

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: subtracção de duas imagens.
- Interacção 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://opencv.org/documentation.html>

Configuração do Code::Blocks

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

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

Para isso, nas **Build options** do Projecto, seleccione **SearchDirectories -> Compiler**.

Inclua o directório onde se localizam os directórios com os vários ficheiros cabeçalho.

Por exemplo: C:\opencv230\build\include

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

Para isso, nas **Build options** do Projecto, seleccione **SearchDirectories -> Linker**.

Inclua o directório os vários ficheiros **lib**.

Por exemplo: C:\opencv230\build\lib

3. Indicar os vários ficheiros das bibliotecas (**lib**) do OpenCV.

Para isso, nas **Build options** do Projecto, seleccione **Linker Settings** e adicione os vários ficheiros.

Atenção — PATH

Para o desenvolvimento em ambiente Windows, a variável **PATH** do sistema deverá incluir a localização do compilador e dos ficheiros **DLL** da biblioteca.

Por exemplo:

```
C:\MinGW\bin;  
C:\opencv230\build_MingW\bin;
```

Atenção — Recompilação

Os ficheiros da instalação do OpenCV podem não ser os adequados para o ambiente de trabalho de cada um.

Pode ser necessário recompilar as bibliotecas!

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 objecto **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.

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: Subtracção 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 **subtracção** para poder visualizar eventuais diferenças entre as duas imagens.

Comente o resultado.

(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 Interacção Simples: Selecção 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).

Tome em atenção que a imagem destino tem que ser criada, com as mesmas características que a imagem original (excepto o número de canais), antes de invocar a função de conversão.

(opcional)

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