

基于 OpenCV 图像处理系统的开发与实现

刘培军¹, 马明栋², 王得玉²

(1. 南京邮电大学 通信与信息工程学院, 江苏 南京 210003;

2. 南京邮电大学 地理与生物信息学院, 江苏 南京 210023)

摘要: 在数字图像处理技术的研究过程中, 基于平时对少量图像进行处理时, 一般都是针对不同需求运用各种图像处理算法进行编程, 存在着许多重复性工作。而当数字图像数据量非常大且对处理的图像实时性要求非常高时, 对所有图像进行处理, 代码以及算法的实现都通过人工进行编写, 存在着相当大的难度且效率十分低下。为有效解决上述问题, 以 OpenCV 计算机视觉库为基础, 在 Qt Creator 集成开发环境下, 采用面向对象编程, 开发并实现了一款界面良好, 方便用户使用的图像处理系统。通过对大量数字图像处理的实践表明, 该系统能够有效快速地对图像进行处理。

关键词: 图像处理算法; OpenCV; Qt Creator; 面向对象; 图像处理系统

中图分类号: TP302

文献标识码: A

文章编号: 1673-629X(2019)03-0127-05

doi: 10.3969/j.issn.1673-629X.2019.03.027

Development and Implementation of OpenCV-based Image Processing System

LIU Pei-jun¹, MA Ming-dong², WANG De-yu²

(1. School of Telecommunications & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. School of Geographical and Biological Information, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

Abstract: In the research of digital image processing technology, when a small number of images are processed, various image processing algorithms are generally used for programming according to different requirements, and there are a lot of repetitive work. However, when the amount of digital image data is large and the real-time requirement of processed images is high, all images are processed, codes and algorithms are written manually, which is quite difficult and inefficient. In order to effectively solve the above problems, based on OpenCV computer vision library and under the integrated development environment of Qt Creator, we develop and implement an image processing system of object-oriented programming with better interface and convenient for users. The practice of digital image processing shows that the system can process images efficiently and quickly.

Key words: image processing algorithm; OpenCV; Qt Creator; object-oriented; image processing system

0 引言

在计算机技术如此成熟的今天, 图像处理理论与技术已经得到了较为全面的发展并在持续不断的优化创新中, 尤其在人工智能、航空航天、工业部件检测、交通、医学等领域得到了广泛应用^[1-2]。目前国内外都有许多优秀的图像处理系统软件, 但由于这些系统的开发环境、应用目的和侧重点不同, 在实际的图像处理过程中都出现了不同程度的问题, 如通用性差, 功能缺

失, 图像处理速度慢, 不能处理多种图像格式, 等等。因此根据目前研究与学习的要求设计出一种图像处理系统是一种较为可行的方法。

文中详细介绍了在 Qt Creator 集成开发环境下, 基于 OpenCV 进行面向对象编程, 开发的一种功能齐全, 运行速度快, 可扩展性强, 可以在不同系统上运行的图形处理系统。其中 OpenCV 计算机视觉库作为常用的图像处理库之一, 功能强大, 支持目前先进的图像

收稿日期: 2018-04-24

修回日期: 2018-08-28

网络出版时间: 2018-12-20

基金项目: 国家自然科学基金(11501302)

作者简介: 刘培军(1993-), 男, 硕士研究生, 研究方向为图像处理与图像通信; 马明栋, 博士, 教授, 研究方向为地理信息系统平台软件设计与开发等; 王得玉, 博士, 副教授, 研究方向为水环境遥感、GIS 软件设计与开发。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20181219.1542.072.html>

处理技术且体系十分完善,几乎涵盖了近十年内的主流算法^[3]。

1 OpenCV 在 Qt Creator 环境下的配置

1.1 OpenCV

如今市面上存在众多处理图像和视频的软件和库,其中 OpenCV、CvImage、CImg、FreeImage 是几种常用的图像处理库,但是对于期望自己开发图像处理软

件的人而言,OpenCV 库是最好的选择。OpenCV (open source computer vision) 是一个开源可跨平台的图像及视频分析库,包含 500 多个优化过的算法^[4-5]。自 OpenCV2 发布后,相较于 1.0 版本其带来了重要的变化,尤其是使用了崭新的 C++ 接口^[6]。自版本 2.2 开始,OpenCV 被划分了多个模块。各个模块的名称及功能如表 1 所示。

表 1 OpenCV 库的主要模块

模块	功能
opencv_highgui 模块	包含了读写图像及视频的函数,以及操作图形用户界面的函数
opencv_core 模块	包含核心功能,尤其是底层数据结构和算法函数
opencv_imgproc 模块	包含图像处理函数
opencv_features2d 模块	包含兴趣点检测子、描述子以及兴趣点匹配架构
opencv_calib3d 模块	包含相机标定、双目集合估算以及立体视觉函数
opencv_video 模块	包含运动估计、特征跟踪以及前景提取函数与类
opencv_objdetect 模块	包含物体检测函数,如脸部与行人检测

1.2 开发环境配置

OpenCV 是一个跨平台计算机视觉库,需要将它编译成二进制的静态库和动态库后,才可以在 IDE 下直接调用图像处理函数。编译 OpenCV 需要用到开源跨平台自动化构建工具 CMake,它可以跨平台生成各种 makefile 或 project 文件支持用各种编译工具生成可执行程序或链接库。Qt Creator 支持 CMake,因此可以用 Qt Creator 在图形界面中对 OpenCV 进行编译。下面是开发环境的配置过程。

(1) 安装 CMake,安装时选择为所有用户添加系统路径,Qt Creator 可以自动找到 CMake,安装 MinGW 版本的 Qt Creator,安装 OpenCV。

(2) 打开 Qt Creator,工具→选项→构建和运行→CMake,确认 Qt 成功找到 CMake,否则手动指定 CMake 的路径。

(3) 在 Qt 中点击 [Open Project],打开 OpenCV 安装目录下的 CMakeLists.txt,点击 [Configure Project] 按钮。进入 [项目],编译构造配置选择 Release,待 CMakeLists.txt 读取完毕,找到 CMAKE_INSTALL_PREFIX,设置编译完成后的输出路径。然后进行构建步骤,勾选 install,点击开始构建。最后显示“正常退出”表示 OpenCV 已经编译完成。文中设计的图像处理系统采用 Qt 5.6.0 与 OpenCV 2.4.12 搭建开发环境。

(4) 将编译完成文件目录下的 bin 文件夹路径添加到系统环境变量 Path 中。

基于以上步骤,开发环境配置完成,在新建 Qt 项目.pro 文件中添加头文件路径和静态库路径以此来

使用 OpenCV 中的图像处理函数。

2 系统总体设计

该系统主要由两个核心模块组成:图像处理模块与显示模块。图像处理模块作为系统最为重要的功能部分,主要负责接收用户指令针对图像进行特定处理^[7];显示模块即在界面上显示原始图像或处理完成后的图像。除了上述两大功能模块外,图像的载入与保存也是该系统必不可少的部分。系统采用 Qt Creator 中的可视化图像编辑器进行 UI (user interface) 设计,界面简洁清晰,具有良好的交互性^[8]。通过对话框操作的形式就可以实现对图像进行相应的处理,而无需考虑具体实现原理,简化了处理图像的工作流程,提高了工作效率。

2.1 系统功能

系统的基本功能主要包含文件处理、基本图像处理、形态学处理、领域处理、点的处理等。

(1) 文件处理:目前图像文件的格式有许多种,文件处理主要是对这些不同格式的图像文件进行打开、恢复图像、关闭、保存当前窗口图像。系统对于打开文件应有两种方式,一是在系统“文件”菜单栏下通过“打开”按钮对图像文件进行载入,二是将图像拖入系统显示窗口进行打开。

(2) 基本图像处理:在图像处理的过程中应先对图像进行一些基本的处理,比如灰度化图像,红(R)、绿(G)、蓝(B)通道的提取,以及图像二值化,其常见算法如 OSTU 法、百分比阈值(P-Tile)法等。

(3) 形态学处理:膨胀与腐蚀是两种最基本也是

最重要的形态学运算,可以消除图像噪声,寻找图像中明显的极大极小值区域,分割独立图像元素,以及连接相邻元素^[9]。运用膨胀与腐蚀这两个基本操作,可以实现较高级的形态学变换,如开运算、闭运算、顶帽与黑帽等。

(4) 领域处理: 图像在采集、传输的过程中不可避免地会受到外界的干扰产生各种噪声和失真。为了抑制噪声,尽量保持图像细节特征,需要对图像进行平滑处理^[10]。边缘检测是领域处理中极为重要的一环,大多数情况下在平滑处理的基础上进行,其主要目的是为了标识图像中亮度变化明显的部分^[11]。图像处理中有多种边缘检测算子,如 Canny 算子、Sobel 算子、拉普拉斯算子等。

(5) 点的处理: 包含几种常用的几何变换操作,如水平翻转、垂直翻转、各种角度旋转等。此外还包含了图像直方图、直方图均衡化等基本图像增强操作。

(6) 自定义阈值化: 图像阈值化是一种最常用的图像分割方法^[12],所谓阈值化就是当图像像素点矩阵中的每个像素点的灰度值小于自定义阈值 T 时为 0 (黑色),大于阈值 T 时为 255 (白色),使图像转化为二值(黑白)图像。

(7) 图像变换: 包括颜色空间变换与图像正交变换。在图像处理中,不同的颜色空间有着不同的特性,为了便于分析各种图像的特性,在实际中需要对图像进行不同颜色空间的转换。常见的颜色空间有 RGB、HSV、Lab、YCrCb 等。图像的正交变换主要是将图像由空间域转换到另一个域进行更方便有效的分析与处理,如傅里叶变换是将图像转变为频域信号,进而对频谱进行分析^[13]。除了傅里叶变换以外,还有小波变换、K-L 变换等。

(8) 其他图像处理: 包含一些特定的角点检测算法,霍夫变换的直线与圆检测,文字提取,亮度对比度调整等一些综合图像处理。

2.2 界面设计

在设计界面时,系统界面应简洁清晰,方便用户使用。根据系统功能需求,设计相应的界面功能模块,主要有菜单栏、图像显示模块,以及其他一些功能显示模块。菜单栏中主要是图像的打开、关闭与保存,以及相应的图像处理功能菜单。图像显示模块是显示打开图像,以及显示处理完成后的图像。其他显示模块,可以设计为显示图像文件的直方图,以及图像的傅里叶变换等,方便用户对图像进行研究。在设计系统界面的过程中应遵循以下几点:

(1) 菜单栏应和功能设计保持一致,菜单名称与相应图像处理功能要一一对应,不能出现菜单名称与处理功能不匹配的现象。

(2) 在菜单层次设计中,子菜单划分应与上一层菜单呈包含关系,方便用户能快速找到相应的图像处理功能。

(3) 各显示模块都能准确显示。

3 系统实现

文中的图像处理系统是在 Windows7 平台下进行开发,采用的开发环境为 Qt Creator。Qt Creator 是一款功能十分强大的集成开发环境,主要包含项目模板生成、代码编辑、UI 设计、QML 界面编辑等丰富功能^[8]。Qt Creator 采用 C++ 面向对象设计,具有跨平台、界面设计美观等特点,因此得到了广泛的应用。

在系统实现的过程中,首先在 Qt Creator 集成开发环境中新建 Qt Widgets Application 窗体项目,该项目中包含了一个基于 Qt 设计师的主窗体,因此可以用于开发传统基于部件的窗体界面程序。

在窗体项目创建的过程中,在类信息设置界面,有三个最为关键的基类选择,分别是 QMainWindow、QWidget、QDialog。

QMainWindow 类: 基于主窗口类的程序,一般用于较为复杂的应用程序。它提供了一个有菜单栏、工具栏、状态栏以及多个可停靠的工具对话框的主应用程序窗口。

QWidget 类: 最简单最基本的窗体程序,里面可以放置多个控件实现程序功能^[14]。

QDialog 类: 是对话框窗口的基类,一般用于弹窗来执行短期任务,也可以用于主界面显示,但没有菜单栏、工具栏、状态栏等。

基于以上叙述,该系统采用 QMainWindow 类来实现主体界面的实现,打开项目中 ui 文件,Qt Creator 会自动进入设计模式,可以对图像界面进行可视化编辑。经过对系统功能分析,以及界面设计需求,系统界面如图 1 所示。

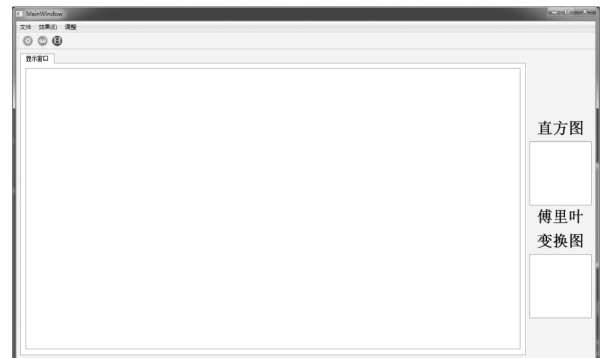


图 1 系统界面

3.1 图像输入输出与保存

从外部文件载入图像,图像处理完成后结果的显示以及存储,要用到 OpenCV2 中的 imread、imshow、

imwrite 函数^[4]。imread 函数用于读取一幅图像,并将其转换成一个 Mat 对象,它有两个参数,其函数原型为:

```
Mat imread( const string& filename , //要加载的图像,一般使用绝对路径
int flags=1//指定加载图像颜色类型,flags>0 三通道彩色图像
//flags=0 灰度图像
//flags<0 包含 Alpha 通道加载的图像
);
函数 imshow 是在特定窗口上显示图像,其函数原型为
void imshow( const string& winname ,//窗口名称
InputArray image //要显示的图像
);
函数 imwrite 是将图像保存到指定目录路径,其函数原型为
bool imwrite( const string& filename ,//写入要保存图像文件名
InputArray img ,//保存的图像,类型为 Mat
const std::vector<int>& params = std::vector<int>() //特定格式保存的参数编码,一般为默认值
);
该图像处理系统打开图像如图 2 所示。
```



图 2 图像系统展示

3.2 图像处理

在图像处理的过程中,根据对图像处理目的的不同,一般先对图像进行预处理(平滑、灰度化、二值化)然后运用各种图像处理算法,比如形态学处理、边缘检测、几何变换等。由于图像处理算法过多,下面仅以大津法(OSTU)二值化为例,并在系统中显示图像处理后的结果,如图 3 所示。

```
Matmy_class::Get_OSTUThresholdIMG( Mat SrcImg3C) //大津法二值化
{
MatSrcGray;
cv::cvtColor( SrcImg3C ,SrcGray ,CV_BGR2GRAY);
doubleThre_value;
int * HistGram =new int [256];
IplImage SrcGray_1 =IplImage( SrcGray);
GetHistGram( &SrcGray_1 ,HistGram); //获取直方图
Thre_value =GetOSTUThreshold( HistGram); //大津法
```

```
MatthreIMG;
threshold( SrcGray ,threIMG ,Thre_value 255 0); //图像二值化
SrcGray. release();
Mat ReturnIMG3C;
if( threIMG. channels() == 1)
{
cvtColor( threIMG ,ReturnIMG3C ,CV_GRAY2BGR);
}
else
ReturnIMG3C =threIMG;
threIMG. release();
return ReturnIMG3C;
}
```



图 3 大津法二值化结果

3.3 系统测试与验证

在 3.1 节与 3.2 节中,系统总体设计都已实现,而系统测试与验证是系统开发过程中的最后阶段,它是检验一个系统是否满足预期设计的必要手段。本节主要从界面交互性、功能模块稳定性、显示模块准确性、扩展性四个方面对该系统进行测试与验证。

(1) 界面交互性。

从上两节可以看出该系统界面简洁清晰,布局合理。打开菜单栏,菜单栏与子菜单栏按钮名称准确,显示规范,层次分明,方便用户操作,有较好的交互性,符合界面设计要求。

(2) 功能模块稳定性。

在处理图形的过程中,系统各个处理功能均能正常运行且运行稳定,反应迅速,没有出现程序意外关闭、崩溃等情况,有效提高了工作效率。由于图像处理功能较多,下面列出了部分功能测试结果以及部分效果图,如图 4 所示。

形态学处理:膨胀、腐蚀、开闭运算、顶帽、黑帽图像处理结果显示准确,达到设计要求。

几何变换:图像在系统中能够翻转,多角度旋转与设计一致。

边缘检测:系统中三种边缘检测算法,Sobel、Canny、Laplacian 算子均能准确实现,符合预期设计。

(3) 显示模块准确性。

经测试,系统图像显示模块均能准确显示与处理功能相匹配的图像,图像直方图与傅里叶变换图也能准确显示。

(4) 扩展性。



(a)形态学腐蚀



(b)图像 60° 旋转



(c)Canny 边缘检测



(d)油画效果

图4 测试效果

4 结束语

采用 OpenCV 研究开发的图像处理系统,实现了许多图像处理方面的通用算法,可以满足对图像的大多数处理需求^[15-16]。由于 OpenCV 具有非常高的通用性,并在 Qt Creator 下实现系统的设计与开发,因此该系统具有较好的可扩展性与可移植性。经测试验证,该系统操作简单,界面友好,能根据不同需求有效地处理不同格式的图像,提高了工作效率。

参考文献:

- [1] 王晨旭. 浅谈计算机图像处理技术的应用[J]. 计算机光盘软件与应用, 2011(6): 139.
- [2] 杜 聪. 浅谈 OpenCV 计算机视觉库[J]. 科技资讯, 2016, 40(28): 7-8.
- [3] BRADSKI G, KAEHLER A. Learning OpenCV [M]. [s. l.]: O'Reilly Media Inc. 2008.
- [4] ZELINSKY A. Learning OpenCV-computer vision with the OpenCV library [J]. IEEE Robotics and Automation Magazine, 2009, 16(3): 100.
- [5] 陈胜勇, 刘 盛. 基于 OpenCV 的计算机视觉技术实现 [M]. 北京: 科学出版社, 2008.
- [6] LAGANIERE R. OpenCV2 计算机视觉编程手册 [M]. 张 静, 译. 北京: 科学出版社, 2009.
- [7] 余松煜, 周源华, 张 瑞. 数字图像处理 [M]. 上海: 上海交通大学出版社, 2007.
- [8] BLANCHETTE J, SUMMERFIELD M. C++ GUI Qt4 编程 [M]. 闫锋欣, 曾泉人, 张志强, 等, 译. 北京: 电子工业出版社, 2011.
- [9] 贾小军, 喻擎苍. 基于开源计算机视觉库 OpenCV 的图像处理 [J]. 计算机应用与软件, 2008, 25(4): 276-278.
- [10] 景晓军. 图像处理技术及其应用 [M]. 北京: 国防工业出版社, 2005.
- [11] 阴国富. 基于阈值法的图像分割技术 [J]. 现代电子技术, 2008(23): 107-108.
- [12] 孙玉胜, 白 克. 一种基于改进的形态学红外图像边缘检测算法 [J]. 计算机与数字工程, 2010, 38(2): 127-129.
- [13] HE K, SUN J, TANG X. Guided image filtering [C]//European conference on computer vision. [s. l.]: [s. n.], 2010: 1-14.
- [14] 殷立峰, 祁淑霞, 房志峰. Qt C++ 跨平台图形界面程序设计基础 [M]. 第 2 版. 北京: 清华大学出版社, 2014.
- [15] 马新明. 基于 OpenCV 的图像处理系统设计与实现 [J]. 河南大学农业学报, 2014, 30(1): 87-89.
- [16] 管焱然, 管有庆. 基于 OpenCV 的仿射变换研究与应用 [J]. 计算机技术与发展, 2016, 26(12): 58-63.