

Assignment1

视觉组考核 (OpenCV)

基础要求

为完成本篇及以后的任务，我们预设你已具有一定C++基础

请注意：在之前的培训中，你应当已经安装了**ROS noetic**，其自带**OpenCV 4.2.0**，因此若非特殊需求，你无需自行编译安装OpenCV

自主学习

1. 什么是 **OpenCV**
2. 了解数字图像的矩阵表示，理解RGB24的意义
3. 了解常用色彩空间：**RGB**，**HSV**，**LAB**，**YCrCb**
4. 了解OpenCV基本数据类型**Mat**及其关键成员函数，学会按需构造和复制Mat对象
5. 学会 `imread()` 和 `imwrite()` 以及 `imshow()`
6. 掌握多种访问任意像素、赋值的方法，特别是多通道图像
7. 学会测量算法运行时间，并比较上一点中几个方法的速度

学习指导

1. OpenCV – the most popular computer vision library in the world
2. 参考：<http://avisynth.nl/index.php/RGB24>
3. 参考：<https://learnopencv.com/color-spaces-in-opencv-cpp-python/>
在完成后面的内容后，你需要了解一下 `cv::split()` 和 `cv::merge()`
4. 参考：https://docs.opencv.org/4.2.0/d3/d63/classcv_1_1Mat.html

一般来说，熟悉以下常用构造函数即可

Public Member Functions

Mat ()
Mat (int rows, int cols, int type)
Mat (Size size, int type)
Mat (int rows, int cols, int type, const Scalar &s)
Mat (Size size, int type, const Scalar &s)

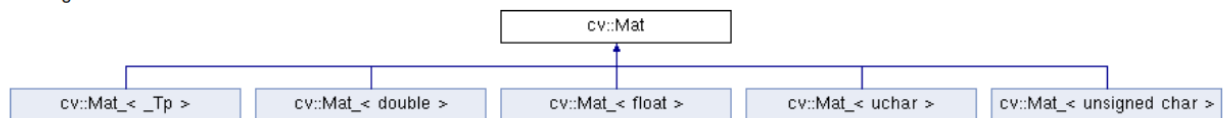
在 **Static Public Member Functions** 段落，你需要了解所有类型是 **static MatExpr** 的函数

！请注意：Mat类非常强大，其成员函数提供的功能涵盖了像素操作，浅拷贝和深拷贝，类型转换等重要项目。

以后你可能需要经常参考以上网页

同时，你需要留意这些派生类：

Inheritance diagram for cv::Mat:



阅读左起第一个即可。在 **Detailed Description** 段落，你将看到它的重要作用：

While **Mat** is sufficient in most cases, **Mat_** can be more convenient if you use a lot of element access operations and if you know matrix type at the compilation time. Note that **Mat::at(int y, int x)** and **Mat::operator()(int y, int x)** do absolutely the same and run at the same speed, but the latter is certainly shorter:

我们不推荐完全使用默认的 `cv::Mat` 来声明图像，很重要的一个原因就是难以直接看出图像的类型。对图像类型不严加控制有时会导致糟糕的错误。因此，可行时，使用 `cv::Mat_` 指明关键的信息，例如：

```
cv::Mat_<cv::Vec3b> red(400, 600); // 或者 cv::Mat3b red(400, 600);  
// 而非：cv::Mat green;
```

对于两个拷贝构造函数：

Mat_ (const Mat_ &m)
copy constructor More...

和 **Mat (const Mat &m)**

https://docs.opencv.org/4.2.0/df/dfc/classcv_1_1Mat_.html#a41f78862f27556e7543b91134b1111ce

https://docs.opencv.org/4.2.0/d3/d63/classcv_1_1Mat.html#a294eaf8a95d2f9c7be19ff594d06278e

请了解它们执行的是深拷贝还是浅拷贝

5. 参考：

https://docs.opencv.org/4.2.0/d4/da8/group_imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56
https://docs.opencv.org/4.2.0/d4/da8/group_imgcodecs.html#gabbc7ef1aa2edfaa87772f1202d67e0ce
<https://learnopencv.com/read-display-and-write-an-image-using-opencv/>

NOTE : when using `imread()`, In the case of color images, the decoded images will have the channels stored in **B G R** order

6. 使用 `at()` 或 `ptr()` ,或利用**Public Attributes**中的data和step

参考 : <https://cloud.tencent.com/developer/article/1457892>

对于Mat_还有 :

`Mat_::operator()(int y,int x)` : returns reference to the specified element (2D case)

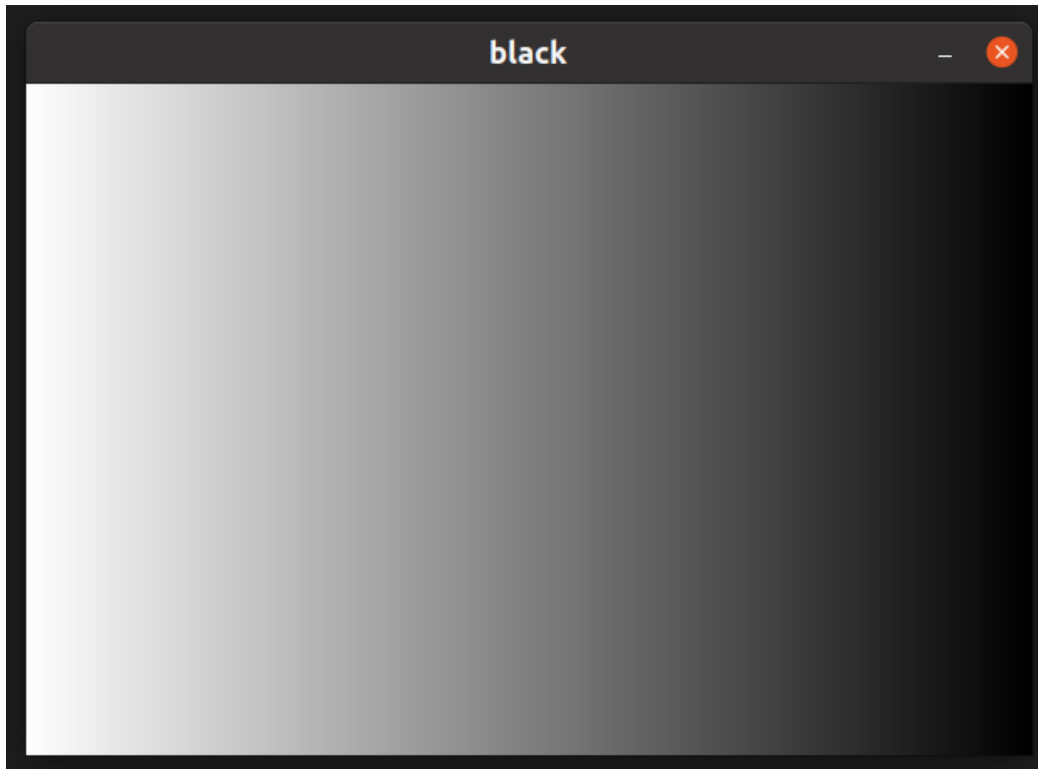
7. 使用 `cv::getTickCount()` 和 `cv::getTickFrequency()` , 参考同上

考核任务

利用OpenCV库, 编写C++代码, 实现简单的功能

1. 题目

1. 创建一张每个像素的值都是0的 400×600 的8位单通道图像, 显示它; 然后利用像素操作, 使其显示如下



2. 创建1张 400×600 的8位三通道图像, 任意显示三原色之一

3. 读取下图, 分割出图中的**棕色叉**和**绿色钩**, 其余图形选做



2. 样例与说明

1. 利用 `cv::Mat_<uchar> black = cv::Mat_<uchar>::zeros(400, 600);`
或 `static cv::Mat_<uchar> black(400, 600);`
2. 略
3. 分割方法自选



3. notes

ubuntu截图命令：

- **Shift(+Fn)+PrtSc**：选区截图保存到~/Pictures
- **(Fn)+PrtSc**：全屏截图保存到~/Pictures

重要的学习参考：

- <https://docs.opencv.org/4.2.0/>
- <https://learnopencv.com/>

以上两个网站包含大量的教程、案例和说明，请积极参考。不熟悉的API和功能，请于<https://docs.opencv.org/4.2.0/>上查询。若你是第一次接触OpenCV，可以从这里开始<https://learnopencv.com/getting-started-with-opencv/>

我们相信你顺利地完成了这些题目。请不妨试试访问越界的像素会发生什么，以及修改它又会如何？

你注意到了 `uchar * data`，请思考CV_32F等是如何实现的
pointer to the data More...

4. 提交要求

1. 你需要提交包含以下内容的zip压缩包：
 - 4道题目的代码(请养成写注释的好习惯)和运行结果截图
 - 一份学习笔记(**PDF**)(原始的书写推荐使用**markdown**，也可以使用 **LibreOfficeWriter** 等)
2. 学习笔记需要包含：
 - 题目1、2、3的思路
 - **心得体会(理论学习的一些笔记、你的思考和试错等)**
 - 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
 - 附件zip包命名同上
 - 若有未完成的题目，请在正文中说明
5. 发送至正确的邮箱