**线扫图像拆分功能开发说明文档**

**一、文件说明：**

1. splitScan.cpp:功能具体实现
2. splitScan.h:功能类与类内函数定义

**二、所需环境**

1、C++11

2、OpenCV 3.1.0+

**三、使用方法：**

1. 配置编译环境；
2. 创建功能类并初始化；
3. 设置参数(待拆分图像或图像路径、拆分数量);
4. 调用类内函数执行功能;
5. 清空类内已有数据；
6. ……(循环3-5)

**四、类与类内函数：**

//功能类

class splitScan

//构造函数1

splitScan();

//构造函数2

//参数1：待拆分的图片路径(包括图片名)

//参数2：需要拆出的数量(默认8)

splitScan(std::string \_path, int \_splitNum=8);

//构造函数3

//参数1：待拆分的Mat图像

//参数2：需要拆出的数量(默认8)

splitScan(cv::Mat \_orignal, int \_splitNum=8);

//析构函数

~splitScan();

//设置读图路径

//参数1：待拆分的图片路径(包括图片名)

void setPath(std::string \_path);

//设置数量

//参数1：需要拆出的数量

void setNum(int \_Num);

//设置待拆分的图像

//参数1：待拆分的Mat图像

void setOrignal(cv::Mat \_orignal);

//清空输入与计算的变量

void clear();

//获得待拆分的图像

//输出1：待拆分的Mat图像

cv::Mat& getOrignal();

//获得拆分出来的结果图像序列

//参数1：已拆分的Mat图像序列

std::vector<cv::Mat>& getSplited();

//将待拆分图像拆成指定数量的图像序列

//输出1：true--成功, false--失败

bool splitImages();

**五、示例程序**

//splitScan Test

string orignal\_path = "D:\\Matlab\\PhotometricStereo\\data\\photometricStereo\\20210728线扫分区\\3\\Orignal.jpg";

Mat orign = imread(orignal\_path, IMREAD\_GRAYSCALE);

splitScan tools(orignal\_path, 8);

splitScan tools(orign, 8);

vector<Mat> splited=tools.getSplited();

for (int i = 0; i < splited.size(); i++)

{

Mat temp = splited[i].clone();

string save\_path = "D:\\Matlab\\PhotometricStereo\\data\\photometricStereo\\20210728线扫分区\\4/splited\_" + to\_string(i) + ".jpg";

imwrite(save\_path, temp);

}

**六、注意事项**

1、为了方便后续光度立体的处理，内部保存的待拆分图像将被转换为灰度图；

2、同时存在待拆分图像与待拆分图像路径时，优先使用已存在的待拆分图像；设置待拆分图像路径后会立即读入指定的图像并覆盖已有的待拆分图像；

3、clear()函数会将待拆分数量置为默认值(8);

**最后更新时间2021-08-14**

**陈星寰**