A Novel GPU Power Model for Accurate Smartphone Power Breakdown

Young Geun Kim, Minyong Kim, Jae Min Kim, Minyoung Sung, and Sung Woo Chung

As GPU power consumption in smartphones increases with more advanced graphic performance, it becomes essential to estimate GPU power consumption accurately. The conventional GPU power model assumes, simply, that a GPU consumes constant power when turned on; however, this is no longer true for recent smartphone GPUs. In this paper, we propose an accurate GPU power model for smartphones, considering newly adopted dynamic voltage and frequency scaling. For the proposed GPU power model, our evaluation results show that the error rate for system power estimation is as low as 2.9%, on average, and 4.6% in the worst case.

Keywords: GPU, power model, power breakdown, power consumption, smartphone.

Manuscript received Dec. 27, 2013; revised May 7, 2014; accepted May 17, 2014.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2012R1A2A2A01013816), the Technology Innovation Program (No. 10041332) funded by the Ministry of Knowledge Economy (MKE) of Korea, and Korea University.

Young Geun Kim (carrotyone@korea.ac.kr), Minyong Kim (mkim05@korea.ac.kr), Jae Min Kim (joist@korea.ac.kr), and Sung Woo Chung (corresponding author, swchung@korea.ac.kr) are with the Division of Computer and Communication Engineering, Korea University, Seoul, Rep. of Korea.

Minyoung Sung (mysung@uos.ac.kr) is with the Department of Mechanical and Information Engineering, University of Seoul, Rep. of Korea.

I. Introduction

Recently, the smartphone has become one of the most popular consumer electronics. To provide high graphics performance to users, many off-the-shelf smartphones adopt a high-end GPU. However, the adoption of a high-end GPU increases the power consumption of the smartphones. In turn, the increased power consumption results in a shorter battery lifetime.

To prolong the battery lifetime, it is essential to reduce GPU power consumption since the GPU is one of the most frequently utilized hardware components and accounts for a notable portion of the system power. For example, a GPU is utilized for the scrolling of a smartphone's home screen, which is one of the most basic operations in smartphones. Moreover, a GPU is heavily utilized for games, which are among the most popular applications. The GPU power consumption during a game's execution usually occupies up to one-fifth of the system power consumption [1].

When developing a GPU power reduction scheme, there should be a method to isolate the GPU power from the system power so as to be able to evaluate such a scheme. Unfortunately, GPU power cannot be directly measured by using a power measurement device, since smartphone GPUs are usually integrated in application processors. Therefore, without an accurate GPU power model, a GPU power reduction scheme cannot be evaluated.

The accuracy of a power model for a hardware component depends on how precisely the model captures the actual power behavior of the hardware. Since recent smartphone GPUs mostly adopt dynamic voltage and frequency scaling (DVFS) [2], GPU power consumption varies depending on the GPU voltage and frequency. Moreover, according to our power analysis on GPUs, GPU power consumption increases with GPU utilization. However, there has not been any technique that models the actual power behavior of a GPU.

In this paper, we propose an accurate GPU power model that considers the power behavior of recent smartphone GPUs. Our technique accurately estimates GPU power consumption based on GPU frequency and utilization in runtime. To isolate the GPU power portion from the system power, we must model the CPU, display, GPS, audio, and Wi-Fi. We evaluate the estimation accuracy of our proposed technic with real smartphone applications.

The rest of this paper is organized as follows. In Section II, we introduce previous online power estimation techniques. In Section III, we propose our GPU power model. In Section IV, we evaluate our proposed technique. Finally, in Section V, we conclude our work.

II. Related Work

1. Online Power Estimation Technique with GPU Power Model

There is only one smartphone power estimation technique that considers GPU power consumption [1]. Kim and Chung analyzed the power consumption of five applications that use a GPU and proposed a power estimation technique based on the analysis. Their technique models GPU power consumption using only the on/off status of the GPU. In other words, their technique assumes that a GPU consumes constant power only when turned on. However, GPU power consumption in recent smartphones varies with voltage and frequency, since recent smartphone GPUs adopt DVFS to reduce their power consumption. In addition, GPU power consumption increases in line with the utilization. Hence, their technique, which models GPU power consumption using only the on/off status of the GPU, is not appropriate for recent smartphone GPUs.

2. Online Power Estimation Techniques without GPU Power Model

PowerTutor is an online power estimation technique that models single-core CPUs, traditional LCD displays, GPSs, Wi-Fi, audio, and 3G cellular modules based on the assumption that the power consumption of a hardware component is proportional to its utilization [3]. In the case of the display power model, for example, PowerTutor assumes that the power consumption of a display is simply proportional to the backlight brightness. PowerTutor shows sufficiently high accuracy on single-core smartphones that employ traditional LCD displays. However, PowerTutor is not suitable for the power estimation of more recent smartphones, since most recent smartphones adopt multi-core CPUs and advanced display components. Note that the power consumption of each core varies depending on the frequency and utilization. For example, when only one core operates at the highest frequency, while the other cores operate at the lowest frequency, the one core at the highest frequency consumes much larger power than the others. In addition, the power consumption of the advanced display is not simply proportional to the backlight brightness. Most importantly, PowerTutor does not consider the power consumption of the GPU, which accounts for a significant portion of the system power in recent smartphones.

Kim and others proposed an advanced online power estimation technique that considers the actual power behavior of multi-core CPUs, advanced display components, and 3G cellular modules [4]. Firstly, in the case of multi-core CPU power consumption, their technique models the power consumption of each core separately. Secondly, taking advantage of a step function, their technique precisely models advanced displays, whose power consumption is not simply proportional to the backlight brightness. Finally, their technique models 3G cellular modules, of which the power consumption exponentially increases as the signal strength weakens. However, their technique also does not include a power model for GPUs.

Pathak and others proposed a power modeling technique for smartphones using system call tracing [5]. Their technique captures some system calls that invoke power-state transitions in runtime. Using the captured system calls, their power model, which is based on state machines, traces the power-state transitions of hardware components. Note that each state of their power model corresponds to a power consumption level of a hardware component and each system call corresponds to a transition between the states. By using the system call–based power model, their technique is able to model the power consumption of some hardware components, such as the Secure Digital card, whose power consumption levels change discretely according to system calls. However, their technique is not appropriate to model the power consumption of hardware components that are either changing in a continuous manner or that are unaffected by system calls. For example, their technique cannot accurately estimate GPU power consumption, which varies in accordance with frequency and utilization.

III. Novel GPU Power Model for Smartphones

In this section, we propose a novel GPU power model. To construct our proposed power model, we analyze the actual GPU power behavior. Based on the analysis, we model the GPU power, which varies in accordance with GPU frequency

http://dx.doi.org/10.4218/etrij.15.0113.1411

and GPU utilization. We also model the power consumption of other hardware components (CPU, display, audio, GPS, and Wi-Fi) to isolate the GPU power from the measured system power, based on the power modeling technique proposed in [4].

1. Proposed GPU Power Model

To analyze the GPU power behavior, we gradually increase GPU utilization at each GPU frequency level available in our target smartphone (128 MHz, 200 MHz, 325 MHz, and 400 MHz) by using a synthetic OpenGL application. At the same time, we measure the power consumption of the smartphone (system power) by using a power measurement device. To clearly observe the GPU power behavior, we exclude the power consumption of the other modelled hardware components (CPU, display, audio, GPS, Wi-Fi, and so on) from the measured system power by modeling their respective power consumption. Furthermore, to explore the impact of GPU frequency and GPU utilization on GPU dynamic power consumption, we also exclude the static power consumption of the GPU, since the static power consumption of a GPU is not affected by GPU frequency and GPU utilization.

As shown in Fig. 1 the GPU dynamic power increases along with GPU utilization at all frequency levels. Moreover, the growth rate of the dynamic power is larger at higher GPU frequency levels. As a result, the dynamic power consumption significantly varies depending on GPU frequency and GPU utilization. These results indicate that GPU frequency and GPU utilization should be considered to accurately model the dynamic power consumption of GPU.

Since the GPU dynamic power is affected by the GPU frequency and GPU utilization, whereas the GPU static power is affected by the GPU on/off status, we separately model the GPU dynamic power and the GPU static power. In the case of the dynamic power consumption of a GPU, our proposed model adopts variables for both the GPU frequency (F_{GPU}) and the GPU utilization (U_{GPU}). Since the GPU dynamic power almost linearly increases with the utilization, our model adopts a selector function ($\beta_{\text{GPU\_freq}}(x)$), which returns the growth rate of the GPU dynamic power at a given frequency x. Our model uses the returned growth rate as the coefficient for U_{GPU}. In the case of the static power consumption of a GPU, our model adopts the variable for the GPU's on/off status ($S_{\text{GPU\_on\_off}}$). The coefficient for $S_{\text{GPU\_on\_off}}$ is the static power consumption of the GPU ($\beta_{\text{GPU\_on}}$). Since a GPU does not consume any power when it is turned off, the variable $S_{\text{GPU\_on\_off}}$ can be added as a term to the GPU dynamic power model. Note that $S_{\text{GPU\_on\_off}}$ is "0" when the GPU is turned off and "1" when the GPU is turned on. Consequently, the power consumption of the GPU (P_{GPU}) is formulated as follows:

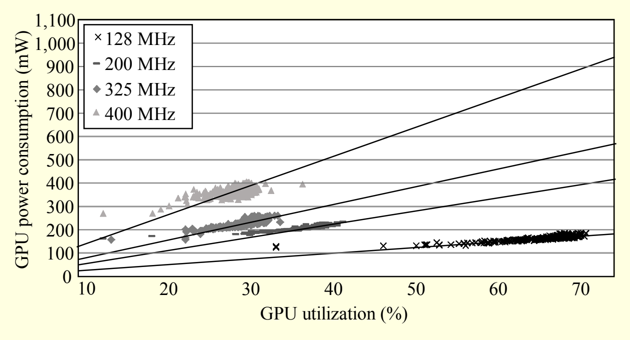

Fig. 1. Power behavior of GPU according to frequency and utilization.

Table 1. Coefficients for proposed GPU power model.

Coefficient.	Value
$\beta_{\text{GPU\_freq\_128}}$	2.5
$\beta_{\text{GPU\_freq\_200}}$	5.5
$\beta_{\text{GPU\_freq\_325}}$	7.5
$\beta_{\text{GPU\_freq\_400}}$	12.6
$\beta_{\text{GPU\_on}}$	90.8

$$P_{\text{GPU}} = \left(\beta_{\text{GPU\_freq}}\left(F_{\text{GPU}}\right) \times U_{\text{GPU}} + \beta_{\text{GPU\_on}}\right) \times S_{\text{GPU\_on\_off}}, \tag{1}$$

$$\beta_{\text{GPU\_freq}}(x) = \begin{cases} \beta_{\text{GPU\_freq\_128}} & \text{if } x = 128\,\text{MHz}, \\ \beta_{\text{GPU\_freq\_200}} & \text{if } x = 200\,\text{MHz}, \\ \beta_{\text{GPU\_freq\_325}} & \text{if } x = 325\,\text{MHz}, \\ \beta_{\text{GPU\_freq\_400}} & \text{if } x = 400\,\text{MHz}. \end{cases}$$

To obtain the coefficients for the dynamic power model, we utilize a linear regression–based method. We log the system power while increasing the GPU utilization at each GPU frequency level. From the logged system power, we exclude the power consumption of the other hardware components by utilizing their power models. In addition, we also exclude the static power consumption of the GPU. After that, we obtain the growth rate of the GPU dynamic power consumption at each frequency level by running a linear regression. To acquire the coefficient for the static power model, we subtract the power consumption of the other hardware components from the system power when the GPU is turned on without any load. Table 1 shows the obtained coefficients for our proposed power model.

2. Power Models for Other Hardware Components

As explained in the previous subsection, we isolate the GPU power from the measured system power by modeling the

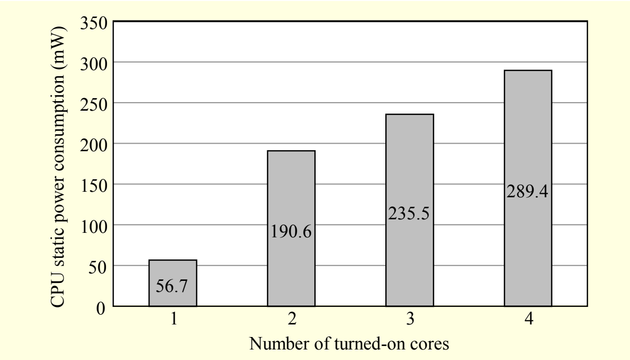

Fig. 2. CPU static power consumption for turned-on cores.

power consumption of the other hardware components that consume substantial power. We model the power consumption of the five hardware components (CPU, display, audio, GPS, and Wi-Fi) based on the power modeling technique proposed in [4]. In addition, we calibrate the coefficients of the power models for our target smartphone.

In the case of a multi-core CPU, the power modeling technique [4] models the power consumption of the CPU as each core consumes unique static power. However, in general, the static power of a multi-core CPU increases along with the number of turned-on cores, as shown in Fig. 2. Hence, we newly model the static power consumption of a multi-core CPU based on the number of cores instead of using the CPU power model proposed in [4]. Furthermore, we also consider the static power of our target CPU, which is not simply proportional to the number of turned-on cores. For example, in our target CPU, the static power difference between one turned-on core and two turned-on cores is 133.9 mW. On the other hand, the static power difference between two turned-on cores and three turned-on cores is only 44.9 mW. We reflect these characteristics of our target CPU in our CPU power model.

IV. Evaluation

In this section, we compare our proposed power model with two conventional power models in terms of the estimation accuracy. We present our evaluation environment in Section IV-1 and describe our evaluation in Section IV-2. In addition, we analyze each case of our evaluation in more detail in the subsections of Section IV-2. Note that we evaluate our proposed power model with representative applications and home screen scrolling (which is one of the most frequently used operations) on a real smartphone.

1. Evaluation Environment

In our evaluation, we compare the power estimation

Table 2. Description of power models for entire system

System power model	Description
$\text{SysPM}_{\text{w/GPU\_proposed}}$	Proposed GPU power model + $\text{SysPM}_{\text{w/oGPU}}$
$\text{SysPM}_{\text{w/GPU\_constant}}$	Constant GPU power model [1] + $\text{SysPM}_{\text{w/oGPU}}$
$\text{SysPM}_{\text{w/oGPU}}$	Power model of CPU, display, audio, GPS, and Wi-Fi [4]

accuracy of three system power models (shown in Table 2): the system power model with our proposed GPU power model ($\text{SysPM}_{\text{w/GPU\_proposed}}$), the system power model with the GPU power model which considers the CPU power to be constant ($\text{SysPM}_{\text{w/GPU\_constant}}$), and the system power model alone ($\text{SysPM}_{\text{w/oGPU}}$). The $\text{SysPM}_{\text{w/GPU\_proposed}}$ consists of our proposed GPU power model and $\text{SysPM}_{\text{w/oGPU}}$. On the other hand, $\text{SysPM}_{\text{w/GPU\_constant}}$ consists of a model that assumes the GPU power to be constant [1] and $\text{SysPM}_{\text{w/oGPU}}$. Note that $\text{SysPM}_{\text{w/oGPU}}$ is the power model for the five hardware components (CPU, display, audio, GPS, and Wi-Fi), except GPU [4]. For a fair comparison, we calibrate the coefficient of the GPU power model in $\text{SysPM}_{\text{w/GPU\_constant}}$ [1] as the average of the GPU power available in our target smartphone. Similarly, we calibrate the coefficients of $\text{SysPM}_{\text{w/oGPU}}$ [4] for our target device as well. Note that we evaluate the accuracy of the GPU power models in terms of system power estimation accuracy since it is impossible to directly measure the GPU power itself. We use Monsoon PowerMonitor [6], an external power measurement tool, to evaluate the accuracy of system power estimations.

We evaluate the accuracy of the system power models with two measures: E_{avg} (average error rate) and E_{abs} (average absolute error rate). We calculate the measures as follows:

$$E_{\text{avg}} = \frac{1}{n}\sum_{t=1}^{n}\frac{P_{\text{measured}}(t) - P_{\text{estimated}}(t)}{P_{\text{measured}}(t)},$$
$$E_{\text{abs}} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{P_{\text{measured}}(t) - P_{\text{estimated}}(t)}{P_{\text{measured}}(t)}\right|, \quad (2)$$

where $P_{\text{measured}}(t)$ is the system power measured by a power measurement device at time t, $P_{\text{estimated}}(t)$ is the system power estimated by the power models at time t, and n is the execution time in seconds. Note that a small E_{avg} means that a GPU power model accurately estimates the overall energy consumption of the GPU, while a small E_{abs} indicates that a GPU power model accurately estimates the transient power consumption of the GPU.

We evaluate the accuracy of the three system power models

Table 3. Description of target application and operation.

Category	Name	Description
Application	Angry Birds	2D game
	Droid Invaders	3D game
	Pie3dDemo (Styrofoam)	3D rendering
	Pie3dDemo (DnD Dice)	3D rendering
Operation	Home screen scrolling	-

on Google Nexus 4, one of the latest smartphones. Nexus 4 adopts a high-end GPU, Adreno 320, which provides four frequency levels: 128 MHz, 200 MHz, 325 MHz, and 400 MHz. The smartphone operates with Android 4.2 (Jelly Bean) and Linux 3.4 kernel.

As described in Table 3, we run three real smartphone applications and one basic smartphone operation that heavily utilizes the GPU in our evaluation. The three applications include two games with different GPU utilizations and one 3D rendering application. The two games are: a 2D game (Angry Birds) and a 3D game (Droid Invaders). For the 3D rendering application (Pie3dDemo), we run two different cases (Styrofoam and DnD Dice) of the application. The one basic operation is home screen scrolling, which is one of the most frequently used operations.

In our experiment, we do not consider the case where multiple applications simultaneously utilize the GPU, since, in reality, such a case does not seem to exist. Note that to execute two applications that utilize the GPU at the same time, one of them should run in the background due to the limited display size of smartphones. In such a case, the smartphone places the application running in the background into a sleep state. As a result, the application in the background does not utilize the GPU.

2. Evaluation Results

As shown in Fig. 3, $\text{SysPM}_{\text{w/GPU\_proposed}}$ precisely estimates the energy consumption and the transient power consumption; its E_{avg} and E_{abs} are always lower than 5%. On the other hand, $\text{SysPM}_{\text{w/GPU\_constant}}$ underestimates the system power consumption, since all the cases in our evaluation heavily utilize the GPU. Note that GPU power in $\text{SysPM}_{\text{w/GPU\_constant}}$ is assumed to be constant, which is the average of the GPU power. Moreover, $\text{SysPM}_{\text{w/oGPU}}$ also underestimates the system power consumption for all cases, since it estimates the GPU power to be zero. As a result, the E_{avg} of $\text{SysPM}_{\text{w/GPU\_proposed}}$ is 56.6% lower than that of $\text{SysPM}_{\text{w/GPU\_constant}}$ (from 5.3% to 2.3%) and 90.2% lower than that of $\text{SysPM}_{\text{w/oGPU}}$ (from 23.4% to 2.3%). Similarly, the E_{abs} of $\text{SysPM}_{\text{w/GPU\_proposed}}$ is 45.4% lower than that of $\text{SysPM}_{\text{w/GPU\_constant}}$ (from 6.6% to 3.6%) and 84.9% lower than that of $\text{SysPM}_{\text{w/oGPU}}$ (from 23.8% to 3.6%). We further analyze each of the evaluation cases in the following subsections.

A. Games

Figures 4 and 5 show the power comparison between the three models for the Droid Invaders and the Angry Birds, respectively. During the execution of the games, the GPU utilization fluctuates. It fluctuates more during the execution of the Droid Invaders than during that of the Angry Birds. In the case of the Droid Invaders, as shown in Fig. 4, $\text{SysPM}_{\text{w/GPU\_proposed}}$ most accurately traces the GPU power consumption, which varies during the execution, compared to the other power models. On the other hand, since $\text{SysPM}_{\text{w/GPU\_constant}}$ and $\text{SysPM}_{\text{w/oGPU}}$ estimate the GPU power as a constant and as zero, respectively, they cannot trace the change in GPU power consumption. For example, between 169 s and 205 s, the GPU power increases by approximately 100 mW due to the increase in GPU utilization (we isolate the GPU power consumption from the measured system power by using $\text{SysPM}_{\text{w/oGPU}}$). In spite of the increased GPU power consumption, the measured system power during the period does not vary by much, since the increase in the GPU power consumption is compensated by the decrease in the CPU power consumption, which is around 100 mW. During the period, $\text{SysPM}_{\text{w/GPU\_proposed}}$ accurately tracks the measured power consumption, since our proposed GPU power model is sensitive to the varying GPU frequency and GPU utilization. However, $\text{SysPM}_{\text{w/GPU\_constant}}$ underestimates the power consumption, since it is not able to detect the change in GPU frequency and GPU utilization. Consequently, the E_{abs} and E_{avg} of $\text{SysPM}_{\text{w/GPU\_proposed}}$ are only 3.1% and 2.0%, respectively.

In the case of Angry Birds, as shown in Fig. 5, $\text{SysPM}_{\text{w/GPU\_proposed}}$ accurately follows the power consumption trend, though it slightly underestimates the power consumption during certain periods. Actually, the underestimation is caused by $\text{SysPM}_{\text{w/oGPU}}$, which models the complex power behavior of the hardware components based on simple regression. Since the simple regression method only reflects the representative features of the complex power behavior of the hardware components, $\text{SysPM}_{\text{w/oGPU}}$ inevitably shows estimation errors in certain cases. In the case of Angry Birds, for example, $\text{SysPM}_{\text{w/oGPU}}$ largely underestimates the power consumption, though the actual GPU power consumption is small. Since we attach our GPU power model to the $\text{SysPM}_{\text{w/oGPU}}$, $\text{SysPM}_{\text{w/GPU\_proposed}}$ also underestimates the power consumption during certain periods of the Angry Birds game. Nevertheless, $\text{SysPM}_{\text{w/GPU\_proposed}}$ still shows the highest estimation accuracy,

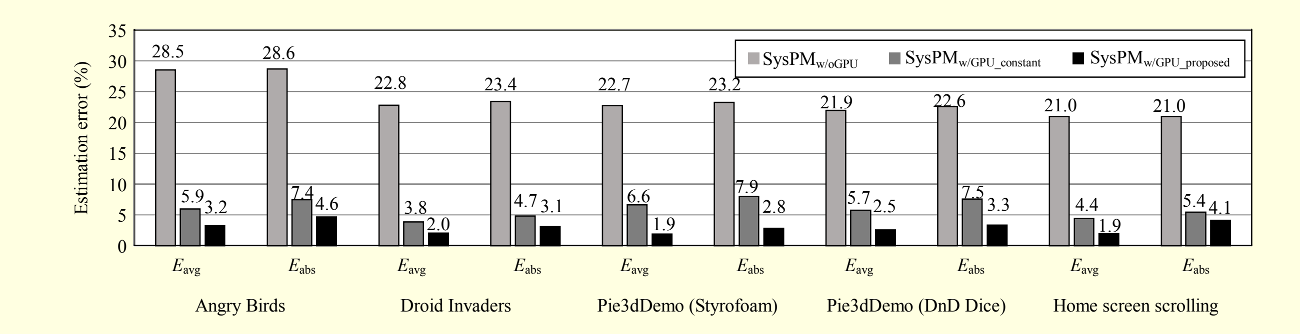

Fig. 3. Power estimation accuracy enhancement by proposed GPU power model.

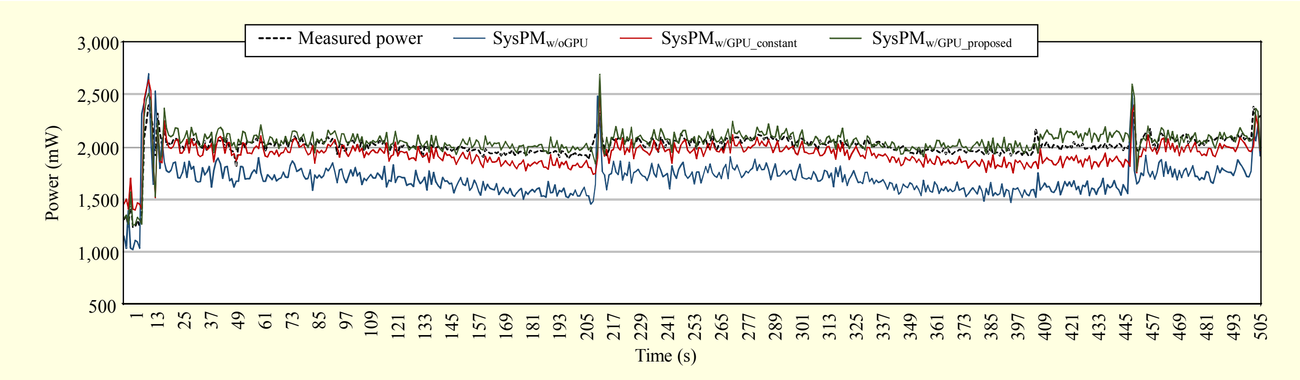

Fig. 4. Power comparison for Droid Invaders.

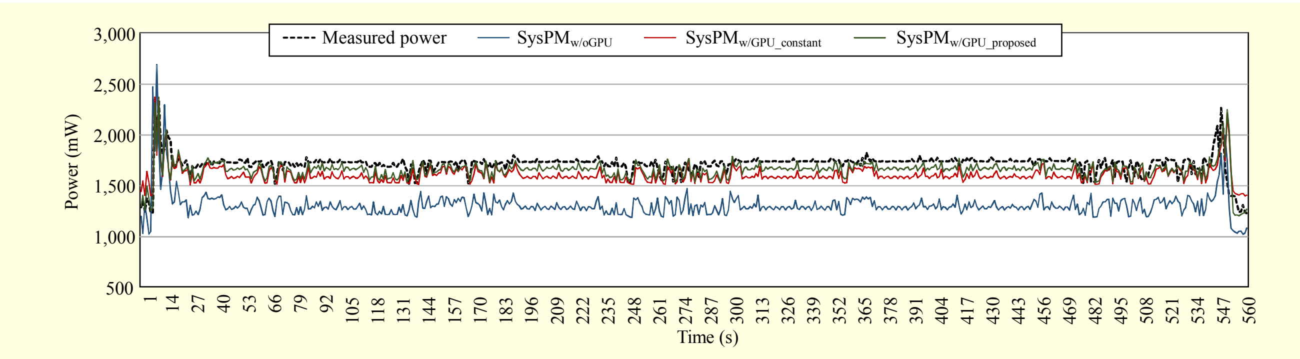

Fig. 5. Power comparison for Angry Birds.

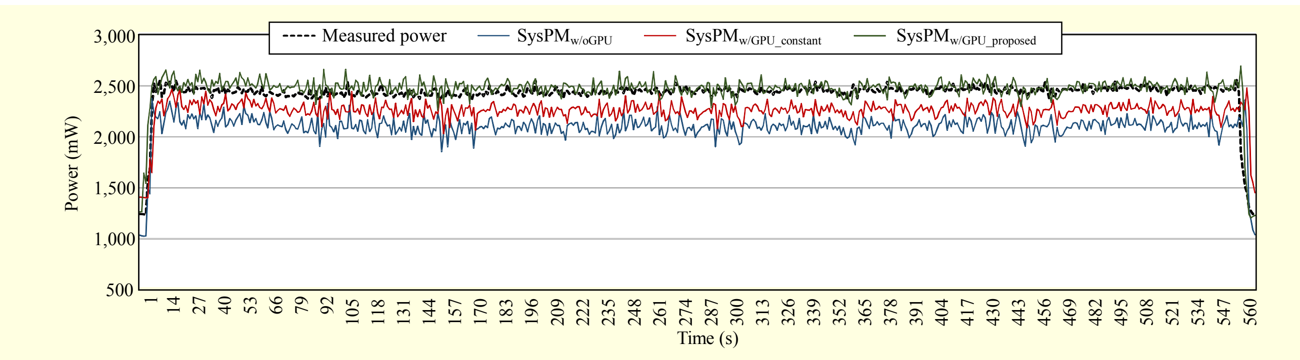

Fig. 6. Power comparison for Pie3dDemo (Styrofoam).

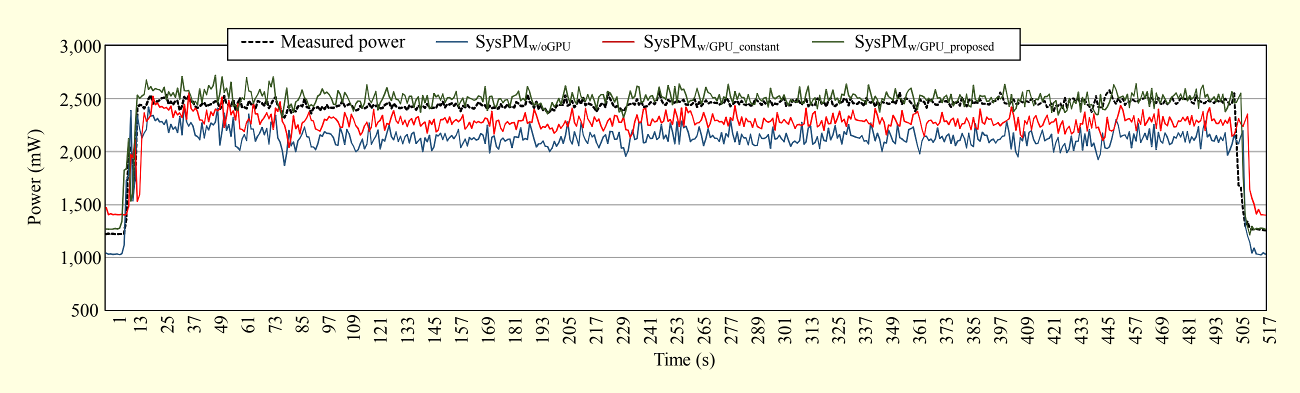

Fig. 7. Power comparison for Pie3dDemo (DnD Dice).

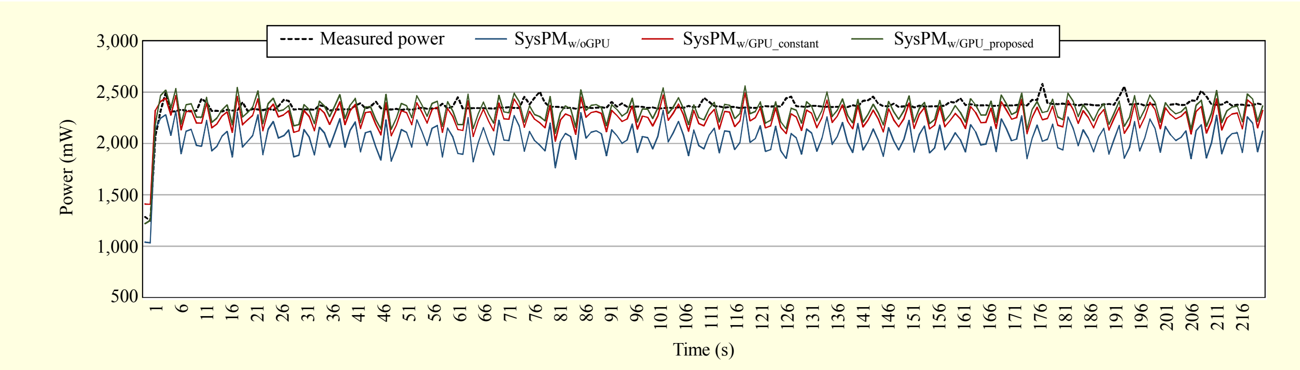

Fig. 8. Power comparison for home screen scrolling.

since our proposed GPU power model accurately tracks the power consumption of the GPU. Consequently, the E_{abs} and E_{avg} of SysPM$_{w/GPU_proposed}$ are only 4.6% and 3.2%, respectively.

B. Rendering Application

Figures 6 and 7 show the power comparison between the three models for the 3D rendering application (Pie3dDemo) in two different rendering cases (Styrofoam and DnD Dice), respectively. Since both rendering cases utilize the GPU heavily, the GPU utilization is kept high at the highest frequency throughout the execution of the rendering cases. For both rendering cases, SysPM$_{w/GPU_proposed}$ accurately traces the transient power consumption obtained from the power measurement device, since it captures the high frequency and utilization of the GPU. However, SysPM$_{w/GPU_constant}$ underestimates the power consumption throughout the execution of both the rendering cases, since the gap between the actual GPU power and the constant value (the GPU power in SysPM$_{w/GPU_constant}$) widens at high frequencies and utilization. Similarly, SysPM$_{w/oGPU}$ also underestimates the power consumption of both the rendering cases, since it estimates the GPU power to be zero. As a result, in the case of Styrofoam, the E_{avg} and E_{abs} of SysPM$_{w/GPU_proposed}$ are only 1.9% and 2.8%, respectively. In addition, in the case of DnD Dice, the E_{avg} and E_{abs} of SysPM$_{w/GPU_proposed}$ are 2.5% and 3.3%, respectively.

C. Home Screen Scrolling

We evaluate the accuracy of the three power models while constantly scrolling the home screen between 6 s and 216 s. Since our proposed GPU power model considers the varying GPU frequency and GPU utilization, SysPM$_{w/GPU_proposed}$ accurately estimates the power consumption throughout the execution. In the case of home screen scrolling, as shown in Fig. 8, SysPM$_{w/GPU_constant}$ does not underestimate the power consumption as much as it does in the other cases (Droid Invaders, Angry Birds, and the two Pie3dDemos), since the GPU power consumption during the home screen scrolling is similar to the constant GPU power of SysPM$_{w/GPU_constant}$. As a result, in this case, the E_{avg} and E_{abs} of SysPM$_{w/GPU_constant}$ are 4.4% and 5.4%, respectively. However, the E_{avg} and E_{abs} of SysPM$_{w/GPU_proposed}$ are still lower than that of SysPM$_{w/GPU_constant}$, which are 1.9% and 4.1%, respectively.

V. Conclusion and Future Work

In this paper, we proposed a novel GPU power model that considers the varying power consumption of recent smartphone GPUs. By taking GPU frequency and GPU utilization into account, our proposed power model accurately traces the actual GPU power behavior. Our evaluation results show that the estimation accuracy of the system power model with our proposed GPU power model (97.1%, on average) is higher than that of the system power model with the constant GPU power model (94.1%, on average). We expect our proposed technique to be widely adopted in many smartphone power modeling techniques and are confident that it can significantly enhance the accuracy of such techniques.

As a future work, we plan to develop a method that automatically calibrates our power model for different smartphones, to encourage end users to adopt our power model.

References

[1] M. Kim and S.W. Chung, "Accurate GPU Power Estimation for Mobile Device Power Profiling," *Proc. IEEE Int. Conf. Consum. Electron.*, Las Vegas, NV, USA, Jan. 11–14, 2013, pp. 183–184.

[2] D. You and K.S. Chung, "Dynamic Voltage and Frequency Scaling Framework for Low-Power Embedded GPUs," *Electron. Lett.*, vol. 48, no. 21, Oct. 2012, pp. 1333–1334.

[3] L. Zhang et al., "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones," *IEEE/ACM/IFIP, Conf. Hardware/Software Codes. Syst. Synthesis*, Scottsdale, AZ, USA, Oct. 24–29, 2010, pp. 105–114.

[4] M. Kim, J. Kong, and S.W. Chung, "Enhancing Online Power Estimation Accuracy for Smartphones," *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, May 2012, pp. 333–339.

[5] A. Pathak et al., "Fine-Grained Power Modeling for Smartphones Using System Call Tracing," *Proc. European Conf. Comput. Syst.*, Salzburg, Austria, Apr. 10–13, 2011, pp. 153–168.

[6] PowerMonitor, Monsoon Solutions, 2008. Accessed Dec. 20, 2013. http://www.msoon.com/LabEquipment/PowerMonitor/

Young Geun Kim received his BS degree in computer and communication engineering from the Division of Computer and Communication Engineering, Korea University, Seoul, Rep. of Korea, in 2014. He is currently an MS student in the Division of Computer and Communication Engineering, Korea University. His research interests include low-power design and battery-aware design.

Minyong Kim received his BS and MS degrees in computer science from the Division of Computer Science, Korea University, Seoul, Rep. of Korea, in 2010 and 2012, respectively. He is currently a PhD student in the Division of Computer and Communication Engineering, Korea University. His research interests include system-on-chip design, user-aware design, and design verification methods.

Jae Min Kim received his BS degree in computer science from the Division of Computer Science, Korea University, Seoul, Rep. of Korea, in 2010. He is currently a PhD student in the Division of Computer and Communication Engineering, Korea University. His research interests include user-aware design, computer architecture, and low-power design.

Minyoung Sung received his PhD degree in computer engineering from the Division of Computer Engineering, Seoul National University, Seoul, Rep. of Korea, in 2002. He is currently an assistant professor in the Department of Mechanical and Information Engineering, University of Seoul, Seoul, Rep. of Korea. He also worked as a senior engineer at Samsung Electronics, Suwon, Rep. of Korea from October in 2002 to August in 2006. His research interests include real-time control systems and system software.

Sung Woo Chung received his BS, MS, and PhD degrees in electrical engineering and computer science from the Division of Electrical Engineering and Computer Science, Seoul National University, Seoul, Rep. of Korea, in 1996, 1998, and 2003, respectively. He is currently an associate professor in the Division of Computer and Communication Engineering, Korea University, Seoul, Rep. of Korea. His research interests include low-power design, temperature-aware design, and user-aware design. He has served (and still serves) on various technical program committees, including on that of the ICCD, at many conferences. He is currently an associate editor of IEEE Transactions on Computers and a senior member of the IEEE.