GPU Computing

Overview The GPU is designed for a particular class of applications with the following characteristics.

- **Computational requirements are large** Real-time rendering requires billions of pixels per second, and each pixel requires hundreds of operations.
- **Parallelism is substantial** Operations on vertices and fragments are mostly independent without little interaction between parallel computations.
- **Throughput is more important than latency** Due to nature of human visual system, latency is less important. As a results, GPU pipelines are very deep, like hundreds to thousands of cycles, with thousands of primitives in flight at any given time.

GPU Programming Model The programmable units of the GPU follow a *single program multiple-data (SPMD)* programming model. For efficiency, the GPU processes many elements in parallel using the same program. Each element is *independent* of the other elements, and in the base programming model, elements cannot communicate with each other. All GPU programs must be structured in this way: *many parallel elements, each processed in parallel by a single program.*

Each element can operate on 32-bit integer or floating-point data with a reasonably generate-purpose instruction set. Elements can read data from a *shared global memory* (a **gather** operation) and, with the newest GPUs, also write back to arbitrary locations in shared global memory (a **scatter** operation).

Branching in GPU elements Allowing a different execution path for each element requires a substantial amount of control hardware. Instead, today's GPUs support arbitrary control flow per thread but impose a penalty for incoherent branching. Elements are grouped together into blocks, and blocks are processed in parallel. If elements branch in different directions within a block, the hardware computes both sides of the branch for all elements in the block. The size of block is known as the **branch granularity** and has been decreasing with recent GPU generations – today, is on the order of 16 elements.

In GPU programs, the branches are permitted but not free. To make the best use of the GPU hardware, blocks should have coherent branches.