

# BmpCV

---

BmpCV：使用C++编写的BMP图像处理库

## 开发环境

CLion 2022.3, MinGW w64 9.0, Eigen 3.4.0, Qt 5.15.2

## 库模块

bmp：BMP图像的读取和保存

```
Img(); // 图像类

// 访问图像中像素
uchar Img::loc(int row, int col, int channel) const;
uchar Img::loc(pt_2d pt, int channel) const;
// 访问/修改图像中像素
uchar& Img::iloc(int row, int col, int channel);
// 读取图像
Img readImg(const std::string& filename);
// 保存图像
void writeImg(const Img& src_img, const std::string& filename);
```

gui：BMP图像的显示

```
// 创建窗口
void createWindow();
// 显示图像
void imgShow(const Img& src_img);
```

img\_proc：BMP图像的基本变换

```

// 旋转
Img rotateImg(const Img& src_img, double angle, int flag);
// 水平/垂直镜像
Img flipImg(const Img& src_img, int flag);
// 缩放
Img resizeImg(const Img& src_img, int size_x, int size_y, int flag);
Img resizeImg(const Img& src_img, double ratio);
// 剪切
Img getRoi(const Img& src_img, point_pair_2d p1, point_pair_2d p2);
// 平移
Img translationImg(const Img& src_img, int offset_x, int offset_y, pt_3d fill_color);
// 去色
Img removeColor(const Img& src_img);

```

## img\_proc\_2: 空域滤波及直方图操作

```

//平滑: f_size*f_size的低通滤波,邻域平均法
Img low_filter(const Img& src_img, int f_size);
//平滑: f_size*f_size的低通滤波,超限邻域平均法T=lim
Img low_filter_p(const Img& src_img, int f_size, int lim);
//平滑: f_size*f_size的中值滤波
Img med_filter(const Img& src_img, int f_size);
//锐化: 3*3的拉普拉斯锐化;l_flag=1是变形模板+高增益滤波;=0或是基本模板
Img Laplace_sharpen(const Img& src_img, int l_flag);
//(归一化后的)直方图的计算及显示
Img show_histogram(const Img& src_img);
//辅助函数,在直方图搜索中调用,灰度图每个灰度的像素值个数
int (*get_vol(const Img& src_img))[256];
//辅助函数,在直方图均衡化中调用,目的是保存灰度映射关系
int *gray_map(int *a, int h_size, int w_size);
//直方图均衡化
Img equalize_histogram(const Img& src_img);
//根据直方图,搜索图片:num是查找图片的数目
Img hist_search(const Img& src_img, int num);

```

## transform: FFT、DCT及逆变换

```

// 图像尺寸填充至2整数次幂
Img imgPadding(const Img& src_img);
// 一维FFT
std::vector<Cp<double> > imgFFT1D(std::vector<Cp<double> > vector, int length, int
flag)
// 二维FFT
Cp_mat imgFFT2D(const Img& src_img);
// FFT结果中心化
Img imgFFTShift(const Cp_mat& mat, int return_flag);
Cp_mat imgFFTShift1(const Cp_mat& mat);
// 二维IFFT
Img imgIFFT2D(const Cp_mat & mat);
// 二维DCT
double_mat imgDCT2D(const Img& src_img);
// 二维DCT结果显示
Img imgDCTNormalize(const double_mat& mat);
// 二维IDCT
Img imgIDCT2D(const double_mat& mat);

```

filter: 频域滤波

```

// 创建滤波器 函数命名规则: create + 滤波器名 + 高/低通 + Kernel 如:
Eigen::MatrixXd createButterLowKernel(const Img& src)img, double D0, int n);
// 对图像作填充和FFT后, 使用特定频域滤波器滤波, 返回频域滤波后图像
Img freqFilterImg(const Img& src_img, const Eigen::MatrixXd& kernel);

```

Qt\_GUI: gui模块调用的前端GUI界面, 调用路径在setting.ini中设置。Qt\_GUI.exe为调用gui模块进程的子进程, 使用Windows 匿名管道实现进程间通信。界面目前支持图片的显示、保存、取色器和img\_proc中实现的基本变换功能。