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

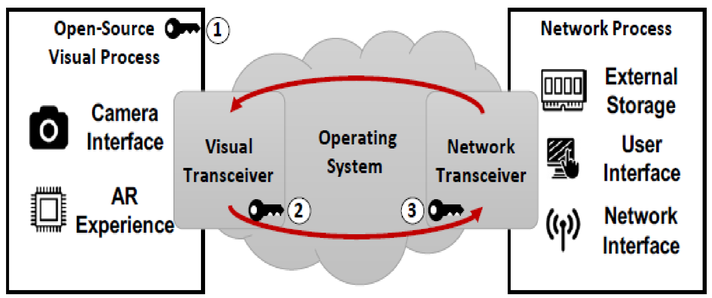

Figure 1:.The security paradigm enforced by the *LensCap* Framework.[1]

Sample application

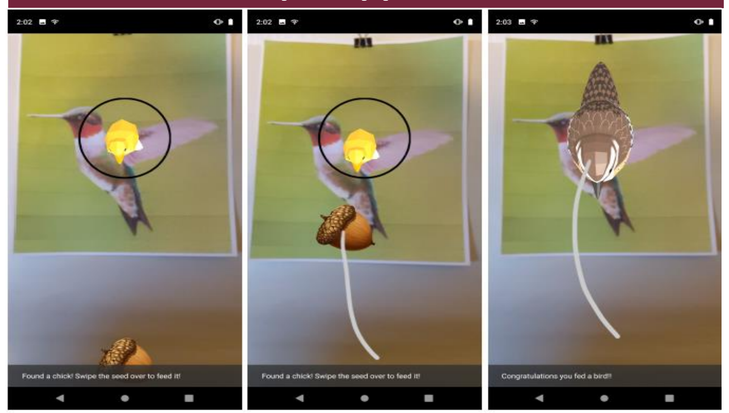

Figure 2: Sample app, that has had *LensCap* added to it. The nut exists in the visual process and it can be thrown by touching the invisible overlay that is the network process.

Framework Results

- Sample *LensCap* applications successfully separated the visual process and the network process
- 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).

Application Performance

Application	Original Latency	LensCap Latency
AR-Images	7.43 ms	13.86 ms
AR-Faces	18.64 ms	32.25 ms
AR-Text	18.36 ms	33.69 ms

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

Application	Original Framerate	LensCap Framerate
AR-Images	34 fps	35 fps
AR-Faces	34 fps	34 fps
AR-Text	42 fps	43 fps

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

Future Work

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

References

[1] Jensen, J., Hu, J., Rahmati, A., & LiKamWa, R. (2019, June). Protecting Visual Information in Augmented Reality from Malicious Application Developers. In The 5th ACM Workshop on Wearable Systems and Applications (pp. 23-28).
[2] Ricardo Jota, Albert Ng, Paul Dietz, and Daniel Wigdor. 2013. How Fast is Fast Enough?: A Study of the Effects of Latency in Direct-touchPointing Tasks. In Proceedings of the SIGCHI Conference on Human
Factors in Computing Systems (CHI '13). ACM, New York, NY, USA, 2291–2300. https://doi.org/10.1145/2470654.2481317

FURI

Ira A. Fulton Schools of Engineering
Arizona State University