As an undergraduate, I worked on projects in parallel processing, digital systems, signal processing, mobile applications, and probability. As a master's student, I worked on dynamic system-level binary translation for x86 virtualization. Several of my projects resulted in papers presented at workshops.

In 2006, I participated in a research project at W3C Semantic Web under Tim Berners-Lee and Daniel Weitzner to create Tabulator, a generic data browser for RDF (Resource Description Framework) data. RDF is a general language for describing graphs whose nodes have global identifier. Because most data can be represented as graphs, RDF is useful for exchanging machine-readable data on the web. The goal of the Tabulator project is to make a browser for traversing data in RDF. While there already existed several semantic web browsers, none of them had the ability to display data in a form that a typical scientific or enterprise users expects. These browsers were also unable to automatically dereference web information by looking up URIs (Uniform Resource Identifier), which is necessary for browsing linked data. To this end, several general "mashup" viewers were implemented to allow users to view and interact with RDF data. As a user browses their current pool of linked data, Tabulator can pull in related data. I implemented a viewer for time-based RDF data, allowing users to view time data as a calendar, schedule or timeline. Since working on Tabulator, the amount of data using semantic web RDF has burgeoned, especially in the field of biology. As a result of my work on Tabulator, I co-authored a paper that appeared in 2006 Semantic Web User Interaction workshop in Athens, Georgia [1].

In December 2006, I worked with Michael Price and Hui-Ying Wen to create a real-time light-saber generator. Input video of a person holding a light-saber handle with embedded inertial sensors is transformed into output video of a person holding a green light-saber. Live video from camera as well as inertial data from accelerometer and gyroscope were processed on an FPGA to create a green light-saber sprite on output XVGA video. While the angle, width, and direction of the light-saber can be deduced from inertial input, the light-saber handle location is detected from camera input so the light-saber is drawn correctly (dead-reckoning for objects that quickly and frequently change direction of movement is hard). I wrote the interfaces to the accelerometer and gyroscope sensors, handled buffering and displaying input and output video, and detecting light-saber handle location [4]. Our project was highly amusing to students working in the Digital Systems Lab.

In 2007, I worked with Alexey Radul and Taylor Campbell on Probabilistic Scheme, an embedding of probabilistic computation in Scheme. While probability theory provides a well-understood mathematical framework for providing multiple sources of information, the computational tools available to programmers for working with probabilistic models is limited. Probabilistic Scheme is a tool for expressing and evaluating probabilistic models, which is important for working with computer vision, natural language processing, bioinformatics and many other fields. Probabilistic Scheme deals with models over arbitrarily structured but discrete and countable probability spaces. The main contributions of Probabilistic Scheme are that it is an embedding of probabilistic computation into a general-purpose programming language, and that it offers anytime approximation with provable upper and lower bounds via restartable partial search. I worked on the implementation of Probabilistic Scheme. Our work on Probabilistic Scheme was presented at 2007 Dynamic Languages Symposium in Montreal, Quebec Canada [5].

In 2008, I worked under the guidance of Hal Abelson with several students on Geolife, a mobile phone application as a submission to Google's Android Developer Challenge.

Geolife is a location-aware to-do list application; unlike typical calendar applications which reminds you when to do something, Geolife also reminds you where to do it. Geolife can remind you to feed your fish when you come home or buy milk when you're by the grocery store. I implemented the interface for selecting locations from a map and tagging them with to-do items. Challenges included developing a user interface for a small screen, power management, and reliability. The project was presented at a public presentation on May 9, 2008 attended by the press, local industry, and technology venture funders [3], where it received exceedingly good reviews.

During mid-2008 to mid-2009, I worked on my master's thesis under the guidance Ole Agesen and Jeffrey Sheldon at VMware and Robert Morris at MIT. My master's thesis was on dynamic binary translation from x86-32 code to x86-64 code for virtualization. The goal of my thesis was to enhance performance of virtual machines and simplify the design of the virtual machine monitor by running 32-bit x86 operating systems in x86-64 mode. This goal was accomplished by translating 32-bit operating system binary code into x86-64 binary code via "widening binary translation"; x86-32 code is "widened" into x86-64 code. As a consequence of my work, VMware's virtual machine monitor was simplified, reducing the human cost of maintaining a complicated virtual machine monitor; the virtual machine monitor code is now 30% smaller. Widening binary translation also improved the performance of 32-bit guest operating systems running in virtual machines and demonstrated the independence of virtual machines from physical hardware. I presented my work at the 2009 International Symposium on Code Generation and Optimization's software and hardware co-design workshop in Seattle, WA [2]. My work resulted in code that will be released next year (2010) to users of VMware's virtualization products.

References

[1] Tim Berners-Lee, Yu hsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lehrer, and David Sheets. Tabulator: Exploring and Analyzing linked data on the Semantic Web. In *Semantic Web User Interaction*, 2006. `http://swui.semanticweb.org/swui06/papers/Berners-Lee/Berners-Lee.pdf`.

[2] Yu hsin Chen. Dynamic Binary Translation from x86-32 code to x86-64 code for Virtualization. In *International Symposium on Code Generation and Optimization (CGO), Workshop on Infrastructures for Software/Hardware co-design (WISH)*, 2009. `http://www.cgo.org/cgo2009/workshops.html#WISH`.

[3] Yu hsin Chen, Evan Gerard, Jeremy Hurwitz, and Yang Xu. Mobile Application Geolife. In *Building mobile applications with Android*, 2008. `http://people.csail.mit.edu/hal/mobile-apps-spring-08/`.

[4] Yu hsin Chen, Michael Price, and Hui Ying Wen. Real-time Light-Saber Generator. In *Introductory Digital Systems Laboratory 6.111*, 2006.

[5] Alexey Radul. Report on the Probabilistic Language Scheme. In *Proceedings of the 2007 symposium on Dynamic Languages.*, 2007. `http://portal.acm.org/citation.cfm?id=1297085`.