American Journal of Applied Sciences, 2012, 9 (12), 2052-2057
ISSN: 1546-9239

doi:10.3844/ajassp.2012.2052.2057 Published Online 9 (12) 2012 (http://www.thescipub.com/ajas.toc)

Measuring Power Consumption for Image Processing on Android Smartphone

Khairul Muzzammil Saipullah,
Ammar Anuar, Nurul Atiqah Ismail and Yewguan Soo

Department of Computer Engineering, Faculty of Electronic and Computer Engineering,
University Teknikal Malaysia Melaka, Melaka, Malaysia

Received 2012-07-05, Revised 2013-01-15; Accepted 2013-01-17

ABSTRACT

The energy consumption of smartphones can be undertaken in multiple levels of hardware and software. Generally, there are two approaches in measuring power consumption of a smartphone application which are the measurement-based and estimation-based methods. The goal of this study is to compare the two power consumption measuring approaches in quantifying the power consumed by image processing applications in Android smartphone. For measurement-based approach, a simple wattmeter is designed whereas for the estimation-based approach, an Android application called the PowerTutor will be utilized. The wattmeter and PowerTutor will measure the power consumption of eight image processing methods running on modified Android library with self-implemented algorithm called the CamTest. According to t-test analysis that has been conducted, the p values of all of the image processing methods show that there are no significant differences between the wattmeter and the PowerTutor application ($p>0.01$). Even though measurement-based method is more accurate than estimation-based method in term of measuring power consumption, PowerTutor application proved it provides accurate, real-time power consumption estimation for Android platform smartphones. Application developers still can use PowerTutor as an option to determine the impact of software design on power consumption.

Keywords: Power Consumption, Wattmeter Android, Image Processing, Consumption Estimation, Differences Between, Estimation-Based, Measurement-Based

1. INTRODUCTION

The mobile phones have become an indispensable and inseparable object in human life. There are 3.3 billion active mobile phones in the world (Lefebvre, 2009). It is common for individuals to own more than one. The increasing dependency upon mobile phone to perform a lot of functions more capability than conventional handsets make mobile phone itself has evolved into an entirely new device that we know as smartphone.

The growth of the smartphone industry has led these devices to keep improving their function such as high resolution camera with flash, bigger screens and support the thousands of application. Since the Open Handset Alliance released the Google Android SDK on November 12, 2007 (OHA), the conception of the Android platform is attracting more programmers to develop Android application. Android application developers also prefer to interface built-in device such as sensors, Wi-Fi, Bluetooth, camera and others into their application.

Application that uses camera usually related to image processing filters such as Gaussian, median, mean, Laplacian, Sobel and others (Genser *et al*., 2009). Image processing applications usually drain more power than other application because it process pixel by pixel of the image with heavy mathematical computation. Besides, image processing also use more memory and time consuming.

Power consumption of the smartphone can be measured using two approaches which are the measurement-base and estimation based approaches. To

Corresponding Author: Khairul Muzzammil Saipullah, Department of Computer Engineering, Faculty of Electronic and Computer Engineering, University Teknikal Malaysia Melaka, Melaka, Malaysia

compare those two power consumption measuring approaches, two tools have been chosen to represent each of the approaches. The tools are the custom made wattmeter and the power meter application called PowerTutor (Zhang *et al*., 2010). In order to measure the power accurately, the wattmeter is designed to measure even very minimal power consumption of eight image processing methods running on modified Android smartphone platform (Anuar *et al*., 2011). The measurement from the wattmeter will be compared with the measurement from the PowerTutor.

According to (Genser *et al*., 2009), software application power profiling can be categorized into measurement-based and estimation-based methods. Measurement-based methods are performed by measuring actual physical measurements. This method give high accuracy compared to other method but requires external measurement tool. On the contrary, power profiling by estimation methods is usually based on power modeling. Compared to the measurement-based method, these techniques are usually less accurate but more portable and provide better flexibility.

1.1. Measurement-Based Methods

Rice and Hay (2010) and Joshua and Lin (2010), the power profiling tool for smartphone applications is introduced. These methods obtain the value of power using Joule's law (Equation 1):

$$P = VI \quad (1)$$

The system applied series circuit combination where the current is the same at any point in the circuit. Measurement resistor is placed between power supply and the load. This resistor value should be as small as possible to reduce voltage drop crossing the measurement resistor. Since the resistance value is fixed and the voltage drop across the resistor is obtained using differential amplifier connected to microcontroller for later analysis. After the value of voltage and resistance are known, by using Ohm's law (Equation 2):

$$I = \frac{V}{R} \quad (2)$$

The current value will be easily calculated. Multiplying the current with the voltage drop across the resistor gives the power dissipated.

An oscilloscope measurement-based profiling technique is proposed by Texas Instrument (TI) in (TI, 2002). The current drawn by a DSP system is profiled and results are viewed on a host computer in TI's software development environment.

1.2. Estimation-Based Methods

Energy profiling by means of estimation techniques can be divided into (i) simulation-based and (ii) hardware-accelerated approaches (Genser *et al*., 2009). Simulation-based power estimation executes programs on simulators to acquire circuit activity information. Power values are obtained using this information. In hardware-accelerated power estimation approaches, power information is derived from power models, which are implemented in hardware.

Early estimation-based method models focused on estimating the power consumption of individual components, using the corresponding performance counters (Pathak *et al*., 2011) (CPU, memory and of the entire system), e.g., (Tiwari *et al*., 1996; Bellosa, 2000; Rawson, 2004). These models do not relate the power consumption of the system with the applications. Recent estimation-based power models tried to estimate the power consumption of application running on desktops (Zeng *et al*., 2002), sensor nodes (Shnayder *et al*., 2004), virtual machines (Kansal *et al*., 2010) and more recently, mobile phones (Zhang *et al*., 2010).

2. MATERIALS AND METHODS

2.1. The Custom Wattmeter

In order to measure power consumption of the smartphone, a device was built to analyze current flowing into the smartphone. This device uses external power supply only for the voltage supplier for the entire device but for the phone it will use its own battery as power supply to maintain the actual current and voltage. Four 0.1Ω measurement resistor in parallel (to get 0.025Ω) are inserted between a battery terminal and its connector on the phone. This is done by replacing the battery of the smartphone with copper tape.

The measurement resistor surely will increase the circuit resistance and its power consumption but the increases are typically less than 1% of the total power. The circuit schematic is shown in **Fig. 1**. The wattmeter hardware to measure power consumption of the smartphone is shown in **Fig. 2**. The hardware consists of a LCD display to show the power consumption value, a microcontroller to calculate of the measured power consumption and the power measuring circuits. The Analog to Digital Converter (ADC) of the microcontroller will read the voltage and the current from the power circuits and convert it into power in Watts unit.

Android platform should always positively recognize a battery's presence through the buffer surges. On most smartphone, the battery contains a tiny, undocumented, microcontroller attached between two pins between V+ and V- on the battery (Joshua and Lin, 2010).

Table 1. Frame processing methods and its description

Frame Processing	Description
RGB	Convert The Original YUV Color Space To RGB Color Space
Grayscale	Convert the Y color space to 0~255 grayscale
Threshold	Threshold the grayscale pixel with 70
Median	Filtering the grayscale frame with average of all the pixel values in a 3×3 window
Gaussian	2D convolution with Gaussian 3×3kernel
Mean	Filtering the grayscale frame with median of all the pixel values in a 3×3 window
Laplacian	2D convolution with Laplacian 3×3 kernel
Sobel	Filtering of the grayscale frame in horizontal and vertical direction using 3×3 Sobel operator

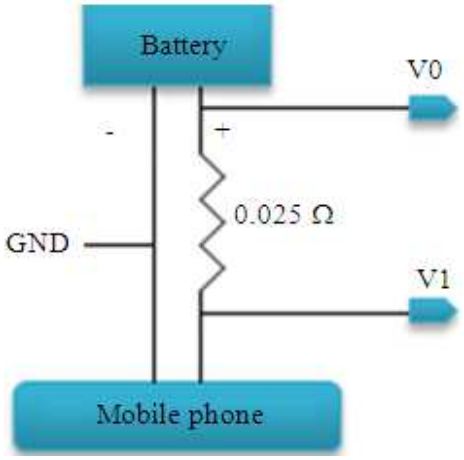

Fig. 1. Block diagram for the hardware component

Fig. 2. The system to measure power consumption of the smartphone

This microcontroller stores some state-of-charge information and can be functioned as thermometer to read temperature from the battery. Without this microcontroller's presence, the smartphone system will unable to boot and if the microcontroller is removed from an actively running system, the smartphone will shut down immediately. The battery should be handled carefully so that it will not disturb the entire power consumption measuring system and shut the smartphone system down.

2.2. The Image Processing Methods

There are eight image processing methods that are tested in the experiment. Each of the methods will be executed on each frame of displayed by the smartphone's camera. **Table 1** shows the brief description the eight image processing methods. The input format from the Android smartphone camera is in usually in the YUV color space. So it needs to be converted to RGB color space for video processing in standard color space. For video processing in grayscale, the luma (Y) is mapped to 0~255 scale. For RGB processing, all the channels in YUV color space are used to convert from the YUV space into the RGB space. And lastly, for the threshold, median, Gaussian, mean, Laplacian and Sobel image processing, the resulting grayscale fram from the grayscale processing method is utilized.

The YUV to RGB conversion formula is calculated using (Equation 3):

$$\begin{aligned} R &= 1.164(Y - 16) + 1.596(V - 128) \\ G &= 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128) \\ B &= 1.164(Y - 16) + 2.018(U - 128) \end{aligned} \tag{3}$$

For image thresholding each pixel is thresholded against a constant number T. If the pixel value larger than T, the pixel value will set to 1, otherwise the pixel value will be set to 0. The image thresholding can be calculated using the following formula (Equation 4):

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) > T \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where, f(x, y) is original frame and g(x, y) is thresholded frame.

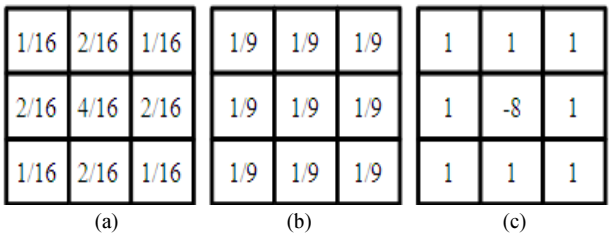

Fig 3. (a) Gaussian mask, (b) Mean filter mask, (c) Laplacian mask

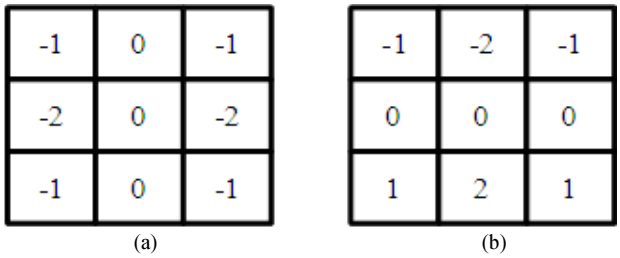

Fig. 4. (a) The vertical direction Sobel 3×3 mask. (b) The horizontal direction Sobel 3×3 mask

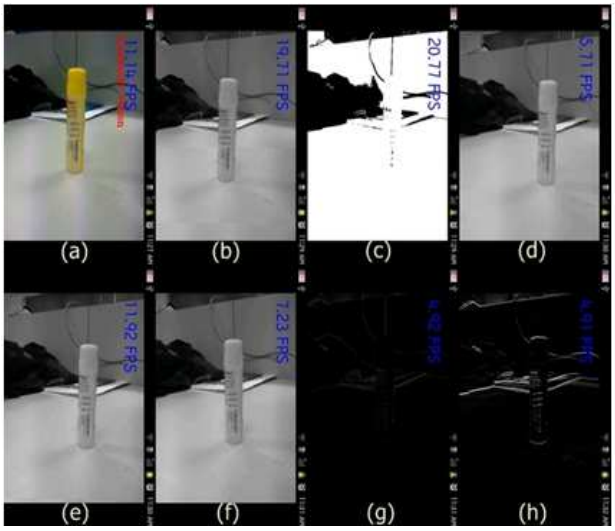

Fig. 5. OpenCV library implementation. Those images were captured when it processes images in real-time video processing. (a) is RGB image, (b) is Greyscale image, (c) is Threshold image, (d) is Mean filter image, (e) is Gaussian image, (f) is Median filter image, (g) is Laplacian filter image, (h) is Sobel filter image

In order to remove the noise from the frame using median filter, each 3×3 window of the original frame is processed by calculating the median value of the whole pixels in 3×3 window. This median value is then will be the new pixel value on the median filtered frame.

For video blurring image processing, each frame is convolved using a 3×3 mask. For Gaussian blurring, the frame will be convolved with the 3×3 mask as shown in **Fig. 3a**. For mean filter, the frame will be convolved with 3×3 mask as shown in **Fig. 3b**. For edge detection, each frame is convolved using a 3×3 mask. For Laplacian, the frame will be convolved with the 3×3 mask as shown in **Fig. 3c**. The sobel edge detection uses two 3×3 masks which are convolved in the vertical and horizontal directions of the frame. The two 3×3 masks are as shown in **Fig. 4**.

Figure 5 shows the image processing methods using CamTest application. Those images were captured in real-time using Xperia Ray Android smartphone camera at the same position for the eight different methods as explained previously.

3. RESULTS

The experiment is done on Xperia Ray smartphone that is powered by 1 GHz Scorpion processor running with Android 4.0 Ice Cream Sandwich Operating System (OS). An Android application is developed using the CamTest library to run the eight image processing methods. The application will run any of the eight image processing methods to each 3264×2448 pixels frame detected by the 8 mega pixels camera.

While the smartphone is running the image processing task, simultaneously, the wattmeter and the PowerTutor application will measure the power consumption of the smartphone. Before that, we need to measure the power consumption of the smartphone before running the image processing application and labeled as the system power. Each of the power measured after running the image processing application will be subtracted with the system power in order to obtain the image processing power consumption.

For each image processing method, the power consumed by the image processing application is recorded for 60 sec. Then the average power in 1 sec value is recorded. The process is iterated 30 times and the mean of the 30 average power consumption per second is recorded for graphical display. The same procedures are conducted with the wattmeter and the PowerTutor.

To measure the differences between the wattmeter and the PowerTutor, the t-test is executed. As explained before, each power measurement for each image processing method using different power consumption measuring method is iterated 30 times. Those 30 readings will be treated as samples for each image processing method class using different power consumption measuring approaches. These samples are then use in the t-test experiment.

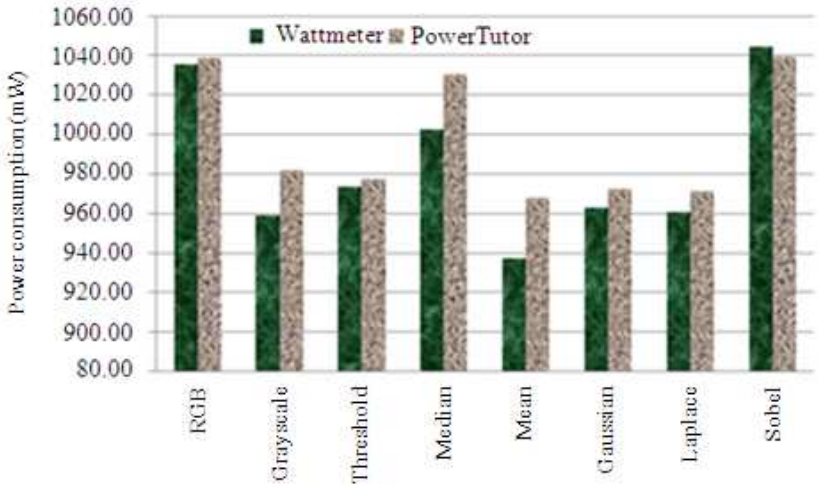

Fig. 6. The mean of 30 average power consumption in one second obtained from wattmeter and PowerTutor application for the eight image processing methods

Table 2. The t-test results of the eight image processing methods measured by wattmeter and PowerTutor application

Frame processing	T-test Result (p value)
RGB	0.96
Grayscale	0.67
Threshold	0.96
Mean	0.23
Gaussian	0.36
Median	0.45
Laplacian	0.80
Sobel	0.92

Figure 6 shows the mean of the 30 average power consumption in 1 sec. of the eight image processing method measured by from the wattmeter and from the PowerTutor application.

As it can be seen, the PowerTutor is slightly higher than the wattmeter for moss of the cases but there are also case where wattmeter measured higher power consumption than the PoweTutor.

To determine whether two samples are likely to have come from the similar source that have same mean, the t-test have been performed. **Table 2** shows the p values for those eight image processing methods. Since all p values are greater than 0.01, measurement value from wattmeter is not significantly different compared to the measurement value from PowerTutor.

4. DISCUSSION

From the average power consumption per second measurement, it can be seen that we are unable to make any decision which power consumption measuring approaches is better since the reading of the power consumption is quite similar for both of the approaches. As explained earlier in the introduction section, the measurement-based approach is the most accurate since it measures the power directly from the source.

However, the t-test confirms that there are no significant differences between the power consumption measured by the wattmeter or the PowerTutor in measuring the power consumption of image processing tasks.

This shows that the PowerTutor application can be an alternative method in measuring the power of the applications that are running in the Android smartphone especially image or video related application that consumes much power of displaying or processing the large data.

5. CONCLUSION

This experiment provides typical values and is useful information for smartphone users and developers. The application developers can benefit from knowing the detail behind these measurements. This study also described the measurement framework and how it can be used to create fine-grained understanding of energy consumption. This work has an easily replicable design by which application developers can measure the impact that their application has on a system battery life on their own without have to buy expensive machine to measure their application power consumption. The system has been designed can effectively help future Android developers especially to optimize power consumption and enhance battery performance.

Even though measurement-based method is more accurate than estimation-based method in term of measuring power consumption, PowerTutor application has proved it can provide accurate, real-time power consumption estimation for Android platform smart phones. Application developers still can use PowerTutor as an option to determine the impact of software design on power consumption.

6. REFERENCES

Anuar, A., K.M. Saipullah, N.A. Ismail and Y. Soo, 2011. OpenCV based real-time video processing using android smartphone. Int. J. Comput. Technol. Elect. Eng., 1: 58-63.

Bellosa, F., 2000. The benefits of event: Driven energy accounting in power-sensitive systems. Proceedings of the 9th Workshop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System, Sep. 17-20, ACM Press, New York, USA. DOI: 10.1145/566726.566736

Genser, A., C. Bachmann, J. Haid, C. Steger and R. Weiss, 2009. An emulation-based real-time power profiling unit for embedded software. Proceeding of the International Symposium on Systems, Architectures, Modeling and Simulation, SAMOS, Jul. 20-23, IEEE Xplore Press, Samos, pp: 67-73. DOI: 10.1109/ICSAMOS.2009.5289243

Joshua, A. and W.W. Lin, 2010. Precise power characterization of modern android devices. Carnegie Mellon University.

Kansal, A., F. Zhao, J. Liu, N. Kothari and A.A. Bhattacharya, 2010. Virtual machine power metering and provisioning. Proceedings of the 1st ACM Symposium on Cloud Computing, Jun. 10-11, ACM Press, New York, USA., pp: 39-50. DOI: 10.1145/1807128.1807136

Lefebvre, C., 2009. Integrating cell phones and mobile technologies into public health practice: A social marketing perspective. Health Promotion Practice, 10: 490-494. DOI: 10.1177/1524839909342849

Pathak, A., Y.C. Hu, M. Zhang, P. Bahl and Y.M. Wang, 2011. Fine-grained power modeling for smartphones using system call tracing. Proceeding of the 6th Conference on Computer Systems, Apr. 10-13, IEEE Xplore Press, New York. DOI: 10.1145/1966445.1966460

Rawson, F., 2004. Mempower: A simple memory power analysis tool set.

Rice, A. and S. Hay, 2010. Decomposing power measurements for mobile devices. Proceedings of the International Conference on 8th IEEE Conference on Pervasive Computing and Communications, Mar. 29 201, IEEE Xplore Press, Mannheim, pp: 70-78. DOI: 10.1109/PERCOM.2010.5466991

Shnayder, V., M. Hempstead, B. Chen, G.W. Allen and M. Welsh, 2004. Simulating the power consumption of large-scale sensor network applications. Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, Nov. 03-05, ACM Press, New York, USA., pp: 188-200. DOI: 10.1145/1031495.1031518

TI, 2002. Analyzing target system energy consumption in code composer studio™ IDE.

Tiwari, V., S. Malik, A. Wolfe and M.T. Lee, 1996. Instruction level power analysis and optimization of software. Proceedings of the VLSI Design, 9th International Conference, Jan. 3-6, IEEE Xplore Press, Bangalore, pp: 326-328. DOI: 10.1109/ICVD.1996.489624

Zeng, H., C.S. Ellis, A.R. Lebeck and A. Vahdat, 2002. ECOSystem: Managing energy as a first class operating system resource. Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems, Oct. 06-10, ACM Press, New York, USA., pp: 123-132. DOI: 10.1145/605397.605411

Zhang, L., B. Tiwana, Z. Qian, Z. Wang and R.P. Dick *et al*., 2010. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. Oct. 24-29, IEEE Xplore Press, Scottsdale, AZ., pp: 105-114. DOI: 10.1145/1878961.1878982