ETH Eidgenössische Technische Hochschule Zürich

\blu:'si:\

The Blue-C.

Martin Näf

Computer Graphics Group

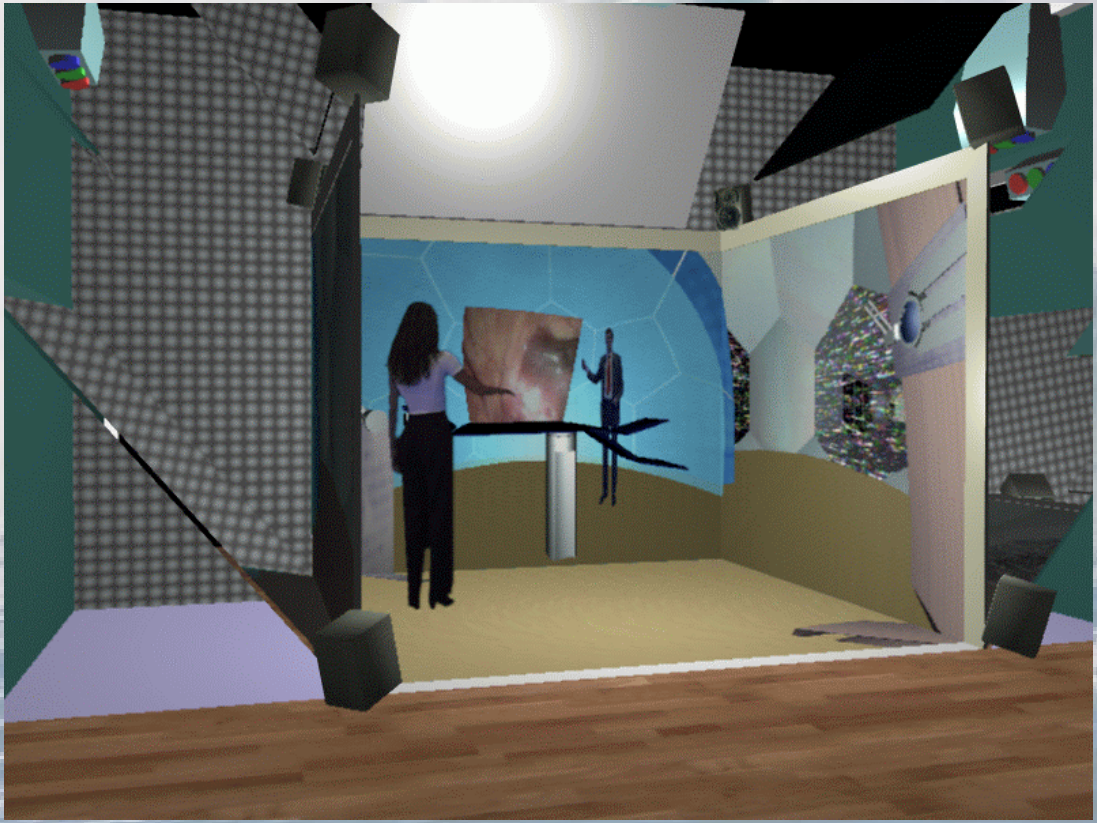
ETH Zürich

State of the Art

- The Cave (Univ. of Illinois)

State of the Art

- Teleport/Virtual Meeting (GMD)



State of the Art

- Office of the Future (UNC)

Limitations

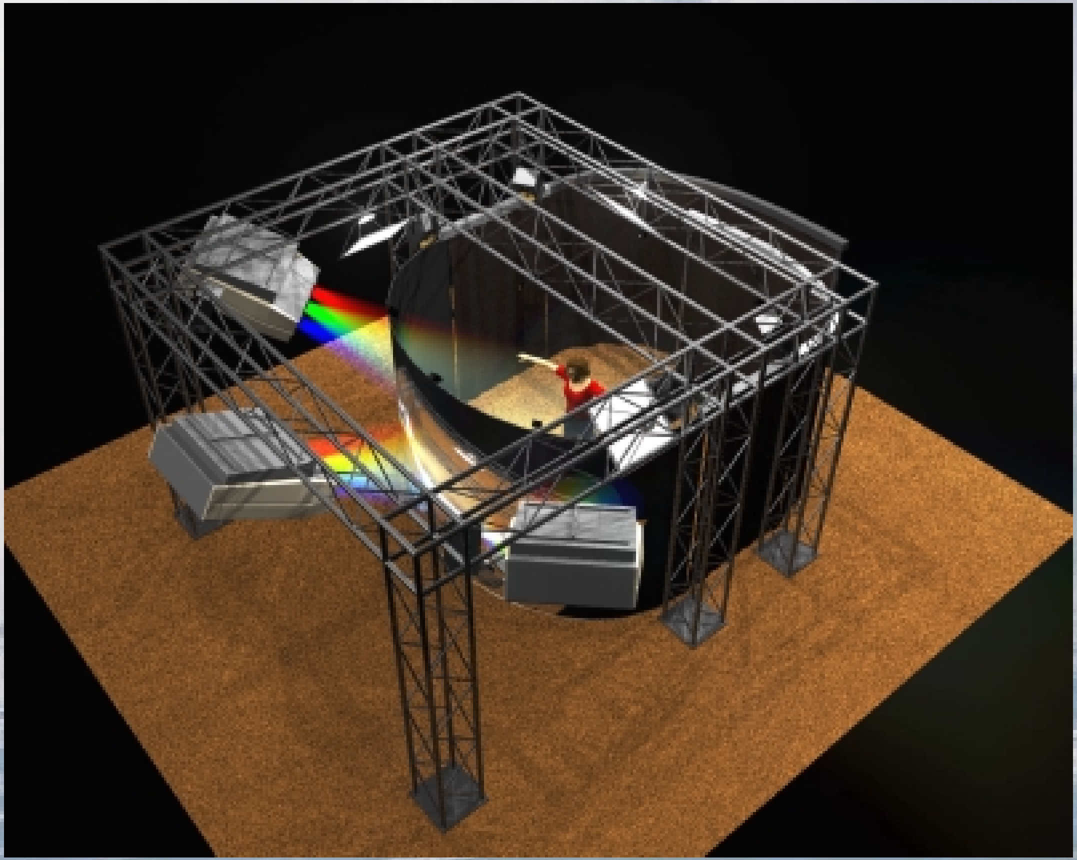
- Integration of 3D projection and real-time video acquisition very limited
- No real time vision systems embedded
- No (truly) hybrid rendering methods
- Current user interfaces are still in children’s booths

Goals

- Build a highly immersive VR environment for collaborative work
- Allow users to freely navigate, meet and collaborate in virtual worlds
- Real time acquisition and 3-D composition of live video streams of real actors in virtual environments

- Polyproject : CGG, CVG, PCCV, CAAD and ZPE

Blue-C.



Eidgenössische Technische Hochschule Zürich

\blu:'si:\

Blue-C.
Application Building Interface

A General Purpose
Collaborative Immersive Virtual Reality
Software Interface

Table of Contents

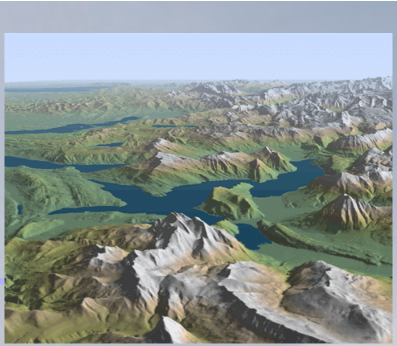
- Collaborative systems
  - Introduction
  - Data types
  - Decision problems
- Scene-graph
  - Introduction
  - Market overview

Collaborative Systems

- Multiple sites
- Distributed static and dynamic data
- Distributed modifications on data
- Distributed decisions

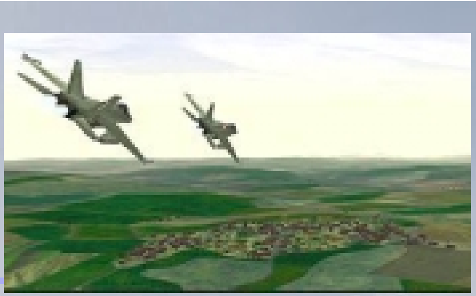
- Extensive research for battlefield simulations (SIMNET)

Data Types: Static



- Terrain, buildings, installations, vehicles operating on predefined paths
- Global data storage required
- Data distribution
  - Network file system (NFS, AFS, etc.)
  - "Web-interfaces" (HTTP, FTP)
  - Proprietary solutions
- Solutions are available
  - CAVERN: HTTP, remote file I/O

Data Types: Dynamic



- Controlled vehicles, actors, particles
- Object data
  - Position and rotation
  - Velocity vector
  - Additional state
- Update events
  - Delivery and consistency guarantees
  - Latency
- Solutions
  - HLA/DIS protocols

Data Types: Streams

- Audio and video data
- Geometry (MPEG 4)

- Strict real-time conditions
- Constant bandwidth and latency

- Compression
- Adaptive algorithms – react to change in conditions

Distributed Modifications

- Consistency problem
- Smooth updates (i.e. vehicle positions)

- Transaction system
- Exclusive locking vs. continuous updates

Distributed Decisions

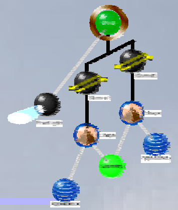
- Problem: Hit-test on outdated data
- Coherent decisions

- Solutions:
  - Client-Server concept
  - Strict locking
  - No guarantees
  - Anticipation

Conclusions

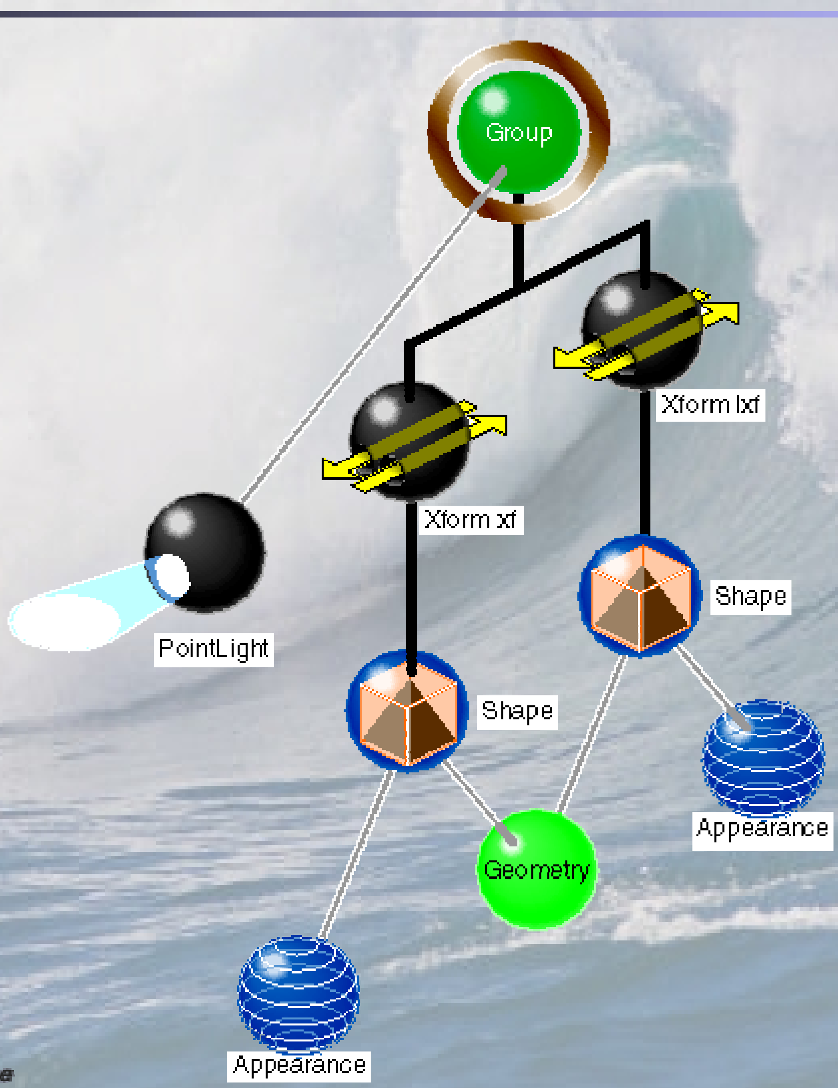
- Distributed database in real-time environment
- No single solution for updates or decisions
- Different solutions available

Scene-Graph Introduction



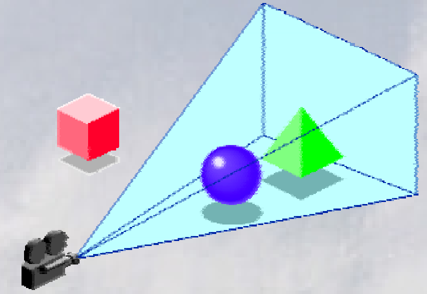
- Main data structure for rendering
  - Geometry
  - Object attributes (material, textures)
  - Environment (Lights, horizon)
- Hierarchical structure

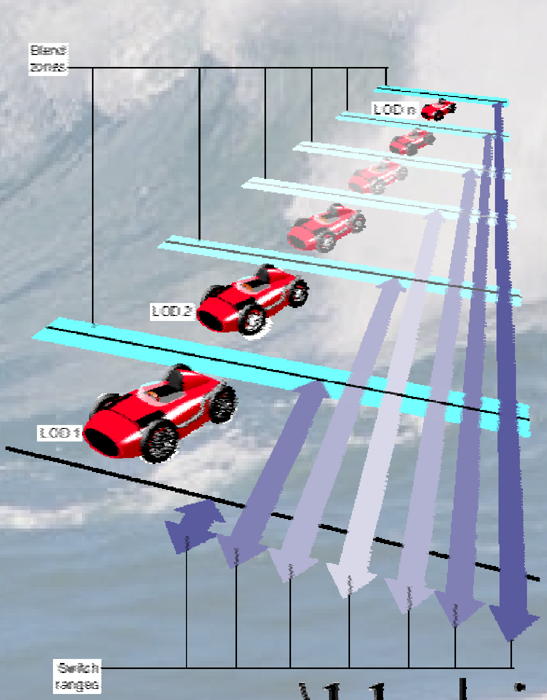
Scene-Graph Structure



Scene-Graph Operations

- Traversal
  - Culling
  - Level of detail selection
  - Rendering
  - Hit-test
- Object modifications
  - Geometry
  - Attributes
  - Meta-data





Eidgenössische Technische Hochschule Zürich

\blu:'si:\

Why not write your own?

- Optimizations
  - Avoid duplicate traversal of nodes
  - Avoid OpenGL-state changes
  - Texture optimizations
    Reduce downloads, clip-mapping, paging
  - Automatic level of detail selection
  - Optimize culling (bounding boxes etc.)
- Multi-CPU / multi-pipe support
- File loaders, format converter tools

Requirements

- Large models
- Real-time support
- Multi-pipe, stereo rendering
- Multi-CPU support
- File format compatibility
  - Geometry (VRML, IV, Alias, ProEngineer)
  - Texture (TIFF, GIF, JPG)

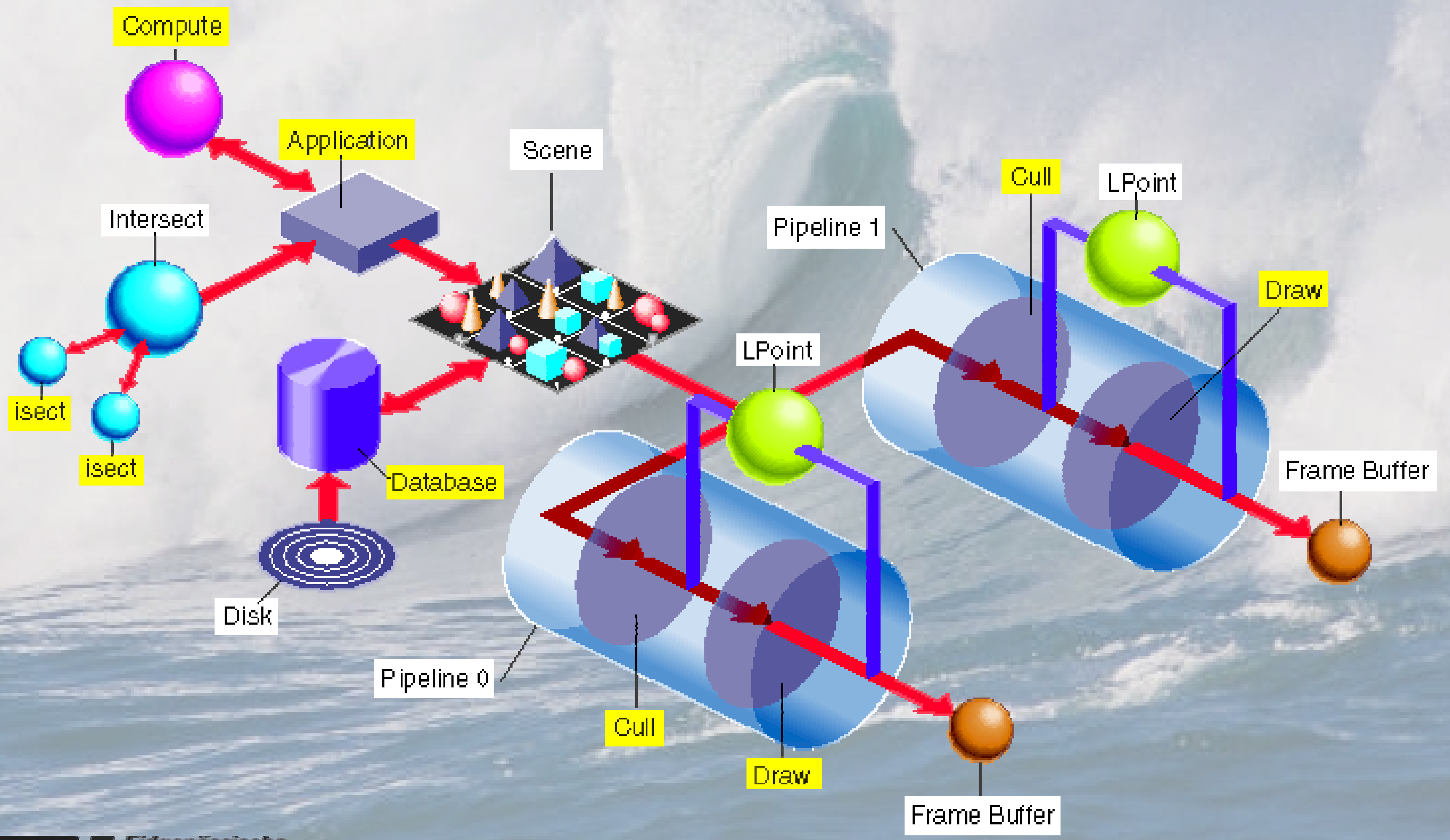
Iris Inventor

- Developed by Silicon Graphics
- Pro
  - Easy to use
  - Extensible, object-oriented design
  - “Active objects” allow interactive applications (➠ VRML)
- Con
  - Efficiency
  - No multi-pipe support
  - No multi-CPU support

Performer

- Successor to Inventor
- *THE* standard for VR-applications
- Pro
  - Fast, with real-time capabilities
  - Multi-pipe, multi-CPU support
  - Clean, extensible design
- Con
  - Specialized for VIS/SIM applications
  - Multiprocessing limited to rendering
  - No freeform-surface support

Performer - Processes



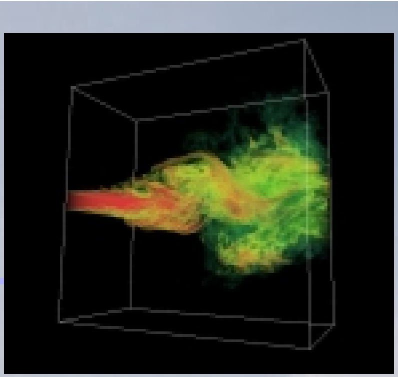
Cosmo3D / Optimizer

- Cross-Platform API developed by SGI
- Base for Java3D
- Pro
  - VRML support
  - Layered approach: Modules for CAD, Vis/Sim
  - Support for large models (enhanced culling)
- Con
  - No real-time features
  - Future support and development?

Open GVS

- Owned by Quantum3D
- Pro
  - Multi-platform support
  - Fast
- Con
  - Limited multi-CPU support
  - Not really "open"

Open RM



- Developed by R3vis
- Pro
  - Open source
  - Volume rendering support
- Con
  - No multi-CPU, multi-pipe support
  - Very limited file format support
  - Limited support

Open Scene Graph

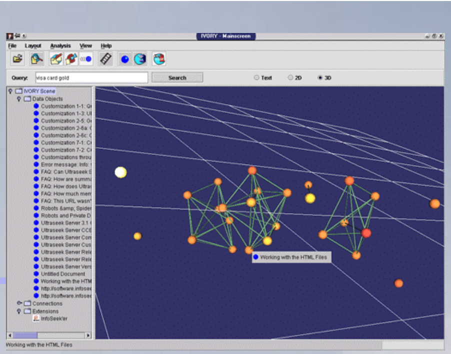
- Open source (ZGDV Darmstadt)

- Pro
  - Open source, cross platform
  - Multi-CPU, multi-pipe support
  - Clean design from scratch
  - Support for free-form surfaces
- Con
  - No real-time features
  - Not yet available

DirectX

- Microsoft Multimedia API
  - Direct3D retained mode
- Pro
  - Supports animation
  - Optimized for inexpensive hardware
- Con
  - “Better display-list”
  - No multi-pipe support
  - Single platform (Win32)

Java 3D

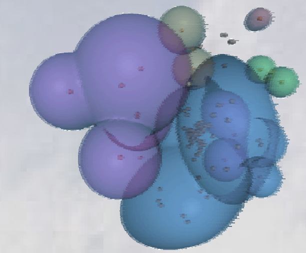


Query: visa card gold

○ Text ○ 2D ● 3D

Working with the HTML Files

- Scene-graph for Java



- Pro
  - Portable
  - Clean, modern interface
  - Good performance for static scenes
- Con
  - Java-only
  - No multi-pipe support
  - Real-time constraints vs. garbage collection

Dead APIs

- Open Inventor
- OpenGL++
- Fahrenheit

- Many more

Conclusions

- No one-fits-all solution
- Performer is still the first choice for VR
- Open Scene Graph?

Discussion