



## PROJETO I

### Espera-se do projeto:

A clara identificação do problema a ser resolvido. Descrição completa da solução proposta e resultados. Identificação objetiva e avaliação dos resultados.

### O que deve ser entregue?

- Relatório feito pelos alunos, em 4 páginas como mínimo e 5 como máximo. Detalhes sobre o relatório, ler o arquivo "LER ANTES DE FAZER OS RELATÓRIOS" no Moodle.
- Código da solução, comentado e com descrição de uso e extensão.

### Questão 1

1.1 - Crie um programa LER\_YUV para ler arquivos YUV (binário). Sendo que a função deve receber como argumento a resolução do vídeo e o número do quadro a ser lido. O arquivo YUV é um arquivo binário que não possui cabeçalho. Simplesmente os pixels Y, seguidos dos pixels U e depois V de um quadro, depois repete para o próximo quadro. Assumir que o arquivo está em formato 4:2:0. Isso significa que se a resolução é 352x288, as componentes U e V tem metade dos pixels em cada direção. A função então deve ser algo do tipo `[Y,U,V] = LER_YUV(foreman.cif, 352, 288, 10)`. Isso significa que vai pular a qtd de bytes igual a 9 quadro (Y,U,V) e ler os próximo quadro Y,U e V.

1.2 - Crie uma função que aumenta o tamanho de uma imagem em múltiplos de 2 em cada direção. Pode ser uma função que sempre aumenta em 2 e é chamada iterativamente.

- Deve ser criada uma coluna adicional com zeros entre duas existentes.
- Deve ser feito o mesmo para as linhas.
- De forma que tem uma coluna e uma linha de zeros pixels pretos entre cada linha e coluna da imagem real
- Depois trocar os pixels pretos por o imediato superior ou imediato a esquerda. Quando possível escolher ambos sempre usar o superior.

1.3 – Faça outro programa similar ao da questão 2. Porém pesquise ou pense uma forma melhor de preencher os pixels pretos.

1.4 – Use as funções das partes 2 e 3 para deixar os componentes U e V do mesmo tamanho que Y. Use funções prontas de matlab ou opencv para transformar YUV em RGB. Mostre o RGB da imagem com os U e V originais e com os interpolados. Comente os resultados

1.5 – Agora aumente em 2 o tamanho do Y, U e V. Ficando no padrão 4:2:0. Similar a parte 4. Mostre os resultados e comente.

## Questão 2

Faça um programa para realizar o processo de aguçamento (sharpening) de imagens, mediante o uso de um Filtro Laplaciano no domínio espacial. As entradas do programa devem ser: (i) a imagem, (ii) o tipo de filtro laplaciano. A imagem que deve ser usada é o arquivo "IMAGE1.PGM". No relatório desta parte deve estar incluído:

2.1 Uma figura com o resultado de fazer o aguçamento utilizando diretamente o filtro Laplaciano de tamanho 3x3, em todas as direções (+/- 8 no centro).

2.2 Uma figura com o resultado de primeiro utilizar um filtro de suavização Gaussiano de tamanho 3x3 com  $\sigma^2=0,5$  e depois, o filtro Laplaciano de tamanho 3x3 com +/- 4 no centro.

2.3 Uma figura com o resultado de primeiro utilizar um filtro de suavização Gaussiano de tamanho 3x3 com  $\sigma^2=1,0$  e depois, o filtro Laplaciano de tamanho 3x3 com +/- 4 no centro

2.4 Indicar qual resultado é melhor subjetivamente e porque.

(Dica: Utilizar as funções 'fspecial', e 'imfilter' no matlab, ou 'cvGaussianBlur', 'cvFilter2D', e 'cvLaplace' no Opencv)

## Questão 3

Faça um programa para realizar o processo de filtro rejeita-**notch**, mediante o uso de filtros passa-altas Butterworth (com  $n=4$ ). A imagem que deve ser usada é o arquivo "MOIRE.TIF". No relatório desta parte deve estar incluído:

3.1 Uma Figura com a imagem original

3.2 Uma Figura com a imagem filtrada utilizando 4 pares notch com as seguintes características (sem usar padding):

1° PAR:  $D_0=10$ ,  $u_k=39$ ,  $v_k=30$

2° PAR:  $D_0=10$ ,  $u_k=-39$ ,  $v_k=30$

3° PAR:  $D_0=5$ ,  $u_k=78$ ,  $v_k=30$

4° PAR:  $D_0=5$ ,  $u_k=-78$ ,  $v_k=30$

### 3.3 Comentários sobre os resultados obtidos.