

Guião 1

- Configuração do IDE e primeiros exemplos com OpenCV.
- Manipulação de imagens, leitura e visualização de imagens com diferentes formatos, manipulação directa de *pixels*.
- Exemplo de uma operação matemática: subtração de duas imagens.
- Interação com a janela: selecção de pontos e desenho na imagem.
- Conversão entre espaços de cores e formatos.

Documentação

A documentação do OpenCV pode ser consultada em: <http://docs.opencv.org>

Instalação do OpenCV em Linux/UBUNTU

O processo de instalação encontra-se bem explicado, por exemplo, em:

<http://www.codebind.com/linux-tutorials/install-opencv-ubuntu-18-04-lts-c-cpp-linux/>

Alternativa – Configuração do Visual Studio em Windows

O processo de configuração do Visual Studio encontra-se bem explicado, por exemplo, em:

<http://opencv-srf.blogspot.pt/2017/11/install-opencv-with-visual-studio.html>

Para poder utilizar a biblioteca OpenCV num projecto em C++ são necessários os passos seguintes (**adaptados à estrutura de directórios de cada um**):

1. Criar um projeto vazio do tipo **Win Console Application**.
2. Indicar que a compilação será efetuada para 64 bits.

Para isso, selecione **Project Property Pages -> All Configurations -> x64**.

3. Indicar onde se encontram os ficheiros cabeçalho do OpenCV.

Para isso, selecione **Project Property Pages -> All Configurations -> Configuration Properties -> C/C++ -> General**.

No campo **Additional Include Directories** inclua o directório onde se localizam os directórios com os vários ficheiros cabeçalho.

4. Indicar onde se encontram as várias bibliotecas do OpenCV.

Para isso, seleccione **Project Property Pages -> All Configurations -> Configuration Properties -> C/C++ -> General**.

No campo **Additional Library Directories** inclua o directório onde se localizam as bibliotecas do OpenCV.

5. Atualizar a variável PATH do sistema.

Para isso, seleccione **Project Property Pages -> All Configurations -> Configuration Properties -> C/C++ -> Debugging**.

No campo **Environment** atualize a variável do sistema, de modo semelhante a `PATH=c:\opencv331\build\x64\vc14\bin;%PATH%`

6. No **Configuration Manager** quer para a configuração **Debug** quer para a configuração **Release** indicar que a compilação será efetuada para **x64**.
7. Associar o ficheiro **opencv_world331d.lib** à configuração **Debug** e o ficheiro **opencv_world331.lib** à configuração **Release**.

Para isso, seleccione **Project Property Pages -> Debug -> Linker -> Input -> Additional Dependencies -> Edit** e acrescente **opencv_world331d.lib**.

E seleccione **Project Property Pages -> Release -> Linker -> Input -> Additional Dependencies -> Edit** e acrescente **opencv_world331.lib**.

1.1 Primeiro exemplo

Compile e teste o código fornecido no ficheiro **OpenCV_ex_01.cpp**

Analise o código e verifique quais as funcionalidades que são usadas.

Note o modo como é declarada, lida e visualizada uma imagem.

Tarefa

Aceda aos atributos do objeto **image** para determinar o tamanho, o número de canais e o número de *bytes* por *pixel* da imagem lida.

1.2 Acesso directo aos *pixels* de uma imagem

Tarefas

Crie uma nova imagem, com a mesma dimensão da imagem lida, usando o método **cv::clone**.

Aceda directamente aos *pixels* da imagem através do atributo **data** e coloque a 0, na nova imagem, todos os *pixels* em que os valores da imagem original são menores que 128.

Uma (melhor) alternativa para acesso aos *pixels* da imagem é a utilização do método genérico **at** que, por exemplo, permite utilizar os índices de linha e de coluna:

```
image.at<uchar>(y,x)
```

Visualize a imagem original e a imagem obtida.

Acrescente a possibilidade de o nome do ficheiro ser lido na linha de comando.

1.3 Operação matemática simples: Subtração entre imagens

Tarefas

A partir do exemplo fornecido, crie um novo programa que permita ler as duas imagens **deti.bmp** e **deti.jpg**.

Visualize cada uma das imagens numa janela diferente.

Efectue a operação de **subtração** para poder visualizar eventuais diferenças entre as duas imagens.

Comente o resultado.

A operação de subtração pode ser efectuada quer usando o respetivo operador (−) quer chamando a função **subtract**.

(opcional)

Pode abrir uma imagem à sua escolha num programa de manipulação de imagens e salvaguardá-la com diferentes opções de compressão no formato **jpeg**.

Observe o resultado da subtracção para compressões diferentes.

1.4 Interação Simples: Seleção de um *pixel* e desenho de um círculo

Tarefa

Acrescente ao programa anterior um *callback* para permitir detectar um *click* do utilizador na imagem, para tal utilize a função:

```
setMouseCallback( const string& winname,  
                  MouseCallback on_mouse, void userdata=0 );
```

A função invocada tem o seguinte protótipo:

```
void on_mouse( int event, int x, int y, int flags, void *param );
```

Ao usar a tecla da direita, o programa deverá desenhar um círculo a cheio centrado no ponto seleccionado pelo utilizador.

1.5 Conversão entre espaços de cor

Tarefa

Escolha uma imagem a cores e utilize a função **cvtColor** para a converter para níveis de cinzento (CV_RGB2GRAY).

(opcional)

Consultando a documentação da função **cvtColor**, modifique o programa para visualizar a mesma imagem noutros espaços de cor (por exemplo: CV_RGB2HLS, CV_RGB2XYZ, CV_RGB2HSV).