

# Relatório Trabalho 03 - Compressão de imagens digitais pela técnica SVD

Nome: Bleno Humberto Claus

Ra: 145444

## Sumário

Introdução	1
Problema	1
Metodologia e Implementação	2
Limitações	5
Conclusão	5

## Introdução

Foi realizado um estudo sobre compressão de imagens digitais pela técnica, *Singular Value Decomposition* - SVD. Foi parametrizado o número de componentes para se verificar seu efeito na qualidade e tamanho em relação o tanto de memória utilizada. Foi utilizado como ferramenta a linguagem de programação *python 2.7* através das bibliotecas *numpy*, *matplotlib* e *scipy* e toda implementação em ambiente linux, distribuição *ubuntu 14.12*.

## Problema

Dado uma matriz  $A_{n \times p}$ , a aplicação da técnica SVD se resume a fatoraçoão de  $A$ , em uma matriz unitária  $U_{n \times n}$ , uma matriz diagonal  $\Sigma_{n \times p}$  com números reais não negativos e outra matriz unitária real  $V_{p \times p}$ .

$$(1) A = U \Sigma V^T$$

Encontrar tais matriz demanda certa complexidades, então neste trabalho se utilizou uma facilidade da biblioteca *Numpy* que possui uma implementação através do método *numpy.linalg.svd*, que dado uma matriz  $A$ , retorna três matrizes  $U$ ,  $S$  e  $V$ , tais que:

$$(2) A = U * diagonal(S) * V$$

Note que a matriz  $V$  retornada pelo método em (2), já a transposta da matriz  $V$  explicada na equação (1).

## Metodologia e Implementação

A implementação se baseou na aplicação da técnica SVD para os três canais RGB da imagem colorida. Foi também parametrizado as componentes das matrizes para determinar a qualidade e o uso de memória da imagem resultante comprimida. O algoritmo implementado pode ser observado logo abaixo:

---

**Algoritmo 1:** Compressão com Análise de Componentes Principais
 

---

```

input : Imagem  $f$  com dimensões  $M \times N$  pixels
          Número de componentes  $k$ 
output: Imagem  $g$  com dimensões  $M \times N$  pixels

1 # dividir a imagem RGB em três canais e aplicar a técnica SVD em cada canal
2 for  $i = 1 : 3$  do
3    $[U_f(:, :, i), S_f(:, :, i), V_f(:, :, i)] = \text{svd}(\text{double}(f(:, :, i)))$ 
4 # considerar apenas  $k$  componentes e combinar novamente os canais
5  $g = \text{zeros}(M, N)$ 
6 for  $i = 1 : 3$  do
7    $U_g(:, 1 : k, i) = U_f(:, 1 : k, i)$ 
8    $S_g(1 : k, 1 : k, i) = S_f(1 : k, 1 : k, i)$ 
9    $V_g(1 : k, :, i) = V_f(1 : k, :, i)^T$ 
10   $g(:, :, i) = U_g(:, :, i) * S_g(:, :, i) * V_g(:, :, i)$ 
11 return  $g$ 
  
```

Figura 1. Algoritmo de compressão

Foi também, para fins avaliativos, calculados duas grandezas, a taxa de compressão  $\rho$  e a raiz do erro médio quadrático RMSE.

$$\rho = \frac{\text{quantidade de memória requerida para representar } g}{\text{quantidade de memória requerida para representar } f}$$

$$\text{RMSE} = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2}$$

Figura 2. taxa de compressão e erro médio quadrático

Foi notado que a medida que quanto menor o número de componentes  $k$ , menor a qualidade da imagem, e seu RMSE tende a ser maior. Quanto menor a qualidade da imagem comprimida, menor quantidade de memória requerida, logo  $\rho$  tende a diminuir. Abaixo será apresentado alguns gráficos representando os valores de RMSE e  $\rho$  para diferentes  $k$ .

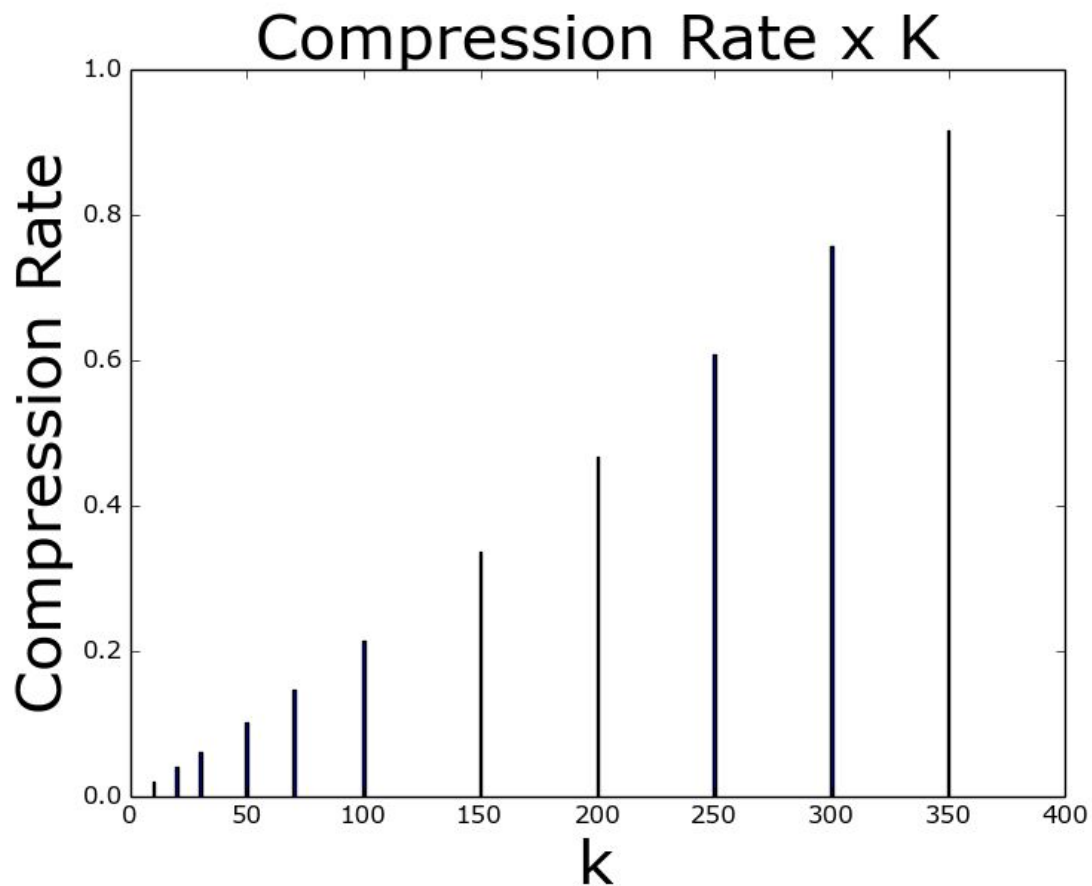


Figura 3. Variação da taxa de compressão em função do número de componentes  $k$

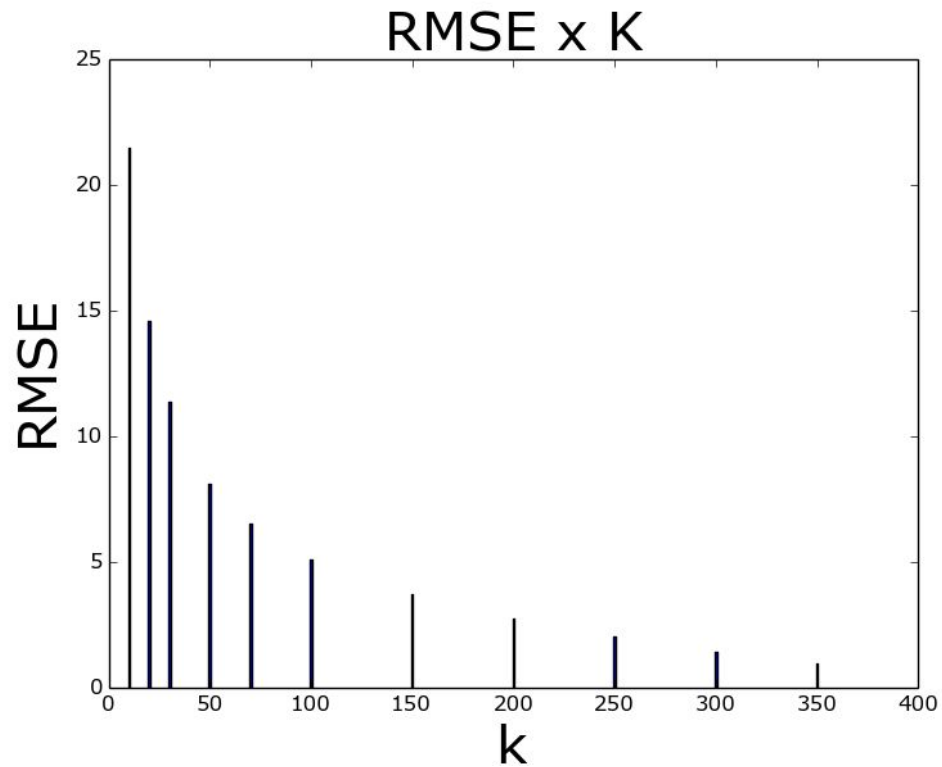


Figura 4. Variação da raiz do erro médio quadrático em função do número de componentes  $k$

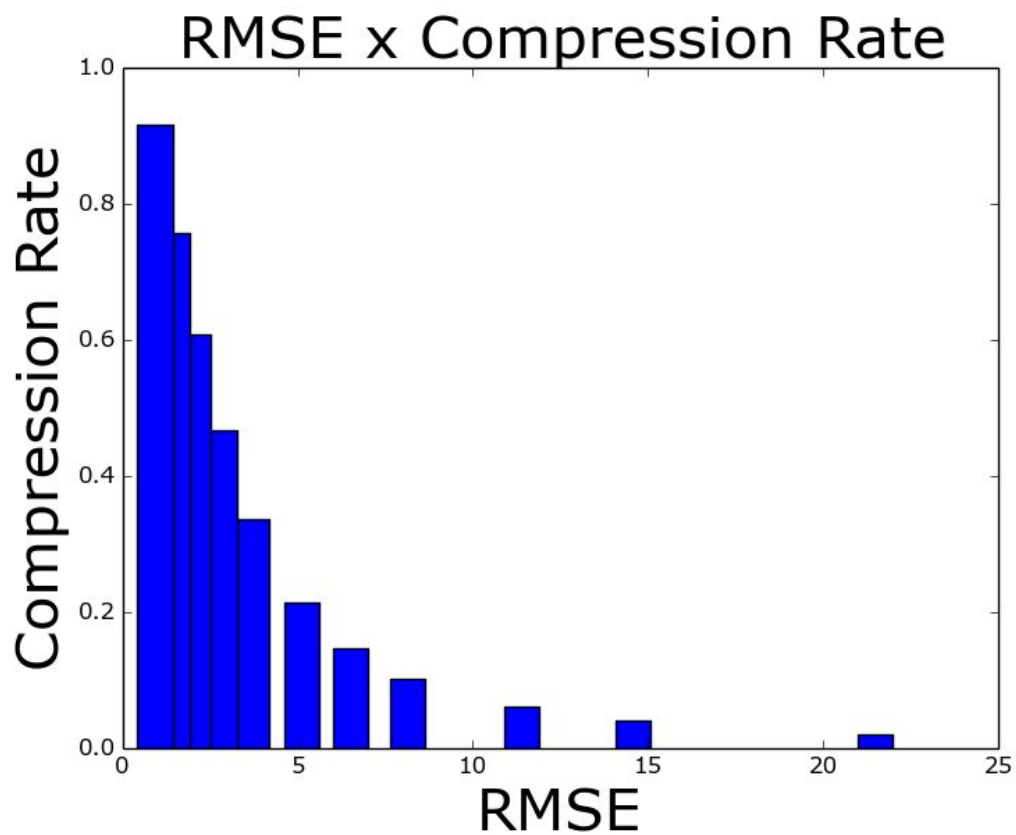


Figura 5. Variação da taxa de compressão em função da raiz do erro médio quadrático

## **Limitações**

Na implementação da taxa de compressão, foi usado o número de bytes utilizados na imagem comprimida sobre o número de bytes da original, ao invés do espaço de memória requerida. Pois é razoável assumir que um Byte assumirá uma quantidade memória constante em um mesmo sistema operacional, e assumindo que isso seja verdade, ao se calcular a taxa, este valor se corta, e uma boa aproximação pode ser usada calculando a razão do número de bytes utilizados.

## **Conclusão**

Foi possível implementar uma técnica de compressão com bastante facilidade e verificar na prática a balança entre qualidade e espaço de armazenamento. Também se observou pela figura 5 que para qualidades maiores, a taxa de compressão foi perto de 1, o que significa um uso de espaço perto do usado pela imagem original.