Intercept Tags
Enhancing Intercept-based Systems

David J. Zielinski
Regis Kopper
Ryan P. McMahan
Wenjie Lu
Silvia Ferrari
Background: Virtual Reality Software

Custom Virtual Reality Software

Desktop Computer Application

Intercept Technique

Virtual Reality Experience
Intercept Technique Example

**Standard Case**

- **Drawing Command**: (e.g. “draw a triangle”)
- **Graphics Driver**: (opengl32.dll)

**Intercept Case**

- **Drawing Command**: (e.g. “draw a triangle”)
- **Intercept Driver**: (new opengl32.dll)
- **Graphics Driver**: (opengl32.dll)
# Existing Intercept Systems

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Techviz</td>
<td>Commercial software</td>
</tr>
<tr>
<td>Conduit from Mechdyne</td>
<td>Commercial software</td>
</tr>
<tr>
<td>Chromium, WireGL</td>
<td>Open-source project</td>
</tr>
<tr>
<td><strong>ML2VR</strong></td>
<td>Open-source framework focused on MATLAB. Presented as a poster at IEEE VR 2013.</td>
</tr>
</tbody>
</table>
Intercept System Challenges

Desktop Application (e.g. MATLAB)

Drawing Command (e.g. “draw a triangle”)

(1) complex realistic simulations take time to compute and will cause low frame rates

(2) large scenes, with lots of objects, take time to send over ethernet, leading to low frame rates

Data: User’s hand position, orientation, buttons.

- There are some bottlenecks in our system
Why is Low Frame Rate Bad?

- **Simulator Sickness**
  - Nausea
  - Headache
  - Dizziness

- **Interaction Latency**
  - frustration, confusion
  - undershoot, overshoot and miss target
  - repeat action (multiple button presses) when getting no response
Motivation: Improving commonly used “Virtual Hand” Interaction Technique

- Opportunity: The VR Application runs at a fast frame rate
- What if we could have the VR Application take control for the Virtual Hand?
Our Solution: Intercept Tags

- Specific geometry calls
- Chosen to not appear in scene
- Interpreted, not rendered

MATLAB:
- `draw_cube`
- `draw_cube2`
- `intercept_tag`
- `draw_sphere`
- `intercept_tag`
- `draw_cube3`
- `draw_cube4`

ML2VR:
- `draw_cube`
- `draw_cube2`
- `draw_triangle(0,0..)`
- `draw_sphere`
- `draw_triangle(0,0..)`
- `load V-Hand transform`
- `draw_sphere`
- `unload transform`
- `draw_cube3`
- `draw_cube4`

1. Desktop Application
   - draw_cube
   - draw_sphere
   - intercept driver
   - user presses and holds button

2. Desktop Application
   - draw_cube
   - <intercept tag>
   - draw_sphere
   - <intercept tag>
   - intercept driver

3. Desktop Application
   - draw_cube
   - <intercept tag>
   - draw_sphere
   - <intercept tag>

4. Desktop Application
   - draw_cube
   - draw_sphere
   - intercept driver
   - user releases button
Hand-Off Techniques

(1) Hand-Off Manipulation 
previously discussed

(2) Hand-Off Slice Plane 
smoothly move slice plane which 
affects only certain objects, even 
if desktop is at low frame rate.
Display Techniques

(1) Display Lists
We can specify sections of static (unchanging) geometry, and thus avoid transferring it each frame.

(2) Level Of Detail
We can use the tags to specify multiple representations of the object (with varying polygon-count).
Visual Enhancements

(1) Interpolated Animations

(2) Advanced Shader Insertion

vertex shaded vs per pixel
Hand-Off Manipulation User Study

- place a solid cube completely inside a wireframe cube
- varying sizes of wireframe cube (10%, 20%, 40%)
- within subjects design
  - hand-off manipulation vs original virtual hand
  - \(54 + 54 = 108\) total tasks
User Study Setup

- Six-sided CAVE-type display. Active stereo glasses
- Tracking provided by Intersense IS-900
- MATLAB script with framework/interception by ML2VR
- 14 unpaid subjects
Time Analysis

- Hand-off manipulation always significantly faster
- Significant Interaction effect was found
  - harder the task, better hand-off manipulation
  - easier the task, less benefit found
**Clutch Analysis**

- Hand-off manipulation caused significantly fewer clutches
- Smaller targets required significantly more clutches
- No significant interaction effect
  - Possibly due to small sample size
Questionnaire Analysis

● Usability Questionnaire (SUS)
  “I think that I would like to use this interaction system frequently.”

● Sickness Questionnaire (SSQ)
  “Fatigue, Headache, Sweating, Nausea, Dizziness, Eyestrain”

● Presence Questionnaire (SUS)
  “Rate your sense of being in the room with the cubes.”

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Usability</th>
<th>Sickness</th>
<th>Presence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand-Off</td>
<td><img src="green.png" alt="Green Arrow" /></td>
<td><img src="red.png" alt="Red Arrow" /></td>
<td><img src="green.png" alt="Green Arrow" /></td>
</tr>
<tr>
<td>Traditional</td>
<td><img src="red.png" alt="Red Arrow" /></td>
<td><img src="green.png" alt="Green Arrow" /></td>
<td><img src="red.png" alt="Red Arrow" /></td>
</tr>
</tbody>
</table>
Limitations:

- Techniques apply only to intercept based systems.
- Tags need to enclose geometry. (May not function in higher order desktop applications that re-order geometry calls).
Contributions:
- Proposed the concept of intercept tags
- Proposed ideas for hand-off techniques, display techniques, and visual enhancements
- Implemented the hand-off techniques for use in ML2VR.
- Conducted a user study of the benefits of using intercept tags for hand-off manipulation.
Future Work
(Features implemented in ML2VR)

hand-off techniques

display techniques

visual enhancements
Thank You!

Questions?

djzielin@duke.edu

Support:
National Science Foundation.
IGERT Grant No. DGE-1068871.