
《OpenCV3 编程入门》勘误-2015.11

1

正文 P17 页。中间的 Note 部分。

将：

X64 表示运行系统环境位 64 位系统，若安装于 32 位系统，应为 X86；

替换为：

X64 表示运行系统环境为 64 位系统，若安装于 32 位系统，应为 X86；

2

正文 P35 页。中间部分读取并播放视频的示例程序代码，用判空的方式修正之前视频播放完成后内存溢出的错误。

将如下代码：

```
#include <opencv2\opencv.hpp>
using namespace cv;

int main()
{
    // 【1】读入视频
    VideoCapture capture("1.avi");

    // 【2】循环显示每一帧
    while(1)
    {
        Mat frame; //定义一个 Mat 变量，用于存储每一帧的图像
        capture>>frame; //读取当前帧
        imshow("读取视频", frame); //显示当前帧
        waitKey(30); //延时 30ms
    }
}
```

```

    }
    return 0;
}

```

替换为：

```

#include <opencv2\opencv.hpp>
using namespace cv;

int main( )
{
    // 【1】读入视频
    VideoCapture capture("1.avi");

    // 【2】循环显示每一帧
    while(1)
    {
        Mat frame;//定义一个 Mat 变量，用于存储每一帧的图像
        capture>>frame; //读取当前帧

        //若视频播放完成，退出循环
        if (frame.empty())
        {
            break;
        }

        imshow("读取视频",frame); //显示当前帧
        waitKey(30); //延时 30ms
    }
    return 0;
}

```

3

正文 P56 页。2.5.1 节的第二段。

将：

argc 和 argc...

替换为：

agrc 和 argv...

4

正文 P76 页底部。

将：

第二个参数，.....指定窗口里每次鼠标时间....

替换为：

第二个参数，.....指定窗口里每次鼠标事件....

5

正文 P89 页。顶部的一段文字中关于有符号型 char 类型的范围的描述有问题。需注意，signed char 的取值范围是 -128 ~ 127，并非之前书中写的-127 ~ 127。修改方式如下所述。

将：

最小的数据类型是 char，占一个字节或者 8 位，可以是有符号型（0 到 255 之间）或无符号型（-127 到+127 之间）。

替换为：

最小的数据类型是 char，占一个字节或者 8 位，可以是有符号型（0 到 255 之间）或无符号型（-128 到+127 之间）。

6

正文 P117 页。5.2.3 节，addWeighted 函数原型名称缺失。

将：

这个函数的作用是计算两个数组（图像阵列）的加权和。原型如下：

```
void (InputArray src1, double alpha, InputArray src2, double beta, double
gamma, OutputArray dst, int dtype=-1);
```

替换为：

这个函数的作用是计算两个数组（图像阵列）的加权和。原型如下：

```
void addWeighted(InputArray src1, double alpha, InputArray src2, double beta,
double gamma, OutputArray dst, int dtype=-1);
```

7

书本正文 P172 页。代码部分。

将：

```
//=====【<1>方框滤波】=====
//创建窗口
namedWindow("【<1>方框滤波】", 1);
//创建轨迹条
createTrackbar("内核值：", "【<1>方框滤波】",&g_nBoxFilterValue,
40,on_BoxFilter );
on_MeanBlur(g_nBoxFilterValue,0);
imshow("【<1>方框滤波】", g_dstImage1);
```

替换为：

```
//=====【<1>方框滤波】=====
//创建窗口
namedWindow("【<1>方框滤波】", 1);
//创建轨迹条
createTrackbar("内核值：", "【<1>方框滤波】",&g_nBoxFilterValue,
40,on_BoxFilter );
on_BoxFilter(g_nBoxFilterValue, 0);
```

8

正文 P192 页。中间部分。

将：

第五个参数，使用 erode()函数的次数

替换为：

第五个参数，使用 dilate()函数的次数

9

正文 P194 页。中间部分。

将：

第五个参数，使用 erode()函数的次数

替换为：

第五个参数，使用 dilate()函数的次数

10

正文 P194 页。下面部分，贴出代码的上面一段。

将：

使用 erode 函数，一般只需要填前面的三个参数.....

替换为：

使用 dilate()函数，一般只需要填前面的三个参数.....

11

正文 P370 页。on_Matching 函数的实现部分。

将：

```
// 【2】 初始化用于结果输出的矩阵
int resultImage_cols = g_srcImage.cols - g_templateImage.cols + 1;
int resultImage_rows = g_srcImage.rows - g_templateImage.rows + 1;
g_resultImage.create( resultImage_cols, resultImage_rows, CV_32FC1 );
```

替换为：

```
// 【2】 初始化用于结果输出的矩阵
int resultImage_rows = g_srcImage.rows - g_templateImage.rows + 1;
int resultImage_cols = g_srcImage.cols - g_templateImage.cols + 1;
g_resultImage.create(resultImage_rows, resultImage_cols, CV_32FC1 );
```