



Segundo trabalho de laboratório - Lab 2: Fundamentos de processamento de sinais visuais

Objectivos: Este trabalho pretende aplicar e consolidar conhecimentos sobre representação de sinais visuais no domínio digital; conhecer técnicas de processamento para melhoramento da imagem e para compressão.

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. *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. Com base no conjunto de funções descritas no guião 2-1, escreva um script Matlab  que:


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

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

Corra o script com as imagens “testRGB.bmp”, “floresVermelhas.bmp”, “folhasVerdes.bmp”, “praia.bmp” e “elephant.bmp”. Compare entre si componentes de cada imagem . Pode testar com outras imagens coloridas bitmap (.bmp).

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

Corra o script com as imagens “testRGB.bmp”, “floresVermelhas.bmp”, “folhasVerdes.bmp”, “praia.bmp” e “elephant.bmp”. 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 vai trabalhar no Matlab utilizando os scripts Matlab “ampliaReduz.m” e “imzoneplate.m” que estão disponíveis no Moodle da UC. alguns destes programas são versões modificadas de programas obtidos do site da MathWorks.

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


O programa de redução/ampliação das dimensões de uma imagem funciona com uma imagem de teste “zone plate” criada durante a sua execução usando a função “imzoneplate.m”.


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

- i) corra o programa “ampliaReduz.m” com diferentes dimensões para a imagem de teste zoneplate e usando diferentes métodos de interpolação da função built-in “imresize.m”. Compare e interprete os resultados. 

3. Experiências de filtragem

3.1. Nestas experiências vai utilizar o programa “filtro2019.m”, o qual permite efectuar diferentes tipos de filtragem de imagens.



Abra o programa no editor e verifique o código para perceber o que o programa faz e com que tipo de informação opera. Corra então o programa usando diferentes imagens e testando os vários tipos de filtro possíveis. Verifique os efeitos de cada filtro .

Corra 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. 

3.2. Nesta experiências vai utilizar o script “filtroMediana.m”, o qual permite eliminar ruído pontual de imagens sem retirar nitidez à imagem, através da aplicação de um filtro de mediana.



Multimédia e Novos Serviços
(EIC0064) 2019-2020

Abra o programa no editor e verifique o código para perceber o que o programa faz e com que tipo de informação opera. Corra então o programa usando diferentes imagens (originalmente com e sem ruído) e testando diferentes quantidades de ruído. Corra as mesmas imagens com filtros de média ou gaussianos e compare e analise os resultados.  .

Nota: existem disponíveis imagens que têm o tipo de ruído pontual aleatório desejado para esta experiência ('ruído1.jpg' e 'ruído2.jpg') e que pode usar para fazer a comparação. Mas também pode introduzir código no script "filtroMediana.m" para guardar as imagens às quais é introduzido ruído.

O relatório deve ser entregue até dia 14 de Março no Moodle