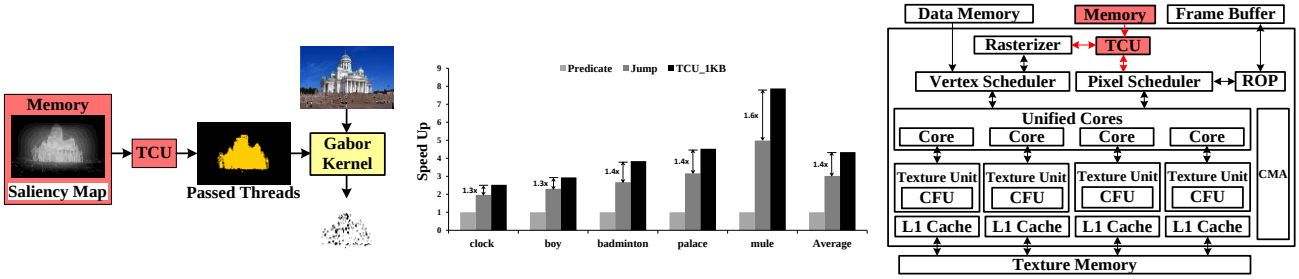
Early-stage Thread Culling Unit for Mobile GPGPU Applications

Yu-Jung Chen, Pai-Shun Ting, Meng-Lin Yu, and Shao-Yi Chien
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University



(a) (b) (c)

Figure 1: *(a) Salient region feature extraction with proposed TCU. (b) Speedup of (a) with predicate, jump and proposed TCU. (c) GPU architecture.*

1. Introduction

Redundant or unnecessary threads execution devastates parallel computing performance in GPUs. To efficiently utilize processor resources in GPUs for rendering applications, various techniques have been proposed in a graphics pipeline. Early-z culling [Hasselgren and Akenine-Möller 2007] avoids the redundant shader execution for occluded fragments; early-stencil culling [del Barrio et al. 2006] and scissor test in the rasterization stage can merely permit performing shader programs for fragments in validated regions. These features greatly enhance the rendering performance.

To utilize the plentiful computing resources of GPUs, it is known that most low-level image processing and computer vision algorithms can be effectively parallelized on GPUs due to the operation independence between regions or pixels, such as image filtering and linear feature extraction; however, it is not the case for high-level applications, where object-based processing and *Region of Interest* (ROI) are often involved. Not every extracted feature is significant for later processing stages and substantial feature information is generally concentrated in salient regions or ROI.

Similar to the early-stage fragment culling in rendering architectures, our work extends the concept to handle such predictable divergent thread behavior for multimedia GPGPU applications. A simple yet efficient configurable early-stage thread culling unit (TCU) is proposed into our mobile GPU architecture to mitigate the execution efforts of divergent threads with identified conditions.

2. Early-stage Thread Culling Unit

Many high-level computer vision applications have the characteristics of determining validated candidates by evaluating the filtered coefficients or masks. Once incorporating such regional scheme, allocated parallel threads are altered into sparse or local grouping threads. It induces low processor utilization for divergent thread execution since the branch instructions repetitively invoke processors to determine the effective branches, and the synchronization among divergent threads required to be carefully managed. Alternatively, predicate instruction tackles the synchronization problem through performing both taken and not taken threads, but only validating the taken fraction. Although predicate execution relieves the synchronization efforts of managing divergent threads, redundant thread execution degrades the performance.

Extending the concept of early-stencil test, our proposed early-stage thread culling unit (TCU) can enhance the processing efficiency by pre-determining the branch condition of divergent threads. Each parallel thread has to fetch the coefficient or pre-processed mask value for determining the passed or failed case. Applying the early fetch-and-compare architecture, the proposed TCU, as shown in Figure 1(c), is integrated with the rasterizer. It can fetch the corresponding coefficients or mask values from a dedicated memory and validate effective threads through certain comparing configurations. However, when the throughput of the culling unit is not high enough, processors are possibly stalled to wait for dispatched threads. For the purpose of efficiency, the culling unit is employed with a cache. The cache size is determined by evaluating the hit rate and processor idle ratio, and a 1KB cache is selected after analysis.

3. Results

To evaluate the performance of the proposed architecture, Viola-Jones face detection framework and salient region linear feature extraction algorithms are chosen for analysis. Our simulation results show that, with early-stage thread culling unit, GPUs can improve up to 24.5x and 1.8x in performance for Viola-Jones face detection framework compared to the predicate execution and branch instruction. Furthermore, 4.3x and 1.4x improvement in salient region linear feature extraction can be achieved as well.

References

DEL BARRIO, V., GONZALEZ, C., ROCA, J., FERNANDEZ, A., AND ESPASA, R. 2006. ATTILA: A cycle-level execution-driven simulator for modern GPU architectures. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*. IEEE, 231–241.

HASSELGREN, J., AND AKENINE-Möller, T. 2007. PCU: the programmable culling unit. In *Proc. of SIGGRAPH '07*, ACM.