Distributed Consistency Maintenance Scheme for Interactive Mixed Reality Environments

Felix G. Hamza-Lup
Univ. of Central Florida

July 23rd, 2004
Outline

- Distributed Collaborative VE
- Motivation
- Consistency in Distributed Interactive VE
- Sensors in Interactive VE
- Hybrid Nodes Architecture
- Experimental Setup & Results
- Conclusion
Distributed Collaborative Environments

- MR/VR & DS paradigms -
Distributed Collaborative Environments
- Examples of MR/VR based DCE & Trend -

- **DCE application**
  - Information/knowledge dissemination
  - Reduced costs, time and risks
  - Increased efficiency through team work

- **Examples & Trend**
  - **Industry**
    - Military simulations: (VR) SIMNET, NPSNET, (MR) MOUT ...
    - Entertainment: (VR) networked games, (MR) Project (ISMR’99) ...
    - Medicine: (AR) training tools (MMVR’03) ...
  - **Academia**: (VR) MASSIVE, DIVE, DEVA, (AR) Studierstube, Coterie...
  - **Trend toward Mixed Reality** (focus on AR)
Outline

- Distributed Collaborative VE
- Motivation
- Consistency in Distributed Interactive VE
- Sensors in Interactive VE
- Hybrid Nodes Architecture
- Experimental Setup & Results
- Conclusion
Motivation

- AR Applications in Medicine -

- AR Diagnostics/Prognostics
- Medical Procedure Training

AR Medical Training Concept

AR Diagnostics Concept

Courtesy of S. Johnson

Courtesy of A. State
Motivation
- AR Applications in Medicine -

AR Medical Training & Diagnostics

Need for interaction and remote information/data sharing
Research Questions

- How can we improve consistency in distributed interactive VE?

- What architecture and data distribution scheme would fit the new trend in distributed VE applications, i.e., the need to capture and distribute sensor data in real-time?
Outline

- Distributed Collaborative VE
- Motivation
- Consistency in Distributed Interactive VE
- Sensors in Interactive VE
- Hybrid Nodes Architecture
- Experimental Setup & Results
- Conclusion
Consistency in Distributed Interactive VE
- Terminology, Dynamic Shared State -

“The dynamic shared state constitutes the changing information that multiple machines must maintain about the networked Virtual Environment” ("Networked Virtual Environments – design and implementation", S. Singhal, M. Zyda)

The cause for inconsistency
- network latency (propagation, transmission, routing)
- computer system latency (rendering, buffering, etc.)
Shared State Maintenance in Distributed VE
- Techniques -

- **Approaches:**
  - centralized information repositories (pull/push architectures)
  - dead-reckoning algorithms (convergence & prediction)
  - frequent state regeneration (blind broadcasts, applications that do not require absolute consistency)

- **Other techniques for resource management:**
  - Communication protocol optimization (packet compression)
  - Visibility of data management (AOI)
  - Human perceptual limitations (LOD)
  - System Architecture
Outline

- Distributed Collaborative VE
- Motivation
- Consistency in Distributed Interactive VE
- Sensors in Interactive VE
- Hybrid Nodes Architecture
- Experimental Setup & Results
- Conclusion
Sensors in distributed VE
- Tracking Systems -

- Give the position and orientation of the 3D objects in the virtual environment

- Types:
  - mechanical
  - magnetic
  - optical
  - acoustic
  - inertial
  - hybrid
Sensors in distributed VE
- Tracking Systems -

- Polaris Tracking System: used to determine the position and orientation of real objects within the environment

- Probe = rigid configuration of IREDs
- IRED probes:
  - HMD probe - tracks the HMD position
  - Mannequin probe - gives the mannequin’s chin position
  - Digitizing probe - gives the 3D position of the probe tip
Outline

- Distributed Collaborative VE
- Motivation
- Consistency in Distributed Interactive VE
- Sensors in Interactive VE
- Hybrid Nodes Architecture
- Experimental Setup & Results
- Conclusion
Hybrid Nodes Architecture
- Passive & Active -

- Hybrid Nodes
  - active nodes (A)
  - passive nodes (P)
  - active forward node (AF)
  - passive forward node (PF)

00 - the state changes from *inactive* to *active*
01 - the state changes from *active* to *inactive*
10 - the state change from forwarding *off* to *on*
11 - the state change from forwarding *on* to *off*.
Hybrid Nodes Architecture
- System Perspective -

- A Distributed System Model
  - dynamic membership
  - graph representation

- Construction of a communication sub-graph
  - Core-Base Tree algorithm
    - guarantees that the path between any node and the core is the shortest
    - only one tree per multicast group
Hybrid Nodes Architecture
- Infrastructure Instance -

G=(V,E)
Hybrid Nodes Architecture

-Participants Sub-graph-

\[ G'(t) = (P(t), E') \subseteq G \]
Hybrid Nodes Architecture

- Communication Sub-graph (CBT) -

Rule: A node cannot join the multicast tree if the accumulated delay on the path to the core is greater than MaxValue.
Hybrid Nodes Architecture
- CBT, multicast tree -
Outline

- Distributed Collaborative VE
- Motivation
- Consistency in Distributed Interactive VE
- Sensors in Interactive VE
- Hybrid Nodes Architecture
- Experimental Setup & Results
- Conclusion
Experimental
- Design & Setup -

- Distributed Artificial Reality Environment (DARE)
  - set of OO libraries for 3D rendering, communication, node monitoring, assessment

- User interacts through an interface by applying a set of consecutive actions (e.g. rotations) on the virtual object
Experimental - Assessment Method -

- Assess orientation drift of a shared 3D virtual object

- Let
  - $q_s$ - orientation of an object at N1
  - $q_c$ - orientation of the same object at N2

- Correction quaternion ($q_E$) expresses the error between the actual orientation of the object and the desired orientation

\[
q_s = q_E q_c
\]

\[
q_E = (\omega_E, v_E)
\]

\[
\alpha = 2 \cos^{-1}(\omega_E)
\]
Experimental
- Passive Participant Setup -

- Two node setup contains an active and a passive participant

- We gradually increase the number of passive participants to five nodes
Experimental
- Results -

Drift behavior at node 2 with no drift compensation

Drift Angle (degrees)

Sample Number

No Update (speed=10 degrees/sec)
No Update (speed=50 degrees/sec)
No Update (speed=100 degrees/sec)

R² = 0.9829
R² = 0.9735
R² = 0.9534
Drift behavior at node 2 using the Event Updates method

- Experimental -

- Results -

Drift Angle (degrees) vs. Sample Number

- Event U. (speed=10 degrees/sec)
- Event U. (speed=50 degrees/sec)
- Event U. (speed=100 degrees/sec)

$R^2 = 0.511$

$R^2 = 0.6713$

$R^2 = 0.3671$
Experimental

- Results -

Drift behavior at node 2 using the Adaptive Synchronization Algorithm (ASA)

<table>
<thead>
<tr>
<th>Speed</th>
<th>Drift Angle (degrees)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 degrees/sec</td>
<td>0.7342</td>
</tr>
<tr>
<td>50 degrees/sec</td>
<td>0.4422</td>
</tr>
<tr>
<td>100 degrees/sec</td>
<td>0.5003</td>
</tr>
</tbody>
</table>
Experimental
- ASA Scalability -

Orientation drift behavior as the number of passive participants increases

\[ y = 0.105x + 2.6186 \quad R^2 = 0.7858 \]

\[ y = 0.097x + 0.9969 \quad R^2 = 0.7352 \]

\[ y = 0.0217x + 0.176 \quad R^2 = 0.8953 \]
Maintains a low and constant drift level

E.g. 100 degrees/s, 50 consecutive actions, 6 participants, 1 A, 5 P

- No Compensation  => $STD(\alpha) = 49.95^\circ$
- Event Update    => $STD(\alpha) = 2.39^\circ$
- ASA            => $STD(\alpha) = 0.62^\circ$
Experimental
- Discussion –

- Scalability regarding the number of participants
  - $\psi$ - average drift
  - $\psi_n \approx \psi_1$ i.e. good scalability

- Experimental results:
  - Passive nodes
    - $\psi_1 = 2.8$ (2 nodes setup)
    - $\psi_5 = 3.1$ (6 nodes setup)
Conclusion

- Consistency maintenance for distributed interactive VE

- Application:
  - Deformable models: deformation distribution and consistency
  - Deformable Lungs 3D Model
Acknowledgements

- Link Foundation
- NSF/ITR
- Optical Diagnostics and Applications Lab.

More info:
www.cs.ucf.edu/~fhamza