

Spatial Database Techniques Oriented to Visualization in 3D GIS

Author: Jun Li, Ning Jing, Maoyin Sun

**Reporter : Jun Li
Junli@nudt.edu.cn**

**School of Electronic Science and Engineering,
National University of Defense Technology,
Changsha, Hunan, 410073, China**



Feature of 3D GIS

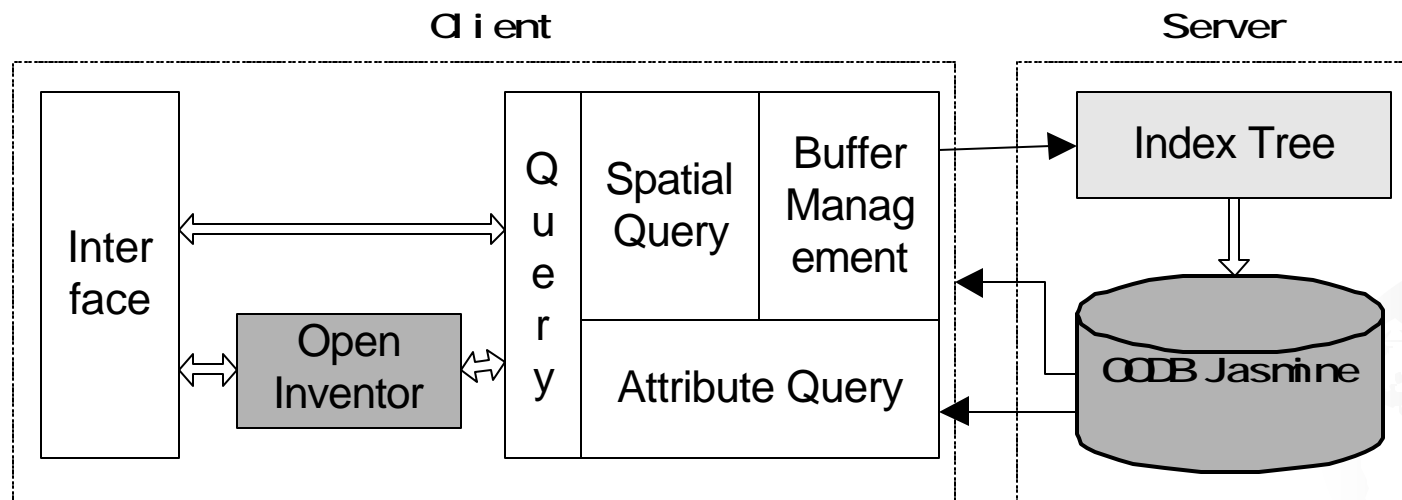
- ◆ All data should be integrated in spatial database
- ◆ Integrated query interface
- ◆ 3D visualization with fast rendering and high performance
- ◆ Better human-computer interactive



Architecture of a 3D GIS Experiment System

◆ Client / Server structure

- Server: spatial database
- Client: client program

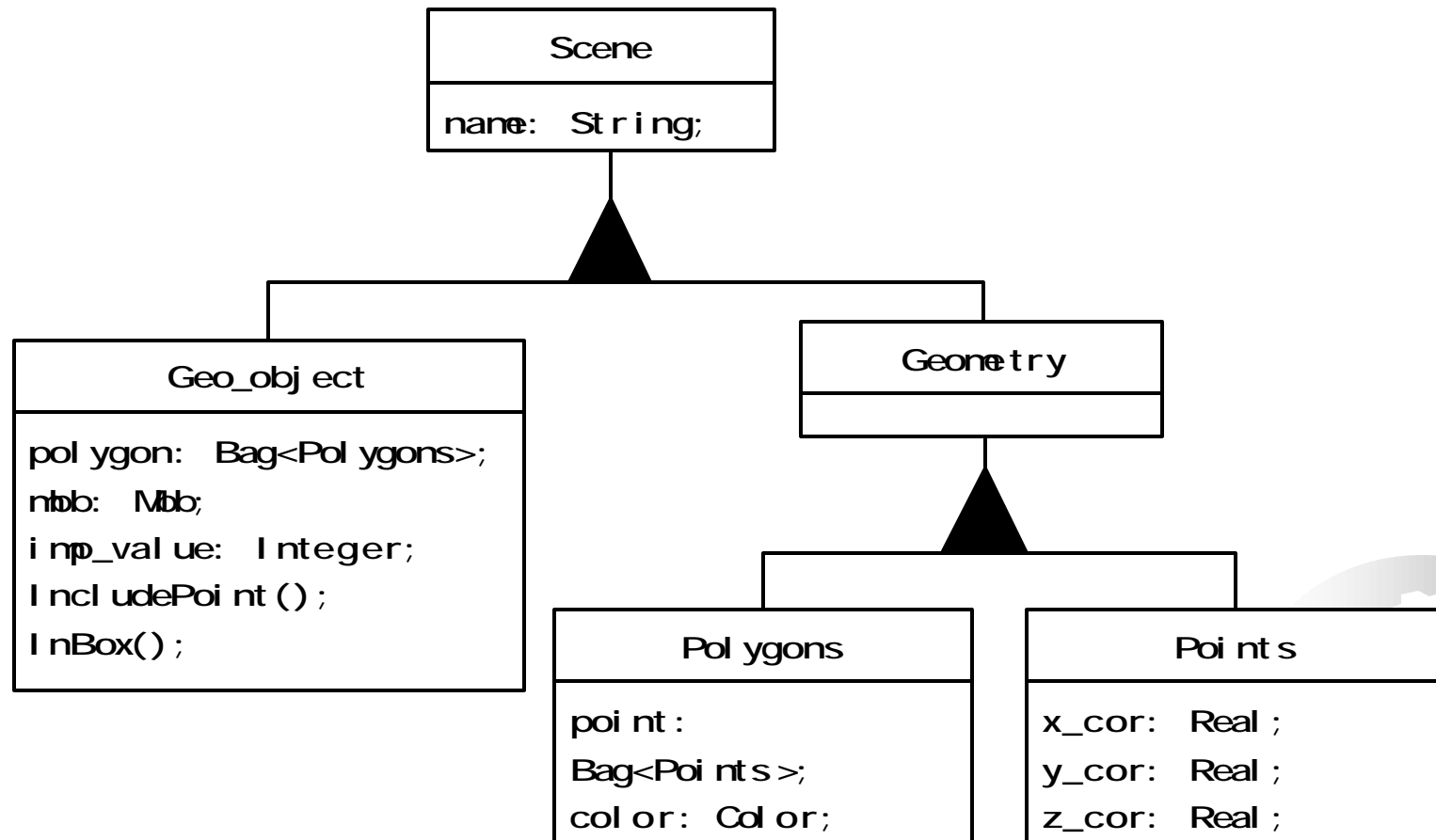


Object-Oriented Database

- ◆ **Limitations for RDBMS in GIS application**
 - Data type can.t be user-defined
 - Spatial operation can.t be user-defined
 - Spatial index can.t be user-defined
 - Spatial query can.t be supported by SQL
- ◆ **Current spatial database**
 - RDBMS and ORDBMS
- ◆ **OODBMS**
 - Jasmine



Object-Oriented Data Structure



Building Spatial Index

◆ Objective

- Based on spatial index, Lod and generalization at multi-scale can be integrated

◆ 3D Visualization with Lod

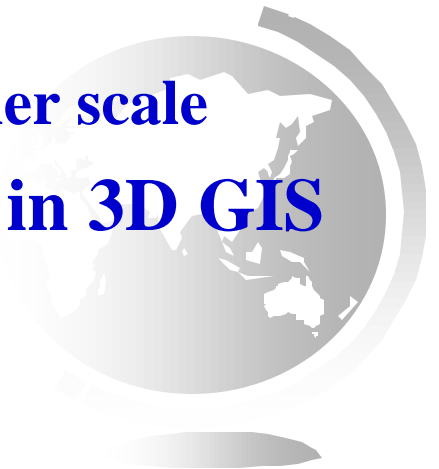
- Store all data of different Lods in advance
- Organize data hierarchically by spatial position

◆ Generalization at multi-scale

- Data at small scale come from data at the higher scale

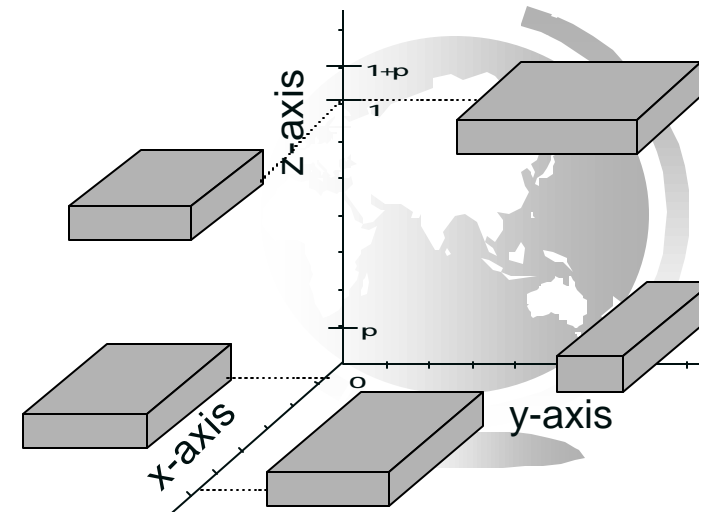
◆ Difficulties for visualization at multi-scale in 3D GIS

- Database management
- 3D visualization



Structure of V-Reactive tree

- ◆ **Lod and generalization rely on the same index tree**
- ◆ **Alternative Reactive Tree**
 - Based on R tree
 - Add a dimension of importance value
 - Selection of parameter p
- ◆ **Concept of V-Reactive tree**
 - 4D R tree structure
 - New spatial query algorithm
 - Optimized for 3D visualization



Principle of Implementing Visualization at Multi-scale

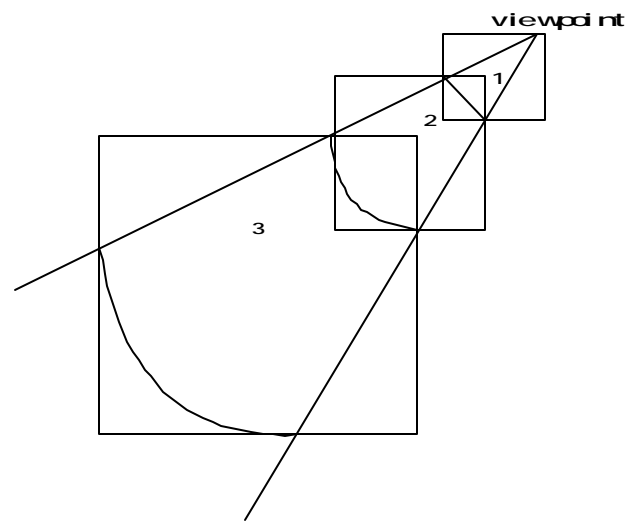
- ◆ **Generalization on V-Reactive tree at different scales**
 - Selection can be implemented by query with importance value
- ◆ **At a certain scale, use the hierarchical structure for Lod**
 - The lowest leaf node represent the greatest details
 - The nodes pointing to leaf node represent less details by MBB
 - The upper node represent the least details by bigger MBB



Perspective query

◆ Concept of perspective query

- By use of LOD, we don't need to visit the nodes from root to the leaf for each spatial query
- By distance from viewpoint, we can divide the visible pyramid into several regions corresponding to LODs. For each region, objects lying in the region are queried on V-Reactive tree from root downwards to the corresponding level.



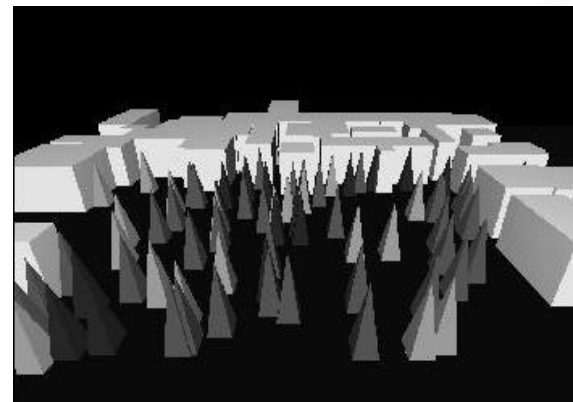
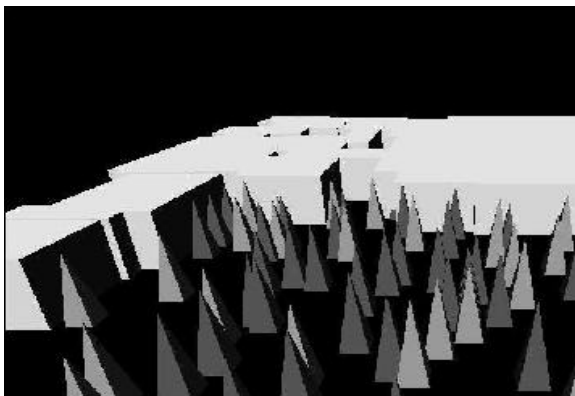
Optimization of V-Reactive Tree

◆ Inherent problems in multi-dimensional Space

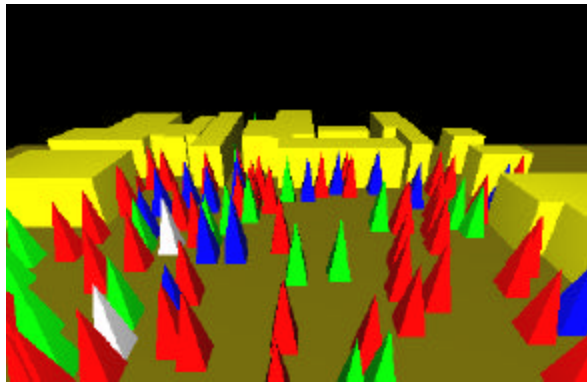
- Added dimension of importance cause that near objects can.t be grouped together in spatial index tree
- Greatly increased overlay among MBB of spatial objects result in overlay at Lod2 in 3D scene

◆ Different choose-path algorithm are used for different level

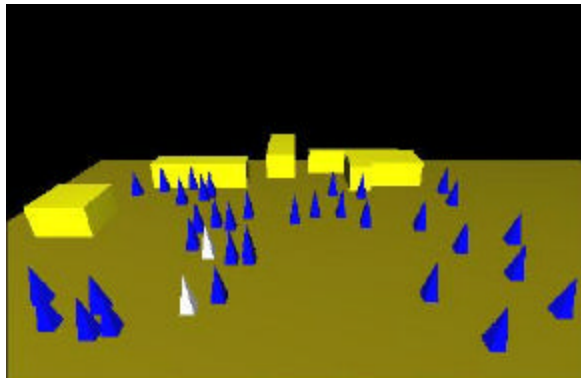
- K represent level at Lod2
- For levels greater than K+1, 3D least volume enlargement is used
- For level equal to K+1, 3D least overlay enlargement is used
- For levels less than K+1, 4D least overlay enlargement is used



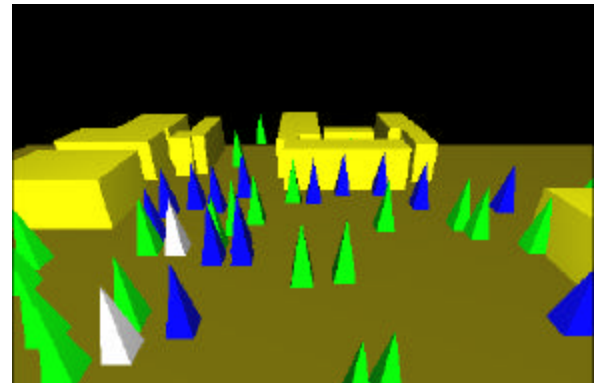
Experiment Results



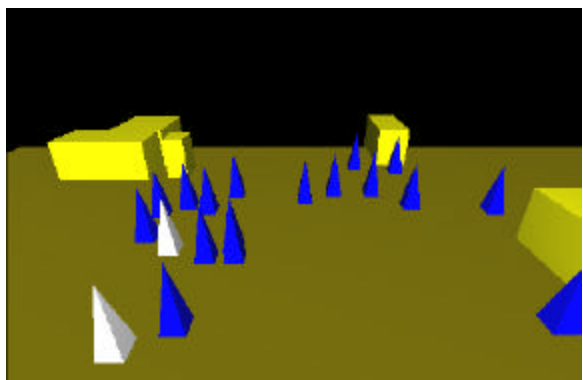
a



b



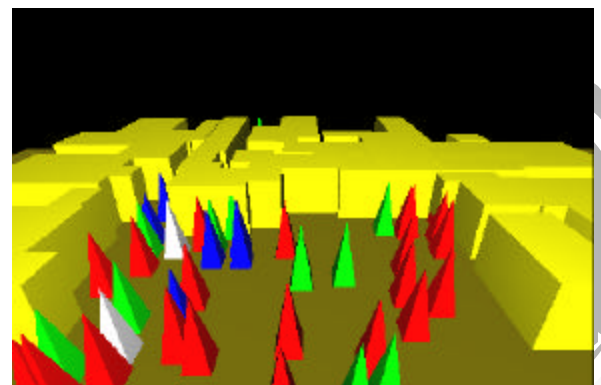
c



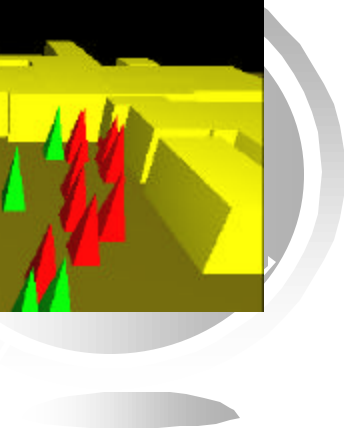
d



e



f



Extension of index tree

- ◆ **Extend V-Reactive index tree in OODB Jasmine**
- ◆ **Two mechanisms**
 - **Building index class**
 - ◆ **Building V-Reactive class in OODB including attributes and methods for operation of index tree**
 - **Building index file**
 - ◆ **Building separate index file by OID of spatial objects stored in OODB**
 - ◆ **Operation of index tree is embeded in applications**



Dynamic Buffer Management

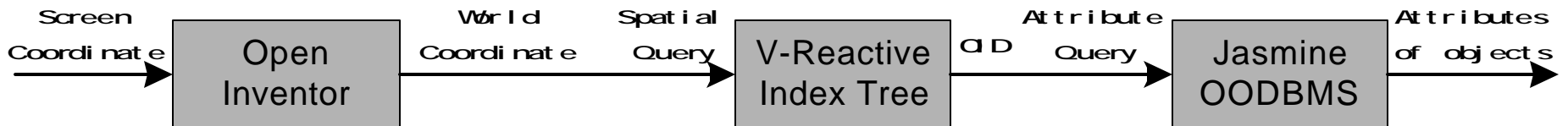
- ◆ Requirement of buffer management in 3D GIS
 - Dynamically load data within visual region
 - Take spatial object as page in buffer
 - ◆ size of spatial objects are different
 - ◆ in a visual region, only different objects are accessed
- ◆ LRU algorithm is optimized for better performance
 - Estimate size of buffer space in the process of data exchangement
 - Set counter for each page in buffer, preferentially hold pages with greater value



Other Relative Techniques

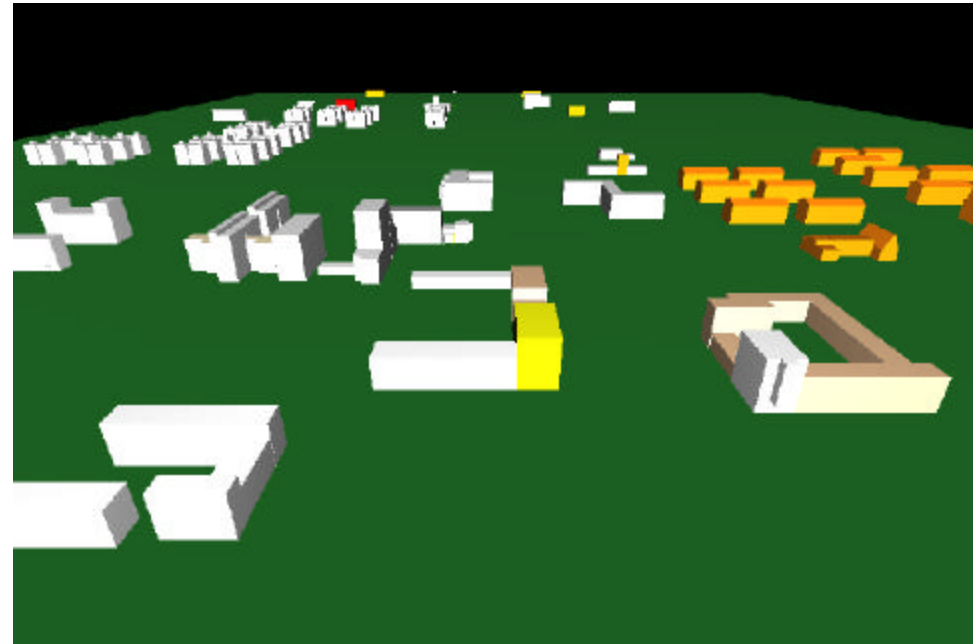
◆ Integrated query interface

- Attribute query is implemented directly by ODQL
- Spatial query is implemented by methods executed at server



◆ Interaction on 3D scene

- Visualization tools
 - ◆ Open Inventor
- Spatial query



Conclusion and Future Work

◆ Features

- Managed by OODB
- Dynamically load data only within visual region
- Implement spatial query on the basis of spatial index
- Integrate Lod with generalization techniques
- Spatial Query can be executed by interaction with 3D scene

◆ Future Work

- Performance should be evaluated for application with mass data
- Data model should be optimized

