

CÓDIGO ANEXO

# Breve comparação entre os métodos Singular Value Decomposition (SVD) e Discrete Cosine Transform (DCT) para compressão de imagens

Gisele G. Tavares<sup>1</sup>, Ruan M. Carvalho<sup>1</sup>

<sup>1</sup>Programa de Pós-graduação em Modelagem Computacional -  
Universidade Federal de Juiz de Fora (PGMC/UFJF)

Trabalho apresentado para a disciplina de Álgebra Linear Computacional - SET/2018

giselegoulart@ice.ufjf.br, ruan.medina@engenharia.ufjf.br

**Resumo.** Este trabalho se propõe em estudar a compressão de imagens com a utilização dos métodos Singular Value Decomposition (SVD) e Discrete Cosine Transform (DCT). Os métodos são comparados por meio da observação de métricas de demonstram suas performances na compressão de imagens em preto e branco. É proposta uma forma automática de obter o número de posições necessárias para gerar compressões sem perdas excessivas de informação utilizando SVD. Os resultados demonstram que o DCT apresenta, na maioria dos testes, maior capacidade de reduzir o uso de memória na compressão em comparação com o SVD, para um dado erro fixado. ☹

Bom tema!!

## 1. Introdução

A demanda por aplicações que consigam extrair conhecimento de dados diversos vêm crescendo nas últimas décadas. Neste sentido, o desenvolvimento e aplicação de métodos da Álgebra Computacional alimentam áreas como a Ciência de Dados, provendo o ferramental para organizar a quantidade de informação presente nos dados coletados [Sarkar 2018]. Este trabalho trata especificamente da compressão de imagens, um problema diretamente ligado a questões de armazenamento e transferência de dados, porém, os métodos aqui estudados podem ser estendidos para compressão de dados em geral, processamento de sinais ou aplicações estatísticas ~~em dados~~. De fato, imagens são matrizes de informação, assim como outras tabelas de dados para uma outra aplicação qualquer. Como vantagem de aplicar os estudos em imagens, tem-se a possibilidade de visualização direta e intuitiva dos resultados dos dados tratados, porém sem a necessidade de se limitar a isso. 7.1

Objetiva-se estudar a compressão de imagens com a utilização dos métodos Singular Value Decomposition (SVD) - do português 'Decomposição em Valores Singulares' - e Discrete Cosine Transform (DCT) - do português 'Transformada Discreta de Cossenos'. Os métodos são comparados por meio da observação de métricas de demonstram suas performances na compressão de imagens em preto e branco. Ademais, propõe-se uma forma automática de gerar compressões sem perdas excessivas de informação.

## 2. Métodos

Nesta seção apresenta-se brevemente os métodos utilizados nas compressões das imagens e a justificativa da aplicação dos mesmos neste problema. Descreve-se ainda as bibliotecas utilizadas nas implementações computacionais e o banco de imagens utilizado para os experimentos.

Texto: bom

- as vezes um pouco informal demais

- tomar cuidado c/ notação e definições

Resultados - bons!

## 2.1. SVD

A Decomposição em Valores Singulares, [Stewart 1993] e [Chen 2000], é uma possível fatoração de uma matriz  $M$ ,  $m \times n$ , real ou complexa em três matrizes como descrito em (1), onde  $U$  é uma matriz  $m \times m$  ortogonal se real (ou unitária se complexa),  $\Sigma$  é uma matriz retangular diagonal  $m \times n$  com números reais não-negativos na diagonal, e  $V^*$  (a matriz adjunta ou conjugada transposta de  $V$ ) é uma matriz  $n \times n$  ortogonal se real (ou unitária se complexa).

$$M_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^* \quad (1)$$

São denominados valores singulares de  $M$  os valores na diagonal de  $\Sigma$ . As  $m$  colunas de  $U$  e as  $n$  colunas de  $V$  são os chamados vetores singulares à esquerda e vetores singulares à direita de  $M$ , respetivamente. Nessa construção, os valores singulares são ordenados do maior para o menor valor na diagonal principal de  $\Sigma$ , ou seja,  $\sigma_1 > \sigma_2 > \dots > \sigma_t$ ;  $t = \min\{n, m\}$ .

A utilização da decomposição SVD no problema da compressão de imagem surge da possibilidade de aproximar a matriz  $M$  por uma matriz  $M'$  formada pela operação matricial de multiplicação entre  $U'$ ,  $\Sigma'$  e  $V'^*$  com ranks menores que os das matrizes que fatoram a matriz original. Além disso, poucos dos valores singulares da decomposição tem valores relativamente altos. Por isso, mesmo que as imagens originais tenham rank completo, muitas dimensões que compõem esse rank não agregam muita informação da imagem [Sadek 2012]. Com isso, em uma compressão, representa-se a matriz  $M$  como uma combinação linear de matrizes de rank unitário e constrói-se uma matriz  $M'$ , aproximação de  $M$ , com  $r$  termos da combinação linear descrita, desconsiderando as posições de dimensões menos representativas, isto é

$$M' = \sigma_1 u_1 v_1^* + \sigma_2 u_2 v_2^* + \dots + \sigma_r u_r v_r^* \quad ; \quad r < \min\{n, m\} \quad (2)$$

## 2.2. DCT

A Transformada de Fourier é um mecanismo utilizado para representação de funções contínuas no domínio da frequência. Ela é formada por uma Transformada de Senos e uma Transformada de Cossenos definidas, respectivamente, pela parte imaginária e pela parte real da Transformada de Fourier [Easton 2010].

A Transformada Discreta de Cossenos (DCT) é a versão discreta da parte real da Transformada de Fourier. Esse processo é vastamente aplicado em compressões pois posiciona os coeficientes mais significativos no início da matriz transformada dos dados. Como os demais valores são muitos pequenos, carregam pouca informação da imagem e podem passar por um processo de eliminação [Newsletter 2014].

A transformada utilizada nesse trabalho é a versão bidimensional da DCT que pode ser observada em (3) para uma matriz  $M$ ,  $n \times n$ . A matriz dessa transformada é composta de vetores ortonormais, sendo por isso uma matriz ortogonal (de rotação). Assim, essa transformada pode ser considerada como uma rotação, ou, mais especificamente, duas rotações consecutivas, uma em cada dimensão.



$$M'_{i,j} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} M_{x,y} \cos\left(\frac{(2y+1)j\pi}{2n}\right) \cos\left(\frac{(2x+1)i\pi}{2n}\right); 0 \leq i, j \leq n-1$$



$$C'_f = \begin{cases} \frac{1}{\sqrt{2}} & , f = 0 \\ 1 & , f > 0 \end{cases}$$

O que implica em  
C.C. ??

(3)

pressão da mai  
detalhe do método!

### 2.3. Biblioteca e Descrição das Imagens

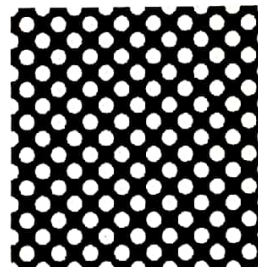
O código para geração dos resultados e comparação dos métodos foi desenvolvido em *Python*, a fim de diminuir a complexidade do algoritmo através da utilização de bibliotecas disponíveis na linguagem. Os principais pacotes utilizados foram o *PIL* (para a leitura da matriz dos pixels da imagem), o *fftpack* da biblioteca *Scipy* (para a aplicação da transformada DCT) e o *Numpy* (para realização dos cálculos com matrizes e decomposição SVD).

para manipulação  
de imagens

Para a realização dos testes, quatro imagens de tamanho  $256 \times 256$  foram dadas como entrada. O algoritmo realiza a leitura da imagem a partir de uma URL e disponibiliza a matriz dos pixels que será utilizada na transformada DCT e decomposição SVD.



(a) Barco



(b) Tabuleiro



(c) Ressonância  
magnética



(d) Lena

Figura 1. Imagens utilizadas nos testes de compressão.

### 3. Experimentos computacionais

Nesta seção, são apresentados os resultados da compressão de quatro imagens para dois métodos estudados. Nesse intuito é comparar suas performances de compressão para cada

O defeituoso

o desempenho

considerados  
para comparação dos

uma das imagens. Para isso, foi necessário propor métricas de erro da compressão como em [Easton 2010] e [Putalapattu 2017]. *apresentado*

Seja o erro em um pixel  $i$  dado por (4). Defina como Raiz do Erro Médio Quadrático (RMS Erro) a raiz da média do somatório dos erros quadráticos em todos os pixels da imagem, dado por (5). Essas métricas são utilizadas para comparar o distanciamento das apresentações visuais da imagem original e da imagem comprimida.

$$\epsilon_i = f(i) - \text{compressed}(f(i))c \quad (4)$$

*ERRO*

$$\text{RMS Erro} = \sqrt{\frac{1}{N} \sum_{i=1}^N \epsilon_i^2} ; N = \text{num\_pixels} \quad (5)$$

Com relação às compressões, define-se como Taxa de Compressão (cp\_ratio) a razão entre o número de posições de memória utilizados pela matriz original em comparação com a matriz após a redução, como em (6). Repare que a medida é aplicada sobre a representação computacional da imagem, que são diferentes após a fatoração SVD ou da transformação DCT. Essa medida não é aplicada diretamente sobre os pixels da imagem. Contudo, a redução do *rank* da matriz de representação das imagens implicará na redução dos detalhes na apresentação da imagem. Isso reflete no erro  $\epsilon$ .

*imagem  
n x m  
ou  
m x n??*

*Referência?*

*o que é rank?*

$$\text{svd\_cp\_ratio} = \frac{mn}{r(m+n+1)} \quad \text{dct\_cp\_ratio} = \frac{mn}{r^2} \quad (6)$$

*definir*

*SVD equivalente em relação aos métodos?*

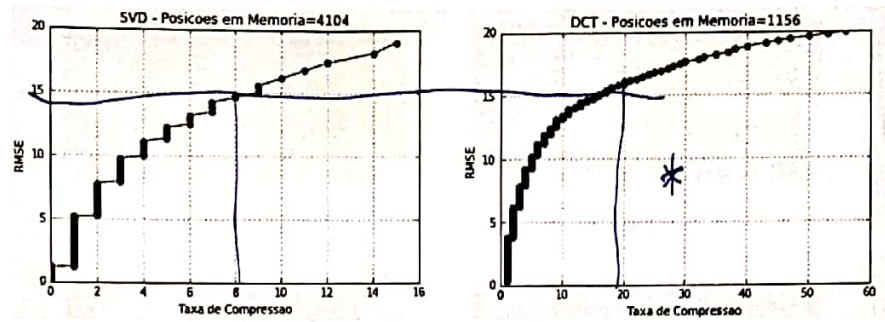
A Figura 2 apresenta o gráfico da relação entre a taxa de compressão e o erro médio quadrático para as diferentes imagens da Figura 1 quando comprimidas pelos métodos SVD e DCT. Para fácil comparação, fixou-se o RMS Erro máximo como 20. Percebe-se que o DCT atinge o teto de erro com uma taxa de compressão maior que do SVD e utilizando menos quantidade de posições em memória para isso (exceto em 2b).

*20  
ou  
20%?*

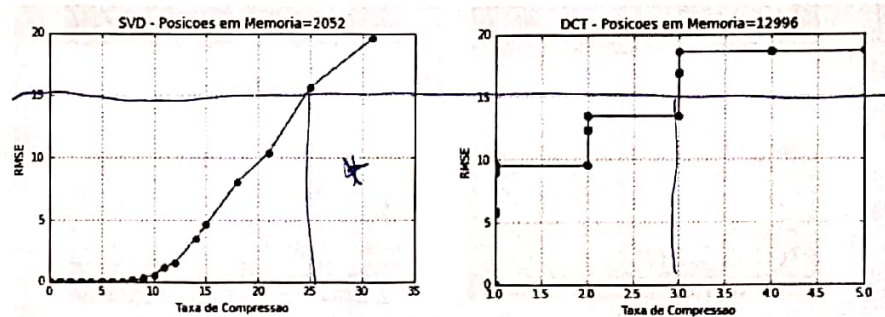
Repare que os métodos tem natureza diferenciada entre si. Mesmo com a redução de linhas e colunas na fatoração SVD, ao efetuar as operações matriciais, obtém-se uma matriz ( $M'$ ) com a mesma dimensão da matriz original. Então, com matrizes de *rank* ( $r$ ) iguais para a matriz  $\Sigma$  do SVD e para a matriz transformada pelo DCT, obtém-se diferentes apresentações da reconstrução das imagens testadas. Tenha em mente que para um dado  $r$ , o SVD representa melhor as imagens, porém utilizando maior número de posições na memória. Isso pode ser observado nas Figuras 3, 4, 5 e 6.

Como já descrito na seção anterior, tanto o método SVD quanto o DCT, concentram a maior parte da descrição da imagem nas primeiras posições das matrizes transformadas (DCT) ou após o processo de fatoração (SVD). Esta tendência pode ser observada comparando as Figuras 3, 4, 5 e 6 com os gráficos das Figuras 7 e 8, uma vez que mesmo com o aumento da taxa de compressão, ou seja, com a diminuição do *rank* da matriz de representação da imagem, a mesma continua sendo reconstruída com detalhes consideráveis.

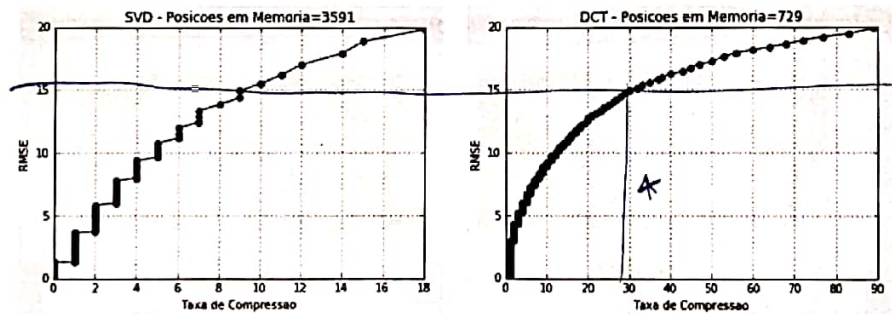




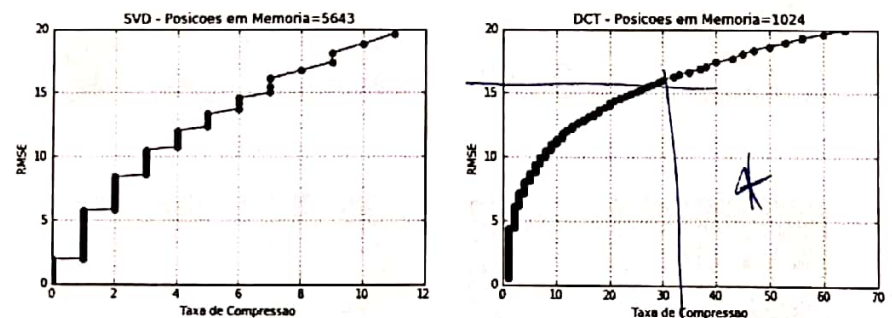
(a) Barco



(b) Tabuleiro



(c) Ressonância



(d) Lena

Figura 2. Relação entre cp\_ratio e RMS\_Erro em compressões de diferentes imagens pelos métodos SVD e DCT.

{ porque  
ocorre isto?  
Figura mais  
uniforme? }

Qual  
m Feb 72

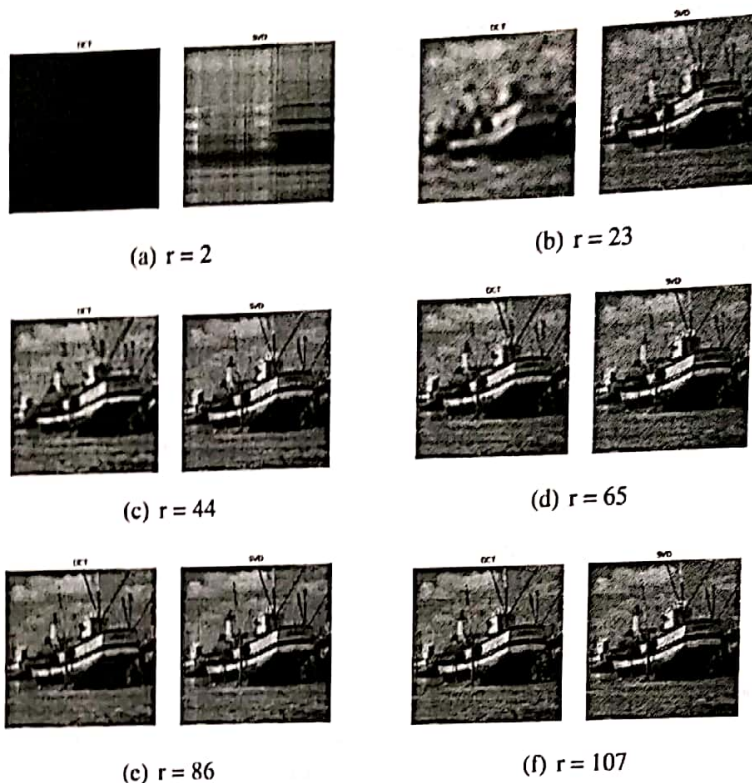


Figura 3. Relação entre cp\_ratio e reconstrução de imagens comprimidas (a).

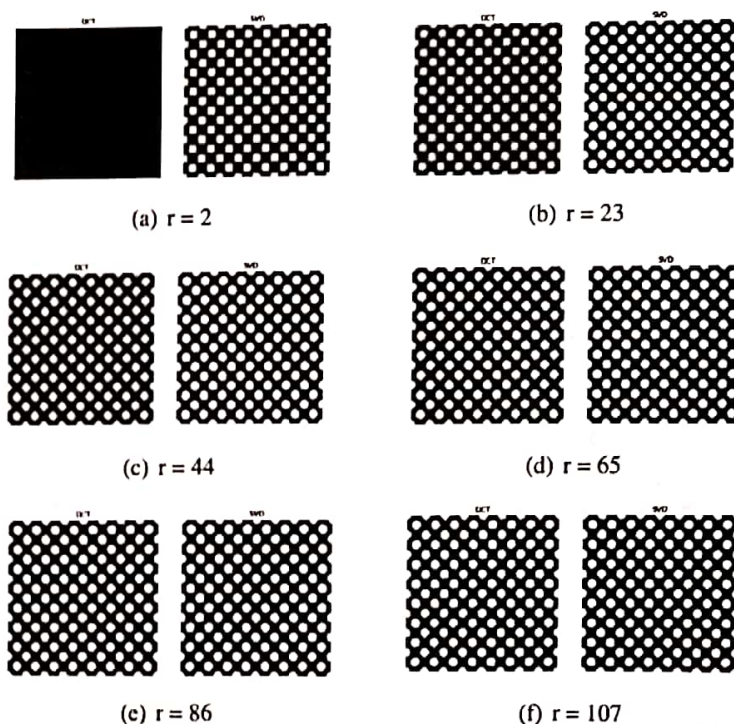


Figura 4. Relação entre cp\_ratio e reconstrução de imagens comprimidas (b).

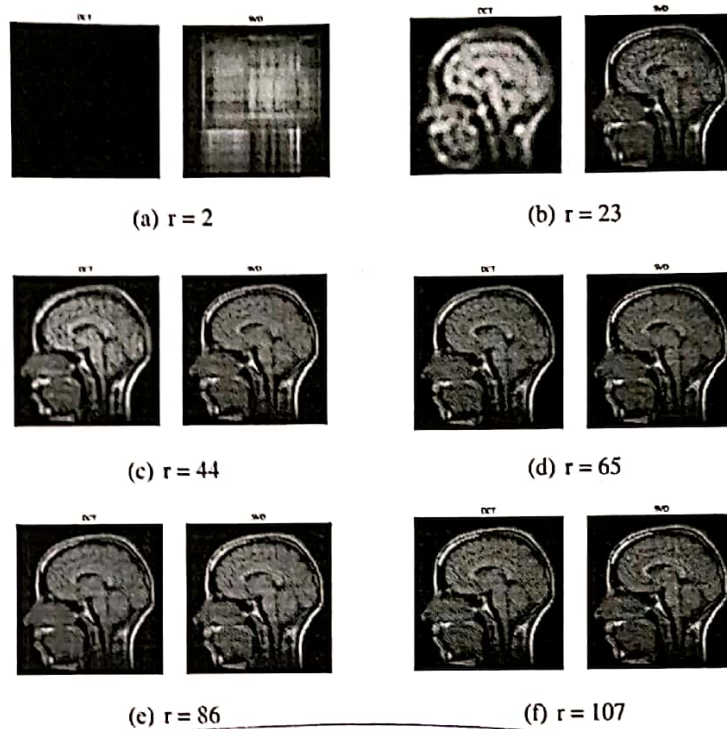


Figura 5. Relação entre cp.ratio e reconstrução de imagens comprimidas (c).

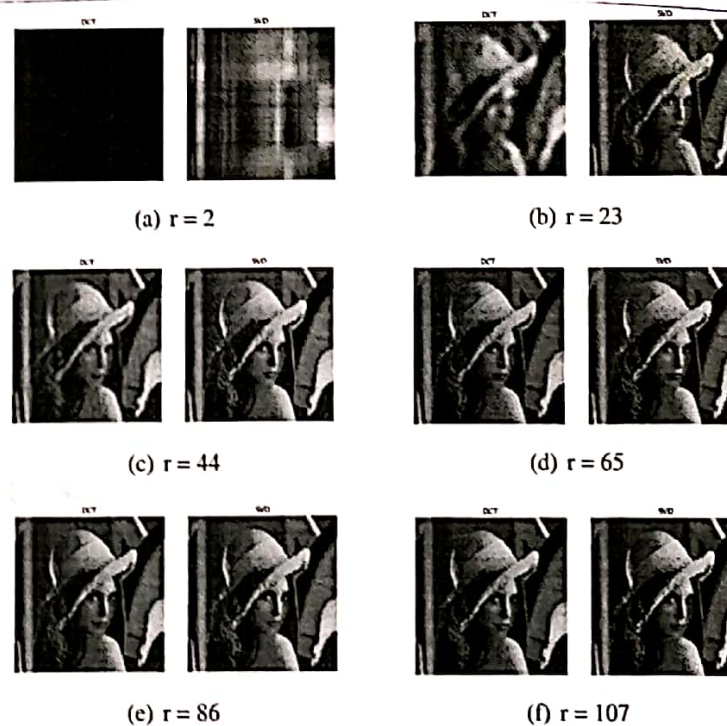
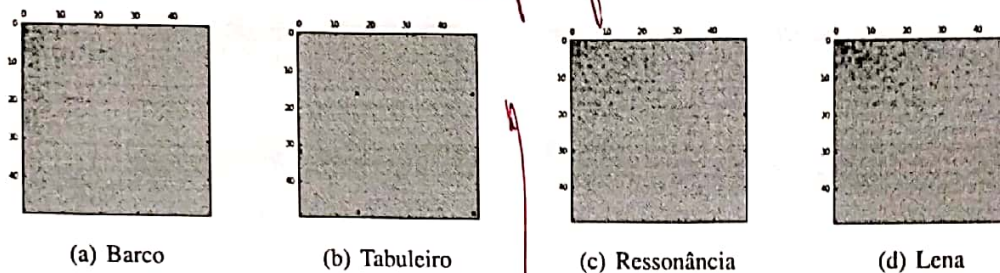


Figura 6. Relação entre cp.ratio e reconstrução de imagens comprimidas (d).





**Figura 7. Grid com as intensidades dos primeiros 2500 coeficientes da transformada DCT. Intensidades dos valores na matriz  $M'$ .**

### 3.1. Quantidade de Informação Representada

Essa disposição dos valores nos direciona à ideia de que poucos *ranks* são suficientes para representar altas porcentagem de detalhes de uma imagem. Contudo, para uma imagem qualquer, qual o número de dimensões necessárias para comprimi-la sem que haja grande perda de informação? Neste caso, considerar um *rank* ( $k$ ) padrão para a compressão não é um parâmetro justo entre diferentes imagens, uma vez que não leva em consideração a quantidade de informação presente em cada uma delas. Mas como estimar a quantidade de informação de uma imagem? Considere a matriz  $\Sigma$  de uma decomposição SVD. O somatório dos valores singulares nos dá uma boa pista da especificidade daquela imagem e de quanta informação ela carrega, como em (7).

$$\text{Quantidade de Informação} = Q_{\text{info}} = \sum_{i=1}^{\min\{n,m\}} \sigma_i \quad (7)$$

Com isso é possível definir um percentual da quantidade de informação que ~~queremos em nossa~~ <sup>deu 9.20 na</sup> compressão e encontrar o número de dimensões necessárias para tal representação [Chen 2000]. Por isso, o percentual de quantidade de informação representada, descrito em (8), é uma grandeza relativa para cada imagem que aproxima o percentual da quantidade de informação da imagem original que está sendo reproduzida na imagem comprimida. A soma dos valores singulares utilizados na compressão nos indica quanto de informação foi projetada no espaço reduzido da compressão. O percentual desse valor em relação à quantidade de informação da imagem original é a porcentagem de informação que conseguiu ser representada na compressão.

$$\%Q_{\text{info}} = \frac{Q_{\text{info}}(\Sigma')}{Q_{\text{info}}(\Sigma)} = \frac{\sum_{i=1}^r \sigma_i}{Q_{\text{info}}(\Sigma)} \quad (8)$$

O número de valores singulares ( $r$ ) necessários para conseguir o percentual de informação desejado na compressão pode ser encontrado facilmente somando gradativamente os valores singulares obtidos na fatoração SVD. A acumulada deste somatório em relação a dimensão da compressão é apresentada na Figura 8.

??  
para todas as  
imagens consideradas.



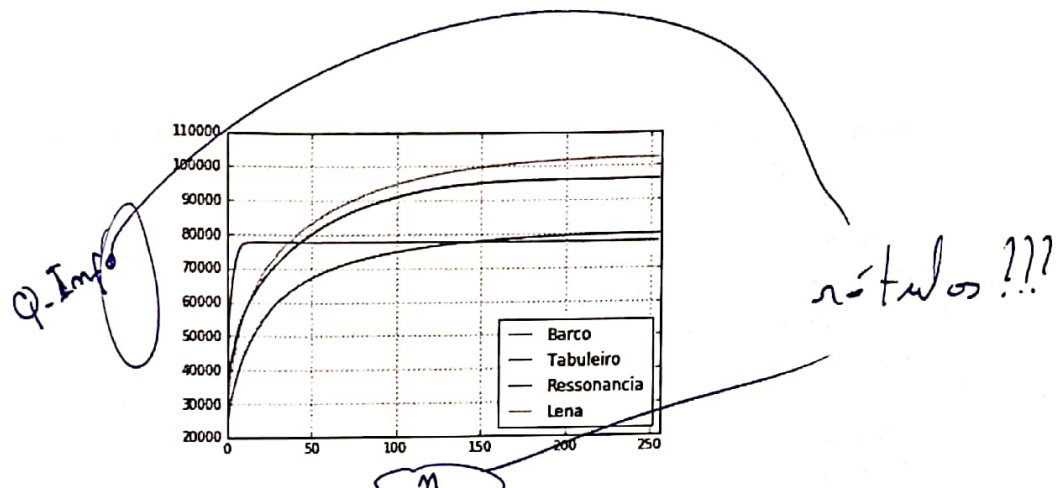


Figura 8. Gráfico com a soma dos autovalores a medida em que aumentamos o K em cada imagem.

#### 4. Discussão e Conclusão

Essa seção apresenta discussões sobre redução do uso de memória com o uso dos métodos estudados e apresenta as conclusões sobre os resultados obtidos.

##### 4.1. Considerações sobre memória

As últimas seções trataram de características da compressão de forma a evitar a perda excessiva de informação da imagem durante o processo. Porém, é importante ter em mente que, por se tratar de um processo de compressão, é preciso garantir que o dado final ocupe menos espaço em memória que o dado original. Para o DCT a discussão é simplificada, uma vez que o processo apenas transforma a matriz original para a redução posterior de sua dimensionalidade. Assim, sempre que reduzirmos o *rank* da matriz transformada, estamos reduzindo o espaço em disco utilizado.

Para o SVD é importante ressaltar que o processo de fatoração 'troca' uma matriz (M) por três matrizes ( $U\Sigma V^*$ ). Assim, para uma compressão efetiva é preciso que a redução de dimensionalidade justifique tal representação. Seja a matriz original quadrada  $m \times m$ , assim são necessários  $m^2$  posições de memória para armazenar essa imagem. Aplicando o SVD na matriz original e utilizando  $r$  valores singulares na representação reduzida, são necessários  $2mr + r$  de posições de memória. Tal número de posições é resultante das duas matrizes cheias  $U$  e  $V^*$  e da diagonal da matriz  $\Sigma$ . Para que o uso de memória seja reduzido neste processo, o valor de  $r$  deve ser tal que:

$$2mr + r < m^2$$

$$r(2m + 1) < m^2$$

$$r < \frac{m^2}{2m+1}$$

→ não explorar?  
(9)  
→ não usar essa ideia?

##### 4.2. Considerações finais

Ambos os métodos apresentaram resultados satisfatórios quanto a visualização das imagens comprimidas. Porém, pelas características de cada método e dependendo do critério de comparação, cada qual possui suas vantagens de uso. O SVD consegue representar as imagens satisfatoriamente com matrizes de *rank* baixo, enquanto na Transformada DCT

a imagem ainda é apresentada de forma manchada ou borrada. Esse comportamento pode ser explicado pela quantidade de informação utilizado por cada método na compressão. O SVD utiliza mais informações da imagem original, ocupando assim mais espaço em memória e consequentemente alcançando menores taxas de compressão que a Transformada DCT na maior parte dos testes realizados.

Boa análise.

A Transformada DCT realiza uma compactação com perdas, e tem sido atualmente utilizada em uma das etapas do processo de compactação do padrão JPEG. Sua aplicação possibilita detectar redundância espacial ao longo da imagem, identificando a semelhança de um pixel com os pixels em sua vizinhança através da matriz de frequências. O SVD distribui os coeficientes mais significantes ao longo da matriz diagonal  $\Sigma$  em ordem decrescente, explorando a compactação da imagem através da representação de frações desses coeficientes. Desse modo, por conta das restrições impostas no uso do SVD para que haja redução no uso de memória, o DCT tem sido mais utilizado em aplicações de compressão.

Considerando valores fixos de *RMS\_Erro* em relação à matriz original, a Transformada DCT apresentou melhor desempenho que o método SVD na maioria dos casos, conseguindo representar a imagem com maiores taxas de compressão. O SVD apresentou melhor resultado somente na imagem (b), mostrada na Figura 4, que apresenta alto padrão de formação. Essa característica da imagem faz com que a concentração de informação seja ainda maior nos primeiros valores singulares da fatoração, como mostrado na Figura 8, e deixa os dados mais espalhados no resultado da transformada, como mostrado na Figura 7. Dessa forma, o SVD apresenta maior capacidade de representação dos dados em direções preferenciais e estudos de padrões, podendo ser interessante em contextos nos quais isso seja desejado e que o uso de memória não seja necessariamente um problema. ;)

### Referências

- Chen, J. (2000). Image compression with ~~SVD~~ <sup>SVD</sup> ecs 289k scientific computation.
- Easton, J. R. L. (2010). *Fundamentals of Digital Image Processing*. Jr. Roger L. Easton, online edition.
- Newsletter, M. L. (2014). Discrete cosine transform and a case study on image compression.
- Putalapattu, R. (2017). Jupyter, Python, image compression and ~~SVD~~ <sup>SVD</sup> an interactive exploration.
- Sadek, R. A. (2012). ~~SVD~~ <sup>SVD</sup> based image processing applications: State of the art, contributions and research challenges. In *(IJACSA) International Journal of Advanced Computer Science and Application*, volume 3, pages 26–34.
- Sarkar, T. (2018). Essential math for data science - 'why' and 'how'.
- Stewart, G. (1993). On the early history of the singular value decomposition. In Siam, editor, *Society for Industrial and Applied Mathematics*, volume 35, pages 551–566.

Boa ideia e bom trabalho.

Houve alguns desequilíbrios na exploração dos 2 métodos.

É importante avaliar o custo computacional. SVD é mais caro.