

## Primeiro trabalho de laboratório da parte Visual - Fundamentos de processamento de sinais visuais

**Objectivos:** Este trabalho pretende aplicar e consolidar conhecimentos sobre representação de sinais visuais no domínio digital; conhecer técnicas básicas de processamento de imagem com o Matlab - convolução e filtragem; detecção de contornos.

### Introdução

O objectivo deste trabalho é o de permitir aos estudantes adquirirem uma melhor compreensão sobre os fundamentos de representação de sinais visuais no domínio digital e de técnicas de processamento que permitem melhorar a qualidade das imagens, extrair características de baixo nível e preparar os sinais para a compressão..

Para a realização destas experiências, sempre que necessário, serão utilizados ficheiros com imagens disponíveis no Moodle da UC como sinais de entrada para os algoritmos/processos implementados pelos programas ou scripts Matlab também disponíveis na página Moodle da UC. A maioria dessas imagens têm o formato BMP (bitmap) o que significa que cada pixel é representado por três valores RGB de 8 bits, ou seja, no total por 24 bits. Comparando resultados obtidos pela aplicação dos diferentes algoritmos, pretende-se que o estudante adquira uma melhor compreensão do papel desempenhado pelas diferentes técnicas e formatos.

Nota: o símbolo 🖋️ significa que deve incluir no seu relatório gráficos ou imagens que resultaram do processamento efectuado ou ainda código que tenha desenvolvido. O símbolo 🧑🔬 indica que deve incluir no relatório uma breve análise aos resultados que obteve.

### Trabalho a desenvolver

#### 1. *Experiências com espaços de cor*

Existem vários espaços de cor para representar sinais visuais, cada um com o seu próprio sistema de coordenadas, tendo finalidades ou áreas de aplicação distintas. Nesta parte do trabalho iremos utilizar três espaços de cor: RGB (Red, Green, Blue), HSV (Hue, Saturation, Value, em que V representa o brilho) e YUV (luminância e sinais diferença de cor).

1.1. Escreva um script Matlab 🖋️ que:

- ii) importe uma imagem com o formato bitmap (espaço de cores RGB) e apresente essa imagem no écran;
- iii) separe cada componente RGB numa matriz diferente e apresente no écran cada uma delas 🖋️;

- iv) converta essa imagem para o espaço de cores HSV e apresente essa imagem no écran;
- v) separe cada componente HSV numa matriz diferente e apresente no écran cada uma delas ✎;

Corra o script com várias imagens tais como “peppers.png”, “lighthouse.png” e outras disponíveis no Matlab, ou “floresVermelhas.bmp”, “folhasVerdes.bmp”, “praia.bmp” e “elephant.bmp”, disponíveis no Moodle. Compare entre si componentes de cada imagem 🕵️.

1.2 Desenvolva um script semelhante mas em vez de converter para HSV converta para YCbCr.

Corra o script com as mesmas imagens. Compare entre si componentes de cada imagem 🕵️. Compare com os resultados obtidos com o script anterior.

1.3 Utilize o script “rgb2yuv.m” e verifique as diferenças que existem relativamente aos resultados obtidos na experiência 1.2. 🕵️

## **2. Variação das dimensões espaciais de imagem usando ou não filtros com imagem de teste “imzoneplate”**

Nesta parte deve desenvolver no Matlab o script “ampliaReduz.m” o qual deve utilizar o script “imzoneplate.m” que está disponível no Moodle da UC (obtido do site da MathWorks).

Antes de iniciar o trabalho, analise o código dos programas fornecidos por forma a perceber as operações realizadas.

O programa a desenvolver “ampliaReduz.m” deve permitir ampliar/reduzir as dimensões espaciais de uma imagem de teste “zone plate” criada durante a sua execução usando a função “imzoneplate.m”. Deve permitir efectuar ampliação/redução por simples repetição/eliminação de pixels e efectuar ampliação/redução recorrendo à função built-in do Matlab “imresize.m”. Pode ter a opção de utilizar um filtro de média ou gaussiano antes de efectuar a ampliação/redução. O programa deve apresentar no écran as imagens originais e as processadas bem como os respectivos gráficos de densidade espectral e de variação do sinal no espaço.

Inicie o Matlab e mude para o seu próprio directório de trabalho. Copie todos os ficheiros necessários (programas e imagens).

- i) desenvolva o seu programa “ampliaReduz.m” e execute-o com diferentes dimensões para a imagem de teste zoneplate e usando diferentes métodos de interpolação da função built-in “imresize.m”. ✎ Analise a função built-in “resample.m” Corra essa função usando a imagem de teste zoneplate. Compare e interprete os resultados. 🕵️

## **3. Experiências de filtragem**

Nestas experiências deve desenvolver dois programas que permita efectuar diferentes tipos de filtragem de imagens e vai aplicar um detector de contornos usando a função built-in do Matlab “edge.m”.

- i) O primeiro programa deve ser desenvolvido recorrendo às funções built-in do Matlab “fspecial” e “imfilter”. ✎

Execute o programa usando como sinal de entrada diferentes imagens e testando os vários tipos de filtro possíveis. Verifique os efeitos de cada filtro ✎.

Execute o programa com valores de dimensão do filtro diferentes para o caso do filtro de média e gaussian. Analise os resultados para os diferentes valores. 🕵️

ii) O segundo programa a desenvolver deve efectuar a convolução explícita entre uma imagem e o kernel de um filtro especificado no programa 🖋️. Os kernels a especificar devem ser os seguintes:

(a) Filtro de média (3X3 e 5x5)

(b) Filtro de mediana

(c) Operadores de Sobel:  $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$   $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$

(d) Operadores de Prewitt:  $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & 1 \end{bmatrix}$

Use filtro de média e de mediana para remover o ruído das imagens ruido1.jpg e ruido2.jpg. Comente os resultados. 🖋️ 🕵️

Aplique os operadores de Sobel e de Prewitt às imagens “casa1.jpg”, “casa2.jpg” e “contorno.jpg” e comente os resultados. 🖋️ 🕵️

iii) Aplique o detector de contornos de Canny à imagem Contorno.jpg usando a função “edge” do Matlab. Teste diferentes valores de “sigma” e “threshold” e interprete os resultados. 🖋️ 🕵️

O relatório deve ser entregue até dia 28 de Março no Moodle.