Landon Rabern

(805) 403-8185
landon.rabern@gmail.com
https://landon.github.io/

Work History

Data Scientist — 2018 - 2019
Facebook — Boston, MA

Worked with the World.AI team to build accurate population density maps from satellite imagery (Python).

Senior Staff Engineer — 2017 - 2018
IQVIA — Philadelphia, PA

Built general Diagram of Things charting engine with arbitrary depth axis-aligned recursively nested, interactive, animated charts. Chart components and databinding are specified by an XML-based markup language used by a team of 100+ engineers in India to build client specific applications(C#, JavaScript).

CTO, Co-founder — 2008 - 2018
LBD Data — Everywhere

My cofounders and I formed LBD Data to build a suite of enterprise software products in collaboration with Safety Vision, a producer of mobile video solutions for police and public transit. Our products are used throughout the United States. I was the sole developer on a team of 3.

- Designed and developed SafetyView, an application to search and explore large video collections for the purposes of identifying video clips and submitting them into evidence for legal proceedings (C#).
  - UI to search and sort through videos with previews (C#, WinForms, WPF).
  - GPU-accelerated 180° fisheye camera dewarping with support for tilt/pan/zoom.
  - Video player (H264/MPEG4) with support for 16 channels plus integrated UI that presents all relevant metadata including GPS traces, allowing you to scrub through the video based on, e.g. location (libavcodec, OpenStreetMap).
  - Video editor with support for annotation and blurring faces (OpenCV).
  - Shareable, permissioned web links to specific videos (Amazon S3, DynamoDB).
  - HTML5 Mobile App for viewing streams with synchronized multichannel playback.
- Developed SafetyNet, a media server to collect video from a fleet of vehicles and support search and playback in SafetyView. It also monitors and reports on the health of the underlying Camera/DVR hardware (C#).
  - High speed multithreaded video import, including scraping GPS and vehicle metadata from a variety of file formats and storing it in a central database (C#, MS-SQL).
  - Automatic wireless download of video from vehicles with support for resume (FTP/SFTP/SSH/Samba/other custom protocols).
  - Wireless Camera/DVR health analysis (camera functioning/positioning/field of view).
  - Management of software/firmware upgrades for Camera/DVR hardware for the whole fleet.
  - Video search backend for SafetyView (SQL).
  - Streaming video to SafetyView.
- Created LiveLook, a realtime web-based UI for public transit systems that allows centralized monitoring and alerting for fleets of thousands of vehicles, including simultaneous viewing of the video streams from their onboard cameras. (C++ / C# / JavaScript / HTML / CSS)
  - Live playback of multiple IP camera streams. (JavaScript / HTML)
  - Map UI for monitoring vehicle locations (built on top of Google Maps).
  - J1939 implementation (C++) for retrieving vehicle diagnostics, including engine exhaust and tire pressure.

Adjunct Assistant Professor, Mathematics — 2014 - 2017
Franklin & Marshall College — Lancaster, PA

- 6 sections of Calculus, 3 sections of Algebra & Trigonometry.

Senior Software Engineer — 2010 - 2011
Wall Street On Demand — Boulder, CO

Worked on the web engineering team, focused on optimizing middleware supporting hundreds of developers.

- Replaced a hodgepodge of thousands of schema-less data exchange formats with autogenerated data types for clearer, more performant code using compiled expression trees. Reduced the number of bugs due to data parsing and the ramp-up time for new developers significantly. (C#)
- Sped up our data-fetching layer by 10x by optimizing serialization/deserialization, affecting hundreds of different services.
- Doubled the capacity of our business-critical chart generation service by writing a lockless thread-safe memory pool to optimize memory use (C#).

Software Engineer   2009 - 2010
Synaptics   Santa Clara, CA

My work at Synaptics was focused on improving the reliability of the Synaptics TouchPad.

- Debugged and fixed issues in the Synaptics TouchPad (kernel-mode) driver, ranging from blue screens to gesture recognition problems (C++).
- Improved the reliability of gesture recognition by building a playback tool to recreate and fix recognition failures from customer log files (C#).
- Reduced time spent to debug configuration-based issues from hours to minutes by building a tool to verify, optimize, and load driver INF files (C#).

Software Engineer   2007 - 2009
Wall Street On Demand   Boulder, CO

WSOD designs and develops charts, reports, and tools for the financial services industry. Major customers included Charles Schwab, Fidelity, Vanguard, JP Morgan, Scottrade, and eTrade.

- Rewrote the company's entire distributed computing layer in C#. Implemented routing (Random, Round-Robin, etc), load balancing, failover, etc.
- Wrote significant portions of Goldman Sachs' (the parent company) new C# financial plotting tool.
- Built our charting framework, including a re-usable API used by all other developers and the core chart rendering service, using vector graphics. This generates thumbnail charts for web pages as well as creating complete stock and research reports (C#, WPF).

Scientific Programmer   2006 - 2007
L-3 Communications - Applied Technologies Division   Santa Barbara, CA
Security Clearance: Secret

Our group at L-3 created software to model the effects of electromagnetic pulses (EMP) on military systems.

- Wrote a parallel Monte Carlo electromagnetic simulation to compute damage probabilities.
- Wrote a C++ application that provides nuclear effects simulation in real-time wargaming exercises for White Sands Missile Range.
- Wrote a finite difference time domain Maxwell's equations visualizer in C++.
- Developed a C# control encapsulating NASA's WorldWind 3D mapping application. Wrote algorithms to display shaded contours, representing nuclear detonations, on the resulting globe control.

Education

Ph.D., Mathematics 2011 - 2013
Arizona State University
Research: Discrete math, graph coloring, games and algorithms.
Advisor: Hal Kierstead

M.A., Mathematics 2003 - 2005
University of California, Santa Barbara

B.A., Mathematics 1999 - 2003
Washington University in St. Louis
Ross Middlemiss Prize (for top graduating Mathematics major)
Study Abroad in the Netherlands (Utrecht University) 2001 - 2002

Honors & Activities

Erdős number 2 2011
\$80k in grants from the NSA to extend my PhD research 2015 - 2017
1st place, Mentor Graphics State Programming Competition 1997 and 1998
Developed Betsy, a strong chess AI, in C 1998 - 2003
Built Tesla coils and produced massive lightning bolts 1997 - 1999

Research Articles

30^+ publications in top-tier discrete mathematics and philosophy journals. A couple of favorites:

D.W. Cranston and L. Rabern. Planar graphs are $\frac{9}{2}$-colorable *Journal of Combinatorial Theory*, 2017.

This article is about coloring countries on a map so that adjacent countries receive distinct colors. It was conjectured in 1852 that any map could be colored thusly using only 4 colors. This was finally proved in 1976, but the proof is not human-checkable; it requires many hours of computer time to check thousands of cases. Finding a human-checkable proof is still an open problem. To prove that 5 colors suffice is relatively simple. We gave a human-checkable proof that 4.5 colors suffice; this means that we get to use 9 colors, but have to assign each country 2 colors.

- Settled a 20-year old conjecture on the existence of such a proof.
- Featured on Computational Complexity, a popular computer science blog by Lance Fortnow & Bill Gasarch.

B. Rabern and L. Rabern. A simple solution to the hardest logic puzzle ever. *Analysis*, **68**(2), April 2008.

Three gods A, B, and C are called, in no particular order, True, False, and Random. True always speaks truly, False always speaks falsely, but whether Random speaks truly or falsely is a completely random matter. Your task is to determine the identities of A, B, and C by asking three yes-no questions; each question must be put to exactly one god. The gods understand English, but will answer all questions in their own language, in which the words for yes and no are da and ja, in some order. You do not know which word means which.

- Showed how to trivialize the puzzle by asking questions that elicit meaningful answers from Random.
- Showed how to solve the puzzle in only two questions by using paradoxes to explode god-heads.
- This article led to the problem getting a lot of press and many follow-up papers have been written.

Technical Skills

- **Languages and Technologies**
  - C#, C/C++, JavaScript, Go, Python, Solidity, F#, Dart, Java, Pascal, VB.NET, Scheme, Fortran, x86 assembly
  - LINQ, LaTeX, SQL, MVC, HTML, CSS, XML, jQuery, AJAX, XSLT, Bridge.NET, Google protobufs
  - SQL Server, MySql, Sqlite, Amazon DynamoDB, Azure SQL, Entity Framework
  - WinForms, WPF, Silverlight, Xamarin

- **Development Tools and Platforms**
  - .NET Reflector, dotPeek
  - Git, Subversion, CVS
  - Windows, Linux, Mac