

Aula prática 6

Imagens e SVD

Lucas Emanuel Resck Domingues

20 de Junho de 2019

1. No ambiente Scilab, para exibir as imagens *img13_Ema.png* e *marinha.png*, foram executados os comandos *imshow(imread("img13_Ema.png"))* e *imshow(imread("marinha.png"))*. A imagem é lida pelo Scilab e automaticamente é exibida na tela. Estes são os resultados:

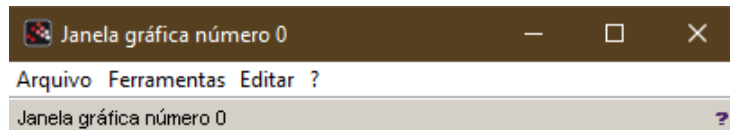


Figura 1: Exibição da imagem *img13_Ema.png*.

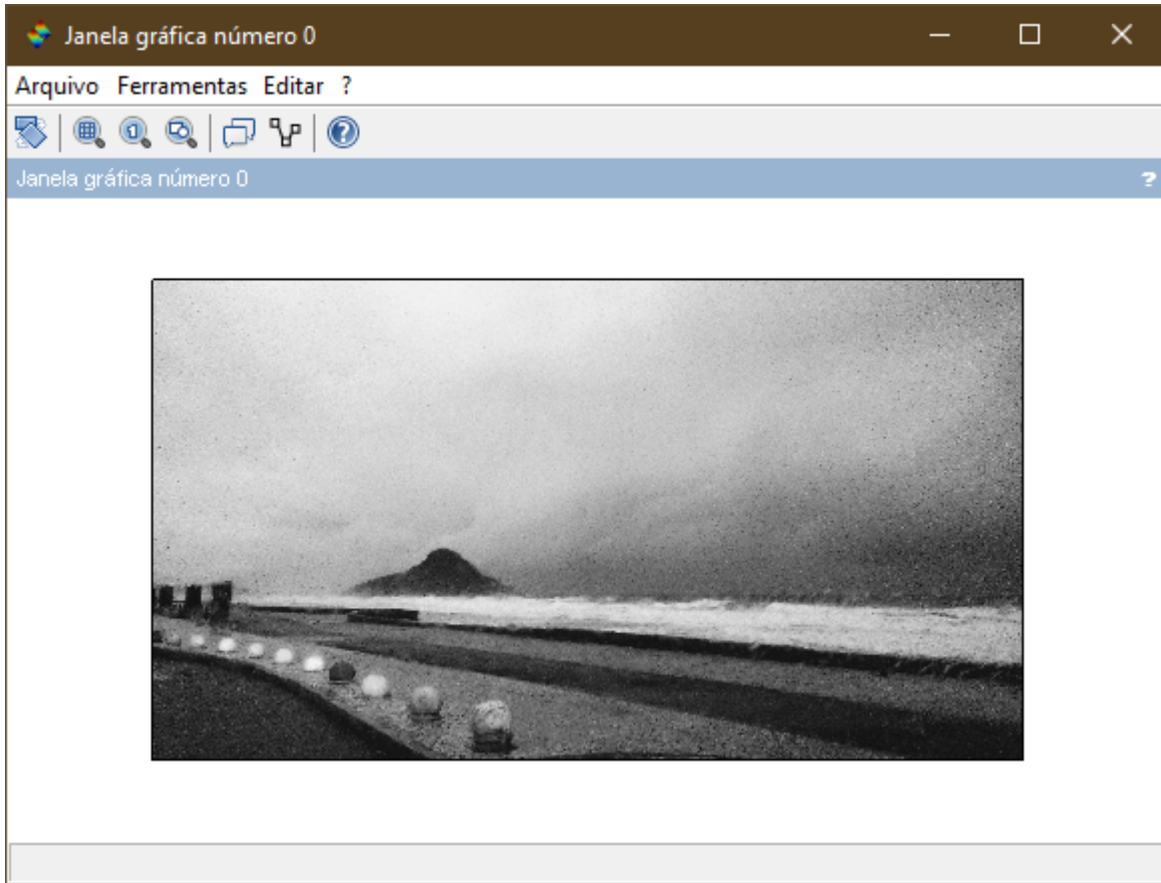


Figura 2: Exibição da imagem *marinha.png*.

2. A função *compression* foi implementada em Scilab e está no arquivo *Image compression function.sce*. Esta recebe uma matriz A correspondente à imagem em preto e branco e um valor p relativo à fração de valores singulares que vamos utilizar na compressão. Vamos testá-la para as imagens com alguns valores diferentes de p .

A imagem *img13_Ema.png* original está à esquerda e a comprimida, à direita:



Figura 3: Comparação para $p = 0,1$



Figura 4: Comparação para $p = 0,15$

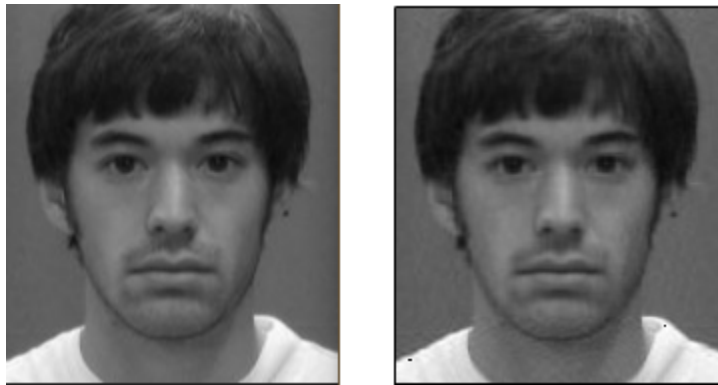


Figura 5: Comparação para $p = 0,2$

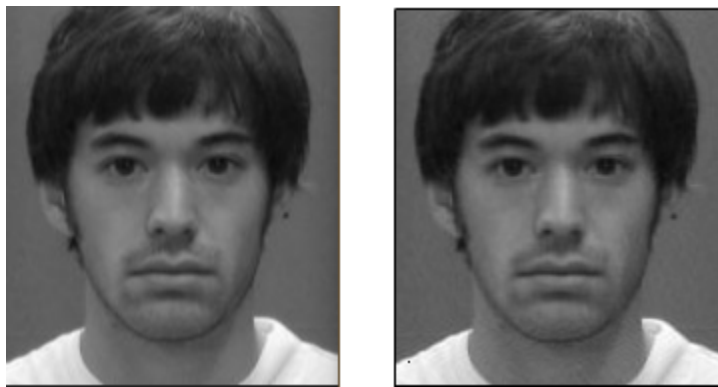


Figura 6: Comparação para $p = 0,25$

Observe que, para essa imagem, $p = 0,25$ já é suficiente para que a imagem esteja relativamente boa, comparada à original. Não é instantâneo olhar para as duas imagens e apontar aquela original.

Repetimos esse processo para a imagem *marinha.png* (porém, agora, a original fica em cima e a comprimida, embaixo):



Figura 7: Comparação para $p = 0,005$



Figura 8: Comparação para $p = 0,015$



Figura 9: Comparação para $p = 0,025$



Figura 10: Comparação para $p = 0,035$

Já para essa imagem, $p = 0,035$ é suficiente para que a imagem fique comparável à original.

Uma compressão de imagens por SVD é eficiente, pois nós podemos salvar as matrizes da decomposição SVD muito reduzidas ao invés da imagem original. Por exemplo, ao invés de guardar uma matriz 3006×5344 que representa a imagem *marinha.png*, podemos guardar as matrizes da decomposição SVD comprimidas com $p = 0,035$, sendo U de ordem 3006×105 , Σ de ordem 105×105 e V , 5344×105 . Reduzimos o armazenamento de mais de 16 milhões de elementos para menos de um milhão, sem perder muita qualidade!