

基于 OpenCV 的汉字绘制研究

柴树杉^{1,2}

(1. 中国地质大学 信息工程学院, 湖北 武汉 430074; 2. 航天远景科技有限公司, 湖北 武汉 430223)

摘 要 :OpenCV 是一种开放源代码的计算机视觉库,在图像处理领域有着广泛的应用。但 OpenCV 只提供了西文字符的输出,并不支持汉字的输出,这就给某些基于 OpenCV 的图像处理系统的开发带来了一定的困难。在对 OpenCV 研究的基础上,基于 FreeType 的字库提取技术实现了 OpenCV 中汉字的输出。该方法和传统的汉字处理方法相比,有很好的可移植性,可运行与 Windows/Linux 等系统,对于其它环境下汉字的输出实现也有一定的指导意义。

关键词 :OpenCV; 计算机视觉; 汉字; TrueType; FreeType

中图分类号 :TP391.41; TP391.11 文献标识码 :A 文章编号 :1000-7024 (2008) 19-5093-03

Research on drawing Chinese characters by OpenCV

CHAI Shu-shan^{1,2}

(1. Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China;

2. VisionTek, Wuhan 430223, China)

Abstract : As a Intel open source computer vision library, OpenCV has already been used in every field, such as image processing and so on. But OpenCV only provides the draw method of western characters, does not support the draw of Chinese characters, and brings many difficulties to the development of some image processing systems. Based on analysis of OpenCV and the FreeType technology, the Chinese characters draws in OpenCV are realized. This technology is also used in many other systems like Linux.

Key words : OpenCV; computer vision; Chinese character; TrueType; FreeType

0 引 言

由 Intel 资助开发的开放源代码库 OpenCV 是计算机视觉方面的一个基础类库,它封装了图形图像处理方面的一些基本算法,具有很强的实用价值。但是 OpenCV 提供的文本绘制功能并不支持汉字输出,这就给用户使用 OpenCV 开发计算机视觉方面的应用带来了困难。本文对 OpenCV 中如何实现汉字的输出进行了探讨。

1 OpenCV 的文本功能

利用 OpenCV 可以很容易地在影像中绘制几何图形或者是文本信息。OpenCV 中与文本输出相关的函数有 cvInitFont、cvPutText、cvGetSize 这 3 个。

1.1 InitFont

初始化字体结构体。函数定义为:

```
void cvInitFont (CvFont* font, int font_face, double hscale, double vscale, double shear, int thickness, int line_type);
```

font 表示被初始化的字体结构体。font_face 字体名称标识符,目前仅支持 CV_FONT_HERSHEY_SIMPLEX 等少数几种字体。hscale 表示字体宽度的放大系数,例如 0.5f 表示宽度

缩小为原来字体宽度的一半。vscale 字体高度的放大系数,意义和 hscale 相似。shear 字体的斜度的斜率,例如 1.0 的斜率表示字体倾斜 45 度。thickness 表示字体笔划的粗细程度。line_type 表示字体笔划的类型,具体参见 cvLine。

函数 cvInitFont 初始化字体结构体,字体结构体可以被传递到文字显示函数中。

1.2 PutText

在图像中显示文本字符串。函数定义为:

```
void cvPutText(CvArr* img, char* text, CvPoint org, CvFont* font, CvScalar color);
```

img 表示要绘制文本的图像。text 表示要显示的字符串,这里是严格的 ASCII 字符,不能接受像中文的 GB2312 等双字节编码。org 表示第 1 个字符左下角的坐标。font 表示字体结构体。color 表示文本的字体颜色。

函数 cvPutText 将具有指定字体的和指定颜色的文本加载到图像中。加载到图像中的文本被感兴趣的矩形框(ROI rectangle)剪切。不属于指定字体库的字符用矩形字符替代显示。

1.3 GetTextSize

获得字符串的宽度和高度。函数定义为:

```
void cvGetTextSize(char* text, CvFont* font, CvSize* text_
```

收稿日期:2007-10-10 E-mail: chaishushan@gmail.com

基金项目:科技型中小企业技术创新基金项目(0177N84417727000006662142)。

作者简介:柴树杉(1983-),男,江苏沐阳人,硕士研究生,研究方向为计算机视觉、汉字字库读写、OpenCV 扩展库等。

size, int* baseline);

font 对应字体结构体。text_string 表示要输入字符串。text_size 表示合成字符串的字符的大小,文本的高度不包括基线以下的部分。baseline 表示相对于文字最底部点的基线的 Y 坐标。

函数 cvGetTextSize 是用于在指定字体时计算字符串的绑定区域(binding rectangle)。

1.4 存在的问题

虽然由 OpenCV 所提供的上述 3 个函数可以实现许多基本的功能。但是在用 cvPutText 输出一个含有双字节的字符串时,却不能得到正确的信息(例如“在 OpenCV 中输出汉字!”中的汉字将被显示为‘?’)。如图 1 所示。



图 1 在 OpenCV 中输出汉字错误

2 绘制汉字的原理

OpenCV 文本输出函数 cvPutText 的源代码位于 cxcvcore\src\cxdrawing.cpp。通过分析代码可知, cvPutText 在处理字符时先获取字符的轮廓坐标,然后通过一个名为 icvPolyLine 的内部函数绘制字符的轮廓形状,如果遇到的是非 ASCII 字符则用‘?’字符简单替换。处理汉字输出也是采用相似的思路:解析汉字的编码,根据汉字编码装载汉字的形状,输出汉字的形状。

2.1 汉字的编码

由于西文字符与汉字在计算机内的编码方式是不同的,所以在处理字符串时需要对汉字进行正确的解码操作。目前,我国一般是采用 GB2312 编码。GB2312 又称为 GB2312-80 字符集,全称为《信息交换用汉字编码字符集·基本集》,由原中国国家标准总局发布,1981 年 5 月 1 日实施,是中国国家标准的简体中文字符集。它所收录的汉字已经覆盖 99.75% 的使用频率,基本满足了汉字的计算机处理需要。在中国大陆和新加坡获广泛使用。Unicode 编码是一个国际标准编码,与 ASCII 兼容(更准确地说,是与 ISO-8859-1 兼容),与 GB 码并不兼容(GB2312、GBK、GB18030 等)。例如“汉”字的 Unicode 编码是 6C49,而 GB 码是 BABA。

GB 码到 Unicode 编码的转换一般可以通过标准的编码映射表来完成。不同字符集之间的转换是一个相对复杂的问题,不过标准的 ANSI-C 语言已经提供不同字符集之间转换的函数。由于 C 语言默认的是采用标准 ASCII 字符集,因此在转换编码之前需要设置当前的编码环境,然后在进行编码转换。下面代码给出了将 char 数组到 wchar_t 数组的转换过程:

```
char bs[10] = "汉字编码";
wchar_t ws[10];
setlocale(LC_ALL, "");
mbstowcs(ws, bs, 10);
```

2.2 提取字型

汉字的字型一般都是保存在各种字库文件中。目前比较流行的是使用 TrueType 字库,对应的字库后缀名为 ttf。在 TrueType 字库中,字型的信息是通过一系列的点来描述的,这些点之后通过直线段或者通过二次贝塞尔曲线来连接,从而形成汉字的轮廓。

由于 TrueType 的具体格式比较复杂,编程实现对 TrueType 字库的提取也相对困难。目前,一个开源的字库文件处理库 FreeType 提供了对 TrueType 字库操作的基本 API,可以方便实现从 TrueType 字库中提取字型轮廓的操作。FreeType 是一个跨平台的字库文件处理库,可以在 Windows/Linux/VxWorks 等多种平台下编译使用。因此,基于 FreeType 实现的 OpenCV 实现的汉字输出功能并不会影响 OpenCV 本身的跨平台特性。

用 FreeType 的 API 提取一个字符的过程如下:

```
int main(){
    FT_Library m_library;    // 字库
    FT_Face m_face;    // 字体
    // 打开字库文件,创建一个字体
    FT_Init_FreeType(&m_library);
    FT_New_Face(m_library, "TrueType.ttf", 0, &m_face);
    // 根据字符的编码得到字符的索引位置
    // 根据字符在字库中索引得到字库轮廓
    // 根据字符的轮廓渲染字符的二值位图
    FT_UInt glyph_index=FT_Get_Char_Index(m_face,'中');
    FT_Load_Glyph(m_face, glyph_index, FT_LOAD_DEFAULT);
    FT_Render_Glyph (m_face->glyph, FT_RENDER_MODE_MONO);
    // 释放 FreeType 库管理的资源
    FT_Done_Face (m_face);
    FT_Done_FreeType(m_library);
    return 0;
}
```

FreeType 库的使用过程一般是先初始化库 (FT_Init_FreeType),然后从字库生成一个字体 (FT_New_Face),然后根据字符的 Unicode 编码从字库中装载字体。具体的细节在代码注释中已经给出。这里需要强调的一点是,上面的代码中 m_face->glyph->bitmap 对应的是当前装载字体的位图。

2.3 绘制汉字

上节中提取的字型对应的是字型的位图(m_face->glyph->bitmap),因此直接把字符对应的位图像素绘制到影像中即可。影像中每个像素的颜色可以通过 OpenCV 中的 cvSet2D 函数设置。实现代码如下:

```
void drawCharacter(IplImage *image, FT_Face *face){
    glyphSlot slot = m_face->gly
    FT_GlyphSlot slot = m_face->glyph;
    int rows = slot->bitmap.rows;
    int cols = slot->bitmap.width;
    for(int i = 0; i < rows; ++i){
        for(int j = 0; j < cols; ++j){
```

```

int off = i* slot->bitmap.pitch+ j/8;
int flag = slot->bitmap.buffer[off] & (0xC0>>(j%8));
cvSet2D(img, i, j, CV_RGB(255,255,255));
}
}
}

```

drawCharacter 在实现代码中还需要处理 IplImage 影像的 origin 标志和 ROI 结构, 这里为了简化省略了具体代码。另外, 当进行 cvSet2D 写字型的每个像素的时候, 还需要检查行列参数是否越界, 否则可能出现导致程序崩溃的错误。

3 具体实现

由于每个字体在装载的时候需要依赖外部的字库文件。作者将具体的操作作用 C++ 类进行封装, 用户采用更简单的接口进行访问。

3.1 CvTextOut 类的实现

为了便于使用, 作者将实现的 OpenCV 绘制功能封装到一个 CvTextOut 类中。CvTextOut 类的接口如下:

```

class CvTextOut{
// 获取/设置字体参数
void getFont(int *type, CvScalar *size, bool *underline,
float *diaphaneity);
void setFont(int *type, CvScalar *size, bool *underline,
float *diaphaneity);
// 输出函数
int putText(IplImage *img, const char *text, CvPoint pos,
CvScalar color);
// ...
};

```

函数 getFont 和 setFont 用于获取和设置当前的输出字体。函数 putText 根据当前的字体, 用 color 颜色在 img 影像的 pos 位置输出 text 文本, 支持汉字输出。

3.2 测试结果

下列代码在 img 影像的 100 行 150 列位置输出文本“在 OpenCV 中输出汉字!”:

```

CvTextOut textOut("wqy-zenhei.ttf");
char *msg = "在 OpenCV 中输出汉字!";
textOut.putText (img, msg, cvPoint (100, 150), CV_RGB
(255,0,0));

```

其中“wqy-zenhei.ttf”为“文泉驿黑正矢量”字库。显示效果如图 2 所示。



图 2 在 OpenCV 中输出汉字

4 结束语

本文对 OpenCV 中汉字的绘制技术进行了详细的介绍, 实现了汉字在 OpenCV 中的输出。由于采用 C 语言中标准的函数处理汉字编码, 采用开源的 FreeType 提取字库, 因此可以很好地推广到其它操作系统和其它不同国家区域的字符集系统。

参考文献:

- [1] Intel Corporation. Intel open source computer vision library reference manual[Z]. 2001.
- [2] 吕学刚, 于明, 刘翠响. IPL 和 OpenCV 在 VC++ 环境下的应用[J]. 微型电脑应用, 2003, 19(1): 33-34.
- [3] 陈磊. 计算机视觉类库 OpenCV 在 VC 中的应用[J]. 微计算机信息, 2007, 23(4-3): 209-210.
- [4] 黎松, 平西建, 丁益洪. 开放源代码的计算机视觉类库 OpenCv 的应用[J]. 计算机应用与软件, 2005, 22(8): 134-136.
- [5] 杨远宾, 刘炯. 基于 VxWorks 嵌入式系统的中文平台研究与实现[J]. 计算机工程与设计, 2006, 27(10): 3890-3893.
- [6] 孙枫, 陈业夫, 郭勇鹏. 在 VxWorks 系统中使用 TrueType 字库[J]. 应用科技, 2003, 30(11): 53-55.
- [7] 李杰, 陈锦昌. OpenGL 中实现三维轮廓汉字的研究[J]. 工程图学学报, 2001(3): 34-37.
- [8] 李雪, 翟正军, 姚和平, 等. Vega 环境下汉字显示技术的实现方法[J]. 测控技术, 2007, 26(4): 80-82.
- [9] 陶青林, 找英俊. VTK 中一种显示汉字的方法[J]. 机床与液压, 2005(2): 152-153.
- [10] 许丽卿, 周力臻. 基于 OpenGL 的三维立体汉字显示的研究[J]. 福建师范大学福清分校学报, 2005, 76(5): 9-12.
- [11] 刘瑞祯, 于仕琪. OpenCV 教程——基础篇[M]. 北京: 北京航空航天大学出版社, 2007: 202-203.
- [12] Samuel P Harbison, Guy L Steele. C 语言参考手册[M]. 5 版. 北京: 机械工业出版社, 2003: 299-301, 327-328.

(上接第 5092 页)

- | | |
|--|--|
| <ol style="list-style-type: none"> [4] 汪培庄, 李洪兴. 模糊系统理论与模糊计算机[M]. 北京: 科学出版社, 1996. [5] 吴贤国. 分形技术在气固流化床检测中的应用研究[D]. 杭州: 浙江大学, 2002: 20-22. [6] 刘文奇, 吴从. 相似关系粗糙集理论与相似关系信息系统[J]. 模 | <ol style="list-style-type: none"> 糊系统与数学, 2002, 16(3): 50-58. [7] 田宏, 王秀坤. 一种基于弱模糊相似关系的广义粗糙集[J]. 大连理工大学学报, 2007, 47(2): 288-230. [8] 方婷婷. TL-模糊粗糙集及群上的 TL-模糊粗糙集研究[D]. 成都: 西南交通大学, 2007: 31-36. |
|--|--|