Processamento de Imagens

Turquesa - 2022/02

Aulas 09 e 10

Manipulação de pixels em imagens

Manipulação de pixels em imagens

- Para acessar os valores de um pixel em uma imagem, primeiro precisamos saber qual o tipo de matriz utilizada para armazenar a imagem:
  - **CV_8UC1**: para imagens em tons de cinza de 1 canal de 8 bits;
  - **CV_32FC1**: para imagens em escala de cinza de 1 canal de ponto flutuante de 32 bits;
  - **CV_8UC3**: para imagens coloridas de 3 canais de 8 bits;
  - **CV_32FC3**: para imagens coloridas de 3 canais de ponto flutuante de 32 bits.

Manipulação de pixels em imagens

- Dentre as diversas funções presentes na biblioteca OpenCv, existe a função *at* que permite manipular diretamente um determinado pixel em uma imagem.
- Para acessar o pixel, é preciso criar um objeto do tipo cv::Mat, atribuir uma imagem a esse objeto e acessar o pixel passando as coordenadas x e y.
- A configuração padrão do cv::imread é a criação de uma matriz do tipo CV_8UC3.

Acessando pixels – Imagem Grayscale

- Como vimos anteriormente, as imagens em escala de cinza possuem somente um valor para cada pixel (geralmente de 0 a 255).
- Desta forma, o acesso ao pixel se dá diretamente pela função *at* passando dois parâmetros, a linha e a coluna, sendo que o tipo de dado mais comumente utilizado nessa situação é o *uchar*
- Exemplo:

```
uchar pixel = image.at<uchar>(linha,coluna);
```

Acessando pixels – Imagem Colorida

- Para acessar os pixels de uma imagem colorida, também podemos utilizar a função *at*, porém alterando o tipo de objeto retornado.
- Considerando uma imagem colorida de 3 canais de 8 bits cada, precisamos criar um objeto do tipo Vec3b para acessar o pixel com a informação das cores.

Acessando pixels – Imagem Colorida

- Exemplo:

```
Vec3b pixel = image.at<Vec3b>(linha,coluna)
```

Como uma variável do tipo Vec3b é um vetor de 3 posições, para acessar cada uma das cores do pixel utilize o índice do vetor, de 0 a 2. Lembrando que a ordem das cores no OpenCV é BGR.

Para alterar o conteúdo de um pixel, faça o procedimento inverso:

```
Vec3b vermelho{0, 0, 255};

image.at<Vec3b>(linha, coluna) = vermelho;
```