

# COMMUNICATIONS

CACM.ACM.ORG

# OF THE ACM

03/2015 VOL.58 NO.03



## Local Laplacian Filters

## Edge-Aware Image Processing with a Laplacian Pyramid

## Python for Beginners

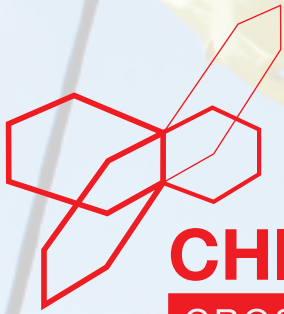
## The Real Software Crisis

## Who Owns IT?

## Privacy Implications of Health Information Seeking on the Web

```
end  
L{end} = G{end}; % residual not affected  
R = reconstruct_laplacian_pyramid(L); % collapse result Laplacian pyramid
```





**CHI 2015**  
**CROSSINGS**  
SEOUL • KOREA

18-23 APRIL 2015

## REGISTER NOW

Come to CHI 2015, the premier international forum for human-computer interaction (HCI). This year's program has more technical content than ever before with breakthrough insights on the impact of human-centered innovation. The conference will take place in the COEX at the heart of Seoul's Gangnam district. Being held in Asia allows this year's conference to showcase groundbreaking content from across the region with special symposia for Chinese, Japanese and ASEAN insights and innovations. You will also hear fresh perspectives from exciting Asian speakers along with special presentations from speakers of diverse backgrounds and viewpoints.

Discounted early registration rates are available until 6 Mar 2015.

Late registration continues until 17 Apr 2015.

See website for registration and up-to-date details on the entire technical program:



**LOU YONGQI**

Dean of the College of Design and Innovation ,  
Tongji University,  
Shanghai, China



**DONGHOON  
CHANG**

Executive Vice  
President, UX Center,  
Samsung Electronics,  
Korea



**DAVID MIN**

Senior Research  
Fellow, Software  
Research Center, LG  
Electronics, Korea



**PSY**

Singer, songwriter,  
record producer and  
view-count breaker



ACM Books



MORGAN & CLAYPOOL  
PUBLISHERS

# Publish your next book in the ACM Digital Library

ACM Books is a new series of advanced level books for the computer science community, published by ACM in collaboration with Morgan & Claypool Publishers.

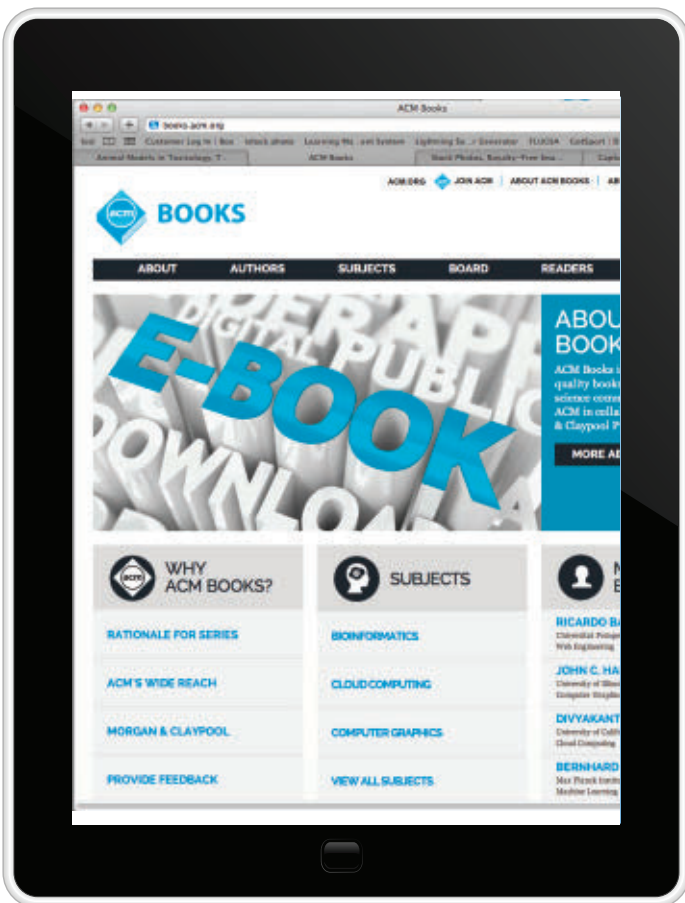
*I'm pleased that ACM Books is directed by a volunteer organization headed by a dynamic, informed, energetic, visionary Editor-in-Chief (Tamer Özsu), working closely with a forward-looking publisher (Morgan and Claypool).*

—Richard Snodgrass, University of Arizona

[books.acm.org](http://books.acm.org)

## ACM Books

- ◆ will include books from across the entire spectrum of computer science subject matter and will appeal to computing practitioners, researchers, educators, and students.
- ◆ will publish graduate level texts; research monographs/overviews of established and emerging fields; practitioner-level professional books; and books devoted to the history and social impact of computing.
- ◆ will be quickly and attractively published as ebooks and print volumes at affordable prices, and widely distributed in both print and digital formats through booksellers and to libraries and individual ACM members via the ACM Digital Library platform.
- ◆ is led by EIC M. Tamer Özsu, University of Waterloo, and a distinguished editorial board representing most areas of CS.



**Proposals and inquiries welcome!**

Contact: **M. Tamer Özsu**, Editor in Chief  
[booksubmissions@acm.org](mailto:booksubmissions@acm.org)



Association for  
Computing Machinery

*Advancing Computing as a Science & Profession*

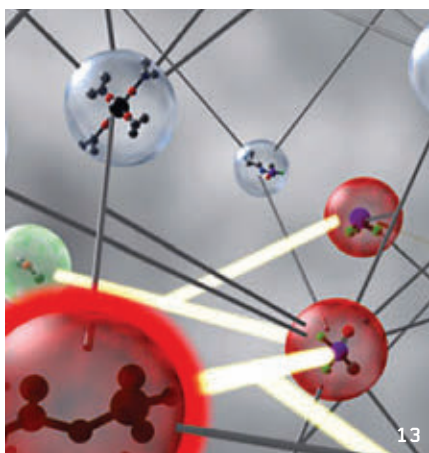
## Departments

- 5 **Letter from ACM's Director of Information Systems**  
**Raising ACM's Digital Library**  
*By Wayne Graves*
- 
- 6 **Letters to the Editor**  
**Make Abstracts**  
**Communicate Results**
- 
- 7 **Cerf's Up**  
**'As We May Think'**  
*By Vinton G. Cerf*
- 
- 8 **Blog@ACM**  
**Advice on Teaching CS, and the Learnability of Programming Languages**  
Valerie Barr considers how attitude can impact teacher effectiveness, while Mark Guzdial suggests the ultimate focus in teaching programming languages should be on usability.
- 
- 33 **Calendar**
- 
- 92 **Careers**

## Last Byte

- 96 **Q&A**  
**Object Lessons**  
The creator of the Eiffel programming language discusses his career in industry and academia, "Design by Contract," and his views on Agile software development.  
*By Leah Hoffmann*

## News



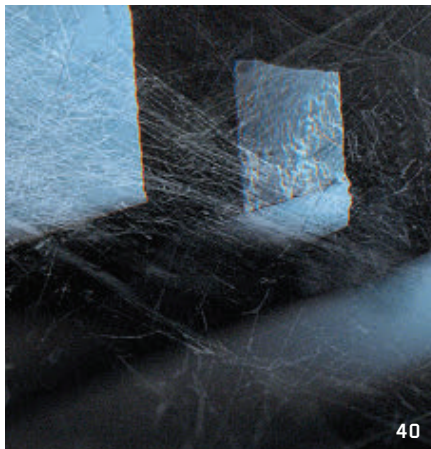
- 13 **Automating Organic Synthesis**  
A machine that could create organic molecules on demand awaits appropriate software and analytical components.  
*By Keith Kirkpatrick*
- 
- 16 **Car Talk**  
Vehicle-to-vehicle communication is coming. Are we ready for it?  
*By Tom Geller*
- 
- 19 **Python for Beginners**  
A survey found the language in use in introductory programming classes in the top U.S. computer science schools.  
*By Esther Shein*

## Viewpoints

- 22 **Legally Speaking**  
**Copyrightability of Java APIs Revisited**  
A recent case challenges the long-standing view that application program interfaces are not protectable under copyright law.  
*By Pamela Samuelson*
- 
- 25 **Broadening Participation**  
**Reaching a Broader Population of Students through "Unplugged" Activities**  
Introducing children to fundamental computing concepts through Computer Science Unplugged.  
*By Thomas J. Cortina*
- 
- 28 **The Profession of IT**  
**A Technician Shortage**  
In our elation about rising CS enrollments, we are overlooking a growing shortage of computing technicians. Our education system is not responding to this need.  
*By Peter J. Denning and Edward E. Gordon*
- 
- 31 **Computing Ethics**  
**Humans in Computing: Growing Responsibilities for Researchers**  
Considering the role of institutional review boards in computing research.  
*By John Leslie King*
- 
- 34 **Viewpoint**  
**The Real Software Crisis: Repeatability as a Core Value**  
Sharing experiences running artifact evaluation committees for five major conferences.  
*By Shriram Krishnamurthi and Jan Vitek*
- 
- 37 **Viewpoint**  
**Why Did Computer Science Make a Hero Out of Turing?**  
Comparing the legacy of Alan Turing in computer science with that of Carl Friedrich Gauss in mathematics.  
*By Maarten Bullynck, Edgar G. Daylight, and Liesbeth De Mol*



Practice



40

- 40 **HTTP/2.0 — The IETF Is Phoning It In**  
Bad protocol, bad politics.  
*by Poul-Henning Kamp*

- 43 **META II: Digital Vellum in the Digital Scriptorium**  
Revisiting Schorre's 1962 compiler-compiler.  
*By Dave Long*

**Q** Articles' development led by [acmqueue.queue.acm.org](http://acmqueue.queue.acm.org)

Contributed Articles



50

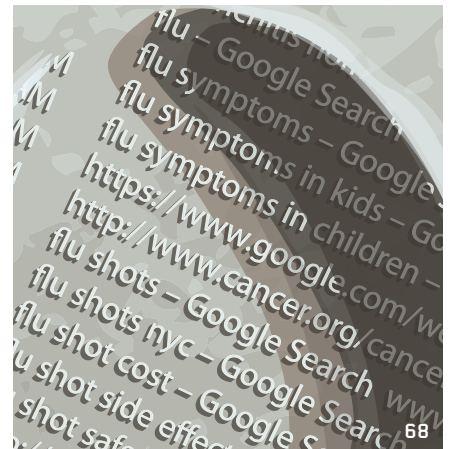
- 50 **Who Owns IT?**  
What was once centralized or federated technology governance is increasingly participatory.  
*By Stephen J. Andriole*

- 58 **Designing Statistical Privacy for Your Data**  
Preparing data for public release requires significant attention to fundamental principles of privacy.  
*By Ashwin Machanavajjhala and Daniel Kifer*



**About the Cover:**  
This month's cover story (p. 81) depicts state-of-the-art edge-aware image processing using simple and flexible Laplacian filters, illustrating sometimes the best approach leads back to basics. Cover design by Andrij Borys Associates, photo by Zora Avagyan.

Review Articles



68

- 68 **Privacy Implications of Health Information Seeking on the Web**  
A revealing picture of how personal health information searches become the property of private corporations.  
*By Timothy Libert*

Research Highlights

- 80 **Technical Perspective**  
**Image Processing Goes Back to Basics**  
*By Edward Adelson*

- 81 **Local Laplacian Filters: Edge-Aware Image Processing with a Laplacian Pyramid**  
*By Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz*



Watch the authors discuss this work in this exclusive *Communications* video.



ACM, the world's largest educational and scientific computing society, delivers resources that advance computing as a science and profession. ACM provides the computing field's premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources.

**Executive Director and CEO**

- John White
- Deputy Executive Director and COO**  
Patricia Ryan
- Director, Office of Information Systems**  
Wayne Graves
- Director, Office of Financial Services**  
Darren Ramdin
- Director, Office of SIG Services**  
Donna Cappel
- Director, Office of Publications**  
Bernard Rous
- Director, Office of Group Publishing**  
Scott E. Delman

**ACM COUNCIL**

- President**  
Alexander L. Wolf
- Vice-President**  
Vicki L. Hanson
- Secretary/Treasurer**  
Erik Altman
- Past President**  
Vinton G. Cerf
- Chair, SGB Board**  
Patrick Madden
- Co-Chairs, Publications Board**  
Jack Davidson and Joseph Konstan
- Members-at-Large**  
Eric Allman; Ricardo Baeza-Yates;  
Cherri Pancake; Radia Perlman;  
Mary Lou Soffa; Eugene Spafford;  
Per Stenström
- SGB Council Representatives**  
Paul Beame; Barbara Boucher Owens;  
Andrew Sears

**BOARD CHAIRS**

- Education Board**  
Mehran Sahami and Jane Chu Prey
- Practitioners Board**  
George Neville-Neil

**REGIONAL COUNCIL CHAIRS**

- ACM Europe Council**  
Fabrizio Gagliardi
- ACM India Council**  
Srinivas Padmanabhuni
- ACM China Council**  
Jiaquan Sun

**PUBLICATIONS BOARD**

- Co-Chairs**  
Jack Davidson; Joseph Konstan
- Board Members**  
Ronald F. Boisvert; Marie-Paule Cani;  
Nikil Dutt; Roch Guerrin; Carol Hutchins;  
Patrick Madden; Catherine McGeoch;  
M. Tamer Ozsu; Mary Lou Soffa

**ACM U.S. Public Policy Office**

Renee Dopplick, Director  
1828 L Street, N.W., Suite 800  
Washington, DC 20036 USA  
T (202) 659-9711; F (202) 667-1066

**Computer Science Teachers Association**  
Lissa Clayborn, Acting Executive Director

# COMMUNICATIONS OF THE ACM

Trusted insights for computing's leading professionals.

*Communications of the ACM* is the leading monthly print and online magazine for the computing and information technology fields. *Communications* is recognized as the most trusted and knowledgeable source of industry information for today's computing professional. *Communications* brings its readership in-depth coverage of emerging areas of computer science, new trends in information technology, and practical applications. Industry leaders use *Communications* as a platform to present and debate various technology implications, public policies, engineering challenges, and market trends. The prestige and unmatched reputation that *Communications of the ACM* enjoys today is built upon a 50-year commitment to high-quality editorial content and a steadfast dedication to advancing the arts, sciences, and applications of information technology.

**STAFF**

**DIRECTOR OF GROUP PUBLISHING**  
Scott E. Delman  
cacm-publisher@cacm.acm.org

- Executive Editor**  
Diane Crawford
- Managing Editor**  
Thomas E. Lambert
- Senior Editor**  
Andrew Rosenbloom
- Senior Editor/News**  
Larry Fisher
- Web Editor**  
David Roman
- Rights and Permissions**  
Deborah Cotton

- Art Director**  
Andrij Borys
- Associate Art Director**  
Margaret Gray
- Assistant Art Director**  
Mia Angelica Balaquiot
- Designer**  
Iwona Usakiewicz
- Production Manager**  
Lynn D'Addesio
- Director of Media Sales**  
Jennifer Ruzicka
- Public Relations Coordinator**  
Virginia Gold
- Publications Assistant**  
Juliet Chance

- Columnists**  
David Anderson; Phillip G. Armour;  
Michael Cusumano; Peter J. Denning;  
Mark Guzdial; Thomas Haigh;  
Leah Hoffmann; Mari Sako;  
Pamela Samuelson; Marshall Van Alstyne

**CONTACT POINTS**

- Copyright permission**  
permissions@cacm.acm.org
- Calendar items**  
calendar@cacm.acm.org
- Change of address**  
acmhelp@acm.org
- Letters to the Editor**  
letters@cacm.acm.org

**WEBSITE**  
<http://cacm.acm.org>

**AUTHOR GUIDELINES**  
<http://cacm.acm.org/>

**ACM ADVERTISING DEPARTMENT**

2 Penn Plaza, Suite 701, New York, NY 10121-0701  
T (212) 626-0686  
F (212) 869-0481

**Director of Media Sales**  
Jennifer Ruzicka  
jen.ruzicka@hq.acm.org

**Media Kit** [acmm mediasales@acm.org](mailto:acmm mediasales@acm.org)

**Association for Computing Machinery (ACM)**  
2 Penn Plaza, Suite 701  
New York, NY 10121-0701 USA  
T (212) 869-7440; F (212) 869-0481

**EDITORIAL BOARD**

**EDITOR-IN-CHIEF**  
Moshe Y. Vardi  
eic@cacm.acm.org

- NEWS**
- Co-Chairs**  
William Pulletyblank and Marc Snir
- Board Members**  
Mei Kobayashi; Kurt Mehlforn;  
Michael Mitzenmacher; Rajeev Rastogi

- VIEWPOINTS**
- Co-Chairs**  
Tim Finin; Susanne E. Hambrusch;  
John Leslie King
- Board Members**  
William Aspray; Stefan Bechtold;  
Michael L. Best; Judith Bishop;  
Stuart I. Feldman; Peter Freeman;  
Mark Guzdial; Rachelle Hollander;  
Richard Ladner; Carl Landwehr;  
Carlos Jose Pereira de Lucena;  
Beng Chin Ooi; Loren Terveen;  
Marshall Van Alstyne; Jeannette Wing

- PRACTICE**
- Co-Chairs**  
Stephen Bourne
- Board Members**  
Eric Allman; Charles Beeler; Bryan Cantrill;  
Terry Coatta; Stuart Feldman; Benjamin Friedl;  
Pat Hanrahan; Tom Limoncelli;  
Kate Matsudaira; Marshall Kirk McKusick;  
Erik Meijer; George Neville-Neil;  
Theo Schlossnagle; Jim Waldo

The Practice section of the CACM Editorial Board also serves as the Editorial Board of [queue](http://queue.acm.org).

**CONTRIBUTED ARTICLES**

- Co-Chairs**  
Al Aho and Andrew Chien
- Board Members**  
William Aiello; Robert Austin; Elisa Bertino;  
Gilles Brassard; Kim Bruce; Alan Bundy;  
Peter Buneman; Peter Druschel;  
Carlo Ghezzi; Carl Gutwin; Gal A. Kaminka;  
James Larus; Igor Markov; Gail C. Murphy;  
Shree Nayar; Bernhard Nebel;  
Lionel M. Ni; Kenton O'Hara;  
Sriram Rajamani; Marie-Christine Rousset;  
Avi Rubin; Krishan Sabnani;  
Ron Shamir; Yoav Shoham; Larry Snyder;  
Michael Vitale; Wolfgang Wahlster;  
Hannes Werthner; Reinhard Wilhelm

**RESEARCH HIGHLIGHTS**

- Co-Chairs**  
Azer Bestavros and Gregory Morrisett
- Board Members**  
Martin Abadi; Amr El Abbadi; Sanjeev Arora;  
Dan Boneh; Andrei Broder; Doug Burger;  
Stuart K. Card; Jeff Chase; Jon Crowcroft;  
Sandhya Dwaekadas; Matt Dwyer;  
Alon Halevy; Maurice Herlihy; Norm Jouppi;  
Andrew B. Kahng; Henry Kautz; Xavier Leroy;  
Kobbi Nissim; Mendel Rosenblum;  
David Salesin; Steve Seitz; Guy Steele, Jr.;  
David Wagner; Margaret H. Wright

**WEB Chair**  
James Landay

**Board Members**  
Marti Hearst; Jason I. Hong;  
Jeff Johnson; Wendy E. MacKay

**ACM Copyright Notice**

Copyright © 2015 by Association for Computing Machinery, Inc. (ACM). Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or fee. Request permission to publish from [permissions@acm.org](mailto:permissions@acm.org) or fax (212) 869-0481.

For other copying of articles that carry a code at the bottom of the first or last page or screen display, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center; [www.copyright.com](http://www.copyright.com).

**Subscriptions**

An annual subscription cost is included in ACM member dues of \$99 (\$40 of which is allocated to a subscription to *Communications*); for students, cost is included in \$42 dues (\$20 of which is allocated to a *Communications* subscription). A nonmember annual subscription is \$100.

**ACM Media Advertising Policy**

*Communications of the ACM* and other ACM Media publications accept advertising in both print and electronic formats. All advertising in ACM Media publications is at the discretion of ACM and is intended to provide financial support for the various activities and services for ACM members. Current Advertising Rates can be found by visiting <http://www.acm-media.org> or by contacting ACM Media Sales at (212) 626-0686.

**Single Copies**

Single copies of *Communications of the ACM* are available for purchase. Please contact [acmhelp@acm.org](mailto:acmhelp@acm.org).

**COMMUNICATIONS OF THE ACM**

(ISSN 0001-0782) is published monthly by ACM Media, 2 Penn Plaza, Suite 701, New York, NY 10121-0701. Periodicals postage paid at New York, NY 10001, and other mailing offices.

**POSTMASTER**

Please send address changes to *Communications of the ACM*  
2 Penn Plaza, Suite 701  
New York, NY 10121-0701 USA

Printed in the U.S.A.



Association for Computing Machinery



DOI:10.1145/2728169

Wayne Graves

# Raising ACM's Digital Library

The ACM Digital Library has been around for quite some time; in fact this year it will turn 17. This makes the ACM Digital Library the youngest of my three children. The other

two are 22 and 19. Over these past 20 years, all three have given me reason to lose sleep and to be proud. I believe they are all on a solid foundation. The core concepts and values are in place and they have reached a level of maturity that can be built upon to accomplish great things as they all move into their 20s.

The ACM DL ([dl.acm.org](http://dl.acm.org)) is a collection of publications serving the needs of approximately five million users worldwide. High-quality ideas, concepts, and views across the breadth of the computing space have been published by ACM for over 60 years. Making critical content discoverable and accessible has been the primary goal of the ACM DL since its conception. Expanding the scope of the ACM DL beyond what ACM publishes to include fully integrated bibliographic data of all computing literature has proven to be an extremely important part of that primary goal.

While the ability to find and access information remains critical, it has become simply an expectation rather than a service. Digital Immigrants<sup>a</sup> can remember the days when this was very exciting, and I will admit that I am still actually excited by this, but our community has certainly changed. The ACM DL is extremely useful to the Digital Immigrants and to the Digital Natives,<sup>b</sup> but

we can now build upon “useful” and explore a new vision.


What was thought of as a library, although in fairness this term may already be somewhat foreign, will move to a space in which independent, distributed, concurrent, and parallel interaction becomes a possibility. The space will contain people, datasets, software, simulations, publications, and more. The set of services layered within this space will provide for interaction rather than simply access. The users of the space will become a natural part of the rich resources. The ACM DL will be a destination where presentation and collaboration allow for relationships to form, extending the boundaries of the past and envisioning the future.

**The ACM DL will be a destination where presentation and collaboration allow for relationships to form, extending the boundaries of the past and envisioning the future.**

## “It takes a village ...”

I would like to engage with the community to help solidify some next steps and some grand ideas. How can the maturation of the ACM DL best fit into your life? What data is important to you and what functionality would you expect or like to see built around it? What kind of interaction would you like to have with fellow consumers or providers? The input I am looking for is absolutely not limited by the few questions here. In my experience dealing with and being a part of this community I have no doubt the strong opinions and creative thinking will come through. I would also appreciate any advice I can pass along to my kids. As with the ACM DL thinking, things were much easier when the problems were “don’t touch that, it’s *hot*.”

In the ACM DL, you will see a “Feedback” link on the right side of any page. This will allow you to comment generally or on a specific aspect of the existing interface. You can also email me: [graves@acm.org](mailto:graves@acm.org).

ACM has always been led by a very dedicated set of visionary volunteers representing the computing community. Interacting with these volunteers is an extremely rewarding aspect of my role as ACM’s Director of Information Systems. Taking these next steps with the community is directly in line with their vision of the ACM DL as well as for the organization itself. 

**Wayne Graves** ([graves@acm.org](mailto:graves@acm.org)) is Director of Information Systems at ACM’s headquarters in New York City.

© 2015 ACM 0001-0782/15/03 \$15.00

a Coined by Marc Prensky in 2001, “Digital Immigrants” refers to those who were not born into a digital world but have come to adopt new technologies.

b Prensky refers to those born in the digital generation of computers, Internet, video games, and social media as “Digital Natives.”

# Make Abstracts Communicate Results

**H**ERMANN MAURER'S VIEWPOINT "Does the Internet Make Us Stupid?" (Jan. 2015) made an important point in its first two sentences, that no one reads full papers anymore. Taking it to heart, ACM should make its publications communicate more effectively by insisting abstracts include a summary of results and key concepts, communicating important information even when readers skip or skim the rest. Maurer also said Internet technology is to blame for superficial reading habits, but I recall when as a graduate student I stopped reading full articles in *AAAS Science*, focusing on just the abstracts to decide whether or not to continue. That was the late 1970s, well before the rise of the Internet, so I cannot blame the Internet. I could, however, rely on abstracts in *Science* because they were so good at summarizing results.

As my focus turned to engineering, I stopped reading *Science* and concentrated on IEEE and ACM publications. However, many engineering papers, and ACM-sponsored conference proceedings and journals in particular, in-

clude notably unenlightening abstracts. It is as if the authors intend to hide their results to force the reader to read the entire paper or at least skip to the final subhead, usually something like "Conclusions and Future Work," to discover whether the results are even interesting. The goal, it appears, is not so much to transmit information but force the reader to appreciate the author's pages of laboriously composed prose.

Newspaper editors and reporters structure their work better in this regard, with meaningful titles conveying the core concept, followed by a one-paragraph introductory summary, or lede, followed by the full narrative. This is perhaps due to professional editorial oversight, whereby a dispassionate editor is able to demand an author concede the major points up front, knowing the reader is unlikely to endure the whole article.

I thus call on authors of scientific papers and articles to follow suit, making sure to include abstracts that summarize results and key concepts. I likewise call on editors, program chairs, and reviewers to enforce this requirement. Good abstracts improve the communication bandwidth of the printed word. It is good engineering and only fair to the reader.

**Steve Trimberger**, Incline Village, NV

## Skip Grand Visions for Software Development

Software development (or engineering if you prefer a more highbrow term) is still a nascent field, as discussed by Ivar Jacobson and Ed Seidewitz in their article "A New Software Engineering" (Dec. 2014). The rapid evolution and progress developers have made over my 25 years in the field remains one of its key attractions. I am never bored. And there is no sign this rapid pace will slow anytime soon. The idea of "a new software engineering" simply does not square with the state of the art as practiced in software organizations all around us. Teams within Google, Microsoft, Amazon, Twitter, and countless other organizations have little

time or patience for grand visions of "how we should do our jobs," as in Software Engineering Method and Theory, or SEMAT, the related Essence kernel and language, or whatever the current fashion is called.

I have worked with many teams in successful and less-successful organizations over the years. The winners are pragmatists, carefully picking and choosing the practices that satisfy their needs. They are rooted in computer science, knowing and using essential data structures, algorithms, and the rest. They are minimalists, minimizing the code they write while constantly proving it through logic and performance tests. They want to get work done, knowing what "done" means.

I know that many people are embarrassed we are not a mature engineering discipline like other, older fields. I gave up on this idea long ago, recognizing software development for what it is—dynamic, growing, in need of improvement, but not anywhere near the point of making grand pronouncements like SEMAT.

**Dean Wampler**, Chicago, IL

## Authors' Response:

*Essence is exactly about practitioners picking and choosing their own practices in order to "get the work done." Despite progress in software engineering, we still seem to reinvent the wheel a lot. Pragmatists often take a long road of hard knocks before finding practices that work for them. Rather than a "methodology to end all methodologies," we propose a common ground to disseminate the community experience needed by practitioners, not from "embarrassment" with traditional software engineering but in hope for better—a hope shared by Google, which even held a three-day workshop on Essence.*

**Ivar Jacobson and Ed Seidewitz**,  
Verbier, Switzerland

*Communications* welcomes your opinion. To submit a Letter to the Editor, please limit yourself to 500 words or less, and send to [letters@cacm.acm.org](mailto:letters@cacm.acm.org).

© 2015 ACM 0001-0782/15/03 \$15.00

**Sketch-Thru Plan: A Multimodal Interface for Command and Control**

**Who Builds a House Without Drawing Blueprints?**

**Use of Formal Methods at Amazon Web Services**

**Security Challenges in Medical Devices**

**Go Static or Go Home**

Plus the latest news about designing medical molecules, current side-channel attacks, and the growth of dynamic pricing.



# 'As We May Think'

I HOPE READERS will forgive me for plagiarizing Vannevar Bush's famous essay title<sup>a</sup> that appeared 70 years ago in the pages of *Atlantic Monthly*. The title is so apt, however, that I dare to use it. Two items arrived in my inbox recently, one is the Winter 2015 edition of the *Journal of the American Academy of Arts and Sciences*, called *Daedalus*, and the other is the recent book by Peter J. Denning, a former president of ACM and editor of *Communications*, and Craig H. Martell titled *Great Principles of Computing*<sup>b</sup> and together, they provoked this essay.

The *Daedalus* issue is focused on neuroscience and spans topics from perception, the role of sleep, consciousness and much else in addition. The *Great Principles* book digs deeply into fundamental principles underlying what we call computer science. Not surprisingly, they deal with some overlapping notions. The one that caught my immediate attention in *Daedalus* is titled "Working Memory Capacity: Limits on the Bandwidth of Cognition." As I read this, I immediately thought of Denning's early work on the Working Set concept: programs had a natural span of memory requirement that had to be met in real memory to avoid thrashing in the implementation of virtual memory space. Apparently the human brain has a working set limitation. In vision, it appears to be roughly two objects in visual space per brain hemisphere. That is, four total, but limited to two supported by the visual cortex in each hemisphere. More than that, and our human ability to recall and process questions about what we have seen diminishes, rather like the thrashing that happens when we do not have sufficient real memory to support the working set needed in virtual space. Denning and Martell address this under their "Principle of Locality."

a Vannevar Bush, "As We May Think;" <http://theatlntc/1ahQVW2>

b P.J. Denning and C.H. Martell. *Great Principles of Computing*. MIT Press, Cambridge, MA, USA, 2015, ISBN 978-0-262-52712-5

Despite the wonders of the human brain, which we are far from understanding, it does not appear to have a convenient way to grow processing capacity while we can achieve that objective with our artificial computers by adding memory or adding processors. This is not to say that adding more conventional computing devices necessarily produces an increase in effective computing, for particular computing tasks. One has only to remember Fred Brooks' *Mythical Man Month*<sup>c</sup> to recall this also applies to programming and the rate at which "finished" code can be produced. Adding more programmers does not necessarily produce more or better code. It might even produce worse code for lack of coordination and Brooks actually draws attention to this phenomenon.

Still, there appears to be a growing sense that computing may in fact benefit from adopting unconventional computational paradigms. The so-called neural chips, such as IBM's TrueNorth,<sup>d</sup> are indicative of this interesting trend as is the rapidly evolving exploration of quantum computing. Adding more state by adding more neural nodes and interconnections seems to improve the scope and accuracy of pattern recognition for example. One then begins to wonder whether there might be utility in combining neural computing with conventional computing to achieve something that neither might be very good at alone. This reminds me of work done by my thesis advisor, Gerald Estrin, in the mid-1950s and early 1960s on what he called "Fixed plus Variable" computing.<sup>e</sup> In these designs, a general-purpose computer was combined with a variable structure computer that, like today's Field Programmable Gate Arrays (FPGAs), could be adapted to special purpose computations. As I understood Estrin's work, this idea also extended to combining analog and digital computation in

c [http://en.wikipedia.org/wiki/The\\_Mythical\\_Man-Month](http://en.wikipedia.org/wiki/The_Mythical_Man-Month)

d <http://www.research.ibm.com/articles/brainchip.shtml>

which the former achieved approximate solutions that could be refined further by digital methods.

These ideas lead me to wonder whether, in the context of today's very flexible and scalable cloud computing environment, one might find ways to harness a variety of computing methods, including neural networks, conventional scalar and vector processing, graphical processors and perhaps even analog or quantum processing to solve various special sorts of problems. Assuming for a moment that any of this actually makes any sense, one is struck by the challenge of organizing the aggregate computation so that the results are reproducible, the appropriate intermediate results reach the right next computing step, and there is an ability to expand and contract the computing element requirements to match need might be preserved.

I hope readers who are far more experienced than I am in the design of complex computing systems may take time to voice their opinions about these ideas. In their book, Denning and Martell dive deeply into the importance of design in all aspects of computing. Without adherence to serious and deep design principles and attention to systems engineering, the usability and utility of computing systems of all kinds suffers. The Internet design adopted a variety of tactics including layering, information hiding and loose coupling, to achieve a scalable and evolvable system. There were only 400 computers on the Internet in 1983 and today there are billions of them. Design and systems engineering should have priority places in the curriculum of computer science. They are the beginning of everything and, in some sense, the end as well. ■

e G. Estrin and C.R. Viswanathan. Organization of a 'fixed-plus-variable' structure computer for computation of eigenvalues and eigenvectors of real symmetric matrices. *JACM* 9, 1 (Jan. 1962).

Vinton G. Cerf is vice president and Chief Internet Evangelist at Google. He served as ACM president from 2012-2014.

Copyright held by author.

The *Communications* Web site, <http://cacm.acm.org>, features more than a dozen bloggers in the BLOG@CACM community. In each issue of *Communications*, we'll publish selected posts or excerpts.

twitter

Follow us on Twitter at <http://twitter.com/blogCACM>

DOI:10.1145/2716345

<http://cacm.acm.org/blogs/blog-cacm>

## Advice on Teaching CS, and the Learnability of Programming Languages

*Valerie Barr considers how attitude can impact teacher effectiveness, while Mark Guzdial suggests the ultimate focus in teaching programming languages should be on usability.*



**Valerie Barr**  
**“Some Thoughts For  
Computer Science  
Teaching Assistants  
(and Faculty)”**

<http://bit.ly/1yBDMbb>

January 4, 2015

I was recently contacted by a student who will be working as a teaching assistant for an upper-level CS course. He remembered that, during a visit to his campus, I had said that the way courses are taught is just as, if not more, important than the race, gender, ethnic background of the instructor. His question was whether I had any advice for him and his fellow TAs about how they could “conduct ourselves or talk about the course material in order to help foster as inclusive and welcoming an environment as possible.”

Here is the advice I gave, which I think can be helpful for faculty as well as for TAs, lab assistants, student help desk workers, and others.

1. The most important thing is tone and attitude when answering questions. Many of us in CS (I include myself in this) have to fight against lapsing into a very patronizing tone. It is easy to slip into the “I cannot believe you don’t understand that” tone of voice, or say things like “it is so obvious.” Every time we use that tone of voice, or those sorts of phrases, we send the “if you don’t get it already, you shouldn’t be here” message. While this is likely more of a problem in lower-level courses, it can happen in upper-level courses as well. An interesting exercise for TAs and lab assistants would be to sit down and collect a list of the sorts of things they have heard over the years (or even said) and then actually say those things in a patronizing tone of voice. That should serve as a good reminder of what it sounds like, hopefully making it easier to avoid talking to students in that tone.

2. If students are working in teams, keep an eye out/ear out for problematic

dynamics amongst the team members. Suggest to teams ways of rotating responsibilities so that everyone has the opportunity to play a leadership role.

3. The electronic submission deadline for homework should be set so that it does not encourage macho behavior. Have homework due no later than 11:00 P.M. or midnight. The problem with a 6:00 A.M. or 9:00 A.M. deadline is that it encourages very macho “I stayed up all night, I’m hardcore, I’m a \*real\* programmer” behavior. People who do not handle their work that way are seen as not being a true hacker, as not having real “programming chops,” even if the person who did not stay up was incredibly organized, completely solved the homework problems, and turned in the assignment hours early!

4. It is fine to suggest the use of Github or others, but understand that not everyone wants to put their work out in the world in that way. If it is required of all students, fine, but do not encourage the use of public repositories as a way of measuring people’s abilities. I am always distressed when I hear that employers are grouping résumés based on what they find in code repositories—this automatically knocks out of the running anyone who does not choose to display their work in that way, and we know that women tend not to use repositories as much because of the level of attack they experience in that world.

5. Periodically look over the grades as a group and see if any implicit bias is in evidence. Are grade distributions roughly the same across all groups of students? Check in about who is com-

ing for help, how much time you spend with people. Are there some people who never come for help, but whose performance indicates they should? You cannot exactly go up to people and say, “what are we doing wrong, why don’t you come for help?” but consider using a quick survey partway through the term to ask students for feedback on the TAs/lab assistants/student helpers as a group.

6. Do not make a big deal about how you are doing so much and working so hard to be inclusive. Bragging about how sensitive you are will backfire and can make people feel singled out. Do it quietly, and trust that people are likely to learn more and enjoy the class more if they are feeling comfortable in the educational setting you have helped create.

I am sure this is not a comprehensive list. What do you do in your classes? What can you add to my list? Post your ideas to the comments. Thanks!



**Mark Guzdial**  
**“Programming Languages Are the Most Powerful, and Least Usable and Learnable User Interfaces”**

<http://bit.ly/1g39mAX>  
 March 27, 2014

Andy Ko wrote a recent blog post (<http://bit.ly/1iVxF3A>) with an important claim: “Programming languages are the least usable, but most powerful human-computer interfaces ever invented.” Ko argues the “powerful” part with points about expressiveness and political power. He uses HCI design heuristics to show how programming languages have poor usability. Obviously, some people can *use* programming languages, but too few people and at great effort.

I see that his argument extends to *learnability*. There are two ways in which programming languages have poor learnability today—(1) in terms of expectancy-value and (2) in terms of social cost.

**What is the benefit of a closure?** Eugene Wallingford tweeted a great quote the other day:

*Educational psychologists measure the cognitive load (http://bit.ly/1ISmGOf) of instruction, which is the effort that a*

*student makes to learn from instruction. Every computer scientist can list a bunch of things which were really hard to learn, and maybe could not even be imagined to start, like closures, recursion in your first course, list comprehensions in Python, and the type systems in Haskell or Scala.*  
<http://bit.ly/1BMu8QD>

Expectancy-value theory (<http://bit.ly/1sctSGD>) describes how individuals balance out the value they expect to get from their actions. Educational psychologists talk about how that expectation motivates learning (<http://edcate.co/1iefV80>). Students ask themselves, “Can I learn this?” and “Do I *want* to learn this? Is it *worth* it?” You do not pursue a degree in music if you do not believe you have musical ability. Even if you love art history, you might not get a degree in it if you do not think it will pay off in a career. Most of us do not learn Dvorak keyboards (<http://bit.ly/1jvaFNC>), even though they are provably better than Qwerty, because the *perceived* costs just are not worth the *perceived* benefit. The actual costs and benefits do not really play a role here—perception drives motivation to learn.

If you cannot imagine closures, why would you want to learn them? If our programming languages have inscrutable features (that is, high cognitive load to learn them) with indeterminate benefits, why go to the effort? That is low learnability. If students are not convinced they can learn it and they are not convinced of the value, then they do not learn it.

**The social cost of going in a new direction.** I was at a workshop on CS Education recently where a learning scientist talked about a study of physicists who did their programming in Fortran-like languages and only used arrays for all their data structures. Computer scientists in the room saw this as a challenge. How do we get these physicists to learn a better language with a better design, maybe object-oriented or functional? How do we get them to use better data structures? Then one of the other learning scientists asked, “How do we *know* that our way is *better*? Consider the possibility that we’re *wrong*.”

We computer scientists are always happy to argue about the value of one programming paradigm over another. But if we think about it from Andy

Ko’s usability perspective, we need to think about it for specific users and uses. How do we know that we can make life better for these Fortran-using physicists?

What if we convinced some group of these Fortran-using physicists to move to a new language with a new paradigm? Languages do not get used in a vacuum—they get used in a community. We have now cut our target physicists off from the rest of their community. They cannot share code. They cannot use others’ libraries, tools, and procedures. The costs of learning a new language (with new libraries, procedures, and tools) would likely reduce productivity enormously. Maybe productivity would be greater later. Maybe. The value is uncertain and in the future, but the cost is high and immediate.

Maybe we should focus on students entering the Fortran-using physics community, and convince them to learn the new languages. Learning scientists talk about student motivation to join a “community of practice” (<http://bit.ly/1kDIhFJ>). Our hypothetical physics student wants to join *that* community. They are learning to value what the community values. Trying to teach them a new language is saying: “Here, use this—it’s way better than what the people you admire use.” The student response is obvious, “Why should I believe *you*? How do *you* know it’s *better*, if it’s not what my community uses?”

**Solution: Focus on usability.** Communities change, and people learn. Even Fortran-using physicists change how they do what they do. The point is that we cannot impose change from the outside, especially when value is uncertain.

The answer to improving both usability and learnability of programming languages is in another HCI dictum: “*Know thy users, for they are not you.*” We improve the usability and learnability of our programming languages by working with our users, figuring out what they want to do, and help them to do it. Then the value is clear, and the communities will adopt what they see as valuable.

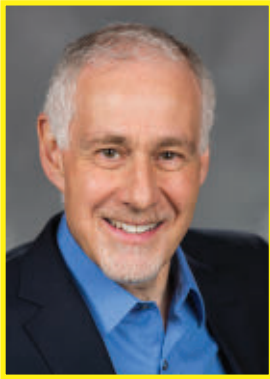
---

Valerie Barr is a professor at Union College, and chair of ACM-W, ACM’s Committee on Women in Computing. Mark Guzdial is a professor at the Georgia Institute of Technology

© 2015 ACM 0001-0782/15/03 \$15.00

# ACM

## ON A MISSION TO SOLVE TOMORROW.



Dear Colleague,

Computing professionals like you are driving innovations and transforming technology across continents, changing the way we live and work. We applaud your success.

We believe in constantly redefining what computing can and should do, as online social networks actively reshape relationships among community stakeholders. We keep inventing to push computing technology forward in this rapidly evolving environment.

For over 50 years, ACM has helped computing professionals to be their most creative, connect to peers, and see what's next. We are creating a climate in which fresh ideas are generated and put into play.

Enhance your professional career with these exclusive ACM Member benefits:

- Subscription to ACM's flagship publication ***Communications of the ACM***
- Online books, courses, and webinars through the **ACM Learning Center**
- Local Chapters, Special Interest Groups, and conferences all over the world
- Savings on peer-driven specialty magazines and research journals
- The opportunity to subscribe to the **ACM Digital Library**, the world's largest and most respected computing resource

We're more than computational theorists, database engineers, UX mavens, coders and developers. Be a part of the dynamic changes that are transforming our world. Join ACM and dare to be the best computing professional you can be. Help us shape the future of computing.

Sincerely,

A handwritten signature in black ink, appearing to read 'Alexander Wolf', written over a horizontal line.

Alexander Wolf  
President  
Association for Computing Machinery



Association for  
Computing Machinery

Advancing Computing as a Science & Profession

# SHAPE THE FUTURE OF COMPUTING. JOIN ACM TODAY.

ACM is the world's largest computing society, offering benefits and resources that can advance your career and enrich your knowledge. We dare to be the best we can be, believing what we do is a force for good, and in joining together to shape the future of computing.

## SELECT ONE MEMBERSHIP OPTION

### ACM PROFESSIONAL MEMBERSHIP:

- Professional Membership: \$99 USD
- Professional Membership plus ACM Digital Library: \$198 USD (\$99 dues + \$99 DL)
- ACM Digital Library: \$99 USD (must be an ACM member)

### ACM STUDENT MEMBERSHIP:

- Student Membership: \$19 USD
- Student Membership plus ACM Digital Library: \$42 USD
- Student Membership plus Print *CACM* Magazine: \$42 USD
- Student Membership with ACM Digital Library plus Print *CACM* Magazine: \$62 USD

- Join ACM-W:** ACM-W supports, celebrates, and advocates internationally for the full engagement of women in all aspects of the computing field. Available at no additional cost.

Priority Code: CAPP

### Payment Information

\_\_\_\_\_  
Name

\_\_\_\_\_  
ACM Member #

\_\_\_\_\_  
Mailing Address

\_\_\_\_\_  
City/State/Province

\_\_\_\_\_  
ZIP/Postal Code/Country

\_\_\_\_\_  
Email

Payment must accompany application. If paying by check or money order, make payable to ACM, Inc., in U.S. dollars or equivalent in foreign currency.

- AMEX    VISA/MasterCard    Check/money order

\_\_\_\_\_  
Total Amount Due

\_\_\_\_\_  
Credit Card #

\_\_\_\_\_  
Exp. Date

\_\_\_\_\_  
Signature

### Purposes of ACM

ACM is dedicated to:

- 1) Advancing the art, science, engineering, and application of information technology
- 2) Fostering the open interchange of information to serve both professionals and the public
- 3) Promoting the highest professional and ethics standards

Return completed application to:  
ACM General Post Office  
P.O. Box 30777  
New York, NY 10087-0777

Prices include surface delivery charge. Expedited Air Service, which is a partial air freight delivery service, is available outside North America. Contact ACM for more information.

**Satisfaction Guaranteed!**

## BE CREATIVE. STAY CONNECTED. KEEP INVENTING.

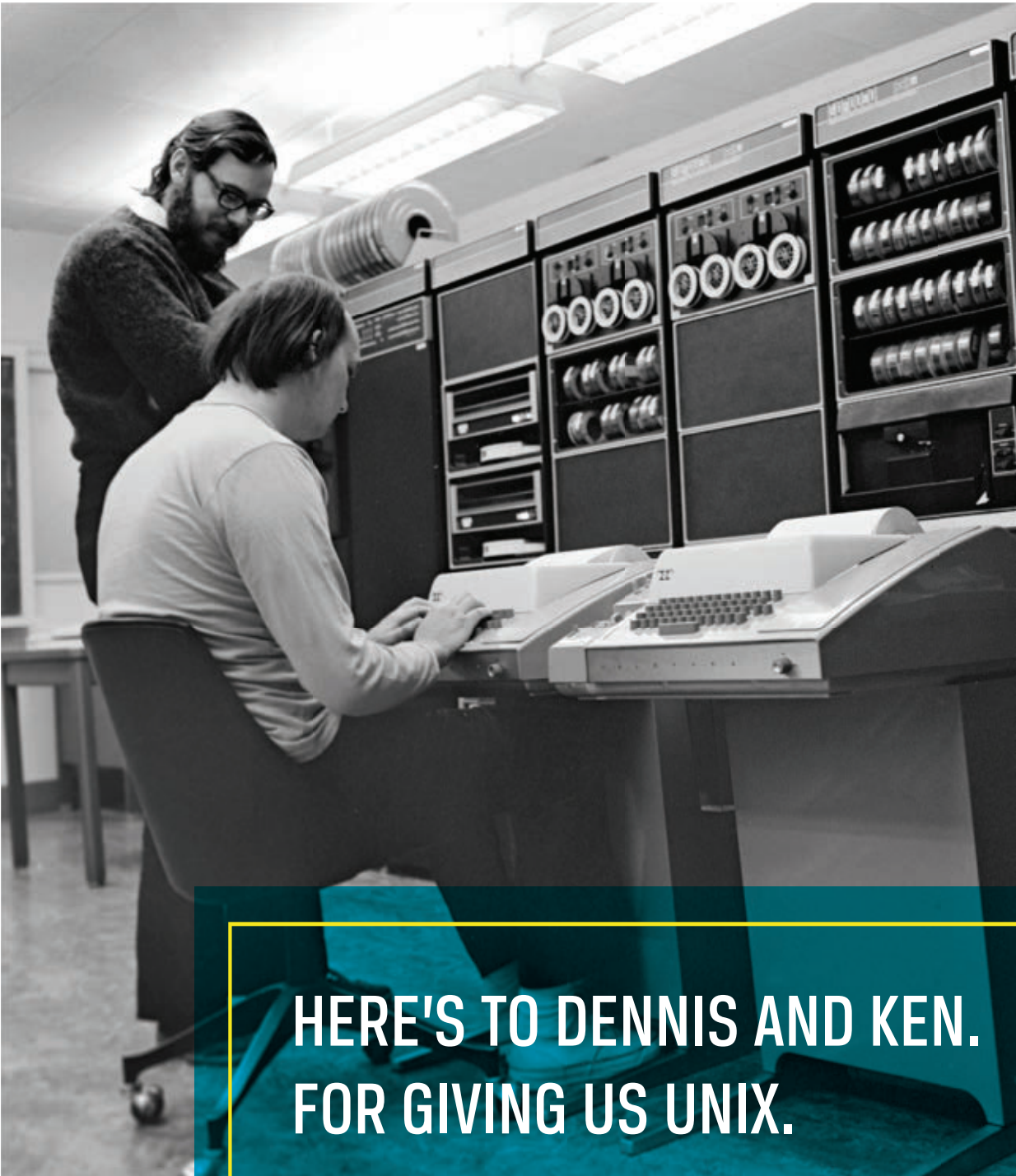


Association for  
Computing Machinery

1-800-342-6626 (US & Canada)  
1-212-626-0500 (Global)

Hours: 8:30AM - 4:30PM (US EST)  
Fax: 212-944-1318

acmhelp@acm.org  
acm.org/join/CAPP



HERE'S TO DENNIS AND KEN.  
FOR GIVING US UNIX.

We're more than computational theorists, database managers, UX mavens, coders and developers. We're on a mission to solve tomorrow. ACM gives us the resources, the access and the tools to invent the future. Join ACM today and receive 25% off your first year of membership.

**BE CREATIVE. STAY CONNECTED. KEEP INVENTING.**

[ACM.org/KeepInventing](https://www.acm.org/KeepInventing)



Association for  
Computing Machinery

# Automating Organic Synthesis

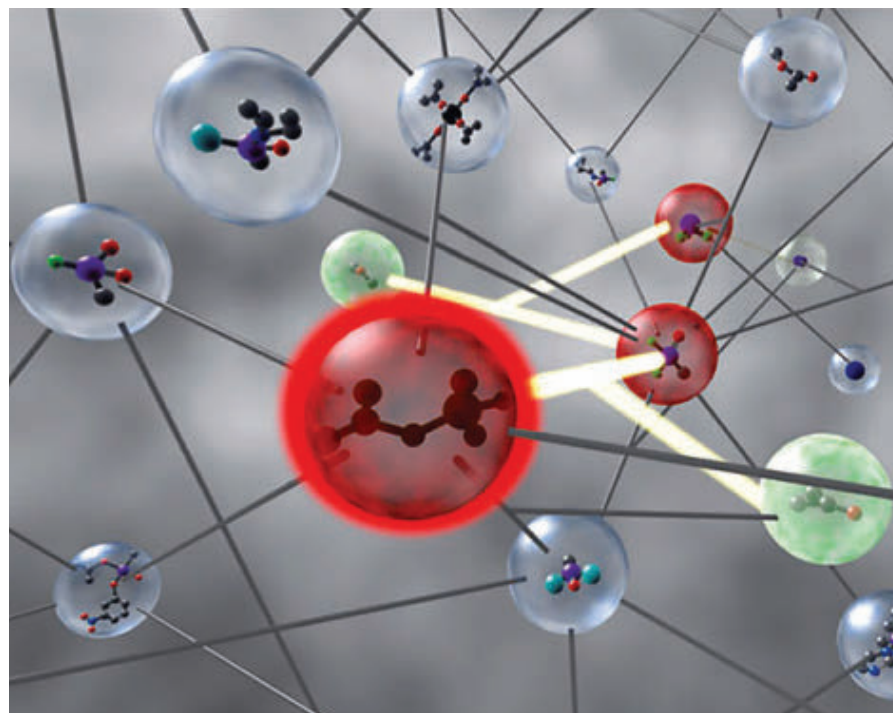
*A machine that could create organic molecules on demand awaits appropriate software and analytical components.*

**T**HE IMAGE OF a chemist slaving away in a lab, haphazardly pouring steaming test tubes of multi-colored liquid into bubbling beakers amid stacks of leather-bound reference books has long been relegated to old Hollywood films or TV shows. However, while today's organic chemists generally spend as much or more time planning their work in advance, thinking and laying out the sequence of reactions that will be required to make a specific molecule, they still largely mix, filter, and combine substances by hand to try to recreate those planned sequences.

The advent of the modern computer and software packages capable of collecting, categorizing, and recombining vast amounts of chemical properties and reaction data may one day help to automate the process of creating molecules. Described as an organic synthesis machine, it would be able to make a huge number of small molecules on demand, speeding the development of new chemical research and of end products across a wide range of industries.

## Organic Chemistry

The process often used to conduct organic synthesis is a technique called



retrosynthetic analysis. Chemists draw a completed molecule and then deconstruct it, erasing the chemical bonds that would be easy to form, while leaving fragments of molecule that are stable or readily available. The chemist then tries to identify the new raw materials needed to connect the missing

pieces of the molecule, based on their experience and expertise. Then, the chemist must actually manually combine the raw material in the lab to synthesize the new molecule.

A few of the challenges involved with conducting organic synthesis in this manner are apparent. First, the hu-

man brain is relatively limited in terms of the number of molecular structures and rules it can quickly recall without needing to refer to a database or reference sources. Similarly, it takes significant time and effort to physically perform a synthesis in the lab, and real-world synthesis results often do not match the theoretical plan.

As such, chemists have increasingly turned to online databases of chemical compounds, reactions, and rules that can be used when trying to construct molecules. Commercial molecular data bases such as SciFinder, an electronic interface to the American Chemical Society's Chemical Abstracts Service, or Reaxys, a commercial database service offered by Elsevier, can provide reference data that can be used as a jumping-off point for the creation of new molecules. And, these data repositories may be the content that helps power the organic synthesis machine of the future.

### Viewing Molecules

One of the visionaries in the space that believes such a machine can and will be built is Richard Whitby, a chemist at the University of Southampton, in the U.K. Whitby is the leader of Dial-A-Molecule, a collaborative project that is working to identify the technical and research requirements to build such a machine.

The key vision of the Dial-A-Molecule project is largely designed around the development of a machine that can quickly develop any molecule, based on a specific set of desired properties.

**“By making the delivery of a new molecule as quick and easy as it is now to order a ‘stock’ chemical, we aim to remove that bottleneck in development.”**

“Every molecule has different properties, and the rate-limiting step in finding the ‘best’ for a particular application is making them,” Whitby says. “The key component is deciding how to put the molecule together. Even for a simple molecule, there are a vast number of possible routes, each comprising many steps.”

However, the organic chemistry community is currently “very poor at judging how well even individual steps will work” even if a reliable route is chosen, Whitby laments. Using a machine that is able to reference a large database of reaction and raw material data, and then automatically synthesize a molecule, could drastically increase the speed, breadth, and depth of chemical creation.

Currently, scientists are often forced to use the best readily available molecule, as manually constructing hun-

dreds or thousands of new molecules is not time- or resource-efficient. “By making the delivery of a new molecule as quick and easy as it is now to order a ‘stock’ chemical, we aim to remove that bottleneck in development and allow the ‘best’ to be used,” Whitby explains.

### Dial-A-Molecule

Dial-A-Molecule began in 2008 as a consultation between the Engineering and Physical Sciences Research Council, the Royal Society of Chemistry, the Institute of Chemical Engineers, and the Chemistry Innovation Knowledge Transfer Network that sought out to identify a “Grand Challenge” in the field of chemistry, which has been defined as an achievement that will have a transformative impact on science or the world at large, and which requires scientists or researchers from many disciplines to accomplish that goal. The Dial-A-Molecule project was one of three projects out of more than 150 submitted that was selected for funding, with \$1.2 million committed via an initial and a continuing grant, and began in 2009.

Whitby says the key hardware components are likely to include a variety of reactors for different functions (which are used to gradually build up the required molecule from simple starting materials); analytical instrumentation (to monitor the process and optimize the chemical process on the fly); and purification equipment (to remove chemical by-products that are present in nearly all chemical reactions). These components would then be linked to-

## Milestones

# Computer Science Awards, Appointments

### WHITE HOUSE HONORS EARLY CAREER SCIENTISTS

More than 100 men and women recently received the U.S. government's highest honor for scientists and engineers in the early stages of their independent research careers—the Presidential Early Career Award for Scientists and Engineers (PECASE).

The PECASE recipients, who received five-year grants from the Faculty Early Career Development

(CAREER) Program, included four computer scientists:

- ▶ Sarah Bergbreiter, University of Maryland, College Park.
- ▶ Daniela A. Oliveira, Bowdoin College.
- ▶ Benjamin Recht, University of California, Berkeley.
- ▶ Noah Snaveley, Cornell University.

### HARD DISK PIONEER GETS MILLENNIUM PRIZE

A British scientist whose work

made it possible for hard disks to radically expand in size has been awarded the million-euro Millennium Technology Prize, Finland's tribute to innovations for a better life.

Stuart Parkin, an IBM Fellow and manager of the Magnetolectronics group at IBM Research-Almaden, and a consulting professor in the Department of Applied Physics at Stanford University, developed a type of data-reading head capable

of detecting weaker and smaller signals than had previously been possible. The innovation allowed more information to be stored on each disk platter.

Technology Academy Finland (TAF), the independent foundation behind the award, said Parkin, who also is director of the IBM-Stanford Spintronic Science and Applications Center, had made Facebook, Google, Amazon and other online services possible.



gether so the material can be routed as needed. While Whitby says that from a hardware engineering perspective, such a machine likely could be constructed today (albeit very expensively), it is the software and analytical components of a machine that have yet to be successfully worked out.

The software used in an organic synthesis machine likely would access databases containing information on chemical compounds and their respective properties, as well as the results from chemical reactions that have been conducted and cataloged by the chemistry community. By using these data pieces, the software would be able to accurately combine materials and automatically produce new molecules with a high degree of accuracy and very little human interaction.

The key issue on the software side revolves around figuring out a way to accurately and efficiently apply the various rules and models that govern the way materials interact in combinational chemistry. The sheer number of rules and models can vary widely based on the raw materials used, as well as the specific combinations of these raw materials, thereby adding significant complexity to a potential machine. In essence, the machine would need to calculate the result of each combination of materials, and then ensure the desired rule or model governing the combination was used, which could result in hundreds or thousands of permutations per combination.

Antony J. Williams, vice president of Strategic Development for the Royal Society of Chemistry (RSC) and leader of the Society's Cheminformatics team (which is working on a collection of reaction data located within its ChemSpider chemical-structure database that will be hosted within the society's developing data repository), notes that this information is key to the development of a machine capable of fully automating the organic synthesis process.

"I am assuming that the machine would be underpinned by a strong software platform that would utilize some form of retrosynthetic analysis using rules extracted from a reaction database," Williams says. "Basic rules will certainly get you some way along the path, but a large database combined with extracted rules is likely the most

powerful approach. We are presently working on building out a 'reaction repository' as part of our development of our RSC data repository and we will be encouraging the community to contribute their reaction data."

The Dial-a-Molecule project is not the only effort focused on finding ways to more quickly synthesize molecules that can be used in research, development, and manufacturing processes. Bartosz Grzybowski, a chemist at Northwestern University in Evanston, IL, is working on a synthesis machine of his own based on Chematica, a software/database that uses algorithms and a collective database of 250 years of organic chemical information to predict and provide synthesis pathways for molecules. Chematica supports 3D modeling of individual molecules, as well as labeling of functional groups, and Grzybowski is negotiating with Elsevier to incorporate the program into its Reaxys database, and also is said to be bidding for a \$2.3-million grant from the Polish government to use Chematica as the brain of a synthesis machine that can plan and execute the synthesis of at least three drug molecules.

Despite the obvious benefits of an organic synthesis machine, it could be years before one actually comes to fruition, according to Whitby, who notes that less than \$100,000 of the Dial-A-Molecule funding grant went to actual research, with the bulk of the money used to "identify how we might get to the target and the key challenges."

"The Grand Challenge has a 30-40-year estimated delivery time, so completion is not imminent," Whitby says, contrasting it with large projects that had a fixed, tangible goal, such as landing on the Moon. However, he notes that achievements made over the next 30 or 40 years on the path to the development of an organic synthesis machine likely will have a substantial impact on chemistry specifically and our world in general.

Still, while Grzybowski did not respond to a request for comment for this article, he has been quoted as stating that an organic synthesis machine could be built and available within five years. Because he has been shopping Chematica to various entities, few independent assessments of Grzybowski's efforts have been conducted.

"I have to believe that it is the chemistry itself that will be the largest limitation, [with] kinetics of reaction, side-products and issues such as precipitation/crystallization," Williams says. "I remember trying to do flow-kinetics in an NMR (nuclear magnetic resonance) probe, only to have solid drop out and clog the lines."

Indeed, work is being done to smooth this process. Jamison Research Group, led by Massachusetts Institute of Technology chemistry professor Tim Jamison, is working on continuous-flow synthesis methods, through which reactions occur as the chemicals move through a machine (rather than in a step-by-step process), which can improve speed and yields. This type of continuous-flow reaction process is better suited to automation, and could be integral to the efficient and error-free design of a fully automated organic synthesis machine.

Furthermore, Williams notes the overall success of any future organic synthesis machine is predicated on the quality of the underlying reaction databases and the various rules or algorithms used to govern the choice of chemical reactions that can be performed.

"Any predictive algorithm, especially for retrosynthetic analysis, is massively influenced by the underpinning training set and extracted models," Williams says, which often renders an imperfect end result. Williams says any machine capable of conducting organic synthesis likely will require some form of self-learning capability, so it can grow more efficient over time. ■

#### Further Reading

Dial-A-Molecule: <http://www.dial-a-molecule.org/wp/>

Chematica: [www.chematica.net](http://www.chematica.net)

ChemSpider: [www.chemspider.com](http://www.chemspider.com)

Jamison Research Group: <http://web.mit.edu/chemistry/jamison/>

What is Organic Synthesis? [https://www.youtube.com/watch?v=rh0Tn\\_oPS30](https://www.youtube.com/watch?v=rh0Tn_oPS30)

Steps in Organic Synthesis: <https://www.youtube.com/watch?v=hZtQuKQbXg&list=PLS7sq10QwYGDqEo5wb65oxGr0Yqkx7&index=13>

Keith Kirkpatrick is principal of 4K Research & Consulting, LLC, based in Lynbrook, NY.

© 2015 ACM 0001-0782/15/03 \$15.00

# Car Talk

*Vehicle-to-vehicle communication is coming. Are we ready for it?*

**D**RIVERLESS CARS ARE the news media's darlings, promising commuters an extra hour's sleep as they whiz down the world's highways. Yet technologies that assist your ride rather than control it will be part of our automotive experience long before this robot-chauffeured vision comes to fruition. Onboard sensors such as back-up cameras already extend our senses by allowing us to observe the world directly; now, vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) technologies—collectively known as “V2X”—stand poised for widespread adoption, appearing in new-model cars as early as 2016, and they are likely to be required eventually, despite current consumer fears.

Like back-up cams, V2X technologies promise safety advantages even if fully driverless cars never become a reality. A report released by the U.S. Department of Transportation (DOT) in August posited that two specific V2V applications would prevent more than 500,000 crashes and 1,000 deaths per year in the U.S.: “Intersection Movement Assist” (IMA), which warns of cross-traffic at intersections, and “Left Turn Assist” (LTA), which watches for traffic approaching from the opposite direction when making a left turn. Other anticipated V2V applications could include collision avoidance in stop-and-go traffic and at highway speeds; speed maximization (and gas savings) for signals and traffic, and parking assistance.

While V2X's proponents tout that drivers remain anonymous under the proposed standards, some applications could also be used by law enforcement, for example to prevent a vehicle from entering a restricted area.

There is a big difference between onboard sensors and V2X technologies. A back-up camera is useful the moment it is installed in a vehicle, regardless of whether any other vehicle has one. Development continues to be



strong in systems with V2X-like features, even though they do not actually communicate with other cars or road infrastructure. For example, the 2009 model year saw Opel introduce its quasi-V2I “Opel Eye” technology, which uses cameras to recognize road signs and lane markings; and for the 2014 model year Mercedes-Benz debuted its quasi-V2V “Distronic Plus” system, which uses radar to judge distances to other cars.

True V2X technology offers capabilities those systems could not provide, such as warning of conditions that are undetectable by sight or radar, at a distance up to 300 meters. The message set uniquely identifies vehicles and infrastructure components using a fast point-to-point signal with relatively few protocol requirements.

## **Governments, Automakers, and the Public**

Yet V2X systems are only as useful as the network to which they are connected, whether that network's nodes are in other cars (V2V) or on lamp-posts, traffic lights, or the roadway itself (V2I). That could cause a stalemate, as automakers are loath to in-

vest in development if no one else does—or if their technologies do not work together.

“Whenever a system has to be standardized, a mandate has to be given,” says Neelam Barua, automotive and transportation industry analyst for Frost and Sullivan. “That was the case for antilock braking systems and back-up cameras.” As a result, industry groups such as Europe's CAR 2 CAR Communication Consortium and the Intelligent Transportation Society of America have been working to establish standards for vehicle-to-vehicle communication, while governments decide how to implement them.

Barua believes V2X requirements will be enacted in Europe soon after European Union-funded trials are completed this year, although European automakers prefer a market-driven approach. In the U.S., the DOT's National Highway Traffic Safety Administration (NHTSA) paired the release of its lengthy report with an “Advance notice of proposed rule-making” (ANPRM) which “initiates [proposals] ... to require vehicle-to-vehicle (V2V) communication capability

for light vehicles”—that is, passenger cars and light trucks.

Despite V2V's readiness and promised benefits, the public may resist such mandates—especially in the U.S. The NHTSA's notice opened a 60-day period for public comment, which elicited nearly 1,000 responses, of which 482 met the NHTSA's submission policy. The consensus was overwhelmingly negative, citing fears that V2V technologies would lead to loss of privacy, inattentive drivers, malicious hackers, and health risks to those claiming the widely discredited condition of “electromagnetic hypersensitivity.” Further, some blanched at the likely cost of about \$300–\$350 to build V2V features into new cars, or retrofit them into older ones.

Barua believes such opposition pales in comparison to the human costs that would come from blocking V2V technologies. “Studies have shown that vehicle-to-vehicle communications could reduce traffic jams, and a lot of lives could be saved,” he says. “Even pedestrians would benefit—they could know when a vehicle is coming, through vibrations on their mobile phones. I think governments should intervene, and take a lead for V2V communications to be functional as soon as possible.”

### On the Road

The U.S. DOT report provides a thorough list of standards for that country, including V2V message contents and transmission performance requirements. The main set of architectural and procedural standards is spelled out in IEEE 1609, “Family of Standards for Wireless Access in Vehicular Environments (WAVE).” Of special interest to those who fear malicious hackers is IEEE 1609.2, “Security Services for Applications and Management Messages,” which was finalized and published in April 2013.

A second set of standards, currently still in development, may be found in SAE J2735 and SAE J2945. These spell out what information each message packet would carry; the latter also includes a section on privacy and security. A third standard, IEEE 802.11p, addresses physical standard specifications for automotive-related “Dedicated Short Range

**“For drivers, I think it will be very seamless; they won't know whether it's V2V, V2I, or sensors that provide guidance.**

Communications” (DSRC). (DSRC has been used for electronic toll collection for over 10 years; this was essentially the first widespread vehicle-to-infrastructure application.)

In the U.S., signals for V2I and V2V communications are carried over 75MHz of spectrum in the 5.9GHz band, which was allocated for DSRC purposes in 1999. China, Europe, and Japan have also reserved DSRC bandwidth near this range; Japan also uses spectrum in the 760MHz band, and Korea is reportedly considering a move to this range.

However, regional standards for DSRC and other aspects of V2V communication are not always compatible with each other, so it is unclear whether cars designed for one regional market would ever work in another. There are efforts to harmonize them somewhat: For example, Japan might implement security according to IEEE Standard 1609.2, currently being used by the U.S. and the EU.

Public V2V tests began in 2006, when DaimlerChrysler got a Mercedes-Benz and Dodge talking to each other, and General Motors demonstrated crash avoidance between two Cadillacs. The first consumer car with true V2V communication capabilities, according to public announcements, will be a 2017 Cadillac equipped with the company's “Super Cruise” technology.

Automakers have generally embraced V2X technology as a logical piece of a larger assisted-driving picture. Hideki Hada, general manager of Integrated Vehicle Systems at Toyota Motor Engineering & Manufacturing North America, described how diverse sensor, V2I, and V2V

## ACM Member News

### LIFELONG LOVE OF MATH SPURS TANNEN IN CS



Val Tannen, professor of Computer and Information Science in the Engineering School of the

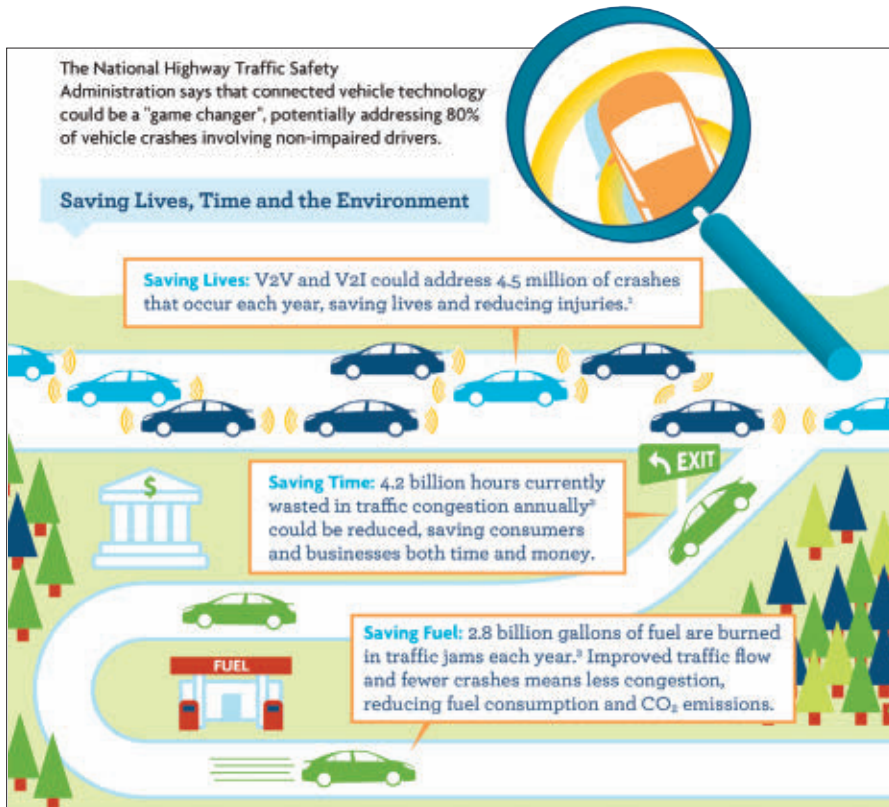
University of Pennsylvania (UPenn), has had a lifelong love of mathematics. “There's no ambiguity about math. The conclusions are inevitable if you accept the assumptions,” he says.

Language and mathematical logic are the dominant influences on Tannen's research. He earned his Computer Engineering degree from the University Politehnica of Bucharest; “There's no Romanian-equivalent master's degree; one degree covers both bachelor's and master's,” he explains. After graduation, he worked as a programmer for a local computer firm and moonlighted as a researcher. Fate intervened in 1982; Tannen came to the U.S. to visit his ailing father and wound up seeking political asylum here. He went to the Massachusetts Institute of Technology, where he received his Ph.D. in Applied Mathematics in 1987; a month later, Tannen joined UPenn.

Tannen's expertise spans programming languages, databases, parallel processing, and logic in computer science and its applications in life sciences. “The word ‘language’ comes up all the time in my research,” Tannen says. During his 27 years at UPenn, he has interwoven mathematical logic and language into his career as a computer scientist. Tannen also specializes in data provenance, a record of the origin and transformation of data.

When he introduced a course at UPenn called “Friendly Logics,” Tannen recalls, the title was approved only after he authored the mathematical definition: “A logic is said to be friendly if and only if, it strikes a good balance between expressiveness and algorithmic tractability.” That neatly sums up Tannen's career as well.

— Laura DiDio



1 Frequency of Target Crashes for IntelliDrive Safety Systems (DOT HS 811 381), October 2010

2 <http://www.its.dot.gov/connected-vehicle/connected-vehicle-research.htm> - December 3, 2013

3 <http://www.its.dot.gov/connected-vehicle/connected-vehicle-research.htm> - December 3, 2013

technologies could work together. "I think they're very complementary. We already have sophisticated radar and camera systems so we can see the car ahead of us. However, with V2V we can also get information about other cars, such as their driving speed, or if they're applying brakes; we don't need to observe it. And it would be great to get information about traffic signals as I approach them so I'll know when to stop, or how fast I should drive to go through the next three signals while they're green. But for drivers, I think it will be very seamless; they won't know whether it's V2V, V2I, or sensors that provide guidance."

### Not Driverless, but "Automated and Connected"

One facility that is actively studying all three is the University of Michigan Transportation Research Institute (UMTRI), which took the lead in a recently completed \$31-million government-funded Connected Vehicle Safety Pilot Model Deployment that placed nearly 3,000 V2V-enabled vehicles on the streets of Ann Arbor, MI, including nearly 2,600 private cars, three passenger buses, and 19 other

commercial vehicles.

The majority of vehicles in the pilot deployment were equipped with a broadcast-only "vehicle awareness device," according to Debby Bezzina, UMTRI's senior program manager. "These are what we call 'target vehicles;' in a nutshell, they transmit position, speed, and heading. So they're saying, 'Here I am! Here I am! Here I am!' 10 times a second."

Approximately 400 vehicles were equipped with devices that could read these signals and react to them with an audible tone. Automakers including Ford, General Motors, Honda, Hyundai-Kia, Mercedes-Benz, Nissan, Toyota, and Volkswagen supplied 64 integrated vehicles that reacted with audible, visible, and tactile warnings. (None of these cars had "self-driving" capabilities—they merely alerted the driver.) The two-year study resulted in approximately 47 terabytes of data from 27 million miles on the road; this collection of information was analyzed by an evaluator within the U.S. DOT, and informed the U.S. DOT's August report.

Bezzina saw a flurry of industry ac-

tivity after that report's release, as well as the announcement by General Motors CEO Mary Barra that the company would be offering advanced intelligent and connected technology on certain 2017 models. Yet Bezzina believes it will take years for sensors, V2I, and V2V to all come together to make "automated and connected" vehicles that look a lot like Google's vision of a "driverless" car. Even when such cars hit the road, she expects to see differences between the two visions.

"With the Google vehicle, everything is standalone; you're not talking to other vehicles, you're not talking to the infrastructure," she says. "But I think a fully automated vehicle is also connected—first with sensors and GPS, and then V2I communication, and then V2V communication with the other vehicles in my lane. I think that, in my lifetime, there will be a special lane on the freeway that you can only get in it if you're in such a car. And then you can take your hands off the wheel and read the newspaper."

### Further Reading

Harding, J., Powell, G.R., Yoon, R., Fikentscher, J., Doyle, C., Sade, D., Lukuc, M., Simons, J., and Wang, J.

(2014, August). *Vehicle-to-vehicle communications: Readiness of V2V technology for application*. (Report No. DOT HS 812 014). Washington, DC: National Highway Traffic Safety Administration <http://1.usa.gov/1wYQ8TY>

CAR 2 CAR Communication Consortium (European) <https://www.car-2-car.org>

Intelligent Transportation Society of America <http://www.itsa.org>

Vehicle-to-Vehicle Communications U.S. National Highway Traffic Safety Administration <http://www.safercar.gov/v2v/>

University of Michigan Transportation Research Institute <http://www.umtri.umich.edu>

Safety Pilot Model Deployment program [http://www.its.dot.gov/safety\\_pilot/spmd.htm](http://www.its.dot.gov/safety_pilot/spmd.htm)

Toyota Collaborative Safety Research Center <http://www.toyota.com/csrtc/>

Tom Geller is an Oberlin, OH-based technology and business writer.

© 2015 ACM 0001-0782/15/03 \$15.00

# Python for Beginners

*A survey found the language in use in introductory programming classes in the top U.S. computer science schools.*

**T**HE WAY TAYLOR POULO sees it, learning to code in Python is comparable “to learning Latin and romantic languages.” Once someone grasps the logic behind Python, the concepts can be more easily transferred to other languages, maintains Poulos, a senior majoring in industrial engineering at the Georgia Institute of Technology (Georgia Tech). “Once you get comfortable thinking in a different type of logic and using different words, it’s much more comfortable to learn new things,” she says, adding that she was required to take three computer science classes at Georgia Tech, all in Python. “Python did that.”

Python, an open source scripting language, has become the most popular introductory teaching language at top U.S. universities—Georgia Tech among them—according to a recent survey by Philip Guo, an assistant professor of computer science at the University of Rochester. Guo decided to conduct the research after noticing anecdotally over the past few years that Python was replacing languages such as Java as the de facto introduction to programming class in more and more computer science classes at universities around the country.

Because it is a scripting language, Python automates tasks that would otherwise need to be performed manually. Java and C++ also are popular and widely used. The main difference is that Python programs tend to run slower than Java programs, but they take significantly less time to develop, according to the Python Software Foundation. Python programs also tend to be shorter than equivalent programs written in Java because of “Python’s built-in high-level data types and its dynamic typing,” the Foundation notes. While the same is true of C++, Python code is generally one-fifth to one-tenth the length of equivalent C++ code, and “Anecdotal evidence suggests that one



Python programmer can finish in two months what two C++ programmers can’t complete in a year,” the Foundation’s website states.

During the summer of 2014, Guo went to the websites of the top 39 U.S. schools for computer science as ranked by *U.S. News & World Report* in 2014, and collected as much data as he could from looking at their introductory computer science courses. He stopped at 39, he explains, because there was an eight-way tie for 40 and “we had to stop somewhere.” At schools including the Massachusetts Institute of Technology (MIT), Carnegie Mellon University, and the University of California, Berkeley, Python emerged as the leading language to teach novices (the full list, along with Guo’s blog on the topic, can be found at <http://bit.ly/W0vt0x>).

Proponents say it is no surprise Python has become the most popular teaching language in colleges, because compared to programs like Java, it is easier to learn and to use to write programs that do practical things with very little code.

With Python, “There’s very little

**In contrast to Java, Python makes more sense for people who are writing small programs.**

overhead in getting to the point where people can start to write interesting programs; the syntax is pretty straightforward,” observes John Guttag, professor of electrical engineering and computer science at MIT, and the author of several books, including one about learning to program in Python. In contrast to Java, which has a “fairly complicated syntax and fairly complicated static semantics,” Python makes more sense for people who are writing small programs, he says. Java is designed to support people writing large, “industrial-quality” programs containing thousands of lines of code, says Guttag, who teaches one of two introductory courses offered by his department.

Another reason Guttag believes more colleges are using Python as an introductory programming language is that it has “a very large set of highly useful libraries that have been built over the years that support things ... that are easy to use from language proper, and that makes Python a particularly useful language for scientists and engineers who want to take advantage of those libraries.”

Python is also very good for “letting you teach conceptual material without getting in the way,” observes Guttag. “So I don’t find myself spending all my time explaining Python to the students. I get to spend a lot of time explaining what I think are more long-lived concepts,” like algorithmic complexity.

Not everyone agrees Python is the be-all-end-all as an introductory programming language. Shriram Krish-

namurthi, a professor of computer science at Brown University, acknowledges Python has many nice features. “It offers a pleasant syntax, a large set of libraries, and an interaction loop ... all of which are very useful for teaching. Compared to the noise and complexity of Java, it is indeed a very nice step forward.” He agrees Python has made people feel more comfortable about exposing programming to a much broader audience of students.

“There are many students I would not dream of teaching Java to that I would happily show Python.” That said, however, it does not take long to discover Python’s weaknesses, Krishnamurthi notes. Among them are that “Creating non-trivial data structures is onerous, because Python does not provide straightforward means for creating new structured data. You have to understand a bunch of unrelated concepts, like classes, and their onerous syntax and tricky semantics, which greatly reduces the benefit of simplicity that Python was supposed to offer.”

Because of this, he believes more and more curricula are ditching the idea of structured data—one of the central concepts in computer science—and doing one of two things: shaping their curriculum to avoid them, or pushing students to encode more-structured data in less-structured formats provided by default in Python.

“This lack of data structuring and classification has a significant nega-

## “Choosing Python is the modern equivalent of the old adage, ‘nobody ever got fired for buying IBM.’”

tive impact on teaching program design,” Krishnamurthi says. “The best program design methods we have right now focus on data-driven design, which derive from the structure of data.”

Additionally, Python has limited support for testing, he says. Even though it has professional testing libraries, he says they can be “onerous for beginning students; the language provides no native support for it, which makes the user interface of testing significantly weaker than it should be.” Testing is not just a matter of finding bugs, he adds. “It also guides students towards the design of solutions and greatly affects how one views debugging. Thus, another vital design and development methodology is taken away.”

Lastly, Python lacks static types, says Krishnamurthi, which “is a central point in teaching programming, and should not be put off for too long. Python offers no good means for

teaching this.”

Guo says he got some backlash from older colleagues who do not view Python as a serious programming language, along with comments that it is not as “industrial strength” as other languages. One comment he received after posting his blog on the topic is that Python is a dynamically typed language and “There’s fair amount of instructors who prefer statically typed languages, like Java,” Guo says. “So they aren’t as happy about this new movement.”

Mark Guzdial, a professor in the School of Interactive Computing at Georgia Tech, says Guo’s research notwithstanding, Java is still the most popular introductory programming language in the U.S. Guo “constrained his search to top U.S. universities,” Guzdial says, “and in general, if you look at book sales, Java is still the most common [language taught] and C++ is second.”

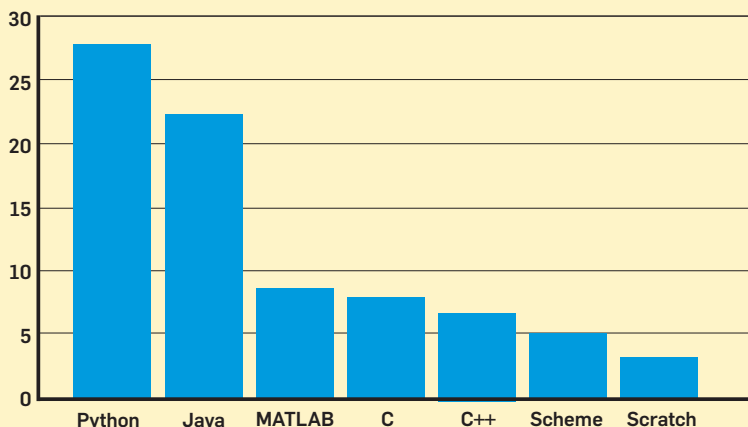
Yet, Guzdial agrees that if someone lacks any prior programming experience, Python is a good language with which to start. “There’s a significant amount of evidence that graphical programming languages are easier for people to get started with than textual,” he says. “If you understand variables in Scratch, it’ll be easier to understand variables or conditionals or loops in Python.”

Like Krishnamurthi, Guzdial thinks Python may broaden the scope of people being able to code. “It is easier and more accessible and ... you can get more done in fewer lines of code.”

Matt Guthmiller, a sophomore majoring in electrical engineering and computer science at MIT, says he was not terribly surprised by Guo’s finding. Guthmiller took his first Python class as a freshman at MIT, but he already knew how to code in C, C++, and JavaScript. “It definitely seems to make a lot of sense as an introductory tool because it’s easy to learn, with lots of functionality built in, and you can do things that in other languages you’d have to build yourself, and [in Python] they’re provided for you.”

He likes that Python allows you to summarize a list of data in one line of code, whereas in other languages it would take multiple lines. “You have to think about the order you want to iterate these items and implement

Number of top 39 U.S. computer science departments that use each language to teach introductory courses



Analysis done by Philip Guo ([www.pgbovine.net](http://www.pgbovine.net)) in July 2014, last updated 2014-07-29

them, so there's a lot more functionality built into one line, so if there's a problem you want to solve in one line of code, you get much closer to solving the problem than in other languages."

Like Guttag, Guthmiller feels the biggest disadvantage of Python is that "the syntax is quite different from most other programming languages," making it trickier to move on to another language once you get all the general concepts down. However, he says, Python's advantages outweigh its disadvantages.

Guthmiller recently used Python to build a controller for a robot to make it follow along a wall, although, generally speaking, his go-to programming language is C++. Python, he says, "gives you a lot of flexibility, and I'm very familiar with it and I am not concerned about having to remember small details."

Abbie Burton, a senior majoring in business at Georgia Tech, was required to take a computer science class and took "Jython," a combination of Java and Python that business students tend to take. She says most engineering students take Python or MATLAB, and she is not surprised by Python's popularity, "I guess because in the real world that's what people use, so they want us to be prepared."

There does appear to be a preference for using Python outside of academia. For the third year in a row, Python was ranked the number one most popular programming language by Codeval, a community of over 24,000 competitive developers, followed by Java, C++, and JavaScript. (<http://bit.ly/1vLiuFj>).

Guo says he has heard some comments that while easy to learn, Python does not have practical applications in the real world and that most coding is done in MATLAB and other languages. He says it all depends on the domain. "MATLAB is used in a lot of scientific domains. I definitely think it's less practical in terms of getting an industry job, because most industry coding would be in other languages, like Java or JavaScript. So I would agree it might not be the language you'd use in your job."


Guttag says Python is a useful tool for people who do not intend to be computer scientists, because it provides a good foundation for learning how to use computation as part of their work.

"For those non-computer science

students," he says, "Python is an excellent choice as the introductory programming language."

Krishnamurthi says Python may be fashionable right now, but he believes it lacks staying power. "Computer science programming education goes in waves of fashion," he says. "Ever since Pascal introduced the idea of 'one programming language for introductory programming education,' the community has been stuck in a rut of trying to find one and then arguing about it. Pascal, C++, Java, Python, Scratch ... take a number."

He likens Python to a "package tour: safe, comfortable, blandly conventional. Choosing Python is the modern equivalent of the old adage, 'nobody ever got fired for buying IBM.'"

Guzdial is also not sure how long Python will be used as the main introductory programming language in academia. "I think Python has hit its tipping point, which may mean we have a couple more years before people say 'Python, what?'" 

#### Further Reading

*Guo, Philip J.*

Online Python Tutor: Embeddable Web-Based Program Visualization for CS Education, Google, Inc. <http://bit.ly/1zB7ugb>

*Guttag, J.V.*

Introduction to Computation and Programming in Python, MIT Press (2013)

*Guzdial, Mark*

Exploring hypotheses about media computation. *Proceedings of the ninth international ACM conference on international computing education research* <http://bit.ly/13SWM0d>

*Enbody, R.J., Punch, W.F., and McCullen, M.,*

Python CS1 as preparation for C++ CS2. *Proceedings of the 40th ACM technical symposium on Computer Science Education* <http://bit.ly/1tJjLu7>

*Pritchard, D. and Vasiga, T.*

CS Circles: An In-Browser Python Course for Beginners. *Proceedings of the ACM technical symposium on Computer Science Education* <http://bit.ly/1z7h8UV>

How to Think Like a Computer Scientist.

*Learning with Python: Interactive Edition 2.0* <http://bit.ly/1tJkknG>

**Esther Shein** is a freelance technology and business writer based in the Boston area.

© 2015 ACM 0001-0782/15/03 \$15.00

#### Research

# ACM Europe Protests H2020 Cuts

ACM Europe Chairman Fabrizio Gagliardi recently contacted European leaders in opposition to proposed budget reductions to Horizon 2020, the European Union's seven-year, 80-billion-euro research funding program.

In January, European Commission President Jean-Claude Juncker unveiled legislation that would remove 2.7 billion euros over 5.5 years from Horizon 2020, the EC's main funding stream supporting research through the year 2020, to devote those funds to economic stimulus through the creation of a European Fund for Strategic Investment. That investment, according to Juncker, would help get the sluggish European economy moving and create new jobs.

The largest share of the cuts would be directed at the European Institute of Innovation and Technology, which aims to spur innovation and entrepreneurship across Europe by bringing together universities, research labs, and companies to form "dynamic cross-border partnerships."

In letters to Juncker, European Council President Donald Tusk, and European Parliament President Martin Shulz, Gagliardi pointed out "the future success of Europe requires Europe to consolidate and advance its position at the forefront of scientific innovation. This goal requires major investments in fundamental research, especially in such critical domains as computing science."

Gagliardi said ACM Europe recommends the European Commission authorities and the European Council "preserve, in the announced cuts to H2020, the support to fundamental research and especially in computing science, given their direct relevance for the focus on innovation of the Investment Plan."

—Lawrence M. Fisher



DOI:10.1145/2723669

Pamela Samuelson

# Legally Speaking

## Copyrightability of Java APIs Revisited

*A recent case challenges the long-standing view that application program interfaces are not protectable under copyright law.*

**F**OR MORE THAN 20 years, the prevailing view has been that application program interfaces (APIs) are unprotectable elements of copyrighted computer programs. According to this view, programmers are free to reimplement other firms' APIs in independently written code. Competition and innovation in the software industry has thrived amazingly well in part because of rulings upholding this understanding.

Challenging this view is the Court of Appeals of the Federal Circuit (CAFC) May 2014 decision in *Oracle v. Google*. The CAFC held that the "structure, sequence, and organization" (SSO) of the Java APIs that Google reimplemented in its Android software are protectable expression under copyright law. It reversed a lower court ruling that the Java APIs were not copyrightable.

Google has asked the U.S. Supreme Court to review the CAFC's ruling. Several amicus curiae (friend of the court) briefs have been filed in support of this effort. Hewlett-Packard, Red Hat, and

Yahoo! are among these amici (as am I and 77 computer scientists).

The Supreme Court may take the case because the CAFC's decision is in conflict with other appellate court rulings that exclude APIs from copyright protection. This column will explain the Oracle and Google theories about the copyrightability of Java APIs and the precedents on which each relies. The stakes in this case could not be higher.

### Oracle's Claims

Developing Java APIs required considerable creativity. Sun's engineers had substantial freedom in the choices they made about how to structure the APIs. The Java APIs are thus easily original enough to qualify for copyright protection, says Oracle (which acquired the intellectual property (IP) rights in Java when it acquired Sun Microsystems).

Java has achieved considerable success, which is why Google wanted to use Java APIs in its software platform for mobile devices. Google entered into negotiations with Sun about licensing

rights in Java, which shows it knew it needed a license.

When these negotiations failed, Google went ahead and copied 37 of the Java APIs anyway in the Android platform for mobile devices. Tens of thousands of Java programmers have written apps to run on the Android platform. These apps have contributed to the extraordinary success of Android devices.

Shortly after acquiring Sun and its assets, Oracle sued Google for copyright infringement. (There were originally some patent claims in the case as well, but a jury ruled against those claims.) Oracle relied on some judicial precedents that had held the SSO of programs is protectable by copyright law as long as there are multiple ways to design that SSO.

### Section 102(b)

At issue in the *Oracle* case is the proper interpretation of Section 102(b) of U.S. copyright law. It states "[i]n no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle or





discovery, regardless of the form in which it is ... embodied in such work.”

Oracle asserts this provision restates the classic distinction between expression (which copyright law protects) and ideas (which are beyond the scope of copyright protection). Because the Java APIs are much more detailed than ideas and may have original elements, they are not ideas alone, but rather expressions of ideas. The CAFC agreed, concluding these Java APIs are copyrightable because of the creativity they embody and the existence of alternative ways in which Google could have developed its own APIs.

### Ninth Circuit Precedents

Google has pointed out the plain language of Section 102(b) makes procedures, systems, and methods of operation unprotectable by copyright law. It asserts the Java APIs at issue are unprotectable under this provision.

Google has relied on several appellate court decisions to support its claims that the Java APIs are unprotectable by copyright law. Especially relevant is the Ninth Circuit Court of Appeals’ ruling in *Sega v. Accolade*.

Sega sued Accolade because it made copies of Sega software in the course of reverse-engineering to get

access to the interface procedures embedded in the Sega code. Accolade needed to know this information to make its videogames compatible with the Sega platform.

The Ninth Circuit held this reverse engineering was a noninfringing fair use because it was done for the legitimate purpose of getting access to interface procedures that were “the functional requirements for [achieving] compatibility” and consequently unprotectable under Section 102(b).

Google claims the CAFC erred by ignoring this aspect of the *Sega* decision. (Ordinarily an appeal from a California federal court would have gone to the Ninth Circuit, but because Oracle originally sued Google for patent as well as copyright infringement, Oracle’s appeal from the copyright loss went to the CAFC instead. The CAFC was supposed to follow Ninth Circuit precedents.)

The CAFC opined that Google’s arguments about compatibility might be relevant to its fair use defense to Oracle’s claim of infringement, but not to whether the Java APIs were protectable by copyright law.

### Origins of Section 102(b) Exclusions

Copyright’s exclusion of systems and methods of operation from the scope

of its protection traces back to the Supreme Court’s 1880 ruling in *Baker v. Selden*. Selden sued Baker because he copied the bookkeeping forms Selden published to illustrate how to implement his new bookkeeping system. Selden won at the trial court level, and Baker appealed.

The Supreme Court perceived the question in *Baker* to be “whether the exclusive property in a system of bookkeeping can be claimed, under the law of copyright, by means of a book in which that system is explained[.]”

The Court ruled Selden’s copyright extended to his *explanation* of the bookkeeping system, but not to the *system* itself, the method of operation it prescribed, or the forms that implemented the system. Such a “useful art” might have been eligible for patent protection, but not for copyright.

The Court observed that “[t]o give to the author of the book an exclusive property in the [useful] art described therein, when no examination of its novelty has ever been officially made, would be a surprise and a fraud upon the public. That is the province of letters-patent, not of copyright.”

Congress codified the *Baker* holding in Section 102(b). A legislative report said it did so “to make clear that the

expression adopted by the programmer is the copyrightable element in a computer program, and that the actual processes or methods embodied in the program are not within the scope of the copyright law.”

The Ninth Circuit in *Sega* recognized copyright law should not protect interface procedures because that would confer patent-like protection on the functional requirements for compatibility without *Sega* meeting the stricter standards required for patents.

The trial judge in *Oracle* expressed concern that Oracle’s copyright claim might be seeking to obtain “an exclusive right to a functional system, process, or method of operation that belongs in the realm of patents, not copyrights.” The court noted that “[b]oth Oracle and Sun have applied for and received patents that claim aspects of the Java API.”

In overturning that decision, the CAFC seemed untroubled about possible overlaps of copyright and patent protection for APIs. In effect, it read the procedure, system, and method exclusions out of the statute.

### Is SSO Protectable by Copyright?

The idea that program SSO is protectable expression as long as there is more than one way to accomplish a programming objective derives from a 1986 Third Circuit ruling in *Whelan Associates v. Jaslow Dental Lab*. Oracle and the CAFC have embraced this theory.

The SSO concept was, however, substantially discredited in the Second Circuit’s 1992 *Computer Associates v. Altai* decision. In the years since *Altai*, courts have largely moved away from conceiving of SSO as protectable expression in programs because it fails to provide a workable framework within which to distinguish protectable and unprotectable structural aspects of programs.

The Second Circuit in *Altai* emphasized the “essentially utilitarian nature of computer programs” makes it difficult to separate protectable and unprotectable structural elements in programs.

*Altai* announced a new “abstraction, filtration, and comparison” test for software copyright infringement. Among the structural elements of programs that must be filtered out before assessing infringement are efficient design elements, elements con-

## Java has achieved considerable success, which is why Google wanted to use Java APIs in its software platform for mobile devices.

strained by external factors, and standard programming techniques.

The Second Circuit in *Altai* was quite explicit that elements of programs “dictated by external factors” such as “compatibility requirements of other programs with which a program is designed to operate in conjunction” lie outside the scope of protection that copyright provides to programs. Such structural similarities must be filtered out before courts can determine whether a defendant infringed copyright.

The Ninth Circuit followed *Altai*’s lead in holding that interface procedures necessary for achieving interoperability among programs were functional elements of programs that copyright did not protect under Section 102(b). In *Sega*, the court cited approvingly to *Altai* for the proposition that computer programs “contain many logical, structural, and visual display elements that are dictated by the function to be performed, by considerations of efficiency, or by external factors such as compatibility requirements and industry demands.”

### Lotus v. Borland

Another important appellate ruling that supports Google’s theory is *Lotus v. Borland*. In 1995, the First Circuit ruled that Borland had not infringed by copying the SSO of the Lotus 1-2-3 command hierarchy for use in the emulation interface of Borland’s Quattro Pro program. Borland had to use the same commands in the same order so that users who had constructed macros of frequently executed functions in the Lotus macro language could con-

tinue to use those macros in the Borland program.


The First Circuit in *Borland* did not find the SSO concept helpful in distinguishing protectable and unprotectable structural elements of computer programs. It held the SSO at issue in *Borland* was an unprotectable method of operation under Section 102(b), akin to the command structure of VCR machines.

The trial judge in the *Oracle* case relied on the *Borland* decision, characterizing the Java APIs as a similar type of command structure. The CAFC chose not to follow *Borland*, and interpreted *Altai* as applicable only when initial designers of APIs are themselves constrained in their choices about structuring the interfaces.

### Conclusion

Twenty years ago, the Supreme Court took Lotus’ appeal from the First Circuit ruling. After oral argument, it split 4-4 on the proper interpretation of Section 102(b) as applied to computer programs. This left the First Circuit opinion intact, but did not make a nationwide precedent. The issues left undecided in that case are before the Court in the *Oracle* case.

Several amicus briefs filed in support of Google’s appeal say that if the Supreme Court does not repudiate the CAFC’s interpretation of copyright law, the result will likely be a new surge in litigation over the protectability of APIs, even though this issue had seemed to be resolved by appellate court rulings going back to 1992.

Oracle filed its brief in opposition to Supreme Court review in December 2014. Google’s petition for review was put on the Court’s calendar in January 2015. The Court decided to ask the Solicitor General to weigh in on whether the Court should hear Google’s appeal (which increases the likelihood the Court will take the case by 46 times). I predict the Court will review this case. How the Court will decide that case remains to be seen. A ruling in support of interoperability is much to be hoped for. 

**Pamela Samuelson** (pam@law.berkeley.edu) is the Richard M. Sherman Distinguished Professor of Law and Information at the University of California, Berkeley.

Copyright held by author.

## Broadening Participation Reaching a Broader Population of Students through “Unplugged” Activities

*Introducing children to fundamental computing concepts through Computer Science Unplugged.*

**T**HE FIRST DECADE of this century saw growth in outreach to raise awareness of computing and the possibility of a career in computing. Some of these efforts were “unplugged,” not requiring a computer, but providing an easy, fast way to present key principles of computer science to a broad audience. This column highlights Computer Science Unplugged (CS Unplugged; [www.csunplugged.org](http://www.csunplugged.org)), activities that are easy to present, require few materials, encourage collaborative work, and do not depend on hardware, compilers, browsers, and Internet connections. They work well when access to computers is limited or nonexistent.

CS Unplugged was developed at the University of Christchurch in New Zealand by Timothy Bell, Ian H. Witten, and Mike Fellows, and adapted for classroom use by Robyn Adams and Jane McKenzie.<sup>2</sup> Activities include basic concepts such as computer data storage, how computers compress information and detect errors, and algorithms for solving common computational problems (searching, sorting, finding minimal spanning trees, using finite automata to model systems). Kids do not simulate a computer (not a particularly interesting endeavor) but learn problem-solving skills that



An illustration from one of the downloadable activities on the CS Unplugged website ([www.csunplugged.org](http://www.csunplugged.org)).

expose fundamental computer science concepts.<sup>1</sup> CS Unplugged activities promote group work, problem-solving skills, and creativity.

For example, a teacher can start with magnets or self-stick notes of two different shapes and ask a child to put a random set of these into a  $7 \times 7$  grid

as shown in the example here:

```
X O X X X X O
X X X X X O X
O X O O O X O
X O X X O X O
O X O O X O O
O O X O X O X
X O O O X X X
```

The teacher, or someone who is in on the “trick,” can then claim to make the problem even more difficult by adding an eighth row and eighth column with seemingly random choices:

```
X O X X X O X
X X X X X O X O
O X O O O X O O
X O X X O X O O
O X O O X O O O
O O X O X O X X
X O O O X X X O
O X O X X O X O
```

The teacher can then leave the room, and a child can change one of the magnets to the other magnet. For example, the child changes the magnet in the second row and third column from X to O:

```
X O X X X X O X
X X O X X O X O
O X O O O X O O
X O X X O X O O
O X O O X O O O
O O X O X O X X
X O O O X X X O
O X O X X O X O
```

The teacher returns and magically picks out the magnet that changed, astounding the children. The teacher asks how this is possible, giving children a chance to discuss solutions with each other, expressing various algorithms or techniques that may have been used. Often, students will eventually see the teacher did not put in a random eighth row and eighth column. Instead, the extra row and column set the number of each magnet in each row and column to be even. The change creates exactly one row and one column with an odd number of each magnet, leading to the magnet that was changed.

The computational thinking principle illustrated in this activity is that of parity, detecting errors in data, which computers have to do constantly. The CS Unplugged activity write-up gives teachers information about parity that they can present to children along with extension activities. For example, what happens if two magnets changed? Can we detect that the change occurred? Can we identify which magnets changed?

One of the keys to the success of CS Unplugged and its use worldwide is the fact the activities do not require a computer at all. Some schools do not

have a computer lab for students to write code. If they do have computer labs, they are often used for word processing and Web surfing for research for other courses. CS Unplugged activities can be done entirely without computers. When the CS4HS (Computer Science for High Schools) workshop was launched at Carnegie Mellon University in 2006, participating high school teachers said they could not teach computer science because they did not have any computers, or enough computers, in their schools. The workshop started with CS Unplugged, and all of the teachers subsequently reported using these activities successfully in their schools the following year with an increase in student interest in computing.

Activities in CS Unplugged support the principle of computational thinking,<sup>5</sup> which promotes the idea that problem-solving skills and computational techniques used in computer science should be a part of every person's education and are applicable to a wide variety of fields, not just computer science. Although one study suggests CS Unplugged activities do not inspire young people to pursue computer science in college,<sup>4</sup> the primary goal of these activities is to expose students to computing as an intellectual discipline that goes beyond their understanding of computers as a tool and a toy. Additionally, these unplugged activities are meant to be supplementary, used for short periods to get kids working together, and to give teachers and students a chance to step away from the comput-

**It is one thing for students to click the Compress option for a file. It is another thing to gain an appreciation for how that process works.**

er and programming-based activities. More formal studies are needed that validate that CS Unplugged is effective in meeting its goals.

Another key to the widespread use of CS Unplugged is its ability to get kids engaged in the activities physically, and most activities encourage group work so kids work together to solve problems, much like computer scientists do when working on large complex software and hardware systems. CS Unplugged exemplifies an educational theory known as experiential learning, where participants learn through activity outside of a standard academic setting.<sup>5</sup> By being physically part of the solution to a problem as it is being solved, kids learn from observations and experiences. Unlike some introductory programming activities that tend to promote solo activity, the CS Unplugged activities put kids physically in the middle of the problem, getting them moving, working together, sharing ideas, and designing solutions.

One activity in CS Unplugged involves compressing text by finding repeated letter sequences. Kids can work together to compress some large paragraphs to a fraction of their size, competing to see who can compress the text the most. As a result of this activity, kids learn one way their computer makes files smaller so they can store more on their hard drive. It is one thing for them to click the Compress option for a file. It is another thing to gain an appreciation for how that process works. And some kids wonder if there are other compression algorithms and why this one works so well, leading to further exploration.

Another activity simulates parallel sorting, where children walk through a parallel sorting network drawn on the ground with chalk, comparing themselves using some measure with other children they encounter, following the appropriate path to another node in the network until they reach the end. They see that no matter how they are organized initially, the network will lead them into sorted order. The activity comes with several networks the teacher can use, and it can be adapted based on the number of students in the activity.

Yet another activity involves a set of

“islands,” with children traveling from one island to another on “pirate ships.” As a child arrives at an island, the island’s overseer (a teacher or another child) gives them two options to travel from that island to another island: A or B. Depending on which letter the child picks, they are sent to one or another island to then answer the same question. Their goal is to find their way to Treasure Island, and as they move from “island” to “island,” filling in a map with their choices, they are forming a finite state automata. Students will share information to find the fastest path to Treasure Island, the longest path (which involves cycles), all of the paths, and so on, describing them as a regular expression. Later, the teacher can show how they can look at this problem abstractly as a set of states with directed edges labeled “A” and “B,” and how automata can be used to describe other complex systems like traffic lights and vending machines.

CS Unplugged activities are gender neutral and encourage participation by all groups. Illustrations in the activities show pictures of boys and girls performing the activities. The National Center for Women in Information Technology (NCWIT), has included CS Unplugged in its materials for teachers to encourage girls to learn about information technology and pursue a career in IT.<sup>a</sup> Exploring Computer Science, a curriculum for secondary-level students that has been used successfully in school districts with significant minority populations like those in Los Angeles and Chicago, has included CS Unplugged in its unit on problem solving.<sup>b</sup> Carnegie Mellon University uses CS Unplugged in its TechNights workshops to encourage middle school girls to learn about computer science,<sup>c</sup> and Howard University has used CS Unplugged to increase awareness and appreciation of computational thinking for African American students.<sup>d</sup>

CS Unplugged has also been used in events sponsored by AccessComputing<sup>e</sup> at the University of Wash-

## With a little creativity, the activities in CS Unplugged can be adapted for any population.

ington for young people with disabilities. For example, blind children with their canes sitting in a row of chairs represents an unsorted array. The children learn various comparison-based sorting algorithms by comparing the lengths of their canes and moving to the appropriate chairs depending on whose cane is longer. When sorting by birthdays they shout out their birthdays according to an algorithm. This allows a group of children to stand up together to move in unison to the next chair, thus demonstrating a parallel sorting algorithm. The concept of a parallel algorithm and broadcast become quite real to the children in the process of executing the algorithm in their chairs.

With a little creativity, the activities in CS Unplugged can be adapted for any population. In fact, the CS Unplugged website lists numerous extensions for each of the original activities, submitted by volunteers all around the world. The activities have had such an impact that the CS Unplugged curriculum, originally published in English, has been translated to a number of languages such as Spanish, German, French, Italian, Portuguese (Brazilian), Polish, Russian, Slovenian, and Japanese.

CS Unplugged is not the only unplugged way to introduce computing ideas to kids. Tinkersmith, an organization based in Oregon, has developed a number of activities for K–12 students that do not require computers including Binary Baubles and My Robot Friend. Binary Baubles involves using hands-on techniques to have kids encode text using ASCII, and other methods. My Robot Friend requires participants to write programs on pa-

per for a “robot” (another participant) to follow to stack a set of cups into a particular configuration. Another organization, Kodable, has activities that teach kids how to program by having them program each other using a graphical set of instructions (for example, squat, jump, rotate, grab) to navigate obstacles and reach a goal. CS Unplugged is included, along with these unplugged activities, as part of the Hour of Code<sup>f</sup> for schools that either do not have computers or that want to include other computing activities beyond computer programming.

Since it was first introduced in 1998, the growth in the use of CS Unplugged by organizations and teachers provides evidence that this method of introducing computing to kids is a valuable resource regardless of whether or not they have access to computers. As computing professionals, we should encourage the addition of unplugged activities in our schools to help children see the ingenuity, creativity and teamwork involved when working on computational problems. We should help to create, study, and evaluate new unplugged activities for teachers to use to reach a more diverse population of children. Through these efforts, we just might connect with young people who never thought computing could be a potential career path, and change their minds. ■

<sup>f</sup> <http://hourofcode.org>

### References

1. Bell, T., Alexander, J., Freeman, I., and Grimley, M. Computer Science Unplugged: school students doing real computing without computers. *Journal of Applied Computing and Information Technology* 13, 1 (2009).
2. Bell, T., Witten, I.B., and Fellows, M. Computer science unplugged: Off-line activities and games for all ages. Computer Science Unplugged, 1998; <http://www.csunplugged.org>.
3. Kolb, D. *Experiential learning: Experience as the source of learning and development*. Prentice Hall, Englewood Cliffs, NJ, 1984.
4. Taub, R., Armoni, M., and Ben-Ari, M. CS Unplugged and middle-school students' views, attitudes and intentions regarding computer science. *Trans. Comput. Educ.* 12, 2 (Apr. 2012).
5. Wing, J. Computational thinking. *Commun. ACM* 49, 3 (Mar. 2006), 33–35.

**Thomas J. Cortina** ([tcortina@cs.cmu.edu](mailto:tcortina@cs.cmu.edu)) is Assistant Dean for Undergraduate Education in the School of Computer Science at Carnegie Mellon University. He is an advisor for Computer Science Unplugged and for the AP Computer Science Principles curriculum. He is a member of SIGCSE, co-chairing SIGCSE 2011, and he is a senior member of ACM. He has demonstrated the use of CS Unplugged at numerous workshops for K–12 teachers.

Copyright held by author.

a <http://www.ncwit.org/resources/computer-science-box-unplug-your-curriculum>

b <http://www.exploringcs.org>

c <http://women.cs.cmu.edu/technights/>

d <http://www.scs.howard.edu/research/PEECS>

e <http://www.washington.edu/accesscomputing/>

## The Profession of IT A Technician Shortage

*In our relation about rising CS enrollments, we are overlooking a growing shortage of computing technicians. Our education system is not responding to this need.*

**O**N THE TENTH anniversary of this column, we took stock of changes in the computing profession since 2001.<sup>2</sup> Computing had become the umbrella term for our field, rather than information technology (IT) as was expected in 2001; IT referred mainly to technology and business applications of computing. Several new professions had appeared within computing to support changes such as big data, cloud computing, artificial intelligence, and cyber security. Certification of important skill sets was more common, but professional licensing had not advanced very much. Finally, there was a sharp drop in enrollments in computer science departments around the world, to about 50% percent of the 2000 peak. Many considered this a paradox because computing jobs were growing and digitization was moving into every field and business.

In 2007 CS enrollments bottomed and began to rise steadily, attaining in 2013 75% of the peak level. Surveys show students are taking up computing not so much because they expect good salaries, but because they perceive computer science as compatible with almost every other field. A major in computer science gives the flexibility of deferring a career choice until graduation.

This reversal has brought great rejoicing among computer science academic leaders. Their attention is focused on coping with the surge of enrollments, which seems like a happy misfortune.



But the surge diverts attention from an underlying big, messy problem. Most CS university graduates are heading for the currently plentiful elite designer jobs, in which they will create and design new computing technology. There are a great many more unfilled technician jobs and more will be needed to support the infrastructure.

Who will operate and maintain the information infrastructure on which so much else depends? That is our worry. Universities say they are not preparing technicians; training is outside their scope. Technician jobs, which do not pay as well as the de-

signer jobs, do not attract the university graduates. Community colleges and two-year colleges do not seem to have enough capacity to meet the need. There are few programs to transition workers displaced by digital automation into these digital technician jobs.

As our graduates find more and more clever ways to automate knowledge work, the number of displaced workers will rise. The displaced would readily take the IT technician jobs but the education system offers them few paths for retraining. To quote *The Economist* (Oct. 4, 2014): “Vast wealth

is being created without many workers; and for all but an elite few, work no longer guarantees a rising income.”

### Technician Shortage

To begin, we acknowledge there is controversy around whether there is a shortage of IT workers.<sup>1</sup> The whole market of IT jobs does not worry us; just the segment we call technicians.

The U.S. Labor Department defines IT technicians as those who diagnose computer problems, monitor computer processing systems, install software, and perform tests on computer equipment and programs. Technicians also set up computer equipment, schedule maintenance, perform repairs, and teach clients to use programs. Technicians need strong knowledge of computers and how they operate, including a broad understanding of hardware and software, operating systems, and basic computer programming. Many technicians must be familiar with electronic equipment, Internet applications, and security. Technicians may also need good communication skills because they interact frequently with people who have varying levels of IT knowledge.

The U.S. Labor Department reported in September 2014 that 16 million mid- and low-skill workers had been displaced by automation and would presumably become employed if they could be retrained. If those people and the underemployed (people with part-time jobs seeking full-time employment) were counted in the unemployment figures, U.S. unemployment rate would have been 11.8% rather than 5.9% in that September. Even retrained workers have had difficulty finding employment. One reason is that employers prefer people with specialized knowledge of their systems. Another is age discrimination—people in their 50s have a much more difficult time finding employment in IT companies than those in their 20s and 30s.

For perspective see the accompanying table, a map of the subdivisions of the computing field (adapted from the 2011 column<sup>2</sup>). The computing departments in the universities are, of course, focusing on the education in the computing core disciplines. Simi-

#### Professional subdivisions of the computing field.

Computing-Core Disciplines	Computing-Intensive Disciplines	Computing-Infrastructure Occupations
Artificial intelligence	Aerospace engineering	Computer technician
Cloud computing	Bioinformatics	Cyber operator
Computer science	Cognitive science	Database administrator
Computer engineering	Computational science	Help desk technician
Computational science	Digital library science	Network operator
Database engineering	E-commerce	Network technician
Computer graphics	Genetic engineering	Professional IT trainer
Cyber security	Information science	Security specialist
Human-computer interaction	Information systems	System administrator
Network engineering	Public policy and privacy	Web identity designer
Operating systems	Instructional design	Web programmer
Performance engineering	Knowledge engineering	Web services designer
Robotics	Management information systems	
Scientific computing	Network science	
Software architecture	Multimedia design	
Software engineering	Telecommunications	

larly other academic departments are focusing on the computational part of their fields. Who is focusing on the third column, the computing infrastructure technicians?

Not the computing departments in four-year colleges. In fact, they call that form of education “training” and say they do not do training. They leave the “training” to two-year colleges, career academies, and a growing number of private firms that offer training certificates.

The Manpower Group (<http://www.manpowergroup.com/talent-shortage-explorer>) lists 10 jobs employers are having most difficulty in filling. The top ones globally include skilled trades, technicians, engineers, sales representatives, and IT staff. Many skilled tradespeople, engineers, and IT staff fit our definition of technician given in this column.

An example of a technician shortage can be seen in the cyber operator category. Cyber operators manage networks and provide for network security. The U.S. Department of Defense has been looking for 6,000 cyber professionals since 2012. In 2014, they had filled 900 and still hoped to fill them all by 2016. Whether they can is an open question.<sup>4</sup>

The report “Job Growth and Education Requirements Through 2020” (<http://cew.georgetown.edu/recovery2020>) says that 66% of job openings by 2020 will be sub-bachelor. Most jobs will require some post-secondary education and will rely more on communication

and analytic skills than on manual skills. Those with only a high school diploma will have fewer employment options. Education at the sub-bachelor level is very important and yet is not well funded. For example, The Brookings Institution in “The Hidden STEM Economy” (<http://www.brookings.edu/research/reports/2013/06/10-stem-economy-rothwell>) notes there are many sub-bachelor STEM jobs, but only one-fifth of U.S. federal spending allocated for STEM education goes to sub-bachelor education such as two-year colleges.

The huge and growing demand for providing training in computer coding to young people (code.org, codecademy.org, khanacademy.org, coderdojo.org, girlwhocode.com and more) demonstrates that coding is a sub-bachelor STEM skill in high demand and that young people are eager to learn it. Coding is the basis of many technician skills in IT. We are also concerned the current surge of interest in coding should not become a dead end, but open a path to the full set of principles making up computing science.

### Investments in Training and Continuing Education

Given the importance of finding qualified employees and keeping them from becoming obsolete, one would think that companies are investing in training of prospective employees and continuing education of on-board employees.

Yet there are worrisome reports that this principle is not widely accepted. An IBM division recently declared it would reduce salaries of employees by 10% for a six-month period while they were receiving training.<sup>5</sup> The training was needed to maintain their qualifications for their future jobs. For IBM, this is a sharp break from its own history of supporting education and professional development of its people. We understand that other IT industries are considering similar policies of “cost sharing” for required training. Such policies would be disastrous if they became widespread.

Another worrisome aspect is that many companies are not investing in R&D, equipment, and training, which all affect their long-term future. Many are plowing their cash into stock buy-backs and some are going into debt to do so. *The Economist* (Sept. 13, 2014) said: “In 2013 38% of [U.S.] firms paid more in buy-backs than their cash-flows could support, an unsustainable position. Some American multinationals with apparently healthy global balance sheets are, in fact, dangerously lopsided. They are borrowing heavily at home to pay for buy-backs while keeping cash abroad to avoid America’s high corporate tax rate.” *Financial Times* listed six major IT companies in the top 10 engaged in buy-backs. The policy climate is drawing companies into short-term decisions that do not align with their long-term interests.

### Finding the Way Out

Education is the key to opening a path for people to move from a displaced position into a technician position that would give them productive work and a chance at rising pay, while easing joblessness and blunting the inequality between the IT elite and the rest of the workforce. Colleges and universities will not be of much help in the short run because they do not see themselves as part of the “training” side of education.

One promising means is a new kind of organization called Regional Talent Innovation Networks (RETAINs).<sup>3</sup> They are non-profit intermediaries that link K–12 schools, two-year colleges, community colleges, and workplace-based training and education.

## Coding is the basis of many technician skills in IT.

Their goal is to produce well-educated STEM talent to support a technology-driven economy. Examples include High School, Inc. in Santa Ana, CA; the Vermillion Advantage in Danville, IL; the New North in northeastern Wisconsin; New Century Careers in Pittsburgh, PA; and the Steinbeck Cluster in Salinas, CA. There are more than 1,000 RETAINs across the U.S. and around the world.

RETAINs are particularly attractive to small business owners because they offer a viable way of pooling their resources in joint programs that will inform, attract, and prepare skilled workers for IT and other growing regional industries. RETAINs link regional employers, educational institutions, and other community organizations together as a collaborative network, thereby reducing the individual company’s investment in employer-provided education and training. RETAINs promote a more positive overall regional business culture of sharing rather than stealing workers from each other. We think RETAINs will play a key role in the reeducation of workers displaced by digital automation.

Another promising means is the career academy. These high schools blend a stronger liberal arts curriculum with specific practical career education courses and internship experiences. Over 2,500 comprehensive career academies are already operating. Many are stand-alone learning communities within larger high schools. Some are stand-alone career high schools in health care, IT, and various STEM areas.

Because the demand for sub-bachelor skills is so obvious, private entrepreneurs have been starting businesses to provide inexpensive online training. The MOOC, which makes

university-level courses widely available, has not yet tackled the technician shortage. The online competency based module (OCBM) is closer to the mark and a growing number of companies are offering them.<sup>6</sup> As these technologies mature, more people will be able to get online training and be certified in a new skill set. With support from their employers, workers can also use these technologies for their continuing education.

The MOOC and OCBM demonstrate that not even the education process is exempt from automation. Before long, students whose only current choice is to enroll in a university may choose instead to enroll in a two-year college or a private company that offers such training. This could displace university faculty by depleting the flow of students seeking enrollment in college. No one is immune from automation of their jobs anymore. □

### References

1. Charette, R. Is there a US IT worker shortage? *IEEE Spectrum* (Sept. 3, 2013); <http://spectrum.ieee.org/riskfactor/computing/it/is-there-a-us-it-worker-shortage>.
2. Denning, P.J. and Frailey, D. Who are we now? *Commun. ACM* 54, 6 (June 2011), 27–30.
3. Gordon, E.E. *Future Jobs: Solving the Employment and Skills Crisis*. ABC-CLIO, 2013.
4. Government Technology. US cyber command looks to fill 6000 jobs. (Oct. 2, 2014); <http://www.govtech.com/federal/US-Cyber-Command-Looks-to-Fill-6000-Jobs.html>.
5. Thibodeau, P. IBM cuts pay by 10% for workers picked for training. *Computerworld* (Sept. 15, 2014). <http://www.computerworld.com/article/2683239/ibm-cuts-pay-by-10-for-workers-picked-for-training.html>.
6. Weise, M. and Christensen, C. *Hire Education: Mastery, Modularization, and the Workforce Revolution*. Christensen Institute (2014); <http://www.christenseninstitute.org/publications/hire/>.

**Peter J. Denning** (pjd@nps.edu) is Distinguished Professor of Computer Science and Director of the Cebrowski Institute for information innovation at the Naval Postgraduate School in Monterey, CA, is Editor of *ACM Ubiquity*, and is a past president of ACM. The author’s views expressed here are not necessarily those of his employer or the U.S. federal government.

**Edward E. Gordon** (imperialcorp@juno.com) is president of Imperial Consulting, a Chicago-based firm that advises corporations, educational institutions, workforce boards, government agencies, and trade associations on talent development and education reform. He is the author of *Future Jobs: Solving the Employment and Skills Crisis* (Praeger, 2013) and has been featured in the *Wall Street Journal*, *New York Times*, *The Futurist*, the PBS *NewsHour*, the CBS Network’s *Early Show*, and CNN.



# Computing Ethics

## Humans in Computing: Growing Responsibilities for Researchers

*Considering the role of institutional review boards in computing research.*

**F**ACEBOOK FOUND ITSELF at the center of heated debate during the summer of 2014. Researchers manipulated Facebook's News Feed feature and published a paper in the *Proceedings of the National Academy of Sciences* showing those viewing positive posts expressed more positive emotions, while those viewing negative posts expressed more negative emotions.<sup>3</sup> The paper's title proclaimed "experimental evidence of massive-scale emotional contagion" among the 689,003 people in the experiment.

Supporters claimed the results were useful, that the researchers had done nothing wrong, and that Facebook users agreed to such uses when they signed up. Critics claimed the experiment had mistreated people by including them in the research without prior knowledge or opportunity to give informed consent to their participation. Companies such as Facebook can conduct research without the oversight of institutional review boards, or IRBs. This was cited in critiques, suggesting that problems would have been avoided if an IRB had reviewed the plan. What role, if any, should IRBs play in computing research?

Research funding is increasingly predicated on human welfare, establishing a connection that is growing stronger for computing researchers.



Thinking about IRBs is useful because they have become a touchstone for ethics in research. IRBs govern much research at universities, medical centers, and other organizations. Federal research agencies sometimes require IRB approval or exemption before making awards. Some computing researchers (for example, human-computer interaction and information

technology for health treatment) have worked with IRBs for years. Researchers who work in education or with people under 18 years of age are in or are heading into the IRB zone. Computing research can show ethical leadership by getting ahead of the curve rather than merely reacting to it. For those who must now deal with IRBs this column suggests a point of view

that will help. For others it suggests a way to get out in front.

The point of view is to recognize the sound justification for the role of the IRB and the power of public opinion behind it. IRBs are the product of evolving political will regarding humane treatment of research subjects. The Nuremberg Trials gathered much public interest following WWII, and put the topic on the table.<sup>6</sup> Protocols evolved as public concern grew. The disclosure of the U.S. Public Health Service's Tuskegee Syphilis Experiment between the early 1930s and the early 1970s caused public alarm, and led to the U.S. National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research. In 1978 the commission issued "The Belmont Report: Ethical Principles and Guidelines for the Protection of Human Subjects of Research." This was soon followed by the U.S. Federal Policy for the Protection of Human Subjects (the "Common Rule"), the creation of the Office of Human Research Protections (OHRP) within the Department of Health, Education and Welfare (now Health and Human Services), and the establishment of IRBs to approve, monitor, and review research involving humans.<sup>2,4,10</sup>

Between the Tuskegee Syphilis Experiment and the Belmont Report the focus on human welfare expanded. Psychologist Stanley Milgram's experiments at Yale University in the early 1960s caused public alarm when authority figures ordered subjects to shock others electrically. No one was actually shocked, but subjects believed they had harmed others. Similarly, the public was concerned about psychologist Philip Zimbardo's experiments at Stanford University in the early 1970s in which students acting as guards in a mock prison psychologically tortured student prisoners. The Belmont Report included mental welfare of research subjects, and IRBs followed suit.

The passage of time does not necessarily reduce public concerns. The papers of Dr. John Charles Cutler disclosed that researchers with the U.S. Public Health Service deliberately infected human subjects in Guatemala with sexually transmitted diseases in

## The Facebook story suggests computing researchers should consider possible connections between their research and human welfare.

the 1940s.<sup>a</sup> President Barack Obama apologized to the government and people of Guatemala, and ordered a thorough investigation. More than half a century had elapsed since the research was done. Cutler gave his papers to the University of Pittsburgh library in 1990s. They remained unexamined until 2010 when a researcher read them and notified library leaders. The records were transferred to the National Archives and Records Administration (NARA), which informed President Obama. The events were controversial even though they were decades old.

The definition of human welfare has continued to expand. *HeLa*, an "immortal" human cell line (the cells can be reproduced indefinitely), was taken from Henrietta Lacks, a cervical cancer patient who died in 1951. *HeLa* became widely used in medical research, including that of Jonas Salk in his efforts to develop the polio vaccine. Rebecca Skloot's best-selling 2010 book explains that neither Lacks nor her family benefited from *HeLa*.<sup>9</sup> Permission to use cells in this way was not required of the patient or the family at the time. Yet in 2013, more than 60 years after the cells were taken, Lacks' descendants reached an agreement with the National Institutes of Health regarding access to *HeLa* DNA code and acknowledgment in scientific papers.<sup>8</sup> The agreement did not award financial com-

a The Cutler Papers were released online in March 2011: <http://www.archives.gov/press/press-releases/2011/nr11-94.html>.

pensation to Lacks' descendants, but the question of who benefits is now open. This area of law and policy is not settled, but the definition of human welfare is expanding.

Information technology is important for human welfare. Connecting computing research to human welfare raises important ethical issues that go beyond avoiding direct physical harm to research subjects. The regulations already include "behavior." Next steps might be finances and reputation (the latter has already arisen in Europe<sup>5</sup>). The regulatory reach of IRBs can grow: a few alterations in legislation or regulations can require funding agencies to demand that researchers seek IRB review or satisfy other requirements *before their proposals will be considered*. While regulatory reach can increase or diminish, computing researchers should get in front of the trends. The simple plea of "Trust Us" does not work. The reputations of the many researchers who know right from wrong and can make good human welfare decisions with no review can be damaged by a few who do the wrong thing and get caught. Arguments to leave researchers alone usually lose. Being proactive is smart.

Two examples illustrate contemporary ethical dilemmas involving computing research and human welfare. One is how research done in the digital world should be treated. Research done using Twitter might be like and unlike research done in the past. If new rules apply, who makes such rules? Many IRBs are grappling with this. Another is "cyberoffense," mimicking those who unlawfully hack into computer systems.<sup>7</sup> Such work might be needed to better secure computing systems against real threats, but what tests should be done, by whom, under whose authority, and for what purposes? Researchers do not become serial murderers to better understand how serial murderers behave. How is this different? How far should efforts to mimic unlawful hackers go? How should the knowledge be used? What if students become expert and unlawful hackers themselves? Such questions need attention. There are no simple answers. Computing researchers can help.

The Facebook story suggests that computing researchers should consider *possible* connections between their research and human welfare. Computing research that goes regularly to the IRB will continue to go there. What about computing research that might now be declared exempt from IRB consideration, or at least be puzzling to IRB experts? It is difficult to pin down the moving “front” between the IRB’s established territory and where the IRB will be in the future. The IRB is not the only mechanism to consider, but public opinion has tended toward more strict control of research, and the IRB is often the most experienced source of guidance available. Computing researchers should watch the IRB and think proactively about important ethical issues.

Although going through IRB review can be a disincentive to writing and submitting proposals, the history of IRBs shows sensitivity to the needs of research. Many institutions have created separate IRBs to deal with biomedical research and behavioral research in recognition of important differences between those research domains. One protocol does not fit all research. In time there might be additional IRBs created. Computing researchers should be engaged at the beginning to forestall senseless regulation and promote ethical practice. The IRB has been at the forefront of ethical discussions regarding the researcher’s “duty of care” toward research subjects and others in the broad realm of “human welfare.” There is much to be learned from the IRB. Finally, the IRB mechanism is likely to persevere and grow in importance as the *primary* device for settling matters of research and human welfare, at least in Federally supported research. Computing researchers should become closer to the IRB, not to accelerate IRB control over computing research, but to understand IRB concerns and establish a sensible and sustainable trajectory for the future.

Open issues regarding human welfare will not be settled using an authoritarian approach. Computing researchers in universities and companies cannot do whatever they like. Doctoral students and postdoctoral

fellows should be aware of science and engineering ethics. Ethical concerns must lead professional practice and regulation, not the other way around. IRBs have not discovered all the ethical issues that should be in the foreground of research. For example, there are major uncertainties regarding what constitutes “informed” consent, many of them brought on by advances in IT.<sup>1</sup> Technological capabilities and social attitudes continue to change. Uncertainties remain, and learning to manage research involving human welfare is not a one-time proposition. Many researchers who assumed they would never be included in IRB review now routinely take their proposed work to the IRB. Computing researchers have the opportunity to develop ethical directions for their work that exemplify humane and responsible conduct. To do so requires individual initiative and institutional support. This is not because IRB control over computing research is inevitable (it might not be), but because this is the right thing to do. **□**

#### References

1. Barocas, S. and Nissenbaum, H. Big data’s end-run around procedural privacy protections. *Commun. ACM* 57, 11 (Nov. 2012), 31–33.
2. Childress, J.F., Meslin, E.M., and Shapiro, H.T., Eds. *Belmont Revisited: Ethical Principles for Research with Human Subjects*. Georgetown University Press, Washington, D.C. (2005); <http://www.hhs.gov/ohrp/humansubjects/commonrule/>.
3. Kramer, A.D.L., Guillory, J.E., and Hancock, J.T. Experimental evidence of massive-scale emotional contagion through social networks. In *Proceedings of the National Academy of Sciences* 111, 8 (June 17, 2014), 8788–8790.
4. Jones, J. *Bad Blood: The Tuskegee Syphilis Experiment*. Free Press, New York, 1981; <http://www.hhs.gov/ohrp/humansubjects/guidance/belmont.html>.
5. Mantelero, A. The EU proposal for a general data protection regulation and the roots of the ‘right to be forgotten.’ *Computer Law and Security Review* 29, 3 (Mar. 2013), 229–235.
6. Marrus, M.R. *The Nuremberg War Crimes Trial 1945–46: A Documentary History*. St. Martin’s Press, Boston, 1997.
7. Nakashima, E. and Soltani, A. The ethics of hacking 101. *Washington Post Live*, (Oct. 7, 2014); [http://www.washingtonpost.com/postlive/the-ethics-of-hacking-101/2014/10/07/39529518-4014-11e4-b0ea-8141703bbf6f\\_story.html](http://www.washingtonpost.com/postlive/the-ethics-of-hacking-101/2014/10/07/39529518-4014-11e4-b0ea-8141703bbf6f_story.html).
8. National Institutes of Health. NIH. Lacks family reach understanding to share genomic data of HeLa cells; <http://www.nih.gov/news/health/aug2013/nih-07.htm>.
9. Skloot, R. *The Immortal Life of Henrietta Lacks*. Crown Publishers, New York, 2010.
10. Vollmer, S.H. and Howard, G. Statistical power, the Belmont Report, and the ethics of clinical trials. *Science and Engineering Ethics* (Dec. 2010), 675–691.

**John Leslie King** (jlkking@umich.edu) is W.W. Bishop Professor in the School of Information at the University of Michigan, Ann Arbor, MI.

## Calendar of Events

### March 2–4

**Fifth ACM Conference on Data and Application Security and Privacy**, San Antonio, TX, Sponsored: SIGSAC, Contact: Jaehong Park, Email: jaehpark@gmail.com

### March 2–5

**ACM/IEEE International Conference on Human-Robot Interaction**, Portland, OR, Sponsored: SIGAI, SIGCHI, Contact: Julie A. Adams, Email: adamsj@ieee.org

### March 4–7

**The 46<sup>th</sup> ACM Technical Symposium on Computer Science Education**, Kansas City, MO, Sponsored: SIGCSE, Contact: Kurt Eiselt, Email: eiselt@cs.ubc.ca

### March 14–18

**ASPLOS ’15: Architectural Support for Programming Languages and Operating Systems**, Istanbul, Turkey, Sponsored: ACM/SIG, Contact: Ozcan Ozturk, Email: ozturk@cs.bilkent.edu.tr

### March 14–18

**The 18th ACM Conference on Computer Supported Cooperative Work and Social Computing**, Vancouver, BC, Sponsored: SIGCHI, Contact: Andrea Forte, Email: aforte@drexel.edu

### March 18–20

**Multimedia Systems Conference 2015**, Portland, OR, Sponsored: SIGMM, SIGCOMM, SIGMOBILE, Contact: Wei Tsang Ooi, Email: ooiwt@comp.nus.edu.sg

### March 29–April 1

**ISPD’15: International Symposium on Physical Design**, Monterey, CA, Sponsored: ACM/SIG, Contact: Azadeh Davoodi, Email: adavoodi@wisc.edu

## Viewpoint

# The Real Software Crisis: Repeatability as a Core Value

*Sharing experiences running artifact evaluation committees for five major conferences.*

**W**HERE IS THE software in programming language research? In our field, software artifacts play a central role: they are the embodiments of our ideas and contributions. Yet when we publish, we are evaluated on our ability to describe informally those artifacts in prose. Often, such prose gives only a partial, and sometimes overly rosy, view of the work. Especially so when the object of discourse is made up of tens of thousands of lines of code that interact in subtle ways with different parts of the software and hardware stack on which it is deployed. Over the years, our community's culture has evolved to value originality above everything else, and our main conferences and journals<sup>a</sup> deny software its rightful place.

Science advances faster when we can build on existing results, and when new ideas can easily be measured against the state of the art. This is exceedingly difficult in an environment that does not reward the production of reusable

**If a paper makes, or implies, claims that require software, those claims must be backed up.**

software artifacts. Our goal is to get to the point where any published idea that has been evaluated, measured, or benchmarked is accompanied by the artifact that embodies it. Just as formal results are increasingly expected to come with mechanized proofs, empirical results should come with code.

Conversations about this topic inevitably get mired in discussions of *reproducibility*, which the act of creating a fresh system from first principles to duplicate an existing result under different experimental conditions. Reproducibility is an expensive undertaking and not something we are advocating. We are after repeatability, which is simply the act of checking the claims made in the paper, usually, but not only, by re-running a bundled software artifact. Repeatability is an inexpen-

sive and easy test of a paper's artifacts, and clarifies the scientific contribution of the paper. We believe repeatability should become a standard feature of the dissemination of research results. Of course, not all results are repeatable, but many are.

Researchers cannot be expected to develop industrial-quality software. There will always be a difference between research prototypes and production software. It is therefore important to set the right standard. We argue the right measure is not some absolute notion of quality, but rather how the artifact stacks up against the expectations set by the paper. Also, clearly, not all papers need artifacts. Even in software conferences, some papers contain valuable theoretical results or profound observations that do not lend themselves to artifacts. These papers should, of course, remain welcome. But if a paper makes, or implies, claims that require software, those claims should be backed up. In short, a paper should not mislead readers: if an idea has not been evaluated this should be made clear, both so program committees can judge the paper on its actual merits, and to allow subsequent authors to get the credit of performing a rigorous empirical evaluation of the paper's ideas. Lastly, artifacts can include data sets, proofs and any other by-product of the research process.

<sup>a</sup> Our central argument applies just as well, and perhaps even more strongly, to journals. However, we do not have experience creating an artifact evaluation process for journals; we also imagine that some journals might be concerned that their submission rate is sufficiently low that further obstacles would be unwelcome, though this is a weak argument for not performing a more thorough review.

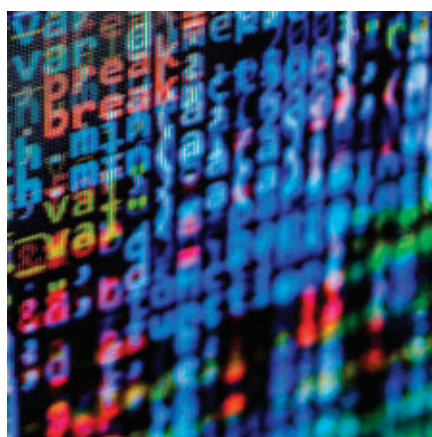
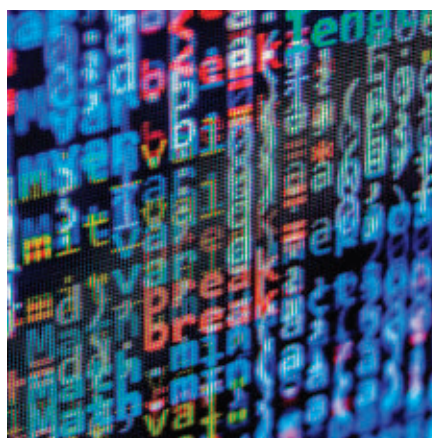
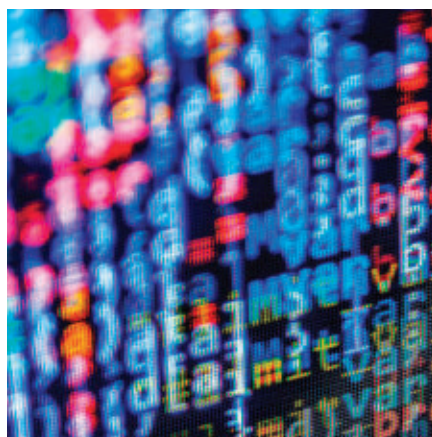
### The artifact evaluation process.

Several ACM SIGPLAN conferences (OOPSLA, PLDI, and POPL) and closely related conferences (SAS, ECOOP, and ESEC/FSE) have begun an experiment intended to move in the direction outlined here. They have initiated an artifact evaluation process that allows authors of accepted papers to submit software as well as many kinds of non-software entities (such as data sets, test suites, and models) that might back up their results.<sup>b</sup> Since 2011 we have run, or helped with, six artifact evaluation committees (AECs). The results so far are encouraging. In 2011, the ESEC/FSE conference had 14 artifact submissions (for 34 accepted papers) and seven of those met or exceeded expectations. In 2013, at ECOOP, nine out of 13 artifacts were found to meet expectations. The same year, ESEC/FSE saw a big jump in artifact submission with 22 artifacts, of which 12 were validated. At SAS, 11 out of 23 papers had artifacts. The 2014 OOPSLA conference had 21 artifacts out of 50 accepted papers, and all but three were judged adequate. In 2014, all the preceding conferences had an artifact evaluation process.

**What are the mechanics of artifact evaluation?** The design of the first artifact evaluation process (conducted by the first author with Carlo Ghezzi<sup>c</sup>) involved discussions with leaders of the software engineering community, and met with more resistance than expected. There was concern that introducing artifact evaluation into the decision-making process would be an abrupt and significant cultural change. As a result, we erected a strict separation between paper acceptance and artifact evaluation in the simplest possible way: using a temporal barrier. Only accepted papers could be submitted for evaluation and their acceptance status was guaranteed to remain unchanged.

<sup>b</sup> For pragmatic and social reasons, artifact evaluation is limited to accepted papers. Integrating artifact evaluation with paper reviewing was felt to be risky, as the standards of what constitutes a valid artifact are still evolving. From a practical perspective, the effort of evaluating a large number of artifacts would overwhelm the committee. On average, an artifact takes a day and a half to evaluate by each of the three evaluators. The process would be difficult to scale to hundreds of submissions.

<sup>c</sup> <http://cs.brown.edu/~sk/Memos/Conference-Artifact-Evaluation/>



This was a necessary compromise to get the process approved at all. In time, it is conceivable that artifact evaluation will become a part of the evaluation of most scientific results.

Initially, we judged artifacts on a five-point scale, with crisp, declarative sentences (inspired by Identify the Champion,<sup>d</sup> which many evaluators are already familiar with) accompanying each level:

- ▶ The artifact greatly exceeds the expectations set by the paper.
- ▶ The artifact exceeds the expectations set by the paper.
- ▶ The artifact meets the expectations set by the paper.
- ▶ The artifact falls below the expectations set by the paper.
- ▶ The artifact greatly falls below the expectations set by the paper.

Over time we have come to think this is too fine-grained, and have settled for the simpler criterion of whether the artifact passes muster or not. Here, “expectations” is interpreted as the claims made in the paper. For instance, if a paper claimed the implementation of a new compiler for the Java programming language, it would be reasonable for the evaluators to expect the artifact would be able to process an arbitrary Java program; on the other hand, if the paper only claimed a subset of the language, say “all loop-free Java programs,” then evaluators would have to lower their expectations accordingly.

In addition to “running” the artifact, the evaluators must read the paper and try to tweak provided inputs or create new ones, to test the limits of the artifact. The amount of effort to be invested is intended to be comparable to the time reviewers spend on evaluating a paper; in practice evaluators have reported spending between one and two days per artifact. Just like when reading a paper, the goal is not to render a definitive judgment but rather to provide a best-effort expert opinion.

### Who should evaluate artifacts?

Some have argued that evaluating artifacts is a job for the conference program committee itself. However, we believe this sits at odds with the reality of scientific reviewing. Due to high submission volumes, program

<sup>d</sup> <http://scg.unibe.ch/download/champion/>

committee members are in high demand. In addition, some of them are not always familiar with modern software tools and systems. We therefore think it best the AECs be populated by senior Ph.D. and post-doctoral researchers. This choice has several benefits. First, they are familiar with the technology needed to build and run artifacts. Second, in our experience, they respond with alacrity and write detailed reviews in a timely manner. Finally, and more subtly, we feel getting junior researchers involved in the process sends a message of its importance to those who will be future research leaders. One caution is that junior researchers can sometimes be overly eager at fault-finding, and their reviews may need moderation. This is why the AEC is chaired by senior researchers.

**What are the benefits of artifact evaluation?** The first benefit of the process is it sends a message that artifacts are valued and are an important part of the contribution of papers published in programming language conferences. Papers found to be at or above threshold get a little extra recognition, both in the proceedings and at the conference. They are marked with a special logo and distinguished in the conference proceedings. A handful of papers are selected for Distinguished Artifact Awards. Another

The ECML/PKDD'13 conference started an open science award process similar to the artifact evaluation process described here.<sup>e</sup> The SIGMOD conference evaluated repeatability from 2008 to 2011.<sup>f,g</sup> The ICERM workshop on reproducibility in computational and experimental mathematics produced a report that argues for a culture shift.<sup>h</sup> Journals such as *Biostatistics* are recognizing papers that are accompanied by artifacts.<sup>i</sup>

e <http://www.ecmlpkdd2013.org/open-science-award/>.

f Manegold, S. et al. Repeatability and Workability Evaluation of SIGMOD 2009. *SIGMOD Record*, September 2009.

g [http://www.sigmod2011.org/calls\\_papers\\_sigmod\\_research\\_repeatability.shtml](http://www.sigmod2011.org/calls_papers_sigmod_research_repeatability.shtml).

h [http://icerm.brown.edu/html/programs/topical/tw12\\_5\\_rcem/icerm\\_report.pdf](http://icerm.brown.edu/html/programs/topical/tw12_5_rcem/icerm_report.pdf).

i [http://www.oxfordjournals.org/our\\_journals/biosts/for\\_authors/msprep\\_submission.html](http://www.oxfordjournals.org/our_journals/biosts/for_authors/msprep_submission.html).

## Artifact evaluation encourages authors to produce reasonable artifacts, which are the cornerstone of future research.

benefit comes from the reviews themselves: several authors have confirmed the evaluators provided valuable feedback and even small bug fixes on the artifacts and on their packaging. At ECOOP 2013, for instance, some authors even claimed the artifact reviews were more useful than the reviews of the paper. For the scientific community at large, artifact evaluation encourages authors to produce reusable artifacts, which are the cornerstone of future research.

### Should artifacts be published?

While there are many good reasons for making the artifact available, there are also arguments against making artifacts public:

- ▶ The artifact may have been produced in a company and may therefore be regarded as proprietary.

- ▶ The data used in the paper's experiments may be proprietary or have high privacy needs.


- ▶ The artifact may depend on expensive or proprietary platforms that are difficult or impossible for anyone but the authors to access.

- ▶ By making the tools public, it becomes easy for others to continue that line of research, which reduces the payoff for the original researchers.

Reasonable people have come to opposite conclusions on each of these issues. In some cases, a different incentive structure might help. At any rate, it is clear that in some situations repeatability may be off limits; but these cases seem rare enough that they should not dominate the discussion.

In the long term, we would like to see evaluated artifacts be made public by mandate, as SAS 2013 did. Even as it remains optional, for authors

who do wish to publish them, there remains the problem of how and where. ACM's digital library would be a natural host, and recent changes have made it possible for authors to deposit artifacts there without surrendering their copyright. Yet, the interface to the digital library is less than optimal; there are also problems with the current terms. We would prefer to use technologies that better support accessing artifacts. Furthermore, the digital library only hosts static artifacts; it would be worthwhile for it to consider combining forces with resources such as [runmycode.org](http://runmycode.org) and [researchcompendia.org](http://researchcompendia.org).

**We have come a long way.** In our efforts to become more "scientific," we have moved away from papers that simply report on software projects to demanding that papers distill the novel contributions of these projects. In the process, however, we may have shifted too far, even as natural science itself has taken a lead on demanding repeatability, data sets, and public access to software; demands we recognize the need for and hence should have spearheaded. We should let the pendulum swing back to a happy medium between scientific contributions and software contributions, recognizing that ultimately, software is indeed the single most distinctive contribution our discipline has to make. So we should embrace it rather than act as if we are ashamed of it. While we report on one particular experiment in the area of programming language research, many other areas in computer science are looking at some of the same issues. References to other initiatives are included in the sidebar; also see <http://www.artifact-eval.org> 

**Shriram Krishnamurthi** ([sk@cs.brown.edu](mailto:sk@cs.brown.edu)) is a professor of computer science at Brown University in Providence, RI.

**Jan Vitek** ([j.vitek@neu.edu](mailto:j.vitek@neu.edu)) is a professor of computer science at Northeastern University in Boston, MA.

The authors thank Andreas Zeller for taking the personal risk involved in initiating this Viewpoint. We thank Matthias Hauswirth for his enthusiasm and artwork, and Matthias, Steve Blackburn, and Camil Demetrescu for several good conversations. We thank our AEC co-chairs Eric Eide, Erik Ernst, and Carlo Ghezzi for their hard work. Most of all, we thank the AEC members for their diligent efforts, often above and beyond the call of duty, and the authors for giving the AEC members something to evaluate.

Copyright held by authors.

## Viewpoint

# Why Did Computer Science Make a Hero Out of Turing?

*Comparing the legacy of Alan Turing in computer science with that of Carl Friedrich Gauss in mathematics.*

**E**VERY DISCIPLINE THAT COMES of age consecrates its own roots in the process. In footnotes, anecdotes, and names of departmental buildings, occasions are found to remember and celebrate personalities and ideas that a discipline considers its own. A discipline needs heroes to help create a narrative that legitimizes and fortifies its own identity. Such a narrative hardly reflects the complexity of historical reality. Rather, it echoes the set of preferences and programmatic choices of those in charge of a discipline at a given moment in a given place. Each name that gets integrated into an officialized genealogy is the result of discussions and negotiations, of politics and propaganda.

To the general public, the genealogies of physics and mathematics are probably more familiar than that of computer science. For physics we go from Galileo via Newton to Einstein. For mathematics we begin with Euclid and progress over Descartes, Leibniz, Euler and Gauss up to Hilbert. Computer science by contrast is a relatively young discipline. Nevertheless, it is already building its own narrative in which Alan Turing plays a central role.

In the past decennia, and especially during the 2012 centenary celebration of Turing, his life and legacy received an increasing amount of attention.



Recently, *Communications* published two columns in which Turing's legacy is put into a more historical context.<sup>7,9</sup> We continue this line of research by focusing on how Turing functioned as a hero within the formation of computer science. We will do so here by comparing the consecration of Turing with that of Gauss in mathematics.

### Making Gauss a Hero

In the early 19<sup>th</sup> century, the Prussian minister Wilhelm von Humboldt sought to introduce mathematics as a discipline per se in higher education. To do so, he needed an icon to represent German mathematics. He turned to the one German who had been praised in a report on the progress of mathematics to emperor Napoleon: Carl Friedrich Gauss (1777–1855). Also, the new generation of mathematicians favored a conceptual approach over computations and saw Gauss as the herald of this new style of mathematics. As such, Gauss became synonymous with German mathematics for both political as well as more internal reasons.

Toward the end of the 19<sup>th</sup> century, the prominent mathematician Felix Klein developed this Gauss image into a programmatic vision. From 1886 onward, he had started to actively transform Göttingen's mathematics department into the world's foremost mathematical center. He promoted a close alliance between pure and applied mathematics and got cooperation with the industry on the way. On a national scale, he worked for the professionalization of mathematics education. To shape this disciplinary empire, Klein, too, used Gauss.

In his 1893 address to the first International Congress for Mathematics in Chicago, Klein talked about the latest developments in mathematics and spoke of: *a return to the general Gaussian programme [but] what was formerly begun by a single master-mind [...] we must now seek to accomplish by united efforts and cooperation.*<sup>10</sup>

The edition of Gauss' collected works (1869–1929) provided an abundance of historical material that Klein used to build an image of Gauss supporting his personal vision on mathematics. Klein portrayed Gauss as the lofty German who was able to pursue practical studies because of his theoret-

ical research, a portrayal that, although very influential, was biased nonetheless.

In the 20<sup>th</sup> century, Klein's interpretation of Gauss was picked up by the international mathematical community and was modified accordingly. In the U.S., following Klein's 1893 address, Gauss's fertile combination of pure and applied struck a note for a mathematical community that often worked closely in alliance with industry.<sup>11</sup> In France, after World War II, the Bourbaki-group emphasized the abstraction of Gauss's work that transcended national boundaries and had helped pave the way for their structural approach to mathematics. However, in contrast with the Kleinian "pure mathematician," Gauss was also "rediscovered" after the birth of the digital computer as a great calculator and explorer of the mathematical discourse.<sup>4</sup>

### Making Turing a Hero

Just like Gauss was instrumental to Humboldt and Klein to further the institutionalization of mathematics, Turing played a similar role in the professionalization of the ACM in the 1960s. This goes back to the 1950s, when some influential ACM members, including John W. Carr III, Saul Gorn, and Alan J. Perlis, wanted to connect their programming feats to modern logic. Stephen Kleene's *Introduction to Metamathematics* (1952), which contained a recast account of Turing's 1936 paper "On computable numbers," was an important source.

In 1954, Carr recommended programmers to deal with "the generation of systems rather than the systems themselves" and with "the 'generation' of algorithms by other algorithms," and hence with concepts akin to meta-

## The first wave of recognition that Turing received posthumously is but a ripple when compared to the second wave.

mathematics.<sup>3</sup> Similarly, around 1955, Gorn became accustomed to viewing a universal Turing machine as a conceptual abstraction of the modern computer (see, for example, Gorn<sup>8</sup>). By the end of the 1950s, Carr and Gorn explicitly used Turing's universal machine to express the fundamental interchangeability of hardware and language implementations. Turing's 1936 theory thus helped influence ACM members to articulate a theoretical framework that could accommodate for what programmers had been accomplishing independently of metamathematics.<sup>6</sup>

In 1965, ACM Vice President Anthony Oettinger (who had known Turing personally), and the rest of ACM's Program Committee proposed that an annual "National Lecture be called the Allen [sic] M. Turing Lecture."<sup>1</sup> Lewis Clapp, the chairman of the ACM Awards Committee, collected information on the award procedures "in other professional societies." In 1966 he wrote: *[a]n awards program [...] would be a fitting activity for the Association as it enhances its own image as a professional society. [...] [I]t would serve to accentuate new software techniques and theoretical contributions. [...] The award itself might be named after one of the early great luminaries in the field (for example, "The Von Neuman [sic] Award" or "The Turing Award", etc.)<sup>2</sup>*

ACM's first Turing Awardee in 1966 was Perlis, a well-established computer scientist, former president of the ACM, and close colleague of Carr and Gorn. Decorating Perlis is in hindsight thus rather unsurprising. Turing, by contrast, was not well known in computing at large, even though his 1936 universal machine had become a central concept for those few who wanted to give computer programming a theoretical impetus and also a professional status.<sup>a</sup>

The first wave of recognition that Turing received posthumously with the Turing award in 1966 is but a ripple when compared to the second wave.

a We speculate that Turing was preferred over von Neumann, because the latter was associated with hardware engineering rather than with theoretical foundations of programming. Moreover, it might be that for the more liberally minded Carr, Gorn, and Perlis, von Neumann was too strongly associated with conservative Cold War politics. There were other potential candidates as well, such as Emil Post. Historians are now starting to investigate these matters (see, for example, Daylight<sup>7</sup>).



This started in the 1970s with the disclosure of some of Turing's war work for the Allies, followed by Andrew Hodges' authoritative 1983 biography, which also added a personal dimension to Turing's story: his life as a gay man in a homophobic world. This made Turing also known outside of computer science. The second wave culminated in the 2012 Turing centenary celebrations that nurtured the perception of Turing as the inventor of the modern computer and artificial intelligence. Some even claim Turing anticipated the Internet and the iPhone.

The year 2012 was full of activities: there were over 100 academic meetings, plaques, documentaries, exhibitions, performances, theater shows, and musical events. The celebrations also brought together a group of people with diverse backgrounds and promoted computer science to the general public, an achievement of which the longer-term impact has yet to be awaited.<sup>12</sup> A discipline has its heroes for good reasons.

As Hodges' biography shows, Turing's work was multifaceted. Not only did Turing contribute in 1936 to the foundations of mathematics, which later proved to be fundamental for theoretical computer science, he also worked at Bletchley Park during World War II to help break the Enigma. He became an experienced programmer of the Ferranti Mark I for which he wrote a programmer's manual and even designed a computer, known as the ACE. He reflected on thinking machines and contributed to the field of morphogenesis.

It is therefore not surprising that for many today the multidisciplinary nature of computer science is personified in Turing who achieved all these different things in one short lifespan. Along these lines, Barry Cooper, the driving force behind the Turing centenary, said the following in 2012: *The mission of [the Turing Centenary] was to address concerns about how science was fragmenting. We wanted to return to more joined-up thinking about computability and how it affects everyone's life. More generally, too, the Turing Year was important in highlighting the need for fundamental thinking.*<sup>12</sup>

From this perspective, Turing's theoretical work gives new impetus to the sciences as a whole, not just to computer science per se. The recent volume

## Is Turing for computer science what Gauss is for mathematics?

*Alan Turing—His Work and Impact*<sup>5</sup>—Turing's collected papers cum essays from renowned scientists—also wants to bring this point home. It echoes even on the political level. The House of Commons has considered naming the new Technology and Innovation elite centers after Turing. According to the chairman of the Science and Technology Committee, "There isn't a discipline in science that Turing has not had an impact upon." As such, computer science, and especially theoretical computer science with its focus on computability, becomes the connecting discipline among the other sciences, and thereby turns into a fundamental science, not unlike mathematics.

The focus on computability and fundamental thinking is certainly not accidental. To a large extent the drive behind the Turing Year came from theoreticians. They do not ignore that Turing also worked in engineering. However, many of them argue that Turing must have invented the computer because of his theoretical 1936 paper. According to this view on science and technology, also present in Klein's Gauss interpretation, theory precedes practice.

### Looking Backward into the Future

Over the past century, the one-dimensional image of Gauss has been replaced by a multitude of images. This shows a discipline in constant evolution assesses its own identity through its heroes and allows for a multiplicity of readings. Certainly, each reading may further the agenda of a particular community, but the diversity of all images taken together, all grounded in some way in Gauss' legacy, positively stimulates the openness and generosity of a field.

Is Turing for computer science what Gauss is for mathematics? Computer science, as its histories show, has many origins, and this should be fostered.

In this sense, the variety of topics and the diversity of approaches of Turing's work, embracing both the practical and the theoretical, reflects an essential aspect of computer science. However, if one celebrates Turing mainly because of his theoretical work, one runs the risk of increasing already existing divides. Instead of favoring one reading of Turing and crowding out others, why not view Turing's own accomplishments as an invitation? The historian could integrate Turing into a more complex historical account. The computer scientist could look back and reflect on the state of computer science, finding new ways of rapprochement between the many branches of computer science, between theory and practice. **□**

### References

1. ACM Council Meeting, August 27, 1965. Available from the "Saul Gorn Papers," the University of Pennsylvania Archives (unprocessed collection).
2. ACM Council Meeting, August 29, 1966. Available from the "Saul Gorn Papers," the University of Pennsylvania Archives (unprocessed collection).
3. Brown, J. and Carr, J.W. III. Automatic programming and its development on the MIDAC. In *Symposium on Automatic Programming for Digital Computers*, Washington D.C., May 1954. Office of Naval Research, Department of the Navy, 84–97.
4. Bullynck, M. Reading Gauss in the computer age: On the U.S. reception of Gauss's number theoretical work. *Archive for the History of the Exact Sciences* 65, 5 (2009), 553–580.
5. Cooper, B.S. and van Leeuwen, J., Eds. *Alan Turing—His Work and Impact*. Elsevier, 2013.
6. Daylight, E.G. Towards a Historical Notion of "Turing—The Father of Computer Science": To appear in *History and Philosophy of Logic*; <http://www.dijkstrascry.com/TuringPaper>.
7. Daylight, E.G. A Turing Tale. *Commun. ACM* 57, 10 (Oct. 2014).
8. Gorn, S. Real solutions of numerical equations by high speed machines. Technical Report 966, Ballistic Research Laboratories, October 1955. Available from the "Saul Gorn Papers" from the University of Pennsylvania Archives (unprocessed collection).
9. Haigh, T. Actually, Turing did not invent the computer. *Commun. ACM* 57, 1 (Jan. 2014), 46–51.
10. Klein, F. The present state of mathematics. In *Mathematical Papers read at the International Mathematical Congress*, held in Connection with the World's Columbian Exposition Chicago (1893), 133–135.
11. Parshall, K.H. and Rowe, D.E., Eds. *The Emergence of the American Mathematical Research Community, 1876–1900*. J.J. Sylvester, Felix Klein, and E.H. Moore. AMS, 1994.
12. Underwood, S. The Alan Turing year leaves a rich legacy. *Commun. ACM* 56, 10 (2013), 24–25.

**Maarten Bullynck** (maarten.bullynck@univ-paris8.fr) is associate professor at Département de mathématiques et histoire des sciences, Université Paris 8. He is currently on sabbatical leave at the research Sphère (UMR 7219, Paris) funded by the CNRS.

**Edgar G. Daylight** (egdaylight@dijkstrascry.com)—also known as Karel Van Oudheusden—has a Ph.D. from KULeuven in Belgium and is a researcher in software engineering and the history of computing at Utrecht University in the Netherlands.

**Liesbeth De Mol** (liesbeth.demol@univ-lille3.fr) is a CNRS researcher at the Université de Lille 3, France. She is the president of the international DHST commission for the History and Philosophy of Computing ([www.hapoc.org](http://www.hapoc.org)).

Copyright held by authors.

Q Article development led by [acmqueue](http://queue.acm.org)  
queue.acm.org

## Bad protocol, bad politics.

BY POUL-HENNING KAMP

# HTTP/2.0 — The IETF Is Phoning It In

A VERY LONG time ago—in 1989—Ronald Reagan was president, albeit only for the final 19½ days of his term. And before the year was over Taylor Swift had been born, and Andrei Sakharov and Samuel Beckett had died.

In the long run, the most memorable event of 1989 will probably be that Tim Berners-Lee hacked up the HTTP protocol and named the result the “World Wide Web.” (One remarkable property of this name is that the abbreviation “WWW” has three times as many syllables and takes longer to pronounce.)

Berners-Lee’s HTTP protocol ran on 10Mbit/s Ethernet, and coax cables, and his computer was a NeXT Cube with a 25MHz clock frequency. Some 26 years later, my laptop CPU is 100 times faster and has 1,000 times as much RAM as Berners-Lee’s machine had, but the HTTP protocol is still the same.

A few weeks ago, the Internet Engineering Steering Group (IESG) asked for “Last Call” comments on new “HTTP/2.0” protocol (<https://tools.ietf.org/id/draft-ietf-httpbis-http2>) before blessing it as a “Proposed Standard.”



Some will expect a major update to the world’s most popular protocol to be a technical masterpiece and textbook example for future students of protocol design. Some will expect that a protocol designed during the Snowden revelations will improve their privacy. Others will more cynically suspect the opposite. There may be a general assumption of “faster.” Many will probably also assume it is “greener.” And some of us are jaded enough to see the “2.0” and mutter: “Uh-oh, Second Systems Syndrome.”

The cheat sheet answers are: no, no, probably not, maybe, no, and yes.

If that sounds underwhelming, it’s because it is.

HTTP/2.0 is not a technical masterpiece. It has layering violations,



inconsistencies, needless complexity, bad compromises, misses a lot of ripe opportunities, and more. I would flunk students in my (hypothetical) protocol design class if they submitted it. HTTP/2.0 also does not improve your privacy. Wrapping HTTP/2.0 in SSL/TLS may or may not improve your privacy, as would wrapping HTTP/1.1 or any other protocol in SSL/TLS. But HTTP/2.0 itself does nothing to improve your privacy. This is almost triply ironic, because the major drags on HTTP are the cookies, which are such a major privacy problem the European Union has legislated a notice requirement for them. HTTP/2.0 could have done away with cookies, replacing them instead with a client-controlled

session identifier. That would put users squarely in charge of when they want to be tracked and when they don't—a major improvement in privacy. It would also save bandwidth and packets. But the proposed protocol does not do this.

The good news is that HTTP/2.0 probably does not reduce your privacy either. It does add a number of “fingerprinting” opportunities for the server side, but there are already so many ways to fingerprint via cookies, Java Script, Flash, among others, that it probably does not matter.

You may perceive webpages as loading faster with HTTP/2.0, but probably only if the content provider has a global network of servers. The individual com-

puters involved, including your own, will have to do more work, in particular for high-speed and large objects like music, TV, and movies. Nobody has demonstrated a HTTP/2.0 implementation that approached contemporary wire speeds. Faster? Not really.

That also answers the question about the environmental footprint: HTTP/2.0 will require a lot more computing power than HTTP/1.1 and thus cause increased CO<sub>2</sub> pollution adding to climate change. You would think a protocol intended for tens of millions of computers would be the subject of some green scrutiny, but surprisingly—at least to me—I have not been able to find any evidence the IETF considers environmental impact at all—ever.

And yes, Second Systems Syndrome is strong.

Given this rather mediocre grade sheet, you may be wondering why HTTP/2.0 is even being considered as a standard in the first place.

### The Answer Is Politics

Google came up with the SPDY protocol, and since they have their own browser, they could play around as they choose to, optimizing the protocol for their particular needs. SPDY was a very good prototype, which showed clearly there was potential for improvement in a new version of the HTTP protocol. Kudos to Google for that. But SPDY also started to smell a lot like a “walled garden” to some people, and more importantly to other companies, and politics surfaced.

The IETF, obviously fearing irrelevance, hastily “discovered” the HTTP/1.1 protocol needed an update, and tasked a working group with preparing it on an unrealistically short schedule. This ruled out any basis for the new HTTP/2.0 other than the SPDY protocol. With only the most hideous of SPDY’s warts removed, and all other attempts at improvement rejected as “not in scope,” “too late,” or “no consensus,” the IETF can now claim relevance and victory by conceding practically every principle ever held dear in return for the privilege of rubber-stamping Google’s initiative.

But the politics does not stop there.

The reason HTTP/2.0 does not improve privacy is the big corporate backers have built their business model on top of the lack of privacy. They are very upset about NSA spying on just about everybody in the entire world, but they do not want to do anything that prevents them from doing the same thing. The proponents of HTTP/2.0 are also trying to use it as a lever for the “SSL anywhere” agenda, despite the fact that many HTTP applications have no need for, no desire for, or may even be legally banned from using encryption.

### Your Country, State, or County Emergency Webpage?

Local governments have no desire to spend resources negotiating SSL/TLS with every single smartphone in their area when things explode, riv-



**The reason HTTP/2.0 does not improve privacy is the big corporate backers have built their business model on top of the lack of privacy.**



ers flood, or people are poisoned. Big news sites similarly prioritize being able to deliver news over being able to hide the fact they are delivering news, particularly when something big happens. (Has everybody in IETF forgotten CNN’s exponential traffic graph from 14 years ago?)

The so-called multimedia business, which amounts to about 30% of all traffic on the Net, expresses no desire to be forced to spend resources on pointless encryption. There are even people who are legally barred from having privacy of communication: children, prisoners, financial traders, CIA analysts, and so on. Yet, despite this, HTTP/2.0 will be SSL/TLS only, in at least three out of four of the major browsers, in order to force a particular political agenda. The same browsers, ironically, treat self-signed certificates as if they were mortally dangerous, despite the fact they offer secrecy at trivial cost. (*Secrecy* means only you and the other party can decode what is being communicated. *Privacy* is secrecy with an identified or authenticated other party.)

History has shown overwhelmingly that if you want to change the world for the better, you should deliver good tools for making it better, not policies for making it better. I recommend that anybody with a voice in this matter turn their thumbs down on the HTTP/2.0 draft standard: It is not a good protocol and it is not even good politics. □

### Related articles on [queue.acm.org](http://queue.acm.org)

#### Making the Web Faster with HTTP 2.0

Ilya Grigorik

<http://queue.acm.org/detail.cfm?id=2555617>

#### Better, Faster, More Secure

Brian Carpenter

<http://queue.acm.org/detail.cfm?id=1189290>

#### The Software Industry IS the Problem

Poul-Henning Kamp

<http://queue.acm.org/detail.cfm?id=2030258>

**Poul-Henning Kamp** ([phk@FreeBSD.org](mailto:phk@FreeBSD.org)) is one of the primary developers of the FreeBSD operating system, which he has worked on from the very beginning. He is widely unknown for his MD5-based password scrambler, which protects the passwords on Cisco routers, Juniper routers, and Linux and BSD systems.

Copyright held by author.  
Publication rights licensed to ACM. \$15.00.

---

## Revisiting Schorre's 1962 compiler-compiler.

---

BY DAVE LONG

---

# META II: Digital Vellum in the Digital Scriptorium

SOME PEOPLE DO living history—reviving older skills and material culture by reenacting Waterloo or knapping flint knives. One pleasant rainy weekend in 2012, I set my sights a little more recently and settled in for a little meditative retro-computing, circa 1962, following the ancient mode of transmission of

knowledge: lecture and recitation—or rather, grace of living in historical times, lecture (here, in the French sense, reading) and transcription (or even more specifically, grace of living post-Post, lecture and reimplementation).

Fortunately, for my purposes, Dewey Val Schorre's paper<sup>10</sup> on META II was, unlike many more recent digital artifacts, readily available as a digital scan.

META II was a “compiler-compiler,” which is to say that when one suspects a production compiler might be a rather large project to write in assembly—and especially if one were in an era in which commercial off-the-shelf, let alone libre and open source, compilers were still science fiction—then it makes sense to aim for an in-

termediate target: something small enough to be hand-coded in assembly, yet powerful enough for writing what one had been aiming for in the first place.

Just as mountain climbers during the golden age of alpinism would set up and stock a base camp before attempting the main ascent, and later expeditions could derive benefit from infrastructure laboriously installed by a prior group, the path to the language ecosystem we now use (cursing only on occasion) was accomplished in a series of smaller, more easily achievable, steps. Tony Brooker (who already in 1958 was faced with the “modern” problem of generating decent code when memory access will


incur widely varying latencies) wrote the compiler-compiler<sup>2</sup> (of which Johnson's later, more famous one was "yet another"<sup>6</sup>) to attack this problem in the early 1960s. According to Doug McIlroy, Christopher Strachey's GPM (general-purpose macrogenerator—a macroexpander of the same era) was only 250 machine instructions, yet it was sufficient to enable Martin Richards's BCPL (Basic Combined Programming Language) implementation, later inspiring Ken Thompson to bootstrap C via B, eventually leading to the self-hosting native-code-generating tool chains we now take for granted.

A horse can pull more than a man, but by exploiting leverage, Archimedes can, with patience, move what Secretariat could not. META II is a fine example of a field-improvised lever: one can see how the beam has been roughly attached to the fulcrum and feel how the entire structure may be springier than one would like, but in the end, no matter how unpolished, it serves to get the job done with a minimum of fuss.


### Why Study META II?

1. There is not much to examine.
2. There is not much to examine because its parts are simply defined.
3. It enables significant consequences.

I will not go into detail, as nearly all of the interest in this exercise comes from doing it yourself. Programming (when not constrained, as it often is in our vocation, by economic concerns) is not a spectator sport. Donald Knuth, who says a simple one-pass assembler should be an afternoon's finger exercise, might wish to make some additional plans to fill his weekend; it might take closer to four or five evenings if you must first refresh dim memories of a university compiler course. Instead, I will describe the general route of my ascent and why I am confident I arrived at the same summit that Schorre described well before my birth. By following Schorre's text, possibly aided by mine, you should also find climbing this peak to be an easy and enjoyable ascent. (An alternative for the hardcore: following the Feynman method, ask yourself one question: What is the square root of



**META II is a fine example of a field-improvised lever: one can see how the beam has been roughly attached to the fulcrum and feel how the entire structure may be springier than one would like, but in the end, it serves to get the job done with a minimum of fuss.**



a compiler?, then head up the mountain without a guide.)

On first reading, Schorre's text may seem horribly naive. We have the benefit of a half-century of experience and a different vocabulary. However, just as it is often amazing how much our fathers seem to have learned in the time between when we turned 14 and when we turned 21, it becomes easy to admire what Schorre accomplished as we follow in his footsteps.

Digression: In examining medieval texts on horses, it is very clear that while equitation has changed very little in the intervening centuries, veterinary science has made giant strides. With this distinction between art and technique in mind—and being thankful that Schorre's text is, albeit in a typewriter font, neither in medieval French nor, worse, handwritten Fraktur—we can take advantage of hindsight to separate the informatics from the technical artifacts of having run on an IBM 1401 (end of digression).

Here is a smattering of the more striking passages to be found:

► "Although sequences can be defined recursively, it is more convenient and efficient to have a special operator for this purpose." With hindsight, we smile and nod as we recognize the Kleene star (cf. the "Thompson construction" *infra*).

► "These assemblers all have the same format, which is shown as:

```
LABEL CODE ADDRESS
1- -6 8- -10 12- -70."
```

Having grown up after the popularity of fixed column formats, I was introduced to the concept that other people might compute in other ways during high school at a summer job: upon attempting to write a PL/I "hello world" under CMS, I had to bring in older and wiser help who shook their heads, stroked their beards, and gravely informed me all that needed to be done was to shift my code right one or two spaces, so it would no longer start in what was obviously the "comment" column.

► "Repeated executions, whether recursive or externally initiated, result in a continued sequence of generated labels. Thus all syntax equations contribute to the one sequence." In the modern style, or even in the late 1960s if you were Rod Burstall (his Cartesian

product<sup>4</sup>), you might call this monadic composition. In the days of small memories and essentially linear card decks, the flattened sequence was the norm rather than the exception, and in our times Rick Hehner’s bunches<sup>5</sup> are a good example of a case where flattening can make the formulae of “formal methods” more easily manipulable than normally nestable sets.

Note that it has taken only two pages for Schorre to describe what we need for META II. The remainder of the article focuses on a description of VALGOL, which might make a suitable destination for another day. Let us take a brief pause, however, to examine a couple of points:

► “The omission of statement labels from the VALGOL I and VALGOL II seems strange to most programmers. This was not done because of any difficulty in their implementation, but because of a dislike for statement labels on the part of the author. I have programmed for several years without using a single label, so I know they are superfluous from a practical, as well as from a theoretical, standpoint. Nevertheless it would be too much of a digression to try to justify this point here.” History agrees the digression would have been superfluous; indeed, now it seems strange that it then seemed strange. *Tempora mutantur, nos et mutamur in illis* (times change, and we change with them).

► Finally, Schorre discusses the problem of backup vs. no backup, which is still a current topic, as the recent popularity of the parsing expression grammar (PEG) and other parsers will attest. In our times, however, we are not so interested in avoiding backup, but in avoiding the need to start at the beginning and process linearly until we reach the end. Luckily for compiler writers, whether or not a production can be matched by an empty string is a property that can be determined by divide and conquer... but it is one of the few<sup>1</sup> that are tackled so simply.

The heart of the matter comes in figures 5 and 6 in the original article, “The META II Compiler Written in its Own Language” (Figure 1 in this article) and “Order List of the META II Machine” (figures 2 through 4 here). Now, it would certainly be possible to

follow in Schorre’s footsteps directly, using the traditional bootstrap:

0. Hand-code the META II machine—this is basically an assembler-like virtual machine: in other words, a

glorified left-fold (mine was about 66 lines of Python).

1. Hand-translate the META II productions to the machine language (211 lines of m2vm opcodes).

**Figure 1. The META II compiler written in its own language (Figure 5 from Schorre’s original paper).**

```
.SYNTAX PROGRAM

OUT1 = '*1' .OUT('GN1') / '*2' .OUT('GN2') /
'*' .OUT('CI') / .STRING .OUT('CL '*)..

OUTPUT = ('.OUT' '('
$ OUT1 ')') / '.LABEL' .OUT('LB') OUT1) .OUT('OUT')..

EX3 = .ID .OUT('CLL' *) / .STRING
.OUT('TST' *) / '.ID' .OUT('ID') /
'.NUMBER' .OUT('NUM') /
'.STRING' .OUT('SR') / '(' EX1 ')' /
'.EMPTY' .OUT('SET') /
'$' .LABEL *1 EX3
.OUT('BT ' *1) .OUT('SET')..

EX2 = (EX3 .OUT('BF ' *1) / OUTPUT)
$(EX3 .OUT('BE') / OUTPUT)
.LABEL *1 ..

EX1 = EX2 $('/' .OUT('BT' *1) EX2 )
.LABEL *1 ..

ST = .ID .LABEL * '=' EX1
'.' .OUT('R')..

PROGRAM = '.SYNTAX' .ID .OUT('ADR' *)
$ ST '.END' .OUT('END')..

.END
```

**Figure 2. Order list of the META II machine (Figure 6.1 in Schorre’s original paper).**

Machine Codes		
TST STRING	TEST	After deleting initial blanks in the input string, compare it to the string given as argument. If the comparison is met, delete the matched portion from the input and set switch. If not met, reset switch.
ID	IDENTIFIER	After deleting initial blanks in the input string, test if it begins with an identifier; that is, a letter followed by a sequence of letters and/or digits. If so, delete the identifier and set switch. If not, reset switch.
NUM	NUMBER	After deleting initial blanks in the input string, test if it begins with a number. A number is a string of digits which may contain imbedded periods, but may not begin or end with a period. Moreover, no two periods may be next to one another. If a number is found, delete it and set switch. If not, reset switch.
SR	STRING	After deleting initial blanks in the input string, test if it begins with a string; that is, single quote followed by a sequence of any characters other than a single quote followed by another single quote. If a string is found, delete it and set switch. If not, reset switch.
CLL AAA	CALL	Enter the subroutine beginning in location AAA. If the top two terms of the stack are blank, push the stack down by one cell. Otherwise, push it down by three cells. Set a flag in the stack to indicate whether it has been pushed by one or three cells. This flag and the exit address go into the third cell. Clear the top two cells to blanks to indicate they can accept addresses which may be generated within the subroutine.

Figure 3. This is Figure 6.2 in Schorre's original paper.

R	RETURN	Return to the exit address, popping up the stack by one or three cells according to the flag. If the stack is popped by only one cell, then clear the top two cells to blanks, because they were blank when the subroutine was entered.
SET	SET	Set branch switch on.
B AAA	BRANCH	Branch unconditionally to location AAA.
BT AAA	BRANCH IF TRUE	Branch to location AAA if switch is on. Otherwise, continue in sequence.
BF AAA	BRANCH IF FALSE	Branch to location AAA if switch is off. Otherwise, continue in sequence.
BE	BRANCH TO ERROR IF FALSE	Halt if switch is off, otherwise, continue in sequence.
CL STRING	COPY LITERAL	Output the variable length string given as the argument. A blank character will be inserted in the output following the string.
CI	COPY INPUT	Output the last sequence of characters deleted from the input string. This command may not function properly if the last command which could cause deletion failed to do so.
GN1	GENERATE 1	This concerns the current label 1 cell; that is, the next to top cell in the stack, which is either clear or contains a generated label. If clear, generate a label and put it into that cell. Whether the label has just been put into the cell or was already there, output it. Finally, insert a blank character in the output following the label.
GN2	GENERATE 2	Same as GN1, except that it concerns the current label 2 cell; that is, the top cell in the stack.
LB	LABEL	Set the output counter to card column 1.
OUT	OUTPUT	Punch card and reset output counter to card column 8.

Figure 4. This is Figure 6.3 from Schorre's original paper.

Constant and Control Codes		
ADR IDENT	ADDRESS	Produces the address that is assigned to the given identifier as a constant.
END	END	Denotes the end of the program.

2. Machine-translate the META II productions to the machine language (using the output from step 1).

Note that Schorre's character set does not include “;” hence his quasi-BNF (Backus-Naur Form) is written within the sequence “.”. Those in search of verisimilitude may wish to use a keypunch simulator to create a “deck” from Figure 1. Type-ahead is anachronistic, however, so if you are going to wear the hairshirt, it may be better to try talking someone else into being your keypunch operator.

Before condemning APL for excessive terseness, you may want to remember both that it was formed before standard character sets, and that at 110 baud, you have much more time to think about each character typed than you do with an autocompleting IDE (integrated development

environment). Before condemning Pascal for excessive verbosity, you may wish to recall the Swiss keyboard has keycaps for the five English vowels, as well as the French accented vowels and German unlauded vowels, and hence does not offer so much punctuation. Before condemning Python and Haskell for whitespace sensitivity, recall that Peter Landin came up with the “offside rule” in 1966,<sup>7</sup> which “is based on vertical alignment, not character width, and hence is equally appropriate in handwritten, typeset, or typed texts.” This was not only prescient with regard to the presentation of code in variable-width fonts, but presumably also catered to the then-common case of one person keypunching code that had been handwritten on a coding sheet by a different person.

As Schorre himself notes, because of the fixpoint nature of this process, it can, if one is fortunate, be forgiving of human error: “Someone always asks if the compiler really produced exactly the program I had written by hand and I have to say that it was ‘almost’ the same program. I followed the syntax equations and tried to write just what the compiler was going to produce. Unfortunately I forgot one of the redundant instructions, so the results were not quite the same. Of course, when the first machine-produced compiler compiled itself the second time, it reproduced itself exactly.”

Being lazy, however, I chose to take a switchback on the ascent, bootstrapping via Python. Much as the Jungfrau-Joch or the Klein Matterhorn can now be approached via funicular and gondola instead of on foot, we can take advantage of string and named tuple library facilities to approach the same viewpoint with little danger of arriving out of breath. The pipeline I first set up was structured as follows:

0. Lexical analysis (unfolding the character-by-character input string into a sequence of tokens and literal strings).

1. Syntax analysis (unfolding the linear lexical list into a syntax tree).

2. Code generation (in a traditional syntax-directed style).

Depending on your programming subculture, you may prefer to call this *syntax-directed translation*, a *visitor pattern*, or even an *algebraic homomorphism*. No matter what it is called, the essence of the matter is the mapping of a composition can be expressed as the composition of mappings, and we use this distributive property to divide and conquer (advice which was probably passed on to Alexander by Aristotle—showing that in certain things the ancients anticipated Hoare and Blelloch by at least a few millennia), pushing the problem of translation out to the leaves of our syntax tree and concatenating the results, thereby folding the tree back down to a sequence of output characters.

Each stage is motivated by a structural transformation: the first two steps take structure that was implicit in the input and make it explicit, while the final step uses this explicit structure to guide the translation but then



forgets it, leaving the structure implicit in the generated code string. Had we included a link phase (in which we would be concerned with flattening out the generated code into a word-by-word sequence), the building up and breaking down of structure would be almost perfectly symmetrical.

Note that you can easily cut corners on the lexical analysis. Schorre notes, “In ALGOL, strings are surrounded by opening and closing quotation marks, making it possible to have quotes within a string. The single quotation mark on the keypunch is unique, imposing the restriction that a string in quotes can contain no other quotation marks.” Therefore, a single bit’s worth of parity suffices to determine if any given nonquote character is inside or outside of a string.

Schorre was even more frugal when it came to numeric literals: “The definition of number has been radically changed. The reason for this is to cut down on the space required by the machine subroutine which recognizes numbers.” Compare Schorre’s decisions with those taken in Chuck Moore’s “Programming a Problem-Oriented-Language”<sup>8</sup> for an example of how much thought our forebears were prepared to put into their literal formats when they had to be implemented on these, by current standards, minuscule machines. (Such frugality reminds one of the Romans, who supposedly, during the negotiations to end the first Punic war, multiplexed a single set of silverware among everyone scheduled to host the Carthaginian delegation.)

The syntax analysis can also profitably cut corners. In trying to arrive at a system that can process grammatical input, you do not actually need the full machinery to analyze the grammar from which you start. In fact, if you are willing to ignore a little junk, the grammar in Figure 5 can be parsed as an expression entirely via precedence climbing, with “.”, “=”, and “/” being the binary operators and “\$” and “.OUT” being unary.

All of these cases are good examples of a general principle when bootstrapping: because you are initially not creating the cathedral, but merely putting up ephemeral scaffolding, you can save a good deal of effort by doing the un-

avoidable work (while still at the lower level, where everything is relatively difficult) in a quick and dirty manner, allowing you to do the desired work later in the proper manner (presumably much more easily, once you have a system operating at the higher level). Schorre’s paper takes two more steps in this manner, moving from META II to VALGOL I to VALGOL II all in the span of a few pages.

Another reason I took this route, rather than Schorre’s direct ascent, is because I had the luxury (much like discovering a fixed line left in place by a previous expedition) of having the skeleton of a precedence-climbing parser left over from a previous project; hence, parsing Schorre’s expressions was simply a matter of changing the operator tables. In this case, my luck was due to having been inspired by Martin Richard’s simple parsers<sup>9</sup>; Richards was a pioneer in the technique of porting and distribution via virtual machine, and his expression parsers are often under a dozen lines each; mine was left over from a reimplement in `sed(1)`, and so (having eschewed integer arithmetic) is comparatively bloated: a score of lines.

At this point, I have climbed a bit and can look down with some satisfaction at the valley below, but the switchback means I have moved a good deal sideways from the original line of ascent. I am parsing Schorre’s original file and generating code, but the code is for his VM (virtual machine), which I have not yet rewritten. Again, rather than aiming directly for the summit, I took another switchback. In this case, it was to rewrite Schorre’s grammar to generate Python code rather than META II. This is another invaluable property of good intermediate positions: I have not yet properly reconstituted Schorre’s system, but there is enough of the machinery in place to use it as intended, as a seed that can be unfolded in different ways to solve different sorts of compilation problems.

Sure enough, Schorre’s system was flexible enough to generate code in a language that would not even have been started until a quarter century later. Because of additional `.LABELS` for the import boilerplate, and an expansion of `EX2` to `EX2` and `EX25` so I

could trivially express META II’s sequential composition in Python as short-circuit conjunction (and) with identity (True), the Python-generating META II grammar grew to 33 lines instead of 30. Now I needed to implement the functionality of the META II VM in Python. The advantage was that by generating Python code, I could implement each piece using a full high-level language, essentially a form of “big step” semantics. This consisted of approximately 85 lines of code, developed largely by the mindless method of iteratively rerunning the program and implementing each operation as execution reached the point where it became necessary. Debugging the null program is not to everyone’s taste, but as A.N. Whitehead remarked: “Civilization advances by extending the number of important operations which we can perform without thinking about them. Operations of thought are like cavalry charges in a battle—they are strictly limited in number, they require fresh horses, and must only be made at decisive moments.”<sup>14</sup>

At this point, I was able to use the Python-generating META II to regenerate itself. This was still a good deal laterally removed from the direct route to the summit, but it gave me confidence that I was heading in the correct direction, and perhaps more importantly, I have far more frequent occasion to use generated Python code than code generated for Schorre’s META II VM.

Most importantly, I now had a good idea which data structures were necessary and how they fit together. (The vocabulary of programming changes as frequently as hemlines rise and fall, but the importance of structured data remains constant; Frederick P. Brooks said, in the language of his times, “Show me your flowcharts and conceal your tables, and I shall continue to be mystified. Show me your tables, and I won’t usually need your flowcharts; they’ll be obvious.”<sup>3</sup>, and before him, John von Neumann not only delineated control flow, but also meticulously tracked representation changes, in his 1947 flow diagrams<sup>13</sup>.) With this structure, it was obvious how to take Schorre’s list of opcodes for his VM and create a Python version. Having gained some experience,

this version was not only cleaner, but also shorter. Each of Schorre's opcodes turned out to be simply implementable in one to three lines of Python, so it was a relatively painless process. I had effectively implemented small-step semantics instead of a big step. To the extent that one could have arrived here directly, by following Schorre's description immediately from the paper, the switchbacks have been a waste of time. I found the diversion useful, however, because instead of needing to work out small-step semantics from scratch, or to read and understand what Schorre had written, the direction to take at each step (as if I were following a well-blazed trail) was almost forced by the data given.

By this time, I appear to have reached a peak. In the distance, I can see the other peaks that Schorre discussed, VALGOL I, VALGOL II, as well as an entire chain of different peaks that might be more attractive to modern sensibilities. But how can I be sure (especially if the clouds have come in, and in the absence of a summit book) that I am standing where Schorre did half a century ago? This is the first time I might actually need to use some intellect, and luckily for me it is known that self-reproducing systems are fixed points, and bootstrap processes should therefore converge. Little need for intellect then: you merely need to confirm that running Schorre's program in Figure 1 through a program for the machine given in figures 2–4 reproduces<sup>12</sup> itself. In fact, if you follow a similar set of switchbacks to mine, you will find that all of the possibilities converge: not only does META II via META II reproduce itself, but Python via Python (as noted supra) reproduces itself, and the two cross terms check as well: META II via Python produces the same output as META II via META II, and Python via META II is identical to Python via Python.

Note well the importance of self-reproduction here. It is not difficult to find self-referential systems: We may take the 1839 Jacquard-woven portrait depicting inventor Joseph Marie Jacquard seated at his workbench with a bunch of punched cards, or the fictional Baron Münchhausen pulling himself up by his pigtail (rather

than by his bootstraps; having needed to lift his horse as well as himself, bootstraps were never an option—he sought a greatest rather than a least fixed point) as entertaining examples, but META II is a useful example of self-reference: it derives almost all of its power, both in ease of propagation and in ease of extension, from being self-applicable: from being the square-root of a compiler.

What has this exercise accomplished? It has resulted in a self-reproducing system, executing both on the original META II VM (working from the original listing) and on Python or another modern language. Obviously, I could use the same process I followed to bootstrap from the Python to the META II machine not only to port to yet another underlying technology, but also to become self-hosting. Less obviously, the basic problem I have solved is to translate (in a “nice” manner) one Kleene Algebra (consisting of sequences, alternations, and repetitions) to another, which is a pattern that, if not ubiquitous in computing, is certainly common anytime we deal with something that has more structure than a linear “shopping list” of data. Compare Thompson's NFA (nondeterministic finite automaton) construction<sup>11</sup> in which a search problem is solved by parsing a specification that is then executed on a virtual (nondeterministic) machine, with the twist that the nondeterministic virtual code has been further compiled into actual deterministic machine code.

Finally, remember that META II lends itself well to this kind of exercise precisely because it was designed to be bootstrapped. As Schorre says in his introduction: “META II is not intended as a standard language which everyone will use to write compilers. Rather, it is an example of a simple working language which can give one a good start in designing a compiler-writing compiler suited to his own needs. Indeed, the META II compiler is written in its own language, thus lending itself to modification.”

I hope the exercise of implementing your own META II will have not only the short-term benefit of providing an easily modifiable “workbench” with which to solve your own prob-

lems better, but also a longer-term benefit, in that to the extent you can arrange for functionality to be easily bootstrappable, you can help mitigate the “perpetual palimpsest” of information technology, in which the paradox of bitrot means many artifacts effectively have a shorter half-life than even oral history.

After all, barbarians may be perfectly adapted to their environment, but to be part of a civilization is to be aware of how other people, in other places and times, have done things, and hence to know how much of what one does oneself is essential and how much accidental. More specifically, barbarians must learn from their own mistakes; civilized people have the luxury of learning from other people's mistakes. Very specifically, for engineers faced with ephemeral requirements, it is often helpful to avoid thinking of the code base at hand as a thing in itself, and instead consider it only a particular instantiation of the classes of related possible programs. **□**

#### References

1. Backhouse, R. Regular algebra applied to language problems. *Journal of Logic and Algebraic Programming* 66 (2006); <http://www.cs.nott.ac.uk/~rcb/MPC/RegAlgLangProblems.ps.gz>.
2. Brooker, R.A., MacCallam, I.R., Morris, D. and Rohl, J.S. The compiler compiler. *Annual Review in Automatic Programming* 3 (1963), 229–275.
3. Brooks, F.P. *The Mythical Man-Month*. Addison Wesley, 1975.
4. Burstall, R.M. Proving properties of programs by structural induction. *Computer Journal* 12, 1 (1969), 41–48.
5. Hehner, E.C.R. A practical theory of programming. *Texts and Monographs in Computer Science*. Springer, 1993.
6. Johnson, S.C. Yacc: Yet another compiler-compiler; <https://www.cs.utexas.edu/users/novak/yaccpaper.htm>.
7. Landin, P.J. The next 700 programming languages. *Commun. ACM* 9, 3 (Mar. 1966), 157–166; <http://doi.acm.org/10.1145/365230.365257>.
8. Moore, C.H. Programming a problem-oriented-language, 1970; <http://www.colorforth.com/POL.htm>.
9. Richards, M. *The MCPL Programming Manual and User Guide*. (2007) 58–63; <http://www.cl.cam.ac.uk/~mr10/mcplman.pdf>.
10. Schorre, D.V. META II: A syntax-oriented compiler writing language. In *Proceedings of the 19th ACM National Conference* (1964), 41.3011–41.3011; <http://doi.acm.org/10.1145/800257.808896>.
11. Thompson, K. Programming techniques: Regular expression search algorithm. *Commun. ACM* 11, 6 (June 1968), 419–422; <http://doi.acm.org/10.1145/363347.363387>.
12. Thompson, K. Reflections on trusting trust. *Commun. ACM* 27, 8 (Aug. 1984), 761–763; <http://doi.acm.org/10.1145/358198.358210>.
13. von Neumann, J., Goldstine, H.H. *Planning and Coding of Problems for an Electronic Computing Instrument*. Institute for Advanced Study, Princeton, NJ, 1947.
14. Whitehead, A.N. *An Introduction to Mathematics*. Henry Holt and Company, New York, NY, 1911.

**Dave Long** divides his time between developing equine area network sensor systems as an application of current technology to the problem of training horses for an old mounted game, and simply enjoying playing it.

Copyright held by author.  
Publication rights licensed to ACM. \$15.00.

# 6th Annual ACM SIGPLAN Conference on

Systems,  
Programming,  
Languages, and  
Applications:  
Software for  
Humanity

## Location

Sheraton Pittsburgh at  
Station Square Hotel

## Events

- 30th Annual OOPSLA
- Onward!
- Wavefront
- Generative Programming and Component Engineering (GPCE)
- Dynamic Languages Symposium (DLS)
- Software Language Engineering (SLE)
- Pattern Languages of Programming (PLoP)
- and more

## General Chair

Jonathan Aldrich  
*Carnegie Mellon University*

## OOPSLA Papers Chair

Patrick Eugster  
*Purdue University*

## Onward! Papers Chair

Gail Murphy  
*University of British Columbia*

## Onward! Essays Chair

Guy Steele  
*Oracle Labs*

<http://splashcon.org>  
[info@splashcon.org](mailto:info@splashcon.org)



DOI:10.1145/2660765

**What was once centralized or federated technology governance is increasingly participatory.**

BY STEPHEN J. ANDRIOLE

# Who Owns IT?

IN THE 20<sup>TH</sup> CENTURY, technology governance was largely about standards and centralized management. Moving into the 21<sup>ST</sup> century, things began to change, first from centralized to federated technology governance models, then to “participatory” models. Commoditization, consumerization, and alternative technology-delivery models changed the way governance is defined and managed in many, though not all, companies. For many of them, the number of technology stakeholders has increased as the importance of technology has expanded to include at least three categories of governance: operational, strategic, and emerging technology. For many companies, the governance mission is evolving toward

a shared, participatory model that recognizes the roles of all internal and external stakeholders, especially as companies acquire, deploy, and support technology through the “cloud” and supply chains globalize and integrate.

Our survey and interview data suggests governance now involves more stakeholders than ever before, many living way beyond the corporate firewall. The data reported here suggests participatory governance is emerging as a major technology governance model for the 21<sup>ST</sup> century, and, for companies that increasingly satisfy business requirements through adoption of cloud computing, the participatory governance model is accelerating. Conversely, the companies that avoid cloud deployment and other alternative deployment models will likely stay within more-traditional centralized/federated governance structures. Our survey and interview data describes how technology governance is changing. As new technologies and technology-delivery models emerge, technology governance is evolving in ways quite different from the dominant models of the 20<sup>TH</sup> and early 21<sup>ST</sup> centuries. Based on the data, this article describes a new participatory governance matrix that recognizes the role internal *and* external stakeholders play in the technology-governance process.

## Technology Governance

Peterson<sup>14</sup> defined information technology governance this way: “IT governance describes the distribution of IT decision-making rights and re-

### » key insights

- **Technology governance was often tightly controlled but is now loosening.**
- **There are many internal *and* external technology “governors” today, something no one would have predicted five years ago.**
- **Governance is now about productivity and partnerships, not just standardization and control.**



sponsibilities among different stakeholders in the enterprise, and defines the procedures and mechanisms for making and monitoring strategic IT decisions.” Technology governance, as in all aspects of corporate governance, concerns decision rights often organized in responsible/accountable/consultative/informed, or RACI, playbooks that describe who is allowed to acquire, deploy, and support business technology.<sup>12,20</sup>

In centralized IT organizations, decision rights involved in the acquisition, deployment, and support of technology belong to a central group reporting to a corporate executive, increasingly the CFO. In decentralized organizations, decision rights are shared across the enterprise and business units; in federated organizations, rights are coordinated across the corporate IT group, the business units, and even specific corporate functions.<sup>1,5-8,11,18,21,23,25</sup> The evolution of research about technology governance is instructive here. Years ago, researchers, including Brown and Magill,<sup>7</sup> Rockart et al.,<sup>18</sup> and Weill and Broadbent,<sup>25</sup> discussed technology governance in the context of organizational realities and the reality of choice, where hardware, software, and communications options were limited. But as organizations changed, especially with the federation of business units, and technology options increased, research on governance offered alternative insights into how companies redefined governance, as well as the role of technology in all business processes and models.<sup>11,20,21,23</sup>

Research on technology alignment and governance is extensive.<sup>3,6,13,24,28</sup> We also know a lot about structures and processes.<sup>26</sup> We know differences across governance structures are often explained through the formalization of arrangements. Historically, technology governance has been more explicit and formalized around operational technology (such as laptops, desktops, networks, storage, and security) than strategic technology (such as business applications and special-purpose hardware) or especially around emerging technology (such as social media, location-based services, wearables, and the



## The new cloud-based, technology-delivery models and proliferation of “consumerized” devices have completely changed the governance equation.



Internet of things).<sup>2,14,17,19</sup> Figure 1 outlines the differences among the three categories of technology, partly based on Weill and Broadbent,<sup>25</sup> Weill and Ross,<sup>26</sup> and Andriole.<sup>2,3</sup>

In the 1970s and 1990s, infrastructure, in-house-developed applications, and databases were often centralized under the command of an enterprise chief information officer (CIO). Part of this command structure can be explained by the relative scarcity of hardware and software diversity at the time, unlike today, when there are many more hardware, software, communications, and delivery options than the popular “command and control” approach to managing corporate assets.

Over time, centralization yielded to decentralization and then federation. Enterprise CIOs countered with “technology standardization,” believing even if the lines of business had some control, so long as they controlled the technology standards around primary devices—servers, desktops, and communications—they were still essentially in charge, even if they did not select every one of the organization’s business applications. The centralization/decentralization/federation game persisted until the Web arrived in the early 1990s, when control was influenced by technology “consumers” who no longer viewed themselves only as end users.

During the mid-to-late 1990s, governance changed due largely to the “irrational exuberance” of the dot-com era and temporary determination that technology was more strategic than tactical. Following the dot-com crash of 2000, governance returned to operational cost control, staying that way until 2003 when technology budgets began to increase again. In the mid-2000s, governance changed again when it was shared by enterprise CIOs and business-unit CIOs (assuming the structure recognized business-unit CIOs) or just “business-unit technology directors,” as they are sometimes called. Companies continued on this path until the financial world melted down again in 2008, and governance changed again, when it was centralized in the hands of a few—or even just one—senior executive(s), the CFO, the COO, or, infrequently after 2008, the CEO.

As more and more business processes and models were converted or

augmented through digital technology, technology also became more accessible through new delivery models, especially cloud-delivery models. This finalized the near-total dependence business has on the reliability, scalability, and security of its digital technology, permanently changing the way companies acquire, deploy, and support technology. That is, businesses of all kinds discovered they could not function—or even exist—without IT and, by extension, the new technology-delivery models.

Old notions of governance are being challenged by technology commoditization, consumerization, and alternative technology-delivery models, along with other emerging technologies about to hit their problem-solving stride. This challenge is not just about the nuances of centralized/decentralized/federated but some very different governance structures that recognize the importance of outside participants.

Business units aggressively pilot and adopt new technologies. Consumerized, cloud-delivered technology has changed the rules around technology acquisition, deployment, and support. Business units no longer ask corporate IT if they can rent software or buy iPads; they just rent and buy as they choose, often without telling IT what they have done. So-called “shadow IT” is more pervasive than ever. The ability to do what they please is fueled by the technology itself. Cloud computing, renting rather than buying technology, and easily supported devices (such as smartphones and tablets) make it easy for anyone to acquire, deploy, and support digital technology. The new cloud-based-technology-delivery models and proliferation of consumerized devices have completely changed the governance equation.

Each governance configuration comes with implications and consequences. The allocation of decision and input rights is simultaneously political and practical. Companies must decide how they want to allocate rights and how far they want to push their political processes.

**Challenges**

The very notion that operational technology is fully commoditized

challenges governance in several important ways. For example, many companies outsource their operational technology to local and/or offshore providers. Sharing outsourcing governance of even operational technology can make sense, especially as companies globalize. Strategic technology (technology facing customers and suppliers) is often “co-governed” by technology and business professionals, as the performance metrics are both technological and functional. Supply-chain partners represent an ongoing challenge to governance, as they often present their own integration and interoperability challenges that must be satisfied by the business units with which they do business.<sup>10</sup>

Renting (versus buying and installing) software calls for whole new governance models. Vendor management has emerged as a core competency for many companies. Service-level agreements must be managed for performance; business units and central IT alike have roles to play here. Similarly, renting hardware through cloud delivery will emerge within the decade as a viable alternative to building and maintaining huge server farms. This trend will challenge governance as well, requiring cooperation between business and technology units, since “control” will now involve third parties—the cloud and supply-chain providers—committed to providing support to the whole company, not just its central IT organization.

Consumerization has changed the way technology is introduced. Technology adoption now often occurs

before employees enter the building. Web 2.0 and social-media technologies (such as wikis, blogs, podcasts, RSS filters, virtual worlds, crowdsourcing, mashups, and social networks) are quickly making their way into companies. Corporate IT departments struggle to keep up with the use of these tools by employees, customers, and suppliers. Mashups are the creation of computing components inside and outside the corporate firewall. Who controls the APIs, the components and widgets that mash into new applications? How do companies prevent blogs and wikis from springing up on employees’ laptops?

Web-based applications also pose a challenge to old governance models. They are built quickly and deployed almost instantly. Changes to existing transaction-oriented Web sites are immediate. If a business unit wants to roll out a revised global pricing schedule, does it need to go through corporate IT? We crossed that authority chasm a decade ago when we invested in user-controlled rules engines and other technologies intended to support real-time decision making. New applications are designed and developed by internal professionals and, increasingly, by outside developers accountable to business units, not to corporate IT. Application development and all varieties of Web-based applications are no longer governed by corporate IT, except, as suggested earlier, at the architectural level (which should remain in the control of the enterprise technology organization). Participatory develop-

**Figure 1. Technology categories.**

<b>Operational Technology</b>	<b>Strategic Technology</b>	<b>Emerging Technology</b>
Shared applications (such as accounting, budgeting, database management, and enterprise resource planning)	Applications that connect to customers, suppliers, and partners	Technologies to improve and disrupt operational effectiveness
Shared services (such as networks, security, risk management, and email)	Business unit applications that differentiate the business unit in the marketplace	Technologies to improve—and disrupt—business models and processes
Shared databases (such as customer, manufacturing, and supplier databases)	Applications and databases that are business unit and vertical-industry specific	Big data analytics, social media, and wearables
Shared “standard” devices (such as laptops, printers, and phones)	Sourced or customized applications and databases with short expected life spans	Operational and strategic digital trends and optimal pilots

**Figure 2. Governance participants.**

Internal Stakeholders			External Stakeholders		
The Enterprise	Corporate Functions	Business Units	Hardware, Software, and Service Providers	Partners and Supplies	The Crowd
The corporate entity that defines the corporate mission and overarching reporting structure of the organization, including technology leaders	Specific activities that define corporate organizations (such as marketing, finance, accounting, human resources, and information technology)	Specific lines of business that focus on specific customer sets with products and services that generate sales and profits and that require information technology	Vendors that provide hardware, software, networks, and other services to the enterprise and business units, increasingly delivered through cloud service providers	Business partners and suppliers that enable business functions, as well as product and service definition, manufacturing, and delivery, along with other activities	All those outside the enterprise, business units, providers, partners, and suppliers that might contribute in any way to the success of the company

**Figure 3. Findings from business (B) and technology (T) professionals.**

Questions for Business (B) and Technology (T) Professionals	B	T
	% Yes	% Yes
Do you have a governance policy?	71%	91%
Is your governance policy effective?	43%	61%
Are the lines of business active governance partners?	44%	15%
Is a central IT group in control of operational IT assets?	67%	82%
Is "Shadow IT" larger because of your governance policies?	89%	42%
Should central IT organizations own operational technology?	76%	86%
Should the lines of business own strategic technology?	89%	59%
Should the lines of business own emerging technology?	69%	44%
Is governance still about standardization and control?	54%	91%
Should the lines of business have more governance power?	88%	32%
Does the cloud represent a governance "game changer"?	92%	69%
Should vendors be part of the governance process?	87%	49%
Should consultants be part of the governance process?	89%	51%
Should suppliers be part of the governance process?	71%	47%
Should the crowd be part of the governance process?	86%	39%
Should the lines of business define governance processes?	87%	37%
Should corporate IT define governance processes?	41%	76%
Should technology governance be more "democratized"?	81%	58%
Is technology "governance" an obsolete concept?	43%	52%

ment is a change from the past, but the prominence of the Web as the emerging dominant transaction platform has changed everything.

Globalization is another major driver of new governance models. As more and more companies expand their global reach, they must adjust the authority they exercise over the business units they encourage to grow. Decentralization and federation are necessary to

enable agile decision making; business units expanding around the globe need the authority to make local and regional decisions. Extending corporate IT from headquarters around the world makes sense infrequently. Servicing an army of technology ex-pats is expensive and inhibiting. Local talent, providers, and local/regional/country support makes sense as companies build sustainable footprints around the world.

Globalization calls for new governance structures. "Headquarters" must decentralize. Standards must become architectural and procedural, not based on brands, models, or vendors. Our data suggests enterprise CIOs and CTOs should focus on infrastructure optimization, alternative technology-delivery models, and architecture—and not much else. Business units should focus on requirements, application development (within architectural standards), and deployment of fast/cheap technologies like those in social media. If companies do not adjust their governance around these activities, the business-technology partnership will collapse. There will be major pushback from the business units that want to move quickly, cheaply, adaptively. If central IT organizations provide roadblocks to these operating principles, the lines of business will end run the IT organization.

### Participatory Technology Governance

Consider this essential finding of our analysis: Where technology governance was essentially something defined and implemented by technology and business professionals in their own companies, the new participatory governance reflects the distribution of decision rights across multiple internal and external participants.

The concept of "participatory governance" emerged from informal discussions validated through formal interviews and surveys with business-technology managers and business executives across the globe (for the surveys) and the U.S. locally/regionally/nationally (for the interviews) on the state of technology governance. The data was collected from both the technology and business sides of multiple companies. Segmenting the groups indicates technology professionals are somewhat less likely to endorse participatory governance, while business professionals are much more likely to endorse it. There is, however, general agreement that cloud and supply-chain computing are the major drivers of participatory governance and that technology vendors and business suppliers should be part of the governance process.



The discussion here contrasts the way corporate leaders have governed technology in the past and where technology governance is likely to go. Participatory governance is a response to the relatively closed governance structures and processes prevalent in the 20<sup>th</sup> and early 21<sup>st</sup> centuries. It is also described by interviewees as a response to the general diffusion of digital technology within and beyond corporate firewalls. The whole notion of governance has expanded. Our data confirms emerging trends in the acquisition and control of technology assets, as well as in the administration of technology processes and services.

Companies routinely look outward to make technology decisions; that is, they find they must consult stakeholders/participants outside their companies to make important technology acquisition, deployment, and support decisions. We consolidated the results from multiple surveys, supplemented by interviews used to (anecdotally) validate/invalidate what the survey data told us.

Figure 2 outlines the governance participants identified in our data. The survey and interview questions focused on governance participants, models, and processes. The result of the surveys and interviews filled a new RACI-based governance matrix presented here. The number of participants in the emerging governance process has increased, and nearly all growth is outside the proverbial corporate firewall. Some new participants are the result of changes in the way technology is acquired, deployed, and supported (such as the vast number of cloud computing providers under contract today). Similarly, integrated supply chains have increased dependency among technology providers. Finally, since many customers are glued to their social networks, companies today must engage them through communication and content networks they do not control in any way, shape, or form. How do they “govern” this activity?

### Survey and Interview Data

The data was collected from surveys conducted by the author at Villanova University through the Cutter Consortium (<http://www.cutter.com>) 2008–



**Segmenting the groups indicates technology professionals are somewhat less likely to endorse participatory governance, while business professionals are much more likely to endorse it.**



2014 and from interviews the author conducted 2004–2014 with CIOs and CTOs who were part of the Villanova University CIO Technology Advisory Council, a rotating group of more than 50 local, regional, and national technology executives. The surveys and interviews involved more than 500 technology managers and executives. Data was collected through surveys and face-to-face interviews repeated every two years. Survey instruments and questionnaires were developed for both data-collection processes. General technology-adoption questions, as well as specific questions, were asked about technology governance and alternative organizational structures. The data reflects input from both the technology and business sides of companies whose participants were asked to identify their roles in technology management. The data reported here represents the combined percentages for the period 2008–2014.

Figure 3 outlines very different governance perceptions and attitudes from those expressed in surveys conducted in the late 20<sup>th</sup> and early 21<sup>st</sup> centuries. Note first the distinctions among operational technology, strategic technology, and emerging technology. Note, too, the range of referenced stakeholders. There are the usual suspects—the corporate and business-unit clients—but there is also a new cast of characters, including vendors, providers, partners, and even “the crowd,” or the random actors who gather on social-media sites. The full cast is why the range of governance and number of governance stakeholders is dramatically different today, and why the whole notion of control will yield to what might be called “participatory” or “shared” control.

Also note the differences between technology and business professionals. While business professionals hold strong views about governance trends, the technology professionals also acknowledge the changes in technology management and acquisition, even though their opinions are measurably different from those of business professionals.

Our interviews with CIOs, CTOs, and senior executives and managers from the business side say participatory governance is inevitable, endors-

ing the “recommendation” to adopt a more collaborative, participatory approach to technology governance:

Here are some insightful quotes from some of our technology executives:

“It was only a matter of time before the businesses demanded more control of technology. I mean, we sort of kept them at arm’s length for years. Once Apple started making stuff that everyone really really wanted, we were toast. So we had to give up some control.” —*CIO of a chemicals company*

“The world really is flat. We sell products all over the world and have databases and applications everywhere, including the cloud. It is impossible to control everything from one address. We had to rethink governance, or there would be a revolution.” —*CIO of a technology company*

“We will be 90% cloud-based in five years. Our vendors have as much to say about how we govern technology as we do. Pretty soon, cloud vendors will be telling us what we can and cannot do.” —*CIO of a pharmaceuticals company*

“Gone are the days when IT calls the shots. And maybe that’s not a bad thing. For a long time, we owned all the technology and the processes for buying and implementing technology. But now we have to move faster and open up our standards to businesses that need more technology faster and cheaper. I guess it’s about time.” —*CIO of a financial services company*

“Working with the businesses is great. But they don’t always understand how complicated IT is or how much work it takes to get technology to work. We have to work with our vendors and consultants constantly to get

all this right. The businesses worry much more about what technology can do for them now, especially for sales. That’s great, but it takes more than hand waving.” —*CIO of a financial services company*

“The lines of business get it. They understand that they need our infrastructure but want more control over the applications they use. Makes sense to me, so long as their decisions keep the infrastructure viable. We can’t have a free-for-all. There have to be some rules, but I get that the rules need to be more flexible. I get that now.” —*CTO of an insurance company*

Here are some quotes from our interviews with business executives:

“We need to move fast. We can’t wait for IT to decide what we should—or should not—be doing. Our problems need technology solutions. While standards are important, and all they bring to stability and security, we still need to solve problems quickly.” —*President of a pharmaceutical business unit*

“IT is the group that tells me what I can’t do, not what I want to do with technology. That has to change or we will fall behind. When I ask for new technology, my question assumes that IT can make it work. Or I will find it somewhere else.” —*CEO of a biotech company*

“Cloud computing has given us all hope. Not just because it represents a good alternative but because it frees us from corporate IT. It used to be that for us to get some new database or application we had to ask IT, which then told us that it would be too hard to do. Since we depended on IT’s infrastructure to get things done, we had to accept their ‘interpretation’ of how easy or hard it

would be to give us what we wanted. Now we can go to the cloud and just rent the damn stuff.” —*Sales manager of a financial services company*

“It’s about time that central IT asks us what we want. And now when we tell them they listen. I always wonder if they listen because they want to be more responsive or because they know we can just go buy it ourselves. Thank God for the cloud. It gives me the ultimate trump card.” —*CIO of an insurance company business unit*

The survey and interview data suggests business and technology professionals understand the governance process is changing and the number of participants in the governance process is increasing. Vendors, service providers, partners, and colleagues in the cloud are now governance stakeholders. Vendors and service providers are special stakeholders since the products and services they offer define de facto governance. Companies that outsource huge amounts of their operational infrastructures outsource many of their technology standards and the governance around those standards. While the standards themselves can be broad, they still define what the hardware, software, and service offerings will be.

Environments that outsource lots of technology and technology services share governance with their providers. Similarly, suppliers and other partners frequently require specific technology-based transaction processing that also results in shared governance.<sup>9,22</sup> The crowd is one of the most dramatic challenges to corporate governance. The crowd is the

Figure 4. RACI participatory governance.

		Technologies			
		Participants	Operational Technology	Strategic Technology	Emerging Technology
Internal	The Enterprise		R/A/C/I	R/A/C/I	R/A/C/I
	Corporate Functions		R/A/C/I	R/A/C/I	R/A/C/I
	Business Units		R/A/C/I	R/A/C/I	R/A/C/I
External	Hardware, Software, and Services Providers		R/A/C/I	R/A/C/I	R/A/C/I
	Partners and Suppliers		R/A/C/I	R/A/C/I	R/A/C/I
	The Crowd		R/A/C/I	R/A/C/I	R/A/C/I

source of a variety of “extensions” to everyone’s technology capabilities. The best example of this is the application programming interfaces (APIs) published by companies and individuals that make it possible for clients and their providers to extend the functionality of applications quickly and cheaply. But are all APIs OK to use? Governance must extend well beyond the corporate firewall to include policies and protocols for the use of externally developed—yet powerful—APIs and other software widgets that can be used to enhance functionality. In addition to APIs and widgets, the crowd can also provide expertise. We are moving quickly toward a free-agent approach to selected corporate problem solving. What if a company must develop a dashboard, a process, a chemical, or a drug? Should it turn to the crowd? What if it moved its help desk to the cloud and paid specialists when they solved problems? Shared governance is at least partially assumed by these trends.

Finally, note in Figure 4 the responsible/accountable/consultative/informed, or RACI, playbook informed by the survey and interview data. The data suggests the participation scale—from responsible to informed—has shifted. Of special importance is the addition of external stakeholders to the governance process.

The RACI playbook suggests the enterprise is responsible (R) and accountable (A) for operational technology but less so for strategic and emerging technology. It also suggests providers are also accountable (A) for operational delivery because so much technology is now outsourced from cloud providers. Partners and suppliers also play an important role in operational technology selection and deployment (As).

Corporate functions and business units are accountable (A) and responsible (R) for strategic and emerging technology. This is a major change from the governance of the 20<sup>th</sup> century, when most if not all strategic and emerging technology was governed by the enterprise CIO.

Providers, partners, and the crowd are now direct participants in technology acquisition, deployment, and support through their consulta-

tive (C) and informed (I) roles, with the exception of providers’ shared accountability (A) for strategic and emerging technology, due primarily to the implications of the integration and support of new technology. This structure is new.

## Conclusion

These findings and analysis indicate governance is changing, “control” is a concept morphing into collaboration and participation, and participatory governance will replace both the rigid conventional governance structures and processes of the 20<sup>th</sup> century and even more-open “federated” structures of the early 21<sup>st</sup> century. Participatory governance acknowledges expansion of the number of governance stakeholders, commoditization of technology, consumerization, and the increasing practice of outsourcing operational, strategic, and emerging technology. The data also suggests the new business technology alignment opportunity is through participatory governance. **C**

## References

1. Agarwal, R. and Sambamurthy, V. Principles and models for organizing the information technology function. *Management Information Systems Quarterly Executive* 1 (Mar. 2002), 1–16.
2. Andriole, S.J. Ready technology. *Commun. ACM* 57, 2 (Feb. 2014), 40–42.
3. Andriole, S.J. Boards of directors and technology governance: The surprising state of the practice. *Communications of the Association for Information Systems* 24 (Mar. 2009), 373–394.
4. Brousell, L. IT Reorgs on the horizon. *CIO Magazine* (Feb. 26, 2013).
5. Brown, C.V. Examining the emergence of hybrid IS governance solutions: Evidence from a single case site. *Information Systems Research* 8, 1 (1997), 69–94.
6. Brown, A.E. and Grant, G.G. Framing the frameworks: A review of IT governance research. *Communications of the AIS* 15, 38 (2005), 696–712.
7. Brown, C.V. and Magill, S.L. Alignment of the IS functions with the enterprise: Toward a model of antecedents. *MIS Quarterly* 18, 4 (Apr. 1994), 371–403.
8. Chan, Y.E. and Reich, B.H. IT alignment: What have we learned? *Journal of Information Technology* 22, 4 (2007), 297–315.
9. Demirkan, H., Cheng, H., and Bandyopadhyay, S. Coordination strategies in an SaaS supply chain. *Journal of Management Information Systems* 26, 4 (2010), 119–143.
10. Dong, H., XU, S.X., and Zhu, K.X. Information technology in supply chains: The value of IT-enabled resources under competition. *Information Systems Research* 20, 1 (Mar. 2009), 18–32.
11. Evaristo, J.R., Desouza, K.C., and Hollister, K. Centralization momentum: The pendulum swings back again. *Commun. ACM* 48, 2 (Feb. 2005), 66–71.
12. Feltus, C., Petit, M., and Dbois, E. Strengthening employee responsibility to enhance governance of IT: COBIT RACI chart case study. In *Proceedings of the ACM Workshop on Information Security Governance* (Chicago, Nov. 13). ACM Press, New York, 2009, 23–32.
13. McElheran, K. Decentralization versus centralization in IT governance. *Commun. ACM* 55, 11 (Nov. 2012), 28–30.
14. Peterson, R. Crafting information technology governance. *EDPACS: The EDP Audit, Control, and Security Newsletter* 32, 6 (Dec. 2004), 1–24.

15. Peterson R. Configurations and coordination for global information technology governance: Complex designs in a transnational European context. In *Proceedings of the Hawaii International Conference on System Science* (Wailea, Maui, Jan. 2–6). IEEE Computer Society Press, 2001.
16. Peterson, R. Information strategies and tactics for information technology governance. In *Strategies for Information Technology Governance*, W. Van Grembergen, Ed., Idea Group Publishing, Hershey, PA, 2003, 37–78.
17. Peterson, R., Parker, M.M., and Ribbers, P. Information technology governance processes under environmental dynamism: Investigating competing theories of decision making and knowledge sharing. In *Proceedings of the 23<sup>rd</sup> International Conference on Information Systems* (Barcelona, Spain, Dec. 15–18). Kluwer Academic Publishers, Norwell, MA, 2002.
18. Rockart, J., Earl, M., and Ross, J. Eight imperatives for the new IT organization. *Sloan Management Review* (Oct. 1996), 43–55.
19. Sambamurthy, V. and Zmud, R.W. Arrangements for information technology governance: A theory of multiple contingencies. *MIS Quarterly* 23, 2 (Feb. 1999), 261–290.
20. Teo, W.L., Manaf, A.A., and Choong, P.H.L. Practitioner factors in information technology. *Journal of Administrative Sciences & Technology* (June 2013), 1–12.
21. Tiwana, A., Konsynski, B., and Venkatraman, N. Information technology and organizational governance: The IT governance cube. *Journal of Management Information Systems (Special Issue)* 30, 3 (Winter 2013–2014), 7–12.
22. Topi, H. and Tucker, A.B., Eds. *Information Systems and Information Technology, Volume 2 (Computing Handbook Set)*. Taylor and Francis, Boca Raton, FL, 2014.
23. Ullah, A. and Lai, R., Requirements engineering and business/IT alignment: Lessons learned. *Journal of Software* 8, 1 (Jan. 2013), 1–10.
24. Wallace, M. and Webber, L. *IT Governance: Policies & Procedures*. Wolters-Kluwer, Alphen aan den Rijn, the Netherlands, 2012.
25. Weill, P. and Broadbent, M. *Leveraging the New Infrastructure: How Market Leaders Capitalize on Information Technology*. Harvard Business School Press, Boston, MA, 1998, 58–62.
26. Weill, P. and Ross, J. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business School Press, Boston, MA, 2004.
27. Weill, P., Subramani, M., and Broadbent, M. Building IT infrastructure for strategic agility. *Sloan Management Review* 44 (Oct. 2002), 57–65.
28. Weill, P. Don’t just lead, govern: How top-performing firms govern IT. *MISQ Executive* 3, 1 (Mar. 2004), 1–17.
29. Winkler, T.J. and Brown, C.V. Horizontal allocation of decision rights for on-premise applications and software-as-a-service. *Journal of Management Information Systems* 30, 3 (Jan. 2014), 13–48.

**Stephen J. Andriole** (stephen.andriole@villanova.edu) is the Thomas G. Labrecque Professor of Business Technology in the Department of Accountancy and Information Systems in the Villanova School of Business at Villanova University, Villanova, PA.

DOI:10.1145/2660766

**Preparing data for public release requires significant attention to fundamental principles of privacy.**

BY ASHWIN MACHANAVAJJHALA AND DANIEL KIFER

# Designing Statistical Privacy for Your Data

IN 2006, AOL RELEASED a file containing search queries posed by many of its users. The user names were replaced with random hashes, though the query text was not modified. It turns out some users had queried their own names, or “vanity queries,” and nearby locations like local businesses. As a result, it was not difficult for reporters to find and interview an AOL user<sup>1</sup> then learn personal details about her (such as age and medical history) from the rest of her queries.

Could AOL have protected all its users by also replacing each word in the search queries with a random hash? Probably not; Kumar et al.<sup>27</sup> showed that word co-occurrence patterns would provide clues about which hashes correspond to which words, thus allowing an attacker to partially reconstruct the original queries. Such privacy concerns are not unique to Web-search data. Businesses, government

agencies, and research groups routinely collect data about individuals and need to release some form of it for a variety of reasons (such as meeting legal requirements, satisfying business obligations, and encouraging reproducible scientific research). However, they must also protect sensitive information, including identities, facts about individuals, trade secrets, and other application-specific considerations, in the raw data. The privacy challenge is that sensitive information can be inferred in many ways from the data releases. Homer et al.<sup>20</sup> showed participants in genomic research studies may be identified from publication of aggregated research results. Greveler et al.<sup>17</sup> showed smart meter readings can be used to identify the TV shows and movies being watched in a target household. Coull et al.<sup>6</sup> showed webpages viewed by users can be deduced from metadata about network flows, even when server IP addresses are replaced with pseudonyms. And Goljan and Fridrich<sup>16</sup> showed how cameras can be identified from noise in the images they produce.

Naive aggregation and perturbation of the raw data often leave exposed channels for making inferences about sensitive information;<sup>6,20,32,35</sup> for instance, simply perturbing energy readings from a smart meter independently does not hide trends in energy use. “Privacy mechanisms,” or algorithms that transform the data to ensure privacy, must be designed carefully according to guidelines set by a privacy definition. If a privacy definition is chosen wisely by the data curator, the sensitive information will be protected.

## » key insights

- **Data snoopers are highly motivated to publicize or take advantage of private information they can deduce from public data.**
- **History shows simple data anonymization and perturbation methods frequently leak sensitive information.**
- **Focusing on privacy design principles can help mitigate this risk.**

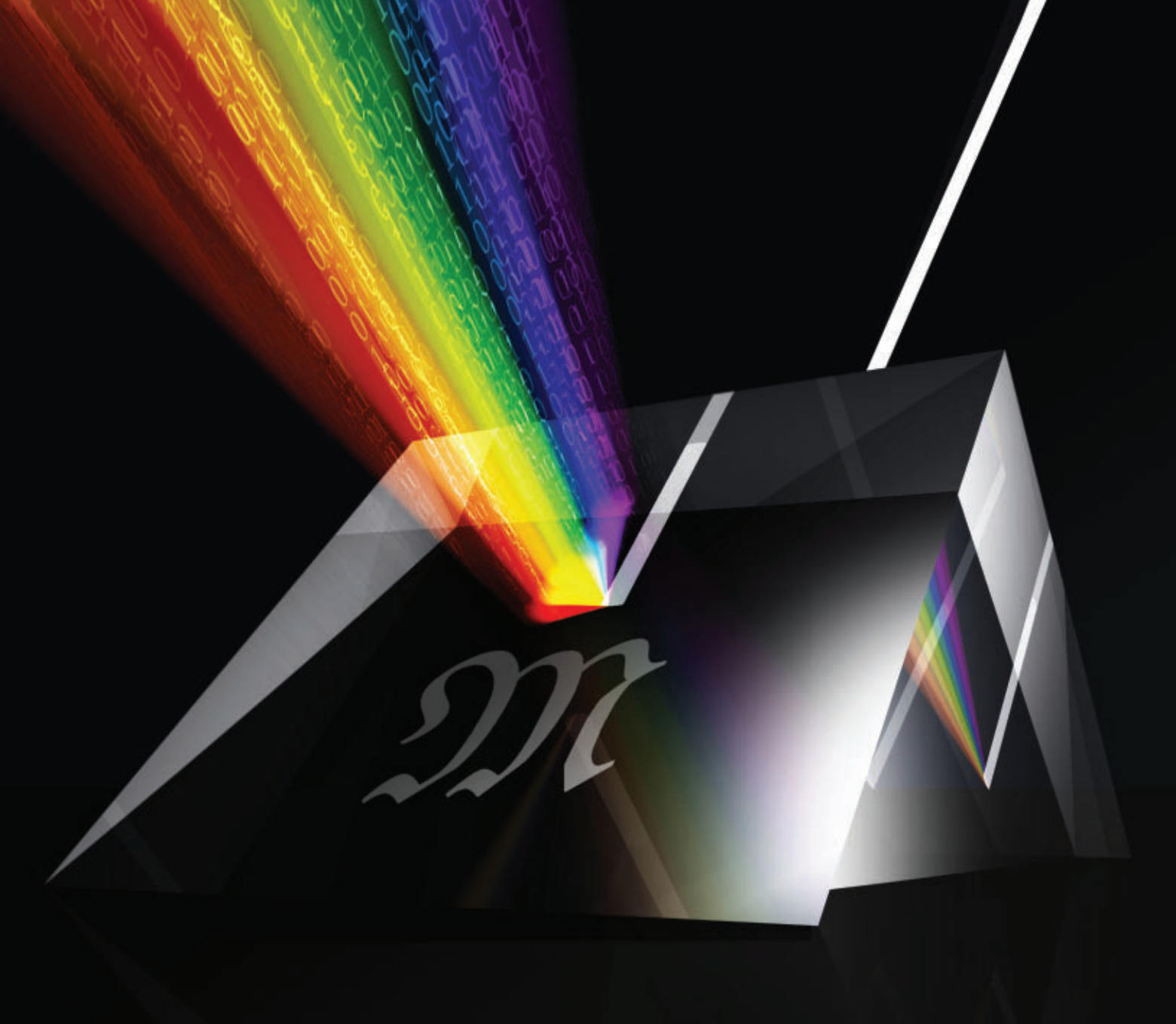


IMAGE BY ANDRÉJ BORYS ASSOCIATES/SHUTTERSTOCK

Unfortunately, privacy definitions are not one-size-fits-all. Each application could have its own unique privacy requirements. Working independently, researchers from disparate fields rediscover similar privacy technologies, along with their weaknesses, new fixes, and other vulnerabilities. Our goal here is to synthesize some of the latest findings in the science of data privacy in order to explain considerations and best practices important for the design of robust privacy definitions for new applications. We begin by describing best practices, then explain how they lead to a generic template for privacy definitions, explore various semantic privacy guarantees achievable with this template, and end with an exam-

ple of a recent privacy definition based on the template and apply it to privacy-preserving  $k$ -means clustering.

#### Desiderata of Privacy Definitions

When data is collected, the curator, with the aid of a privacy definition, puts it in a form that is safe to release. A privacy definition is a specification for the behavior of randomized and deterministic algorithms. Algorithms that satisfy the spec are called privacy mechanisms. The curator first chooses a privacy definition, then a privacy mechanism  $\mathcal{M}$  satisfying the definition. The curator will run  $\mathcal{M}$  on the sensitive data, then grant external users access to the output of  $\mathcal{M}$ , or the “sanitized output.”

There is a long history of proposed privacy definitions, new vulnerabilities discovered, and amended privacy definitions developed only to be broken once again. As privacy concerns spread, parallel copies of this process are spawned in many research areas. Fortunately, current research has identified many best practices for engineering robust privacy protections for sensitive data. Although they can be formalized in a mathematically rigorous way, we present them at a more intuitive level, leveraging the following privacy definitions as sources of examples.

*Definition 1* ( $\epsilon$ -differential privacy<sup>9,11</sup>). An algorithm  $\mathcal{M}$  satisfies  $\epsilon$ -differential privacy if for each of its possible outputs  $\omega$  and for every pair

of databases  $D_1, D_2$  that differ on the addition or removal of a single record,  $P(\mathfrak{M}(D_1) = \omega) \leq e^\epsilon P(\mathfrak{M}(D_2) = \omega)$ .

Intuitively,  $\epsilon$ -differential privacy guarantees that adding or removing a record from the data will have little effect on the output of a privacy mechanism  $\mathfrak{M}$ . For small  $\epsilon$ , it means  $\mathfrak{M}$  will probably produce the same sanitized output regardless of whether or not Bob's record is in the data.

How should a data curator choose  $\epsilon$ ? Consider a highly targeted query about an individual (such as asking if Bob's record is in the data). For  $\epsilon$ -differential privacy, the most revealing privacy mechanism answers truthfully with probability  $e^\epsilon/(1 + e^\epsilon)$  and falsely with probability  $1/(1 + e^\epsilon)$ .<sup>24</sup> When  $\epsilon$  is close to 0, both these probabilities are close to  $1/2$ , and little information is provided; the mechanism is almost as likely to lie as respond truthfully; for example, when  $\epsilon = 0.1$ , the true answer probability is  $\approx 0.525$ , and when  $\epsilon = 0.01$ , the probability is  $\approx 0.502$ . We recommend choosing  $\epsilon$  based on how close the curator wants this value to be to  $1/2$ .

The Laplace mechanism is a popular mechanism for  $\epsilon$ -differential privacy. Let  $f$  be a function that computes a vector of query answers on the data. To

each query answer, the Laplace mechanism adds an independent Laplace random variable with mean 0 and standard deviation  $\sqrt{2}S(f)/\epsilon$ , where  $S(f)$  is the global sensitivity of  $f$ —the largest possible change in  $f$  due to the addition of one record, or the maximum of  $\|f(D_1) - f(D_2)\|_1$  over pairs of databases  $D_1, D_2$  that differ in one record. Intuitively, the noise masks the influence of any single record on the result of  $f$ . Now consider:

**Definition 2** (*k*-anonymity.<sup>34,35</sup>) Given a set  $Q$  of attributes, known as the quasi-identifier, a table is *k*-anonymous if every record in it has the same quasi-identifier values as *k*−1 other records. An algorithm satisfies *k*-anonymity if it outputs only *k*-anonymous tables.

*k*-anonymity defends against one type of attack called a “linkage attack”—joining an external dataset that associates an identity (such as name) with the quasi-identifier (such as ZIP code and age) to a *k*-anonymous table containing this publicly available quasi-identifier. Its goal is to prevent the matching of an identity to a single tuple in the *k*-anonymous table; clearly, there will always be at least *k* candidates in the join result. *k*-anonymity mechanisms usually operate by coarsening attributes (such as dropping digits from ZIP codes and changing ages to age ranges); see Figure 1 for two examples of *k*-anonymous tables.

Likewise, privacy-mechanism designers should always assume attackers are smarter than they are. Just because the designer of a privacy mechanism cannot deduce sensitive information from the output of a piece of software, an adversary will also fail. A well-engineered privacy definition will overcome these disadvantages, protecting sensitive information from clever attackers who know how  $\mathfrak{M}$  operates. We explain how in subsequent sections.

**Post-processing.** A privacy definition determines the mechanisms that data curators can trust to not leak sensitive information. Let  $\mathfrak{M}$  be one such mechanism, and suppose  $\mathcal{A}$  is some algorithm that can be applied to the output of  $\mathfrak{M}$ ; for example, suppose  $\mathfrak{M}$  creates synthetic data from its inputs, and  $\mathcal{A}$  builds a decision tree. Let the notation  $\mathcal{A} \circ \mathfrak{M}$  denote the composite algorithm that first applies  $\mathfrak{M}$  to the sensitive data and then runs  $\mathcal{A}$  on the sanitized output of  $\mathfrak{M}$ .

If  $\mathfrak{M}$  is trusted, should this composite algorithm  $\mathcal{A} \circ \mathfrak{M}$  also be trusted? Intuitively, the answer is yes. It would be very strange if a data curator released privacy-preserving synthetic data but then claimed building statistical models from this data is a violation of privacy.

A privacy definition is closed under post-processing if  $\mathcal{A} \circ \mathfrak{M}$  satisfies the constraints defining the privacy definition whenever  $\mathfrak{M}$  does. Differential privacy<sup>11</sup> satisfies this property, but *k*-anonymity does not.<sup>23</sup> Closure under post-processing has two important consequences: First, it ensures compatibility between a privacy definition and Kerckhoffs's principle; for example, some algorithms that satisfy *k*-anonymity are susceptible to a so-called minimality attack.<sup>13,36</sup> For each such *k*-anonymity mechanism  $\mathfrak{M}$ , it is possible to craft a post-processing algorithm  $\mathcal{A}$  that takes the output of  $\mathfrak{M}$ , undoes some of its data transformations, and outputs a new dataset having records with possibly unique quasi-identifier values that are vulnerable to linkage attacks with external data. That is, the composite algorithm  $\mathcal{A} \circ \mathfrak{M}$  does not satisfy the same conditions as  $\mathfrak{M}$ , or *k*-anonymity, and often reveals sensitive records.

By contrast, suppose an  $\epsilon$ -differentially private algorithm  $\mathfrak{M}$  is applied to the

### Security Without Obscurity

The process of sanitizing sensitive data through a privacy mechanism  $\mathfrak{M}$  must follow Kerckhoffs's principle<sup>21</sup> and ensure privacy even against adversaries who might know the details of  $\mathfrak{M}$ , except for the specific random bits it may have used. Better yet, the mechanism  $\mathfrak{M}$  must be revealed along with the sanitized output.

The reasons for making the mechanism  $\mathfrak{M}$  publicly known are twofold: First, history shows “security through obscurity” is unreliable in many applications; and, second, the output of  $\mathfrak{M}$  must be useful. This sanitized output often takes the form of a dataset or statistical model and could be intended to support scientific research. In such a case, statistical validity is an important concern, and statistically valid conclusions can be drawn only when scientists know precisely what transformations were applied to the base sensitive data.

**Figure 1. Examples of *k*-anonymity: (a) 4-anonymous table; (b) 3-anonymous table.**

Zip Code	Age	Disease
130**	25–30	None
130**	25–30	Stroke
130**	25–30	Flu
130**	25–30	Cancer
902**	60–70	Flu
902**	60–70	Stroke
902**	60–70	Flu
902**	60–70	Cancer

(a)

Zip Code	Age	Disease
130**	< 40	Cold
130**	< 40	Stroke
130**	< 40	Rash
1485*	≥ 40	Cancer
1485*	≥ 40	Flu
1485*	≥ 40	Cancer

(b)

data  $D$ , and the result  $\mathcal{M}(D)$  is published. Given knowledge of  $\mathcal{M}$ , a clever adversary can design an attack algorithm  $\mathcal{A}$  and run it on the published data to obtain the result  $\mathcal{A}(\mathcal{M}(D))$ . Note  $\mathcal{A}(\mathcal{M}(D))$  is the result of applying the composite algorithm  $\mathcal{A} \circ \mathcal{M}$  to the data  $D$ . Since  $\epsilon$ -differential privacy is closed under post-processing, the composite algorithm  $\mathcal{A} \circ \mathcal{M}$  still satisfies  $\epsilon$ -differential privacy and hence has the same semantics; the output of  $\mathcal{A} \circ \mathcal{M}$  is barely affected by the presence or absence of Bob's (or any other individual's) record in the database.


The second important consequence of closure under post-processing is how a data curator must express privacy definitions. Consider  $k$ -anonymity and  $\epsilon$ -differential privacy. By analogy to database-query languages, the definition of  $k$ -anonymity is declarative; that is, it specifies what we want from the output but not how to produce this output. On the other hand, differential privacy is more procedural, specifying constraints on the input/output behaviors of algorithms through constraints on probabilities (such as  $P(\mathcal{M}(D) = \omega)$ ). This is no coincidence; in order to achieve closure under post-processing, it is necessary that the privacy definition impose conditions on the probabilities (even when  $\mathcal{M}$  is deterministic) rather than on the syntactic form of the outputs.<sup>22</sup>

**Composition.** We introduce the concept of composition with an example. Suppose the 4-anonymous table in Figure 1 was generated from data from Hospital A, while the 3-anonymous table in Figure 1 was generated by Hospital B. Suppose Alice knows her neighbor Bob was treated by both hospitals for the same condition. What can Alice infer about, say, Bob's private records? Bob corresponds to an anonymized record in each table. By matching ZIP code, age, and disease, Alice can deduce that Bob must have had a stroke. Each anonymized table individually might have afforded Bob some privacy, but the combination of the two tables together resulted in a privacy breach. The degradation in privacy that results from combining multiple sanitized outputs is known as “composition.”<sup>14</sup>

*Self-composition.* “Self-composition” refers to the scenario where the sanitized outputs are all produced



**The privacy challenge is that sensitive information can be inferred in many ways from the data releases.**



from privacy mechanisms that satisfy the same privacy definition. Fundamental limits on a privacy definition's ability to withstand composition are part of a growing literature inspired by the results of Dinur and Nissim<sup>7</sup> who showed that the vast majority of records in a database of size  $n$  can be reconstructed when  $n \log(n)^2$  statistical queries are answered, even if each answer has been arbitrarily altered to have up to  $o(\sqrt{n})$  error; that is, a distortion that is less than the natural variation of query answers that an adversary would get from collecting a sample of size  $n$  from a much larger population.

Despite such negative results that limit the number of times a private database can be queried safely, there can be a graceful degradation of privacy protections, as in the case of  $\epsilon$ -differential privacy. If  $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k$  are algorithms such that each  $\mathcal{M}_i$  satisfies  $\epsilon_i$ -differential privacy, then the combination of their sensitive outputs satisfies  $\epsilon$ -differential privacy with  $\epsilon = \epsilon_1 + \dots + \epsilon_k$ ;<sup>30</sup> more formally, this privacy level is achieved by the algorithm  $\mathcal{M}$  running mechanisms  $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k$  on the input data and releases all their outputs. The end result thus does not reveal any record deterministically while still satisfying differential privacy but with a linear degradation in the privacy parameter.

Self-composition has another practical benefit—simplifying the design of privacy-preserving algorithms. Complicated mechanisms can be built modularly from simpler mechanisms in the same way software is built from functions. By controlling the information leakage of each component individually, a privacy-mechanism designer can control the information leakage of the entire system. In the case of  $\epsilon$ -differential privacy, the privacy parameter  $\epsilon$  of the final mechanism is at most the sum of the privacy parameters of its components.<sup>4,30</sup>

*Composition with other mechanisms.* The data curator must also consider the effect on privacy when the mechanisms do not satisfy the same privacy definition. As an example,<sup>24,26</sup> consider a database where each record takes one of  $k$  values. Let  $x_1, x_2, \dots, x_k$  denote the number of times each of these values appears in the database; they are histogram counts. Let  $\mathcal{M}_1$  be

a mechanism that releases the sums  $x_1 + x_2, x_2 + x_3, \dots, x_{k-1} + x_k$ . Note  $\mathcal{M}_1$  does not satisfy  $\epsilon$ -differential privacy. Moreover, the knowledge of any one count  $x_i$ , combined with the output of  $\mathcal{M}_1$ , would reveal all the original counts. Now consider a mechanism  $\mathcal{M}_2$  that adds noise drawn from a Laplace distribution, with variance  $2/\epsilon^2$ , independently, to each histogram count, so its output consists of  $k$  noisy counts  $\tilde{x}_1, \dots, \tilde{x}_k$ . Mechanism  $\mathcal{M}_2$  does satisfy  $\epsilon$ -differential privacy;<sup>9</sup> it is the Laplace mechanism mentioned earlier.

What is the effect of the combined release of the sanitized outputs of  $\mathcal{M}_1$  and  $\mathcal{M}_2$ ? From  $\tilde{x}_1$ , we have a noisy estimate of  $x_1$ . From the quantity  $x_1 + x_2$  and the noisy value  $\tilde{x}_2$ , we can obtain another independent estimate of  $x_1$ . Combining  $x_1 + x_2, x_2 + x_3$ , and  $\tilde{x}_3$  we get yet another estimate. Overall, there are  $k$  independent noisy estimates of  $x_1$  that can be averaged together to get a final estimate with variance  $2/(k\epsilon^2)$ , which is  $k$  times lower than what we could get from  $\mathcal{M}_2$  alone. This example illustrates why there is a recent push for creating flexible privacy definitions that can account for prior releases of information (such as the output of  $\mathcal{M}_1$ ) to control the overall inference.<sup>2,15,25</sup>

**Convexity.** Consider a privacy definition satisfied by two mechanisms,  $\mathcal{M}_1$  and  $\mathcal{M}_2$ . We can create an algorithm  $\mathcal{M}^{(p)}$ , or their “convex combination,” that randomly chooses among them; with probability  $p$  it applies  $\mathcal{M}_1$  to its input and with probability  $1-p$  it applies  $\mathcal{M}_2$ . Why consider mechanisms like  $\mathcal{M}^{(p)}$ ? Convex combinations like  $\mathcal{M}^{(p)}$  could provide better worst-case error guarantees for some queries than either  $\mathcal{M}_1$  or  $\mathcal{M}_2$  for reasons similar to why mixed strategies may be preferred over pure strategies in game theory.

Now, should we trust  $\mathcal{M}^{(p)}$  to protect privacy? It is reasonable to do so because the only thing  $\mathcal{M}^{(p)}$  does is add additional randomness into the system.<sup>22,23</sup> We say a privacy definition is convex if every convex combination of its mechanisms also happens to satisfy that privacy definition. Convex privacy definitions have useful semantic properties we discuss in more detail in the next section.

**Minimizing probabilistic failure.** Consider a private record that can be expressed in one bit; that is, 1 if Bob

## A naive implementation of a privacy-preserving algorithm may not satisfy the requirements of a chosen privacy definition.

has cancer and 0 otherwise. We add noise from a standard Gaussian distribution and release the result, which happens to be 10. If Bob’s bit is 1, then we are 13,000 times more likely to observe a noisy value of 10 than if Bob’s bit is 0. We have thus almost certainly discovered the value of Bob’s bit.

One can argue that observing a noisy value this large is so unlikely (regardless of the value of Bob’s bit) that such a privacy breach is very rare and hence can be ignored. Such reasoning has led to relaxations of privacy definitions that allow guarantees to fail with a small probability  $\delta$ ; one example is the relaxation  $(\epsilon, \delta)$ -differential privacy, which can produce more accurate data-mining results.

*Definition 3*  $(\epsilon, \delta)$ -differential privacy.<sup>10,11</sup> Let  $\mathcal{M}$  be an algorithm and  $\mathcal{S}$  be its set of possible outputs.  $\mathcal{M}$  satisfies  $(\epsilon, \delta)$ -differential privacy if for all subsets  $\mathcal{B} \subset \mathcal{S}$  and for all pairs of databases  $D_1, D_2$  differing on the value of a single record,  $P(\mathcal{M}(D_1) \in \mathcal{B}) \leq e^\epsilon P(\mathcal{M}(D_2) \in \mathcal{B}) + \delta$ .

The decision whether to always provide guarantees or allow privacy protections to fail with a small probability is application-specific and depends on the stakes involved. It is a privacy/utility trade-off having consequences with different levels of subtlety. For instance, let  $\mathcal{M}$  be the algorithm that outputs  $\perp$  with probability  $1-\delta$  and outputs the input dataset with probability  $\delta$ . This  $\mathcal{M}$  satisfies the more relaxed conditions of  $(\epsilon, \delta)$ -differential privacy. Similarly, consider an algorithm  $\mathcal{M}^*$  that returns the record of a randomly selected individual from the input dataset. If the number of records is  $N$  and if  $N > 1/\delta$  then  $\mathcal{M}^*$  satisfies  $(\epsilon, \delta)$ -differential privacy yet always violates the privacy of some individual.

Worth noting is that  $\epsilon$ -differential privacy and the relaxed  $(\epsilon, \delta)$ -differential privacy both offer the same high-level guarantee—the output distribution of a mechanism is barely affected by the value of any individual record. Still, privacy relaxation may consistently cause privacy violations, while the former will not. Reasoning about attackers can help data curators set parameter values that limit such information leakages<sup>14</sup> and (as we discuss later) provide new perspectives on achievable guarantees.



**Implementation concerns.** As with many aspects of security, moving from theory to practice requires great care. In particular, a naive implementation of a privacy-preserving algorithm may not ensure privacy even though the algorithm is theoretically proven to satisfy the requirements of a chosen privacy definition. One problem arises from side-channels. Consider a program that processes a sensitive database and behaves as follows: If Bob’s record is in the database, it produces the output 1 after one day; if Bob’s record is not in the database, it outputs 1 right away. The output is the same, no matter what the database is. But by observing the time taken for a result to be output, we learn something about the database.<sup>18</sup>

Another concern arises when the theoretical algorithms base their security properties on exact computation that may be beyond the limits of digital computers.<sup>5,31</sup> The most common example is the addition of noise from continuous distributions (such as Gaussian and Laplace). For most floating-point implementations, an analysis of the bit patterns yields additional information about the input data.<sup>31</sup>

Finally, many privacy mechanisms rely on a random number generator. A provably secure implementation of a privacy-preserving algorithm must be tailored to the quality of the randomness of the bits.<sup>8</sup>

## A Generic Recipe

Privacy definitions that are convex, closed under post-processing, and require protections for all outputs of a mechanism  $\mathcal{M}$  all have a similar format and can be written in terms of linear constraints,<sup>28</sup> as in the following generic template:

*Definition 4* (a generic privacy definition). Let  $D_1, D_2, \dots$  be the collection of possible input datasets. For some fixed constants  $S_1^{(1)}, S_1^{(2)}, \dots, S_2^{(1)}, S_2^{(2)}, \dots, S_3^{(1)}, S_3^{(2)}, \dots$  an algorithm  $\mathcal{M}$  must satisfy the following conditions for every possible output the algorithm  $\omega$  can produce:

$$\sum_j S_j^{(1)} P(\mathcal{M}(D_j) = \omega) \leq 0; \quad \sum_j S_j^{(2)} P(\mathcal{M}(D_j) = \omega) \leq 0$$

$$\sum_j S_j^{(3)} P(\mathcal{M}(D_j) = \omega) \leq 0; \quad \dots \text{ etc.}$$

To evaluate a proposed privacy definition, a good sanity check for current best practices is thus to verify whether

or not the algorithm  $\mathcal{M}$  can be expressed as linear constraints on the probabilistic behavior of algorithms, as in Definition 4; for example,  $k$ -anonymity does not fit this template,<sup>28</sup> but with  $\epsilon$ -differential privacy, there is a linear constraint  $P(\mathcal{M}(D_{j_1}) = \omega) - e^\epsilon P(\mathcal{M}(D_{j_2}) = \omega) \leq 0$  for every pair of datasets  $D_{j_1}, D_{j_2}$  that differ on the presence of one rec-ord. We next discuss some semantic guarantees achievable through this template.

**Good and bad disclosures.** Even when published data allows an analyst to make better inferences about Bob, Bob’s privacy has not necessarily been violated by this data. Consider Bob’s nosy but uninformed neighbor Charley, who knows Bob is a life-long chain-smoker and thinks cancer is unrelated to smoking. After seeing data from a smoking study, Charley learns smoking causes cancer and now believes that Bob is very likely to suffer from it. This inference may be considered benign (or unavoidable) because it is based on a fact of nature.

Now consider a more nuanced situation where Bob participates in the aforementioned smoking study, the data is processed by  $\mathcal{M}$ , and the result  $\omega$  (which shows that smoking causes cancer) is published. Charley’s beliefs about Bob can change as a result of the combination of two factors: by him learning that smoking causes cancer, and since Bob’s record may have affected the output of the algorithm. This latter factor poses the privacy risks. There are two approaches to isolate and measure whether Charley’s change in belief is due to Bob’s record and not due to his knowledge of some law of nature—“counterfactuals”<sup>12,25,33</sup> and “simulatability.”<sup>2,15,29</sup>

**Privacy via counterfactuals.** The first approach<sup>12,25,33</sup> based on counterfactual reasoning is rooted in the idea that learning the true distribution underlying the private database is acceptable, but learning how a specific individual’s data deviates from this distribution is a privacy breach.

Consider pairs of alternatives (such as “Bob has cancer” and “Bob is healthy”). If the true data-generating distribution  $\theta$  is known, we could use it to understand how each alternative affects the output of  $\mathcal{M}$  (taking into account uncertainty about the data) by

considering the probabilities  $P_\theta(\mathcal{M} \text{ outputs } \omega \mid \text{Bob has cancer})$  and  $P_\theta(\mathcal{M} \text{ outputs } \omega \mid \text{Bob is healthy})$ . Their ratio is known as the “odds ratio.” It is the multiplicative factor that converts the initial odds of Bob having cancer (before seeing  $\omega$ ) into the updated odds (after seeing  $\omega$ ). When the odds ratio is close to 1, there is little change in the odds, and Bob’s privacy is protected.

Why does this work? If the reasoning is done using the true distribution, then we have bypassed the change in beliefs due to learning about laws of nature. After seeing  $\omega$ , the change in Charley’s beliefs depends only on the extent to which  $\omega$  is influenced by Bob (such as it was computed using Bob’s record).

What if the true distribution is unknown? To handle this scenario, the data curator can specify a set  $\Xi$  of plausible distributions and ensure reasoning with any of them is harmless; the corresponding odds ratios are all close to 1. A counterfactual-based privacy definition would thus enforce constraints like  $P_\theta(\mathcal{M} \text{ outputs } \omega \mid \text{Bob has cancer}) \leq e^\epsilon P_\theta(\mathcal{M} \text{ outputs } \omega \mid \text{Bob is healthy})$  for all possible  $\omega$ , for various pairs of alternatives and distributions  $\theta$ . When written mathematically, these conditions turn into linear constraints, as in the generic template (Definition 4).

**Privacy via simulatability.** The second approach,<sup>2,15,29</sup> based on simulatability, is motivated by the idea that learning statistics about a large population of individuals is acceptable, but learning how an individual differs from the population is a privacy breach. The main idea is to compare the behavior of an algorithm  $\mathcal{M}$  with input  $D$  to another algorithm, often called a “simulator,”  $\mathcal{S}$  with a safer input  $D'$ ; for example,  $D'$  could be a dataset that is obtained by removing Bob’s record from  $D$ . If the distribution of outputs of  $\mathcal{M}$  and  $\mathcal{S}$  are similar, then an attacker is essentially clueless about whether  $\omega$  was produced by running  $\mathcal{M}$  on  $D$  or by running  $\mathcal{S}$  on  $D'$ . Now  $\mathcal{S}$  does not know anything about Bob’s record except what it can predict from the rest of the records in  $D'$  (such as a link between smoking and cancer). Bob’s record is thus protected. Similarly, Alice’s privacy can be tested by considering different alterations where Alice’s record is removed instead of Bob’s record. If

$\mathcal{S}$  can approximately simulate the behavior of  $\mathcal{M}$  no matter what the true data  $D$  is and no matter what alteration was performed, then every individual record is protected.

Privacy definitions based on simulatability are generally more complex than those based on counterfactuals. To check whether  $\mathcal{M}$  satisfies the definitions, it is often necessary to find the appropriate simulator  $\mathcal{S}$ . However, in some cases, the privacy definitions can also be expressed using linear constraints, as in the generic privacy definition template.

**Counterfactuals vs. simulatability.** The differences between counterfactual and simulatability approaches depend on the nature of the data that must be protected. When the data records are independent of each other, properties of the data-generating distribution and properties of a population are essentially the same (due to the law of large numbers), in which case both approaches provide similar protection.

*Data correlations.* A difference arises when there is correlation between individuals. First, we consider a scenario when counterfactuals would be more appropriate. Suppose a database contains records about Bob and his relatives. Even if Bob's record is removed, Bob's susceptibility to various diseases can be predicted from the rest of the data because it contains his family's medical history. The general goal of privacy definitions based on simulatability is not to hide this inference but to hide how Bob's actual record differs from this prediction. On the other hand, if we include probabilistic models of how diseases are passed through genetics, then privacy definitions based on counterfactuals will try to prevent predictions about Bob and his family. Intuitively, this happens because the actual family medical history is not a property of the data-generating distribution but of a sample from that distribution. Since the family medical history is correlated with Bob's record, it would allow better predictions about how Bob deviates from the data-generating distribution; hence, it must be protected as well.

Next, we examine a situation where simulatability-based privacy definitions are more appropriate. Consider a social network where many profiles

of individuals are public. Private information about individuals is often predictable directly from the public profiles of their friends and contacts.<sup>37</sup> Even if Bob's profile is private, it is easy to collect information that is correlated with Bob. Here, privacy definitions based on simulatability are applicable, allowing data curators to process the social network data with algorithms  $\mathcal{M}$  that create outputs from which it is difficult to tell if Bob's record was used in the computation.

*Data constraints.* One difficulty in designing privacy definitions is accounting for public knowledge of constraints the input database must satisfy. Constraints may correlate the values of different records, arising due to, say, functional dependencies across attributes or prior exact releases of histograms. Correlations arising from constraints provide inference channels attackers could use to learn sensitive information. A privacy definition must thus account for them; for example, while Census data records must be treated confidentially, certain coarse population statistics must, by law, be released exactly in order to determine the number of Congressional Representatives for each state. More generally, if a histogram  $H$  of the data has been released exactly, how can a data curator choose a privacy definition, and hence constraints on  $\mathcal{M}$  to account for the information in the histogram so any subsequent data release via  $\mathcal{M}$  is able to ensure privacy? Complete solutions to this problem are open but appear to be easier for approaches based on counterfactuals if we use data-generating distributions that are conditioned on the histogram, or  $P(D|H)$ .<sup>25</sup> For approaches based on simulatability, there is more of a challenge since data-alteration techniques consistent with previously released information must be developed; recall, they provide the guarantee that an attacker would not be able to reliably determine whether the original dataset or altered dataset was used in the computation. It is important to note, too, that constraints on the input data, and especially those arising from prior releases, can be exploited for better utility.

*Interpretations of differential privacy.* These two approaches for defin-

ing semantics for privacy definitions also provide two ways of interpreting  $\epsilon$ -differential privacy. The simulatability argument shows an algorithm satisfying  $\epsilon$ -differential privacy provides the following protection: an attacker cannot detect whether  $\mathcal{M}$  was run on the original data or on altered data from which any given record was removed.<sup>2,14</sup> This is true no matter how knowledgeable the attacker is, as long as the data alteration is consistent with what is known about the data; if not, additional leakage can occur, as explained in the earlier discussion on composition with other mechanisms. From a different perspective, the counterfactual argument shows an algorithm  $\mathcal{M}$  satisfying  $\epsilon$ -differential privacy prevents an attacker from learning how an individual differs from the data-generating distribution precisely when all records are independent.<sup>25</sup>

### Example: Blowfish

We illustrate this discussion with Blowfish,<sup>19</sup> a new class of privacy definitions that follows the generic privacy template in Definition 4. Like differential privacy, Blowfish definitions satisfy a number of desirable properties we outlined earlier, including Kerckhoffs's principle, self-composition, convexity, and closure under post-processing. The privacy goals of Blowfish definitions have both a counterfactual and a simulatability interpretation. In addition to satisfying these properties, Blowfish definitions improve on differential privacy by including a generalized and formal specification of what properties of an individual in the data are kept private and by accounting for external knowledge about constraints in the data. Blowfish thus captures part of the privacy design space. In the rest of this section, we describe how data owners can use Blowfish to customize privacy protections for their applications.

Blowfish definitions take two parameters: privacy  $\epsilon$  (similar to differential privacy) and policy  $P = (\mathcal{T}, \mathcal{G}, \mathcal{I}_Q)$  allowing data curators to customize privacy guarantees. Here,  $\mathcal{T}$  is the set of possible record values,  $\mathcal{Q}$  is a set of publicly known constraints on the data, and  $\mathcal{I}_Q$  is the set of all possible datasets consistent with  $\mathcal{Q}$ . Specifying  $\mathcal{I}_Q$  allows a data curator to create

privacy definitions that can compose with prior deterministic data releases, thus avoiding some of the difficulties discussed earlier in the section on desiderata. To simplify the discussion, we set  $\mathcal{Q}$  to be the single constraint that the dataset has  $n$  records, in which case  $\mathcal{I}_{\mathcal{Q}} = \mathcal{T}^n$ ; for more complicated constraints, see He<sup>19</sup> on Blowfish and Kifer and Machanavajjhala<sup>25</sup> on Pufferfish frameworks.

The final component of the policy is  $G = (\mathcal{T}, E)$ , or the “discriminative secret graph.” The vertices in  $G$  are the possible values a record can take. Every edge  $(x, y) \in E$  describes a privacy goal with both counterfactual and simulatability interpretations. From the simulatability viewpoint, changing a single record from  $x$  to  $y$  (or vice versa) will not cause a significant change in the probability of any output. From the counterfactual viewpoint, if records are independent, an attacker could estimate the odds of a new record having value  $x$  vs.  $y$ , but estimated odds about any individual in the data would not differ significantly from this value. Using this graph  $G$ , we define the concept of neighboring databases, then formally define the Blowfish framework:

*Definition 5 (G-Neighbors).* Let  $P = (\mathcal{T}, G, \mathcal{T}^n)$  be a discriminative secret graph. Two datasets  $D_1, D_2 \in \mathcal{T}^n$  are called  $G$ -neighbors if for some edge  $(x, y) \in E$  and some dataset  $D \in \mathcal{T}^{n-1}$ ,  $D_1 = D \cup \{x\}$  and  $D_2 = D \cup \{y\}$ .

*Definition 6 (( $\epsilon, P$ )-Blowfish Privacy).* Let  $P = (\mathcal{T}, G, \mathcal{T}^n)$  be a policy. An algorithm  $\mathfrak{M}$  satisfies  $(\epsilon, P)$ -Blowfish privacy if for all outputs  $\omega$  of the algorithm  $\mathfrak{M}$  and all  $G$ -neighbors  $D_1, D_2$  we have  $P(\mathfrak{M}(D_1) = \omega) \leq e^{\lfloor d(x,y)/10 \rfloor \epsilon} P(\mathfrak{M}(D_2) = \omega)$ .

This privacy definition clearly matches the generic template of Definition 4. We now examine some policies and their applications.

*Full domain.* Consider a policy  $P_K = (\mathcal{T}, G, \mathcal{T}^n)$  where  $K$  is a complete graph, and every pair of values in the domain  $\mathcal{T}$  are connected. The result is that two datasets are neighbors if they differ (arbitrarily) in any one record.  $(\epsilon, P_K)$ -Blowfish privacy is equivalent to a popular variant of differential privacy<sup>11</sup> that requires  $P(\mathfrak{M}(D_1) = \omega) \leq e^{\lfloor d(x,y)/10 \rfloor \epsilon} P(\mathfrak{M}(D_2) = \omega)$  for all  $\omega$  and for all pairs of datasets  $D_1, D_2$  that differ (arbitrarily) in the value (rather than presence/absence) of one record.

*Partitioned.* Let us partition the do-

## Learning population statistics is acceptable, but learning how an individual differs from the population is a privacy breach.

main  $\mathcal{T}$  into  $p$  mutually exclusive subsets, with  $\mathcal{P} = \{P_1, \dots, P_p\}$ . Consider a graph  $G^{\mathcal{P}} = (\mathcal{T}, E)$ , where any two values  $x, y$  are connected by an edge if and only if  $x$  and  $y$  appear in the same partition. Each connected component of  $G^{\mathcal{P}}$  is thus a clique corresponding to one of the  $P_i$ . Now, two datasets  $D_1$  and  $D_2$  are neighbors if  $D_2$  can be obtained from  $D_1$  by replacing the value of one record with a new value belonging to the same partition. For example, let  $\mathcal{T}$  be the set of all disease outcomes, partitioned into three subsets: healthy cases, communicable diseases, and non-communicable diseases. Let us use the graph  $G^{\mathcal{P}}$  corresponding to this partition in our Blowfish policy. An algorithm  $\mathfrak{M}$  satisfying Definition 6 comes with the guarantee that the probabilities of its outputs do not change substantially if one communicable disease is replaced with another communicable disease or a healthy case with another healthy case, or a simulatability interpretation.

What about replacing a noncommunicable disease with a communicable disease? Can the algorithm’s output probabilities be significantly different in such a case? The answer is yes. In fact, this policy allows algorithms to publish the exact status of each individual—healthy, contagious, or noncontagious—and approximate histograms of each disease. However, specific details (such as which person has which contagious disease) are protected. Such behavior may be desirable in certain health-care applications where some facts must be disclosed but further details kept confidential.

*Distance threshold.* Many applications involve a concept of distance between records; for instance, the distance between two age values can be the absolute difference, and the distance between two points on a plane can be the straightline Euclidean distance or the Manhattan distance along a grid. Given a distance metric  $d$ , one can define a discriminative secret graph  $G^{d,\theta}$  in which only nearby points are connected. That is,  $(x, y) \in E$  only when  $d(x, y) < \theta$  for some threshold  $\theta$ ; for example, if  $\mathcal{T}$  is the set of all points on Earth, and  $d$  is the orthodromic distance between pairs of points, we can set  $\theta = 10$  miles, so valid record locations are connected to other valid rec-

ord locations that are within 10 miles of each other. In general, if an individual's location  $x$  (in dataset  $D_1$ ) was changed to another point  $y$  (resulting in a neighboring dataset  $D_2$ ), then an algorithm satisfying Blowfish with this policy will have the guarantee that for all outputs  $\omega$

$$P(\mathfrak{M}(D_1) = \omega) \leq e^{\lfloor d(x,y)/10 \rfloor \epsilon} P(\mathfrak{M}(D_2) = \omega)$$

An adversary may thus be able to detect the general geographic region of a target individual but unable to infer the location with a resolution better than 10 miles. Such a relaxed notion of privacy is reasonable when dealing with location data; individuals may not want disclosure of their precise locations but be less worried about disclosing their information at a coarser granularity (that may be obtained from other sources). As we show later, data output by mechanisms that satisfy such relaxed notions of privacy permit data mining results with greater accuracy than if data is generated using mechanisms that satisfy the stricter notion of differential privacy.

**Attribute.** Let  $\mathcal{T}$  be a multi-attribute domain with  $m$  attributes  $\mathcal{T} = A_1 \times A_2 \times \dots \times A_m$ . Consider a graph  $G^{attr,c}$  connecting any two values  $x$  and  $y$  that differ in at most  $c$  attribute values. A Blowfish policy with this graph is useful for location traces and genome data. For the former, attributes correspond to locations of an individual at different times. Neighboring datasets thus dif-

fer in at most  $c$  locations of a person, hiding the specific details about every sequence of  $c$  consecutive locations of an individual. In the genome case, an attribute corresponds to a specific position on the genome. Under this policy, an algorithm's output would be insensitive to changes to a block of up to  $c$  positions on the genome.

**Answering queries with Blowfish.**

Recall that adding Laplace noise with 0 mean and  $\sqrt{2}S(f)/\epsilon$  standard deviation to a function  $f$  (where  $S(f)$  is the sensitivity of  $f$ ) ensures  $\epsilon$ -differential privacy. Blowfish, with a policy  $P = (\mathcal{T}, G, \mathcal{T}^n)$  is also compatible with additive Laplace noise and requires an often smaller standard deviation of  $\sqrt{2}S(f, G)/\epsilon$  where  $S(f, G)$  is the policy-specific global sensitivity of  $f$ —the largest difference  $\|f(D_1) - f(D_2)\|_1$  over all datasets  $D_1, D_2$  that are  $G$ -neighbors.

Consider a multidimensional record domain  $\mathcal{T} = A_1 \times A_2 \times \dots \times A_m$  where each attribute is numeric. Let  $q_{sum}$  denote the function that sums all the records together; that is, for each attribute, it computes the sum of the values that appear in the data. Let  $a_i$  and  $b_i$  denote the maximum and minimum values in attribute  $A_i$ . The global sensitivity  $S(q_{sum})$  of  $q_{sum}$  is  $\sum_{i=1}^m \max\{|a_i|, |b_i|\}$ . The policy-specific global sensitivity of  $q_{sum}$  under Blowfish policies is usually much smaller. In the case of the distance threshold policy  $G^{d,\theta}$  with  $d$  being the  $L_1$  Manhattan distance,  $S(q_{sum}, G^{d,\theta})$  is only  $\theta$ . Consider a single attribute domain Age and further suppose the age

values range from 0 to 100. The global sensitivity of  $q_{sum}$  is 100. The policy-specific sensitivity of  $q_{sum}$  under  $G^{7^{1.5}}$  is only 5. If, instead, the policy used a partition graph  $G^P$  that partitions age into ranges (such as  $\{0 - 10, 11 - 20, 21 - 30, \dots, 91 - 100\}$ ), then the policy-specific global sensitivity is only 10. Finally, with the attribute policy,  $S(q_{sum}, G^{attr,1}) = \max(a_i - b_i)$ .

**K-means clustering.** For a specific data-mining result, consider an application of Blowfish to  $k$ -means clustering.

*Definition 7 (K-means clustering).* Given a set of  $n$  vectors  $\{x_1, \dots, x_n\}$ , the  $k$ -means clustering problem is to divide these  $n$  records among  $k$  clusters  $S = \{S_1, \dots, S_k\}$ , where  $k \leq n$ , so as to minimize the objective function

$$\sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|_2^2, \quad (1)$$

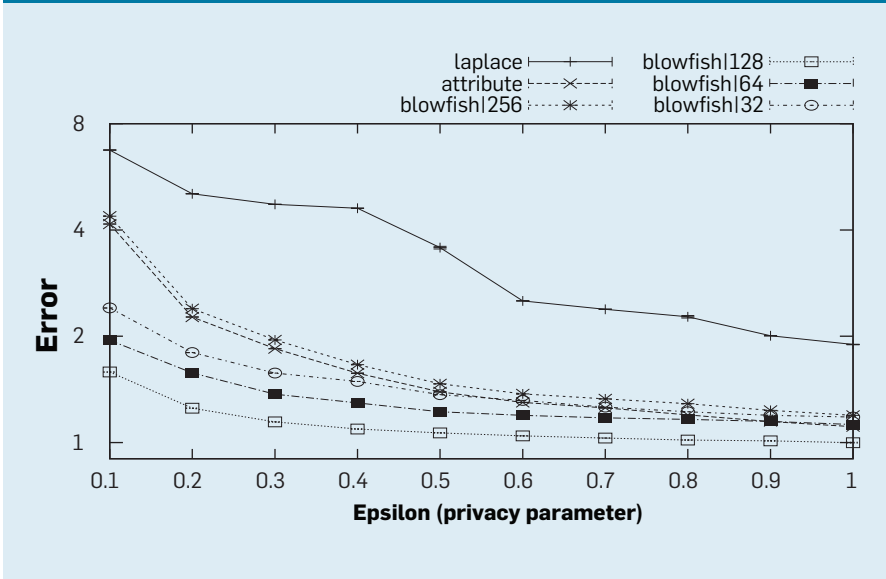
where  $\mu_i = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j$  is the mean of cluster  $S_i$ .

The iterative (non-private)  $k$ -means clustering algorithm initializes a set of  $k$  centroids  $\{\mu_1, \mu_2, \dots, \mu_k\}$ , one for each cluster. These centroids are iteratively updated in two steps: assign each  $x_j$  to the cluster with the nearest centroid, and set each centroid  $\mu_i$  to be the mean of the vectors of its corresponding cluster. The algorithm terminates after a certain number of iterations or when the centroids do not change significantly.

Each iteration (the two steps) are easily modified to satisfy  $\epsilon$ -differential privacy<sup>4,30</sup> and Blowfish.<sup>19</sup> These steps require access to the answers to two queries:  $q_{hist}$ , which returns the number of points in each cluster, and  $q_{sum}$ , or the sum of the points in each cluster. As discussed earlier,  $q_{sum}$  can be answered through the Laplace mechanism. Analogously,  $q_{hist}$  can be answered with the Laplace mechanism because it has global sensitivity  $S(q_{hist}) = 1$  (for differential privacy) and policy-specific global sensitivity  $S(f, G) = 2$  for all Blowfish policies discussed here. The policy-specific sensitivity of the  $q_{sum}$  query under Blowfish policies is typically much smaller than its global sensitivity so we would thus expect more accurate clustering under the Blowfish privacy definitions.

Figure 2 confirms this improvement in utility. For the clustering task,

Figure 2. K-means under several Blowfish policies.



we used a small sample of the skin-segmentation dataset,<sup>3</sup> or 1%, which is approximately 2,500 instances, in order to make the problem challenging. Each instance corresponds to the RGB intensities from face images, and each intensity ranges from 0 to 255. The  $x$ -axis is the privacy parameter  $\epsilon$ , and on the  $y$ -axis (note the log scale) we report the error incurred by the privacy-preserving algorithms. We measure the error as the ratio between the squared error (Equation 1) attained by the privacy-preserving algorithms to that achieved by the non-private  $k$ -means algorithm after 10 iterations that was sufficient for the convergence of the non-private algorithm. The Laplace mechanism for  $\epsilon$ -differential privacy incurred the most error. Using the  $G^{attr,1}$  policy already reduces the error by at least a factor of 1.5. The error is further reduced when using  $G^{L_{1,\theta}}$ , for  $\theta \in \{256, 128, 64, 32\}$ . It is interesting to note the error does not increase monotonically as we increase  $\theta - G^{L_{1,128}}$ —an improvement of 3x and 2x over differential privacy for  $\epsilon \leq 0.5$  and  $\epsilon > 0.5$ , respectively. One explanation is that small amounts of error can help avoid local minima while clustering.

### Conclusion

Privacy definitions are formal specifications an algorithm must satisfy to protect sensitive information within data. Our experience shows that designing robust privacy definitions often requires a great deal of subtlety. Our goal is to present some of the major considerations in this design process, along with example privacy definitions and resulting privacy mechanisms. We hope this discussion inspires additional curiosity about the technical nature of privacy.

### Acknowledgment

This work is supported by the National Science Foundation under Grants 1054389 and 1253327. 

### References

1. Barbaro, M. and Zeller, T. A face is exposed for AOL searcher no. 4417749. *The New York Times* (Aug. 9, 2006).
2. Bassily, R., Groce, A., Katz, J., and Smith, A. Coupled-worlds privacy: Exploiting adversarial uncertainty in statistical data privacy. In *Proceedings of the 54<sup>th</sup> IEEE Annual Symposium on Foundations of Computer Science* (Berkeley, CA, Oct. 27–29). IEEE Computer Society Press, Washington, D.C., 2013, 439–448.
3. Bhatt, R. and Dhall, A. *Skin Segmentation Dataset*. Machine Learning Repository Center for Machine Learning and Intelligent Systems, University of

- California, Irvine, 2012: <https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation/>
4. Blum, A., Dwork, C., McSherry, F., and Nissim, K. Practical privacy: The SUQ framework. In *Proceedings of the 24<sup>th</sup> ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Baltimore, MD, June 13–16). ACM Press, New York, 2005, 128–138.
5. Chaudhuri, K., Monteleoni, C., and Sarwate, A.D. Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12 (July 2011), 1069–1109.
6. Coull, S., Collins, M., Wright, C., Monrose, F., and Reiter, M. On Web browsing privacy in anonymized netflows. In *Proceedings of 16<sup>th</sup> USENIX Security Symposium* (Boston, MA, Aug. 6–10). USENIX Association, Berkeley, CA, 2007, 23:1–23:14.
7. Dinur, I. and Nissim, K. Revealing information while preserving privacy. In *Proceedings of the 22<sup>nd</sup> ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (San Diego, CA, June 9–12). ACM Press, New York, 2003, 202–210.
8. Dodis, Y., López-Alt, A., Mironov, I., and Vadhan, S.P. Differential privacy with imperfect randomness. In *Proceedings of the 32<sup>nd</sup> Annual Cryptology Conference* (Santa Barbara, CA, Aug. 19–23). Springer-Verlag, Berlin, Heidelberg, 2012, 497–516.
9. Dwork, C. Differential privacy. In *Proceedings of the 33<sup>rd</sup> International Colloquium on Automata, Languages and Programming* (Venice, Italy, July 9–16). Springer-Verlag, Berlin, Heidelberg, 2006, 1–12.
10. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24<sup>th</sup> Annual International Conference on the Theory and Applications of Cryptographic Techniques* (Saint Petersburg, Russia, May 28–June 1). Springer-Verlag, Berlin, Heidelberg, 2006, 486–503.
11. Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography Conference* (Columbia University, New York, Mar. 4–7). Springer-Verlag, Berlin, Heidelberg, 2006, 265–284.
12. Evfimievski, A., Gehrke, J., and Srikant, R. Limiting privacy breaches in privacy-preserving data mining. In *Proceedings of the 22<sup>nd</sup> ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (San Diego, CA, June 9–12). ACM Press, New York, 2003, 211–222.
13. Fang, C. and Chang, E.-C. Information leakage in optimal anonymized and diversified data. In *Proceedings of the 10<sup>th</sup> Information Hiding* (Santa Barbara, CA, May 19–21). Springer-Verlag, Berlin, Heidelberg, 2008, 30–44.
14. Ganta, S.R., Kasiviswanathan, S.P., and Smith, A. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Las Vegas, Aug. 24–27). ACM Press, New York, 2008, 265–273.
15. Gehrke, J., Lui, E., and Pass, R. Towards privacy for social networks: A zero-knowledge-based definition of privacy. In *Proceedings of the Theory of Cryptography Conference* (Providence, RI, Mar. 28–30). Springer-Verlag, Berlin, Heidelberg, 2011, 432–449.
16. Goljan, M. and Fridrich, J. Camera identification from scaled and cropped images. In *Proceedings of Electronic Imaging, Forensics, Security, Steganography, and Watermarking of Multimedia Contents* (Feb. 26, 2008); <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=812538>
17. Greveler, U., Justus, B., and Loehr, D. Forensic content detection through power consumption. In *Proceedings of the IEEE International Conference on Communications* (Ottawa, Canada, June 10–15). IEEE Press, Piscataway, NJ, 2012, 6759–6763.
18. Haeblerl, A., Pierce, B.C., and Narayan, A. Differential privacy under fire. In *Proceedings of the 20<sup>th</sup> USENIX Conference on Security* (San Francisco, CA, Aug. 8–12). USENIX Association, Berkeley, CA, 2011, 33–33.
19. He, X., Machanavajjhala, A., and Ding, B. Blowfish privacy: Tuning privacy-utility trade-offs using policies. In *Proceedings of the ACM SIGMOD/PODS International Conference on Management of Data* (Snowbird, UT, June 22–27). ACM Press, New York, 2014, 1447–1458.
20. Homer, N., Szlinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J.V., Stephan, D.A., Nelson, S.F., and Craig, D.W. Resolving individuals

- contributing trace amounts of DNA to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genetics* 4, 8 (Aug. 2008).
21. Kerckhoffs, A. La cryptographie militaire. *Journal des Sciences Militaires* 9 (Jan. 1983), 5–83.
22. Kifer, D. and Lin, B.-R. Towards an axiomatization of statistical privacy and utility. In *Proceedings of the 29<sup>th</sup> ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Indianapolis, IN, June 6–11). ACM Press, New York, 2010, 147–158.
23. Kifer, D. and Lin, B.-R. An axiomatic view of statistical privacy and utility. *Journal of Privacy and Confidentiality* 4, 1 (2012), 5–49.
24. Kifer, D. and Machanavajjhala, A. No free lunch in data privacy. In *Proceedings of the ACM SIGMOD/PODS International Conference on Management of Data* (Athens, Greece, June 12–16). ACM Press, New York, 2011, 193–204.
25. Kifer, D. and Machanavajjhala, A. A rigorous and customizable framework for privacy. In *Proceedings of the 31<sup>st</sup> Symposium on Principles of Database Systems* (Scottsdale, AZ, May 20–24). ACM Press, New York, 2012, 77–88.
26. Kifer, D. and Machanavajjhala, A. Pufferfish: A framework for mathematical privacy definitions. In *Transactions on Database Systems* 39, 1 (Jan. 2014), 3:1–3:36.
27. Kumar, R., Novak, J., Pang, B., and Tomkins, A. On anonymizing query logs via token-based hashing. In *Proceedings of the 16<sup>th</sup> International World Wide Web Conference* (Banff, Alberta, Canada, May 8–12). ACM Press, New York, 2007, 629–638.
28. Lin, B.-R. and Kifer, D. Towards a systematic analysis of privacy definitions. *Journal of Privacy and Confidentiality* 5, 2 (2014), 57–109.
29. Machanavajjhala, A., Gehrke, J., and M. Götz. Data publishing against realistic adversaries. In *Proceedings of the 35<sup>th</sup> International Conference on Very Large Data Bases* (Lyon, France, Aug. 24–28, 2009), 790–801.
30. McSherry, F.D. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of ACM SIGMOD/PODS International Conference on Management of Data* (Providence, RI, June 29–July 2). ACM Press, New York, 2009, 19–30.
31. Mironov, I. On significance of the least significant bits for differential privacy. In *Proceedings of the 19<sup>th</sup> ACM Conference on Computer and Communications Security* (Raleigh, NC, Oct. 16–18). ACM Press, New York, 2012, 650–661.
32. Narayanan, A. and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (Oakland, CA). IEEE Computer Society Press, Washington, D.C., 2008, 111–125.
33. Rastogi, V., Hay, M., Miklau, G., and Suciu, D. Relationship privacy: Output perturbation for queries with joins. In *Proceedings of the 28<sup>th</sup> ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Providence, RI, June 29–July 2). ACM Press, New York, 2009, 107–116.
34. Samarati, P. Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering* 13, 6 (Nov. 2001), 1010–1027.
35. Sweeney, L.  $k$ -anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems* 10, 5 (Oct. 2002), 557–570.
36. Wong, R., Fu, A., Wang, K., and Pei, J. Minimality attack in privacy-preserving data publishing. In *Proceedings of the 33<sup>rd</sup> International Conference on Very Large Data Bases* (University of Vienna, Austria, Sept. 23–27). VLDB Endowment, 2007, 543–554.
37. Zheleva, E. and Getoor, L. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18<sup>th</sup> International World Wide Web Conference* (Madrid, Spain, Apr. 20–24). ACM Press, New York, 2009, 531–540.

**Ashwin Machanavajjhala** (ashwin@cs.duke.edu) is an assistant professor in the Department of Computer Science at Duke University, Durham, NC.

**Daniel Kifer** (dkifer@cse.psu.edu) is an associate professor in the Department of Computer Science & Engineering at Penn State University, University Park, PA.

**A revealing picture of how personal health information searches become the property of private corporations.**

BY TIMOTHY LIBERT

# Privacy Implications of Health Information Seeking on the Web

PRIVACY ONLINE IS AN increasingly popular field of study, yet it remains poorly defined. “Privacy” itself is a word that changes according to location, context, and culture. Additionally, the Web is a vast landscape of specialized sites and activities that may only apply to a minority of users—making defining widely shared privacy concerns difficult. Likewise, as technologies and services proliferate, the line between on- and offline is increasingly blurred. Researchers attempting to make sense of this rapidly changing environment are frequently stymied by such factors.

Therefore, the ideal object of study is one that is inherently sensitive in nature, applies to the majority of users, and readily lends itself to analysis. The study of health privacy on the Web meets all of these criteria.

Health information has been regarded as sensitive since the time of the ancient Greeks. In the 5<sup>th</sup> century B.C., physicians taking the Hippocratic Oath were required to swear that: Whatever I see or hear in the lives of my patients...I will keep secret, as considering all such things to be private.<sup>21</sup> This oath is still in use today, and the importance of health privacy remains universally recognized. However, as health-information seeking has moved online, the privacy of a doctor’s office has been traded in for the silent intrusion of behavioral tracking. This tracking provides a valuable vantage point from which to observe how established cultural norms and technological innovations are at odds.

Online health privacy is an issue that affects the majority of Internet users. According to the Pew Research Center, 72% of adult Internet users in the U.S. go online to learn about medical conditions.<sup>9</sup> Yet only 13% of these begin their search at health-specific sites. In fact, health information may be found on a wide spectrum of sites ranging from newspapers, discussion forums, to research institutions. In order to discover the full range of sites users may visit when seeking health information, I used a search engine to

## » key insights

- **Over 90% of the 80,142 health-related Web pages initiate HTTP requests to third-parties, oftentimes outside the view of the user.**
- **Some 70% of third-party requests transmit information on specific symptoms, treatments, and diseases in the URI string.**
- **Page visitors are at risk of their health interests being publicly identified as well as being blindly discriminated against by marketers.**
- **Extant policy and legal protections are few in number and weak in effect, demonstrating a need for interventions.**

...not - Google Search www.google.com  
flu - Google Search www.google.com  
std testing nyc - Google Search www.google.com  
std information hotline - Google Search www.google.com  
https://www.google.com/webhp?saq=1  
std information - Google Search www.google.com  
stds testing - Google Search www.google.com  
stds list - Google Search www.google.com  
stds - Google Search www.google.com  
std - Google Search www.google.com  
hiv medicine cost - Google Search www.google.com  
hiv medicine - Google Search www.google.com  
hiv research - Google Search www.google.com  
hiv - Google Search www.google.com  
hiv treatment - Google Search www.google.com  
hiv treatment cost - Google Search www.google.com  
hiv symptoms - Google Search www.google.com  
hiv information - Google Search www.google.com  
hiv testing - Google Search www.google.com  
hiv testing nyc - Google Search www.google.com  
chemotherapy side effects - Google Search www.google.com  
chemotherapy drugs - Google Search www.google.com  
chemotherapy hair loss - Google Search www.google.com  
chemotherapy information - Google Search www.google.com  
lung cancer prognosis - Google Search www.google.com  
lung cancer stage 4 - Google Search www.google.com  
lung cancer symptoms - Google Search www.google.com  
lung cancer stages - Google Search www.google.com  
bronchitis curable - Google Search www.google.com  
bronchitis cure - Google Search www.google.com  
bronchitis symptoms in children - Google Search www.google.com  
bronchitis contagious - Google Search www.google.com  
bronchitis home remedies - Google Search www.google.com  
flu - Google Search www.google.com  
flu symptoms - Google Search www.google.com

...ing - Google Search www.google.com  
stds list - Google Search www.google.com  
stds - Google Search www.google.com  
std - Google Search www.google.com  
hiv medicine cost - Google Search www.google.com  
hiv medicine - Google Search www.google.com  
hiv research - Google Search www.google.com  
hiv - Google Search www.google.com  
hiv treatment - Google Search www.google.com  
hiv treatment cost - Google Search www.google.com  
hiv symptoms - Google Search www.google.com  
hiv information - Google Search www.google.com  
hiv testing - Google Search www.google.com  
hiv testing nyc - Google Search www.google.com

...enitis new - Google Search www.google.com  
flu - Google Search www.google.com  
flu symptoms - Google Search www.google.com  
flu symptoms in kids - Google Search www.google.com  
flu symptoms in children - Google Search www.google.com  
https://www.google.com/cancer/...  
http://www.cancer.org/web...  
flu shots - Google Search www.google.com  
flu shots nyc - Google Search www.google.com  
flu shot cost - Google Search www.google.com  
flu shot side effects - Google Search www.google.com  
flu shot safety - Google Search www.google.com  
http://www.cancer.org/cancer/pa...  
flu shot side effects - Google Search www.google.com  
flu shot - Google Search www.google.com  
std testing nyc - Google Search www.google.com  
std information hotline - Google Search www.google.com  
https://www.google.com/webhp?saq=1  
std information - Google Search www.google.com  
stds testing - Google Search www.google.com  
stds list - Google Search www.google.com  
stds - Google Search www.google.com  
std - Google Search www.google.com

identify 80,142 unique health-related Web pages by compiling responses to queries for 1,986 common diseases. This selection of pages represents what users are actually visiting, rather than a handful of specific health portals.

Having identified a population of health-related Web pages, I created a custom software platform to monitor the HTTP requests initiated to third parties. I discovered that 91% of pages make requests to additional parties, potentially putting user privacy at risk. Given that HTTP requests often include the URI of the page currently being viewed (known as the “Referer” [sic]), information about specific symptoms, treatments, and diseases may be transmitted. My analysis shows 70% of URIs contains such sensitive information.

This proliferation of third-party requests makes it possible for corporations to assemble dossiers on the health conditions of unwitting users. In order to identify which corporations are the recipients of this data I have also analyzed the ownership of the most requested third-party domains. This has produced a revealing picture of how personal health information becomes the property of private corporations.

This article begins with a short primer on how third-party HTTP requests

work, reviews previous research in this area, details methodology and findings, and concludes with suggestions for protecting health privacy online.

**Background: Third-Party HTTP Requests**

A real-world example is the best way to understand how the information is leaked to third parties on a typical Web page. When a user searches online for “HIV” one of the top results is for the U.S. Centers for Disease Control and Prevention (CDC) page with the address <http://www.cdc.gov/hiv/>.<sup>a</sup> Clicking on this result initiates what is known as a “first-party” Hypertext Transfer Protocol (HTTP) request to the CDC Web server (Figure 1.1). A portion of such a request is as follows:

```
GET /hiv/
Host: www.cdc.gov
User-Agent: Mozilla/5.0 (Macintosh...
```

This request is sent to the CDC Web server (“Host: www.cdc.gov”) and is an instruction to return (“GET”) the page with the address “/hiv/.” This request also includes “User-Agent” information that tells the server what kind

<sup>a</sup> As of April, 2014

of browser and computer the user is on. In this case, the user employs the Mozilla Firefox browser on a Macintosh computer. Such information is helpful when loading specially optimized pages for smartphones or tablets.

Once this request has been made, the CDC Web server sends the user an HTML file. This file contains the text of the page as well as a set of instructions that tells the Web browser how to download and style additional elements such as images (Figure 1.2). In order to get the CDC logo, the following HTTP request is made:

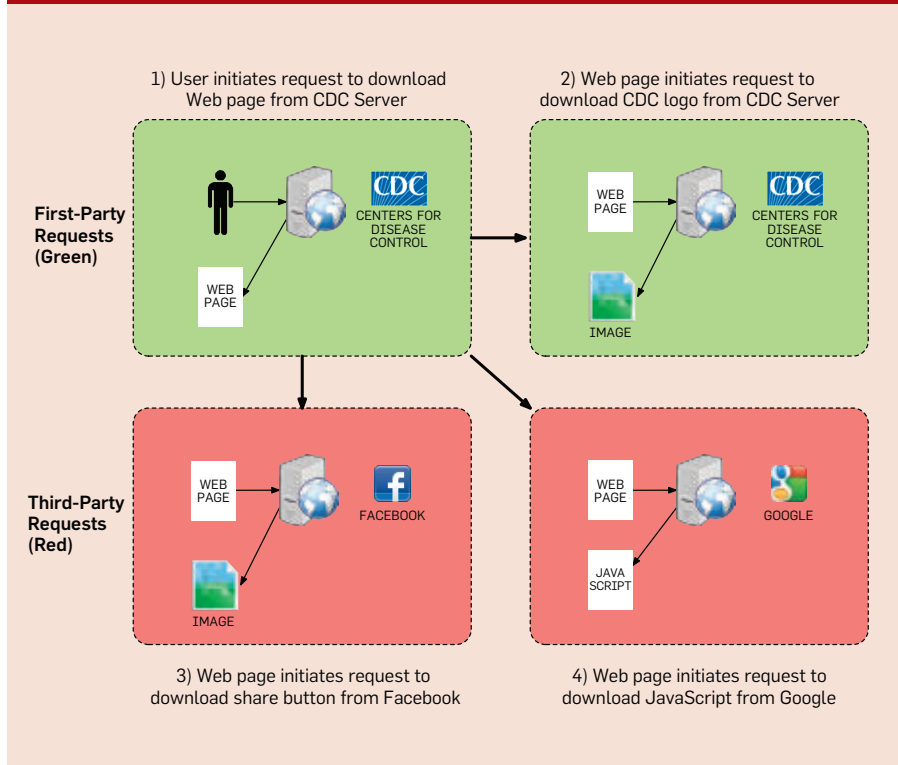
```
GET /TemplatePackage/images/cdcHeaderLogo.gif
Host: www.cdc.gov
User-Agent: Mozilla/5.0 (Macintosh...
Referer: http://www.cdc.gov/hiv/
```

This request introduces a new piece of information called the Referer, which contains the address of the page the user is currently viewing. The CDC Web server may keep records of all HTTP requests in order to determine what pages and content are being requested most often.

Because the “Host” for both requests is identical (www.cdc.gov), the user is only interacting with a single party and such requests are called “first-party requests.” The only two parties who know the user is looking up information about HIV are the user and the CDC. However, the HTML file also contains code that makes requests to outside parties. These types of third-party requests typically download third-party elements such as images and JavaScript. Due to the fact that users are often unaware of such requests, they form the basis of the so-called “Invisible Web.”

On the CDC’s HIV page, third-party requests are made to the servers of Facebook, Pinterest, Twitter, and Google. In the case of the first three companies, the requested elements are all social media buttons, which allow for the sharing of content via the “Recommend,” “Tweet,” or “Pin It” icons (Figure 1.3). It is unlikely that many users would understand the presence of these buttons indicates that their data is sent to these companies. In contrast, the Google elements on the page are entirely invisible and there is no Google logo present. One of

**Figure 1. First- and third-party requests on the CDC Web page for HIV/AIDS.**





these requests is sent to Google's Analytics service (Figure 1.4) to download a file containing JavaScript code:

```
GET /ga.js
Host: www.google-analytics.com
User-Agent: Mozilla/5.0 (Macintosh...
Referer: http://www.cdc.gov/hiv/
```


Again, the Referer field reveals the user is visiting a page about HIV. By pairing information about the User-Agent, Referer, and user's IP address, it is possible for companies like Google and Facebook to identify people who are concerned with HIV.<sup>34</sup> Those visiting this page likely are unaware of this fact, and would not be happy to find out.

### Prior Research


Prior research has demonstrated that while users are uncomfortable with this type of tracking, it is performed in a number of highly sophisticated ways, and it is increasingly widespread.

**Attitudes.** There has long been anxiety about how personal data will be used on the Web. A 1999 study determined that "only 13% of respondents reported they were 'not very' or 'not at all' concerned" about their privacy online.<sup>2</sup> Such anxiety remained in 2003 when 70% of survey respondents reported they were nervous that websites had information about them.<sup>29</sup> A 2009 follow-up study revealed that 67% of respondents agreed with the statement they had "lost all control over how personal information is collected and used by companies."<sup>30</sup> These surveys demonstrate the activities of many businesses run directly counter to public preferences.

As with general concerns with online privacy, there is excellent research exploring attitudes toward health information. In 2012, Hoofnagle et al. determined that only 36% of survey respondents knew that advertisers are allowed to track their visits to health-related websites.<sup>12</sup> An extensive study from the year 2000 found that 85% of Internet users in poor health were concerned that websites would share their data, and only 3% were comfortable with websites sharing their data with other sites, companies, and advertisers.<sup>10</sup> Despite these fears, 44% of respondents felt their information was safe



## The proliferation of third-party requests makes it possible for corporations to assemble dossiers on the health conditions of unwitting users.



with institutions such as the National Institutes of Health (NIH).<sup>10</sup> The CDC example detailed earlier indicates this trust is potentially misplaced.

**Mechanisms.** Once a third-party request is made, a user may be tracked using a number of ever-evolving technical mechanisms. Researchers have been tracing the development of such mechanisms for years, often analyzing the code and behaviors that take place within the Web browser. These are often called "client-side" techniques for they take place on the user's computer. Traditional client-side techniques typically involve storing data on the user's computer in small text files known as cookies—this functions as a sort of digital name tag.<sup>3</sup> Users are getting more adept at evading such practices, therefore newer techniques often employ so-called "browser fingerprinting" to identify users based on characteristics of their computers. This area of research has proven very popular of late with numerous studies investigating fingerprinting techniques.<sup>1,7,13,14,23</sup> In addition, Miller et al. have recently demonstrated sophisticated attacks on HTTPS that are able to reveal "personal details including medical conditions."<sup>20</sup>

Turning attention to the server-side, Yen et al. have recently demonstrated a tracking technique that utilizes a combination of IP address, User-Agent string, and time intervals when HTTP requests were made. This team was able to identify users 80% of the time, which is on par with what is typically accomplished with client-side cookies.<sup>34</sup> Furthermore, identification rates remained essentially static even when removing the final octet of the IP address, which is a common technique by which major advertisers claim to anonymize data. Yen et al.'s findings indicate that while novel techniques may be needed on the client-side, the lowly HTTP request is sufficient for advanced server-side techniques.

**Measurement.** The final area of related research is measurement. Measurement of Web tracking generally entails two steps: selecting a population of pages, and performing automated analysis of how user data is transmitted to third parties. Many studies have relied on popular site lists provided by the Alexa

company,<sup>4,17,18,25</sup> but often utilize their own methodologies for analysis. Krishnamurthy and Wills have conducted many of the most important studies in this area<sup>18</sup> and developed the idea of a privacy footprint<sup>17</sup> based upon the number of nodes a given user is exposed to as they surf the Web. This team has consistently found there are high levels of tracking on the Web, including on sites dealing with sensitive personal information such as health.<sup>17</sup> Other teams have performed comparative analyses between countries<sup>4</sup> as well as explored general trends in tracking mechanisms.<sup>19,25</sup> A common theme among all measurement research is the amount of tracking on the Web is increasing, and shows no signs of abating. The data presented in this article updates and advances extant findings with a focus on how users are tracked when they seek health information online.

### Methodology

In order to quickly and accurately reveal third-party HTTP requests on health-related Web pages, my methodology has four main components: page selection, third-party request detection, request analysis, and corporate ownership analysis.

**Page selection.** A variety of websites such as newspapers, government agencies, and academic institutions provide health information online. Thus, limiting analysis to popular health-centric sites fails to reach many of the sites users actually visit.<sup>16</sup> To wit, the Pew Internet and American Life Project found “77% of online health seekers say they began at a search engine such as Google, Bing, or Yahoo”<sup>9</sup> as opposed to a health portal like WebMD.com. In order to best model the pages a user would visit after receiving a medical diagnosis, I first compiled a list of 1,986 diseases and conditions based on data from the Centers for Disease Control, the Mayo Clinic, and Wikipedia. Next, I used the Bing search API in order to find the top 50 search results for each term.<sup>b</sup> Once duplicates and binary files (pdf, doc, xls) were filtered out, a set of 80,142 unique Web pages remained. A major contribution of this study to prior work is the fact that my analysis is focused on the pages that users seeking medical information

<sup>b</sup> Search results were localized to U.S./English.



**Prior research has demonstrated that while users are uncomfortable with this type of tracking, it is performed in a number of highly sophisticated ways, and it is increasingly widespread.**



are most likely to visit, irrespective of if the site is health-centric.

**Third-party request detection.** To detect third-party HTTP requests, my methodology employs a “headless” Web browser named PhantomJS.<sup>24</sup> PhantomJS requires no GUI, has very low resource utilization, and is therefore well suited for large-scale analyses. Due to the fact it is built on WebKit, PhantomJS’s underlying rendering engine is capable of executing JavaScript, setting and storing cookies, and producing screen captures. Most important for this project, PhantomJS allows for the direct monitoring of HTTP requests without the need to resort to browser hacks or network proxies.

It should be noted that the most recent versions of PhantomJS (1.5+) do not support the Adobe Flash browser plug-in. To address this potential limitation, I conducted testing with an older version of PhantomJS (1.4) and Flash. The inclusion of Flash led to much higher resource utilization, instability, and introduced a large performance penalty. While this method successfully analyzed Flash requests, I determined that Flash elements were comparatively rare and had negligible effect on the top-level trends presented below. Therefore, I made the decision to forgo analysis of Flash requests in favor of greater software reliability by using the most recent version of PhantomJS (1.9).

In order to fully leverage the power of PhantomJS, I created a custom software platform named WebXray that drives PhantomJS, collects and analyzes the output in Python, and stores results in MySQL. The workflow begins with a predefined list of Web page addresses that are ingested by a Python script. PhantomJS then loads the given Web address, waits 30 seconds to allow for all redirects and content loading to complete, and sends back JSON-formatted output to Python for analysis. This technique represents an improvement over methods such as searching for known advertising elements detected by popular programs such as Ghostery or AdBlock.<sup>4</sup> As of March 2014, Ghostery reports the WebMD Web page for “HIV/AIDS” contains four trackers. In contrast, WebXray detects the same page initiating requests to thirteen distinct third-party domains.

This is due to the fact that Ghostery and Adblock rely on curated blacklists of known trackers, rather than reporting all requests.

**Request analysis.** The primary goal of WebXray is to identify third-party requests by comparing the domain of the Web page being visited to the domains of requests being made. For example, the address “http://example.com” and the request “http://images.example.com/logo.png” both share the domain “example.com,” thus constituting a first-party request.

Alternately, a request from the same page to “http://www.googleanalytics.com/ga.js,” which has the domain “google-analytics.com,” is recognized as a third-party request. The same technique for HTTP requests is also applied toward evaluating the presence of third-party cookies. The method is not flawless, as a given site may actually use many domains, or a subdomain may point to an outside party. However, when evaluating these types of requests in aggregate, such problems constitute the statistical noise that is present in any large dataset.

Finally, in order to evaluate larger trends in tracking mechanisms, third-party requests are dissected to extract arguments (for example, “?SITEID=123”) and file extensions such as .js (JavaScript), .jpg (image), and .css (cascading style sheet).

Removing arguments also allows for a more robust analysis of which elements are the most prevalent, as argument strings often have specific site identifiers, making them appear unique when they are not.

**Corporate ownership.** A specific focus of this investigation is to determine which corporate bodies are receiving information from health-related Web pages. While it is possible to programmatically detect requests to third-party domains, it is not always clear who belongs to the requested domains. By examining domain registration records, I have been able to pair seemingly obscure domain names (for example, “2mdn.net,” “fbcdn.net”) with their corporate owners (for example, Google, Facebook). This process has allowed me to follow the data trail back to the corporations that are the recipients of user data. To date, the literature has given much more attention to technical

mechanisms, and much less to the underlying corporate dynamics. This fresh analytical focus highlights the power of a handful of corporate giants.

**Limitations.** While this methodology is resource efficient and performs well at large scale, it comes with several potential limitations, many of which would produce an under-count of the number of third-party requests. First, given the rapid rate by which pages are accessed, it is possible that rate-limiting mechanisms on servers may be triggered (that is, the requests generated by my IP would be identifiable as automated), and my IP address could be blacklisted, resulting in an under-count. Second, due to the fact I use PhantomJS without browser plugins such as Flash, Java, and Silverlight, some tracking mechanisms may not load or execute properly, resulting in an under-count. Third, many tracking mechanisms are designed to be difficult to detect by a user, and an under-count could result from a failure to detect particularly clever tracking mechanisms. Therefore, the findings presented here constitute a lower bound of the amount of requests being made.

## Findings

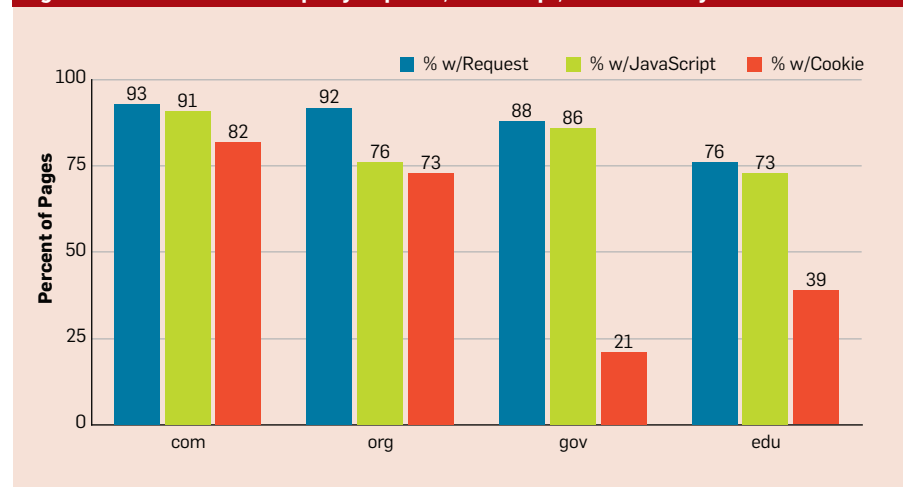
In April 2014, I scanned 80,142 Web pages that were collected from search results for 1,986 common diseases with the intent of detecting the extent and the ways in which the sensitive health data of users was being leaked.

**General trends.** I have broken up my top-level findings into five general categories based on information gleaned from the TLDs used. They are: all pages,

commercial pages (.com), non-profit pages (.org), government pages (.gov), and education-related pages (.edu). This information is illustrated in Figure 2. Of all pages examined, 91% initiate some form of third-party HTTP request, 86% download and execute third-party JavaScript, and 71% utilize cookies. Unsurprisingly, commercial pages were above the global mean and had the most third-party requests (93%), JavaScript (91%), and cookies (82%). Education pages had the least third-party HTTP requests (76%) and JavaScript (73%), with a full quarter of the pages free of third-party requests. Government pages stood out for relatively low prevalence of third-party cookies, with only 21% of pages storing user data in this way. Figure 2 details these findings.

**Mechanisms.** Given that 91% of pages make third-party HTTP requests, it is helpful to know what exactly is being requested. Many third-party requests lack extensions, and when viewed in a browser display only blank pages that generate HTTP requests and may also manipulate browser caches. Such requests accounted for 47% of the top 100 requests and may point toward emerging trends in the ongoing contest between user preferences and tracking techniques. The second most popular type of requested elements were JavaScript files (33%). These files are able to execute arbitrary code in a user’s browser and may be used to perform fingerprinting techniques, manipulate caches and HTML5 storage, as well as initiate additional requests. The third most popular type of content is the tried-and-true image file, which ac-

Figure 2. Prevalence of third-party requests, JavaScript, and cookies by TLD.



counts for 8% of the top requested elements. Table 1 presents additional detail into the file extensions found.

Given that tracking occurs on the so-called Invisible Web, it initially appears odd that so many mechanisms are images. However, when investigating the images themselves, it is clear they provide little indication as to whom they belong to, and thus users are kept in the dark as to their purpose or presence. An examination of the top 100 requested images determined that only 24% contained information that would alert the user they had initiated contact with a third party. Many images were only a single pixel in size, and are often referred to as tracking pixels as their only purpose is to initiate HTTP requests. The most popular image, found on 45% of pages, was a single tracking pixel with the name `utm.gif`, which is part of the Google Analytics service. The second most popular image is the clearly identifiable Facebook “Like” button that was found on 16% of pages. It is unclear how many users elect to “Like” an illness, but Facebook is able to record page visits regardless if a user clicks the “Like” button, or if they

even have a Facebook account in the first place. Google and Facebook are not alone, however, there are a number of companies tracking users online.

**Corporate ownership.** While security and privacy research has often focused on how user privacy is violated, insufficient attention has been given to who is collecting user information. The simple answer is that a variety of advertising companies have developed a massive data collection infrastructure that is designed to avoid detection, as well as ignore, counteract, or evade user attempts at limiting collection. Despite the wide range of entities collecting user data online, a handful of privately held U.S. advertising firms dominate the landscape of the Invisible Web.

Some 78% of pages analyzed included elements that were owned by Google. Such elements represent a number of hosted services and use a variety of domain names: they range from traffic analytics (`google-analytics.com`), advertisements (`doubleclick.net`), hosted JavaScript (`googleapis.com`), to videos (`youtube.com`). Regardless of the type of services provided, in some way all of these HTTP requests funnel information back to Google. This means a single company has the ability to record the Web activity of a huge number of individuals seeking sensitive health-related information without their knowledge or consent.

While Google is the elephant in the room, they are far from alone. Table 2 details the top 10 firms found as part

of this analysis along with the rankings of two data brokers. In second place is comScore who are found on 38% of pages, followed by Facebook with 31%. It is striking that these two companies combined still have less reach than Google.

Additionally, companies were categorized according to their type of revenue model. Some 80% of the top 10 companies are advertisers. The only exceptions to this rule are Adobe and Amazon. Adobe offers a mix of software and services, including traffic analytics. Amazon is in the business of both consumer-retail sales as well as Web hosting with the Amazon Web Services (AWS) division. At present it is unclear if AWS data is integrated into Amazon product recommendations or deals, but the possibility exists.

While advertisers dominate online tracking, I was also able to detect two major data brokers: Experian (5% of pages), and Acxiom (3% of pages). The main business model of data brokers is to collect information about individuals and households in order to sell it to financial institutions, employers, marketers, and other entities with such interest. Credit scores provided by Experian help determine if a given individual qualifies for a loan, and if so, at what interest rate. Given that a 2007 study revealed that “62.1% of all bankruptcies ... were medical,”<sup>11</sup> it is possible that some data brokers not only know when a given person suffered a medical-related bankruptcy, but perhaps even when they first searched for information on the ailment that caused their financial troubles.

**Health information leakage.** The HTTP 1.1 protocol specification warns the source of a link [URI] might be private information or might reveal an otherwise private information source and advises that “[c]lients SHOULD NOT include a Referer header field in a (non-secure) HTTP request if the referring page was transferred with a secure protocol.”<sup>8</sup> In simpler terms, Web pages that include third-party elements, but do not use secure HTTP requests, risk leaking sensitive data via the Referer field. Of the pages analyzed, only 3.24% used secure HTTP, the rest used non-encrypted HTTP connections and thereby potentially transmitted sensitive information to third parties. Unsurprisingly, a significant amount of

**Table 1. Types of file extensions.**

Type	%
No Extension	47
JavaScript	33
Image	8
Dynamic Page	4
Other	8

**Table 2. Corporate ownership and risk assessment (N=80,142).**

Rank	% Pages	Company	Revenue	Identification	Blind Discrimination
1	78	Google	Advertising	X	X
2	38	comScore	Advertising	—	X
3	31	Facebook	Advertising	X	X
4	22	AppNexus	Advertising	—	X
5	18	Add This	Advertising	—	X
6	18	Twitter	Advertising	—	X
7	16	Quantcast	Advertising	—	X
8	16	Amazon	Retail and Hosting	—	X
9	11	Adobe	Software and Services	—	X
10	11	Yahoo!	Advertising	—	X
...	—	—	—	—	—
31	5	Experian	Data Broker	X	—
...	—	—	—	—	—
47	3	Acxiom	Data Broker	X	—

sensitive information was included in URI strings.

Based on a random sample of 500 URIs taken from the population of pages analyzed ( $N=80,142$ ), 70% contained information related to a specific symptom, treatment, or disease. An example of an URI containing specific symptom information is:

[http://www.nhs.uk/conditions/breast-lump/...](http://www.nhs.uk/conditions/breast-lump/)

a URI containing no such information is:


<http://www.ncbi.nlm.nih.gov/pubmed/21722252>

Given the former type of URI was by far the most prevalent, it may be seen that third parties are being sent a large volume of sensitive URI strings that may be analyzed for the presence of specific diseases, symptoms, and treatments. This type of leakage is a clear risk for those who wish to keep this information out of the hands of third parties who may use it for unknown ends.


## Discussion

Defining privacy harms is a perennially difficult proposition. Health information, however, presents two main privacy risks that are interrelated. The first is personal identification, where an individual's name is publicly associated with their medical history. The second is blind discrimination, where an individual's name is not necessarily revealed, but they may be treated differently based on perceived medical conditions.

**Personal identification.** While most people would probably consider details of their health lives to be of little interest or value to others, such details form the basis of a lucrative industry. In 2013, the U.S. Senate Committee on Commerce, Science and Transportation released a highly critical review of the current state of the so-called data broker industry. Data brokers collect, package, and sell information about specific individuals and households with virtually no oversight. This data includes demographic information (ages, names, and addresses), financial records, social media activity, as well as information on those who may be suffering from "particular ailments, in-



**While security and privacy research has often focused on how user privacy is violated, insufficient attention has been given to who is collecting user information.**



cluding Attention Deficit Hyperactivity Disorder, anxiety, depression ... among others."<sup>26</sup> One company, Medbase200, was reported as using proprietary models to generate and sell lists with classifications such as rape victims, domestic abuse victims, and HIV/AIDS patients.<sup>6</sup>

It should also be noted that such models are not always accurate. For example, individuals looking for information on the condition of a loved one may be falsely tagged as having the condition themselves. This expands the scope of risk beyond the patient to include family and friends. In other cases, an individual may be searching for health information out of general interest and end up on a data broker's list of sufferers or patients. Common clerical and software errors may also tag individuals with conditions they do not have. The high potential for such errors also highlights the need for privacy protections.

Furthermore, criminals may abuse poorly protected health information. The retailer Target has used datamining techniques to analyze customers' purchase history in order to predict which women may be pregnant in order to offer them special discounts on infant-related products.<sup>5</sup> Even if shoppers and surfers are comfortable with companies collecting this data, that is no guarantee it is safe from thieves. In 2013, 40 million credit and debit card numbers were stolen from Target.<sup>15</sup> While a stolen credit card may be reissued, if Target's health-related data were leaked online, it could have a devastating impact on millions of people. Merely storing personally identifiable information on health conditions raises the potential for loss, theft, and abuse.

**Blind discrimination.** Advertisers regularly promise their methods are wholly anonymous and therefore benign, yet identification is not always required for discriminatory behavior to occur. In 2013, Latanya Sweeney investigated the placement of online advertisements that implied a given name was associated with a criminal record.<sup>27</sup> She found the presence of such ads were not the result of particular names being those of criminals, but appeared based on the racial associations of the name, with African-American names more often resulting in an implication of criminal record. In this way, extant societal injustices may be replicated

through advertising mechanisms online. Discrimination against the ill may also be replicated through the collection and use of browsing behavior.

Data-mining techniques often rely on an eclectic approach to data analysis. In the same way a stew is the result of many varied ingredients being mixed in the same pot, behavioral advertising is the result of many types of browsing behavior being mixed together in order to detect trends. As with ingredients in a stew, no single piece of data has an overly deterministic impact on the outcome, but each has some impact. Adding a visit to a weather site in the data stew will have an outcome on the offers a user receives, but not in a particularly nefarious way. However, once health information is added to the mix, it becomes inevitable it will have some impact on the outcome. As medical expenses leave many with less to spend on luxuries, these users may be segregated into data silos<sup>28</sup> of undesirables who are then excluded from favorable offers and prices. This forms a subtle, but real, form of discrimination against those perceived to be ill.

**Risk assessment.** Having collected data on how much tracking is taking place, how it occurs, and who is doing it, it is necessary to explicate how this constitutes a risk to users. As noted earlier, there are two main types of harm: identification and blind discrimination. Table 2 shows a breakdown of how data collection by 12 companies (top 10 and data brokers) impacts the two types of risk. The two data brokers most obviously entail a personal identification risk as their entire business model is devoted to selling personal information. It is unlikely they are selling raw Web tracking data directly, but it may be used as part of aggregate measures that are sold.

Despite the fact that Google does not sell user data, they do possess enough anonymous data to identify many users by name. Google offers a number of services that collect detailed personal information such as a user's personal email (Gmail), work email (Apps for Business), and physical location (Google Maps). For those who use Google's social media offering, Google+, a real name is forcefully encouraged. By combining the many types of information held by

Google services, it would be fairly trivial for the company to match real identities to anonymous Web browsing data. Likewise, Facebook requires the use of real names for users, and as noted before, collects data on 31% of pages; therefore, Facebook's collection of browsing data may also result in personal identification. In contrast, Twitter allows for pseudonyms as well as opting-out of tracking occurring off-site.

The potential for blind discrimination is most pronounced among advertisers. As noted here, online advertisers use complex data models that combine many pieces of unrelated information to draw conclusions about anonymous individuals. Any advertiser collecting and processing health-browsing data will use it in some way unless it is filtered and disposed of.

### Policy Implications

The privacy issues raised by this research are of a technical nature and invite technical solutions. These solutions often come in the form of add-on software users may install in their Web browsers. Such browser add-ons have proven effective at blocking certain types of behavioral tracking.<sup>19,25</sup> However, this type of solution places a burden on users and has not been broadly effective. As measurement research has shown, tracking has only increased over the past decade despite technical efforts to rein it in.

Purely technical solutions are problematic, as they require a relatively high level of knowledge and technical expertise on the part of the user. The user must first understand the complex nature of information flows online in order to seek out technical remedies. Next, the user must be proficient enough to install and configure the appropriate browser additions. This may seem trivial for the well educated, but many who use the Internet have little education or training in computing. Despite this, these users deserve to have their health privacy protected.

Furthermore, add-ons are often unavailable on the default browsers of smartphones and tablets, making it difficult for even the highly skilled to protect their privacy. A final reason that browser add-ons provide insufficient remedy is the fact that advertisers devote significant resources to

overcoming such barriers and will always find creative ways to bypass user intent. Thus, on one hand we have users who are poorly equipped to defend themselves with available technical measures, and on the other, highly motivated and well-funded corporations with cutting-edge technologies.

In order to effectively tackle the issue of tracking on health-related pages, attention toward the underlying social dynamics is needed. Government and corporate policies formalize these dynamics. By addressing policy issues directly, rather than combating obscure tracking techniques, we may produce durable solutions that outlast today's technology cycle. Unfortunately, extant policies are few in number and weak in effect.

**Extant policies and protections.** Health information is one of the few types of personal information that has been granted special protections. The Health Insurance Portability and Accountability Act (HIPAA)<sup>31</sup> is a U.S. law that stipulates how medical information may be handled, stored, and accessed. HIPAA is not meant to police business practices in general; rather it is tailored to those providing health-specific services such as doctors, hospitals, and insurance claims processors. Yet, even within this realm, HIPAA provides incomplete protections. Contrary to popular perceptions, HIPAA permits the disclosure of patient information between health providers and insurance claims processors without patient notification or consent. HIPAA generally does not allow patients to restrict the flow of their sensitive data; therefore, extending HIPAA in the online domain does not present an effective approach to privacy protection.

Nevertheless, the U.S. Federal Trade Commission (FTC) has established a Health Breach Notification Rule that requires entities holding personally identifiable health records to notify users if such records have been stolen.<sup>32</sup> However, merely providing health information (rather than storing doctor's notes or prescription records) does not place a business under the jurisdiction of HIPAA or associated rules. Many businesses that handle health information are subject to virtually no oversight and the main source of policy regarding the use of health informa-

tion online comes in the form of self-regulation by the parties that stand to benefit the most from capturing user data: online advertisers.

However, self-regulation has proven wholly insufficient. No lesser authority than the FTC determined that “industry efforts to address privacy through self-regulation have been too slow, and up to now have failed to provide adequate and meaningful protection.”<sup>33</sup> When self-regulations are present, there are no serious sanctions for violating the rules that advertisers draw up among themselves. Nevertheless, the Network Advertising Initiative (NAI) has produced a Code of Conduct that requires opt-in consent for advertisers to use precise information about health conditions such as cancer and mental health.<sup>22</sup> Yet the same policy also states that “member companies may seek to target users on the basis of such general health categories as headaches.”<sup>22</sup> Given the range of ailments between cancer and a headache is incredibly broad, this directive provides virtually no oversight. Likewise, the Digital Advertising Alliance (DAA) provides rules that also appear to protect health information, but legal scholars have determined that “an Internet user searching for information about or discussing a specific medical condition may still be tracked under the DAA’s principles.”<sup>12</sup>

**Potential interventions.** Although this problem is complex, it is not intractable and there are several ways health privacy risks may be mitigated. First, there is no reason for non-profits, educational institutions, or government-operated sites to be leaking sensitive user information to commercial parties. While advertising revenue keeps commercial sites running, non-profits gain support from donors and grants. Fixing this situation could be as simple as an internal policy directive on a per-institution basis, or as expansive as adopting language that would deny funding to institutions that leak user data.


As for commercial-oriented sites, it is true they rely on ad-tracking revenue. However, regulatory and legislative bodies have the authority to draft and implement policies that would require a mandatory limitation on how long information from health-related websites could be retained and how it could

be used. Such policy initiatives could have significant impact, and would reflect the preferences of the public.

Finally, talented engineers may devote a portion of the time they spend analyzing data to developing intelligent filters to keep sensitive data quarantined. The spark of change could be the result of a single engineer’s 20% time project. If the mad rush to ingest ever more data is tempered with a disciplined approach to filtering out potentially sensitive data, businesses and users may both benefit equally.

## Conclusion

Proving privacy harms is always a difficult task. However, this study has demonstrated that data on health information seeking is being collected by an array of entities that are not subject to regulation or oversight. Health information may be inadvertently misused by some companies, sold by others, or even stolen by criminals. By recognizing that health information deserves to be treated with special care, we may mitigate what harm may already be occurring and proactively avoid future problems.

**Acknowledgments.** The author thanks the anonymous reviewers for wise revisions. Thanks to A. Blanford, M. Delli Carpini, S. González-Bailón, J. Goodwin, B. Hoffman, B. Kroeger, D. Liebermann, N. Maruyama, T. Patel, V. Pickard, J. Poinsett, J. Rosen and J. Smith for their invaluable feedback. 

## References

1. Acar, G. et al. Fpdetective: Dusting the Web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. ACM, 1129–1140.
2. Ackerman, M.S., Cranor, L.F. and Reagle, J. Privacy in e-commerce: examining user scenarios and privacy preferences. In *Proceedings of the 1<sup>st</sup> ACM Conference on Electronic Commerce*. ACM, 1999, 1–8.
3. Ayenson, M., Wambach, D., Soltani, A., Good, N. and Hoofnagle, C. Flash cookies and privacy ii: Now with HTML5 and etag respawning. Available at SSRN 1898390, 2011.
4. Castellucia, C., Grumbach, S., Olejnik, L. et al. Data harvesting 2.0: From the visible to the Invisible Web. In *Proceedings of the 12<sup>th</sup> Workshop on the Economics of Information Security*, 2013.
5. Duhigg, C. How companies learn your secrets. *New York Times*, (2012), 2012.
6. Dwoskin, E.D.E. Data broker removes rape-victims list after journal inquiry. *Wall Street Journal*, 2013.
7. Eckersley, P. How unique is your Web browser? *Privacy Enhancing Technologies*. Springer, 2010, 1–18.
8. Fielding, R. et al. Hypertext transfer protocol (1999), <http://1.1>.
9. Fox, S. and Duggan, M. Health online 2013. Pew Internet and American Life Project.
10. Grimes-Gruczka, T., Gratz, C. and Dialogue, C. Ethics: Survey of Consumer Attitudes about Health Web Sites. California HealthCare Foundation, 2000.
11. Himmelstein, D.U., Thorne, D., Warren, E. and Woolhandler, S. Medical bankruptcy in the United

- States, 2007: Results of a national study. *Amer. J. Med.* 122, 8 (2009), 741–746.
12. Hoofnagle, C., Urban, J. and Li, S. Privacy and modern advertising: Most us Internet users want do not track to stop collection of data about their online activities. In *Proceedings of the Amsterdam Privacy Conference*, 2012.
13. Jackson, C., Bortz, A., Boneh, D. and Mitchell, J.C. Protecting browser state from Web privacy attacks. In *Proceedings of the 15<sup>th</sup> International Conference on World Wide Web*. ACM, 2006, 737–744.
14. Jang, D., Jhala, R., Lerner, S. and Shacham, H. An empirical study of privacy-violating information flows in JavaScript Web applications. In *Proceedings of the 17<sup>th</sup> ACM conference on Computer and Communications Security*. ACM, 2010, 270–283.
15. Krebs, B. Sources: Target investigating data breach (2013); <http://krebsonsecurity.com/2013/12/sources-target-investigating-data-breach/>.
16. Krishnamurthy, B., Naryshkin, K. and Wills, C. Privacy leakage vs. protection measures: The growing disconnect. In *Proceedings of the Web 2.0 Security and Privacy Workshop*, 2011.
17. Krishnamurthy, B. and Wills, C. Privacy diffusion on the Web: A longitudinal perspective. In *Proceedings of the 18<sup>th</sup> International Conference on World Wide Web*. ACM, 2009, 541–550.
18. Krishnamurthy, B. and Wills, C.E. Generating a privacy footprint on the Internet. In *Proceedings of the 6<sup>th</sup> ACM SIGCOMM Conference on Internet Measurement*. ACM, 2006, 65–70.
19. Mayer, J.R. and Mitchell, J.C. Third-party Web tracking: Policy and technology. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*. IEEE, 413–427.
20. Miller, B., Huang, L., Joseph, A. and Tygar, J. I know why you went to the clinic: Risks and realization of https traffic analysis. arXiv preprint arXiv:1403.0297, 2014.
21. National Institutes of Health, History of Medicine Division. Greek medicine (2002); [http://www.nlm.nih.gov/hmd/greek/greek\\_oath.html](http://www.nlm.nih.gov/hmd/greek/greek_oath.html)
22. Network Advertising Initiative. NAI code of conduct, 2011.
23. Nikiforakis, N. et al. Cookieless monster: Exploring the ecosystem of Web-based device fingerprinting. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2013.
24. PhantomJS. PhantomJS is a headless Webkit scriptable [browser] with a JavaScript API, 2013; <http://phantomjs.org/>.
25. Roesner, F., Kohno, T., and Wetherall, D. Detecting and defending against third-party tracking on the Web. In *Proceedings of the 9<sup>th</sup> USENIX Conference on Networked Systems Design and Implementation*. USENIX Association, 2012, 12.
26. Staff of Chairman Rockefeller. A review of the data broker industry: Collection, use, and sale of consumer data for marketing purposes. U.S. Senate, 2013.
27. Sweeney, L. Discrimination in online ad delivery. *Commun. ACM* 56, 5 (May 2013), 44–54.
28. Turow, J. *The Daily You: How the New Advertising Industry is Defining Your Identity and Your Worth*. Yale University Press, 2012.
29. Turow, J. and Center, A.P.P. Americans & online privacy: The system is broken. Annenberg Public Policy Center, University of Pennsylvania, 2003.
30. Turow, J. King, J., Hoofnagle, C.J., Bleakley, A. and Hennessy, M. Americans reject tailored advertising and three activities that enable it. Available at SSRN 1478214, 2009.
31. United States. Health Insurance Portability and Accountability Act of 1996. Public Law, 1996, 104–191.
32. U.S. Federal Trade Commission. Complying with the FTC’s health breach notification rule, 2010; <http://www.business.ftc.gov/documents/bus56-complying-ftcs-health-breach-notification-rule/>
33. U.S. Federal Trade Commission. Protecting consumer privacy in an era of rapid change preliminary staff report, 2010; [http://www.ftc.gov/sites/default/\\_les/documents/reports/federal-trade-commission-bureau-consumer-protection-preliminary-ftc-sta-report-protecting-consumer/101201privacyreport.pdf](http://www.ftc.gov/sites/default/_les/documents/reports/federal-trade-commission-bureau-consumer-protection-preliminary-ftc-sta-report-protecting-consumer/101201privacyreport.pdf)
34. Yen, T.-F., Xie, Y., Yu, F., Yu, R.P. and Abadi, M. Host fingerprinting and tracking on the Web: Privacy and security implications. In *Proceedings of NDSS*, 2012.

**Timothy Libert** (tlibert@asc.upenn.edu) is a doctoral student in the Annenberg School for Communication at the University of Pennsylvania, Philadelphia, PA.

Copyright held by author.  
Publication rights licensed to ACM. \$15.00.



# Distinguished Speakers Program

*talks by and with technology leaders and innovators*

*Chapters • Colleges and Universities • Corporations • Agencies • Event Planners*

## A great speaker can make the difference between a good event and a WOW event!

The Association for Computing Machinery (ACM), the world's largest educational and scientific computing society, now provides colleges and universities, corporations, event and conference planners, and agencies – in addition to ACM local Chapters – with direct access to top technology leaders and innovators from nearly every sector of the computing industry.

Book the speaker for your next event through the ACM Distinguished Speakers Program (DSP) and deliver compelling and insightful content to your audience. **ACM will cover the cost of transportation for the speaker to travel to your event.** Our program features renowned thought leaders in academia, industry and government speaking about the most important topics in the computing and IT world today. Our booking process is simple and convenient. Please visit us at: [www.dsp.acm.org](http://www.dsp.acm.org). If you have questions, please send them to [acmdsp@acm.org](mailto:acmdsp@acm.org).

### *The ACM Distinguished Speakers Program is an excellent solution for:*

**Corporations** Educate your technical staff, ramp up the knowledge of your team, and give your employees the opportunity to have their questions answered by experts in their field.

**Colleges and Universities** Expand the knowledge base of your students with exciting lectures and the chance to engage with a computing professional in their desired field of expertise.

**Event and Conference Planners** Use the ACM DSP to help find compelling speakers for your next conference and reduce your costs in the process.

**ACM Local Chapters** Boost attendance at your meetings with live talks by DSP speakers and keep your chapter members informed of the latest industry findings.

### *Captivating Speakers from Exceptional Companies, Colleges and Universities*

DSP speakers represent a broad range of companies, colleges and universities, including:

AMD	Imperial College London	Nanyang Technological University	UCLA
Carnegie Mellon University	INTEL	Raytheon BBN Technologies	University of British Columbia
Google	Lawrence Berkeley Nat'l Laboratory	Stanford University	University of Cambridge
IBM	Microsoft	Tsinghua University	University of Texas at Austin

### *Topics for Every Interest*

Over 500 lectures are available from more than 120 different speakers with topics covering:

Software	Web Topics	Career-Related Topics	Computer Graphics, Visualization and Interactive Techniques
Cloud and Delivery Methods	Computer Systems	Science and Computing	High Performance Computing
Emerging Technologies	Open Source	Artificial Intelligence	Human Computer Interaction
Engineering	Game Development	Mobile Computing	

### *Exceptional Quality Is Our Standard*

The same ACM you know from our world-class Digital Library, magazines and journals is now putting the affordable and flexible Distinguished Speaker Program within reach of the computing community.

Microsoft  
**Research**  
The DSP is sponsored  
in part by Microsoft Europe



**Association for  
Computing Machinery**

*Advancing Computing as a Science & Profession*



---

P. 80

**Technical  
Perspective**  
**Image Processing  
Goes Back to Basics**

By Edward Adelson

P. 81

**Local Laplacian Filters:  
Edge-Aware Image Processing  
with a Laplacian Pyramid**

By Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz



Watch the authors discuss  
this work in this exclusive  
*Communications* video.

# Technical Perspective

## Image Processing Goes Back to Basics

By Edward Adelson

IN RECENT YEARS, the image sensors in digital cameras have improved in many ways. The increases in spatial resolution are well known. Equally important, but less obvious, are improvements in noise level and dynamic range. At this point digital cameras have gotten so good it is challenging to display the full richness of their image data. A low noise imager can capture subtly varying detail that can only be seen by turning up the display contrast unnaturally high. A high dynamic range (HDR) imager presents the opposite problem: its data cannot be displayed without making the contrast unnaturally low. To convey visual information to a human observer, it is often necessary to present an image that is not physically correct, but which reveals all the visually important variations in color and intensity. A discipline known as *computational photography* has emerged at the intersection of photography, computer vision, and computer graphics, and the twin problems of detail enhancement and HDR range compression (also called tone mapping) have become recognized as important topics.

Given an individual image patch, it is not difficult to find display parameters that will effectively convey the local visual information. The problem is this patch must coexist with all the other image patches around it, and these must join into a single, globally coherent image. Many techniques have been proposed to find an image that simultaneously displays everything clearly, while still looking like a natural image. In struggling to bring about a global compromise between all the local constraints, these techniques tend to introduce visually disturbing artifacts, such as halos around strong edges, or distortions of apparent contrast, sharpness, and position of local features.

Performance has improved through the use of increasingly sophisticated

image processing techniques, which can manipulate information smoothly across multiple spatial scales, while preserving the integrity of sharp edges. Recent progress in “edge-aware” processing builds on a foundation of work in such topics as anisotropic diffusion, regularization, and sparse image coding. New classes of edge-aware filters have been devised, utilizing ideas from robust estimation. Novel forms of wavelet decomposition have been introduced, specifically to deal with the challenges of processing sharp edges within a multiscale representation. However, none of the methods has proven entirely satisfactory, and some of them are quite complex.


In the following paper, Paris et al. made a surprising move. They chose to build a system on the Laplacian pyramid, which is a very simple multiscale representation that predates wavelets. It lacks an impressive mathematical pedigree, but is still widely used because of its simplicity and reliability; it serves as a basic building block for many image-processing schemes. At the same time, the Laplacian pyramid seems ill suited to any tasks involving specialized processing near edges. Its basic functions are smooth, overlapping, and non-oriented, whereas edges are sharply localized and oriented.

The authors also eschew a wide range of modern techniques. Indeed, the most striking thing about the paper is what is missing: There are no statistical image models, no machine learning, no PDEs, no fancy wavelets, and no objective functions. Instead, the authors return to an old-fashioned style rarely seen today: carefully considering a problem at the level of pixels and patches, and specifying the requirements in the most direct possible way. It should be noted that these authors are fully capable of developing elaborate machinery when

they need it, but they have chosen to avoid it here. They want to rethink the problem from the ground up, setting out basic principles about the behavior they desire with edges, textures, and smooth regions.

Their new direction is quite unexpected. To make an analogy, it is almost as if some experts in 3D manufacturing decided to abandon their CAD systems and 3D printers in order to sculpt marble with a hammer and chisel. Sometimes the fancy tools get in the way, and the best thing is to get back in direct contact with the material.

The results in this case are stunning. The authors are able to achieve extreme levels of detail enhancement and HDR range compression. There are almost no visible artifacts. It is difficult to believe anyone can do much better, and in that sense one could say the problems have been solved.

So, is this paper the last word? No, because beautiful pictures are not enough. It is still important to situate the work intellectually within the greater worlds of image processing and computational photography. How do these techniques relate to the many other approaches to detail enhancement and HDR range compression? How can the insights from this paper be integrated into methods that are couched in other languages, such as wavelets or image statistics? More generally, what does this paper teach us about the underlying problems of edge-aware image processing? There is already progress on these questions, as noted in the revised research that Paris et al. present here. We can expect more insights to follow, as people digest the results of this refreshing original paper. 

Edward Adelson ([adelson@csail.mit.edu](mailto:adelson@csail.mit.edu)) is the John and Dorothy Wilson Professor of Vision Science in the Department of Brain and Cognitive Sciences at MIT, Cambridge, MA.

Copyright held by author.

# Local Laplacian Filters: Edge-Aware Image Processing with a Laplacian Pyramid

By Sylvain Paris, Samuel W. Hasinoff, and Jan Kautz

## Abstract

The Laplacian pyramid is ubiquitous for decomposing images into multiple scales and is widely used for image analysis. However, because it is constructed with spatially invariant Gaussian kernels, the Laplacian pyramid is widely believed to be ill-suited for representing edges, as well as for edge-aware operations such as edge-preserving smoothing and tone mapping. To tackle these tasks, a wealth of alternative techniques and representations have been proposed, for example, anisotropic diffusion, neighborhood filtering, and specialized wavelet bases. While these methods have demonstrated successful results, they come at the price of additional complexity, often accompanied by higher computational cost or the need to postprocess the generated results. In this paper, we show state-of-the-art edge-aware processing using standard Laplacian pyramids. We characterize edges with a simple threshold on pixel values that allow us to differentiate large-scale edges from small-scale details. Building upon this result, we propose a set of image filters to achieve edge-preserving smoothing, detail enhancement, tone mapping, and inverse tone mapping. The advantage of our approach is its simplicity and flexibility, relying only on simple point-wise nonlinearities and small Gaussian convolutions; no optimization or postprocessing is required. As we demonstrate, our method produces consistently high-quality results, without degrading edges or introducing halos.

## 1. INTRODUCTION

Laplacian pyramids have been used to analyze images at multiple scales for a broad range of applications such as compression,<sup>6</sup> texture synthesis,<sup>18</sup> and harmonization.<sup>32</sup> However, these pyramids are commonly regarded as a poor choice for applications in which image edges play an important role, for example, edge-preserving smoothing or tone mapping. The isotropic, spatially invariant, smooth Gaussian kernels on which the pyramids are built are considered almost antithetical to edge discontinuities, which are precisely located and anisotropic by nature. Further, the decimation of the levels, that is, the successive reduction by factor 2 of the resolution, is often criticized for introducing aliasing artifacts, leading some researchers (e.g., Li et al.<sup>21</sup>) to recommend its omission. These arguments are often cited as a motivation for more sophisticated schemes such as anisotropic diffusion,<sup>1,29</sup> neighborhood filters,<sup>19,34</sup> edge-preserving optimization,<sup>4,11</sup> and edge-aware wavelets.<sup>12</sup>

While Laplacian pyramids can be implemented using simple image-resizing routines, other methods rely on more sophisticated techniques. For instance, the bilateral filter relies on a spatially varying kernel,<sup>34</sup> optimization-based methods (e.g., Fattal et al.,<sup>13</sup> Farbman et al.,<sup>11</sup> Subr et al.,<sup>31</sup> and Bhat et al.<sup>4</sup>) minimize a spatially inhomogeneous energy, and other approaches build dedicated basis functions for each new image (e.g., Szeliski,<sup>33</sup> Fattal,<sup>12</sup> and Fattal et al.<sup>15</sup>). This additional level of sophistication is also often associated with practical shortcomings. The parameters of anisotropic diffusion are difficult to set because of the iterative nature of the process, neighborhood filters tend to over-sharpen edges,<sup>5</sup> and methods based on optimization do not scale well due to the algorithmic complexity of the solvers. While some of these shortcomings can be alleviated in postprocessing, for example, bilateral filtered edges can be smoothed,<sup>3,10,19</sup> this induces additional computation and parameter setting, and a method producing good results directly is preferable. In this paper, we demonstrate that state-of-the-art edge-aware filters can be achieved with standard Laplacian pyramids. We formulate our approach as the construction of the Laplacian pyramid of the filtered output. For each output pyramid coefficient, we render a filtered version of the full-resolution image, processed to have the desired properties according to the corresponding local image value at the same scale, build a new Laplacian pyramid from the filtered image, and then copy the corresponding coefficient to the output pyramid. The advantage of this approach is that while it may be nontrivial to produce an image with the desired property everywhere, it is often easier to obtain the property locally. For instance, global detail enhancement typically requires a nonlinear image decomposition (e.g., Fattal et al.,<sup>14</sup> Farbman et al.,<sup>11</sup> and Subr et al.<sup>31</sup>), but enhancing details in the vicinity of a pixel can be done with a simple S-shaped contrast curve centered on the pixel intensity. This local transformation only achieves the desired effect in the neighborhood of a pixel, but is sufficient to estimate the fine-scale Laplacian coefficient of the output. We repeat this process for each coefficient independently and collapse the pyramid to produce the final output.

The original version of this paper was published in *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2011)* 30, 4 (Aug. 2011), 68:1–68:12.

We motivate this approach by analyzing its effect on step edges and show that edges can be differentiated from small-scale details with a simple threshold on color differences. We propose an algorithm that has a  $\mathcal{O}(N \log N)$  complexity for an image with  $N$  pixels. While our algorithm is not as fast as other techniques, it can achieve visually compelling results hard to obtain with previous work. We demonstrate our approach by implementing a series of edge-aware filters such as edge-preserving smoothing, detail enhancement, tone mapping, and inverse tone mapping. We provide numerous results, including large-amplitude image transformations. None of them exhibit halos, thereby showing that high-quality halo-free results can be indeed obtained using only the Laplacian pyramid, which was previously thought impossible.

**Contributions.** The main contribution of this work is a flexible approach to achieve edge-aware image processing through simple point-wise manipulation of Laplacian pyramids. Our approach builds upon a new understanding of how image edges are represented in Laplacian pyramids and how to manipulate them in a local fashion. Based on this, we design a set of edge-aware filters that produce high-quality halo-free results (Figure 1).

## 2. RELATED WORK

**Edge-aware image processing.** Edge-aware image manipulation has already received a great deal of attention and we refer to books and surveys for an in-depth presentation.<sup>1,20,27</sup> Recently, several methods have demonstrated satisfying results with good performance (e.g., Chen et al.,<sup>7</sup> Farbman et al.,<sup>11</sup> Fattal,<sup>12</sup> Subr et al.,<sup>31</sup> Criminisi et al.,<sup>8</sup> He et al.,<sup>17</sup> and Kass and Solomon<sup>19</sup>). Our practical contribution is to provide filters that consistently achieve results at least as good, have easy-to-set parameters, can be implemented with only basic image-resizing routines, are noniterative, and do not rely on optimization or postprocessing. In particular, unlike gradient-domain methods (e.g., Fattal et al.<sup>13</sup>), we do not need to solve the Poisson equation which may introduce artifacts with nonintegrable gradient

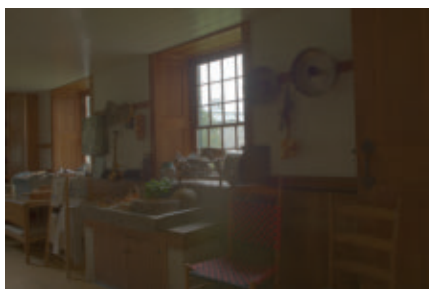
fields. From a conceptual standpoint, our approach is based on image pyramids and is inherently multiscale, which differentiates it from methods that are expressed as a two-scale decomposition (e.g., Chen et al.,<sup>7</sup> Subr et al.,<sup>31</sup> and He et al.<sup>17</sup>).

**Pyramid-based edge-aware filtering.** As described earlier, pyramids are not the typical representation of choice for filtering an image in an edge-preserving way, and only a few techniques along these lines have been proposed. A first approach is to directly rescale the coefficients of a Laplacian pyramid; however, this typically produces halos.<sup>21</sup> While halos may be tolerable in the context of medical imaging (e.g., Vuylsteke and Schoeters,<sup>36</sup> and Dippel et al.<sup>9</sup>), they are unacceptable in photography.

Fattal et al.<sup>13</sup> avoid halos by using a Gaussian pyramid to compute scaling factors applied to the image gradients. They reconstruct the final image by solving the Poisson equation. In comparison, our approach directly manipulates the Laplacian pyramid of the image and does not require global optimization. Fattal et al.<sup>14</sup> use a multiscale image decomposition to combine several images for detail enhancement. Their decomposition is based on repeated applications of the bilateral filter. Their approach is akin to building a Laplacian pyramid but without decimating the levels and with a spatially varying kernel instead of a Gaussian kernel. However, their study is significantly different from ours because it focuses on multi-image combination and speed. In a similar spirit, Farbman et al.<sup>11</sup> compute a multiscale edge-preserving decomposition with a least-squares scheme instead of bilateral filtering. This work also differs from ours since its main concern is the definition and application of a new optimization-based filter. In the context of tone mapping, Mantiuk et al.<sup>23</sup> model human perception with a Gaussian pyramid. The final image is the result of an optimization process, which departs from our goal of working only with pyramids.

Fattal<sup>12</sup> describes wavelet bases that are specific to each image. He takes edges explicitly into account to define the basis functions, thereby reducing the correlation between

**Figure 1.** We demonstrate edge-aware image filters based on the manipulation of Laplacian pyramids. Our approach produces high-quality results, without degrading edges or introducing halos, even at extreme settings. Our approach builds upon standard image pyramids and enables a broad range of effects via simple point-wise nonlinearities (shown in corners). For an example image (a), we show results of tone mapping using our method, creating a natural rendition (b) and a more exaggerated look that enhances details as well (c). Laplacian pyramids have previously been considered unsuitable for such tasks, but our approach shows otherwise.



(a) Input HDR image tone-mapped with a simple gamma curve (details are compressed)



(b) Our pyramid-based tone mapping, set to preserve details without increasing them



(c) Our pyramid-based tone mapping, set to strongly enhance the contrast of details

pyramid levels. From a conceptual point of view, our work and Fattal’s are complementary. Whereas he designed pyramids in which edges do not generate correlated coefficients, we seek to better understand this correlation to preserve it during filtering.

Li et al.<sup>21</sup> demonstrate a tone-mapping operator based on a generic set of spatially invariant wavelets, countering the popular belief that such wavelets are not appropriate for edge-aware processing. Their method relies on a corrective scheme to preserve the spatial and intrascale correlation between coefficients, and they also advocate computing each level of the pyramid at full resolution to prevent aliasing. However, when applied to Laplacian pyramids, strong corrections are required to avoid halos, which prevents a large increase of the local contrast. In comparison, in this work, we show that Laplacian pyramids can produce a wide range of edge-aware effects, including extreme detail amplification, without introducing halos.

Gaussian pyramids are closely related to the concept of Gaussian scale-space defined by filtering an image with a series of Gaussian kernels of increasing size. While these approaches are also concerned with the correlation between scales created by edges, they are used mostly for purposes of analysis (e.g., Witkin<sup>37</sup> and Witkin et al.<sup>38</sup>).

**Background on Gaussian and Laplacian pyramids.** Our approach is based on standard image pyramids, whose construction we summarize briefly (for more detail, see Burt and Adelson<sup>6</sup>). Given an image  $I$ , its *Gaussian pyramid* is a set of images  $\{G_\ell\}$  called *levels*, representing progressively lower resolution versions of the image, in which high-frequency details progressively disappear. In the Gaussian pyramid, the bottom-most level is the original image,  $G_0 = I$ , and  $G_{\ell+1} = \text{downsample}(G_\ell)$  is a low-pass version of  $G_\ell$  with half the width and height. The filtering and decimation process is iterated  $n$  times, typically until the level  $G_n$  has only a few pixels. The *Laplacian pyramid* is a closely related construct, whose levels  $\{L_\ell\}$  represent details at different spatial scales, decomposing the image into roughly separate frequency bands. Levels of the Laplacian pyramid are defined by the details that distinguish successive levels of the Gaussian pyramid,  $L_\ell = G_\ell - \text{upsample}(G_{\ell+1})$ , where  $\text{upsample}(\cdot)$  is an operator that doubles the image size in each dimension using a smooth kernel. The top-most level of the Laplacian pyramid, also called the *residual*, is defined as  $L_n = G_n$  and corresponds to a tiny version of the image. A Laplacian pyramid can be *collapsed* to reconstruct the original image by recursively applying  $G_\ell = L_\ell + \text{upsample}(G_{\ell+1})$  until  $G_0 = I$  is recovered.

### 3. DEALING WITH EDGES IN LAPLACIAN PYRAMIDS

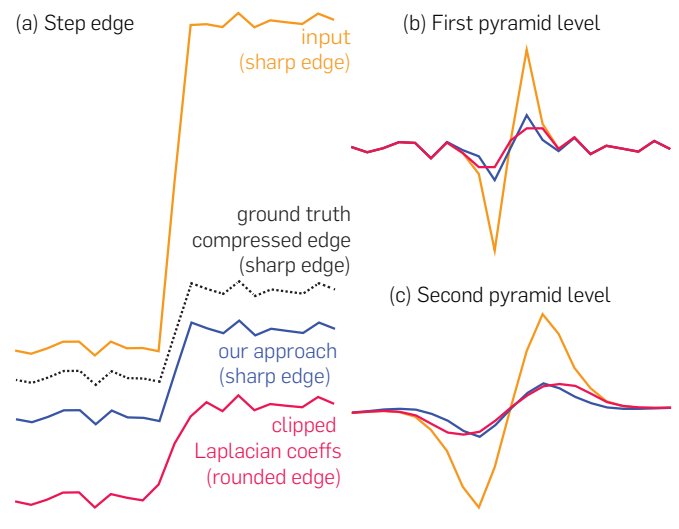
The goal of edge-aware processing is to modify an input signal  $I$  to create an output  $I'$ , such that the large discontinuities of  $I$ , that is, its edges, remain in place, and such that their profiles retain the same overall shape. For example, the amplitude of significant edges may be increased or reduced, but the edge transitions should not become smoother or sharper. The ability to process images in this edge-aware fashion is particularly important for techniques that manipulate the image in a spatially varying way, such

as image enhancement or tone mapping. Failure to account for edges in these applications leads to distracting visual artifacts such as halos, shifted edges, or reversals of gradients. In the following discussion, for the sake of illustration, we focus on the case where we seek to reduce the edge amplitude—the argument when increasing the edge amplitude is symmetric.

In this work, we characterize edges by the magnitude of the corresponding discontinuity in a color space that depends on the application; we assume that variations due to edges are larger than those produced by texture. This model is similar to many existing edge-aware filtering techniques (e.g., Aubert and Kornprobst<sup>1</sup> and Paris et al.<sup>27</sup>); we will discuss later the influence that this assumption has on our results. Because of this difference in magnitude, Laplacian coefficients representing an edge also tend to be larger than those due to texture. A naive approach to decrease the edge amplitude while preserving the texture is to truncate these large coefficients. While this creates an edge of smaller amplitude, it ignores the actual “shape” of these large coefficients and assigns the same lower value to all of them. This produces an overly smooth edge, as shown in Figure 2.

Intuitively, a better solution is to scale down the coefficients that correspond to edges, to preserve their profile, and to keep the other coefficients unchanged, so that only the edges are altered. However, it is unclear how to separate these two kinds of coefficients since edges with different profiles generate different coefficients across scales. On the other hand, according to our model, edges

**Figure 2. Range compression applied to a step edge with fine details (a). The different versions of the edge are offset vertically so that their profiles are clearly visible. Truncating the Laplacian coefficients smooths the edge (red), an issue which Li et al.<sup>21</sup> have identified as a source of artifacts in tone mapping. In comparison, our approach (blue) preserves the edge sharpness and very closely reproduces the desired result (black). Observing the shape of the first two levels (b, c) shows that clipping the coefficients significantly alters the shape of the signal (red vs. orange). The truncated coefficients form wider lobes whereas our approach produces profiles nearly identical to the input (blue vs. orange).**

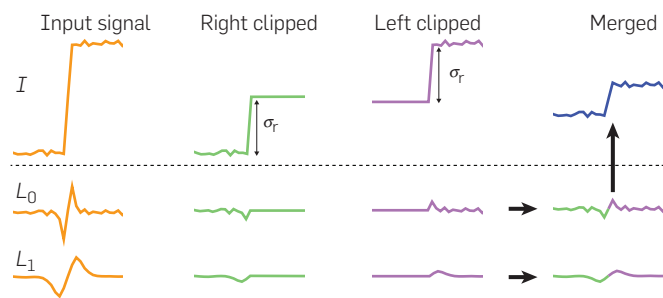


are easy to identify in image space; a threshold on color differences suffices to differentiate edges from variations due to texture. This is a key aspect of our approach: we generate new pyramid coefficients by working primarily on the *input image* itself, rather than altering the pyramid coefficients directly.

The overall design of our algorithm derives from this insight: we build an approximation of the desired output image specific to *each pyramid coefficient*. This is a major difference with the existing literature. Whereas previous techniques are formulated in terms of optimization (e.g., Farbman et al.<sup>11</sup>), PDEs (e.g., Perona and Malik<sup>29</sup>), or local averaging (e.g., Tomasi and Manduchi<sup>34</sup>), we express our filter through the computation of these local image approximations together with standard image pyramid manipulations. In practice, we use locally processed versions of the input to recompute values for each pyramid coefficient, and combine all of these new coefficient values into the final result. For each coefficient at location  $(x, y)$  and level  $\ell$ , we first determine the region in the input image on which this coefficient depends. To reduce the amplitude of edges, for example, we clamp all the pixels values in that region so that the difference to the average value does not exceed a user-provided threshold. This processed image has the desired property that edges are now limited in amplitude, to at most twice the threshold. This also has the side effect of flattening the details across the edge. As we discuss below, these details are not lost, they are actually captured by pyramid coefficients centered on the other side of the edge as illustrated in Figure 3. Then, we compute the Laplacian pyramid of this processed image to create coefficients that capture this property. In particular, this gives us the value of the coefficient  $(x, y, \ell)$  that we seek. Another way of interpreting our method is that we locally filter the image, for example, through a local contrast decrease, and then determine the corresponding coefficient in the Laplacian pyramid. We repeat this process, such that each coefficient in the pyramid is computed.

**Detail preservation.** As mentioned earlier, a reasonable concern at this point is that the clamped image has lost

**Figure 3. Simple view of our range compression approach, which is based on thresholding and local processing. For a step-like signal similar to the one in Figure 2, our method effectively builds two Laplacian pyramids, corresponding to clipping the input based on the signal value to the left and right of the step edge, then merging their coefficients as indicated by the color coding.**



details in the thresholded regions, which in turn could induce a loss in the final output. However, the loss of details does not transfer to our final result. Intuitively, the clamped details are on “the other side of the edge” and are represented by other coefficients. Applying this scheme to all pyramid coefficients accurately represents the texture on each side of the edge, while capturing the reduction in edge amplitude (Figure 3). Further, clamping affects only half of the edge and, by combining coefficients on “both sides of the edge,” our approach reconstructs an edge profile that closely resembles the input image, that is, the output profiles do not suffer from oversmoothing. Examining the pyramid coefficients reveals that our scheme fulfills our initial objective, that is, that the edge coefficients are scaled down while the other coefficients representing the texture are preserved (Figure 2).

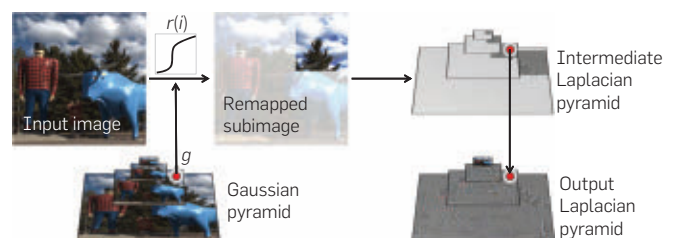
#### 4. LOCAL LAPLACIAN FILTERING

We now formalize the intuition gained in the previous section and introduce *Local Laplacian Filtering*, our new method for edge-aware image processing based on the Laplacian pyramid. A visual overview is given in Figure 4 and the pseudo-code is provided in Algorithm 1.

In Local Laplacian Filtering, an input image is processed by constructing the Laplacian pyramid  $\{L[I']\}$  of the output, one coefficient at a time. For each coefficient  $(x, y, \ell)$ , we generate an intermediate image  $\tilde{I}$  by applying a point-wise monotonic remapping function  $r_{g,\sigma}(\cdot)$  to the original full-resolution image. This remapping function, whose design we discuss later, depends on the local image value from the Gaussian pyramid  $g = G_\ell(x, y)$  and the user parameter  $\sigma$  which is used to distinguish edges from details. We compute the pyramid for the intermediate image  $\{L[\tilde{I}]\}$  and copy the corresponding coefficient to the output  $\{L[I']\}$ . After all coefficients of the output pyramid have been computed, we collapse the output pyramid to get the final result.

A direct implementation of this algorithm yields a complexity in  $\mathcal{O}(N^2)$  with  $N$  being the number of pixels in the image, since each coefficient entails the construction of another pyramid with  $\mathcal{O}(N)$  pixels. However, this cost can be

**Figure 4. Overview of the basic idea of our approach. For each pixel in the Gaussian pyramid of the input (red dot), we look up its value  $g$ . Based on  $g$ , we remap the input image using a point-wise function, build a Laplacian pyramid from this intermediate result, then copy the appropriate pixel into the output Laplacian pyramid. This process is repeated for each pixel over all scales until the output pyramid is filled, which is then collapsed to give the final result. For more efficient computation, only parts of the intermediate pyramid need to be generated.**



reduced in a straightforward way by processing only the subpyramid needed to evaluate  $L_\ell[\bar{I}](x, y)$ , illustrated in Figure 4. The base of this subpyramid lies within a  $K \times K$  subregion  $R$  of the input image  $I$ , where  $K = \mathcal{O}(2^\ell)$ ; for Laplacian pyramids built using a standard 5-tap interpolation filter, it can be shown that  $K = 3(2^{\ell+2} - 1)$ . Put together with the fact that level  $\ell$  contains  $\mathcal{O}(N/2^\ell)$  coefficients, each level requires the manipulation of  $\mathcal{O}(N)$  coefficients in total. Since there are  $\mathcal{O}(\log N)$  levels in the pyramid, the overall complexity of our algorithm is  $\mathcal{O}(N \log N)$ . Later we will see that some applications only require a fixed number of levels to be processed or limit the depth of the subpyramids to a fixed value, reducing the complexity of our algorithm further.

**Remapping function for gray-scale images.** We assume the user has provided a parameter  $\sigma$  such that intensity

**Algorithm 1**  $\mathcal{O}(N \log N)$  Version of Local Laplacian Filtering

**input:** image  $I$ , parameter  $\sigma$ , remapping function  $r$   
**output:** image  $I'$

- 1: compute input Gaussian pyramid  $\{G[I]\}$
- 2: **for all** coefficients at position  $(x, y)$  and level  $\ell$  **do**
- 3:  $g \leftarrow G_\ell(x, y)$
- 4: determine subregion  $R$  of  $I$  needed to evaluate  $L_\ell(x, y)$
- 5: create temporary buffer  $\bar{R}$  of the same size
- 6: **for all** pixels  $(u, v)$  of  $R$  **do**
- 7: apply remapping function:  $\bar{R}(u, v) \leftarrow r(R(u, v), g, \sigma)$
- 8: **end for**
- 9: compute subpyramid  $\{L[\bar{R}]\}$
- 10: update output pyramid:  $L_\ell[I'](x, y) \leftarrow L_\ell[\bar{R}](x, y)$
- 11: **end for**
- 12: collapse output pyramid:  $I' \leftarrow \text{collapse}(\{L_\ell[I']\})$

Our algorithm considers the pyramid coefficients one by one (Step 2). Each of them is computed using the pixels from the finest resolution (Step 4) by applying the remapping function to them (Step 7) and building a Laplacian pyramid of the remapped data (Step 9). We copy the relevant coefficient into the output pyramid (Step 10) and once all the coefficients have been computed, we collapse pyramid the get the final result (Step 12).

variations smaller than  $\sigma$  should be considered fine-scale details and larger variations are edges. As a center point for this function we use  $g = G_\ell(x, y)$ , which represents the image intensity at the location and scale where we compute the output pyramid coefficient. Intuitively, pixels closer than  $\sigma$  to  $g$  should be processed as details and those farther than  $\sigma$  away should be processed as edges. We differentiate their treatment by defining two functions  $r_d$  and  $r_e$ , such that  $r(i) = r_d(i)$  if  $|i - g| \leq \sigma$  and  $r(i) = r_e(i)$  otherwise. Since we require  $r$  to be monotonically increasing,  $r_d$  and  $r_e$  must have this property as well. Furthermore, to avoid the creation of spurious discontinuities, we constrain  $r_d$  and  $r_e$  to be continuous by requiring that  $r_d(g \pm \sigma) = r_e(g \pm \sigma)$ .

The function  $r_d$  modifies the fine-scale details by altering the oscillations around the value  $g$ . In our applications we process positive and negative details symmetrically, letting us write:

$$r_d(i, g, \sigma) = g + \text{sign}(i - g) \sigma f_d(|i - g| / \sigma) \quad (1)$$

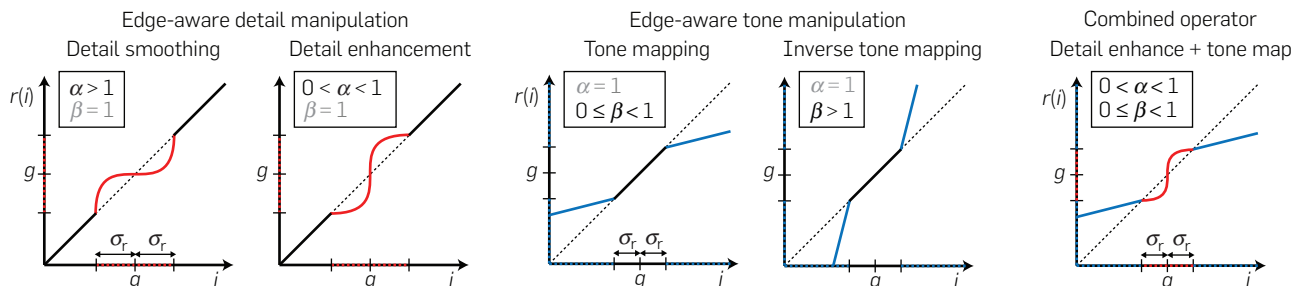
where  $f_d$  is a smooth function mapping from  $[0, 1]$  to  $[0, 1]$  that controls how details are modified. The advantage of this formulation is that it depends only on the amplitude of the detail  $|i - g|$  relative to the parameter  $\sigma$ , that is,  $|i - g| / \sigma = 1$  corresponds to a detail of maximum amplitude according to the user-defined parameter. Analogously,  $r_e$  is a function that modifies the amplitude of edges that we again formulate in a symmetric way:

$$r_e(i, g, \sigma) = g + \text{sign}(i - g) (f_e(|i - g| - \sigma) + \sigma) \quad (2)$$

where  $f_e$  is a smooth nonnegative function defined over  $[0, \infty)$ . In this formulation,  $r_e$  depends only on the amplitude above the user threshold  $\sigma$ , that is,  $|i - g| - \sigma$ . The function  $f_e$  controls how the edge amplitude is modified since an edge of amplitude  $\sigma + f_e(a)$  becomes an edge with amplitude  $\sigma + f_e(a)$ . For our previous 1D range compression example, clipping edges corresponds to  $f_e = 0$ , which limits the amplitude of all edges to  $\sigma$ . Useful specific choices for  $r_d$  and  $r_e$  are described in the next section and are illustrated in Figure 5.

The advantage of the functional forms defined in Equations (1) and (2) is that they ensure that  $r$  is continuous and increasing, and the design of a specific filter boils down to defining the two point-wise functions  $f_d$  and  $f_e$  that each

**Figure 5. Family of point-wise functions for edge-aware manipulation described in Sections 5.2 and 5.3. The parameters  $\alpha$  and  $\beta$  let us control how detail and tone are processed respectively. To compute a given Laplacian coefficient in the output, we filter the original image point-wise using a nonlinear function  $r(i)$  of the form shown. This remapping function is parametrized by the Gaussian pyramid coefficient  $g$ , describing the local image content, and a threshold  $\sigma$  used to distinguish fine details (red) from larger edges (blue).**



have clear roles:  $f_d$  controls the amplification or attenuation of details while  $f_e$  controls the amplification or attenuation of edges.

**Extension to color images.** To handle color, it is possible to treat only the luminance channel and reintroduce chrominance after image processing (Section 5.3). However, our approach extends naturally to color images as well, letting us deal directly with 3D vectors representing, for example, the RGB or CIE-Lab channels. Algorithm 1 still applies, and we need only to update  $r_d$  and  $r_e$ , using bold typeface to indicate vectors:

$$r_d(\mathbf{i}, \mathbf{g}, \sigma) = \mathbf{g} + \text{unit}(\mathbf{i} - \mathbf{g})\sigma f_d(\|\mathbf{i} - \mathbf{g}\|/\sigma) \quad (3a)$$

$$r_e(\mathbf{i}, \mathbf{g}, \sigma) = \mathbf{g} + \text{unit}(\mathbf{i} - \mathbf{g})[f_e(\|\mathbf{i} - \mathbf{g}\| - \sigma) + \sigma] \quad (3b)$$

with  $\text{unit}(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|$  if  $\mathbf{v} \neq \mathbf{0}$  and  $\mathbf{0}$  otherwise. These equations define details as colors within a ball of radius  $\sigma$  centered at  $\mathbf{g}$  and edges as the colors outside it. They also do not change the roles of  $f_d$  and  $f_e$ , letting the same 1D functions that modify detail and edges in the gray-scale case be applied to generate similar effects in color. For images whose color channels are all equal, these formulas reduce to the gray-scale formulas of Equations (1) and (2).

## 5. APPLICATIONS AND RESULTS

We now demonstrate how to realize practical image processing applications using our approach and discuss implementation details. First we address edge-preserving smoothing and detail enhancement, followed by tone mapping and related tools. We validate our method with images used previously in the literature<sup>10–13, 27</sup> and demonstrate that our method produces artifact-free results.

### 5.1. Implementation

We use the pyramids defined by Burt and Adelson,<sup>6</sup> based on  $5 \times 5$  kernels. On a 2.26 GHz Intel Xeon CPU, we process a one-megapixel image in about a minute using a single thread. This can be halved by limiting the depth of the intermediate pyramid to at most five levels, by applying the remapping to level  $\max(0, \ell - 3)$  rather than always starting at the full-resolution image. This amounts to applying the remapping to a downsampled version of the image when processing coarse pyramid levels. The resulting images

are visually indistinguishable from the full-pyramid process with a PSNR on the order of 30 to 40 dB. While this performance is slower than previous work, our algorithm is highly data parallel and can easily exploit a multicore architecture. Using OpenMP, we obtain an  $8\times$  speed-up on an 8-core machine, bringing the running time down to 4 seconds.

### 5.2. Detail manipulation

To modify the details of an image, we define an S-shaped point-wise function as is classically used for the local manipulation of contrast. For this purpose, we use a power curve  $f_d(\Delta) = \Delta^\alpha$ , where  $\alpha > 0$  is a user-defined parameter. Values larger than 1 smooth the details out, while values smaller than 1 increase their contrast (Figures 5 and 6). To restrict our attention to the details of an image, we set the edge-modifying function to the identity  $f_e(a) = a$ .

In the context of detail manipulation, the parameter  $\sigma$  controls how at what magnitude signal variations should be considered edges and therefore be preserved. Large values allow the filter to alter larger portions of the signal and yield larger visual changes (Figure 7). In its basic form, detail manipulation is applied at all scales, but one can also control which scales are affected by limiting processing to a subset of the pyramid levels (Figure 6c, d, e). While this control is discrete, the changes are gradual, and one can interpolate between the results from two subsets of levels if continuous control is desired. Our results from Figures 6 and 7 are comparable to results of Farbmán et al.<sup>11</sup>; however, we do not require the complex machinery of a multiresolution preconditioned conjugate gradient solver. Note that our particular extension to color images allows us to boost the color contrast as well (Figures 6, 7, and 8).

**Reducing noise amplification.** As in other techniques for texture enhancement, increasing the contrast of the details may make noise and artifacts from lossy image compression more visible. We mitigate this issue by limiting the smallest  $\Delta$  amplified. In our implementation, when  $\alpha < 1$ , we compute  $f_d(\Delta) = \tau\Delta^\alpha + (1 - \tau)\Delta$ , where  $\tau$  is a smooth step function equal to 0 if  $\Delta$  is less than 1% of the maximum intensity, 1 if it is more than 2%, with a smooth transition in between. All the results in this paper and supplemental material are computed with this function.

**Figure 6. Smoothing and enhancement of detail, while preserving edges ( $\sigma = 0.3$ ). Processing only a subset of the levels controls the frequency of the details that are manipulated (c, d, e). The images have been cropped to make the flower bigger and its details more visible.**





### 5.3. Tone manipulation

Our approach can also be used for reducing the intensity range of a high-dynamic-range (HDR) image, according to the standard tone mapping strategy of compressing the large-scale variations while preserving (or enhancing) the details.<sup>35</sup> In our framework, we manipulate large-scale variations by defining a point-wise function modifying the edge amplitude,  $f_c(a) = \beta a$ , where  $\beta \geq 0$  is a user-defined parameter (Figure 5).

In our implementation of tone manipulation, we process the image intensity channel only and keep the color unchanged.<sup>10</sup> We compute an intensity image  $I_i = \frac{1}{61}(20I_r + 40I_g + I_b)$  and color ratios  $(\rho_r, \rho_g, \rho_b) = \frac{1}{I_i}(I_r, I_g, I_b)$ , where  $I_r, I_g,$  and  $I_b$  are the RGB channels. We apply our filter on the log intensities  $\log(I_i)$ ,<sup>35</sup> using the natural logarithm. For tone mapping, we set our filter with  $\alpha \leq 1$  so that details are preserved or enhanced, and  $\beta < 1$  so that edges are compressed. This produces new values  $\log(I_i')$ , which we must then map to the displayable range of  $[0, 1]$ . We remap the result  $\log(I_i')$  by first offsetting its values to make its maximum 0, then scaling them so that they cover a user-defined range.<sup>10, 21</sup> In our implementation, we estimate a robust maximum and minimum with the 99.5th and 0.5th percentiles, and we set the scale

factor so that the output dynamic range is 100:1 for the linear intensities. Finally, we multiply the intensity by the color ratios  $(\rho_r, \rho_g, \rho_b)$  to obtain the output RGB channels, then gamma correct with an exponent of 1/2.2 for display. We found that fixing the output dynamic range not only makes it easy to achieve a consistent look but also constrains the system. As a result, the  $\sigma$  and  $\beta$  parameters have similar effects, both controlling the balance between local and global contrast in the rendered image (Figure 9). From a practical standpoint, we advise keeping  $\sigma$  fixed and varying the slope  $\beta$  between 0, where the local contrast is responsible for most of the dynamic range, and 1, where the global contrast dominates. Unless otherwise specified, we use  $\sigma = \log(2.5)$ , which gave consistently good results in our experiments. Since we work in the log domain, this value corresponds to a ratio between pixel intensities. It does not depend on the dynamic range of the scene, and assumes only that the input HDR image measures radiance up to scale.

Our tone mapping operator builds upon standard elements from previous work that could be substituted for others. For instance, one could instead use a sigmoid to remap the intensities to the display range<sup>30</sup> or use a different color management method (e.g., Mantiuk et al.<sup>24</sup>). Also, we did not apply any additional “beautifying curve” or increased saturation as is commonly done in photo editing software. Our approach produces a clean output image that can be post-processed in this way if desired.

Range compression is a good test case to demonstrate the abilities of our pyramid-based filters because of the large modification involved. For high compression, even subtle inaccuracies can become visible, especially at high-contrast edges. In our experiments, we did not observe aliasing or oversharpening artifacts even on cases where other methods suffer from them (Figures 10 and 11). We also stress-tested our operator by producing

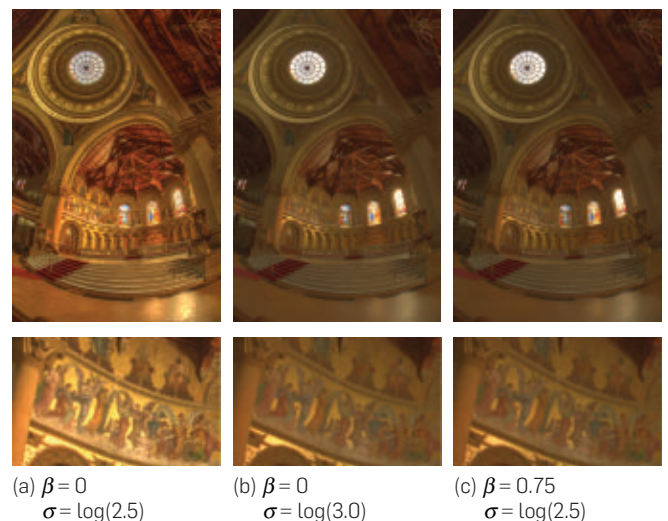
**Figure 7. Effect of the  $\sigma$  parameter for detail enhancement ( $\alpha = 0.25$ ). Same input as Figure 6.**



**Figure 8. Filtering only the luminance (b) preserves the original colors in (a), while filtering the RGB channels (c) also modifies the color contrast ( $\alpha = 0.25, \beta = 1, \sigma = 0.4$ ).**



**Figure 9.  $\beta$  and  $\sigma$  have similar effects on tone mapping results, they control the balance between global and local contrast.  $\alpha$  is set to 1 in all three images.**

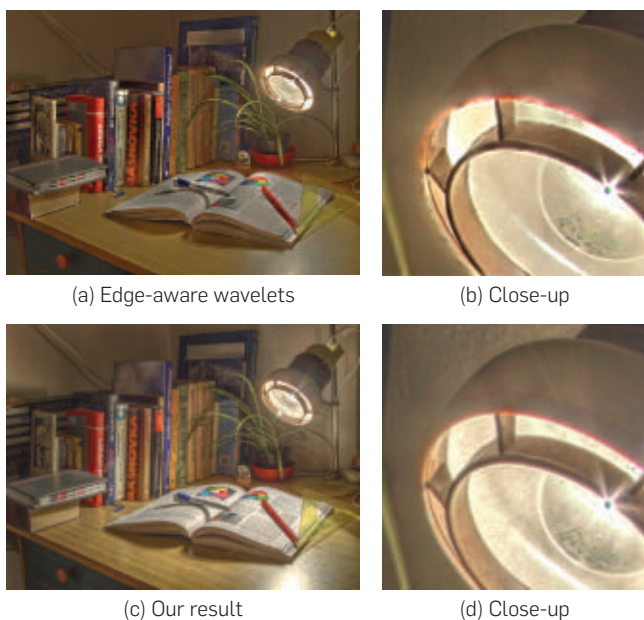


results with a low global contrast ( $\beta = 0$ ) and high local details ( $\alpha = 0.25$ ). In general, the results produced by our method did not exhibit any particular problems (Figure 12). We compare exaggerated renditions of our method with Farbman et al.<sup>11</sup> and Li et al.<sup>21</sup> Our method produces consistent results without halos, whereas the other methods either create halos or fail to exaggerate detail (Figure 13).

One typical difficulty we encountered is that sometimes the sky interacts with other elements to form high-frequency textures that undesirably get amplified by our detail-enhancing filter (Figures 8b and 14). Such “misinterpretation” is common to all low-level filters without semantic understanding of the scene, and typically requires user feedback to correct.<sup>22</sup>

We also experimented with inverse tone mapping, using slope values  $\beta$  larger than 1 to increase the dynamic range of a normal image. Since we operate on log intensities, roughly speaking, the linear dynamic range gets exponentiated by  $\beta$ . Applying our tone-mapping operator on these range-expanded results gives images close to the originals, typically with a PSNR between 25 and 30 dB for  $\beta = 2.5$ . This shows that our inverse tone mapping preserves the image content well. While a full-scale study on an HDR monitor is beyond the scope of this paper, we believe that our simple approach can complement other relevant techniques (e.g., Masia et al.<sup>25</sup>). Sample HDR results are provided in supplemental material.

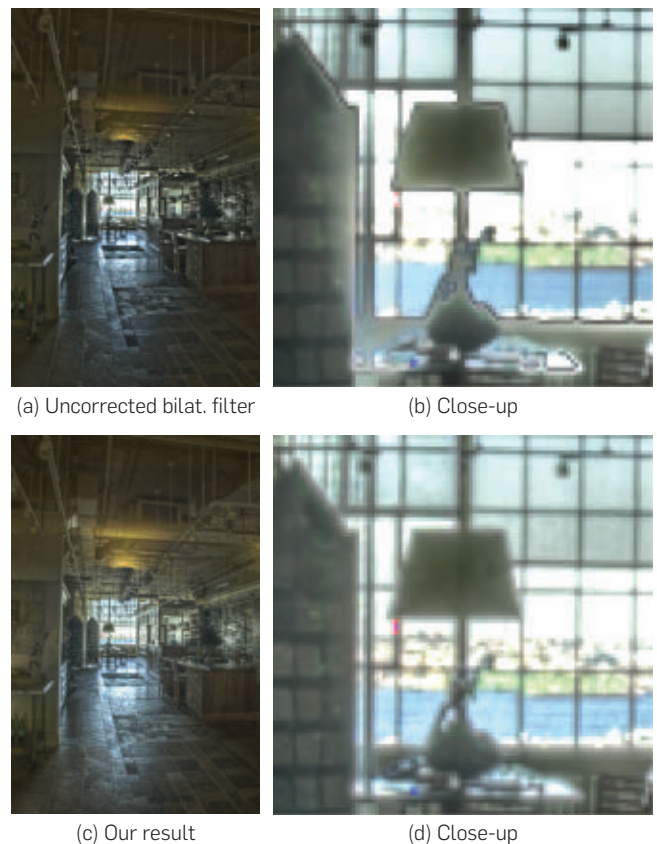
**Figure 10. The extreme contrast near the light bulb is particularly challenging. Images (a) and (b) are reproduced from Fattal.<sup>12</sup> The edge-aware wavelets suffer from aliasing and generate an irregular edge (b). In comparison, our approach (d) produces a clean edge. We set our method to approximately achieve the same level of details ( $\sigma = \log(3.5)$ ,  $\alpha = 0.5$ ,  $\beta = 0$ ).**



### 5.4. Discussion

While our method can fail in the presence of excessive noise or when extreme parameter settings are used (e.g., the *lenna* picture in supplemental material has a high level of noise), we found that our filters are very robust and behave well over a broad range of settings. Figure 15 shows a variety of parameters values applied to the same image and the results are consistently satisfying, high-quality, and halo-free; many more such examples are provided in supplemental material. While the goal of edge-aware processing can be ill-defined, the results that we obtain show that our approach allows us to realize many edge-aware effects with intuitive parameters and a simple implementation. The current shortcoming of our approach is its running time. We can mitigate this issue, thanks to the multiscale nature of our algorithm, allowing us to generate quick previews that are faithful to the full-resolution results (Figure 16). Furthermore, the algorithm is highly parallelizable and should lend itself to a fast GPU implementation. Beyond these practical aspects, our main contribution is a better characterization of the multiscale properties of images.

**Figure 11. The bilateral filter sometimes oversharpens edges, which can lead to artifacts (b). We used code provided by Paris and Durand<sup>26</sup> and multiplied the detail layer by 2.5 to generate these results. Although such artifacts can be fixed in postprocessing, this introduces more complexity to the system and requires new parameters. Our approach produces clean edges directly (d). We set our method to achieve approximately the same visual result ( $\sigma = \log(2.5)$ ,  $\alpha = 0.5$ ,  $\beta = 0$ ).**



**Figure 12.** We stressed our approach by applying a strong range compression coupled with a large detail increase ( $\alpha = 0.25$ ,  $\beta = 0$ ,  $\sigma = \log(2.5)$ ). The results are dominated by local contrast and are reminiscent of the popular, exaggerated “HDR look” but without the unsightly halos associated with it. In terms of image quality, our results remain artifact-free in most cases. We explore further parameter variations in the supplemental material.



**Figure 13.** We compare exaggerated, tone-mapped renditions of an HDR image. The wavelet-based method by Li et al.<sup>21</sup> is best suited for neutral renditions and generates halos when one increases the level of detail (a). The multiscale method by Farbman et al.<sup>11</sup> performs better and produces satisfying results for intermediate levels of detail (b), but halos and edge artifacts sometimes appear for a larger increase, as in this image for instance; see the edge of the white square on the blue book cover and the edge of the open book (c). In comparison, our approach achieves highly detailed renditions without artifacts (d). These results as well as many others may be better seen in the supplemental material.

- (a) Li et al.<sup>21</sup> (detailed rendition using parameters suggested by the authors)
- (b) Farbman et al.<sup>11</sup> (detailed rendition using parameters suggested by the authors)
- (c) Farbman et al.<sup>11</sup> (exaggerated rendition using parameters suggested by the authors)
- (d) Our result with exaggerated details ( $\alpha = 0.25$ ,  $\beta = 0$ )



Many problems related to photo editing are grounded in these properties of images and we believe that a better understanding can have benefits beyond the applications demonstrated in this paper.

## 6. CONCLUSION

**Link to recent work.** We first presented this work at the ACM SIGGRAPH conference in 2011. The main difference

**Figure 14.** Our approach is purely signal-based and its ignorance of scene semantics can lead to artifacts. For a large increase in local contrast (b), at a level similar to Figure 12, the sky gets locally darker behind clouds, because it forms a blue-white texture amplified by our filter. Our result for this example is good elsewhere, and this issue does not appear with a more classical rendition (a).



(a) No detail increase ( $\alpha = 1$ )      (b) Detail increased ( $\alpha = 0.25$ )

with our original article is Section 3 that now focuses on qualitative properties of edges. A formal discussion of these properties can be found in Paris et al.<sup>28</sup> Since then, we also extended this work with a fast algorithm that makes Local Laplacian Filters practical, an analysis that shows their relationship to the Bilateral Filter, an application to the transfer of gradient histograms applied to photographic style transfer, and additional comparisons with existing techniques such as the Guided Filter.<sup>17</sup> These results are described in Aubry et al.<sup>2</sup>

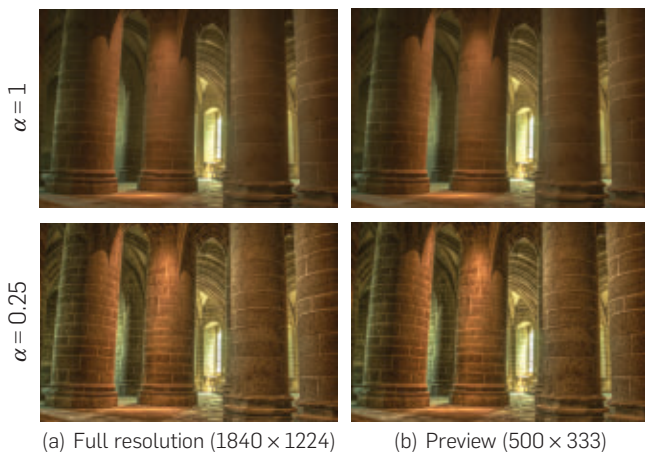
Although Local Laplacian Filters can reduce image details, Xu et al.<sup>39, 40</sup> have shown that they do not fully remove them and have proposed filters that completely suppress details for applications such as cartoon rendering and mosaic texture removal. By addressing the extreme detail removal problem, this work is complementary to Local Laplacian Filters that perform well at extreme detail increase. Hadwiger et al.<sup>16</sup> have introduced a dedicated data structure to process very large images efficiently and have demonstrated its application to Local Laplacian Filtering.

**Closing note.** We have presented a new technique for edge-aware image processing based solely on the Laplacian pyramid. It is conceptually simple, allows for a wide range of edge-aware filters, and consistently produces artifact-free images. We demonstrate high-quality results over a large variety of images and parameter settings, confirming the method's robustness. Our results open new perspectives on multiscale image analysis and editing since Laplacian pyramids were previously considered as ill-suited for manipulating edges. Given the wide use of pyramids and the need

**Figure 15.** Our filter to enhance and reduce details covers a large space of possible outputs without creating halos.




**Figure 16. Our approach generates faithful previews when applied to a low-resolution version of an image ( $\beta = 0$ ,  $\sigma = \log(2.5)$ ).**



for edge-aware processing, we believe our new insights can have a broad impact in the domain of image editing and its related applications.

### Acknowledgments

We thank Ted Adelson, Bill Freeman, and Frédo Durand for inspiring discussions and encouragement; Alan Erickson for the *Orion* image; and the anonymous reviewers for their constructive comments. This work was supported in part by an NSERC Postdoctoral Fellowship, the Quanta T-Party, NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, and gifts from Microsoft, Google, and Adobe. We thank Farbman et al. and Li et al. for their help with comparisons. 

### References

- Aubert, G. and Kornprobst, P. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Vol. 147 of Applied Mathematical Sciences. Springer, 2002.
- Aubry, M., Paris, S., Hasinoff, S.W., Kautz, J., and Durand, F. *Fast and Robust Pyramid-based Image Processing*. Tech. Rep. MIT-CSAIL-TR-2011-049. MIT, 2011.
- Bae, S., Paris, S., and Durand, F. Two-scale tone management for photographic look. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25, 3 (2006), 637–645.
- Bhat, P., Zitnick, C.L., Cohen, M., and Curless, B. Gradientshop: A gradient-domain optimization framework for image and video filtering. *ACM Trans. Graph.* 29 (2010), 2.
- Buades, A., Coll, B., and Morel, J.-M. The staircasing effect in neighborhood filters and its solution. *IEEE Trans. Image Process.* 15 (2006), 6.
- Burt, P.J. and Adelson, E.H. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.* 31 (1983), 4.
- Chen, J., Paris, S., and Durand, F. Real-time edge-aware image processing with the bilateral grid. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26 (2007), 3.
- Criminisi, A., Sharp, T., Rother, C., and Perez, P. Geodesic image and video editing. *ACM Trans. Graph.* 29 (2010), 5.
- Dippel, S., Stahl, M., Wiemker, R., and Blaffert, T. Multiscale contrast enhancement for radiographies: Laplacian pyramid versus fast wavelet transform. *IEEE Trans. Med. Imaging* 21 (2002), 4.
- Durand, F. and Dorsey, J. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph. (Proc. SIGGRAPH)* 21 (2002), 3.
- Farbman, Z., Fattal, R., Lischinski, D., and Szeliski, R. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 27 (2008), 3.
- Fattal, R. Edge-avoiding wavelets and their applications. *ACM Trans. Graph. (Proc. SIGGRAPH)* 28 (2009), 3.
- Fattal, R., Lischinski, D., and Werman, M. Gradient domain high dynamic range compression. *ACM Trans. Graph. (Proc. SIGGRAPH)* 21 (2002), 3.

- Fattal, R., Agrawala, M., and Rusinkiewicz, S. Multiscale shape and detail enhancement from multi-light image collections. *ACM Trans. Graph. (Proc. SIGGRAPH)* 26 (2007), 3.
- Fattal, R., Carroll, R., and Agrawala, M. Edge-based image coarsening. *ACM Trans. Graph.* 29 (2009), 1.
- Hadwiger, M., Sicat, R., Beyer, J., Krüger, J., and Möller, T. Sparse PDF maps for non-linear multiresolution image operations. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 31 (2012), 5.
- He, K., Sun, J., and Tang, X. Guided image filtering. In *Proceedings of European Conference on Computer Vision (Proc. ECCV)* (2010).
- Heeger, D.J. and Bergen, J.R. Pyramid-based texture analysis/synthesis. In *Proceedings of the ACM SIGGRAPH Conference (Proc. SIGGRAPH)* (1995).
- Kass, M. and Solomon, J. Smoothed local histogram filters. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29 (2010), 3.
- Kimmel, R. *Numerical Geometry of Images: Theory, Algorithms, and Applications*. Springer, 2003.
- Li, Y., Sharan, L., and Adelson, E.H. Compressing and companding high dynamic range images with subband architectures. *ACM Trans. Graph. (Proc. SIGGRAPH)* 24 (2005), 3.
- Lischinski, D., Farbman, Z., Uyttendaele, M., and Szeliski, R. Interactive local adjustment of tonal values. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25 (2006), 3.
- Mantiuk, R., Myszkowski, K., and Seidel, H.-P. A perceptual framework for contrast processing of high dynamic range images. *ACM Trans. Appl. Percept.* 3 (2006), 3.
- Mantiuk, R., Mantiuk, R., Tomaszewska, A., and Heidrich, W. Color correction for tone mapping. *Comput. Graph. Forum (Proc. Eurographics)* 28 (2009), 2.
- Masia, B., Agustín, S., Fleming, R.W., Sorkine, O., and Gutierrez, D. Evaluation of reverse tone mapping through varying exposure conditions. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 28 (2009), 5.
- Paris, S. and Durand, F. Tone-mapping code. <http://people.csail.mit.edu/sparis/code/src/tonemapping.zip>.
- Paris, S., Kornprobst, P., Tumblin, J., and Durand, F. Bilateral filtering: Theory and applications. *Found. Trends Comput. Graph. Vision* 4, 1 (2009), 1–74.
- Paris, S., Hasinoff, S.W., and Kautz, J. Local Laplacian Filters: Edge-aware image processing with a Laplacian pyramid. *ACM Trans. Graph. (Proc. SIGGRAPH)* 30 (2011), 4.
- Perona, P. and Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (1990), 7.
- Reinhard, E., Stark, M., Shirley, P., and Ferwerda, J. Photographic tone reproduction for digital images. *ACM Trans. Graph. (Proc. SIGGRAPH)* 21 (2002), 3.
- Subr, K., Soler, C., and Durand, F. Edge-preserving multiscale image decomposition based on local extrema. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 28 (2009), 5.
- Sunkavalli, K., Johnson, M.K., Matusik, W., and Pfister, H. Multiscale image harmonization. *ACM Trans. Graph. (Proc. SIGGRAPH)* 29 (2010), 3.
- Szeliski, R. Locally adapted hierarchical basis preconditioning. *ACM Trans. Graph. (Proc. SIGGRAPH)* 25 (2006), 3.
- Tomasi, C. and Manduchi, R. Bilateral filtering for gray and color images. In *Proceedings of the IEEE International Conference on Computer Vision (Bombay, India, 1998)*.
- Tumblin, J. and Turk, G. Low curvature image simplifiers (LCIS): A boundary hierarchy for detail-preserving contrast reduction. In *Proc. SIGGRAPH* (1999).
- Vuylsteke, P. and Schoeters, E.P. Multiscale image contrast amplification (MUSICA). In *Proc. SPIE*, Volume 2167 (1994).
- Witkin, A. Scale-space filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, Volume 2 (Karlsruhe, Federal Republic of Germany (a.k.a. West Germany), 1983).
- Witkin, A., Terzopoulos, D., and Kass, M. Signal matching through scale space. *Int. J. Comput. Vision* 1 (1987), 2.
- Xu, L., Lu, C., Xu, Y., and Jia, J. Image smoothing via LO gradient minimization. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 30 (2011), 5.
- Xu, L., Yan, Q., Xia, Y., and Jia, J. Structure extraction from texture via relative total variation. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 31 (2012), 5.

**Sylvain Paris** (sparis@adobe.com), Adobe Research.

**Jan Kautz** (j.kautz@ucl.ac.uk), University College London.

**Samuel W. Hasinoff** (hasinoff@google.com), Google Inc.

Images credits: Martin Čadik, Paul Debevec, Frédéric Drago, Frédo Durand, Mark Fairchild, Dani Lischinski, Byong Mok Oh, Erik Reinhard, and Gregory J. Ward.



Watch the authors discuss this work in this exclusive *Communications* video.

# CAREERS

## The Ohio State University Department of Integrated Systems Engineering Tenure-Track Faculty, All Ranks

The Department of Integrated Systems (ISE) at The Ohio State University, invites applications for a tenure-track position focusing on visual analytics and sensemaking, human-computer interaction, data mining and the design of cognitive tools to support data analytics. All faculty ranks will be considered.

View full posting at:

<https://discovery.osu.edu/career-opportunities/open-positions/data-visualization-for-decision-making-assistant-or-associate.html>

## Tufts University Full-Time Computer Science Lecturer or Visiting Faculty

The Department of Computer Science in the School of Engineering at Tufts University invites applications for a full-time, non-tenure track Lecturer or Visiting Faculty beginning in September 2015.

We are seeking an engaged individual committed to excellent teaching, student mentoring, academic advising, and curriculum development. Applicants are expected to teach advanced courses in their areas of expertise. Applicants must hold a PhD in Computer Science or closely related field at time of appointment, or have a solid track record of classroom instruction and curricular innovation. The initial appointment is for two years with possibility of longer contracts.

The Department of Computer Science has grown tremendously in the past decade in faculty and student size and in research expenditures. Located in the Boston area, the department benefits from outstanding undergraduate and graduate students, collaborative faculty, and cross-disciplinary research and educational opportunities. Tufts University is one of the smallest universities ranked as a Research 1 university, and it offers the best of a liberal arts college atmosphere, coupled with the intellectual and technological resources of a major research university. Tufts University supports and encourages a culture of interdisciplinary research and there are numerous such opportunities within the School of Engineering, the School of Arts and Sciences, and through graduate and professional schools.

Tufts' School of Engineering distinguishes itself by the interdisciplinary focus and integrative nature of its engineering education within the intellectually rich environment of a research university. Located only six miles from historic downtown Boston, faculty members on the Tufts Medford/Somerville campus have extensive opportunities for academic and industrial collaboration as well as participation in the rich intellectual life of the area. The School of Engineering is in the midst of a period of exciting growth that has seen recruitment of outstanding new faculty, a quadrupling of funded research over the last ten years, addition of new laboratory space, an emphasis on building diversity in engineering, and major curricular initiatives at both the undergraduate and graduate levels.

Review of applications will begin February 1, 2015 and continue until the position is filled.

Application materials should be submitted online through Academic Jobs Online (AJO) at <https://academicjobsonline.org/ajo/jobs/5317>

For more information about the department, the positions, and the application procedure please visit <http://www.cs.tufts.edu/>. Inquiries should be emailed to [cssearch2@cs.tufts.edu](mailto:cssearch2@cs.tufts.edu).



## Computer Science – Department Head

The Department of Computer Science at Virginia Tech seeks applications from creative and visionary leaders for the position of Department Head. The Department Head's principal responsibility is to provide leadership and management of the department's programs, faculty, staff, and students. This entails leadership of departmental programs and administrative responsibility for planning, fiscal management, human resources, and communication within the department. The Department Head is expected to advance the research and teaching missions of this prominent department, nurture interdisciplinary collaborations, and work to achieve strategic goals in both the department and university. The successful candidate will be located at the Blacksburg, VA campus and lead a department with faculty there and in the National Capital Region campus ([www.ncr.vt.edu](http://www.ncr.vt.edu)). Faculty in NCR are located in Falls Church, VA as well as in the Virginia Tech Research Center ([www.ncr.vt.edu/arlington](http://www.ncr.vt.edu/arlington)) in Arlington, VA.

Candidates should have a Doctoral degree in Computer Science or a closely related field; demonstrated intellectual leadership and administrative skills in an academic/university environment or equivalent; a clear vision for the future of computing as a discipline; ability to communicate effectively, concisely, and clearly at all levels; dedication to the instructional mission of the university; interest in the development and expansion of sponsored research programs; an established record of professional activities and leadership in professional organizations; strong interpersonal skills; experience in enhancing the representation and success of underrepresented populations; and credentials commensurate with appointment as full professor with tenure in the department.

The Department has 37 research oriented tenure-track faculty and ~10 postdocs/research faculty. There are 12 NSF/DOE CAREER awardees in the department. Research expenditures for FY2014 were \$334 thousand per tenure-track faculty member (i.e., a total of \$12.2 million); total research funding at the beginning of FY2015 was \$42.8 million. Research strengths and several world-class centers in the department span human-computer interaction, high-performance computing, computational biology and bioinformatics, software engineering, data analytics, and computer science education. BS, MS, and PhD degrees are offered, with a growing enrollment of over 610 undergraduate majors (14% women) and over 225 PhD/MS students. In 2010, CS@VT was ranked 5th in the country in recruiting quality of CS undergrads by the *Wall Street Journal*. The Department is in the College of Engineering, the premier engineering school in the Commonwealth of Virginia, whose undergraduate program was ranked 8th and graduate program was ranked 12th among public engineering schools in 2014 by *U.S. News & World Report*.

Applications should include a curriculum vitae, a cover letter, a vision statement, a statement of leadership style and experience, and contact information for at least five individuals providing references. References will only be contacted for those candidates who are selected for the short list/phone interviews. Applications must be submitted online to <http://jobs.vt.edu> for posting **TR0140155**. Inquiries should be directed to Dr. Dennis Kafura, Search Committee Chair ( [kafura@cs.vt.edu](mailto:kafura@cs.vt.edu), 540.231.5568).

Applicant screening will begin on February 1, 2015 and continue until the position is filled. Early applications are encouraged. We welcome applications from women or minorities. Salary for suitably qualified applicants is competitive and commensurate with experience. Selected candidates must pass a criminal background check prior to employment.

**About Blacksburg:** Blacksburg is consistently ranked among the country's best places to live and raise a family (<http://www.liveinblacksburg.com/>). Educational and economic information, crime rates, amenities, air quality, and diversity are typical factors considered in the nationwide ranking. Blacksburg is a high-tech hub located in a scenic and vibrant community in the New River Valley between Alleghany and Blue Ridge Mountains. The town is proximal to state parks, trails, and other regional attractions of Southwest Virginia, renowned for their history and natural beauty. Virginia Tech has been recognized as a Tree Campus USA from the Arbor Day Foundation for its dedication to campus forestry management and environmental stewardship.

*Virginia Tech is an AA/EEO employer; applications from members of underrepresented groups are especially encouraged.*

Tufts University is an Affirmative Action/Equal Opportunity employer. We are committed to increasing the diversity of our faculty. Members of underrepresented groups are strongly encouraged to apply.

**University of Central Florida CRCV**  
**UCF Center for Research in Computer Vision**  
*Assistant Professor*

CRCV is looking for multiple tenure-track faculty members in the Computer Vision area. Of particular interest are candidates with a strong track record of publications. CRCV will offer competitive salaries and start-up packages, along with a generous benefits package offered to employees at UCF.

Faculty hired at CRCV will be tenured in the Electrical Engineering & Computer Science department and will be required to teach a maximum of two courses per academic year and are expected to bring in substantial external research funding. In addition, Center faculty are expected to have a vigorous program of graduate student mentoring and are encouraged to involve undergraduates in their research.

Applicants must have a Ph.D. in an area appropriate to Computer Vision by the start of the appointment and a strong commitment to academic activities, including teaching, scholarly publications and sponsored research. Preferred applicants should have an exceptional record of scholarly research. In addition, successful candidates must be strongly effective teachers.

To submit an application, please go to: <http://www.jobswithucf.com/postings/34681>

Applicants must submit all required documents at the time of application which includes the following: Research Statement; Teaching Statement; Curriculum Vitae; and a list of at least three references with address, phone numbers and email address.

Applicants for this position will also be considered for position numbers 38406 and 37361.

UCF is an Equal Opportunity/Affirmative Action employer. Women and minorities are particularly encouraged to apply.

**University of the District of Columbia**  
**Department of Computer Science and**  
**Information Technology**  
*Assistant/Associate Professor*

The Department of Computer Science and Information Technology at the University of the District of Columbia seeks applications for one tenure-track position at the level of Assistant/Associate Professor beginning in August 2015. We welcome all candidates in all areas of Computer Science and Information Technology to apply. Applicants must hold a Ph.D. in Computer Science, IT, or closely related disciplines. We are particularly interested in candidates with research experiences in the following areas: networks, cyber-security, mobile computing, cloud computing, computer vision, robotics, artificial intelligence or operating systems.

Candidates who have strong practical expertise in Confidentiality, Integrity, and Availability in Technology, Policy & Practice, and Education & Awareness of information assurance and incorporating protection, detection, and reaction capabilities are encouraged to apply. Faculty duties include teaching undergraduate and graduate students, conducting high-quality research by collaborating closely with the department's established teams, participating in and developing externally funded research projects, and performing academic duties, university services, and professional services.

The University of the District of Columbia is a comprehensive urban land-grant institution and is classified as a Historically Black College and University. It is the only public university in the District of Columbia, the U.S. Capital.

Applicants should submit a CV with three references (names and contact information) and teaching & research statement. All applicants should submit required materials, in electronic formats through UDC website: Click Here to Apply (<http://udc.applicantstack.com/x/detail/a2hbyxfur3>) Reviews will continue until position is filled. The University of the District of Columbia is an Equal Opportunity/ Affirmative Action Employer.



**ACM**  
**Transactions on**  
**Reconfigurable**  
**Technology and**  
**Systems**

ACM Transactions on Reconfigurable Technology and Systems

SPECIAL SECTION ON THE 15TH INTERNATIONAL SYMPOSIUM ON FPGAs

Articles 1-12 pages	W. Bui, W. Lee	Introduction
Articles 13-24 pages	A. Dutt, M. Reagin	Guest Editorial
Articles 25-36 pages	T. Makhadmeh, M. Haid, T. Szymanski, M. Suda, T. Sakaguchi, T. Ishigaki	Specialized FPGAs: Design Freedom in FPGAs using Multiple Configurations
Articles 37-48 pages	S. Ghemawat, K. Skaragan	Statistical Analysis and Program Version-Aware Mapping and Area Assignment for FPGAs
Articles 49-60 pages	S. Lu, H. A. Thompson, G. Li, M. Suda, M. Reagin	A Clustering Compiler with a Reconfigurable Backend

Downloaded from [dl.acm.org](http://dl.acm.org)

ACM  
 Association for Computing Machinery  
 1515 Broadway, New York, NY 10019

◆ ◆ ◆ ◆ ◆

This quarterly publication is a peer-reviewed and archival journal that covers reconfigurable technology, systems, and applications on reconfigurable computers. Topics include all levels of reconfigurable system abstractions and all aspects of reconfigurable technology including platforms, programming environments and application successes.

◆ ◆ ◆ ◆ ◆

[www.acm.org/trets](http://www.acm.org/trets)  
[www.acm.org/subscribe](http://www.acm.org/subscribe)

ACM  
 Association for Computing Machinery



**ADVERTISING IN CAREER OPPORTUNITIES**

**How to Submit a Classified Line Ad: Send an e-mail to [acmm mediasales@acm.org](mailto:acmm mediasales@acm.org). Please include text, and indicate the issue/or issues where the ad will appear, and a contact name and number.**

**Estimates: An insertion order will then be e-mailed back to you. The ad will be typeset according to CACM guidelines. NO PROOFS can be sent. Classified line ads are NOT commissionable.**

**Rates: \$325.00 for six lines of text, 40 characters per line. \$32.50 for each additional line after the first six. The MINIMUM is six lines.**

**Deadlines: 20th of the month/2 months prior to issue date. For latest deadline info, please contact: [acmm mediasales@acm.org](mailto:acmm mediasales@acm.org)**

**Career Opportunities Online: Classified and recruitment display ads receive a free duplicate listing on our website at:**

<http://jobs.acm.org>

**Ads are listed for a period of 30 days.**

**For More Information Contact:**  
**ACM Media Sales**  
**at 212-626-0686 or**  
**[acmm mediasales@acm.org](mailto:acmm mediasales@acm.org)**

# Are you looking for your next IT job? Do you need Career Advice?

The **ACM Career & Job Center** offers ACM members a host of career-enhancing benefits:

- A **highly targeted focus** on job opportunities in the computing industry
- **Access to hundreds** of industry job postings
- Resume posting **keeping you connected** to the employment market while letting you maintain full control over your confidential information
- **Job Alert system** that notifies you of new opportunities matching your criteria
- **Career coaching** and guidance available from trained experts dedicated to your success
- **Free access** to a content library of the best career articles compiled from hundreds of sources, and much more!



Visit **ACM's Career & Job Center** at:  
<http://jobs.acm.org>



Association for  
Computing Machinery

Advancing Computing as a Science & Profession

The **ACM Career & Job Center** is the perfect place to begin searching for your next employment opportunity!

Visit today at <http://jobs.acm.org>



[CONTINUED FROM P. 96] Wirth had retired from ETH and I received an invitation to take on a chair. It took me a year to organize the transfer, because I had my company to deal with, and I couldn't just drop it overnight. And it's a long flight from Santa Barbara to Zurich. Nonetheless, I dived headfirst into the academic world.

**Yet you have remained involved with Eiffel Software.**

The trick for me has been not to develop a dual personality. I'm the same person whether I work as a professor or whether I'm devoting time to the company, and my research is still in the context and philosophy of Eiffel. If you want to survive as a company, you have to do what the customers want. In academia, you can play with crazy ideas and experiment without regard to what you can sell. You're not going to hit the scientific jackpot every time. But once in a while, you are ahead of the game and do things that you cannot do in a company if you are focused on the bottom line.

**What specific elements of EiffelStudio have evolved through your academic work?**

Eiffel offers you the ability to test programs automatically—and it really is completely automatic in the sense that everything is generated by the software, even the test cases. That started out in an academic context at ETH and was refined through several excellent Ph.D. theses.

The much more ambitious goal, which has been made possible by a succession of Ph.D. theses, is the idea of fully verified software. For a long time, that looked like a purely intellectual pursuit, but it's now becoming a reality. We call it EVE, the Eiffel Verification Environment. There's still a lot of work to be done, and we have benefited from tremendous advances in technology, in particular the Boogie tools from Microsoft Research. But it's becoming realistic to imagine that we can take large, real-life programs and prove that they're correct. And if they're not correct, we can also—that's another part of the EVE research—automatically suggest fixes.

**On the teaching side, you helped produce a MOOC (Massive Open Online Course) that enables students any-**

**“The trick for me has been not to develop a dual personality. I'm the same person whether I work as a professor or whether I'm devoting time to the company.”**

**where to take the introductory programming course offered at ETH.**

Coming from industry, the last thing I expected to do was to teach bright-eyed 19-year-olds the rudiments of programming, let alone in German, or my imitation of it. Yet it has been one of the most exciting things I have done at ETH. I started looking in depth at how we can teach programming today to kids who have been using smartphones and videogames all their lives, and need the competitive advantage of becoming true professionals.

I use Eiffel and Design by Contract right from the start. The experience resulted in a textbook, *Touch of Class*, and more recently I dived head-on into MOOCs, first producing a kind of skunkworks MOOC outside of any organizational structure at the initiative of my colleague Marco Piccioni. Now we have redone it in a completely official context and it's an edX offering. It applies the same pedagogical principles as our course and also benefits from our research on distributed software development; students can compile and run programming exercises online, as quizzes in the course, and see the results right away.

**You also recently published a book on Agile development methods with the subtitle “The Good, the Hype and the Ugly.”**

Usually, when you see a novel idea in software engineering, you can quickly recognize whether it's good or bad. What's special about Agile is that it's a mix of the best and the worst. My book is a cool-headed attempt to separate the wheat from the chaff.

**What are the best elements of Agile, in your opinion?**

Among the best is the idea of developing in short iterations of two to six weeks. This has profoundly transformed the software industry for the better, and no one develops software anymore by assembling a few groups who tackle different parts of the program and will see each other in six months.

Another example is what I call the closed-window rule, an absolutely brilliant idea—that when you have an iteration, you schedule a certain number of tasks, and absolutely no one, regardless of rank, is permitted to add anything during that iteration. This rule has a number of benefits. It stabilizes the whole process and prevents bosses and managers from interrupting the iteration. It also has the benefit of weeding out bad ideas, because many suggestions that seem brilliant don't look so good when you wake up sober the next day.

**And the worst?**

In the opposite camp, you have the general rejection of what's derisively called “big upfront anything”—big upfront requirements, big upfront design. The Agile credo is that you should start implementing part of the system right away and not engage in long, early phases of architecture or investigation. The Agile world has this phobia of not producing anything that's not deliverable to the customer. And as is so often the case with Agile ideas, there's a grain of truth in this rejection, because the customer does not need specifications. The customer needs results. But the idea that it's bad to spend an appropriate time at the beginning of the project to clarify the overall requirements and design is nonsense. You do need at some point to focus on the deliverables; but until then you should take all the time necessary to address the big issues of specification and design. I've seen projects fail miserably for blindly applying the Agile catechism: we're Agile, we don't need to stop and think, we just go ahead and code! Not surprisingly, what they code is junk and they have to redo it, but maybe at that point the money has run out and the customer has lost faith. □

Leah Hoffmann is a technology writer based in Piermont, NY.

© 2015 ACM 0001-0782/15/03 \$15.00

## Q&A

# Object Lessons

*The creator of the Eiffel programming language discusses his career in industry and academia, “Design by Contract,” and his views on Agile software development.*

**FRENCH-BORN COMPUTER SCIENTIST** Bertrand Meyer—best known as an early advocate of object-oriented programming techniques and creator of the programming language and environment Eiffel—has enjoyed a varied career in industry and academia. Currently a professor of software engineering at ETH Zurich, the Swiss Federal Institute of Technology, he also has worked at Électricité de France (EDF) and at the University of California, Santa Barbara, and he continues to serve as CEO and chief architect at the California-based company he founded in 1985, Eiffel Software.

**After receiving degrees from Stanford and the University of Nancy, you spent nearly 10 years in industry at Électricité de France (EDF). When did you begin working on Eiffel, the object-oriented language and environment that you continue to refine and develop?**

In 1983, I got to spend a sabbatical at the University of California, Santa Barbara. A Japanese company got excited about a structured editor we had developed, and in 1985, we decided to found a company that’s now called Eiffel Software.

We were looking for a programming language. We didn’t like what was available, so I designed a notation that became Eiffel. Initially, I didn’t pay much attention to it. But in 1986, we attended the first OOPSLA (Object-Oriented Programming, Systems, Languages, and Applications) conference in Portland, and that’s where we realized that what I thought obvious was



**Bertrand Meyer, professor of software engineering at ETH Zurich, and CEO and chief architect of California-based Eiffel Software.**

actually ahead of the curve. So coming back from Portland, we refocused the company on Eiffel.

**You are also known for the idea of “Design by Contract,” a method of assuring a program’s correctness by specifying the conditions that each element must satisfy; there are preconditions, which state what an operation expects, and postconditions, which state what an operation guarantees. Did that idea evolve in tandem with Eiffel?**

Yes. After reading the works of Hoare and Dijkstra, it seemed like a direct, practical application of everything

we knew about program correctness. Design by Contract is a transposition to software of concepts that everyone is familiar with. If I want to buy something from you, I have a certain set of obligations to satisfy, and on your side, you also have obligations. Your obligations map into my benefits and conversely. So, too, with preconditions and postconditions in software.

**Let’s talk about your move to ETH Zurich, where you’ve been since 2001.**

It was not planned, but it has worked out very nicely. In 2000, Niklaus [CONTINUED ON P. 95]

# Systor2015

The 8th ACM International Systems  
and Storage Conference

May 26 – 28 • Haifa, Israel

We invite you to submit original and innovative papers, covering all aspects of computer systems technology, such as file and storage technology; operating systems; distributed, parallel, and cloud systems; security; virtualization; and fault tolerance, reliability, and availability. SYSTOR 2015 accepts both full-length and short papers.

**Paper submission deadline: March 5, 2015**

#### Program committee chairs

Gernot Heiser, NICTA and UNSW,  
Australia

Idit Keidar, Technion

#### General chair

Dalit Naor, IBM Research

#### Posters chair

David Breitgand, IBM Research

#### Steering committee head

Michael Factor, IBM Research

#### Steering committee

Ethan Miller, University of California  
Santa Cruz

Liuba Shrira, Brandeis University

Dan Tsafir, Technion

Yaron Wolfsthal, IBM

Erez Zadok, Stony Brook University

[www.systor.org/2015/](http://www.systor.org/2015/)

Sponsored by



In cooperation with



Platinum sponsor



Gold sponsors



Sponsors



# Computing Reviews



Association for  
Computing Machinery

ThinkLoud

---

## BEST OF COMPUTING

### 19th Annual

---



BEST REVIEWS  
NOTABLE BOOKS & ARTICLES

---

April 2015 Online & Print

[www.computingreviews.com](http://www.computingreviews.com)