Poster: Towards Mobile GPU-Accelerated Context Processing for Continuous Sensing Applications on Smartphones

Chulhong Min[1], Wookhyun Han[2], Inseok Hwang[1], SangJeong Lee[1], Youngki Lee[1], Insik Shin[3], Junehwa Song[1]
Computer Science Department, KAIST
{chulhong, inseok, peterlee, youngki, junesong}@nclab.kaist.ac.kr[1]
hanwook@cps.kaist.ac.kr[2] insik.shin@cs.kaist.ac.kr[3]

Categories and Subject Descriptors
K.8 [**Personal Computing**]: General; C.3 [**Special-Purpose and Application-based Systems**]: Real-time and embedded systems

Keywords
Acceleration, Context, GPU, Mobile sensing, Smartphone

1. MOTIVATION
Mobile sensing applications (MS apps) are coming into the spotlight as next generation applications on smartphones. These applications continuously monitor a user's situational contexts using mobile sensors and provide the user with proactive services. To prevent obesity and help weight control, for instance, the smartphone can track daily calories expended by physical activity. A number of such applications will be running simultaneously and contribute to the quality of user experiences through diverse services.

Despite their usefulness, many of these applications are not widely deployed yet. This is because they require a considerable amount of resources in a consistent manner. According to our measurement, indoor place tracking with ambient sound fingerprint continuously consumes about 50% of CPU utilization and 600mW of energy on Nexus One phone; note that GPS consumes about 450mW on average. With such high CPU overhead, the smartphone is able to support only a small number of the applications. The performance of other user-interactive applications would be degraded due to the CPU conflict. Also, such continuous energy consumption shortens the battery lifetime.

2. VISION
In this poster, we present the vision of mobile GPU-accelerated context processing for sensing applications on smartphones. The key idea is to leverage Graphics Processing Units (GPUs) on smartphones for context processing. By offloading a part of computation into GPUs, the GPU-accelerated context processing disburdens a smartphone's CPUs and thus achieves performance and energy benefits. A number of GPU-enhanced MS apps would be able to run in the background while little causing the foreground applications to slug or the smartphone battery to be depleted in the middle of a day.

The key design philosophies of GPUs share many analogies with the major workload characteristics of MS apps. GPUs are inherently designed to support highly parallel, iterative computation workload; continuous stream of voluminous data under low-latency and high bandwidth requirement. A typical example would be the real-time processing of a large dataset of textures and polygons through a well- defined pipeline consisting of Gouraud shading, anti-aliasing, etc. Similarly, many MS apps consist of pipelines with multi-step operations, each of which is an iterative signal processing operation such as FFT. Executing these operations often involves piece-wise computation on a large volume of windowed frames of raw data, in which we can expect a large potential of parallelism as in the common GPU tasks. More similarities lie in that many MS apps impose latency requirements, e.g., a few hundreds of milliseconds in hand gesture recognition. Moreover, the use of GPUs instead of CPUs often yields great energy-saving benefits, which is a critical issue in mobile systems.

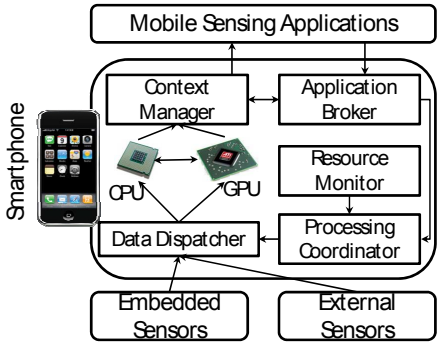

Figure 1. Overall architecture

Figure 1 shows the overall architecture. The *Application Broker* interacts with MS apps through a set of APIs we provide; the support for APIs is essential due to the lack of knowledge about underlying execution on GPUs. The *Processing Coordinator* carefully partitions the context processing computation into CPU and GPU parts in the way of achieving the maximum benefit. According to the decision, the *Data Dispatcher* properly dispatches the sensing data to CPU and GPU.

We believe that mobile GPU-accelerated context processing will be one of the key enabling forces to take the MS apps out of the lab and make them everyday reality. With them, people could enjoy diverse useful and proactive services without compromising the user experiences with their phones. It also would have substantial impact on mobile computing system as a first attempt to leverage new opportunities of mobile GPU for MS apps.

MobiSys'12, June 25–29, 2012, Low Wood Bay, Lake District, UK.
ACM 978-1-4503-1301-8/12/06