Demo: Optimizing Smartphone Power Consumption through Dynamic Resolution Scaling

Songtao He[1,2], Yunxin Liu[1], Hucheng Zhou[1]

[1]Microsoft Research, Beijing, China
[2]University of Science and Technology of China, Hefei, China

hst@mail.ustc.edu.cn, {yunliu, huzho}@microsoft.com

ABSTRACT

The extremely-high display density of modern smartphones imposes a significant burden on power consumption, yet does not always provide an improved user experience and may even lead to a compromised user experience. As human visually-perceivable ability highly depends on the user-screen distance, a reduced display resolution may still achieve the same user experience when the user-screen distance is large. This provides new power-saving opportunities.

We present a flexible dynamic resolution scaling system for smartphones. The system adopts an ultrasonic-based approach to detect the user-screen distance at low-power cost and makes scaling decisions automatically for maximum user experience and power saving. App developers or users can also adjust the resolution manually and dynamically as their needs. Our system is able to work on the existing commercial smartphones and support the legacy apps, without requiring re-building the ROM or any changes from apps.

Categories and Subject Descriptors

I.3.4 [**Computer Graphics**]: Graphics Utilities—*Software support*

General Terms

Experiments; Measurement; Performance

Keywords

Smartphone; GPU; Power Consumption; Display Resolution; Dynamic Resolution Scaling

1. INTRODUCTION

The display resolution of smartphones has become increasingly high. Since the “Retina display” of the iPhone 4 released in 2010 [5], display resolution of smartphones has continued to rise, from 960x640 pixels of the iPhone 4 to 1920x1080 pixels (Full HD) and recently to 2560x1440 pixels (2K). As a result, latest smartphones have an extremely-high display density. For example, the LG G3 [3] has a display of 538 pixels per inch (ppi), while the Samsung Galaxy S5 LTE-A [4] has a display of 577 ppi.

However, this extremely-high display density does not always help bring an improved user experience. The Retina display of the iPhone 4 has a display density of only 326 ppi. The high display density of the LG G3 and the Samsung Galaxy S5 LTE-A is far beyond the human ability of visual acuity, and thus does not always lead to visibly-sharper content and interfaces displayed on the screen. Nevertheless, a high-resolution display consumes a large amount of system resources, especially the GPU computation, and thus results in high system power consumption. As energy is a paramount concern on smartphones, it is desirable to study the tradeoff between display density and power consumption.

In this demo, we propose to optimize the power consumption of high-density displays through Dynamic Resolution Scaling (DRS). DRS dynamically adjusts the display resolution based on the viewing distance (i.e., the screen-user distance) but ensures that the pixels are always small enough to be individually unobservable by human eye. The philosophy of DRS is that the system should render the interface with exactly as much detail as is perceptible to a user, in order to maximize both user experience and battery life. When a display resolution is too high for a viewing distance, DRS reduces the display resolution to save power.

Enabling DRS imposes several requirements. First, DRS must seamlessly adjust display resolution in real-time as the viewing distance changes, without compromising the user experience. Second, DRS must be done systematically and is transparent from applications. Third, DRS must be able to determine the viewing distance of a user in real-time and with minimal energy cost.

To meet the requirements, we develop per-frame DRS technique for seamless and real-time display-resolution adjusting. We intercept the system graphics pipeline to enable system-wide DRS, without requiring any changes or recompiling from applications, or re-building a new smartphone ROM. And we propose to use ultrasonic sensors to detect the viewing distance of a user, which is accurate and low-power. Based on the detected viewing distance, we utilize existing knowledge of the human visual system to define the required display-pixel density for optimal user experience, and adjust the extravagant display resolution accordingly.

We have implemented a DRS system that works on existing commercial smartphones including the Galaxy S5 LTE-A and the Nexus 6. We conduct comprehensive experi-

MobiCom'15, September 7–11, 2015, Paris, France.
ACM ISBN 978-1-4503-3619-2/15/09.
DOI: http://dx.doi.org/10.1145/2789168.2789175.

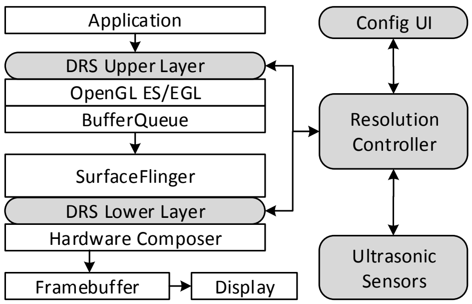


Figure 1: DRS System Architecture.

ments and a user study to evaluate the DRS implementation against 30 real gaming and benchmark applications. Experimental results show that all the 30 applications can run successfully with per-frame, real-time dynamic resolution scaling. The energy per frame can be reduced by 30.1% in average and up to 60.5% at most when the resolution is halved, for 15 applications. Even for other less GPU-intensive applications such as Adobe Reader and web browser, we still reduce the energy per frame for about 10%. A user study with 10 users indicates that our system remains good user experience: none of the 10 users could perceive the resolution changes in the user study although the resolution changed for more than 100 times for each user.

2. SYSTEM OVERVIEW

Enabling DRS on existing smartphones imposes a couple of challenges. First, we must reduce the number of pixels in all the pixel-related computations from the complex procedures of rasterization and pixel-processing involving many OpenGL function calls. Second, OpenGL libraries are close source and thus we cannot revise and re-compile the source code to add new functionalities. And we cannot require rebuilding a smartphone ROM or making application-specfic modifications. Third, we must design a lightweight approach to decide the user-screen distance for automatic resolution scaling.

To address the above challenges, we employ binary-rewriting techniques that are able to hook and intercept the function calls in existing binaries and add new functionalities. As a result, we can enable DRS on commercial smartphones without requiring any source-code level changes. Meanwhile, to detect the user-screen distance, we design a lower-power ultrasonic based approach to minimize the overhead.

Figure 1 shows the architecture of our DRS system. To enable resolution scaling, we add two new layers into the existing android graphic system[1].

The first layer is the *DRS Upper Layer* that sits between the application layer and and the OpenGL ES/EGL layer. It intercepts the necessary OpenGL ES function calls and EGL function calls and applies a scaling factor to the parameters of the those function calls to transform the default display region to a targeted smaller one, which locates at the bottom-left corner of the original graphic buffer.

The second layer is the *DRS Lower Layer* that locates between the SurfaceFlinger layer and the Hardware Composer. This layer intercepts the function call passed to the Hardware Composer, and let the Hardware Composer scale up the bottom-left region of the passed in graphic buffer to the original size using the same scaling factor.

By using this approach, we can reduce the energy cost of generating the high resolution image, and still make the image appear at the original region of the screen with a smaller virtual resolution.

In fact, the detailed parameter interception strategy is carefully designed to ensure the correctness of the scaling transformation. Meanwhile, a synchronization mechanism is also designed to synchronize the DRS upper layer and DRS lower layer. Due to the BufferQueue locates between these two layers, we need this synchronization mechanism to make sure that they use the same target display resolution for the same graphics buffer, while the resolution is changing for each frame.

To decide a proper targeted display resolution, we need to know the viewing-distance of a user. We add ultrasonic sensors[2] into the system for instant and accurate viewing-distance detection. Based on the measured viewing-distance, the *Resolution Controller* calculates the best display resolution for maximum user experience and maximum power optimization, and indicates the two DRS layers to do scaling transformation for the calculated display resolution.

In order to connect the user-screen distance and resolvable pixels, we use angular resolving acuity to define the visual acuity of human. According to the standards for visual acuity[6], Normal vision[1] can be represented with a angular resolving acuity of $\delta_{normal} = 2.9 \times 10^{-4}$ radians, and $\delta_{optimal} = 1.45 \times 10^{-4}$ radians for optimal vision[2].

For a specific device, we consider the number of pixels at the longer side of the display as resolvable pixel number when the pixel density of it can just meet the users' visual acuity. Resolvable pixel number varies with different user-screen distance and different user visual acuity. The relationship among resolvable pixel number, user-screen distance and user visual acuity can be approximately described using equation

$$N = \frac{L}{2D\ tan\left(\frac{\delta}{2}\right)} \tag{1}$$

where N is the resolvable pixel number, L is the length of the longer side of the display, D is the user screen distance and δ is the angular resolving acuity of the user.

Figure 2 demonstrates the variation of resolvable pixels number with user screen distance for users with normal vision and optimal vision on the Galaxy S5 LTE-A, the high display density of which can surpass the visual acuity of users with normal vision when the user-screen distance is beyond 5.4 inches, and 10.8 inches for users with optimal vision. After the user-screen distance reach these thresholds, our DRS system will dynamically reduce the display resolution based on equation(1) for maximum power optimization without compromising user experience.

As shown in Table 1, we evaluate the potential power optimization room for 15 apps (16 scenes) on the Galaxy S5 LTE-A. When locking the GPU Frequency to 500MHz, the energy per frame (whole system) can be reduced by 30.1% in average and up to 60.5% at most when the resolution is halved.

[1]Normal vision is commonly referred as 20/20 vision (vulgar fraction expression) or 1.0 (decimal number expression).
[2]Optimal vision is commonly referred as 20/10 vision (vulgar fraction expression) or 2.0 (decimal number expression).

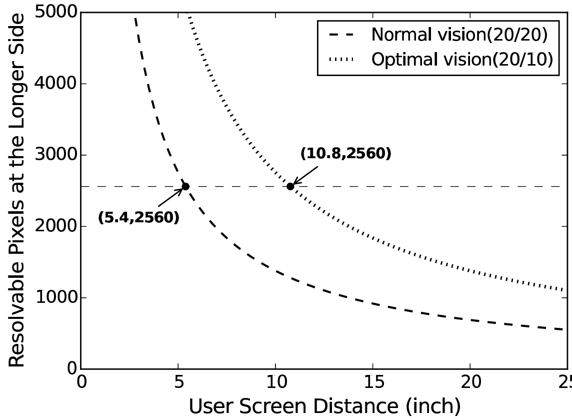


Figure 2: Relationship among resolvable pixel number, user-screen distance and user visual acuity.

Application(Scene) Name	Scale Factor 0.9	0.8	0.65	0.5
Games				
Badland	95.1%	91.2%	86.7%	83.5%
Hitman Sniper	94.0%	85.5%	80.6%	73.0%
Iron man 3	94.5%	90.1%	83.4%	75.9%
Leo's Fotune	95.4%	92.8%	88.7%	84.3%
Minecraft PE	94.5%	90.4%	82.6%	76.1%
NFS Most Wanted	94.3%	89.6%	79.7%	72.3%
Over Kill 3	95.7%	92.0%	88.3%	82.5%
Ridge Racer Slipstream	94.5%	85.0%	76.8%	65.9%
Riptide GP2	95.3%	89.0%	77.5%	68.7%
Shine Runner	95.9%	90.1%	81.2%	74.2%
Smash Hit	93.9%	89.5%	81.1%	72.8%
Temple Run Brave	93.7%	85.6%	78.9%	71.2%
Tiny Troopers 2	94.4%	90.0%	82.7%	76.1%
Warships	95.3%	90.7%	84.3%	76.4%
Graphics Benchmark				
GFX Bench Manhattan	84.2%	70.4%	56.7%	39.5%
GFX Bench T-Rex	85.9%	73.1%	57.6%	48.0%

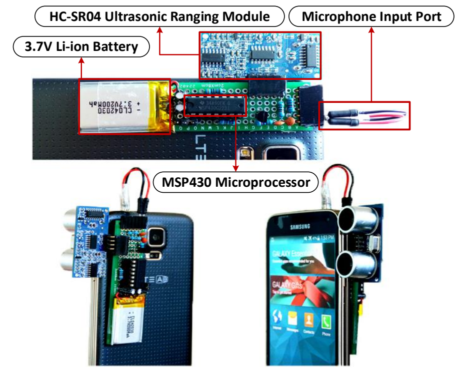
Table 1: Normalized energy per frame of games and benchmarks in different scaling factors.

3. DEMONSTRATION

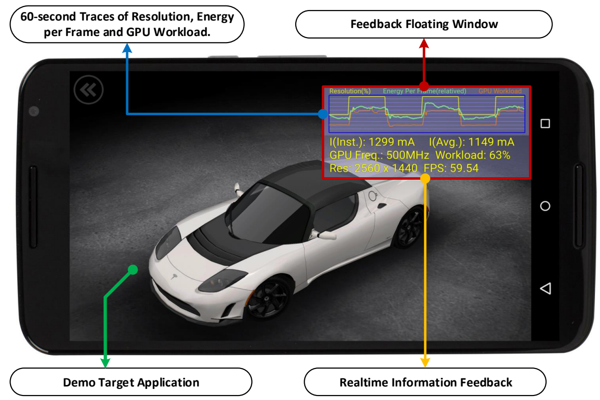
In this demo, we will show the power consumption optimizing opportunity by using our dynamic resolution scaling (DRS) system. We have implemented our prototype system on two smartphones, a Galaxy S5 LTE-A and a Nexus 6. Figure 3(a) shows a prototype system based on Galaxy S5 LTE-A. In the demo, we will provide each participant with our prototype system, and let them have a video game or watching a game scene. Meanwhile, they are encouraged to moving the prototype freely, so that the user-screen distance can vary and the display resolution will change properly.

As shown in Figure 3(b), we put a floating window at the top-right corner of the screen. In that window, we show the feedback information in the past 60 seconds of our prototype system, including the display resolution, GPU information and power consumption measured by smartphone's internal sensor. Thus, the power consumption's variation and other variations caused by DRS can be easily observed by participants.

Equipments and Setup. Our demo equipments are simple. We will bring our prototype system and a laptop by



(a) Demo System Based on Galaxy S5 LTE-A



(b) Demo Realtime Feedback UI

Figure 3: Demonstration System

ourselves. And the setup time of our demo is within 5 minutes.

4. DISCUSSION

To the best of our knowledge, we are the first to design and implement a DRS system for smartphones. Based on the measured user-screen distance, our system can enable automatic resolution scaling for maximum user experience and power saving. Besides the user-distance driven resolution scaling, our DRS system can also be used as an infrastructure to support a series of use cases which are remained to be studied in the future work toward alleviating performance, energy and over heating issue on smartphones and improving user experience.

5. REFERENCES

[1] Android Graphics Architecture. `https://source.android.com/devices/graphics/architecture.html`.

[2] Ultrasonic Ranging Module HC - SR04. `http://www.micropik.com/PDF/HCSR04.pdf`.

[3] LG G3. `http://www.lg.com/us/mobile-phones/g3`, Feb. 2015.

[4] Samsung Galaxy S5 LTE-A. http://www.samsung.com/us/news/23327, Feb. 2015.

[5] Apple iPhone. `http://www.apple.com/iphone/features/retina-display.html`, Jun. 2010.

[6] E. Messina. Standards for visual acuity. *National Institute for Standards and Technology*, 2006.