System support for protecting visual privacy in augmented reality

Andrei Iosifescu, Computer Science (BS)

Mentor: Dr. Robert LiKamWa, Assistant Professor, School of Arts, Media and Engineering & the School of Electrical, Computer and Energy Engineering
Co-Mentor: Jinhan Hu, School of Arts, Media and Engineering

Research question and motivation

Motivation: As Augmented Reality and Virtual Reality applications are becoming increasingly popular so are the privacy concerns regarding them, specifically in protecting users' visual data whilst using these applications.

Research question: How can users' visual data be protected while they are using Ar/Vr applications?

Proposed Solution

• A security framework, by the name of LensCap.
• Split-access control: the network process and the visual process (figure 1)
• Network Process:
  • User Interface
  • Network and external write permissions
• Visual process:
  • Camera Interface
  • AR Model interface
• Can only communicate with each other using signed and encrypted communication

Sample application

• Sample LensCap applications successfully separated the visual process and the network process

Framework Results

• This network process was a transparent overlay on top of the visual process, which passed user interactions to the visual process (figure 2).
• Only allowed signed encrypted messages are sent to the network process from the visual process.
• LensCap adds minimal latency to applications (Table 1), it is not enough to be noticeable.[2].
• Framerate remains relatively unchanged (Table 2).

Figure 1: The security paradigm enforced by the LensCap Framework.[1]

Table 1: Comparing the average latency times between the touch of a button and the action triggering, with and without LensCap.

<table>
<thead>
<tr>
<th>Application</th>
<th>Original Latency</th>
<th>LensCap Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR-Images</td>
<td>7.43 ms</td>
<td>13.86 ms</td>
</tr>
<tr>
<td>AR-Faces</td>
<td>18.64 ms</td>
<td>32.25 ms</td>
</tr>
<tr>
<td>AR-Text</td>
<td>18.36 ms</td>
<td>33.69 ms</td>
</tr>
</tbody>
</table>

Table 2: Comparing the framerate, with and without LensCap.

<table>
<thead>
<tr>
<th>Application</th>
<th>Original Framerate</th>
<th>LensCap Framerate</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR-Images</td>
<td>34 fps</td>
<td>35 fps</td>
</tr>
<tr>
<td>AR-Faces</td>
<td>34 fps</td>
<td>34 fps</td>
</tr>
<tr>
<td>AR-Text</td>
<td>42 fps</td>
<td>43 fps</td>
</tr>
</tbody>
</table>

Future Work

• Make LensCap more developer friendly and streamlined to use.
• Implement LensCap in various engines and platforms.
• Optimize LensCap performance.

References