RESEARCH INTERESTS
Human-computer interaction, productivity tools for programmers and data scientists, end-user programming, online learning at scale, computing education

CURRENT RESEARCH DIRECTIONS
Two-page Research Statement summarizing my early faculty career work:

Studying why and how people from diverse backgrounds (e.g., older adults, non-native English speakers, end-user programmers) are learning programming and data science.

Building new kinds of interactive tutorial interfaces for learning programming and data science, such as mixed-media interfaces that combine video and text.

Building new kinds of scalable programming environments for novices, such as run-time visualization and real-time peer tutoring built upon my Python Tutor platform, which has been used by over 5 million people in over 180 countries to visualize over 75 million pieces of Python, Java, JavaScript, C, C++, and Ruby code. [NSF CAREER]

ACADEMIC POSITIONS

07/2016 – present
University of California, San Diego (UCSD), La Jolla, CA
Assistant Professor of Cognitive Science
Affiliate Assistant Professor of Computer Science and Engineering
Faculty Affiliate: Design Lab, Halıcıoğlu Data Science Institute

07/2014 – 06/2016
University of Rochester, Rochester, NY
Assistant Professor of Computer Science

06/2013 – 06/2014
Massachusetts Institute of Technology, Cambridge, MA
Visiting Research Scientist – edX (summer 2013)
Postdoctoral Researcher – CSAIL User Interface Design Group – Host: Rob Miller

EDUCATION

09/2006 – 06/2012
Stanford University, Stanford, CA
Ph.D. in Computer Science
Dissertation: Software Tools to Facilitate Research Programming
Advisor: Dawson Engler

06/2005 – 06/2006
Massachusetts Institute of Technology, Cambridge, MA
Master of Engineering in Electrical Engineering and Computer Science
Master’s Thesis: A Scalable Mixed-Level Approach to Dynamic Analysis of C and C++ Programs, Advisor: Michael D. Ernst

09/2001 – 06/2005
Massachusetts Institute of Technology, Cambridge, MA
Bachelor of Science in Electrical Engineering and Computer Science, GPA: 5.0/5.0
AWARDS AND HONORS

10/2019  VL/HCC Best Paper Award [C.47]
07/2019  ISSTA 10-Year Impact Paper Award [C.6]
05/2019  CHI Honorable Mention Paper Award [C.44]
04/2019  NSF CAREER Award
04/2019  Google Faculty Research Award
10/2018  UIST Best Paper Award [C.42]
04/2018  CHI Honorable Mention Paper Award [C.38]
10/2017  UIST Honorable Mention Paper Award [C.36]
05/2017  CHI Honorable Mention Paper Award [C.31]
08/2015  Google Faculty Research Award
04/2014  CHI Honorable Mention Paper Award [C.18]
06/2012  ICSE Software Engineering In Practice Best Paper Award [C.13]
07/2009  ACM SIGSOFT Distinguished Paper Award [C.6]
04/2009  CHI Honorable Mention Paper Award [C.3]
09/2009 – 06/2011 National Science Foundation (NSF) Graduate Fellowship
05/2006  MIT Charles and Jennifer Johnson Thesis Award for Outstanding Computer Science Master of Engineering Thesis

FUNDING

National Science Foundation. CAREER: Interactive Systems for Learning Programming at Scale. $479,860 (2019–2024)

Google Faculty Research Award. Automated IDE Activity Tracing for Creating, Testing, and Updating Developer Tutorials. $65,609 (2019)


National Science Foundation. CRII: CHS: Scaling Up Online Peer Tutoring of Computer Programming. $175,000 + $32,000 REU (undergrad) supplement (2015–2018)

Google Faculty Research Award. Enabling Learners to Create Hierarchical Tutorials from How-To Videos on YouTube. $64,295 (2015)

University of Rochester. University Research Award: Enabling Fast and Scalable Feedback on Writing. $50,000 (2015)

Microsoft Research. Online Python Tutor for Office Mix. $61,308 (2014)
Note that in many areas within computer science and human-computer interaction, conferences (not journals) are the primary publication venues.

I have provided contribution summaries for all papers that I worked on after starting my first faculty position in July 2014, starting with [C.22]. To see the full PDFs of all publications, visit http://pgbovine.net/publications.htm

**Conference Papers**


*Demonstration-based technique that observes users invoking command-line apps and synthesizes custom GUIs for those apps in a computational notebook.*


*Augments the browser with data wrangling, visualization, and API hooks to turn the web into a contextualized prototyping environment for machine learning.*


*Software developers want to learn machine learning for high-level aspirational reasons but often get stuck on low-level code and math issues.*


*End-user programmers are well-positioned to train more diverse populations of future programmers since they may empathize better with novice frustrations.*


*Combines flexibility of live coding with structure of slides to reduce cognitive load and foster Mayer’s multimedia learning principles when teaching programming.*


*Data science instructors want to teach authentic workflows but struggle to keep up with fast-changing software and to find pedagogically-meaningful datasets.*


*Mashup technique for prototyping web apps by extracting UI components from existing webpages and hooking them together using transclusion and glue code.*

Visualization technique that helps tutorial creators overcome expert blind spots by letting them see how novices followed their tutorials and where they struggled.


Students from different countries used Python Tutor’s debugger in ways that may indicate cultural preferences for self- versus instructor-directed learning.


Computer vision technique that extracts source code edits from programming screencast videos and turns existing videos into mixed-media tutorial interfaces.


Meta-analysis of all 69 papers from the first four years of Learning@Scale, which cover diverse learner motivations and interactions with sociotechnical systems.


Conversational programmers [C.28] are frustrated since they want high-level conceptual resources but mostly find low-level coding tutorials from web searches.


Most programming learners worldwide are non-native English speakers; they often struggle to read, write, and talk about English-centric coding resources.


Augments the browser with data wrangling and visualization hooks to turn any webpage into an in-situ computational notebook for prototyping data science.


Program visualization technique that displays entire history of all numeric runtime values in code, which aids debugging, mental models, and tutor dialogue.

*Technique that generates mixed-media (video+text) tutorials for a set of GUI and command-line applications by observing how the user invokes those apps.*


*Lightweight technique to collect programmer feedback using a medical pain scale; user-reported frustrations mirror those found by more heavyweight techniques.*


*Students attending college hackathons value the time-limited, socially-augmented energy and learn coding concepts incidentally, opportunistically, and from peers.*


*(Honorable Mention Paper Award)*

*Many people over age 60 now want to learn programming for mental enrichment and social bonding but get frustrated by cognitive, social, and technical barriers.*


*Reduces cognitive barriers to managing software complexities by integrating version control, testing, and real-time collaboration into a novice-oriented IDE.*


*Women face unique challenges when using and contributing to programming discussion forums and question-and-answer websites such as Stack Overflow.*


*Conversational programmers are professionals who learn to code to communicate better with engineers, not because they need to write code in their jobs.*


*Visualization and multiplexing technique that lets a single programming tutor monitor dozens of students coding in real-time and help several at once via chat.*

Enables remote pair programming and tutoring by extending Python Tutor’s run-time state visualizations with live chat, shared editing, and shared cursors.


Learnersourcing technique that organizes a crowd of novices to create step-by-step worked examples for basic code that can overcome some expert blind spots.


Text-based discussion forums are inefficient for teaching programming concepts; they should be redesigned to be more visual, interactive, and domain-specific.


Identifies an emerging category of college students who aspire to become conversational programmers [C.28] rather than professional or end-user programmers.


Digital textbooks complement both classroom and MOOC-based lessons by being a more self-paced, non-linear, and interactive instructional format.


(Honorable Mention Paper Award)


**INVITED PUBLICATIONS**

**MAGAZINE ARTICLES**


INVITED TALKS


- Learning Programming at Scale. Coursera, Berkeley Institute for Data Science, South Park Commons, Google, May 2018; Caltech CMS Department, April 2018.

- The Design Space of Tools for Learning Programming at Scale. UCSD Design at Large Seminar, October 2016.


- Interactive Systems for Learning Programming at Scale. Stanford Human-Computer Interaction Seminar, Stanford, CA, Feb 2016; Recurse Center (née Hacker School), New York, NY, Jan 2016; University of Maryland, College Park, MD, Dec 2015.


- Programming On Demand: Wrangling, Iterating, and Opportunistic Learning. (faculty candidate job talk, all in CS or EECS departments) University of Utah, North Carolina State University, Dartmouth College, University of San Francisco, Oregon State University, Northeastern University, University of Rochester, Washington University in St. Louis, Feb–Mar 2013.


---

**SERVICE**

**Program Committee Member**

• Blocks and Beyond Workshop 2019 (located at VL/HCC 2019)
• UIST 2019 (ACM Symposium on User Interface Software and Technology)
• L@S 2019 (ACM Conference on Learning at Scale)
• CHI 2019 (ACM Conference on Human Factors in Computing Systems)
• UIST 2018 (ACM Symposium on User Interface Software and Technology)
• VL/HCC 2018 (Symposium on Visual Languages & Human-Centric Computing)
• L@S 2018 (ACM Conference on Learning at Scale)
• EDM 2018 (International Conference on Educational Data Mining)
• LIVE 2018 (Workshop on Live Programming, located at SPLASH 2018)
• ICER 2017 review committee (Int’l Computing Education Research Conference)
• LIVE 2017 (Workshop on Live Programming, located at SPLASH 2017)
• EDM 2017 (International Conference on Educational Data Mining)
• VISSOFT 2016 (IEEE Working Conference on Software Visualization)
• EDM 2016 (International Conference on Educational Data Mining)
• L@S 2016 (ACM Conference on Learning at Scale)
• L@S 2015 (ACM Conference on Learning at Scale)
• PLOOC 2015 (Workshop on Programming Languages Technology for MOOCs)
• CHESE 2015 (Int’l Code Hunt Workshop on Educational Software Engineering)
• SPLASH-E 2015 (Systems, Programming, Languages and Applications: Software for Humanity – Education Symposium)
• PLATEAU 2012 (Workshop on Evaluation and Usability of Programming Languages and Tools)
• TaPP 2012 (Workshop on the Theory and Practice of Provenance)

**External Paper Reviewer**


**Steering Committee**

VL/HCC (Symposium on Visual Languages & Human-Centric Computing), elected to a four-year term (2019–2023)

**Grant Reviewer**


**Artifact Eval. Committee**

ESEC/FSE 2011 (Symposium on the Foundations of Software Engineering)
TEACHING

Instructor

- UCSD COGS 120/CSE 170: Human-Computer Interaction Design (Fall 2016, 2017, 2018)
- UCSD COGS 124: Human-Computer Interaction Technical Systems Research (Fall 2017, 2018)
- UCSD COGS 127: Data-Driven UX/Product Design (Fall 2019)
- UCSD CSE 219/COGS 229/DSGN 119: Design at Large seminar series (Fall 2017)
- UCSD COGS 121: Human-Computer Interaction Programming Studio (Spring 2017, 2018, 2019)
- UCSD COGS 231: Graduate Seminar on Human-Centered Programming (Spring 2017, 2018, 2019)
- University of Rochester CSC 210: Principles of Web Application Development (Fall 2014, 2015)
- University of Rochester CSC 253: Dynamic Languages and Software Development (Fall 2014)
- MIT 6.813: User Interface Design & Implementation, co-taught as a postdoc with PI Rob Miller and others (Spring 2014)

Graduate Teaching Assistant

- Stanford CS343 – Advanced Topics in Compilers (Spring 2012)
- Stanford CS242 – Programming Languages (Autumn 2009)
- Stanford CS243 – Advanced Compiling Techniques (Winter 2008)
- MIT 6.170 – Laboratory in Software Engineering (Spring 2006)

Undergraduate Laboratory Assistant

- MIT 6.170 – Laboratory in Software Engineering (Fall 2004)
- MIT 6.111 – Introductory Digital Systems Laboratory (Fall 2004)
- MIT 6.001 – Structure and Interpretation of Computer Programs (Spring 2002)

RESEARCH STUDENTS SUPERVISED

Ph.D.

- Ian Drosos [C.33]
- Logan Gittelson
- Sean Kross [C.39,C.44,C.46]
- Sam Lau
- Jaime Montoya
- Xiong Zhang [C.36,C.43,C.48]

Masters

- Davide Berdin (visiting student from Uppsala University, Sweden)
- Charles (Hsien-che) Chen [C.45] – undergrad+masters, first position: NVIDIA
- Hyeonsu Kang [C.35] – first position: research engineer at MIT, then Ph.D. student at CMU Human-Computer Interaction Institute
- Kandarp Khandwala [C.40] – first position: MathWorks
- Alok Mysore [C.34,C.42] – 2017 UCSD CSE masters student research award winner, first position: Yelp
- Dan Scarafoni – undergrad+masters, first position: MIT Lincoln Laboratory, then Ph.D. student in Machine Learning at Georgia Tech
- Priyan Vaithilingam [C.49]
- April Wang [C.38] – masters at Simon Fraser University, member of thesis committee, first position: Ph.D. student at University of Michigan School of Information
- Jeffery White [C.24,C.26]
• Renan Zanelatto [C.24,C.26]

Undergraduate
• Karina Banda
• Lenny Brown
• Irene Chen – first position: Google
• Jennifer (Kate) Godziecki
• Dan Hassin
• Sara Lickers
• Emy Lin – first position: Intel
• Douglas Miller – first position: Jump Trading
• Anvisha Pai [P.1] – first position: Dropbox
• Annie Zhang – first position: Bank of America
• Joyce Zhu [C.24] – 2015 CRA Outstanding Undergraduate Researcher Award honorable mention, first position: Quip

Ph.D. Committee Member
Erin Brady, Anna Loparev, Phyo Thiha, Eric Seidel, Tricia Ngoon, Adam Rule, Ailie Fraser, Benjamin Cosman, Mario Alvarez, Amy Rae Fox, Yan Chen, Toby (Jia-Jun) Li, Sander Valstar

Outreach@Scale
My personal website http://pgbovine.net contains over 700 articles, videos, and podcast episodes on topics ranging from research to education to scientific outreach; it receives over 750,000 page views per year. I have also recorded over 600 videos on research, education, and scientific outreach for my YouTube channel, which has over 6,000 subscribers and 700,000+ total video views.

OTHER EMPLOYMENT

07/2018 – 09/2018 Google, Mountain View, CA (worked remotely)
Visiting Researcher, People + AI Research (PAIR) group, Host: Carrie Cai

07/2015 – 08/2015 Microsoft Research, Redmond, WA
Visiting Researcher, Research in Software Engineering (RiSE) group, Host: Rishabh Singh

07/2012 – 02/2013 Google, Mountain View, CA
Software Engineer – online education group – Google Research

09/2006 – 06/2012 Stanford University, Stanford, CA
Ph.D. Student – Department of Computer Science

09/2011 – 01/2012 Harvard University, Cambridge, MA
Visiting Research Fellow, Computer Systems Group, Host: Margo Seltzer

06/2011 – 09/2011 Google, Mountain View, CA
Software Engineering Intern – refined and deployed CDE

06/2009 – 09/2009 Microsoft Research, Redmond, WA
Research Intern, Research in Software Engineering (RiSE) group, Host: Thomas Zimmermann

06/2007 – 09/2007 Google, Mountain View, CA
Software Engineering Intern – prototyped memory allocators for C and C++ programs
01/2004 – 06/2006 Massachusetts Institute of Technology, Cambridge, MA
Research Assistant – Program Analysis Group – Advisor: Michael D. Ernst
Undergraduate and master’s research on tools for analyzing C and C++ programs

09/2003 – 01/2004 Massachusetts Institute of Technology, Cambridge, MA
Research Assistant – Computer Graphics Group – Advisor: Fredo Durand
Developed an HDR (high dynamic range) image editing tool for photographers

06/2004 – 08/2004, Teradyne, Agoura Hills, CA
06/2003 – 08/2003 Software Engineering Intern – wrote simulators for semiconductor test hardware

09/2002 – 06/2003 Massachusetts Institute of Technology, Cambridge, MA
Research Assistant – Teacher Education Program – Advisor: Eric Klopfer
Developed a suite of 5 educational games for Palm OS devices

06/2002 – 08/2002 Codehost, Culver City, CA
Software Engineering Intern – wrote embedded Linux tablet PC software