

Assignment2

视觉组考核 (OpenCV)

一些补充

对于每个像素都要特别赋值的情形，静态函数 `zeros()`，`ones()`，`eye()` 等难以满足需求。你可以利用重载的 `<<` 运算符：

```
1. cv::Mat kernel1 = (cv::Mat_<float>(3, 3) << 0, 1, 0,
2.                                     1, 0, 1,
3.                                     0, 1, 0);
```

或者将数据直接拷贝进去：

```
1. unsigned char rgb_data[12] = {1, 2, 3,
2.                               4, 5, 6,
3.                               7, 8, 9,
4.                               10, 11, 12};
5.
6. cv::Mat3b rgb_img(1, 4);
7. memcpy(rgb_img.data, rgb_data, 3 * 4 * sizeof(uchar));
```

图像滤波

图像滤波可以理解为根据图像全局或局部或两者综合的信息，为像素点赋予新值的行为

就较常使用的形式而言，可以分为两种：

$$\left\{ \begin{array}{l} \text{卷积滤波} \\ \text{排序滤波} \end{array} \right.$$

卷积滤波

卷积滤波涉及到**卷积核**(kernel)的概念，至于**卷积**本身，可以理解成加权求和。他们的严格定义可以自行学习。

- **均值滤波**是帮助理解的好例子：<https://zhuanlan.zhihu.com/p/76188487>
- **高斯滤波**同样可以用于图像平滑，其原理参考：<https://www.cnblogs.com/wj-1314/p/11693364.html>
以上链接也同时包含了均值滤波，中值滤波等的详细说明
- **bilateral filtering**：Bilateral filtering essentially applies a 2D Gaussian (weighted) blur to the

image, while also considering the variation in intensities of neighboring pixels to minimize the blurring near edges (which we wish to preserve). What this means is that the shape of the kernel actually depends on the local image content, at every pixel location.

你可以通过 **bilateral filtering** 实现简单的美颜算法(磨皮)

卷积滤波的特别应用：图像差分

对一元函数 $f(x)$ 求微分是耳熟能详的，同样的，可以对二元函数 $g(x, y)$ 求微分。因为图像是离散值，我们换一个名字，叫做**差分**。具体的理论涉及到高数下册的知识，但是其实现依然是通过类似如下的卷积核：

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

参考：<https://zhuanlan.zhihu.com/p/79153871>

图像差分的一个应用：**Edge Detection** [Sobel, Canny]

- 关于Sobel算子的教程中提到：*in the sample code below, you will notice that above the code for Sobel() function there is also code for the Scharr() function commented.*
但是在sample code中并无相关的内容。你需要仔细阅读sample code，思考一下这是为什么。[API的说明](#)同样可以告诉你这个答案。
- [Canny的API说明](#)中L2gradient这个参数你需要留意一下

还可以使用二阶差分：**Laplace**

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{\partial}{\partial x} \frac{\partial f}{\partial x} = \frac{\partial f}{\partial x} \Big|_x - \frac{\partial f}{\partial x} \Big|_{x-1} = f(x+1) - f(x) - [f(x) - f(x-1)] \\ &= f(x+1) + f(x-1) - 2f(x)\end{aligned}$$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$x\text{方向上: } \frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$y\text{方向上} : \frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\therefore \nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

模板实现为：

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

非常重要的函数： `cv::filter2D()`

`cv::filter2D()` 允许我们使用自己构造的卷积核对图像进行卷积，输入的卷积核即可以是一个小的Mat

一个例子：`cv::dilate()`(将在下文提到) 又称最大值滤波，是耗时的函数。若目标仅仅是粗暴地膨胀，可以尝试在`cv::filter2D()`中用一个全为1的卷积核

总结

以上几乎全部理论和API使用方法都已涵盖在以下网页中：

- <https://learnopencv.com/image-filtering-using-convolution-in-opencv/>
- <https://learnopencv.com/edge-detection-using-opencv/>

你已经熟悉：

- `cv::blur()`
- `cv::GaussianBlur()`
- `cv::Laplacian()`
- `cv::filter2D()`
- `cv::Sobel()`
- `cv::Canny()`

排序滤波

排序滤波的代表是**中值滤波**。同样，理应存在**最大值滤波**和**最小值滤波**。OpenCV中，它们的功能分别由两个函数`cv::dilate()`和`cv::erode()`执行

(https://docs.opencv.org/4.2.0/db/df6/tutorial_erosion_dilatation.html)

`cv::dilate()` 和 `cv::erode()`是形态学操作的重要基础

其他的非线性滤波

我们将形态学归纳到这一部分

Morphology Transformations

https://docs.opencv.org/4.2.0/d3/dbe/tutorial_opening_closing_hats.html

very important : Opening , Closing

Hit-or-Miss

https://docs.opencv.org/4.2.0/db/d06/tutorial_hitOrMiss.html

Image Thresholding

请学习 : <https://learnopencv.com/opencv-threshold-python-cpp/>

THRESH_OTSU 未被提及。这是一个有效的全局自适应方法，利用最大类间方差求取optimal threshold value。你需要学习如何使用它。

请思考一下，全局阈值具有什么缺点？什么样的图像不能被良好的分割？寻找这样的图像并测试

`cv::adaptiveThreshold()` 可以解决上面的问题吗？

位操作

from **bitwise_and()** to **bitwise_xor()**

思考：

```
1. cv::Mat_<uchar> input = cv::Mat_<uchar>::ones(99, 99) * 81;
2. cv::Mat_<uchar> output;
3. // 以下两句的结果是否相同？若相同，谁更高效？
4. cv::bitwise_not(input, output);
5. output = 255 - input;
```

提交要求

1. 一份学习笔记(PDF)
2. 学习笔记需要包含：
 - 简要总结新学习到的知识
 - **对 Image Thresholding 和 位操作 章节中问题的思考**
 - **心得体会**
 - 对于本教程的建议(Optional)
3. 截止日期：**见群通知**
4. 邮件格式：
 - 邮件以 **视觉考核-姓名-年级** 的格式命名
5. 发送至正确的邮箱