A Virtual Memory System for Real-Time Visualization of Multi-Resolution 2D Objects

Sergio Pinheiro
Luiz Velho

Problem

• Real-Time visualization of 2D objects defined by large data-sets.
• Difficulties
  – High-speed memories have small storage capacity.
  – Short time interval to perform rendering process.

Motivation

• 2D objects are widely used in computer graphics applications.

Satellite Image  Virtual Panorama  Terrain Data

A Solution

• Memory management system
  – Virtual memory model.
  – On-demand paging mechanism.
  – Predictive Caching.

✓ Requirements
  • Facilities to add and to remove storage levels.
  • Application-independent.

➢ Based on Multi-resolution representation

Multi-Resolution Representation

➢ Rectangular geometric support.
➢ Regular decomposition.
➢ Discrete multi-resolution representation (2^j).
Multi-Resolution
• Normal Multi-Resolution Method

Multi-Resolution
• Adaptive Multi-Resolution Method

Multi-Resolution
• Adaptive Multi-Resolution Method (Bottom-Up)

Memory Management System
• Virtual Memory System

Memory Management System
• Logical address identifies a tile
Memory Management System

- Application Communication

- Application
- Application Interface

- Visualization System
- Memory Manager

- Geometric access
- Texture access
- Information about View and Object

- I/O System
- Page Loader
- Page Unloading System
- Page Table

- Memory Device
- RAM
- Memory Maneger
- On demand Paging
- Predictive Paging

- Level 0
- Level 1

- Graphical Object
- Texture

- Information about View and Object

Memory Management System

• Predicting Problem
  - Let \((t_0, t_1, ..., t_n) \in \mathbb{R}^n\) and \((p_0, p_1, ..., p_n) \in \mathbb{R}^n\)
  - Define a mapping function \(f\) as \(f(t_i) = p_i\), where \(0 \leq i \leq n\)
  - Calculate the value of the function \(f(t_{i+1})\), where \(k > 0\)

• Predictive Mechanism
  - Calculate velocities: \(v_{n+1} \rightarrow v_n = v_{n-1} + v_{n-2} + ... + v_0\)
  - Calculate acceleration: \(a_n = a_{n-1} + a_{n-2} + ... + a_0\)
  - Define \(f\) as \(f(t_{n+k}) = p_n + v_n + a_n\), where \(k > 0\)

Memory Management System

• Page Loading Rules
  - Rule 1: Pages that contain textures of the lowest resolution cannot be liberated.
  - Rule 2: Hierarchical dependence.
  - Rule 3: Pages that contain textures of the current frame cannot be liberated.

• Page Unloading Rules
  - Rule 1: Pages that contain textures of the lowest resolution cannot be liberated.
  - Rule 2: Hierarchical dependence.
  - Rule 3: Pages that contain textures of the current frame cannot be liberated.

• Page Unloading Criterion
  - Based on resolution level and distance.
**Application**

- Virtual Panorama

**Configuration**

- **Hardware**
  - Pentium III-850
  - 768 MB of RAM
  - 40 GB SCSI – 80 MB/s
  - Oxygen GVX-420 graphics card
- **Software**
  - Windows NT
  - C++
  - Open GL

**Conclusion**

- The system Guarantees a frame rate of 30 fps.
- The system can be extended for other applications.
- It's easy to add others storage devices.
- It's easy to use others predictive algorithms.
- It's easy to modify the loading and replacement system.
- The system doesn't need a powerful machine.

**Future Work**

- Extend the management system to work with geometric data.
- Make possible to management animated texture.
- Modify the loading and replacement system to allow terrain data real-time visualization.
- Incorporate the storage network level.
- Extend the system to do memory management for others object graphic operations.