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

Sergio Pinheiro
Luiz Velho

Problem

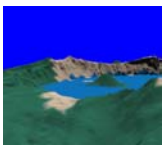
- Real-Time visualization of 2D objects defined by large data-sets.
- Difficulties
  - High-speed memories have small storage capacity.
  - Short time interval to perform rendering process.

Motivation

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

Satelite Image | Virtual Panorama | Terrain Data





A Solution

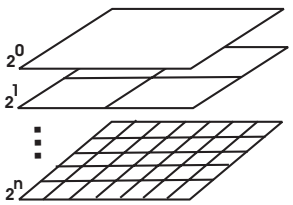
- Memory management system
  - Virtual memory model.
  - On-demand paging mechanism.
  - Predictive Caching.

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

➢Based on Multi-resolution representation

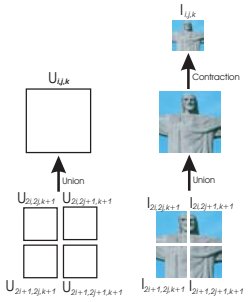
Multi-Resolution Representation

- Rectangular geometric support.
- Regular decomposition.
- Discrete multi-resolution representation (2^j).



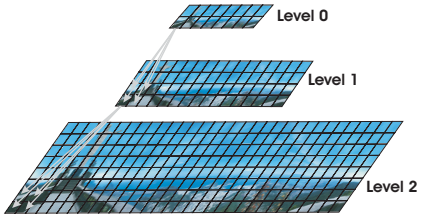
Multi-Resolution

- Normal Multi-Resolution Method (Bottom-Up)



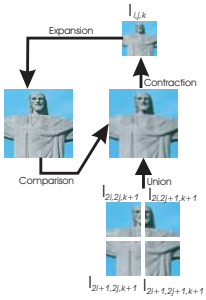
Multi-Resolution

- Normal Multi-Resolution Method



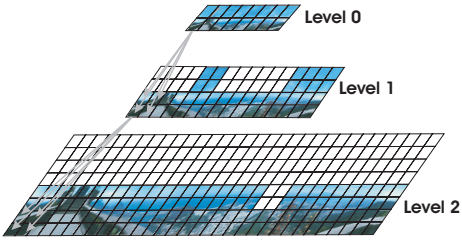
Multi-Resolution

- Adaptive Multi-Resolution Method (Bottom-Up)



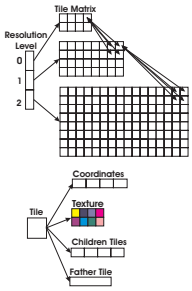
Multi-Resolution

- Adaptive Multi-Resolution Method



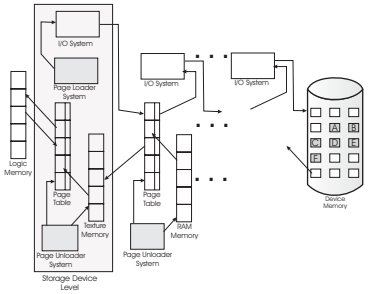
Multi-Resolution

- Data Structures



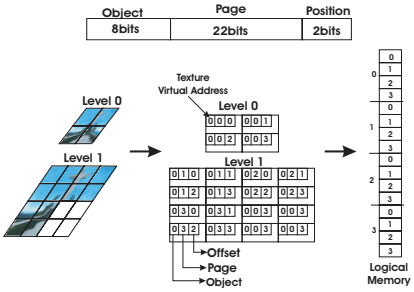
Memory Management System

- Virtual Memory System



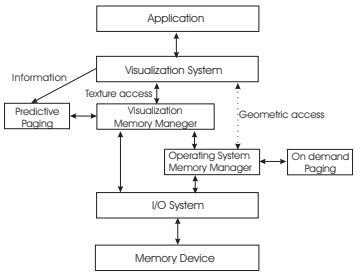
Memory Management System

- Logical address identifies a tile



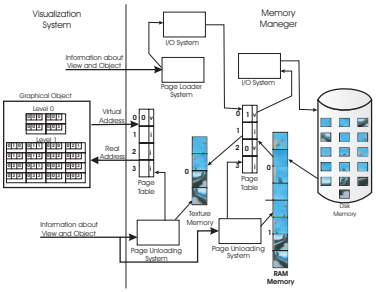
Memory Management System

- Application Communication



Memory Management System

- Application Interface

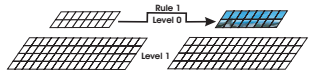


Memory Management System

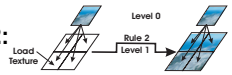
- Predicting Problem
  - Let $(t_0, t_1, t_2, ..., t_n) \in \Re$ and $(p_0, p_1, p_2, ..., p_n) \in \Re^m$
  - Define a mapping function *f* as $f(t_i) = p_i$, where $0 \le i \le n$
  - Calculate the value of the function $f(t_{n+k})$, where $k > 0$
- Predictive Mechanism
  - Calculate velocities: $v_{n-1} = \frac{p_{n-2} - p_{n-1}}{t_{n-2} - t_{n-1}}, v_n = \frac{p_n - p_{n-1}}{t_n - t_{n-1}}$
  - Calculate acceleration: $a_{n-1} = \frac{v_n - v_{n-1}}{t_n - t_{n-1}}$
  - Define *f* as $f(t_{n+k}) = p_n + v_n t_{n+k} + \frac{a_n t_{n+k}^2}{2} = p_{n+k}$, where $k > 0$

Memory Management System

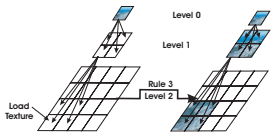
- Page Loading Rules
  - **Rule 1:**



  - **Rule 2:**

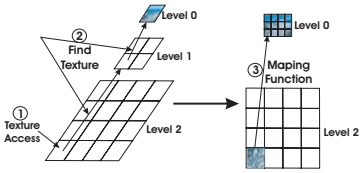


  - **Rule 3:**



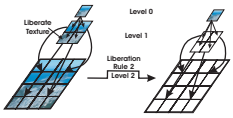
Memory Management System

- Page Fault



Memory Management System

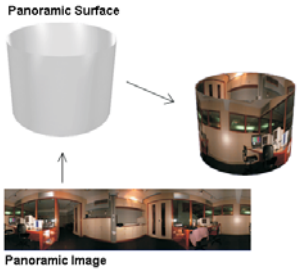
- Page Unloading Rules
  - **Rule 1:** Pages that contain textures of the lowest resolution cannot be liberated.
  - **Rule 2**:Hierarchical dependence.



  - **Rule 3:** Pages that contain textures of the current frame cannot be liberated.
- Page Unloading Criterion
  - Based on resolution level and distance.

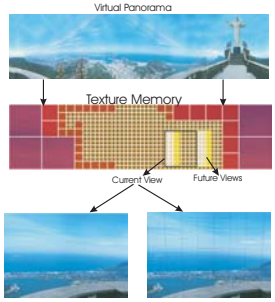
Application

- Virtual Panorama



Application

- Real-Time Visualization of Virtual Panorama



Configuration

- Hardware
  - Pentium III-850
  - 786 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.