Adaptive AR Visual Output Security using Reinforcement Learning Trained Policies

Joseph DeChicchis*, Surin Ahn†, Maria Gorlatova*

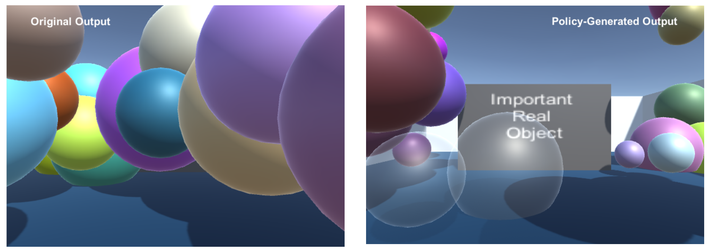
* Duke University, Durham, NC
† Stanford University, Palo Alto, CA

Intelligent AR Output Security

- Current Augmented Reality (AR) is limited in dynamic and adaptive behavior
- Previous work used a simulated environment to investigate the output security application and did not evaluate models on a physical AR device
- Our previous work showed that reinforcement learning trained policies could be applied to output security

Output Security

- Regulating visual content displayed to the user to reduce distraction & obstruction of real-world context
- Prevent holograms from obscuring other holograms with higher priority



Example of visual output before (left) and after (right) a reinforcement learning generated policy is applied

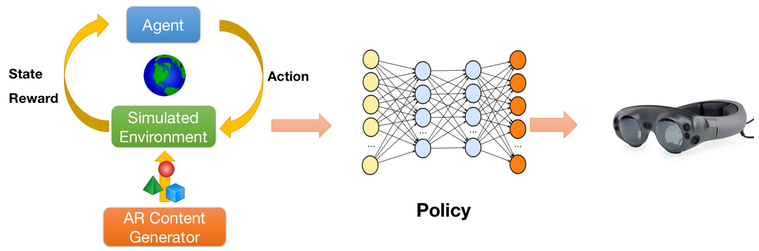
Objectives

1. Investigate the feasibility of running reinforcement learning models on the Magic Leap One AR device
2. Train and evaluate reinforcement learning models for the output security application

Methods

Use Unity to build the training environment → Use ml-agents to train reinforcement learning models → Deploy trained models on a physical device

Output Security Model Development



System overview

Reward Function

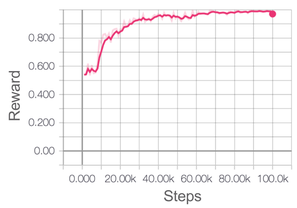
$$R = \alpha \sum_{\substack{o' \in O' \\ r \in R}} \left(Dist_{xy}(r, o') \right) - \beta \sum_{\substack{o \in O \\ o' \in O'}} Dist_{xy}(o, o')$$

R is the set of real-world object positions, O is the set of original hologram positions, and O' is the set of new hologram positions

Adapted from Ahn et al.

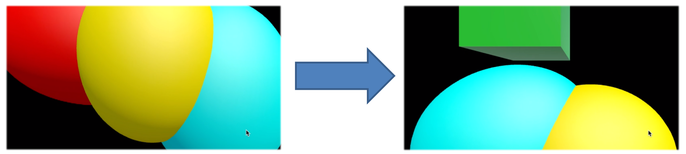
Calculate Direction

- Models converge quickly and move holograms away from real world object
- Requires heuristics to know when to stop moving holograms



Training with one hologram

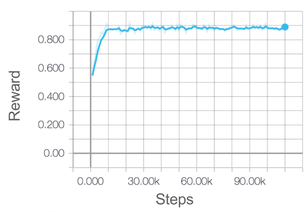
Training with three holograms



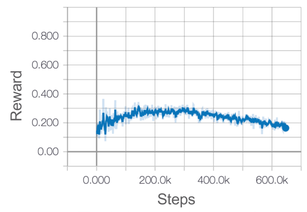
Simulated training environment

Calculate Position

- Models converged when not penalizing for moving hologram away from original position and did not converge otherwise
- Does not require heuristics

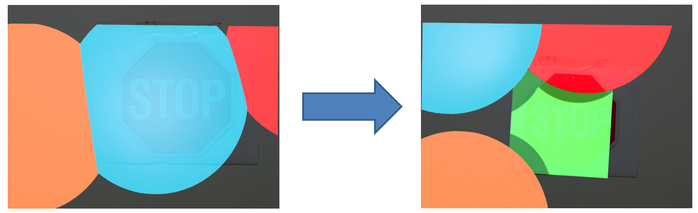


Training without penalty



Training with penalty

Deploying on Magic Leap One



Example visual output before (left) and after (right) the security policy is applied. A green square is placed over the real-world object to indicate that it is tracked.

Successfully deployed on Magic Leap One without any noticeable performance impact

Conclusion

1. Reinforcement learning models can be deployed without noticeable performance degradation
2. An output security model which calculates the direction to move holograms converges quickly
3. An output security model which calculates the position to move holograms requires more fine tuning

Future Work

- Improve the reinforcement learning model and apply it to new scenarios
- Improve the object detection mechanism
- Develop other machine learning and computer vision applications for AR, test its limits and potentially offload computation to edge servers

References

- Surin Ahn, Maria Gorlatova, Parinaz Naghizadeh, Mung Chiang, and Prateek Mittal. 2018. Adaptive Fog-Based Output Security for Augmented Reality. In *Proc. ACM VR/AR Network Workshop'18*. Kiron Lebeck, Kimberly Ruth, Tadayoshi Kohno, and Franziska Roesner. 2017. Securing Augmented Reality Output. In *Proc. IEEE S&P'17*.
- Suman Jana, David Molnar, Alexander Moshchuk, Alan Dunn, Benjamin Livshits, Helen J. Wang, and Eyal Ofek. 2013. Enabling Fine-grained Permissions for Augmented Reality Applications with Recognizers. In *Proc. USENIX Security'13*.
- Franziska Roesner, Tadayoshi Kohno, and David Molnar. 2014. Security and Privacy for Augmented Reality Systems. *Commun. ACM* 57, 4 (April 2014). Unity. ml-agents. https://github.com/Unity-Technologies/ml-agents
- Unity. Unity. https://unity.com/

Acknowledgements: This work is supported in part by the NSF award CSR-1903136 and by the Defense Advanced Research Projects Agency (DARPA) under contract No. HR001117C0048.