Virtual Textures

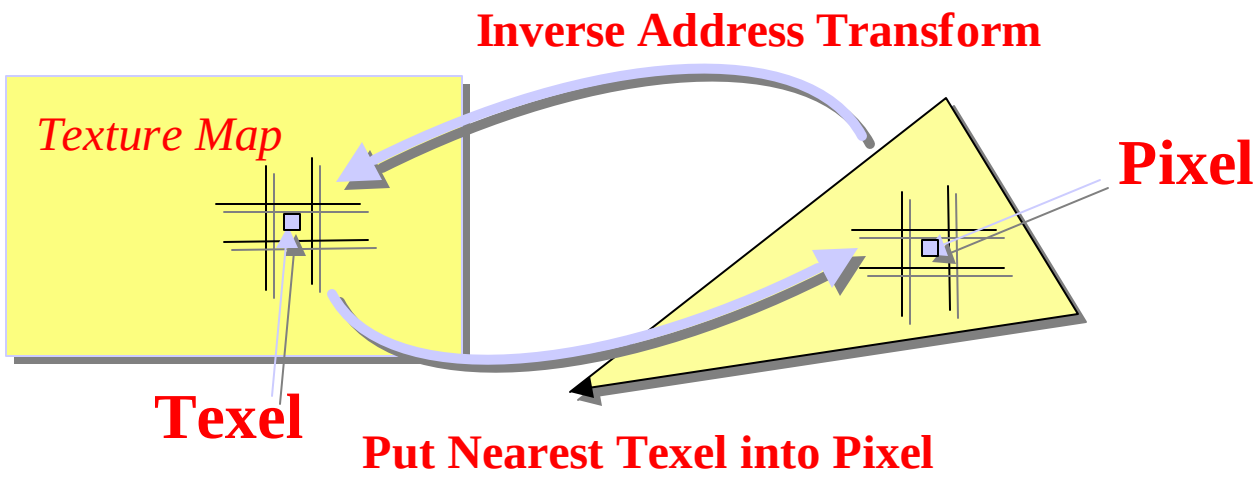
Texture Management in Silicon

Chris Hall
Director, Product Marketing
3Dlabs Inc.

A Few Definitions

Texture Mapping

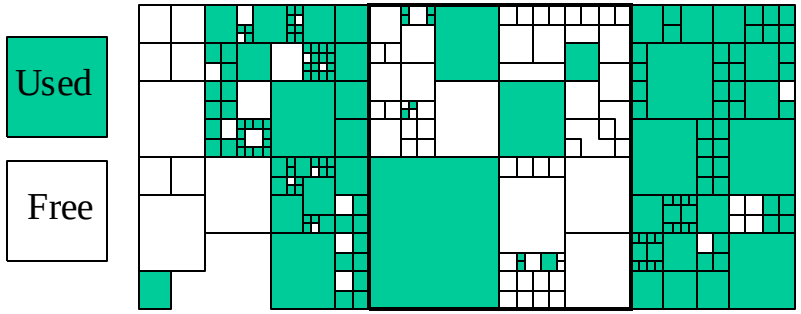
- **Texture Map**
  - Bitmap images used to provide extra detail in rendered images
- **A Mipmap Texture**
  - A series of bitmaps containing the same image at different sizes
- **A Texel is a single pixel in a texture**



Texture Management

A non-trivial problem

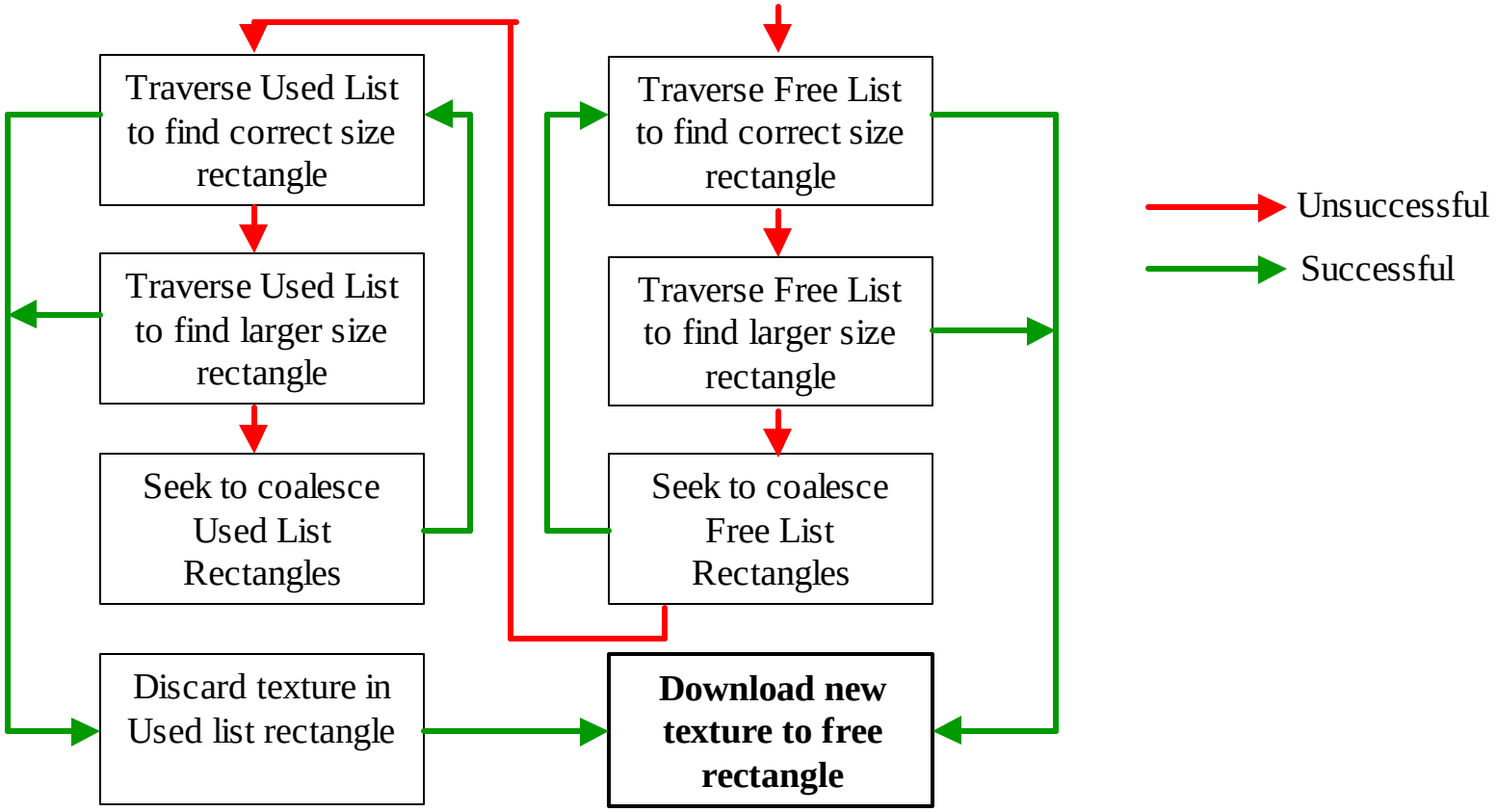
- **Fitting textures into memory is a classical NP complete CS problem**
  - No easy solution
- **Textures use large amounts of memory**
  - A 256x256 true-color texture uses 384KBytes of memory
  - Applications tend to use many (100s) textures simultaneously
- **Textures must be resident on card for maximum performance**
  - Up to 8 texels accessed per drawn pixel
  - ~4GB/s of memory bandwidth to read texel data
- **Onboard texture memory is limited, and must be managed**
  - Bitmaps are 2D entities, and must be fitted inside of a 2D texture store
  - Typically, the textures are square, and sized as a power of 2
- **Bandwidth between the texture store and main memory is limited**
  - AGP 2X is 512MB/s
  - AGP 4X is ~1GB/s



Traditional Solution

Software Management

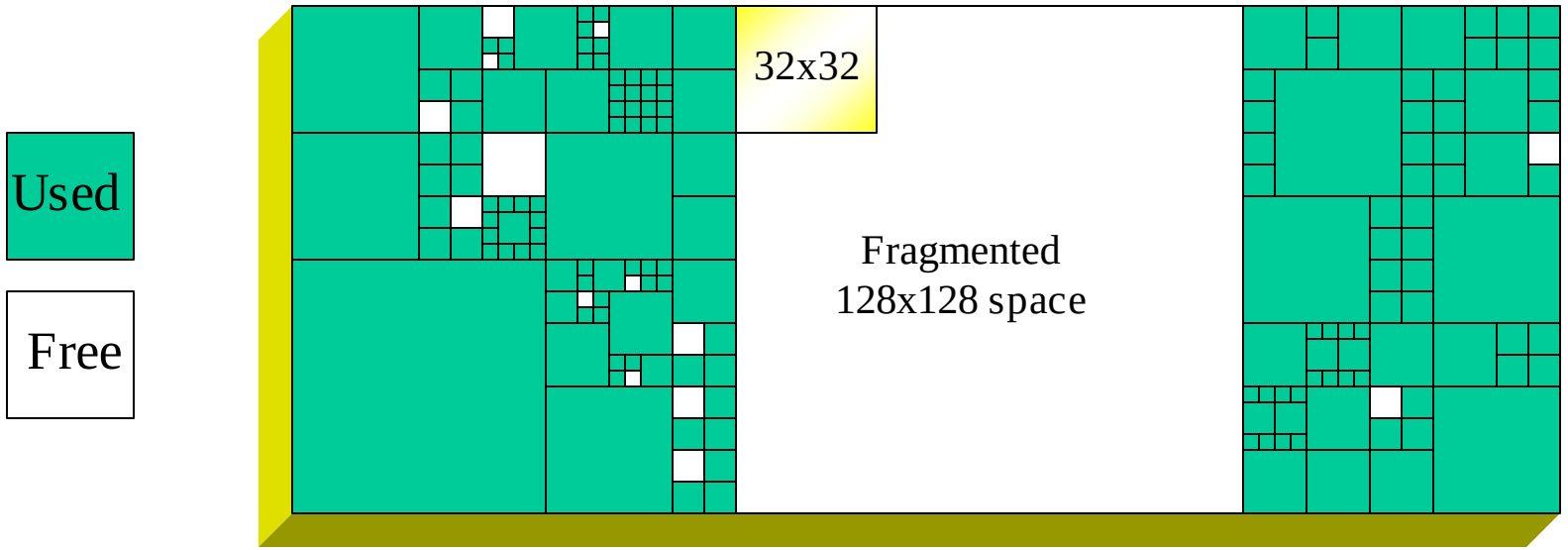
- **Maintain two size-sorted lists of rectangular areas in texture memory**
  - Free List, the "empty" rectangles in texture memory
  - Used List, the " full" rectangles which currently contain textures
- **Complex sequence to download a new texture**



SW Texture Management

Inevitable Problems

- **Wasted Space**
  - We have a 32x32 texture, but the only space available is 128x128
  - Memory Fragmentation becomes a big problem
- **Garbage Collection**
  - Should we coalesce blocks that are freed?
  - Is there an optimal sized block that we should keep uncoalesced?
- **Texture Memory Thrashing**
  - Fitting in new textures of means throwing textures out of the on-board memory that are immediately required again

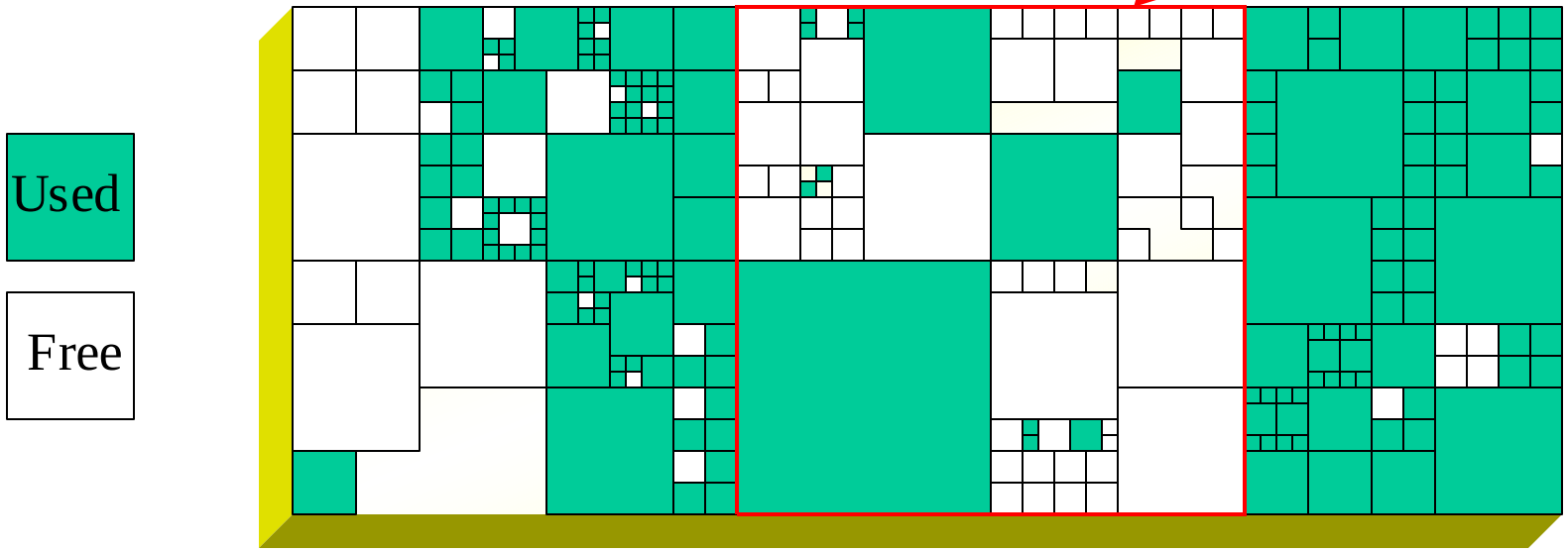


SW Texture Management

More Problems

- **Texture memory management is a pan-application problem**
  - must be handled by part of the driver that has access to all applications
- **Must download complete textures**
  - Includes all mipmap levels
  - Software has no way to know which texels will be needed

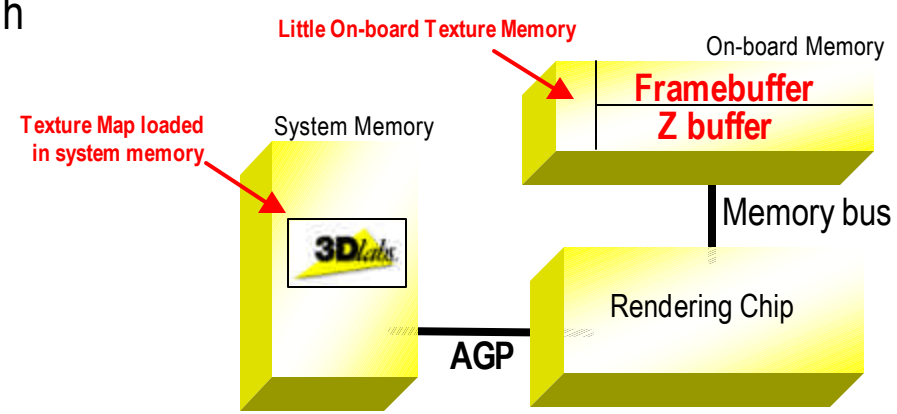
To fit this texture - a lot of shuffling and discarding is needed



An Alternative

AGP Texture Execute

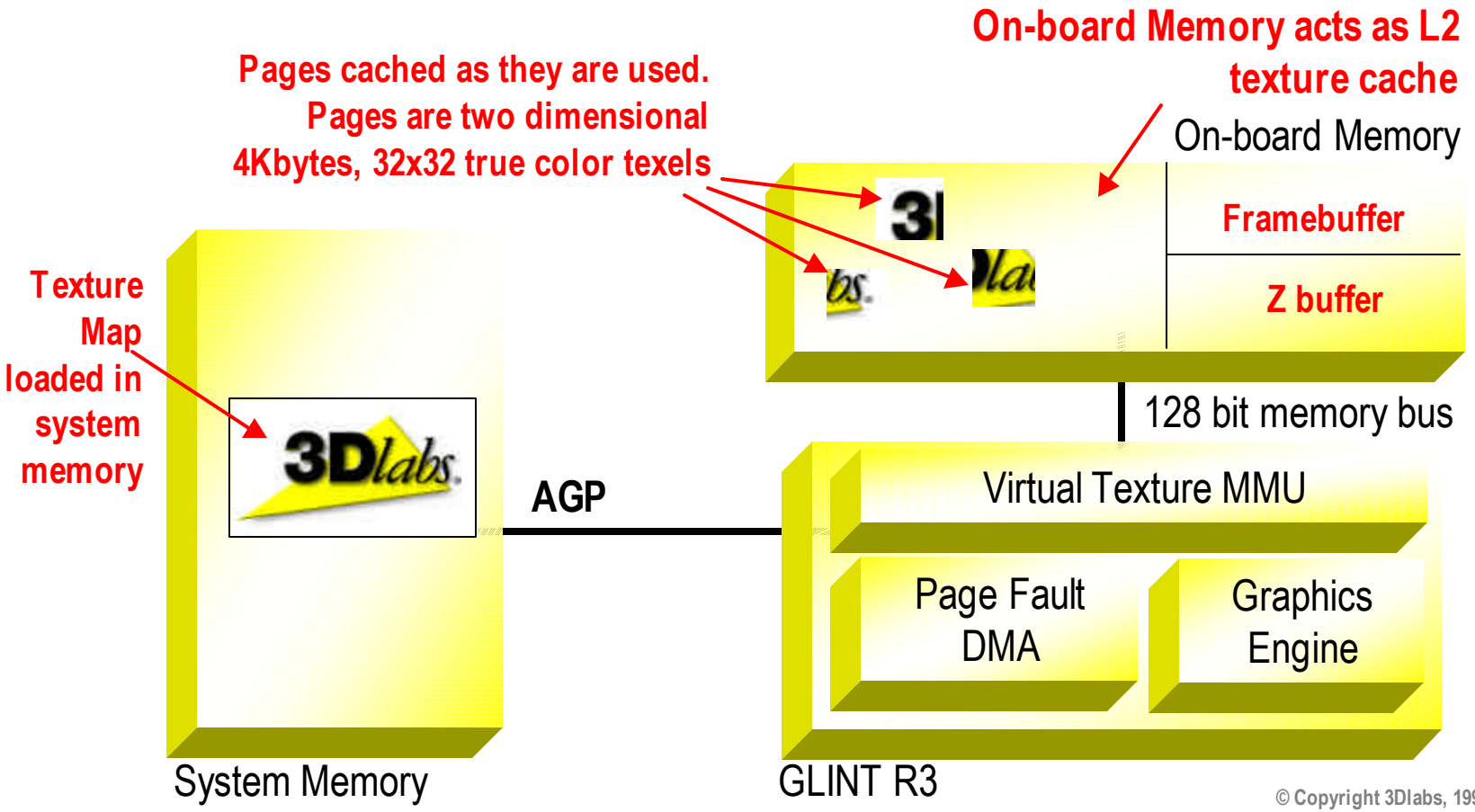
- **Avoid the texture memory management issue all together**
  - The host system has “infinite” memory available
- **Poor texture access latency**
  - Texturing pipeline must stall while the texel data is brought across the bus
- **Generates AGP bus traffic each time a texel is used**
  - On-chip caches help, but not enough
- **Texture traffic clashes with vertex traffic**
  - In a typical graphics system, the AGP bus is still being used to transfer vertex data, while texturing is occuring
- **An ineffective solution**
  - Performance cost is too high



Virtual Textures

3Dlabs' unique texture management system

- **On-chip virtual memory management unit - similar to a CPU**
  - Virtual to physical address translation unit
  - Dedicated page-fault DMA engine fetches pages with no CPU intervention
  - Handles 256MB Virtual Texture address space



Details of the Paging Mechanism

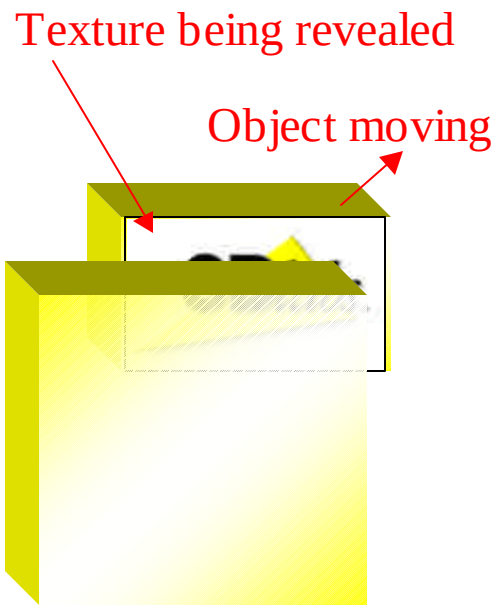
Very much like a CPU Virtual Memory System

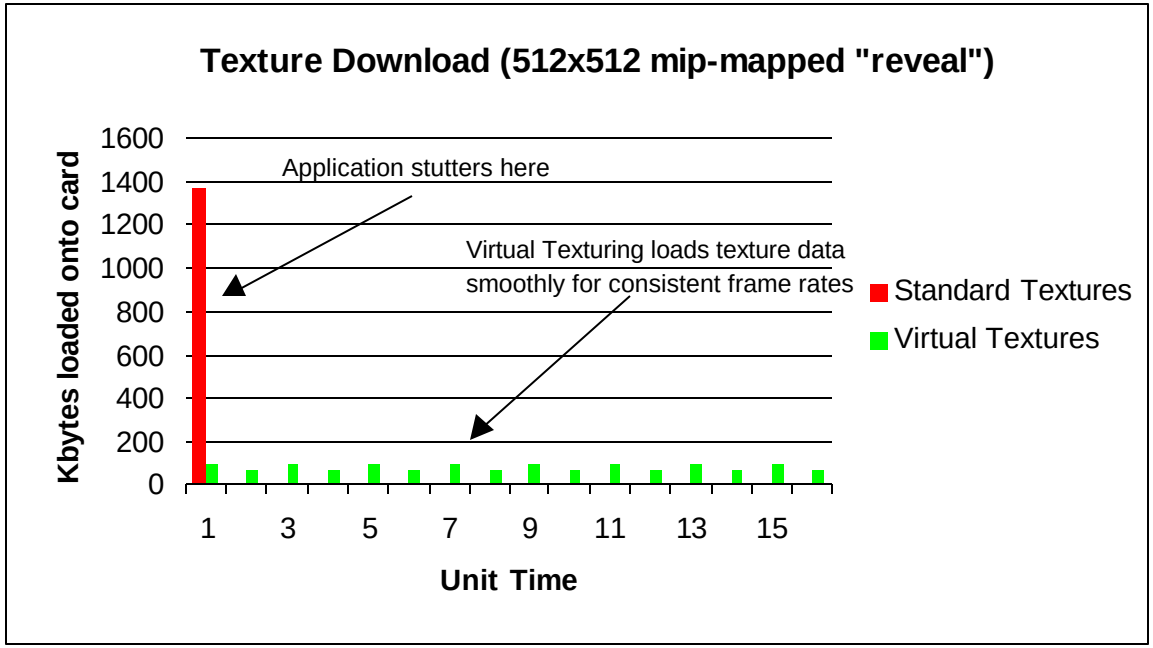
- **Operates on 4KB pages**
  - 32x32 true-color pixels
  - Smaller textures can be combined into a single page
  - True size of data in page is used to determine transfer size on AGP bus
- **Manages up to 256MB of texture data**
  - 8 bytes per page of page table overhead
  - 256MB uses 512KB of onboard memory
- **True Demand Paged Texture Management**
  - Textures do not need to be completely resident on the graphics card
  - Only accessed pages are brought down to the graphics card
- **Textures do not need to be physically contiguous**
  - Not in onboard memory
  - Not in system memory
- **No CPU Intervention Required**
  - Autonomous DMA engine automatically loads pages into on-board working set

Virtual Textures

Significant Load Balancing Benefits

- **A normal graphics board has to download a complete texture as soon as the first texel is accessed**
  - For a 2048x2048x32 texture, that is **16MBytes!**
- **With Virtual Texturing, only the accessed pages are downloaded**
  - Avoids large texture download spikes and application "stuttering"



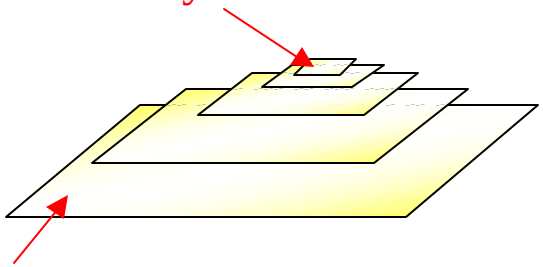


Mip-map Level Loading

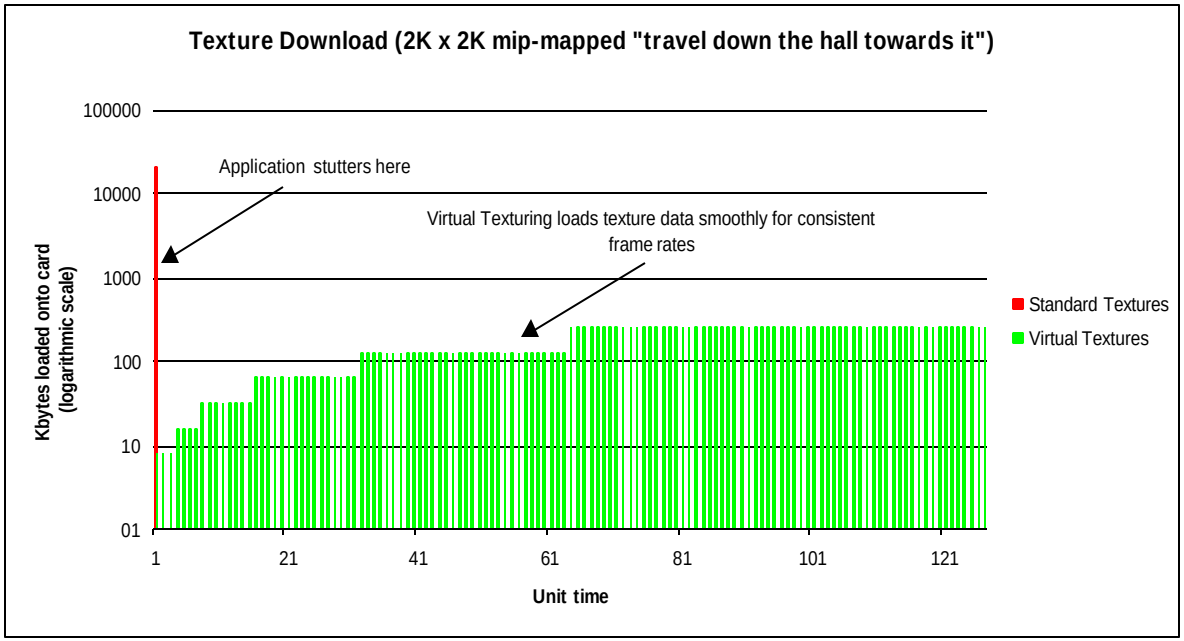
Virtual Textures Score Again

- **Texture loading on demand works for mip-map levels**
  - A far-away object need only load low-resolution mip-map levels
- **If the object never comes close - can result in huge savings**
  - The lowest mip-map levels of even a 2048x2048x32 texture fits in just a few bytes
- **As object gets closer mip-map level pages are smoothly loaded**

Small mip-map level needed when object is far away



Large Texture



Other Virtual Textures Benefits

Texture management software becomes trivial

- **Software simply identifies location of texture in host memory to the graphics sub-system**
- **No CPU Intervention needed on a page miss**
  - Geometry pipeline doesn't need to understand whether or not to load a new texture down to the graphics sub-system
- **Ability to (effectively) use textures that are larger than available memory**
- **Easily handles thousands of small textures**
- **Can be extended to access System Virtual Memory**
  - Interrupt generated upon page fault
  - ISR causes operating system to page in required data
  - ISR causes graphics sub-system to load required data

Conclusion

Virtual Memory for Graphics Cards

- **The Virtual Textures mechanism is a dramatically improved texture memory management technique.**
- **Improved Performance - up to 50% better real world performance over hardware with similar raw fill-rates**
- **Simplified Software - fewer bugs, less CPU time**
- **Improved usage of available texture memory**
  - Entire textures don't have to be present
  - Only used Mipmap levels need to be downloaded
- **Virtual Textures is available today in the Permedia3 Create!, Oxygen VX1, Oxygen GVX1, and the new Oxygen GVX210 products**