

The surprising usefulness of sloppy arithmetic

January 4 2011, By Larry Hardesty



Graphic: Christine Daniloff

Ask a computer to add 100 and 100, and its answer will be 200. But what if it sometimes answered 202, and sometimes 199, or any other number within about 1 percent of the correct answer?

Arithmetic circuits that returned such imprecise answers would be much smaller than those in today's computers. They would consume less power, and many more of them could fit on a single chip, greatly increasing the number of calculations it could perform at once. The question is how useful those imprecise calculations would be.

If early results of a research project at MIT are any indication, the answer is, surprisingly useful. About a year ago, Joseph Bates, an adjunct professor of [computer](#) science at Carnegie Mellon University, was giving a presentation at MIT and found himself talking to Deb Roy, a researcher at MIT's Media Lab. Three years earlier, before the birth of his son, Roy had outfitted his home with 11 video cameras and 14 microphones, intending to flesh out what he calls the “surprisingly incomplete and biased observational data” about human speech acquisition. Data about a child's interactions with both its caregivers and its environment could help confirm or refute a number of competing theories in developmental psychology. But combing through more than 100,000 hours of video for, say, every instance in which either a child or its caregivers says “ball,” together with all the child's interactions with actual balls, is a daunting task for human researchers and artificial-intelligence systems alike. Bates had designed a chip that could perform tens of thousands of simultaneous calculations using sloppy arithmetic and was looking for applications that leant themselves to it.

Roy and Bates knew that algorithms for processing visual data are often fairly error-prone: A system that identifies objects in static images, for instance, is considered good if it's right about half the time. Increasing a video-processing algorithm's margin of error ever so slightly, the researchers reasoned, probably wouldn't compromise its performance too badly. And if the payoff was the ability to do thousands of computations in parallel, Roy and his colleagues might be able to perform analyses of video data that they hadn't dreamt of before.

High tolerance

So in May 2010, with funding from the U.S. Office of Naval Research, Bates came to MIT as a visiting professor, working with Roy's group to determine whether video algorithms could be retooled to tolerate sloppy arithmetic. George Shaw, a graduate student in Roy's group, began by

evaluating an algorithm, commonly used in object-recognition systems, that distinguishes foreground and background elements in frames of video.

To simulate the effects of a chip with imprecise arithmetic circuits, Shaw rewrote the algorithm so that the results of all its numerical calculations were either raised or lowered by a randomly generated factor of between 0 and 1 percent. Then he compared its performance to that of the standard implementation of the algorithm. “The difference between the low-precision and the standard arithmetic was trivial,” Shaw says. “It was about 14 pixels out of a million, averaged over many, many frames of video.” “No human could see any of that,” Bates adds.

Of course, a really useful algorithm would have to do more than simply separate foregrounds and backgrounds in frames of video, and the researchers are exploring what tasks to tackle next. But Bates’ chip design looks to be particularly compatible with image and video processing. Although he hasn’t had the chip manufactured yet, Bates has used standard design software to verify that it will work as anticipated. Where current commercial computer chips often have four or even eight “cores,” or separate processing units, Bates’ chip has a thousand; since they don’t have to provide perfectly precise results, they’re much smaller than conventional cores.

But the chip has another notable idiosyncrasy. In most commercial chips, and even in many experimental chips with dozens of cores, any core can communicate with any other. But sending data across the breadth of a chip consumes much more time and energy than sending it locally. So in Bates’ chip, each core can communicate only with its immediate neighbors. That makes it much more efficient — a chip with 1,000 cores would really be 1,000 times faster than a conventional chip — but it also limits its use. Any computation that runs on the chip has to be easily divided into subtasks whose results have consequences mainly for small

clusters of related subtasks — those running on the adjacent cores.

On the grid

Fortunately, video processing seems to fit the bill. Digital images are just big blocks of pixels, which can be split into smaller blocks of pixels, each of which is assigned its own core. If the task is to, say, determine whether the image changes from frame to frame, each core need report only on its own block. The core associated with the top left corner of the image doesn't need to know what's happening in the bottom right corner.

Bates has identified a few other problems that his chip also handles well. One is a standard problem in computer science called “nearest-neighbor search,” in which you have a set of objects that can each be described by hundreds or thousands of criteria, and you want to find the one that best matches some sample. Another is [computer](#) analysis of protein folding, in which you need to calculate all the different ways in which the different parts of a long biological molecule could interact with each other.

Bob Colwell, who was the chief architect on several of Intel's Pentium processors and has been a private consultant since 2000, thinks that the most promising application of Bates' chip could be in human-computer interactions. “There's a lot of places where the machine does a lot of work on your behalf just to get information in and out of the machine suitable for a human being,” Colwell says. “If you put your hand on a mouse, and you move it a little bit, it really doesn't matter where exactly the mouse is, because you're in the loop. If you don't like where the cursor goes, you'll move it a little more. Real accuracy in the input is really not necessary.” A system that can tolerate inaccuracy in the input, Colwell, argues can also tolerate (some) inaccuracy in its calculations. The type of graphics processors found in most modern computers are another example, Colwell says, since they work furiously hard to

produce 3-D images that probably don't need to be rendered perfectly.

Bates stresses that his chip would work in conjunction with a standard processor, shouldering a few targeted but labor-intensive tasks, and Colwell says that, depending on how Bates' chip is constructed, there could be some difficulty in integrating it with existing technologies. But he doesn't see any of the technical problems as insurmountable. But "there's going to be a fair amount of people out in the world that as soon as you tell them I've got a facility in my new chip that gives sort-of wrong answers, that's what they're going to hear no matter how you describe it," he adds. "That's kind of a non-technical barrier, but it's real nonetheless."

This story is republished courtesy of MIT News (web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.

Provided by Massachusetts Institute of Technology

Citation: The surprising usefulness of sloppy arithmetic (2011, January 4) retrieved 27 April 2024 from <https://phys.org/news/2011-01-sloppy-arithmetic.html>

<p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p>
--