Lista de Exercícios de Sistemas de TV – 2019-1

Luiz Felipe da S. Coelho
lfscoelho@ieee.org
DETEL – UERJ
Departamento de Engenharia Eletrônica e Telecomunicações
Universidade do Estado do Rio de Janeiro

27 de Maio de 2019

1 Amostragem e Subamostragem (Redução de Resolução)

Objetivo: O objetivo principal das atividades desta seção é verificar empiricamente aspectos relativos à amostragem espacial de imagens e o impacto da resolução da imagem em função da distância de visualização e de captura. Temos ainda o emprego de filtros e máscaras bidimensionais. O secundário é expandir os conhecimentos sobre manipulação de matrizes e exibição de imagens usando o Matlab.

1.1 Subamostragem

1. **Tarefa:** Carregue a imagem ZELDA_S.TIF e a mostre na tela usando 256 níveis de cinza.

2. **Tarefa:** Subamostre a imagem de 2 em cada direção (isto é, retenha somente os pixeis com índices de linha e coluna ímpares ou somente os pares) os outros pixels devem ser descartados. Obtém-se assim uma imagem com a metade das linhas e das colunas daquelas da imagem original.

3. **Tarefa:** Visualize a imagem obtida acima (apresente-a na tela). Compare a imagem obtida acima com a original.

   Dica: Usar sempre o comando truesize para que cada pixel da imagem corresponda a um pixel na tela.

   Dica: Considere que o fator de subamostragem é r para realizar as subamostragens nos itens acima. Assim retenha somente um pixel a cada r pixeis nas direções horizontal e vertical.

4. **Tarefa:** Faça uma função Matlab, que receba como parâmetros a imagem e r, e retorne a imagem subamostrada de r. Considere que somente a parte central da imagem será retida. Isto é, se as dimensões da imagem são L e C (em linhas e colunas) e se $L/r = M + resto_L$ e $C/r = N + resto_C$, onde $L,\ C$ e r são inteiros, sobrarão $resto_L$ linhas e $resto_C$ colunas que deverão ser eliminadas da imagem antes de realizar a subamostragem. Essas deverão ser eliminadas retirando a quantidade de linhas e colunas necessárias no topo, fundo, esquerda e direita da imagem original. Considere a Figura 3 como ilustrativa do

Figura 1: Imagem `zelda_s.tif`, original.

Figura 2: Imagem `zelda_s.tif`, subamostrada.

que deve ser feito. Observe que se $resto_L$ ou $resto_C$ forem ímpares ajustes devem ser realizados (tirando uma linha e uma coluna a mais).

A função que realiza a operação de subamostragem está no *script* `funcs5.py`.

5. **Tarefa:** Usando a função desenvolvida acima subamostre a imagem de 4, 8, 16 e 32 em cada direção, mostrando na tela cada uma das imagens resultantes.

6. **Pergunta:** Comente o que você observa. O que podemos comentar sobre o espectro de frequências de imagens subamostradas nos itens acima? Analise-o!

   Pode-se observar, através da Figura 4 que a qualidade visual das imagens foi drasticamente reduzida. Podemos observar também que a frequência da imagem não foi muito afetada, temos transições rápidas de pixels claros para pixels escuros.

7. **Pergunta:** Explique / averígue se as subamostragens de 4, 8, 16 e 32 solicitadas podem ser obtidas pela aplicação iterada da subamostragem de 2. Responda matematicamente.

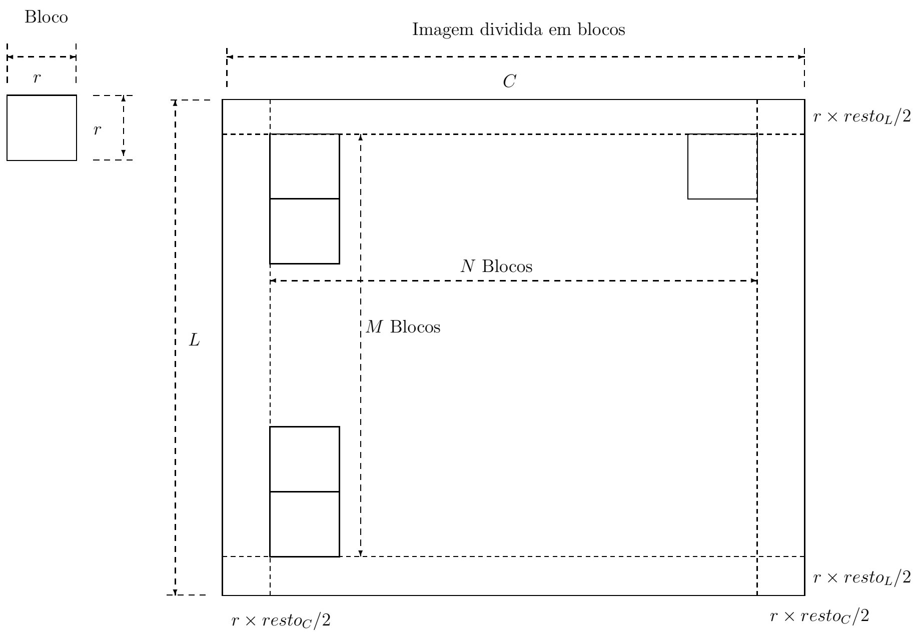

Figura 3: Divisão de uma imagem de $L \times C$ pixeis em blocos em blocos de tamanho $r \times r$.

Sim, se realizarmos o procedimento da subamostragem de 2 repetidas vezes, podemos obter os resultados das subamostragens de 4, 8, 16, 32. Isso é demonstrado no *script* `pergunta7.py`

1.2 Subamostragem Continuação

1. **Tarefa:** Repita a seção 5.1 considerando agora não a retenção de um pixel somente, mas fazendo com que os pixeis da imagem subamostrada correspondam à média dos pixeis do bloco de dimensões $r \times r$. Para isso, altere a função desenvolvida no item 5.1.4. Apresente e explique essa nova função.

Dica: Gere uma matriz ou máscara de dimensões $r \times r$ com entradas 1 e passe a sobre a imagem divida o resultado por r^2 e assim obtenha a média.

Dica: Por exemplo, veja que a aplicação de uma máscara $A(m,n)$ de dimensões $M \times N$ (sendo M e N ímpares) sobre uma imagem $I(l,c)$ de dimensões $L \times C$ gera a imagem $\hat{I}(l,c)$

$$\hat{I}(l,c) = \sum_{m=\frac{-(M-1)}{2}}^{\frac{M-1}{2}} \sum_{n=\frac{-(N-1)}{2}}^{\frac{N-1}{2}} I(l+m, c+n)A(m,n), \tag{1}$$

onde consideramos que a origem do eixo de coordenadas (dos índices de linhas e colunas da máscara) está no pixel central, conforme indicado na equação (2).

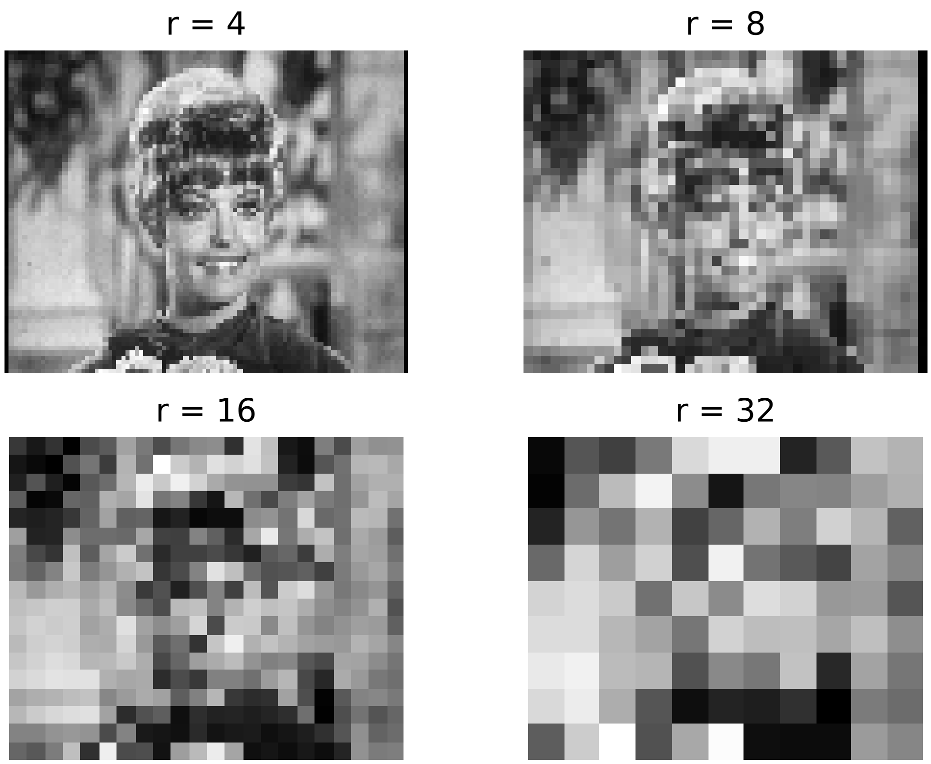

Figura 4: Quatro imagens de `zelda_s.tif` subamostradas pelo fato $r = 4,\ 8,\ 16$ e 32.

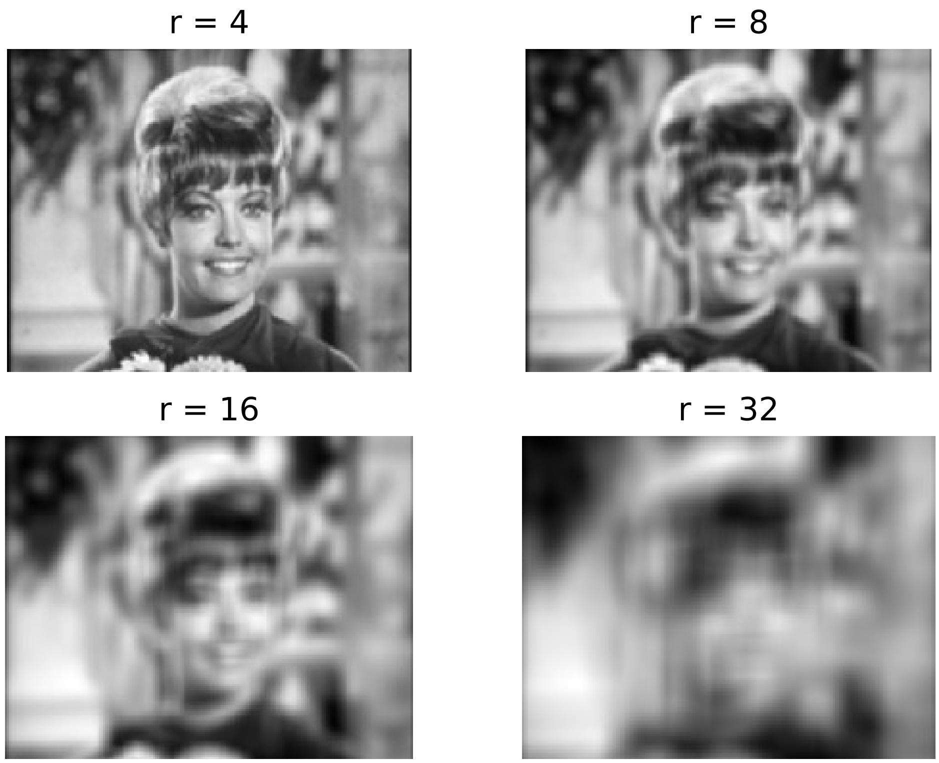

Figura 5: Quatro imagens de `zelda_s.tif` subamostradas pelo fato $r = 4,\ 8,\ 16$ e 32 utilizando o método de filtragem bidimensional.

$$A = \begin{bmatrix}
a\left(\frac{M-1}{2}, \frac{-(N-1)}{2}\right) & a\left(\frac{M-1}{2}, \frac{-(N-1)}{2}+1\right) & \cdots & a\left(\frac{M-1}{2}, 0\right) & \cdots & a\left(\frac{M-1}{2}, \frac{N-1}{2}-1\right) & a\left(\frac{M-1}{2}, \frac{N-1}{2}\right) \\
a\left(\frac{M-1}{2}-1, \frac{-(N-1)}{2}\right) & a\left(\frac{M-1}{2}-1, \frac{-(N-1)}{2}+1\right) & \cdots & a\left(\frac{M-1}{2}-1, 0\right) & \cdots & a\left(\frac{M-1}{2}-1, \frac{N-1}{2}-1\right) & a\left(\frac{M-1}{2}-1, \frac{N-1}{2}\right) \\
\cdots & \vdots & & \cdots & & \vdots & \cdots \\
a\left(0, \frac{-(N-1)}{2}\right) & \vdots & & a(0,0) & & \vdots & a\left(0, \frac{N-1}{2}\right) \\
\cdots & \vdots & & \cdots & & \vdots & \cdots \\
a\left(\frac{-(M-1)}{2}+1, \frac{-(N-1)}{2}\right) & a\left(\frac{-(M-1)}{2}+1, \frac{-(N-1)}{2}+1\right) & \cdots & a\left(\frac{-(M-1)}{2}+1, 0\right) & \cdots & a\left(\frac{-(M-1)}{2}+1, \frac{N-1}{2}-1\right) & a\left(\frac{-(M-1)}{2}+1, \frac{N-1}{2}\right) \\
a\left(\frac{-(M-1)}{2}, \frac{-(N-1)}{2}\right) & a\left(\frac{-(M-1)}{2}, \frac{-(N-1)}{2}+1\right) & \cdots & a\left(\frac{-(M-1)}{2}, 0\right) & \cdots & a\left(\frac{-(M-1)}{2}, \frac{N-1}{2}-1\right) & a\left(\frac{-(M-1)}{2}, \frac{N-1}{2}\right)
\end{bmatrix} \tag{2}$$

A função feita está no *script* `funcs5.py` e é definida como `msk_dwnsp`. Sua implementação é realizada em `52_contin.py`.

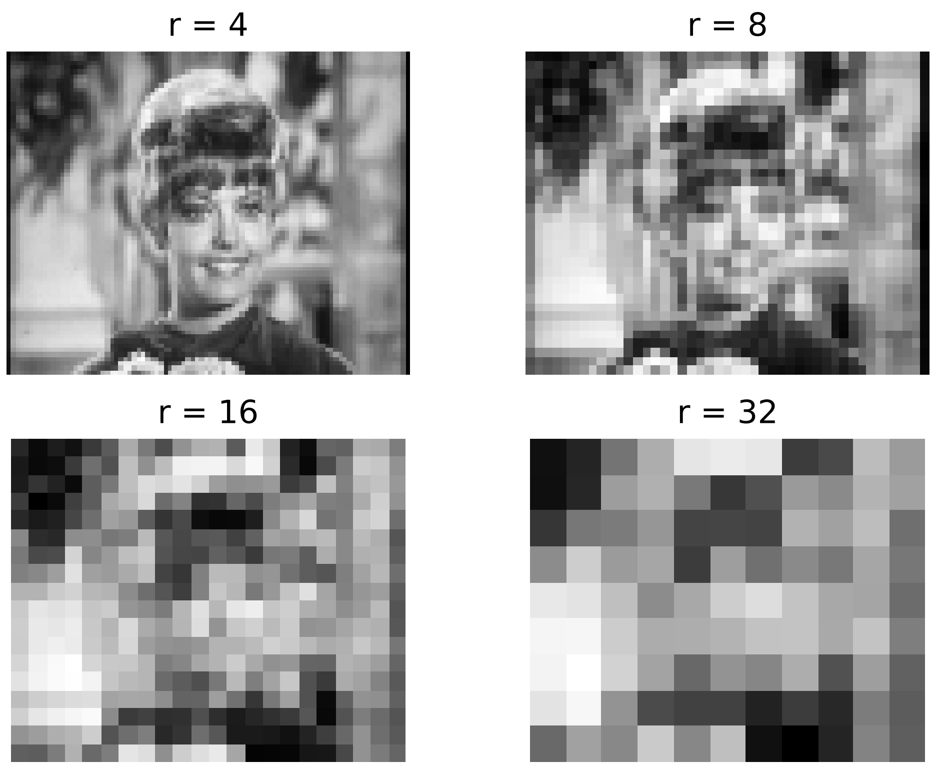

Figura 6: Quatro imagens de `zelda_s.tif` subamostradas pelo fato $r = 4,\ 8,\ 16$ e 32 utilizando o método do cálculo simples da média por soma e divisão.

2. **Pergunta:** Apresente e explique as considerações realizadas para a implementação solicitada.

   Para esta realização, foi considerada a convolução bidimensional de um arranjo $r \times r$ de uns por uma imagem e em seguida o resultado da convolução foi dividido por r^2.

3. **Pergunta:** Explique / averígue se os pixeis gerados usando a forma sugerida equivalem aos que seriam gerados por uma implementação por cálculo simples da média por soma e divisão.

   Como pode ser observado à partir de uma rápida anlálise das Figuras 5 e 6, os resultados abtidos pelo método sugerido e pelo cálculo simples da média por soma e divisão trazem resultados diferentes.

4. **Pergunta:** Compare as imagens obtidas com esta nova abordagem com aquelas obtidas com a subamostragem dos itens 5.1.3 e 5.1.5 e comente as diferenças entre os procedimentos e as qualidades das imagens subamostradas observadas.

   As imagens obtidas nesse procedimento têm transições mais suaves, o que as tornam mais nítidas que aquelas dos processos anteriores. Porém com o aumento do fator de subamostragem as imagens ficam mais embaçadas.

Dica: A passagem de uma máscara sobre uma imagem pode ser obtida a partir da convolução. O Matlab fornece a função conv2 para a convolução bidimensional.
Sejam $F(m,n)$ e $G(m,n)$ duas sequencias bidimensionais, isto é, m e $n \in \mathbb{Z}$, a convolução 2D dessas sequencias é

$$F(m,n)*G(m,n) = G(m,n)*F(m,n) = \sum_k \sum_l F(k,l)G(m-k,n-l) = \sum_l \sum_k G(k,l)G(m-k,n-l). \tag{3}$$

Observe que a imagem resultante da convolução de uma imagem de dimensões $M \times N$ por um filtro de dimensões $r \times r$ tem dimensões $(r+M-1) \times (r+N-1)$.

5. **Pergunta**: Sendo assim, explique como a aplicação de uma máscara a uma imagem pode ser obtida a partir da convolução bidimensional.

Dica: Não se esqueça de considerar dois aspectos:

a) Quais transformações deverá sofrer uma máscara de forma a gerar uma matriz A que pode ser convoluída bidimensionalmente com a imagem e obter o mesmo resultado da aplicação da máscara?
Para que o resultado da convolução bidimensional seja do mesmo da aplicação da máscara, a matriz A deverá ser composta de termos $1/r^2$; isto para A com dimensões $r \times r$.

b) Qual a eliminação de linhas e colunas a ser aplicada à imagem resultante da convolução de forma a obter uma imagem que possua as mesmas dimensões da imagem original, de forma tal que a área da imagem resultante seja o mais próxima possível da área da imagem original?
Para que a imagem resultante tenha a mesma dimensão da imagem original, devemos reduzir as linha e as colunas em $r-1$; isto para A com dimensões $r \times r$.