AYIA NAPA SUMMER SEMINAR 2006

Modeling and streaming large terrain database for network-based interactive visualization

Christian Bouville (http://perso.rd.francetelecom.fr/bouville/)
Patrick Gioia, Jérôme Royan
France Telecom R&D

(Unrestricted)

Network-based 3D terrain visualization

- Many applications :
  - Locations based services
  - Tourism : information, vacation rentals
  - Cultural heritage
  - Real estate : real estate projects
  - Civil engineering, impact simulation
  - Games
  - …
- Leverages low-cost 3D hardware and high-speed internet

State-of-the-art Platforms and Standards

- Commercial platforms:
  - GoogleEarth (Keyhole), Skylinesoft, Geonova,…
- Open source:
  - Terravision: http://www.ai.sri.com/TerraVision/
- Standards:
  - X3D GeoSpatial (GeoVRML):
    - Set of dedicated nodes (geo. Coordinates, QuadLOD,…)
  - MPEG-3DGC: http://www.mpeg-3dgc.org/
    - Adaptive streaming solutions for:
      - *Terrain: Wavelet Subdivision Surfaces*
      - *Urban environments: PBTree technique*

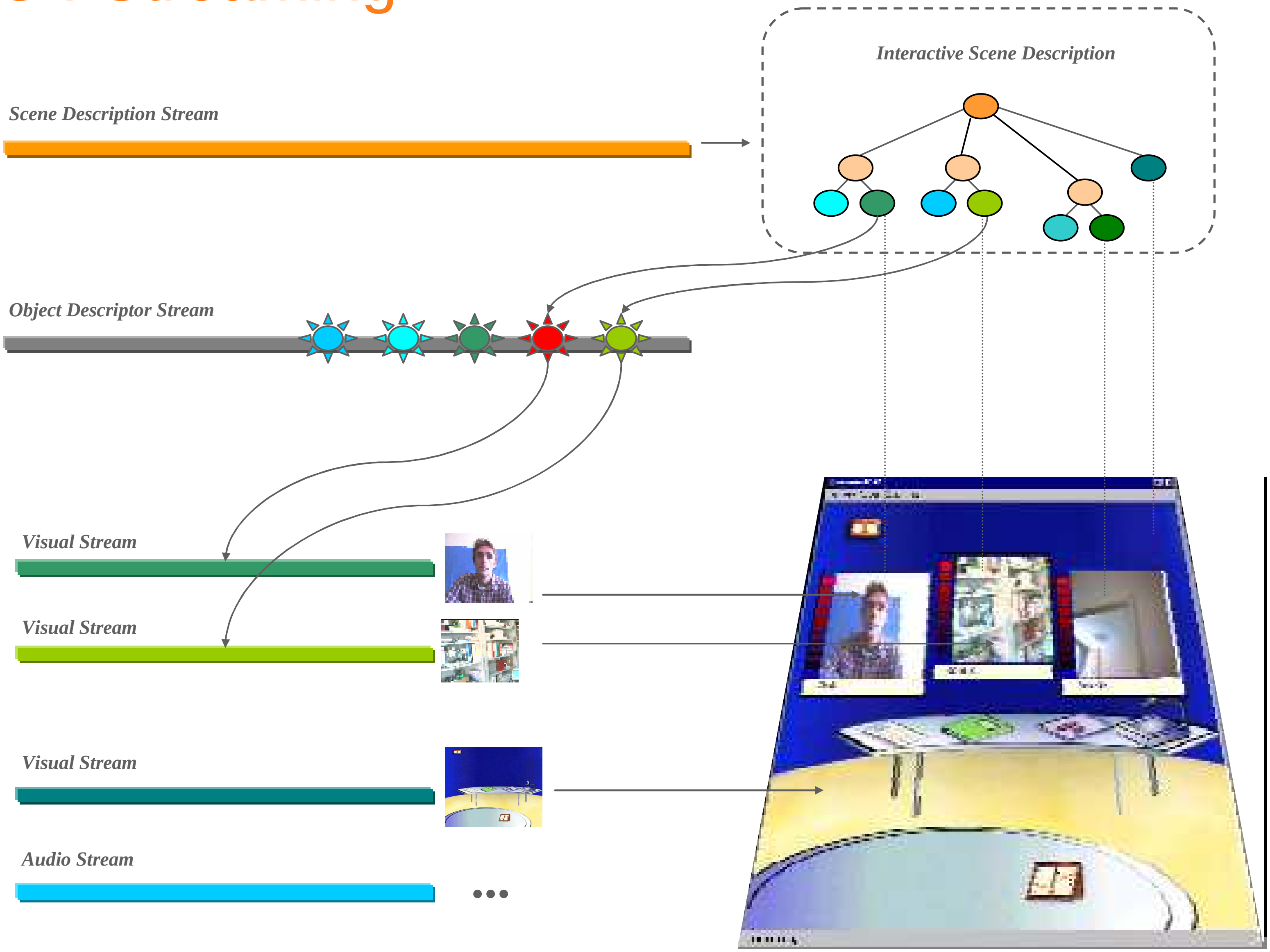
Possible architectures for 3D adaptive streaming

- Image based rendering : pre-computed images or 3D rendering on the server side
- Immediate drawing mode : PEX [Rost89], distributed GL
- Local copy:
  - Flash card memory
  - Download-and-play : VRML'97, X3D
- Real-time streaming (Push mode) : MPEG4-BIFS
- Adaptive streaming or view-dependent streaming:
  - On demand downloading: [Schmalstieg96]
  - Multi-resolution streaming: [Hoppe96, ChimA98]
  - Visibility-based streaming: [Cohen-Or98, Marvie03, Bouville05]

MPEG4 Streaming for 3D scenes

- Modeling with a scene graph (id. VRML'97)
- Two kinds of stream :
  - BIFS-Update :
    - Scene graph downloading
    - Modifications of scene graph by adding – removing nodes
  - BIFS-Anim :
    - Animation data: transformation, colors
- Packets of data with time stamps :
  - Control of decoding and visualization time

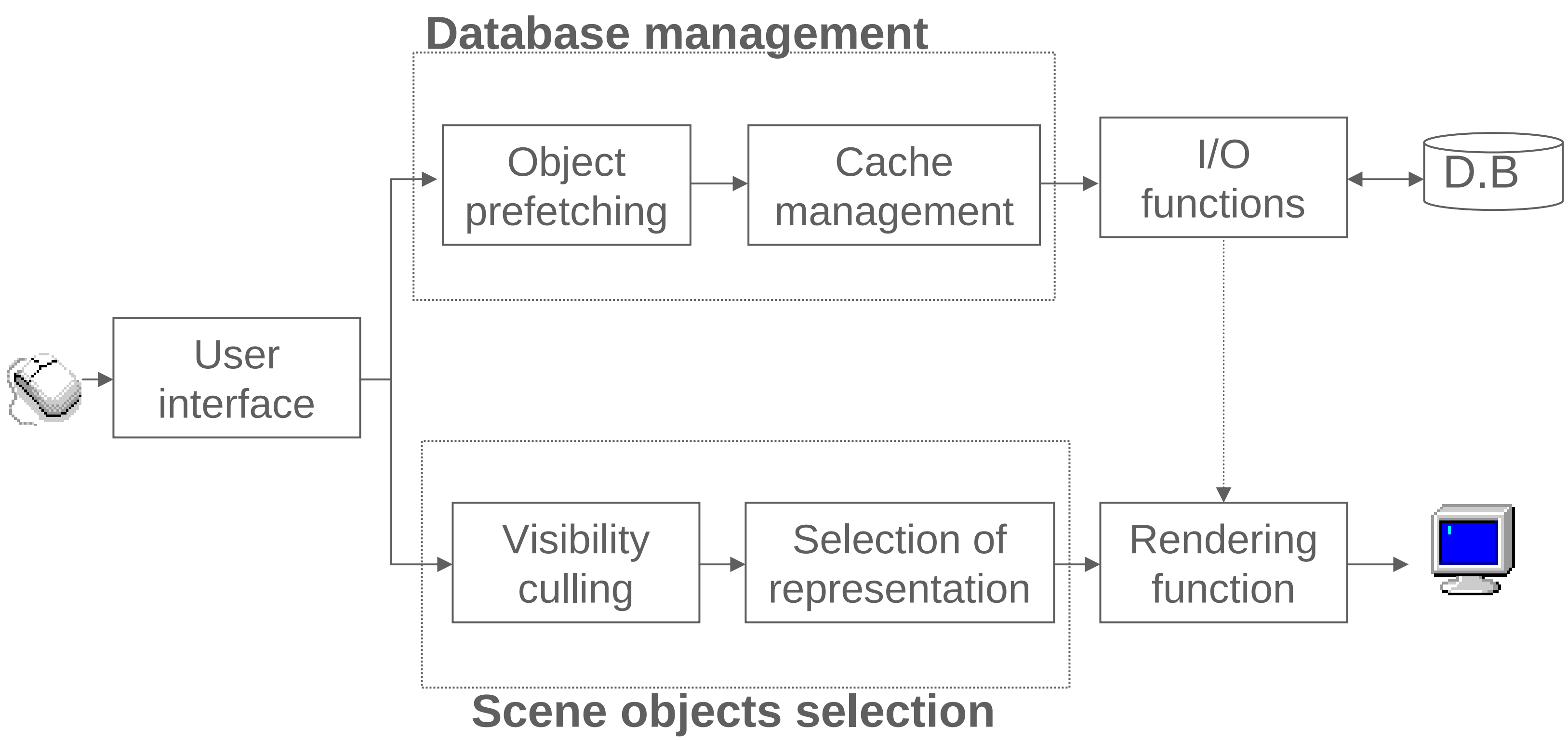
MPEG4 Streaming



Adaptive streaming

- ➔ Scene data transmission is adapted to the current viewpoint position
  - Objectives: minimize network and terminal resource usage
- ➔ Two criteria:
  - Visibility: FOV, inter-object occlusion
  - Distance to object: LOD selection (geometry and textures)
- ➔ Constraints:
  - Minimize server workload to allow many client connections
  - Allow P2P architecture

Interactive visualization with adaptation



Visualization processing flow [Funkhouser93]

Specific network-based visualization issues

- Minimizing network resource:
  - Fast transmission: data compression
  - Reduction of data to be transmitted :
    - Adaptive streaming
    - Locality and reversibility of Refinements/Simplifications in LODs
    - Optimized model selection:
      - *Cost/Benefit analysis, [Hesina98, Teller01]: maximizes benefit given bandwidth constraint*
      - *Distortion/Lifetime analysis [Pasman03]: compares image-based models to geometry simplification, minimizes bandwidth at given distortion*
    - Client caching:
      - *LRU, multi-resolution, MRM (Most required Movement) [ChimB98]*

Specific network-based visualization issues

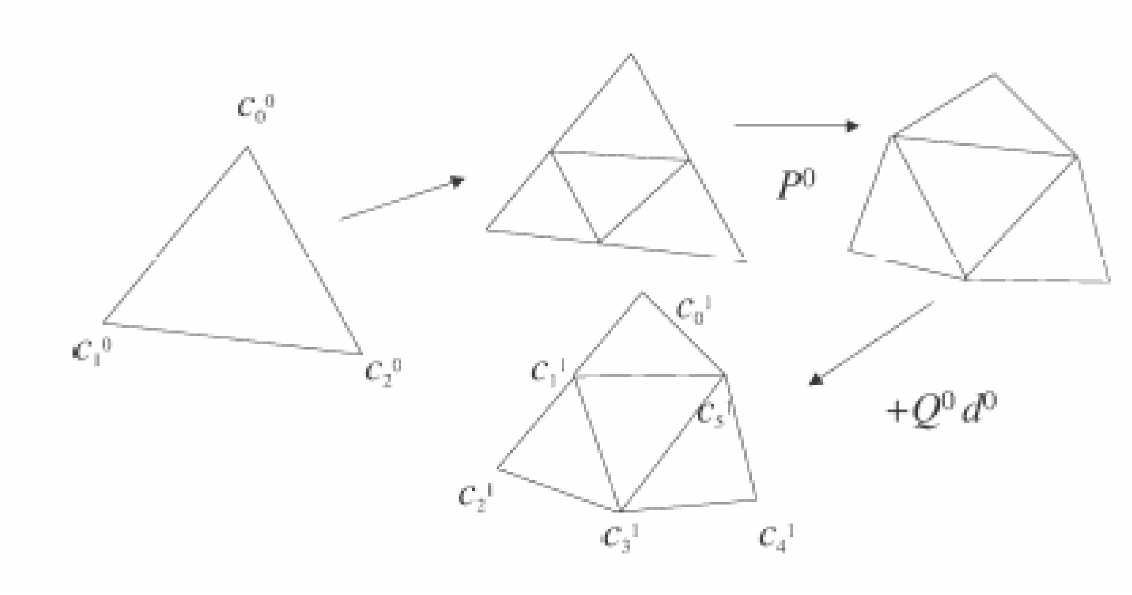
- Compensation of latencies:
  - Data prefetching by:
    - Widening the FOV
    - Lookahead through viewpoint motion prediction [ChimA98, Chan01, Chan05, PopescuA02]
  - Optimization of data packing
    - according to priority, internal dependencies or visual error [PopescuB02]
  - Use of non reliable protocols (UDP),
    - Faster streaming (no protocol handshaking) but needs to compensate for packet loss and unordered transmission [Harris02]
    - or minimizes a probabilistic error criteria (connectivity weighs more than geometry) [Al-Regib04]

Client-server architectural issue: the data selection task

- Server-side selection: clients send their current viewpoint position to the server
  - Ideal for optimum selection: direct access to object models, PVSs,…
  - No additional data for the clients (scene structure, PVS,…)
  - But:
    - Redundant data streaming (client cache content not known from the server)
    - Higher server workload
- Client-side selection:
  - No redundant data streaming
  - Low server workload
  - Allow dynamic adaptation to client storage and computing resource
  - But:
    - Additional data must be sent to clients (scene structure, PVS,…)

Wavelet subdivision surfaces for terrain visualisation using MPEG4-3DGC

- Use triangle quadrisections
  - No connectivity encoding required
- P: low-pass (smoothing operator)
- Q: high-pass (wavelets)
- Wavelet coefficient compression: zerotree coding technique [Khodakovsky00]

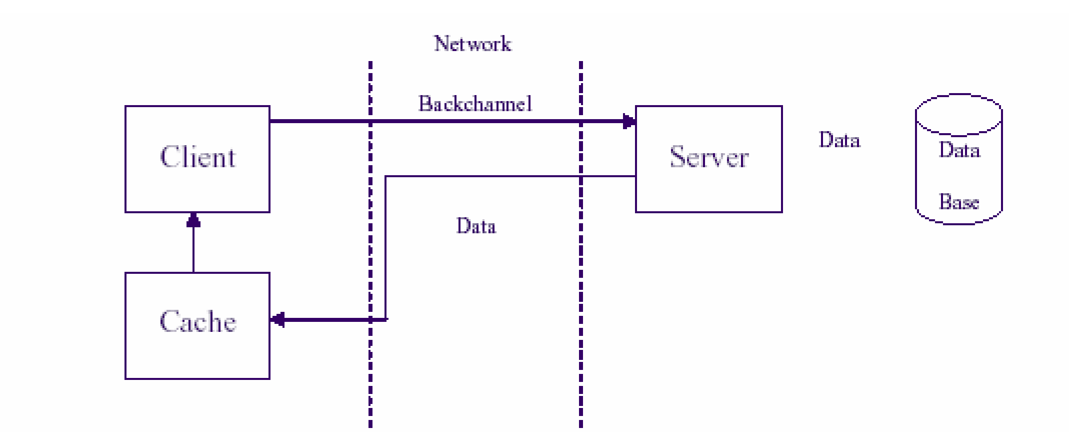


Subdivision Surfaces as a geometric predictor

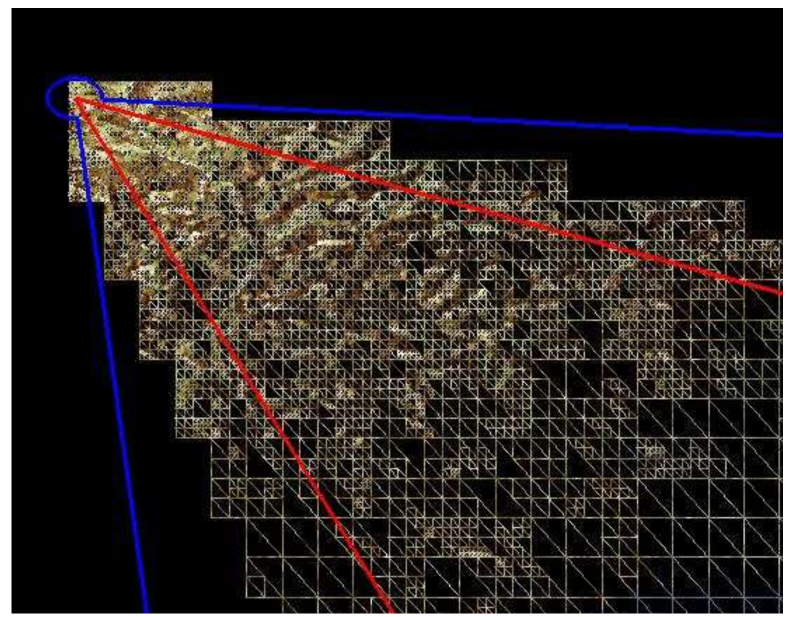
- More complex models → additional information between levels

Adaptive streaming with wavelet subdivision surfaces [Gioia04]

- Details managed according to the FOV and the distance to viewpoint
- Server-side selection: the server sends detail (wavelet coefficients) according to received viewpoint data



Backchannel stream: viewpoint +
cache content description

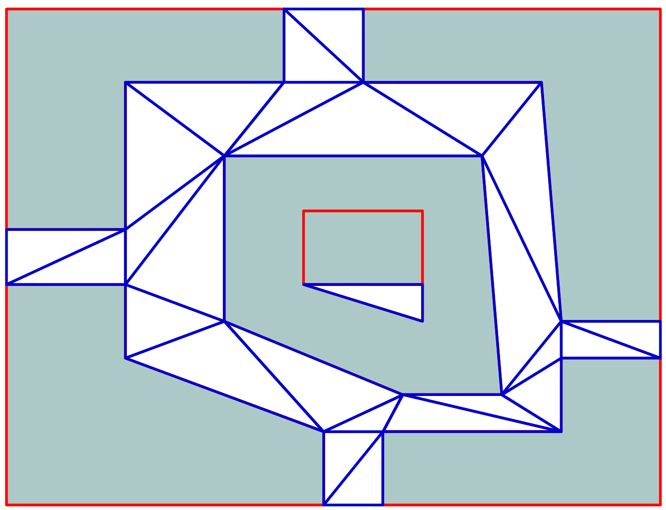


Urban environment: the case of 2D1/2 building geometry [Royan03]

- Building models based on prisms (footprint + elevation)
- Cannot be considered as a mesh
- Imposter bandwidth cost too high [Pasman03]

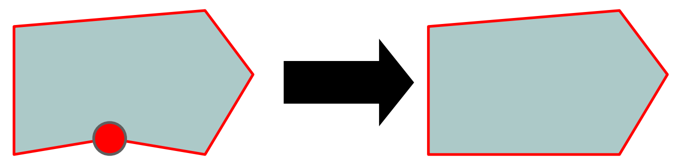
Construction of a LOD hierarchy: the PBTree

- Reconstruct the topological information of the scene
  - Triangulation of the empty space of the scene, constrained by the building footprints

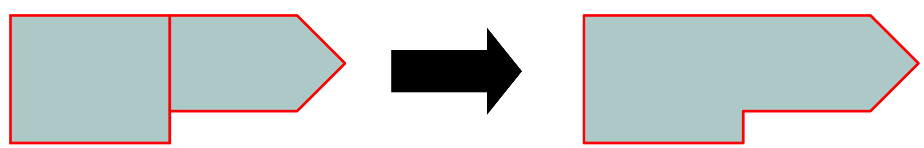


Simplification operations

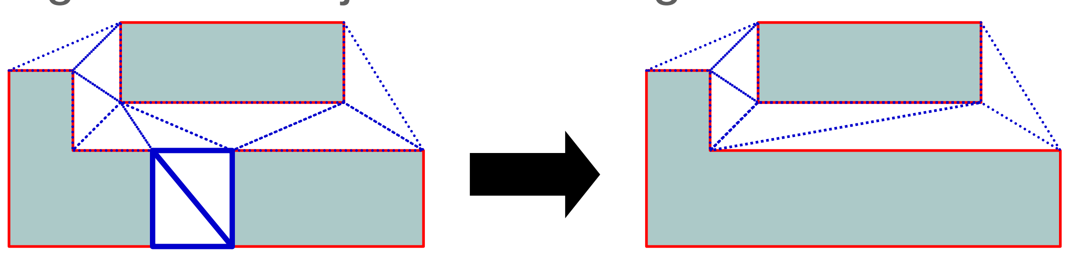
- The different possible simplifications of the scene mesh :
  - Suppression of a vertex



  - Merge of adjacent buildings



  - Merge of non adjacent buildings

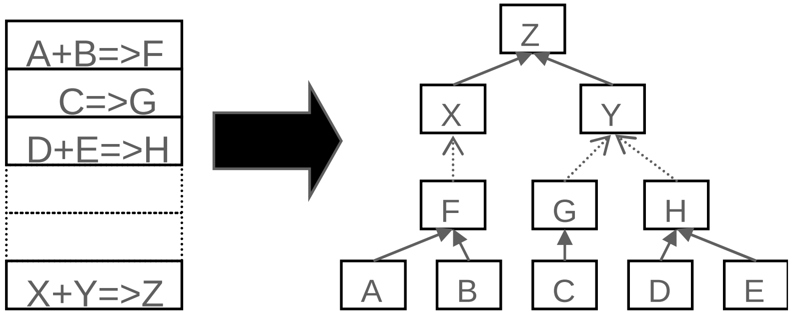


PBTree construction

- ➔ Original buildings are the leafs of the tree
- ➔ Search for all the potential simplifications of the scene.
- ➔ Assign each potential simplification an error metric taking as parameters:
  - The volume difference between the original buildings and their merging.
  - The altitude difference between the merged buildings
  - The height difference between the merged buildings.
  - The minimal distance between the merged buildings.

Construction of the PBTree

- Generation of the « Progressive Building Tree »
  - Bottom-up tree building
  - The generation of the « Progressive Building Tree » does not depend on the point of view.
  - The selection of the level of detail depends on the point of view.



List of potential simplifications sorted according to their costs

PBtree

PBtree important property

➔ Main characteristics of the street geometry are kept

Pbtree rendering: simple method

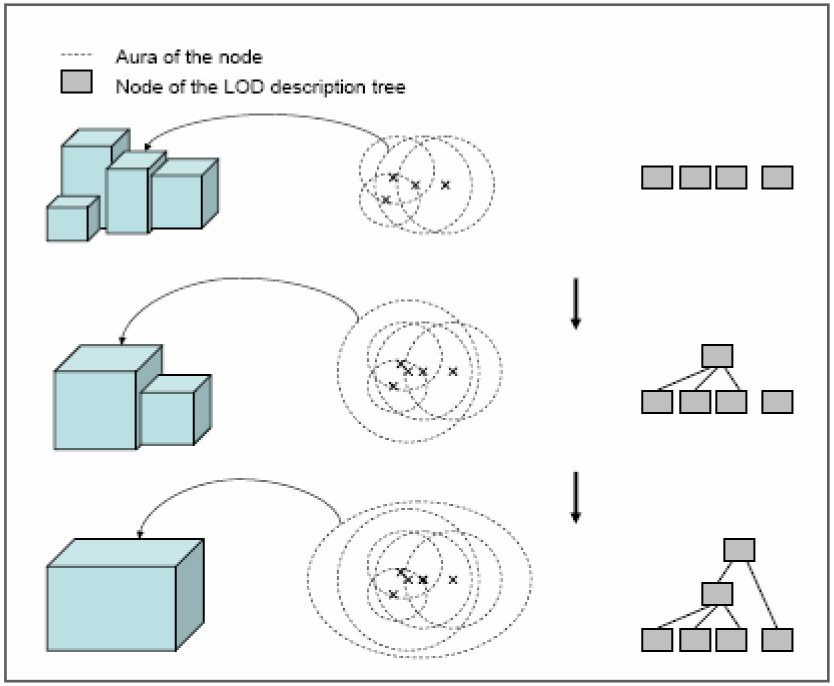
- A geometric error is assigned to each node
- Until geometric error of new nodes not visible {
  - Request children nodes to server }
- Advantage:
  - No knowledge of scene structure required for the client
  - Simple request
- Drawback:
  - Scene loading at start-up could be slow: many server requests necessary to reach the appropriate PBTree nodes

Video

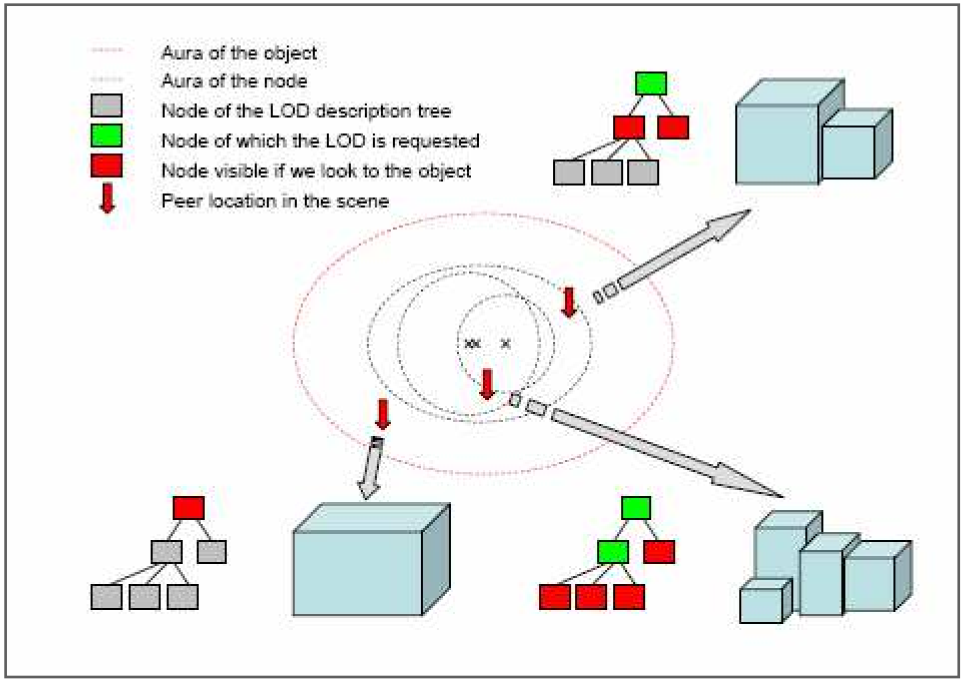
http://perso.rd.francetelecom.fr/bouville/index_fichiers/video_wavelet_PBTree.avi

PBTree rendering: fast method

- ➔ Use a LOD description tree: the LODDT
- ➔ LODDT can be downloaded in a progressive fashion



LODDT building



Rendering with LODDT

Conclusion

- Work in progress in 3DGC:
  - PBTree will be adopted in the next MPEG-3DGC amendment
  - Procedural modeling of roof and facades under discussion [Royan06]
- Techniques well-suited to P2P networking:
  - Low server burden
  - Easy scene partitioning
  - Reduced network resource usage
  - But not so easy to get self-organizing properties (work in progress)
- Bibliography:
  - http://perso.rd.francetelecom.fr/bouville/index_fichiers/Ayia_Napa.html