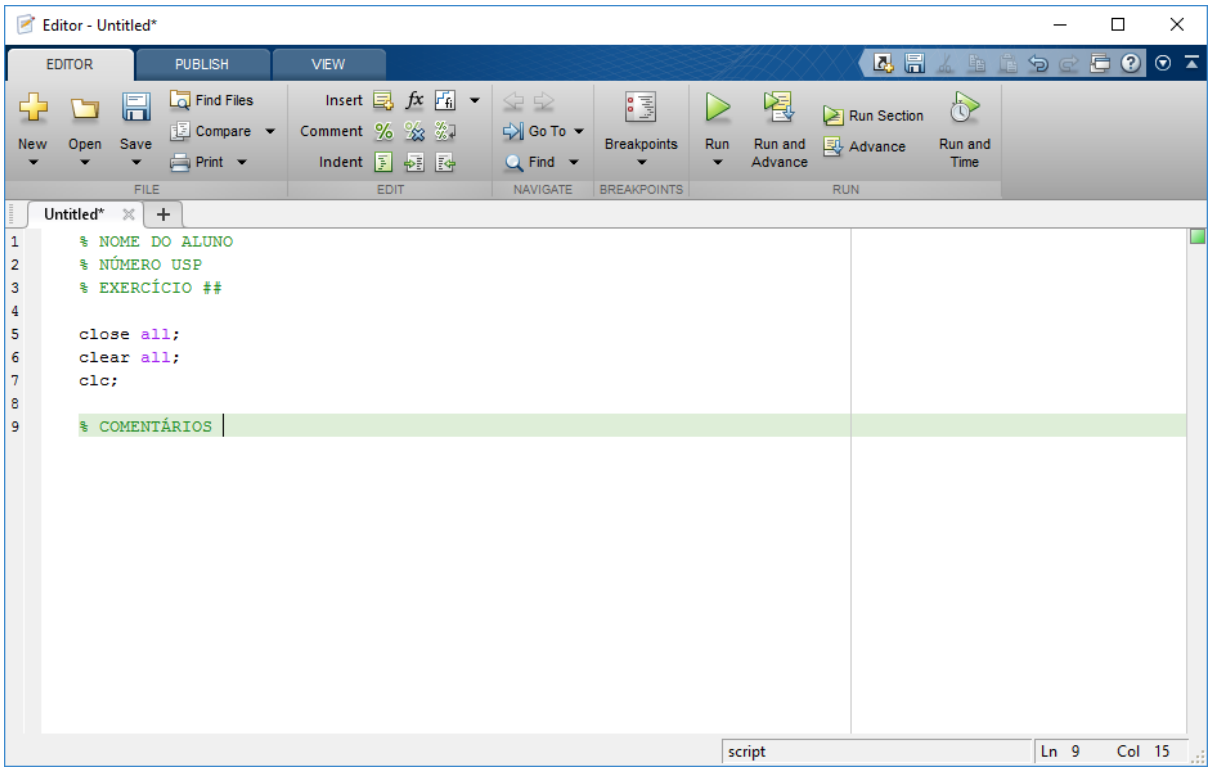
SEL0449 –Processamento Digital de Imagens Médicas
Prof. Dr. Marcelo Andrade da Costa Vieira

Lista de Exercícios
#7

Instruções:

- Essa lista consiste de 4 exercícios (E_1 a E_4).
- Deve ser gerado um arquivo no editor do Matlab (extensão *.m) para cada exercício pedido.
- Deve-se colocar comentários nos programas desenvolvidos.
- As perguntas devem ser respondidas também como comentários no arquivo.
- Deve-se tornar o diretório onde estão as figuras e os arquivos *.m como um diretório padrão do Matlab.
- Depois de terminado os exercícios, todos os arquivos *.m devem ser comprimidos em um único arquivo e enviado ao professor pelo sistema *Moodle* até a data máxima de entrega.

- Utilizar o padrão mostrado na Figura abaixo para seus arquivos *.m:
  - Colocar um cabeçalho contendo seu nome, número USP e o número do exercício correspondente (E1, E2, E3...);
  - Iniciar todos os exercícios com os 3 comandos mostrados na Figura abaixo, que servem para limpar as variáveis e as figuras abertas, além de limpar a tela de comando do Matlab;
  - Colocar comentários nas linhas de programa.



1) Ruído:

A função do MATLAB que contamina uma imagem com ruído é a ***imnoise*** que tem a seguinte sintaxe:

```
g = imnoise(f, type, parameters)
```

Onde *f* é a imagem original e *type* é o tipo de ruído conforme mostra a Tabela 1.

Tabela 1: Tipos de ruídos da função imnoise

Value	Description
`'gaussian'`	Gaussian white noise
`'localvar'`	Zero-mean Gaussian white noise with an intensity-dependent variance
`'poisson'`	Poisson noise
`'salt & pepper'`	On and off pixels
`'speckle'`	Multiplicative noise

E_1: Ruído sal e pimenta e ruído gaussiano

a) Insira ruído na imagem ***barco.gif*** usando a função *imnoise*. Gere duas imagens ruidosas diferentes: uma com ruído "sal e pimenta" e outra com ruído "gaussiano".
b) Filtre as duas imagens ruidosas utilizando um filtro da média 3 x 3 e um filtro da mediana 3 x 3. Utilizar as funções *imfilter* e *medfilt2*, respectivamente.
c) Mostre a imagem original, as imagens ruidosas e as imagens filtradas.
d) Comente as diferenças encontradas nas imagens. Qual a melhor técnica para cada tipo de ruído?

E_2: Média de Múltiplas Imagens

a) A partir da imagem original (***barco.gif***), gerar 15 amostras de imagens ruidosas contaminadas por ruído 'gaussiano' com média zero e variância 0.05 (usando a função *imnoise*).
b) Filtrar o ruído dessas imagens usando média de múltiplas imagens usando 5 e 25 amostras, respectivamente.
c) Filtrar o ruído dessa imagem usando o filtro de média 5 x 5 ('*average'*).
d) Comente os resultados encontrados com o filtro da média e com a média de múltiplas imagens (5 e 15 amostras).

2) Restauração de imagens:

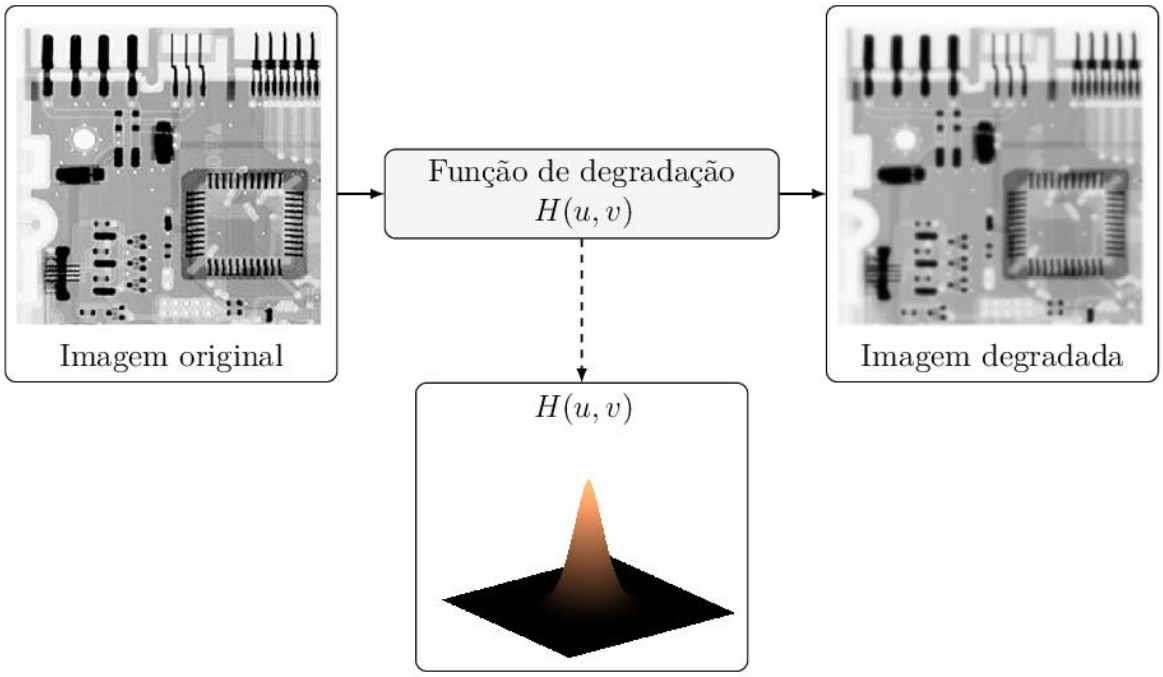


Figura 1: Modelo de degradação utilizado na construção de "circuit_blur.tif"

E_3: Criar uma "m-file" que funcione para a imagem "circuit.tif" e sua versão degradada, "circuit_blur.tif".

A imagem foi degradada por uma função de espalhamento de ponto modelada a partir da seguinte equação ($k = 5.10^{-4}$):

$$H(u,v) = \exp\left(-k * (u^2 + v^2)^{\frac{5}{6}}\right)$$

a) Visualizar a imagem degradada;
b) Construir *padding* simétrico (espelhamento) *. Visualizar a imagem no domínio da frequência (Espectro de Fourier);
c) Carregar a função de degradação contida em "E3_EXTRA.mat". Visualizar a magnitude da função de degradação em 3D (função *surf*, *mesh*, *view*) e sua imagem (em níveis de cinza).
d) Restaurar a imagem degradada utilizando filtro inverso não limitado;
e) Visualizar a imagem restaurada e sua versão não degradada. Comentar os resultados;
f) Restaurar a imagem degradada utilizando filtro inverso limitado (por duas frequências de corte diferentes); **
g) Visualizar as imagens restauradas e sua versão não degradada. Comentar os resultados;

* O *padding* simétrico pode ser construído utilizando a função `padarray` com o argumento: `'symmetric'`. A direção de construção do *padding* pode ser modificada com os argumentos: `'post'`,`'pre'` ou `'both'` (*Default*).

** DICA: Pode-se limitar o filtro inverso implementando $H(u,v) = 1$ para todas as frequências maiores que uma determinada frequência de corte.

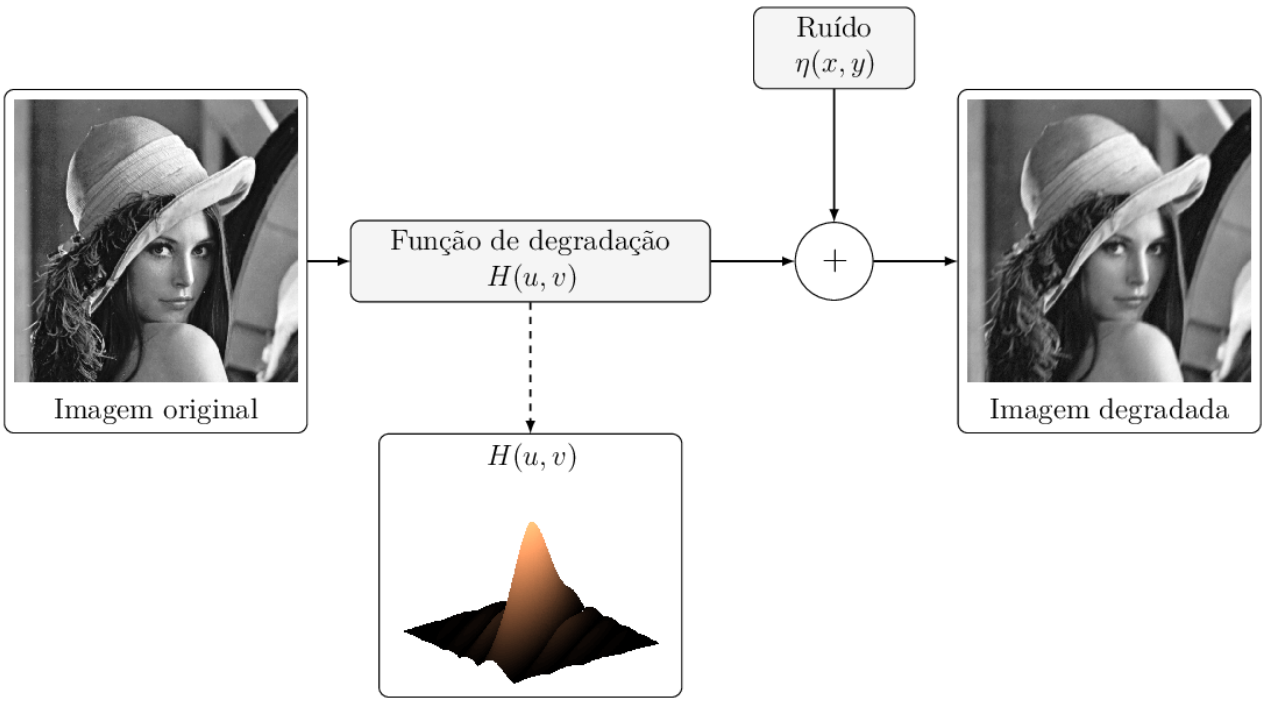


Figura 2: Modelo de degradação utilizado para a construção de "lena_deg.tif"

E_4: Criar uma "m-file" que funcione para a imagem "lena.tif" e sua versão degradada, "lena_deg.tif".

A imagem foi degradada por uma função de espalhamento de ponto e movimento construídos a partir das seguintes equações:

$$H(u,v) = H_{blur}(u,v).H_{motion}(u,v)$$
$$H_{blur}(u,v) = e^{-k.(u^2+v^2)^{5/6}}$$
$$H_{motion}(u,v) = \gamma.sinc(\alpha u + \beta v).e^{-j\pi(\alpha u+\beta v)}$$

Sendo $k = 5.10^{-5}$, $\alpha = -6.10^{-3}$, $\beta = 4.10^{-3}$ e $\gamma = 1$. A imagem foi também corrompida por ruído aditivo gaussiano com variância igual a $\sigma^2 \cong 6{,}5$.

a) Visualizar a imagem degradada;
b) Construir *padding* simétrico (espelhamento) e visualizar a imagem degradada no domínio da frequência (Espectro de Fourier);
c) Carregar a função de degradação contida em "E4_EXTRA.mat". Visualizar a magnitude da função de degradação em 3D (função *surf, mesh, view*) e sua imagem (em níveis de cinza).
d) Restaurar a imagem degradada utilizando filtro inverso não limitado;
e) Visualizar a imagem restaurada e sua versão não degradada. Comentar os resultados;
f) Restaurar a imagem degradada utilizando filtro inverso limitado (por duas frequências de corte diferentes); **
g) Visualizar as imagens restauradas e sua versão não degradada. Comentar os resultados;
h) Restaurar a imagem degradada utilizando filtro de Wiener com parâmetro $K = 3.10^{-3}$;
i) Essa "m-file" deve mostrar todas as figuras obtidas nos itens anteriores; Colocar título nas figuras geradas.