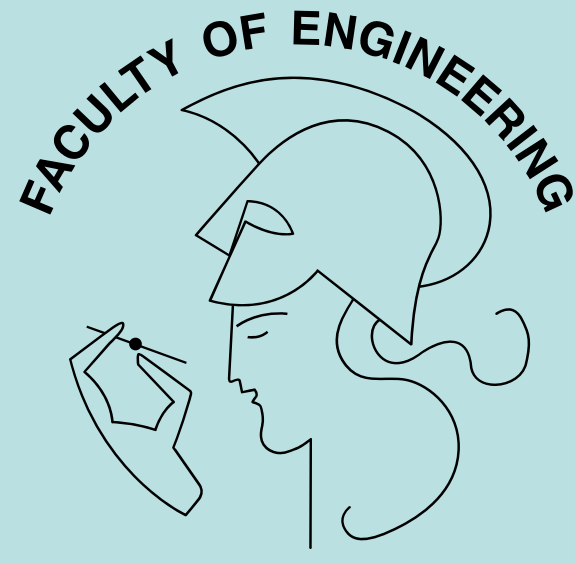


Accelerating Virtual Texturing Using CUDA



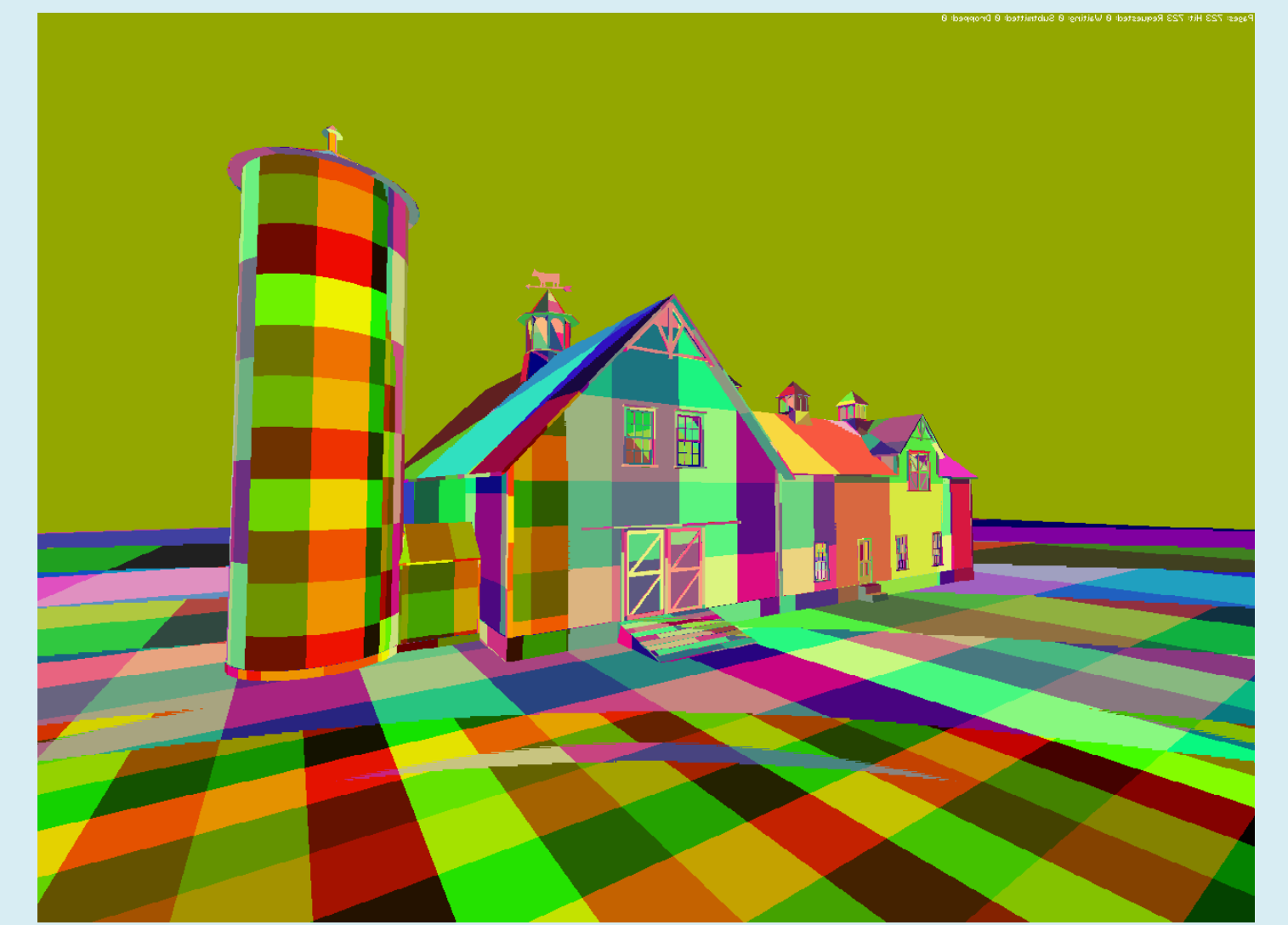
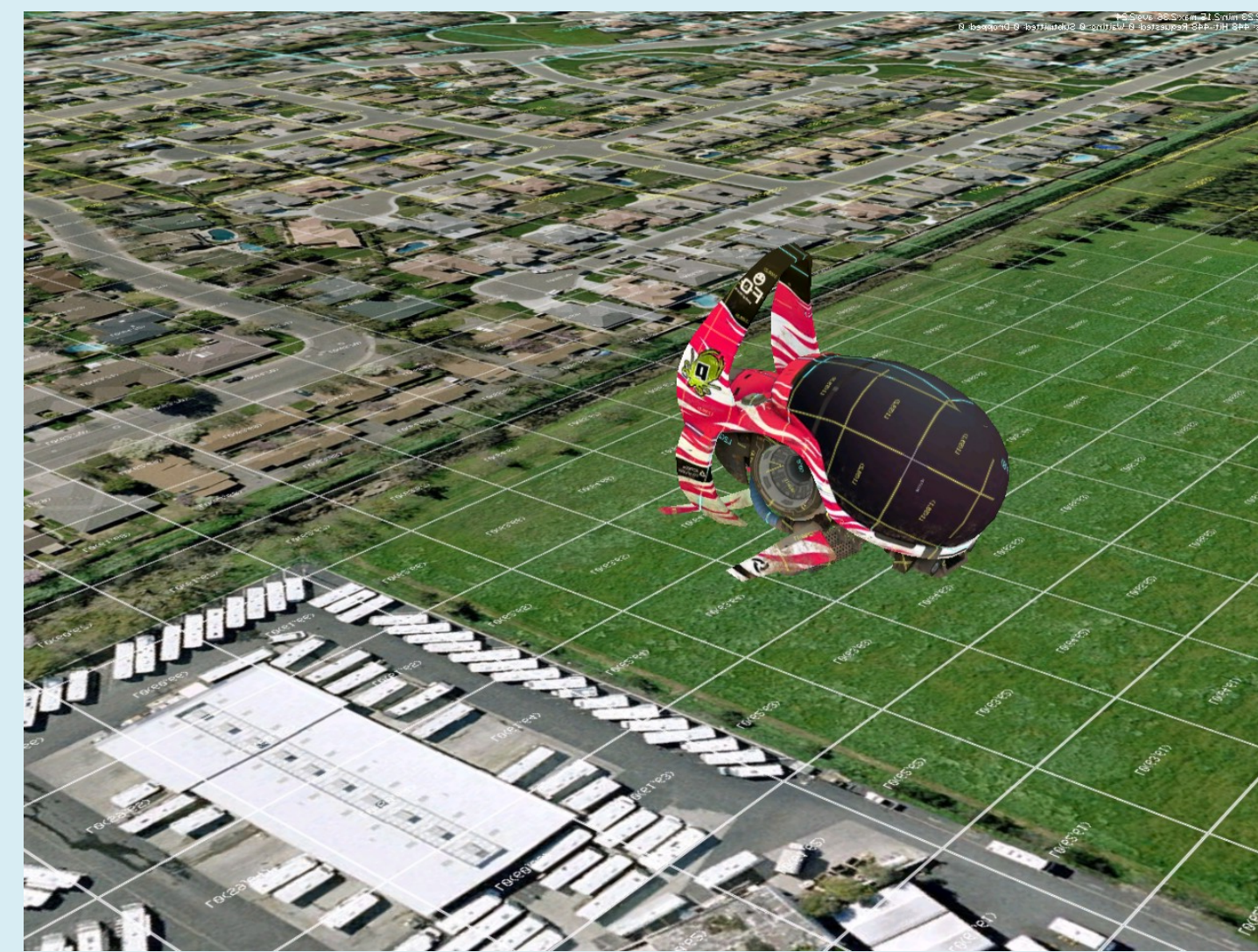
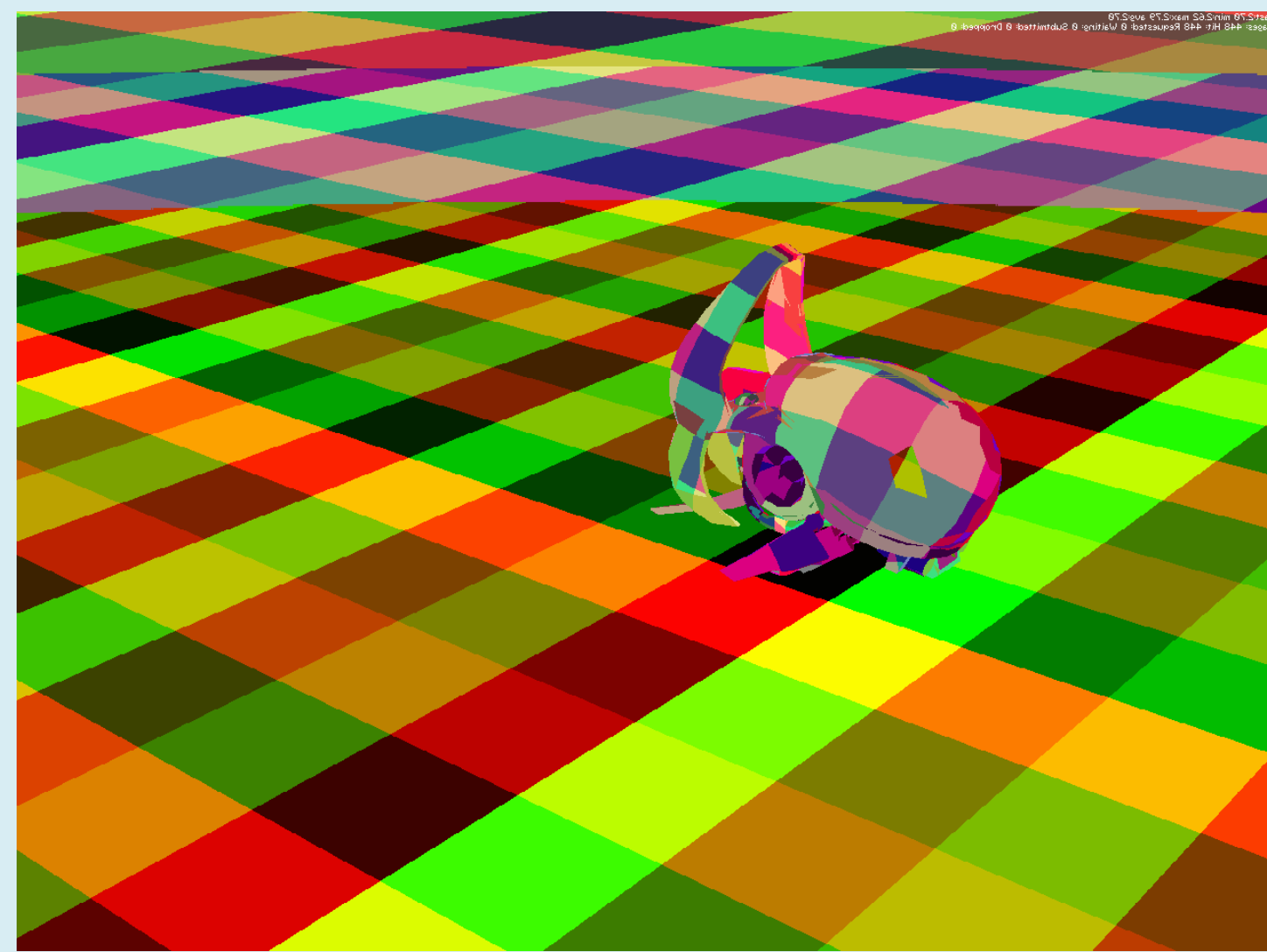
Charles-Frederik Hollemeersch, Bart Pieters,
Peter Lambert, and Rik Van de Walle

Ghent University - IBBT, ELIS Department/Multimedia Lab,
Gaston Crommenlaan 8 bus 201, B-9050 Ledeborg - Ghent, Belgium



e-mail: charlesfrederik.hollemeersch@ugent.be

<http://multimedialab.elis.ugent.be>

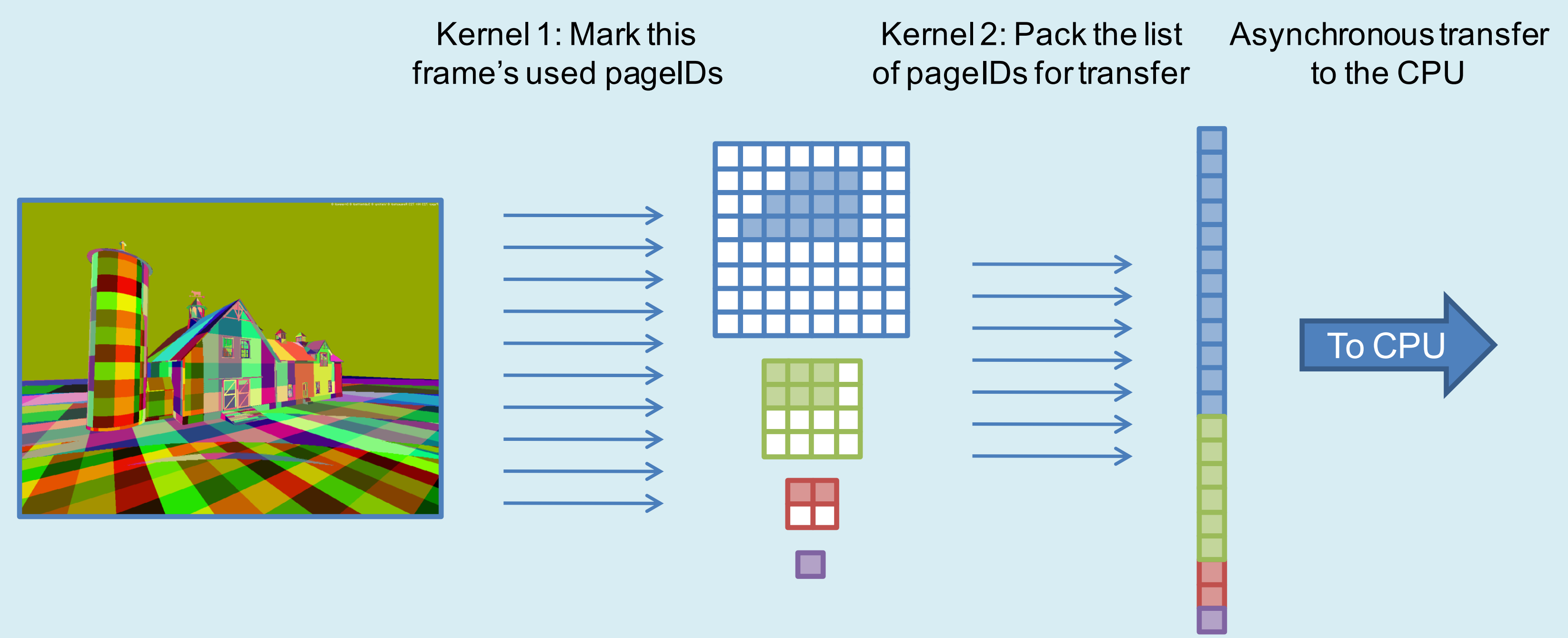


Introduction

- Increasing texture use requires dynamic loading of textures
- Traditionally ad-hoc methods are used
 - Mipmap streaming
 - Clipping (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

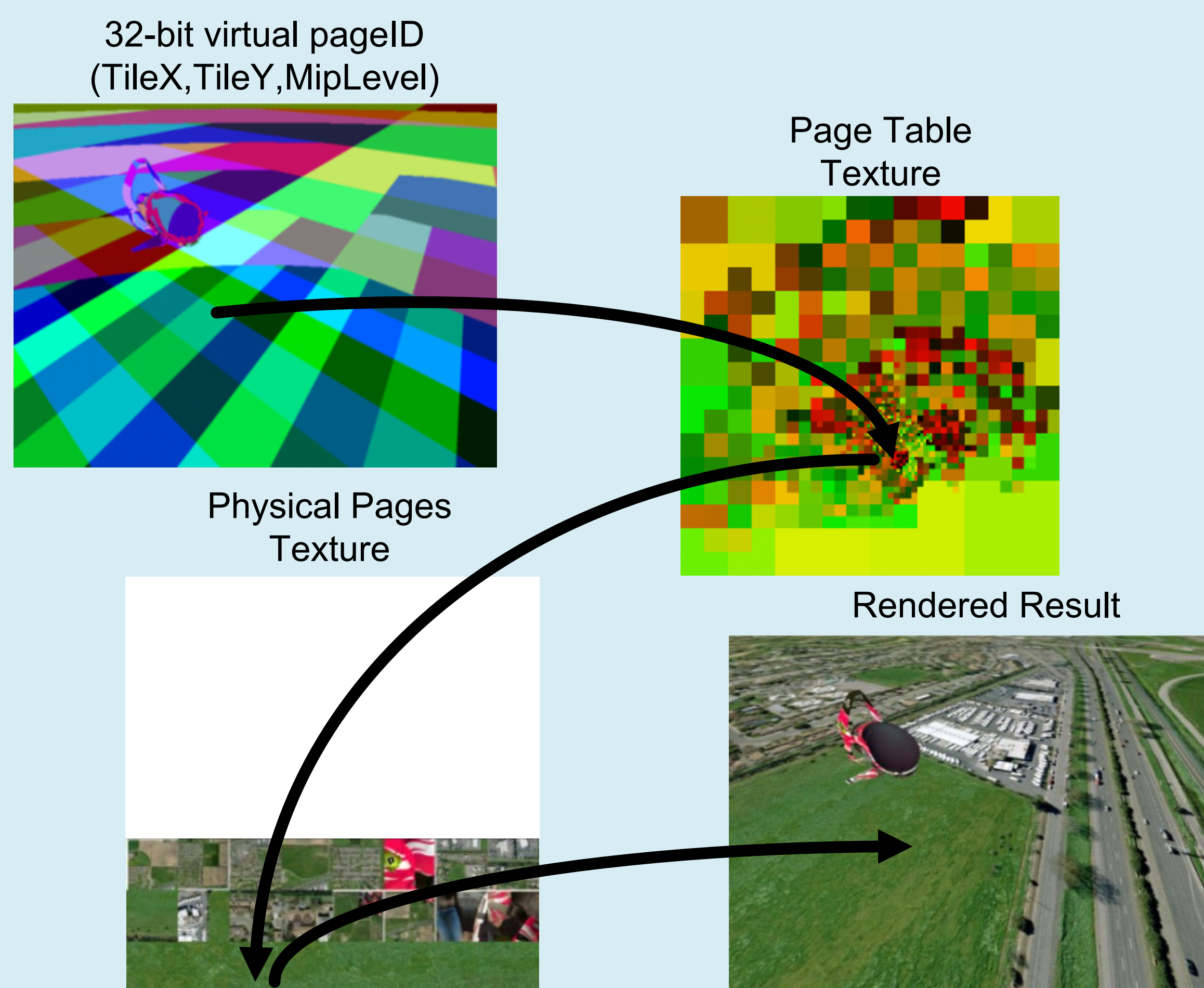
Determining the Working Set using CUDA

- Determines the list of pages needed by the current view
- Analyses buffer rendered using special shader (OpenGL)



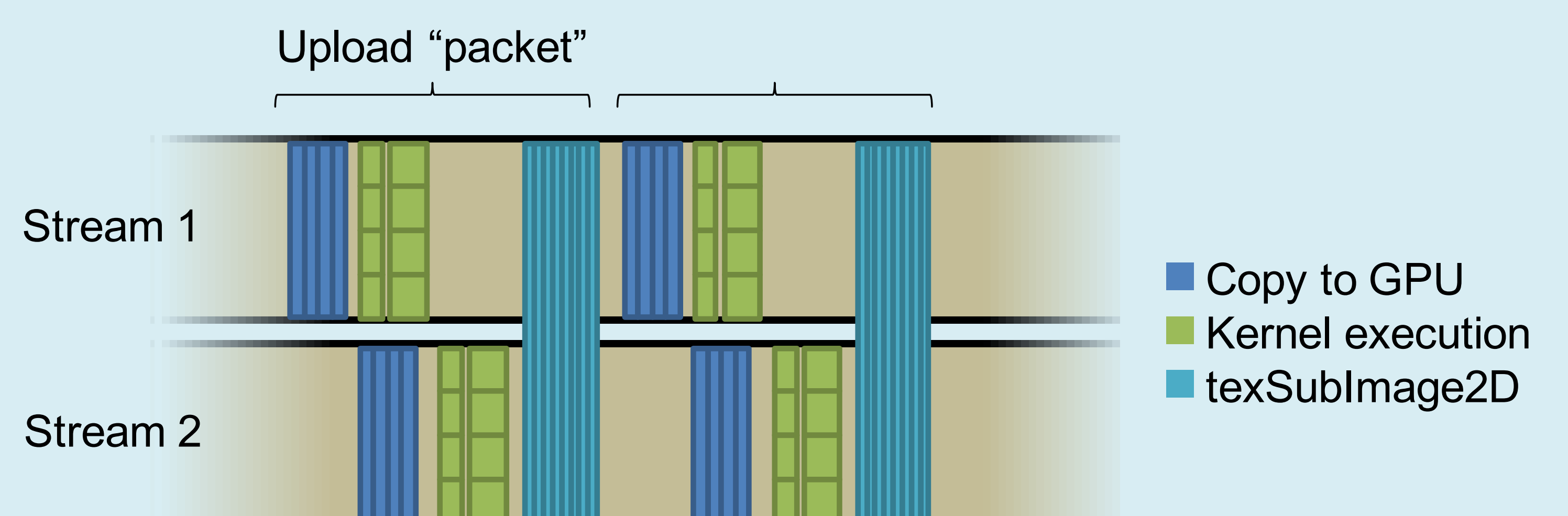
Virtual Texturing: an Overview

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

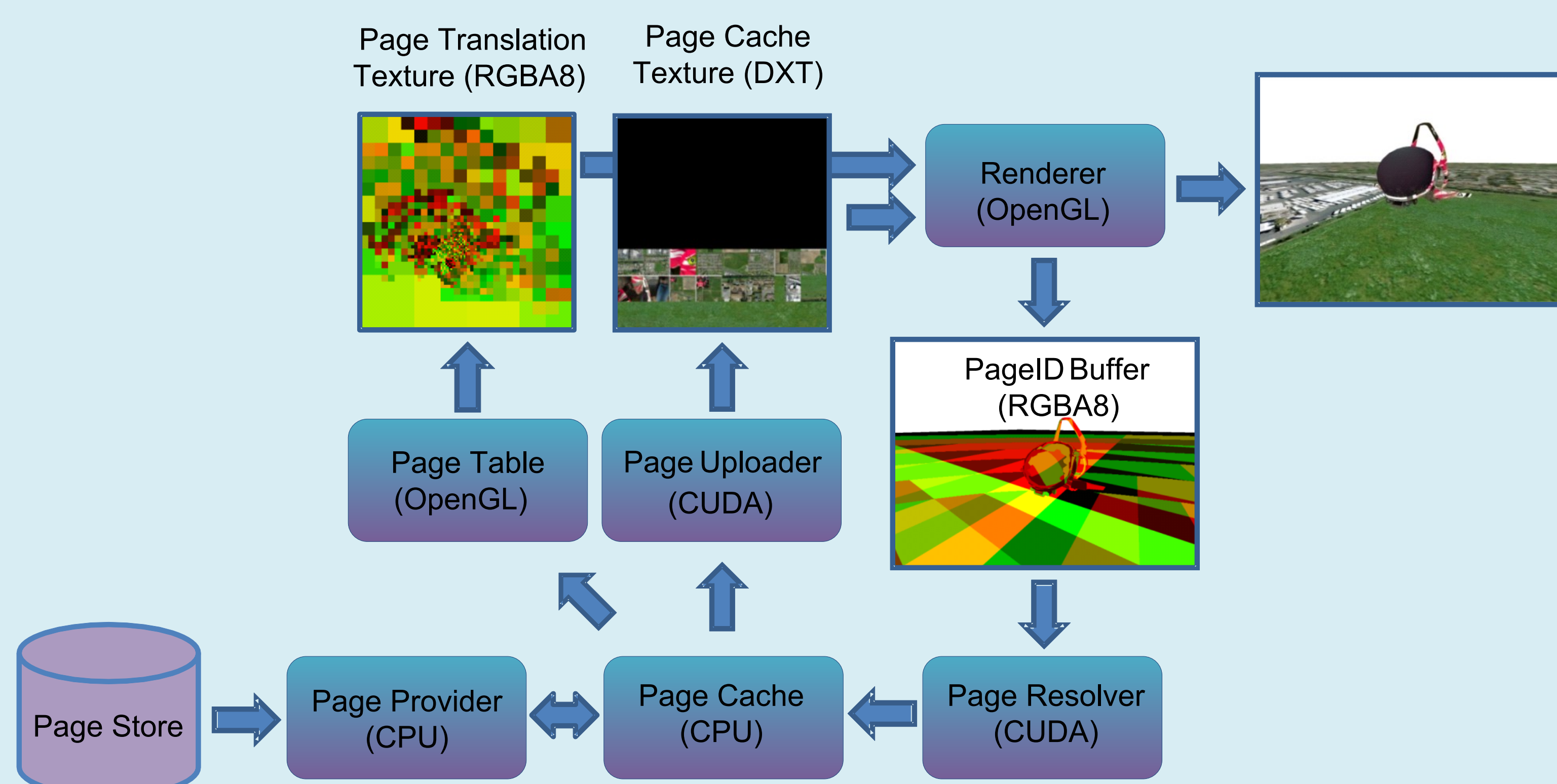


Tile Uploading & Processing using CUDA

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



Software Architecture



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

- Charles-Frederik Hollemeersch, Bart Pieters, Peter Lambert, and Rik Van de Walle. (2009). Accelerating virtual texturing using CUDA. In: Wolfgang Engel *GPU Pro - Advanced Rendering Techniques*. A K Peters.