Fundação Getulio Vargas
Escola de Matemática Aplicada (EMAp)

Nome: Lucas Machado Moschen
Data: 25 de junho de 2019
Professor Antônio Branco

Aula Prática 6

1) Após instalar a ToolBox de imagens e realizar os devidos procedimentos para não ter problemas com a inicialização de seus métodos, inicializei a primeira matriz. Também inicializei a segunda matriz, que representa a imagem "marinha.png". Também realizei a função visualization, que dados os parâmetros da função subplot, mostram a imagem que uma matriz representa.

```
--> i = imread('C:\Users\lucas\img14_Mos.png');

--> A = i;

--> size(A)
 ans  =

   189.   166.


--> A(1:10,1:10)
 ans  =

  166  165  165  164  163  162  162  162  161  161
  166  166  165  164  163  162  161  161  161  161
  167  166  165  164  163  162  161  161  161  161
  167  166  165  164  163  162  161  160  161  161
  166  166  165  164  163  162  161  161  161  160
  166  165  165  164  163  163  162  162  160  160
  165  164  164  164  163  163  163  163  160  160
  164  164  164  164  164  164  163  163  160  160
  164  164  164  164  164  163  163  163  162  162
  165  165  164  164  163  163  163  162  161  161
```

```
--> j = imread('marinha.png');

--> B = j;

--> size(B)
 ans  =

   3006.   5344.


--> B(1:10,1:10)
 ans  =

  177  198  221  223  211  212  218  212  214  215
  207  213  222  219  211  214  219  216  210  216
  203  206  212  213  209  211  215  213  214  218
  215  216  213  203  198  204  215  221  219  218
  216  220  215  206  205  213  220  222  218  213
  207  212  212  213  221  223  215  210  213  208
  224  219  210  210  214  207  198  201  206  207
  213  208  203  207  207  189  180  191  198  206
  219  209  197  211  205  198  195  201  198  205
  216  209  197  210  211  212  213  217  215  216
```

A partir desse momento, por razões pessoais, tive de aplicar o resto do trabalho em MatLab. Espero que o entendimento do trabalho não seja prejudicado.

2) Desenvolvi uma função chamada singular_values(A,p) que faz o trabalho esperado por essa questão. A função está escrita em Matlab, mas é facilmente utilizada em Scilab, desde

que se tenha o ToolBook de imagens do Scilab instalado. Porém no Matlab, a função iconvert não se faz necessária, por ela faz isso dentro do imshow. A minha função recebe, no lugar de p, um vetor de valores entre 0 e 1, para poder já plor as diferenças. Utilizei para esse primeiro teste p = [0.05;0.1;0.15;0.2;0.25;0.3;0.35;0.4;0.45], com a primeira imagem.

Observo que para p = 0.2 a imagem já fica suficientemente boa. A última imagem, quando p = 0.45, a imagem é praticamente idêntica a final. Também noto que para p = 0.1, a segunda imagem, já fica interessante o resultado.

O resultado para p = 1 está à extrema esquerda.

Façamos agora com uma imagem com dimensão bem maior. Considere esta imagem.

Agora, faremos o mesmo processo que fora feito com a imagem anterior. A diferença agora é que estamos lidando com uma paisagem, que pode obter resultados extremamente melhores.









É claro que a passagem da imagem produzida no MatLab para este documento acarreta a perda de qualidade da imagem. Mas vê-se que mesmo para p = 0.05, a a imagem está claramente bem compactada. Eu diria que a partir de p = 0.10 as diferenças são quase irreconhecíveis. Farei agora com valores de p ainda menores.

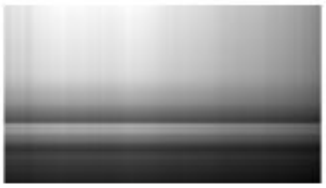

Você adivinharia o valor de p para a primeira imagem?

O valor é p = 0.01, e a imagem continua ainda extremamente semelhante.

p = [0.01;0.04;0,16;0.64] nesse caso.

Para encerrar essa observação, veja:


p = [0.0001;0.001;0.01;0.1;1]



De fato, para as primeiras duas imagens, p é de fato muito pequeno. Para se ter uma ideia, a matriz que representa essa imagem tem 3006 valores singulares não nulos. A primeira imagem mostra se tivéssemos apenas 1 valor singular sendo levado em conta. Na segunda imagem, já temo 3. Porém, a partir de aproximadamente 30 valores singulares (terceira imagem), já nos aproximamos muito do resultado esperado.

Obrigado, Branco, por esse semestre! Certamente vou levar tudo que eu aprendi em minha bagagem!

```
function singular_values(A,p)
% A is an image represented by a matrix and p is vector with
numbers between 0 and 1.
[U,S,V] = svd(double(A));

r = diag(S);
r = r(r > 0);

c = ceil(sqrt(size(p,1)));
f = floor(sqrt(size(p,1)));

for j = 1:size(p,1)
    s = max(1,floor(p(j)*size(r)));
    subplot(c, f, j), imshow(U(:,1:s)*S(1:s,1:s)*(V(:,1:s))', []);
end
end
```