

Parallel Stereo Visualization for Clusters with OpenInventor: A Case Study for the Automotive Industry

F. Vega Higuera[•], G. Sußner[•], T. Rheuding[‡], G. Greiner[•]



- Computer Graphics Group, University of Erlangen-Nuremberg
- ‡ BMW AG

Contents



- ◆ Motivation
 - ◆ OpenInventor Stereo
 - ◆ Eye-based parallelization
- ◆ System Design
- ◆ Implementation
 - ◆ **OISLC** library
- ◆ Results
- ◆ Conclusion and perspectives

OpenInventor® Library

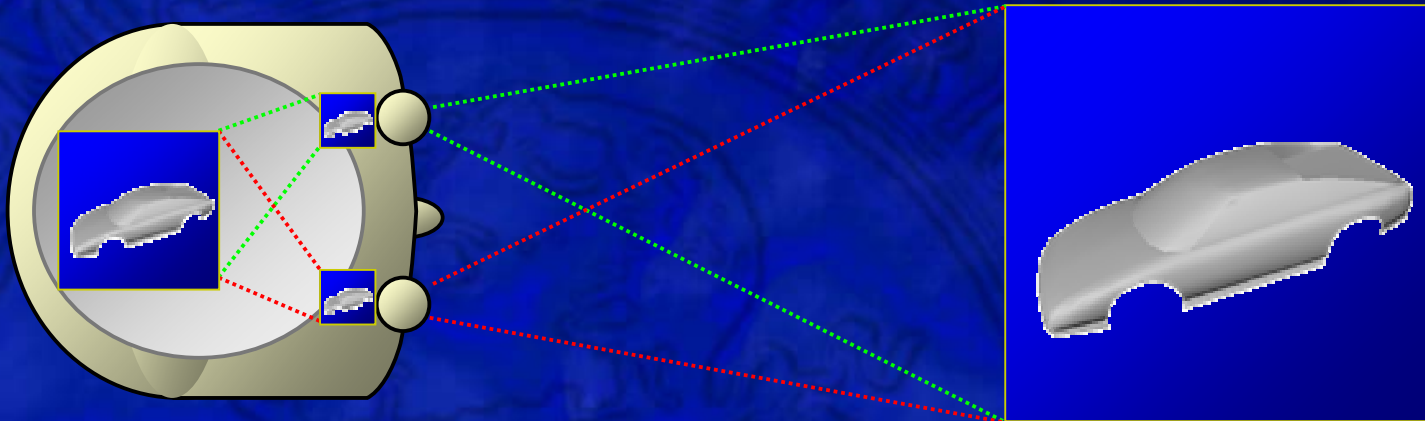


- ◆ “Retained Mode” library.
- ◆ Based on the Scene Graph concept.
- ◆ Nodes are the building blocks; fields describe their properties.
- ◆ It is **NOT** “Thread-Safe” (Open source).
- ◆ Intensively used for CAD within BMW.

Stereo Viewing

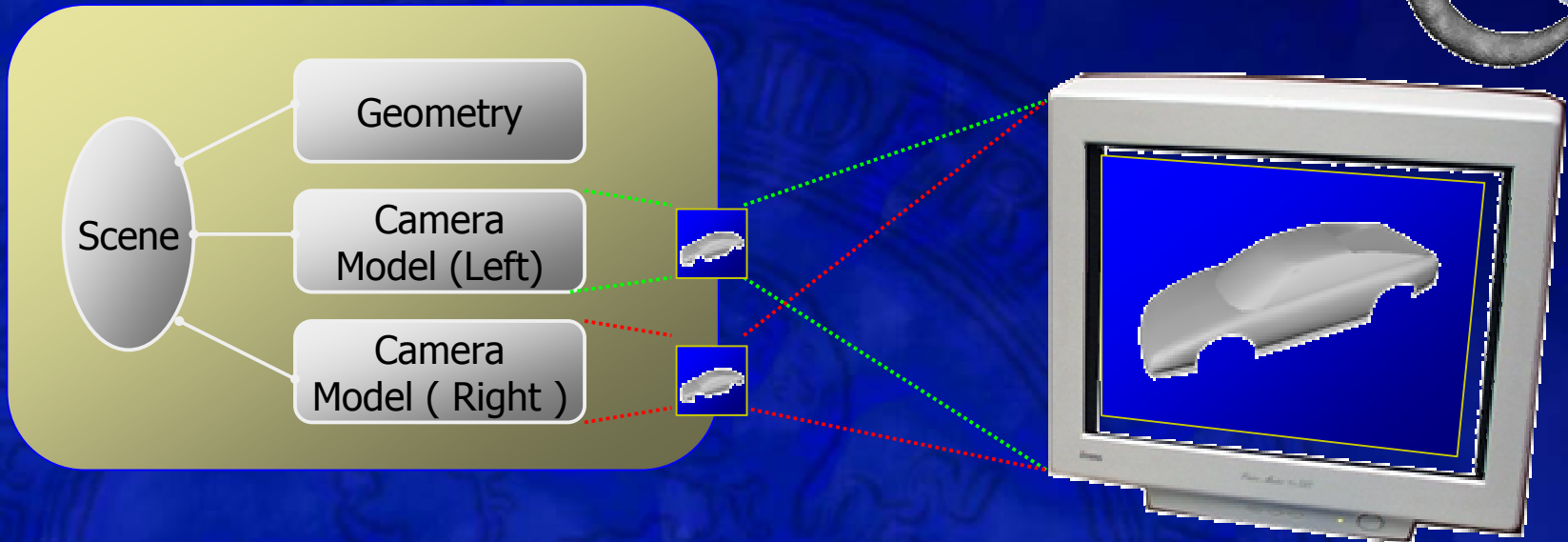


- ◆ Human beings have stereo vision



- ◆ Provides very important spatial hints.
- ◆ It can play a major role for the CAD process.

Standard OpenInventor® Stereo

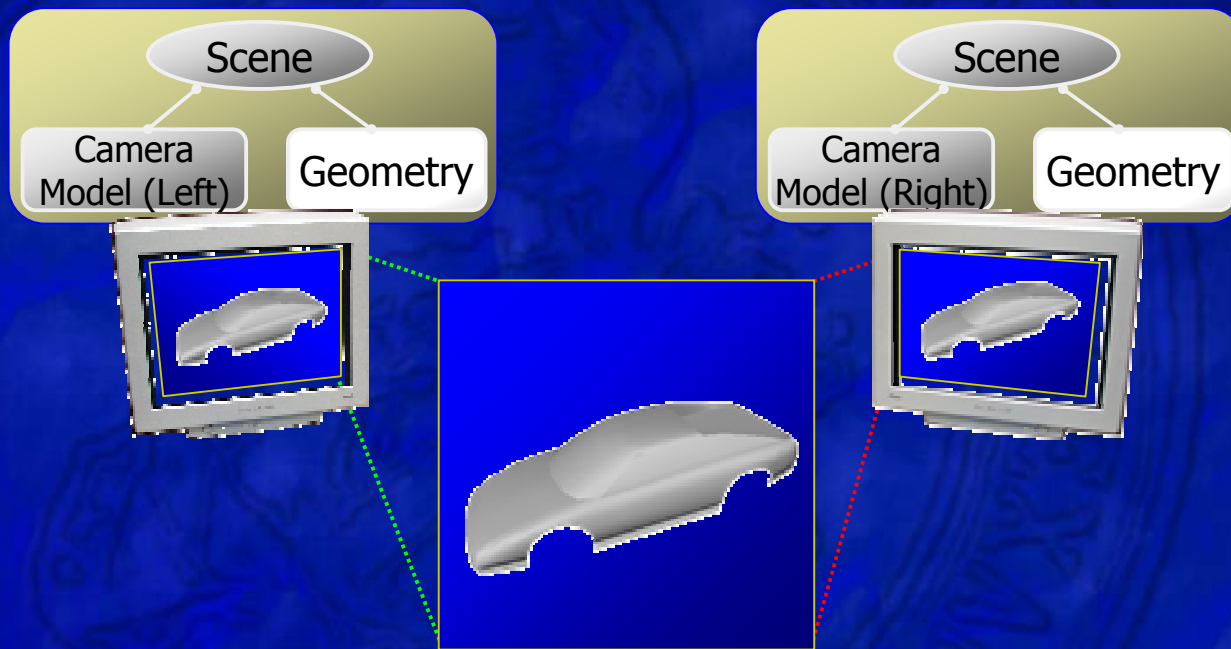


- ◆ One CPU computes the scene for two different camera positions.
- ◆ Resulting frames are displayed alternatively. This means 25 stereo FPS for a 50 FPS capable system.

Idea: Eye based Parallelization



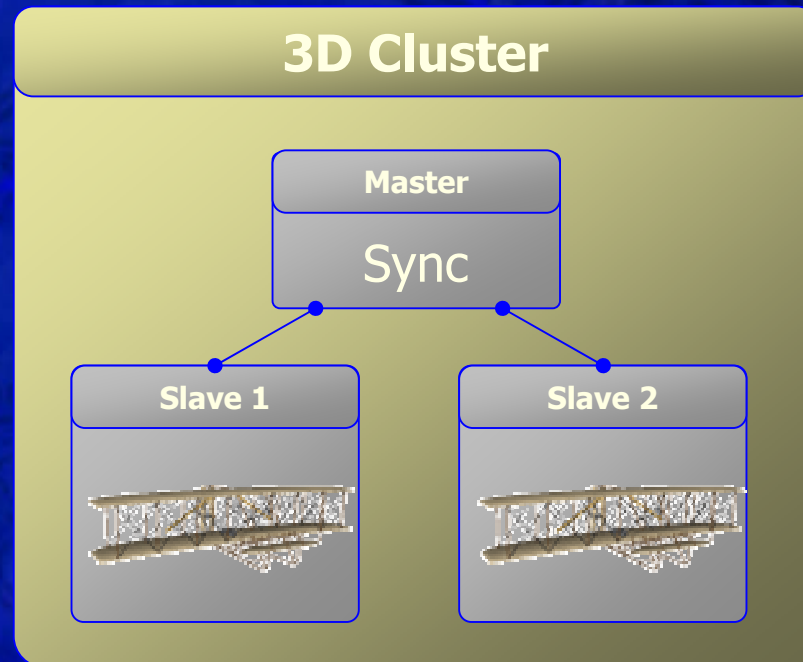
- ◆ Use a CPU for each camera position.
- ◆ First CPU creates the right image while the second CPU creates the left image.



A PC cluster can do the job



- ◆ A master node for synchronization of rendering slaves.

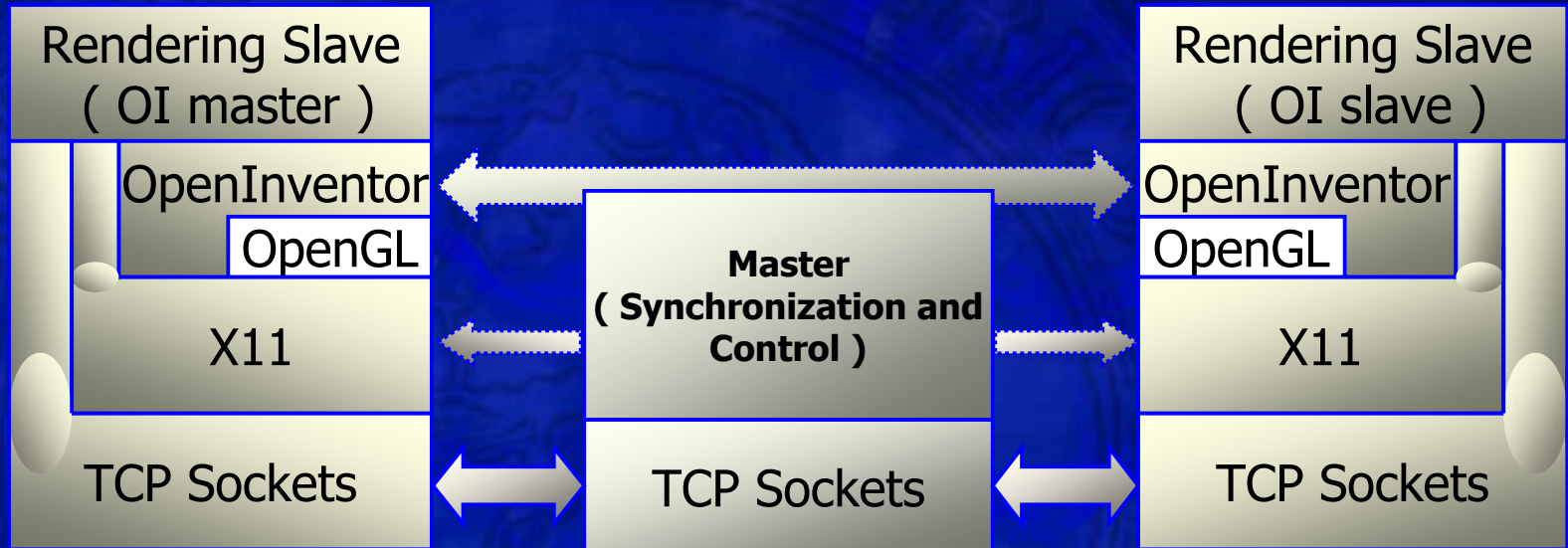


System Design



- ◆ What do we need / want ?
 - ◆ “Standard Desktop”: both rendering computers produce the same image for 2D segments of the screen. Standard functionality is available (moving, resizing of windows).
 - ◆ OpenInventor applications provide parallel stereo rendering.
 - ◆ Both tasks must be absolutely synchronized!

Two level architecture



Implementation

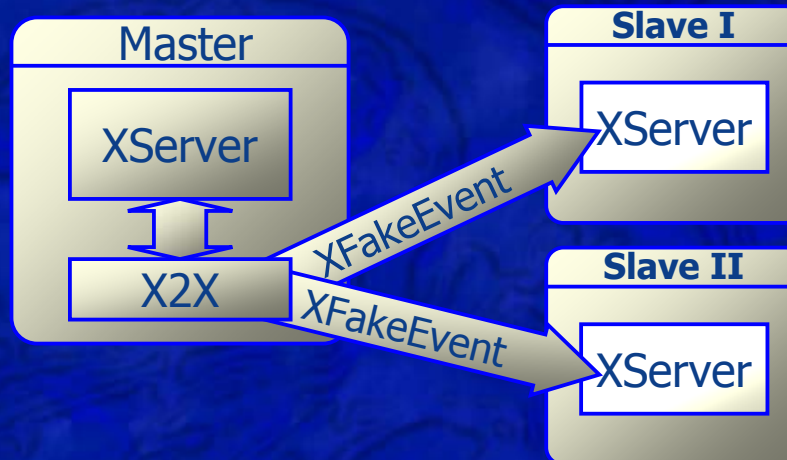


- ◆ Rendering slaves are identical systems. Equipment and software.
- ◆ X11 is synchronized from the master node.
- ◆ Both slaves have a copy from the Scene-Graph. One of the slaves is used as reference and processes changes on the Scene-Graph. The second slave “just” replicates the reference scene-graph.
 - ◆ X11 Synchronization does NOT ensure OI synchronization.
- ◆ Only CHANGES are transmitted.

X11 Synchronization



- ◆ **x2x** provides functionality for controlling multiple “slave” X-Servers by a “master” X-Server.

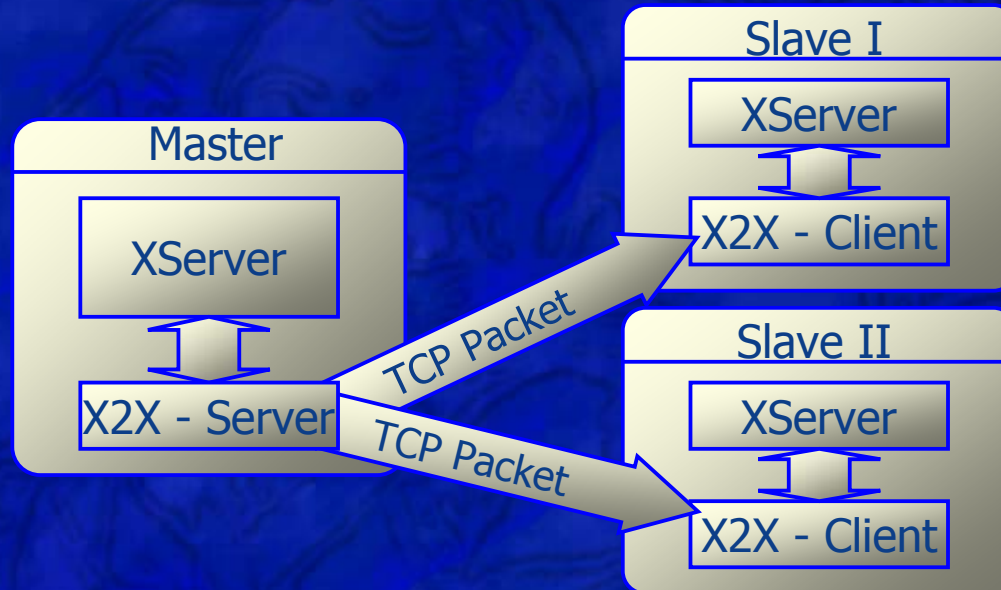


- ◆ **x2x** provides “Multipurpose” functionality; we have a fixed architecture. Optimize!

Optimized X2X



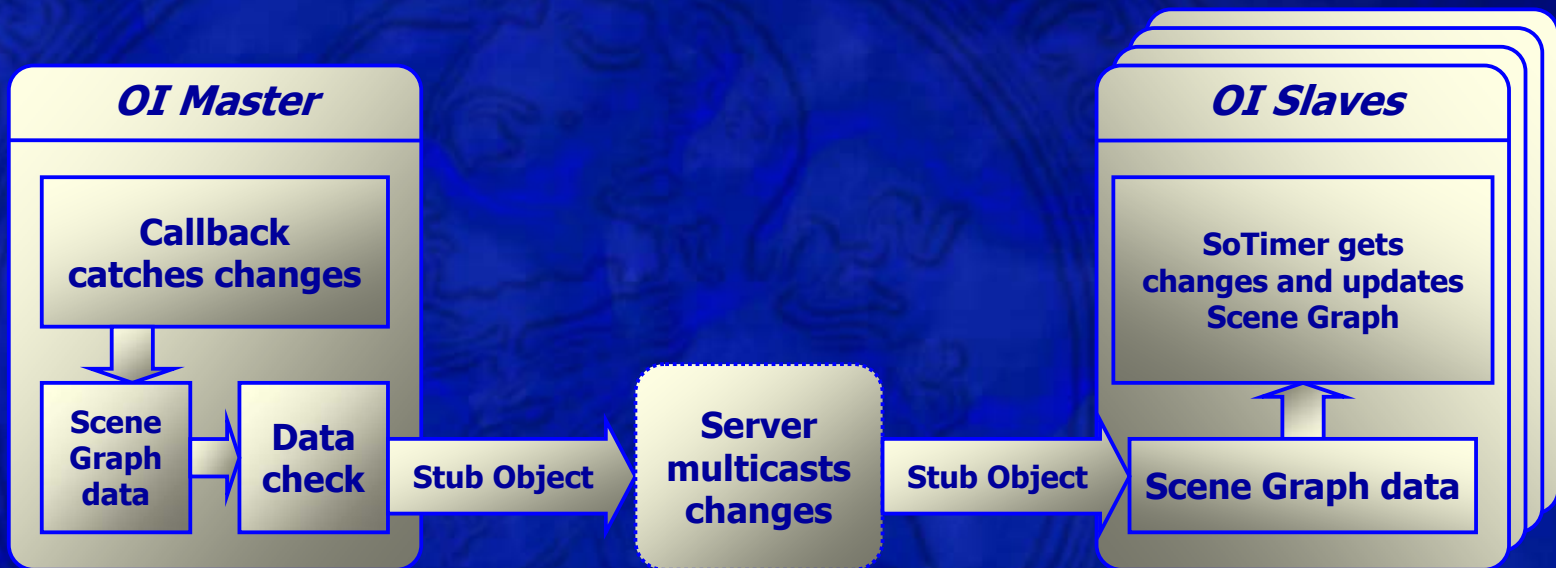
- ◆ Avoid using XProtocol over the net
 - ◆ 31% less overhead as **x2x**.
 - ◆ Perceptible increase of "smoothness".



OpenInventor® Synchronization



- ◆ OpenInventor functions can be called only from the SAME thread.



Frame Synchronization





- ◆ Only three instructions are necessary in order to port applications to the system:
 - ◆ `ClientOISLC.ConnectToServer(port, port)`
 - ◆ `ClientOISLC.ShareNode(SoNode*)`
 - ◆ `ClientOISLC.RemoveNode(SoNode*)`
- ◆ `XtViewers` \Leftrightarrow `XtStereoViewers`
- ◆ `SoGLRenderAction` \Leftrightarrow `StereoRenderAction`

Results



◆ Prototype System

- ◆ PIII@600 MHz.

 - ◆ A master and 2 rendering slaves.

- ◆ GeForce 4Ti 4200 at the slaves.

- ◆ Fast Ethernet hub.

- ◆ Output viewed on a passive stereo system.

 - ◆ Semi-Portable projection system.

Results



◆ Frame Synchronization

- ◆ Time difference between frame displays at the slaves.

Tests	Min	Max	Avg.	Pct. under 8ms	Avg. over 8ms
10.158	0ms	12ms	1,45ms	99,9%	10ms

* Time-outs : 43 (0,42%)

Performance



◆ Graphics load

Triangle Count	Triangles/sec mono*	Triangles/sec stereo**	Graphics load
700.398	4.146.356	3.172.109	89,52%
350.199	4.188.380	2.917.157	69,64%
66.137	2.702.357	1.693.107	62,65%

* Single PC based

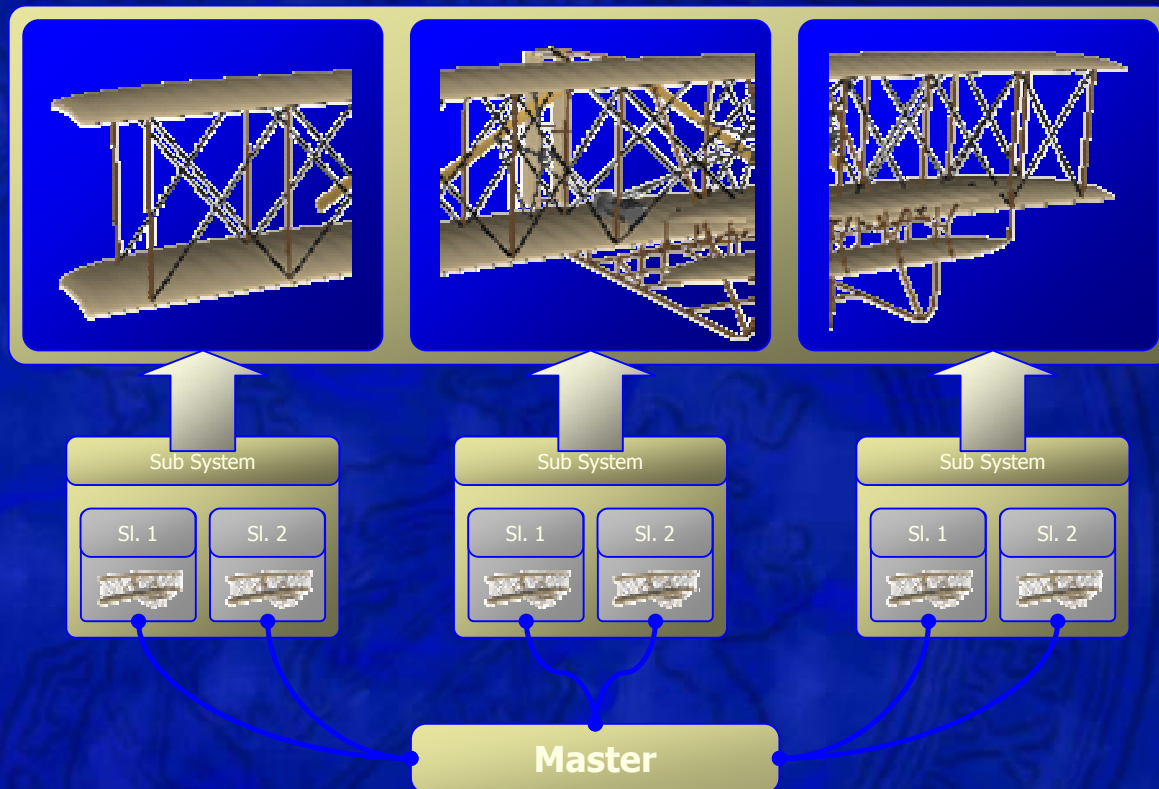
**Parallel Stereo System

Conclusions and Perspectives



- ◆ Eye-based parallelization for Open Inventor based on PC clusters.
- ◆ A standard X-Stereo Desktop is provided.
- ◆ OISLC makes it possible to port existing applications.
- ◆ System suited for static CAD data visualization.
- ◆ Starting point for multi-tiled PC-based Stereo desktop .

Conclusions and Perspectives



Thank you



Questions ?