

Debug Your Hardware as You Would Your Software

Kathy Creath
Optineering, USA

Back in 1979 when I started doing research for my master's thesis as a senior undergrad at the University of Rochester, I noticed two basic types of theses that students were working on: hardware and theoretical. Most of my fellow students were focusing on experiments requiring hardware. These types of projects required being able to design, build, and test electronics to get your data. Film and oscilloscopes were standard lab staples. Since I didn't really want to solder circuits together, I first chose a topic that involved a vacuum system that we could never get to work for lack of funding to buy the necessary parts to fix it. While working on that, I had my first real "aha" in an optics lab. I was told to line something up using Brewster's angle. When I rotated the polarizer and saw that Brewster's angle really did change the polarization state, I thought WOW, this stuff they're teaching really works.

Once our group realized that we weren't going to fix the vacuum, I switched topics to characterize an infrared detector. This required me to do the soldering I hadn't initially wanted to do to build an electronic amplifier and filter for biasing the detector so we could get a signal out. Compared to the other project, it turned out to be very straightforward to take data. What wasn't straight-forward was figuring out why this detector was behaving as it was. This required adapting a theoretical framework to fit the observations. When fitting the data, I did some basic programming on a large tabletop HP calculator/plotter. That was the beginning of my seeing how software was going to take over.

When I went to work on my PhD at the University of Arizona in 1982 I saw I had a choice of electronics vs. software for processing images. Seeing the complex boxes of electronics being used in the optics shop for the interferometers they were using, I thought that software was the way to go. From the modeling I did for my master's

thesis, I saw the power of computation to crunch data. As you know, I chose to work on interferometry. I poked at a lot of different topics before Professor Jim Wyant dropped a draft of a book chapter on my desk on electronic speckle-pattern interferometry. Up to this point, ESPI had required very complex electronics to get fringe patterns and “analyze” them. He asked if I thought I could do it in software. I took a look at the book chapter and said “sure.” The rest (as they say) is history.

Since then, I’ve basically become known as an “algorithms” person. Yes, I certainly did a lot of that along the way. But when people from lots of different application areas within optics come to me as a consultant and ask for help solving a problem, the answers are usually a lot more basic than fixing an algorithm.

I have found over the past three decades that many of these projects have not had good starting points. Often, the optical system is not producing the best possible image, or the illumination is not uniform. It could also be that the camera being used hasn’t been calibrated or maybe someone simply forgot to turn off the automatic gain. The algorithms assume that you’re getting a one-to-one correspondence between the object and image. But so many little things can disrupt that conjugate relationship. If you don’t start with a good image, you’re not going to get a good measurement result. This is a basic tenet of optical metrology.

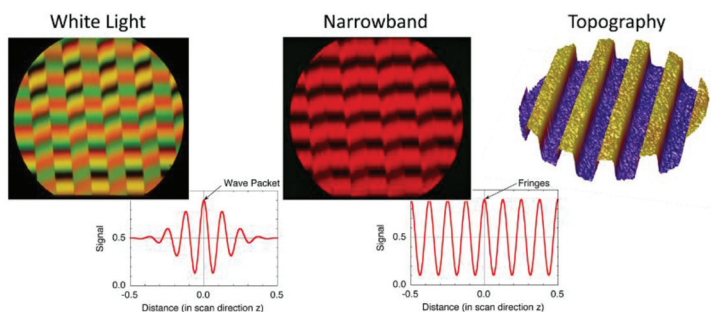
So often I find that the questions I get asked are at a very high level. However, the real question that needs to be addressed is usually at a low level. Often, it could be something as basic as setting up a way to align an optical component the same way on every system. But, what’s happening instead is that there are too many degrees of freedom and you end up with every system having a different problem. Those are tough problems to tease out.

Unfortunately, this type of problem solving and finding solutions is generally not taught in the classroom. I have found that those who've worked on real-world problems have a leg-up on those who haven't. A lot of this knowledge only comes from experience doing it and by learning from fixing our mistakes. Once you've had to fix something more than once, you probably will not do it the same way again, and you'll come up with a way to do it better.

How do we train our younger optical scientists to understand these things and to learn how to learn from what they do? I advocate more hands-on training—the more of it the better. Summer jobs are great. Undergraduate senior team projects and master's theses research more clearly reflect how we work in the real world than do lectures and exams. Lectures and preparing for exams help us know what material we need to understand and where to find it, as well as the answers to specific direct questions, but actually rotating the polarizer to get Brewster's angle is what we need to be able to recognize as the fix to debug systems we are working on.

My takeaway message is to not just look at the final outcome. Dig in and figure out places where you can check your results. Intermediate measurements and calculating results using known inputs can help you understand how to improve your hardware as well as your software. *Debug your hardware as you would your software.*

Ultimately, I think we should be thinking more about the questions, and how we mentor young scientists and engineers so they can benefit from what we have learned to pass on these skills. This, I argue, is more important than memorizing answers to specific problems that will be asked on an exam.



Deep grating on silicon substrate. (left) White light fringes taken through Mirau interference microscope. (middle) Narrowband fringes taken through Mirau interference microscope [images by K. Creath]. (right) Topographic map of grating generated from interferograms (images by J. Schmit).

Field Guide to

Optics Education: A Tribute to John Greivenkamp

SPIE Field Guides
Volume FG53

J. Scott Tyo, Series Editor
Eric Pepper, Guest Editor

SPIE PRESS
Bellingham, Washington USA