UNIVERSITY of WASHINGTON

Undergraduate Research Symposium May 17, 2019 Mary Gates Hall

Online Proceedings

1R

Computer Security, Privacy, Accessibility, and Graphics

Session Moderator: Franziska Roesner, Computer Science and Engineering

JHN 026

12:30 PM to 2:15 PM

* Note: Titles in order of presentation.

Project Sidewalk: A Web-based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data at Scale

Aileen Zeng, Junior, Computer Science
Mary Gates Scholar
Mentor: Jon Froehlich, Allen School of Computer Science

We introduce Project Sidewalk, a new web-based tool that enables online crowdworkers to remotely label pedestrian-related accessibility problems by virtually walking through city streets in Google Street View. To train, engage, and sustain users, we apply basic game design principles such as interactive onboarding, mission-based tasks, and progress dashboards. In an 18-month deployment study, 797 online users contributed 205,385 labels and audited 2,941 miles of Washington DC streets. We compare behavioral and labeling quality differences between paid crowdworkers and volunteers, investigate the effects of label type, label severity, and majority vote on accuracy, and analyze common labeling errors. To complement these findings, we report on an interview study with three key stakeholder groups (N=14) soliciting reactions to our tool and methods. Our findings demonstrate the potential of virtually auditing urban accessibility and highlight tradeoffs between scalability and quality compared to traditional approaches.

Analysis of the Susceptibility of Smart Home Interfaces to End User Error

Mitali Vishwesh Palekar, Senior, Computer Science
UW Honors Program
Mentor: Franziska Roesner, Computer Science and Engineering
Mentor: Earlence Fernandes, CSE

Trigger-action platforms enable end-users to program their smart homes using simple conditional rules of the form: if condition then action. Although these rules are easy to program, subtleties in their interpretation can cause users to make errors that have consequences ranging from incorrect and undesired functionality to security and privacy violations. Based on prior work, we enumerate a set of nine error classes that users can make, and we empirically study the relationship between these classes and the interface design of eight commercially available trigger-action platforms. Particularly, we examine whether each interface prevents (e.g., via good design) or allows each class of error. Based on this analysis, we develop a framework to classify errors and extract insights that lay a foundation for the design of future trigger-action programming interfaces where certain classes of errors can be mitigated by technical means or by alerting the user to the possibility of an error. For instance, we identify that an analysis of a dataset of functionally-similar trigger-action rules could be used to predict whether certain types of error patterns are about to occur. We believe that this work is a first step towards trigger-action interface designs that significantly mitigate user error.

Secure Multi-User Content Sharing for Augmented Reality Applications

Kimberly Christine Ruth, Senior, Computer Engineering, Mathematics
Goldwater Scholar, Mary Gates Scholar, UW Honors Program, Washington Research Foundation Fellow
Mentor: Franziska Roesner, Computer Science and Engineering
Mentor: Tadayoshi Kohno, Computer Science and Engineering

Augmented reality (AR) for the consumer market is gaining momentum and public attention. Besides smartphone platforms, early-stage head-mounted displays such as the Microsoft HoloLens are now publicly available. Many compelling uses of these AR technologies are multi-user: for instance, in-person collaborative tools, multiplayer gaming, and telepresence. Although multi-user AR technologies enable new forms of interaction, they also raise new security

and privacy challenges, not only from untrusted applications but also from other users' malicious or unthinking behavior. It is imperative that these challenges be addressed while the technology is still new and highly malleable. In this work, I explore emerging challenges in securing multi-user AR content sharing from user-to-user threats. I argue that supporting secure and private AR content sharing when users can augment each other's reality requires careful consideration of AR's tight integration with the physical world. I systematize design goals for security and functionality that an AR content sharing module should support, and I design and prototype an application-level module for the HoloLens that meets these goals. By evaluating my module against representative application case studies, I show that it meets desired security and functionality goals flexibly across a range of use cases. I further demonstrate that applications' content sharing needs can be achieved in relatively few lines of code and with low performance overhead. I am currently converting my research prototype into an open-source toolkit so that developers can address these challenges in practice. This work opens up directions for future research on supporting developers in effectively addressing these issues in practice, including making recommendations on how these underlying paradigms should manifest in user interface design. By building foundations for secure multi-user AR content sharing, my work takes steps toward allowing AR to securely reach its full potential.

Privacy Preserving Screening of Personal Documents
Devin Daniel Reich, Junior, Computer Science and Systems
Mary Gates Scholar
Mentor: Martine De Cock, School of Engineering and Technology, UW Tacoma
Mentor: Anderson Nascimento, School of engineering and technology

The ability to derive information through automated scanning of personal documents has significant economic and societal value, stemming from applications in surveillance and digital forensics, e-commerce, tailored advertising, recommender systems, human resource management, mental health care, and more. Giving applications access to one's personal text messages and e-mails can easily lead to (un)intentional privacy violations. We have developed and implemented cryptographic protocols to scan personal documents in a privacy-preserving manner, using techniques from Machine Learning (ML) and Secure Multiparty Computation (SMC). In a typical scenario of interest for our research, there are two parties, nick-named Alice and Bob. Bob has a trained ML model that can automatically classify texts like e-mails, for instance inferring whether the author is depressed, suicidal, a terrorist threat, or whether the e-mail is a spam message. Our SMC based protocols allow for the classification of a personal text written by Alice with Bob's ML model in such a way that Bob does not learn anything about Alice's text (other than the class label resulting from the classification) and Alice does not learn anything about Bob's model. We demo the cryptographic protocols in an application for privacy-preserving detection of hate speech against women and immigrants in text messages, built on top of the SMC framework Lynx developed at UW. In this use case, Bob has a boosted decision tree model that flags texts as hateful based on the occurrence of particular words. We show that Bob can label Alice's texts as hateful or not without learning which words occur in Alice's texts, and Alice does not learn which words are in Bob's hate speech lexicon, no how these words are used in the classification process.

Greedy Face Meshing: An Efficient Meshing Algorithm for Polygon Rendering in Computer Graphics
Ryan Raghav Pachauri, Senior, Computer Science
Mentor: Kevin Zatloukal, Computer Science and Engineering, Allen School

In computer graphics, a voxel (volume element) is a point in a 3D world coordinate system (i.e. the coordinate system of a virtual world). In games like Toca Blocks or Minecraft, voxels are used to store the texture of a particular terrain. Sometimes, voxels next to each other have the same texture. When voxels of homogeneous textures form polygons, rendering systems will optimize memory storage by storing the polygons' vertices rather than every single voxel in the polygon. The process of choosing polygons that cover the voxels is known as meshing. We refer to these polygons as quads and the collection of quads as a mesh. Current methods for polygon meshing require too much data storage or require a drastic change in the mesh after a small change in the world coordinate system. We propose the Greedy Face Meshing (GFM) Algorithm, a linear time algorithm for meshing voxels into quads. We prove that our algorithm is within a constant factor of the optimal solution (in terms of number of quads) and can update in constant time for a single-voxel change in the world coordinate system. We also show how the GFM Algorithm can be implemented using the Segment Tree data structure. Rendering systems can use the GFM algorithm to mesh polygons since its storage is no worse than any existing algorithm and its updates take constant time.

Synthesizing Programs that Generate Plant Graphics
Caleb Hansel (Caleb) Winston, Sophomore, Pre-Sciences
Mentor: Rastislav Bodik, CSE

Within the domains of graphic and video game design, there is often need for tools to quickly develop convincingly realistic models of plants. A common tool applied to this problem is L-systems, a kind of rewriting system that can be used to define rules for iteratively transforming plant models to increasingly fine detail. However, the connection between L-systems and the graphics they generate can sometimes be un-

intuitive. To enable more intuitive development of plant models, we propose a method for generating models of branching structures from simple specifications of a few given iterations of the model. Our approach involves encoding plant models as bracketed L-systems and applying SMT (Satisfiability Modulo Theory) solvers to solve a form of the inverse L-system problem. Iterations of growth in the form of simple vector graphics are compiled to formal constraints for an L-system that can indefinitely generate further growth iterations. The satisfactory system is then found using an SMT solver. This technique allows for branching structures to be conveniently developed by providing meaningful specifications.