Introduction

- Increasing texture use requires dynamic loading of textures
- Traditionally ad-hoc methods are used
  - Mipmap streaming
  - Clipmapping (Mega Texture)
- Virtual texturing offers generalized texture streaming
  - Up to several gigapixels of texture data
  - Works on arbitrary geometry & unwraps
- Expensive to implement
  - Requires frame analysis to determine working set
  - Dynamic streaming & compression of tiles needed
  - However, several tasks are suitable for parallel processing

Virtual Texturing: an Overview

- Based on virtual memory
  - Split texture into fixed size pages
  - Also uses pages for mipmap levels

Determining the Working Set using CUDA

- Determines the list of pages needed by the current view
- Analyses buffer rendered using special shader (OpenGL)
  - Kernel 1: Mark this frame’s used pageIDs
  - Kernel 2: Pack the list of pageIDs for transfer
  - Asynchronous transfer to the CPU

Tile Uploading & Processing using CUDA

- Pages are asynchronously uploaded in separate streams
- Mipmaps are generated (for anisotropic filtering)
- DXT compression is executed
- Copy to page texture (Using OpenGL)

Preliminary Results

- Measured on a Intel Core 2 Quad and NVIDIA GeForce GTX 285, averaged over 100 frames
  - Task | Time
  - Resolve frame | 1.2 ms
  - Regenerate page table | 0.7 ms
  - Upload, mipmap and DXT encode | 0.21 ms

References