é unicamente determinada para A , entretanto as matrizes U e V não são unicamente determinadas, consideraremos que na decomposição SVD os elementos diagonais da matriz Λ estão listados em ordem decrescente.

Pode-se aplicar a SVD para aproximar uma matriz retangular A de posto r utilizando uma matriz retangular \tilde{A} de posto $k \leq r$. Considera-se o problema de minimizar a norma de Frobenius da diferença entre A e \tilde{A} para a qual exige-se que o posto de \tilde{A} seja igual a k (GOLUB; VAN LOAN, 1996; STEWART, 1993). Desta forma, pode-se enunciar o chamado Teorema de Eckart–Young (ECKART; YOUNG, 1936).

Teorema 2 (Eckart–Young): A matriz \tilde{A} é obtida pela fatoração SVD da matriz A , ou seja,

$$\tilde{A} = U \tilde{\Lambda} V^T = \sum_{i=1}^k \sigma_i u_i v_i, \quad (1)$$

em que u_i e v_i denotam a i -ésima coluna de U e V , respectivamente; $\tilde{\Lambda}$ coincide com Λ nos k maiores valores singulares os demais valores singulares são substituídos por zero na diagonal de $\tilde{\Lambda}$.

Demonstração:

Primeiramente, tem-se que:

$$\|A - A_k\|_F^2 = \left\| \sum_{i=k+1}^n \sigma_i u_i v_i^T \right\|_F^2 = \sum_{i=k+1}^n \sigma_i^2.$$

Portanto, é necessário mostrar que se $B_k = X Y^T$, onde X e Y tem k colunas, então

$$\|A - A_k\|_F^2 = \sum_{i=k+1}^n \sigma_i^2 \leq \|A - B_k\|_F^2.$$

Pela desigualdade triangular com a norma espectral, se $A = A' + A''$, então $\sigma_1(A) \leq \sigma_1(A') + \sigma_1(A'')$. Suponha que A'_k e A''_k denotem, respectivamente, o posto k para a aproximação de A' e A'' pelo método SVD. Então, para quaisquer $i, j \geq 1$, tem-se:

$$\begin{aligned} \sigma_i(A') + \sigma_j(A'') &= \sigma_1(A' - A'_{i-1}) + \sigma_1(A'' - A''_{j-1}) \\ &\geq \sigma_1(A - A'_{i-1} - A''_{j-1}) \\ &\geq \sigma_1(A - A_{i+j-2}) \quad \left(\text{pois posto}(A'_{i-1} + A''_{j-1}) \leq \text{posto}(A_{i+j-2}) \right) \\ &= \sigma_{i+j-1}(A). \end{aligned}$$

Desde que $\sigma_{k+1}(B_k) = 0$, quando $A' = A - B_k$ e $A'' = B_k$ conclui-se que, para $i \geq 1$ e $j = k + 1$,

$$\sigma_i(A - B_k) \geq \sigma_{k+i}(A).$$

Portanto,

$$\|A - B_k\|_F^2 = \sum_{i=1}^n \sigma_i(A - B_k)^2 \geq \sum_{i=k+1}^n \sigma_i(A)^2 = \|A - A_k\|_F^2,$$

como se queria demonstrar.

2.2 Exemplo numérico

A seguir, é apresentado um exemplo numérico da SVD, para um melhor entendimento de seus cálculos e para que seja possível converter uma imagem digital em uma matriz para o desenvolvimento das aplicações computacionais.

Exemplo 3 A matriz A , dada a seguir, será decomposta na forma $A = U\Lambda V^T$

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Resolução: Sabendo que a matriz A^T é da forma:

$$\mathbf{A}^T = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix},$$

segue que o produto $A^T A$ é tal que:

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix},$$

com autovalores distintos $\lambda_1 = 7$ e $\lambda_2 = 2$ e seus respectivos autovetores $w_1 = (\frac{1}{2}; 1)$ e $w_2 = (-2; 1)$. Logo, pelo Teorema da Decomposição em Valores Singulares tem-se que:

$$\mathbf{\Lambda} = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{bmatrix} = \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{2} \end{bmatrix}.$$

Normalizando os autovetores w_1 e w_2 , obtêm-se que:

$$v_1 = \left(\frac{w_1}{|w_1|} \right) = \left(\frac{\frac{1}{2}}{\sqrt{(\frac{1}{2})^2 + 1^2}}; \frac{1}{\sqrt{(\frac{1}{2})^2 + 1^2}} \right) = \left(\frac{\sqrt{5}}{5}; \frac{2\sqrt{5}}{5} \right)$$

$$v_2 = \left(\frac{w_2}{|w_2|} \right) = \left(\frac{-2}{\sqrt{(-2)^2 + 1^2}}; \frac{1}{\sqrt{(-2)^2 + 1^2}} \right) = \left(\frac{-2\sqrt{5}}{5}; \frac{\sqrt{5}}{5} \right).$$

Segue que:

$$\mathbf{V} = [v_1 \quad v_2] = \begin{bmatrix} \frac{\sqrt{5}}{5} & \frac{-2\sqrt{5}}{5} \\ \frac{2\sqrt{5}}{5} & \frac{\sqrt{5}}{5} \end{bmatrix}$$

e

$$u_1 = \frac{1}{\sqrt{\lambda_1}} Av_1 = \frac{1}{\sqrt{7}} \cdot \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{5}}{5} \\ \frac{2\sqrt{5}}{5} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{35}}{7} \\ \frac{3\sqrt{35}}{35} \\ \frac{-\sqrt{35}}{35} \end{bmatrix}$$

$$u_2 = \frac{1}{\sqrt{\lambda_2}} Av_2 = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \frac{-2\sqrt{5}}{5} \\ \frac{\sqrt{5}}{5} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{-\sqrt{10}}{10} \\ \frac{-3\sqrt{10}}{10} \end{bmatrix}.$$

Então:

$$\mathbf{U} = [u_1 \quad u_2] = \begin{bmatrix} \frac{\sqrt{35}}{7} & 0 \\ \frac{3\sqrt{35}}{35} & \frac{-\sqrt{10}}{10} \\ \frac{-\sqrt{35}}{35} & \frac{-3\sqrt{10}}{10} \end{bmatrix}.$$

Portanto:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T = \begin{bmatrix} \frac{\sqrt{35}}{7} & 0 \\ \frac{3\sqrt{35}}{35} & \frac{-\sqrt{10}}{10} \\ \frac{-\sqrt{35}}{35} & \frac{-3\sqrt{10}}{10} \end{bmatrix} \cdot \begin{bmatrix} \sqrt{7} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} \frac{\sqrt{5}}{5} & \frac{-2\sqrt{5}}{5} \\ \frac{2\sqrt{5}}{5} & \frac{\sqrt{5}}{5} \end{bmatrix}^T = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

3 Cor e imagem

Como este trabalho tem como objetivo reconstruir uma imagem de modo que ainda se possa visualizar a imagem original, porém com menor número de informação, como proposto no Teorema de



Eckart-Young, esta seção apresenta os conceitos de cor e imagem, fundamentais para a compreensão desse processo.

Na física, a cor é definida por uma radiação eletromagnética que varia entre 380 e 740, chamado de espectro visível (medida espectral) nanômetros. A energia que é liberada por essa onda e chamada de energia radiante (HELERBROCK, 2021).

Os físicos Young e Helmholtz desenvolveram, no século XIX, um modelo de percepção de cor, conhecido como RGB (*Red, Green e Blue*), que consiste nas iniciais das cores “vermelho, verde e azul.” Essas são as cores captadas pelo olho humano através de células fotosensíveis. A cor vermelha em amostras de baixa frequência, a cor verde em amostras de média frequência e a cor azul em amostras de alta frequência. Dessa forma o sistema RGB pode ser representado pelo \mathbb{R}^3 (GOMES; VELHO, 2008), como uma matriz. Uma imagem colorida é composta por três matrizes, sendo a primeira com os valores reais associados a cor vermelha, a segunda aos valores associados a cor verde, e a terceira, a cor azul. Cada entrada é chamada de pixel. Portanto, define-se (GOMES; VELHO, 2008):

Definição 4 A matriz A definida como $A(m, n, p)$ é uma matriz com m linhas, n colunas e p camadas. Se a imagem for cor de cinza, $p = 1$ e se a imagem for colorida, $p = 3$.

Como os computadores armazenam as informações em *bits*, que podem ser 0 ou 1, o agrupamento de 8 *bits* é feito, formando o *byte*. Assim, o *byte* pode armazenar $2^8 = 256$ valores diferentes. Portanto, para representar uma imagem em preto e branco (escala de cinza), basta associar cada pixel a um valor numérico que tem o valor 0 associado à cor preta e o valor 255 à cor branca. Os valores intermediários são tons de cinza. As imagens coloridas tem cada pixel associado a três componentes, que são as cores vermelha, verde e azul (RGB). Assim, a cor branca em uma imagem colorida tem as componentes $RGB = (255, 255, 255)$ e a cor preta tem $RGB = (0, 0, 0)$.

4 Aplicação para compressão de imagens

Nessa seção, são apresentadas aplicações da SVD, conforme descrito na seção anterior, mostrando exemplo de como decompor a imagem em forma de matriz, comprimir e reconstruir uma imagem, utilizando a Decomposição em Valores Singulares, utilizando MATLAB. É possível reconstruir uma imagem utilizando menos informação do que a imagem original, necessitando de menos armazenamento e tempo de transmissão da imagem.

A reconstrução da imagem é feita utilizando o Teorema 2. Considera-se que os valores singulares estejam ordenados do maior para o menor. A imagem reconstruída é então composta por uma quantidade menor de parcelas do somatório da Equação (1) (valores singulares), desta forma com menos informações do que a imagem original, devido à decomposição.

É importante ressaltar que o MATLAB possui uma função chamada ‘*svd(A)*’, que retorna um vetor contendo os valores singulares da matriz A . Porém, neste trabalho, foi desenvolvida a função ‘*decomposicao(A)*’, descrita a seguir, que calcula todos os passos da SVD e retorna as matrizes U , Λ e V do Teorema 1.

```
function [U,S,V] = decomposicao( A )
```

```
B=A'*A;
```

```
[A1, D]=eig(B,'nobalance');
```

```
%construção da matriz S
```



```
d=diag(D);
m=length(d);
for i=1:m
    if d(i,1)<exp(-10)
        d(i,1)=0;
    end
end
d=sort(d,'descend');
d=sqrt(d);
[x,y]=size(A);
if x<y
    for i=1:x+1
        if d(x+1,1)==0
            d(x+1,:)=[];
        end
        h=length(d);
        if h==x
            break
        end
    end
end
D=diag(d);
[p,q]=size(D);
if p<x
    S=[D;zeros(x-p,y)];
elseif q<y
    S=[D zeros(x,y-q)];
else
    S=D;
end
%Construção da matriz V
X=orth(B); %base ortonormal do espaço coluna de B
Y=null(A); %base ortonormal do espaço nulo de A
V=[X Y];
%Construção da matriz U a partir da V
if x<=y
    for i=1:x
        U(:,i)=D(i,i)^(-1)*A*V(:,i);
    end
else
    for i=1:y
        W(:,i)=D(i,i)^(-1)*A*V(:,i);
    end
    Y=null(A');
    U=[W Y];
end
end
end
```



A função ‘decomposicao(A)’ é então utilizada nas funções ‘comppeb’ e ‘compcolor,’ descritas a seguir. A função ‘comppeb’ consiste em comprimir uma imagem e exibi-la em preto e branco (escalas de cinza), utilizando apenas uma matriz de mesma dimensão (em pixels) da figura original. O comando ‘imread’ lê a imagem (colorida ou em escalas de cinza), cujo nome deve ser inserido entre aspas simples. Em seguida, o comando ‘rgb2gray’ converte a imagem do sistema RGB (vermelho, verde e azul) para uma imagem em escalas de cinza. E seguida, a SVD é calculada utilizando-se a função ‘decomposicao,’ de acordo com o Teorema 1, e a imagem é reconstruída de acordo com o Teorema 2.

```
function [ I ] = comppeb( ~ )
A=imread(input('Insira o nome da imagem (entre apóstrofes): '));
B=rgb2gray(A);
C=im2double(B);
[U S V]=decomposicao(C);
k=input('Insira o valor de k que define a quantidade de valores
singulares a serem considerados na Equação (1): ');
A=0;
V1=V';
for i=1:k;
    A=A+S(i,i)*U(:,i)*V1(i,:);
end
I=imshow(A);
end
```

A função ‘compcolor’ consiste em comprimir uma imagem e exibi-la de forma colorida, no sistema RGB (vermelho, verde e azul), utilizando 3 matrizes de mesma dimensão (em pixels) da figura original. O comando ‘imread’ também é utilizado, assim como a função ‘decomposicao’ para a execução da SVD para cada uma das 3 matrizes, e a imagem é também reconstruída de acordo com o Teorema 2, utilizando-se as 3 matrizes.

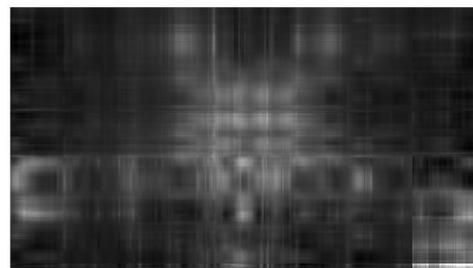
```
function [ I ] = compcolor( ~ )
A=imread(input('Insira o nome da imagem (entre apóstrofes): '));
B=im2double(A);
B1=B(:,:,1);
B2=B(:,:,2);
B3=B(:,:,3);
[m n]=size(B1)
[U1 S1 V1]=decomposicao(B1);
[U2 S2 V2]=decomposicao(B2);
[U3 S3 V3]=decomposicao(B3);
k=input('Insira o valor de k que define a quantidade de valores
singulares a serem considerados na Equação (1): ');
A1=0;
A2=0;
A3=0;
```

```
for i=1:k;
    A1=A1+S1(i,i)*U1(:,i)*V1(:,i)';
    A2=A2+S2(i,i)*U2(:,i)*V2(:,i)';
    A3=A3+S3(i,i)*U3(:,i)*V3(:,i)';
end
C(:,:,1)=A1;
C(:,:,2)=A2;
C(:,:,3)=A3;
I=imshow(C);
end
```

A Figura 1(a) apresenta uma imagem original com 720×1280 pixels. Essa imagem é colorida, porém, como a função 'comppeb' será aplicada inicialmente, a imagem será convertida em escalas de cinza. Por isso, o comando 'rgb2gray' é executado inicialmente. Desta forma, a imagem é modelada matematicamente por uma única matriz de dimensão: 720×1280 . Dessa forma, seria necessário armazenar 720×1280 pixels = 921.600 números reais. A Figura 1 exibe algumas comparações a partir das imagens reconstruídas utilizando uma quantidade pequena de valores singulares, ou seja, uma quantidade menor de informações a serem armazenadas quando comparada à imagem original.



(a) Imagem original



(b) Imagem com k=5 valores singulares



(c) Imagem com k=50 valores singulares



(d) Imagem com k=100 valores singulares

Figura 1: Aproximação da imagem via SVD

Utilizando $k = 5$ valores singulares, conforme ilustra a Figura 1(b), tem-se, de acordo com o Teorema 2, $720 \times 5 + 1280 \times 5 + 5 = 10.005$ números reais a serem armazenados para apenas 1 canal de cor (escalas de cinza), que representa 1,09% do armazenamento original. Porém, a imagem ainda não está bem visível, devido ao baixo número de valores singulares considerado. Com $k = 50$ valores singulares, obtém-se uma imagem mais nítida, com dimensão $720 \times 50 + 1280 \times 50 + 50 = 100.050$

números reais a serem armazenados para o canal, representando 10,86% do armazenamento original, conforme a Figura 1(c). Se forem utilizados 100 valores singulares, a imagem aparece como a imagem original, porém em escalas de cinza, com $(720 \times 100 + 1280 \times 100 + 100) = 200.100$ valores reais, ou seja, apenas 21,71 % das informações, como pode ser visto na Figura 1(d).

A Figura 2(a) apresenta uma imagem original com 800×600 pixels. Essa imagem colorida é então modelada matematicamente por três matrizes de mesma dimensão da figura: 800×600 pixels, sendo uma matriz para cada canal de cor do sistema RGB (vermelho, verde e azul).

Dessa forma, seriam necessários armazenar 800×600 pixels = 480.000 números reais para cada um dos três canais de cores, totalizando 1.440.000 números reais a serem armazenados. Algumas comparações são exibidas na Figura 2, mostrando as imagens reconstruídas, executando-se a função ‘compcolor,’ a partir de uma quantidade pequena de valores singulares ($k = 5$), que vão sendo acrescentados até obter uma imagem mais clara e limpa, porém com uma quantidade menor de informações (números reais a serem armazenados) quando comparada à imagem original.

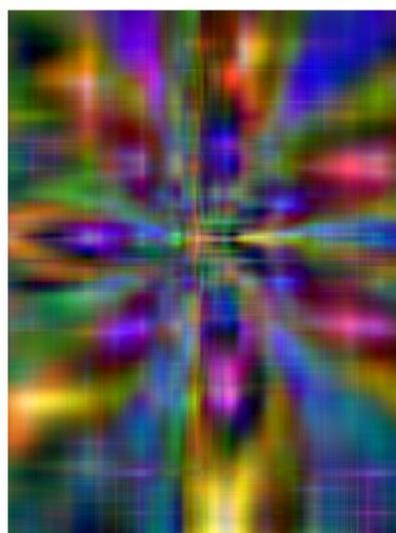
Com $k = 5$ valores singulares, conforme mostra a Figura 2(b), tem-se, de acordo com o Teorema 2, $800 \times 5 + 600 \times 5 + 5 = 7.005$ números reais a serem armazenados para cada um dos 3 canais de cores, ou seja: 21.015 valores, que representa 1,46% do armazenamento original. Porém, nota-se que a imagem ainda não está perceptível. Com $k = 50$ valores singulares, obtém-se uma imagem melhorada, com $800 \times 50 + 600 \times 50 + 50 = 70.050$ números reais a serem armazenados para cada canal, ou seja: 210.150 valores, representando 14,59% do armazenamento original, conforme ilustra a Figura 2(c). Se forem utilizados 100 valores singulares, a imagem se mostra praticamente a mesma imagem original, sendo que serão necessários armazenar $(800 \times 100 + 600 \times 100 + 100) \times 3 = 420.300$ valores reais, que representa apenas 29,19 % das informações, como pode ser visto na Figura 2(d).

5 Conclusão

A decomposição em valores singulares possui inúmeras aplicações em diversas áreas do conhecimento. Neste trabalho, foi apresentada a aplicação na compressão e reconstrução de imagens, que se mostrou possível por meio do Teorema de Eckart–Young (Teorema 2). Destaca-se que os programas descritos em MATLAB, na Seção 4, permitem a fácil reprodução dos exemplos numéricos apresentados na Seção 2. Nos exemplos de compressão de imagens apresentados, pode-se notar como o uso desta ferramenta baseada na SVD permite uma grande economia de memória de armazenamento de valores reais para envio e para a transmissão de imagens.



(a) Imagem original



(b) Imagem com $k=5$ valores singulares



(c) Imagem com $k=50$ valores singulares



(d) Imagem com $k=100$ valores singulares

Figura 2: Aproximação da imagem via SVD

6 Referências Bibliográficas

ANTON, H.; RORRES, C. **Álgebra linear com aplicações**. 10. ed. Porto Alegre: Bookman, 2012.



- DEMMELE, J. W. **Applied numerical linear algebra**. Philadelphia: Siam, 1997.
- ECKART, C.; YOUNG, G. The approximation of one matrix by another of lower rank. **Psychometrika**, v. 1, n. 3, p. 211–218, 1936.
- GOLUB, G. H.; HOFFMAN, A.; STEWART, G.W. A generalization of the Eckart-Young-Mirsky matrix approximation theorem. **Linear Algebra and its Applications**, v. 88/89, p. 317-327, 1987.
- GOLUB, G. H.; VAN LOAN, C. F. **Matrix computations**. 3rd. ed. Baltimore: Johns Hopkins Univeisty Press, 1996.
- GOMES, J.; VELHO, L. **Fundamentos da computação gráfica**. Rio de Janeiro: IMPA, 2008.
- HELERBROCK, R. **Espectro eletromagnético**. Brasil Escola. Goiânia: Rede Omnia, 2021. Disponível em: <https://brasilecola.uol.com.br/fisica/espectro-eletromagnetico.htm>. Acesso em: 12 out. 2021.
- LEON, S. J. **Álgebra linear com aplicações**. 9. ed. Rio de Janeiro: LTC, 2018.
- MOUNIKA, K.; LAKSHMI, D. S. N.; ALEKYA, K. SVD based image compression. **International Journal of Engineering Research and General Science**, v. 3, n. 2, p. 1271–1278, 2015.
- SADEK, R. A. SVD based image processing applications: state of the art, contributions and research challenges. **International Journal of Advanced Computer Science and Applications**, v. 3, n. 7, p. 26-34, 2012.
- STEINBRUCH, A.; WINTERLE, P. **Álgebra linear**. 2. ed. Rio Grande do Sul: Pearson, 1995.
- STEWART, G. W. On the early history of the singular value decomposition. **SIAM Review**, v. 35, n. 4, p. 551–566, 1993.