

# 用OpenCV和C++轻松玩转图像处理

作者: 陈若愚

日期: 2020年/2月/29日

## Abstract

图像处理是我们很多学生进入科研或者科创的一个选则, 因为他入门的门槛很低, 实践性很强, 而且不像仪器类学科需要科研设备或者深度学习需要强大的计算服务器。目前主流的图像处理是matlab, 因为matlab是数据科学常用的一门语言, 操作简单, 但是matlab在计算速度方面存在问题, 而且并不适合工业领域。C++, 面向对象的编程, 可以算是非常难的一门语言了, 但是由于很多软件工程师的努力, 我们现在可以非常简单地使用C++和Opencv开源软件进行数字图像处理, 希望我的学习记录能够让大家快速入门。

## 1. 图像读入、显示与保存

本人不习惯使用命名空间, 这里只记录一下命名空间与非命名空间的比较, 后面代码全部使用非命名空间研式

- 不含命名空间格式:

```
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc.hpp>

int main()
{
    // 工作台输出
    std::cout << "Hello OpenCV!" << std::endl;
    // 读入一张图片
    cv::Mat img = cv::imread("Lena.jfif");
    // 创建一个名为 "图片"窗口
    cv::namedWindow("Lena");
    // 在窗口中显示图片
    cv::imshow("Lena", img);
    // 等待6000 ms后窗口自动关闭
    cv::waitKey(6000);
    // 将图片保存为jpg格式
    cv::imwrite("Lena.jpg", img);
    return 0;
}
```

- 含命名空间格式:

```
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc.hpp>

using namespace std;
```

```
using namespace cv;

int main()
{
    // 工作台输出
    cout << "Hello OpenCV!" << endl;
    // 读入一张图片
    Mat img = imread("Lena.jfif");
    // 创建一个名为 "Lena"窗口
    namedWindow("Lena");
    // 在窗口中显示图片
    imshow("Lena", img);
    // 等待6000 ms后窗口自动关闭
    waitKey(6000);
    // 将图片保存为jpg格式
    imwrite("Lena.jpg", img);
    return 0;
}
```

- 输出结果:



并且可以看到保存的jpg格式的图片，以上便是最基本的读取，显示与保存图像。

## 2. 图像处理基础

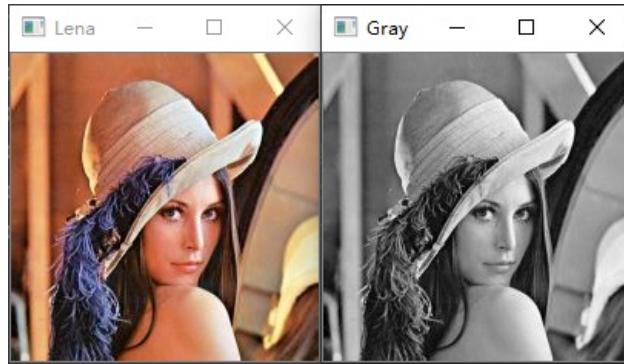
---

### 2.1 灰度图转化

```
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc.hpp>

int main()
{
    // 工作台输出
    std::cout << "Hello OpenCV!" << std::endl;
    // 读入一张图片
    cv::Mat img = cv::imread("Lena.jfif");
    // 定义灰度图
    cv::Mat gray_img;
    cv::cvtColor(img, gray_img, cv::COLOR_BGR2GRAY); //旧版用CV_RGB2GRAY
    // 在窗口中显示图片
    cv::imshow("Lena", img);
    cv::imshow("Gray", gray_img);
    // 手动窗口关闭后结束
    cv::waitKey(0);
    return 0;
}
```

- 输出结果:



## 2. 2 获取图像属性

```
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/opencv.hpp>
#include <opencv2/imgproc.hpp>

int main()
{
    cv::Mat img = cv::imread("Lena.jfif");
    //宽度
    int col = img.cols;
    std::cout << "宽度: " << col << std::endl;
    //高度
    int row = img.rows;
    std::cout << "高度: " << row << std::endl;
    //通道数
    int channel = img.channels();
    std::cout << "通道数: " << channel << std::endl;
    system("pause");
    return 0;
}
```

- 输出结果:

```
宽度: 200
高度: 200
通道数: 3
请按任意键继续. . .
```

## 2. 3 读取与修改像素

## 3. 图像运算

---

## 4. 类型转换

---

## 5. 几何变换

---

## 6. 直方图

---

7. 图像滤波

---

8. 阈值分割

---

9. 形态学操作

---

10. 图像梯度

---

11. Canny边缘检测

---

12. 傅里叶变换

---