

# 基于 OpenCV 的车牌图像定位

胡雪会, 王波涛

(北京工业大学 电子信息与控制工程学院, 北京 100124)

**摘要:** 现有的车牌定位方法几乎都需要先对白天夜晚的场景进行分类, 在图像中存在其他灰度剧烈变化区域时, 这种场景分类容易影响到车牌定位的准确率; 为了对获取到的车辆图像进行准确地定位, 提出了一种改进的灰度跳变车牌定位算法, 首先加入了光照补偿, 使得不需要对白天和晚上的场景进行分类; 然后进行了长竖线的噪声去除, 可以在背景比较复杂的情况下准确定位车牌; 并对精确定位出的车牌进行边框的去除, 有利于下一步的字符分割; 另外, 对 OpenCV 计算机视觉库的库函数的利用, 算法的复杂度得到了简化, 从而更好地满足了车牌识别系统的实时性; 通过对出入口处采集到的分辨率为  $704 \times 576$  的 300 张图片进行测试, 定位率高于 95%, 对白天和夜晚获取的图像有很好的适应性, 满足设计要求。

**关键词:** 车牌定位; 光照补偿; 中值滤波; 边缘检测; 灰度变换

## A Open CV—based Vehicle License Plate Location

Hu Xuehui, Wang Botao

(College of electronic and control engineering, Beijing University of Technology, Beijing 100124, China)

**Abstract:** Almost all existing methods need to classify the day and night scene, and the accuracy of the positioning of the plate by this classification is easily affected when there are other dramatic changes in the gray area in the image. An improved gray level jump License Plate Location Algorithm is promoted for the purpose of accurate positioning. First the illumination compensation is joined, makes no need for day and night scene classification. Then the long vertical lines of noise is removed, in the case of complicated background. And the frame is removed after the accurate location of license plate, which benefited to the next character segmentation. In addition, the OpenCV library is used in the algorithm, which is made the algorithm more simplify and real-time. The accuracy is more than 95% by testing 300 images in resolution ratio of  $704 \times 576$ . The algorithm is adapted to the picture both from the day and night. The design requirement is satisfied.

**Keywords:** license plate localization; illumination compensation; median filtering; edge detection; gray level transformation

## 0 引言

车牌识别是当前国内外研究的一个热点方向, 主要用于高速公路、停车场、公路自动收费站、居民小区等环境下, 所以国内外的学者将计算机视觉、数字图像处理、模式识别等技术运用于车牌识别系统, 用于对图像或视频中的车牌号码进行识别, 实现对车辆进行自动管理和控制。车牌定位的主要目的就是确定车辆牌照的具体位置并将包含车牌号码的子图像从整个图像中划分出来, 以便进行后续的字符分割与字符识别, 因此定位的好坏直接影响到整个系统识别率的高低。车牌定位是典型的在复杂背景中进行目标检测的问题, 车辆图像是在室外环境下拍摄的, 易出现污损变形失真, 且分辨率较低。一般情况下, 车牌区域只会占据整个车辆图像中的一小块区域, 位置不固定, 而且经常会受到污染的影响, 并且在处于不同的光照条件下时, 车牌区域的亮度、颜色、对比度也会不同。目前有以下几种常用的车牌定位方法:

- 1) 基于颜色特征的车牌定位方法<sup>[1]</sup>;
- 2) 基于灰度直方图的定位方法;
- 3) 基于形态学的车牌定位方法<sup>[2]</sup>;
- 4) 基于 BP 神经网络的车牌定位方法<sup>[3]</sup>。

方法 1) 定位准确, 精确度比较高, 但适应性差, 无法对褪色、偏色以及背景颜色干扰等情况进行有效处理; 方法 2)

速度比较快, 算法较直观, 但容易受到背景和噪声的影响; 方法 3) 算法比较简单, 计算速度较快, 但在大多数情况下不能将车牌区域边界精确定位出; 方法 4) 自适应能力强, 准确率较高, 但计算量大, 每当车牌尺寸发生的变化比较大时, 就需要重新训练神经网络。

本文提出一种针对出入口环境下的车牌定位算法, 算法比较简单, 准确率较高, 而且对光照不敏感。在预处理阶段, 首先对图像进行光照补偿, 用来提高图像的质量, 接着利用 OpenCV 计算机视觉库的库函数进行图像的灰度化, 然后利用两次二值化来消除路面的噪声。中值滤波后利用抑制噪声能力较好的 Sobel 算子得到完整的边缘图像, 接着对长横线和长竖线进行了去除。由于车框对后续的车牌字符分割影响较大, 粗定位和细定位后对车牌边框进行去除。本文详细介绍了从车辆图像预处理到车牌精确定位的过程, 并通过 300 张真实的车牌图像进行了测试, 定位率高于 95%。

## 1 本文车牌定位系统的基本思路

我国的车牌主要有以下特点<sup>[4]</sup>:

- 1) 车牌组成是两种或 3 种颜色;
- 2) 车牌的大小构成是  $440 \text{ mm} \times 140 \text{ mm}$ , 宽高比在 3 左右;
- 3) 车牌字符面积占整个车牌的 20% 左右;
- 4) 相邻字符的间隔一般是  $12 \text{ mm}$ , 第二个和第三个字符的距离是  $34 \text{ mm}$ , 车牌字符是  $45 \text{ mm} \times 90 \text{ mm}$ 。

本文提出的定位方法的限制条件如下<sup>[5]</sup>:

- 1) 获取图像的距离范围一般设置在  $2 \sim 4 \text{ m}$  之间, 而且需

收稿日期: 2016-03-24; 修回日期: 2016-04-25。

作者简介: 胡雪会(1990-), 女, 山东聊城人, 硕士研究生, 主要从事图像处理方向的研究。

要为汽车正面角度的图像。如果车牌图像获取的距离过远或过近, 则会给车牌定位增加难度, 还会降低运算速度;

2) 车牌位置没有较大的破坏, 最好是在肉眼能够分辨出车牌位置和号码情况下。

测试图像如图 1。



图 1 原始测试图

本文利用 VC6.0, 结合 OpenCV 计算机视觉库进行算法的实现。VC 提供了功能强大的 MFC (microsoft foundation class), 可方便友好地通过界面对设计结果进行展示。MFC 是一套面向对象的函数库, 以类的方式将函数提供给用户使用。利用这些类, 可以快速地完成 Windows 应用程序的开发。

本文对出入口处采集到的车辆图像进行分析和处理, 主要包括图 2 的几个步骤。

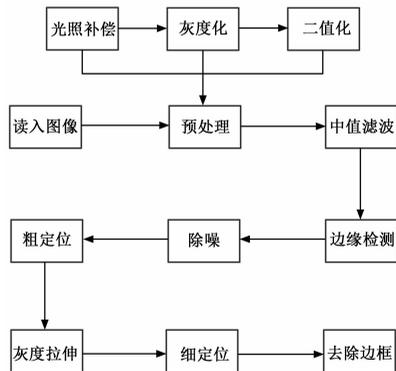


图 2 车牌定位流程图

## 2 车牌定位的设计

### 2.1 读入图像

由于本文使用的开发环境是 Windows 操作系统下的 VC6.0, BMP 文件正是 Microsoft Windows 所定义的图片文件格式, 最早应用在 Microsoft 公司的 Microsoft Windows 窗口系统, 所以本文进行处理的图片均为 BMP 格式。

本文算法首先需要把输入的图像转换为 DIB 位图的格式, 但 MFC 中没有封装实现这个功能的类, 为了在程序中对位图方便的处理, 所以需要建立一个 IplImage 的类, 这样就可以在对 DIB 位图进行使用直接调用函数, 而且在此基础上进行开发, 可以避免重复性的工作。设计的 IplImage 类的基本操作功能包括:

- 1) 对 BMP 图像进行读取和保存;
- 2) 将彩色图像转换成灰度图像;
- 3) 获取图像的宽度和高度。

在实现的过程中, 可以定义多个 DIB 处理变量, 并且各

个步骤的处理结果可以进行保存, 直观地显示出来, 其中任何一个中间过程的位图都可以进行处理和输出, 可以通过中间得到的图像进行有针对性的处理和分析。

利用 OpenCV 库函数 cvLoadImage 对图像进行读取。函数原型: IplImage \* cvLoadImage (const char \* filename, int flags=CV\_LOAD\_IMAGE\_COLOR);

利用 OpenCV 库函数 cvSaveImage 对图像进行保存。函数原型: IplImage \* cvSaveImage (const char \* filename, const CvArr \* image);

### 2.2 图像预处理

#### 2.2.1 光照补偿

由于采集图像的手段和方法日新月异地发展, 所以采集到图像的种类很多, 获得的视觉质量也各不相同。这样多年来就形成了许多对图像进行增强处理的不同技术。根据图像增强技术处理所进行的空间不同, 目前常用的可分为空间域法和频率域法两大类。

空间域法是在原图像上直接对像素的灰度值进行处理, 进行数据运算。常见的处理方法有对数变换、指数变换、直方图均衡化、分段线性化。

频率域法是在图像的变换域上进行处理, 增强感兴趣的频率分量, 然后再进行反变换, 最后得到增强的图像, 是一种间接增强的算法。常见的频率增强方法有: 同态滤波、低通滤波、高通滤波等等。

除了上面这些传统的图像增强方法外, 近几年又出现了一些新兴的图像增强方法, 如基于小波变换的增强方法、模糊增强方法、基于神经网络的增强方法以及基于 Retinex 理论的光照补偿增强方法等<sup>[6]</sup>。

基于 Retinex 理论的图像增强方法很好地解决传统图像增强方法的缺陷, 可以保持原图像色彩恒常性, 提高图像对比度, 使图像增强思想有了全新的扩展。由于该理论结合人眼视觉特性, 使得增强后的图像在色彩保真性、对比度上均优于传统的图像增强算法。

Retinex 理论的基础是色彩恒常性, 色彩恒常性即在光线发生变化的时候, 人眼能对物体保持恒定不变的观察效果。基于 Retinex 理论的 MSRRCR (Multiscale Retinex) 增强算法具有对比度高、动态范围大、色彩失真小等特点, 它能在灰度范围内动态压缩, 边缘增强和颜色恒常性 3 个方面达到平衡, 所以可对各种不同种类的图像进行自适应增强, 从根本上说是一种基于光照补偿的图像增强算法。

本文所采用的带色彩恢复的 MSRRCR 算法增强, 不但可以突出原图像中所包含的细节信息, 使图像的整体效果得到提升, 而且处理后的图像色彩保真性更好, 使图像光照均匀, 更方便的进行图像处理。

测试表明, 加光照补偿后的车牌定位算法不需要对白天夜间的场景分类, 而且可以避免图像中其他灰度变化剧烈区域对车牌定位的影响, 比如车辆标志区域、车灯区域, 可准确定位出车牌。

#### 2.2.2 灰度化和二值化

图像灰度化的方法主要有一下 4 种:

- 1) 最大值法;
- 2) 平均值法;
- 3) 加权平均法; 该方法根据人眼对绿色最为敏感, 对红

色次之,对蓝色的敏感性最低。

4) 分量法;将  $R$ ,  $G$ ,  $B$  三个分量中的一个分量作为彩色图像转换成灰度图像的灰度值。

其中:  $Y$  是转换后灰度图像的值,  $R$ ,  $G$ ,  $B$  则代表原彩色图像中红、绿、蓝 3 个分量。

考虑到人眼对不同颜色敏感度不一样,本文根据方法(3)对彩色图像到灰度图像转换,可利用 OpenCV 库函数 `cvCvtColor` 进行灰度化。函数原型: `void cvCvtColor (const CvArr * src, const CvArr * dst, int code)`。其中的参数: `src` 表示要处理的图像, `dst` 将处理后的图像进行保存, `code` 则表示色彩空间转换的模型,这里 `code` 的值是 `CV_BGR2GRAY`,即加权平均法<sup>[7]</sup>。

本文采用两次二值化的方法,该方法利用了图像中车牌字符区域对比度比较强的特点,经过一次二值化后只产生黑度带和白度带;而路面的灰度值分布均匀,经过两次二值化后生成黑度带、灰度带和白度带。在进行后续的边缘检测时,只需记录黑度带和白度带之间产生的边缘,这样可以消除路面区域边缘的干扰。

### 2.2.3 滤波和边缘检测

经常使用的图像去噪方法有均值滤波和中值滤波<sup>[8]</sup>。

均值滤波是一种典型的线性滤波算法。均值滤波不能很好的保护图像的细节,在去除图像噪声的同时也破坏了图像的细节部分。中值滤波是一种避免边缘模糊的非线性平滑方法,中值滤波可以很好的消除图像的噪声,并且经过滤波后,图像边缘的不存在明显的模糊情况,在图像处理中经常被用来保护图像边缘信息。

合理设置运算模块的大小是中值滤波的关键,本文设置一个  $5 \times 5$  像素大小子模块,这样可以对图像平滑滤波的过程中,既能抑制噪声的又能很好地保持边缘信息,还兼顾了运行的速度。

图像的边缘是指一些像素的集合,这些像素周围像素灰度变化是不连续的,也就是指图像局部亮度变化最显著的部分。传统边缘检测的算子有许多,作为代表性的有 Roberts、Sobel、Prewitt、Laplacian、Kirsch 等。随着图像理论的快速发展,出现了一些有代表性的新的算子,比如基于小波变换、基于形态学以及神经网络等的检测算子。

Roberts 算子对于边缘定位有较好的效果,但不能完整的对边缘信息进行保留,而且受到噪声的影响比较大。Sobel 算子和 Prewitt 算子非常相似,只是在权值选取上有一些不同,这两者在一定程度上都可以对噪声进行抑制。Laplacian 算子保留边缘不完整,并且抑制噪声的能力比较差。Kirsch 算子抑制噪声比较有效,但是计算量太大<sup>[9]</sup>。新出现的代表性的边缘检测算子虽定位出的边缘信息比较准确,但计算量太大也是一大劣势。

与 Prewitt 算子比较起来, Sobel 算子对于像素位置的影响做了一定的加权,可以对边缘模糊程度进行降低,所以效果比较好,因此本文采用 Sobel 算子对车辆灰度图像进行边缘检测。利用像素点上下、左右邻点的灰度进行加权的算法,并且根据在边缘点处达到极值这一现象进行边缘的检测。Sobel 算子受噪声的影响较小,而且对噪声具有平滑作用,可提供较为精确的边缘方向信息,是一种比较好用的边缘检测方法。

Sobel 算子是一种传统滤波算子的形式,运算时可以利用

快速卷积函数,用来提高算法的速度,依据式(2)进行运算:

$$S(x) = \max\{|(N_0 + N_1 \times 2 + N_2) - (N_4 + N_5 \times 2 + N_6)|, \\ |(N_0 + N_7 \times 2 + N_6) - (N_2 + N_3 \times 2 + N_4)|\} \quad (2)$$

### 2.2.3 除噪

现有算法中除噪都是针对于长横线,但长竖线对车牌的定位也会造成影响。本文中的除噪是为了去除边缘图像中的长横线和长竖线。当图像中的连续白色线超过一定长度或高度时,就可以认为是噪声,进行滤除。除噪效果如图 4 所示,除噪后的车牌字符边缘点几乎没有减少,汽车标志和护栏周围的边缘点则减少了很多。



图 3 除噪前图



图 4 除噪后图

### 2.2.4 粗定位

由于拍摄角度的问题,车辆图像中的车牌部分可能会存在一定的倾斜,为了保证定位出来车牌图像的完整性,在粗定位时可稍微放大车牌的区域。

本文通过利用矩形区域对图像进行扫描的方法来进行车牌的粗定位,矩形区域为  $160 \times 40$ 。通过对矩形区域内白色的像素数进行统计,可以很快的找到车牌的大致区域,在粗略定位区域的位置进行确定后,便对该区域进行截取,将上下左右 4 个边界所围区域以外的所有位置的像素点赋予 255 的值,从而通过“变白”的方式去除定位区域以外的无关图像部分,以达到车牌图像截取的目的。

### 2.2.5 灰度变换

对比度增强是指利用各种数学方法和频域变换手段提高图像中的目标跟非目标的对比度与图像清晰度。目标是指所需要研究的对象,非目标是指目标以外的背景部分。在空间域中,常用的对比度增强方法如灰度变换、直方图均衡法。本文采用灰度变换的方法进行图像的对比度增强。

车牌图像中对比度不足主要是因为被拍摄物体的远近不同,造成中间部分和边缘部分的灰度不平衡,或者是由于曝光不足或过度使得图像的灰度值范围比较小。这样得到的图像由于没有灰度层次而模糊不清。灰度变换的原理如下:在原图像  $f(x, y)$  中,大部分像素的灰度值范围较小,设为  $[a, b]$ 。经过灰度变换后,在图像  $g(x, y)$  中,灰度扩展到中一个比

较大的范围  $[c, d]$ , 其中  $g(x, y)$  即为灰度变换后的图像<sup>[10]</sup>。 $f(x, y)$  和  $g(x, y)$  之间的灰度变换关系如下式(3)所示:

$$g(x, y) = \begin{cases} c, & f(x, y) < a \\ \frac{d-c}{b-a}[f(x, y) - a] + c; & a \leq f(x, y) < b \\ d, & f(x, y) \geq b \end{cases} \quad (3)$$

车牌图像灰度变换的效果如图 5~6 所示。



图 5 灰度变换前图



图 6 灰度变换后图

### 2.2.6 细定位

经过粗定位后, 图像中的大部分背景区域被移除。但是粗定位后的结果并非只有车牌区域, 还包含较多非车牌区域, 所以需要再进行第二次精确定位。

本文中的细定位过程首先经过膨胀与腐蚀, 去除孤立噪声点的同时形成车牌区域的连通, 将无关区域去掉。

膨胀是将与目标对象接触的所有背景点合并到对象中的过程, 结果是使目标增大, 孔洞缩小, 可用来填补目标物体中的孔洞, 形成连通域。

腐蚀是一种消除边界点的过程, 可使目标对象缩小, 孔洞增大, 因而可有效的消除孤立的噪声点。

车牌区域都是一些孤立的点, 为了进一步判断车牌区域的位置, 可利用数学形态学中的膨胀, 使目标对象增大, 孔洞缩小, 以填补目标对象中的孔洞, 形成车牌区域的连通。再经过腐蚀处理, 就可以去除孤立的噪声点。

经过膨胀与腐蚀的操作后, 车牌的位置就相对比较精确了, 但大部分情况下车牌的边框还是存在, 所以需要继续对边框和铆钉进行去除。

### 2.2.7 去除车框

经过细定位后的车牌区域相对比较精确, 但如果在车牌区域含有边框和铆钉, 会影响车牌字符分割的效果。因而, 在车牌字符分割前需要对车牌进行去除边框和铆钉的处理, 突出字符, 最后定位后的车牌无边框, 便于进行后续的字符分割。

本文根据车牌图像在水平方向上的灰度跳变去除上下边框, 根据车牌图像的垂直投影去除左右边框。

理想情况下, 二值车牌图像在水平方向上的像素黑白跳变有以下规律:

- 1) 车牌水平上下边框黑白跳变次数为 0;
- 2) 车牌第二个至第六个字符上下铆钉处所在行的黑白跳变各自 2 次, 所以至少为 4;
- 3) 由于车牌字符有 7 个, 因而在计算水平投影跳变时至少有  $7 \times 2 = 14$  次。

本文选择的阈值为 12, 大于 12 的认为是字符区域, 否则为背景, 即非字符区域。从中间处向上扫描, 第一个小于 12 以上的区域即可认为是背景; 从中间处向下扫描, 第一个小于 12 以下的区域即可认为是背景。

理想情况下, 二值车牌图像在垂直方向上的投影有以下规律:

车牌边框处的投影值局部最大。

本文对图像进行从中间向左的扫描, 得到第一个极大值,

即可认为是车牌的左边框; 对图像进行从中间向右的扫描, 得到第一个极大值, 即可认为是车牌的右边框。

最终定位出的车牌图像如图 7~8 所示。



图 7 细定位图



图 8 去边框后图

## 3 车牌定位的测试

在车牌定位算法完成后, 需要将其定位的效果进行测试, 测试算法是否满足使用需求, 是否正常运行。为了对本算法进行有效的测试, 将出入口处采集到的分辨率为  $704 \times 576$  的 300 张图片进行测试。测试结果表明, 本文的算法对光照不敏感, 而且定位出的车牌无边框。

表 1 算法测试结果

光照条件	车辆图像	定位出的车牌
正常光照		京N·F2059
		京P·5G023
车牌区域弯折 (光照不均匀)		京Q·J9Y65
光照较暗		京K·04945
		京J·L7689

## 4 结论

本文设计了一种改进的车牌定位方法, 结合 OpenCV 计算机视觉库, 有效提高了算法的准确性和效率。通过光照补偿改善了图像的质量, 然后利用具有平滑作用的 Sobel 边缘检测算子检测出了完整的图像边缘, 利用图像的水平跳变和垂直投影进行了边框的去除, 最后得到了无边框的车牌定位图像, 有利于进一步的字符分割和字符识别的处理。在本文所阐述的车牌定位的编程环境下, 将出入口处采集到的分辨率为  $704 \times 576$  的 300 张图片进行测试, 定位率高于 95%, 满足设计要求。

参考文献:

[1] 潘寒飞. 基于颜色特征的车牌定位与分割技术研究 [D]. 重庆: 重庆大学, 2012.

[2] 程磊, 程伟. 基于数学形态学的车牌定位与倾斜校正的方法 [J]. 科技视界, 2015, 23: 10-11.

[3] 赵涛, 杨晓莉, 王绪本, 等. 一种用于车牌定位的改进 BP 神经网络方法 [J]. 计算机仿真, 2007, 24 (2): 240-243.

[4] 朱梦哲, 陈志华, 赵钟, 等. 基于 OpenCV 的车牌定位和校正方法 [J]. 计算机应用, 2013, 33 (S1): 223-226.

[5] 胡将胜, 侯建华. 基于区域标注的车牌提取算法研究 [J]. 科学技术

术与工程, 2011, 11 (2): 378-382.

[6] 刘钦堂. 基于 Retinex 算法的彩色图像增强研究 [D]. 哈尔滨: 哈尔滨工程大学, 2012.

[7] 谭同德, 王三刚. 基于 OpenCV 的车牌定位方法 [J]. 计算机工程与设计, 2013, 34 (8): 2816-2820.

[8] 鲁恋情. 复杂环境下的车牌定位算法 [D]. 镇江: 江苏大学, 2013.

[9] 籍美苹. 车牌定位算法的研究 [D]. 重庆: 重庆大学, 2014.

[10] 韩立明, 王波涛. 车牌识别中关键技术的研究与实现 [J]. 计算机工程与设计, 2010, 31 (17): 3919-3923.

(上接第 205 页)

先分别记录 Android 移动终端在充满电的状态下分别在待机状态下分别运行 5 周的电量使用情况, 之后在上面启动我们的情景感知策略再在分别充满电的状态下分别运行 5 周, 同样记录这 5 周内待机状态下的电量使用情况。

图 5 为系统在待机状态下分别在 5 周的电量消耗, 在没有运用情景感知策略的情况下, 分别用掉系统电量的 58%、52%、61%、59%、56%, 虽然有所波动, 但是也在误差允许范围内。在运用情景感知策略的情况下分别用掉系统电量的 49%、52%、51%、48%、43%。计算可以得出运用情景感知策略后可以比未运用情景感知策略的情况下节省 15.1% 电量。

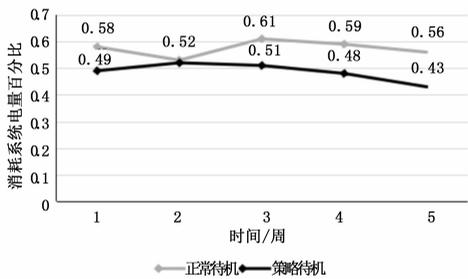


图 5 系统消耗电量图

4 结语

Android 移动终端的高性能与电池容量受限的矛盾使得减少功耗已成为其发展阻碍, 为了有效地减少待机状态下的功耗, 提出了一种基于情景感知的电源管理策略及其框架, 利用移动终端的情景感知信息进行决策来预测当前是否需要网络接口, 并智能地对其进行动态管理, 减少待机状态下的系统能耗。我们对复杂的情景信息进行了预处理, 并优化了获取定位数据的方法。决策算法是决策树算法, 根据信息增益生成决策树。根据生成的决策树来决定待机状态下的网络连接的状态, 并分别进行相关实验和对比。结果表明我们提出的情景感知策略更加节省能耗。情景感知策略的实验结果表明其在智能管理网络接口的同时, 能够在待机状态下节省 15% 左右的系统功耗。以上实验结果充分验证了我们提出的策略的可行性和有效性。

参考文献:

[1] IBM and MontaVista Software. Dynamic Power Management for Embedded Systems V1.1 [Z]. 2002.

[2] Zhao Xia, Chen Xiangqun, Guo Yao, et al. A Survey on Operating System Power Management [J]. Journal of Computer Research and Development, 2008, pp817-824.

[3] Wang C H, Wu A. A Predictive System Shutdown Method For Energy Saving Of Event Driven Computation [A]. In: Proc of Int. Conf on Computer Aided Design [C]. Los Alamitos: IEEE Computer Society Press, 1997. 28-32.

[4] Soumya Kanti Datta, Christian Bonnet, Navid Nikaein, Sophia Antipolis. 2013. Minimizing Energy Expenditure in Smart Devices [A]. In Proceedings of 2013 IEEE Conference on Information and Communication Technologies (ICT 2013) [C]. IEEE, 2012: 578-585.

[5] Pering T, Burd T, Brodersen R. The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms [A]. Proceedings of the 1998 International Symposium on Low Power Electronics and Design [C]. Monterey, CA, USA: ACM, 1998.

[6] Grunwald D, Morrey III C B, Levis P, et al. Policies for Dynamic Clock Scheduling [A]. Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation [C]. CA, USA: USENIX Association, 2000: 6.

[7] Indumathi G, Ramakrishnan K V. Study and Analysis of Power Optimization Techniques for Embedded Systems [A]. 2011 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) [C]. Chennai: IEEE, 2011: 365-366.

[8] Seo Y, Kim J, Seo E. Effectiveness Analysis of DVFS and DPM in Mobile Devices [J]. Journal of Computer Science and Technology, 2012, 27 (4): 781-790.

[9] Kim J, Park J. Reducing Power Consumption Using Improved Wakelock on Android Platform [A]. The Eighth International Multi-Conference on Computing in the Global Information Technology [C]. Nice, France: IARIA, 2013: 171-174.

[10] Carroll A, Heiser G. An Analysis of Power Consumption in a Smartphone [A]. Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference [C]. CA, USA: USENIX, 2010: 21.

[11] Benini L, Bogliolo A, De Micheli G. A Survey of Design Techniques for System-level Dynamic Power Management [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2000, 8 (3): 299-316.

[12] 陈璟, 陈平华, 李文亮. Android 内核分析 [J]. 现代计算机, 2009 (11): 112-115.

[13] 方胜. 基于 Android 平台的电源管理机制的研究 [J]. 现代计算机, 2012 (2): 41-43.

[14] Android wakelocks and the Linux power management controversy [EB/OL]. <http://www.perforce.com/blog/110713/android-wakelocks-linux-power-management-controversy>.

[15] Learn and talk about Sysfs, Free special purpose file systems [EB/OL]. <http://www.digplanet.com/wiki/Sysfs>.