**光度立体功能开发说明文档**

**一、文件说明：**

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

**二、所需环境**

1、C++11

2、OpenCV 3.4.0+

3、Eigen 3.3.0+

4、OpenMP 2.0+

**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(图像、检测设置、光源参数等);
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：**

//枚举值

//光源类型枚举

enum lightType //光源类型枚举

{

Point\_light = 0, //点光源

Circle\_light = 1, //环形光

Rect\_IMG = 2 //矩形光

};

//计算方式枚举

enum calcType //表面法向量计算方式枚举

{

minus\_calc = 0, //理想情况下

divide\_calc = 1, //带光源修正

iter\_calc = 2 //带光源修正迭代

};

//信息结构体

struct lightInfo //光源信息

struct detectInfo //检测设置信息

//功能类

class photometst

//类内函数

photometst(); //构造

photometst(getImages \_MatImages, detectInfo \_settings, lightInfo \_lights); //构造2

~photometst(); //析构

void setDirImgs(std::vector<cv::Mat> \_dirImg); //设置4方向图像

void setAllDirImg(cv::Mat \_alldirImg); //设置正常图像

void setStdImgs(std::vector<cv::Mat> \_standardImg); //设置标定图像

void setInfos(detectInfo \_settings, lightInfo \_lights); //设置信息

void clearCalc(bool IsAll = false); //清除内部计算(值)数据

void clearAll(); //清除所有(输入&计算)数据

////////////////////////////You may call///////////////////////////////////

std::vector<cv::Mat> getEp(); //返回光源亮度修正图像

std::vector<cv::Mat> getangleMap(); //返回光源角度修正图像

std::vector<cv::Mat> getlightAngle(); //返回光源角度修正矩阵(裁切后)

std::vector<cv::Mat> getprocedImgs(); //返回4方向图像(预处理后)

cv::Mat getIa(); //返回总和图像

std::vector<cv::Mat> getsurfaceNormals(); //返回表面法向量

cv::Mat getalbedo(); //返回总和图像

std::vector<cv::Mat> getcurvature(bool IsCalc4Height=false);//返回曲率图(是否由高度图计算)

cv::Mat getsurfHeight(); //返回表面高度图像

getImages MatImages; //4方向|正常|标定Mat图像

/////////////////////////////////Function/////////////////////////////////////////

bool lightCorrect(); //光源修正：光照强度、光照方向修正

bool preprocess(); //预处理：检测区域处理、ROI裁切、(可选的)光照强度修正

bool calcNormalVec(); //计算表面单位法向量

bool surfcurvature(bool isHeight = false); //计算曲率

bool calcSurface(); //计算高度图

**五、示例程序**

//photometst

getImages images("D:/Matlab/MyPhotometricStereo/Photometric-Stereo/Matlab/data");

//Mat fullIMG = images.getAllDirImg();

images.getDirImg("3/\_Dir\*.jpg");

images.getStdImg();

detectInfo settings(27, Rect(4200, 700, 300, 400), divide\_calc);//Rect(4200, 700, 300, 400)

lightInfo lights(0.0175, 77, 53, 230, 7, 90, Point\_light);

photometst toCalc(images, settings, lights);

vector<Mat> procImg = toCalc.getprocedImgs();

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

{

Mat temp2 = procImg[i];

string save\_path = "D:\\Matlab\\MyPhotometricStereo\\Photometric-Stereo\\Matlab/read\_" + to\_string(i) + ".jpg";

}

Mat Ia = toCalc.getIa();

Mat heightMap = toCalc.getsurfHeight();

Mat heightMap\_re;

reScaleImg(heightMap, heightMap\_re,0,1);

**六、注意事项**

1、OpenvCV的Mat类型变量名称多以Img/Mat结尾，Eigen的MatrixXd类型变量名称多以Mtx结尾;本功能类内部变量均保存为Eigen库中的MatrixXd类型，仅在需要时转化为Mat类型，与Matlab中的变量名称及其作用几乎一致。

2、枚举值中矩形光源(Rect\_IMG)和带光源修正迭代(iter\_calc)还有待研究与编写，调用上述值将无法计算结果；使用带光源修正(divide\_calc)需要额外指定光源信息与标定图像，使用理想条件下(minus\_calc)只需要输入待测4方向照明图像；

3、采用用先判断后执行的方式，如果未清空已有数据，则一直返回已有数据，请在希望更新数据前先清空已有数据；考虑到专用计算的需要，清除计算值时可选是否清除光源修正矩阵与稀疏矩阵(默认不清除)；

4、大矩阵（1800w像素）求解高度时，迭代求解器(求解器1)很慢：22s/100ite，但也是唯一可用的选项，迭代次数与误差：100ite/0.24、600ite/0.06、900ite/0.04；

5、在AMD Ryzen5 3600\16G主机中运行通过，其他硬件配置下未知。

**最后更新时间2021-08-25**

**陈星寰**