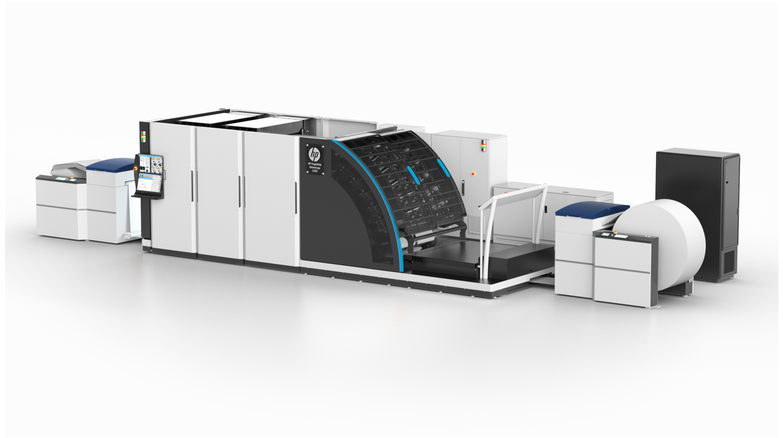
Faster Calibration Loading

Rye Gleason [1]

[1]*HP, San Diego, 6/21-9/16*

Overview

I worked on the HP PageWide Advantage 2200 printer, which needs to print a specific calibration image every time it's turned on. This image used to take 1 minute to load, and I decreased the load time to 30 seconds by creating a new, more efficient way to store the images.



Technical Challenges

The main technical challenge was how to properly design the new image format so it could store images with high compression, but still decode quickly. The overall concept for the new format was to encode large images as a set of smaller "snippet" images and an xml document detailing where to paste copies of the snippets, like a sprite graphics system. My contribution was coming up with the "patterns" used in the xml. One example is the "full tile" pattern, which takes a snippet and repeats it, end-to-end, down the full width or length of the image.

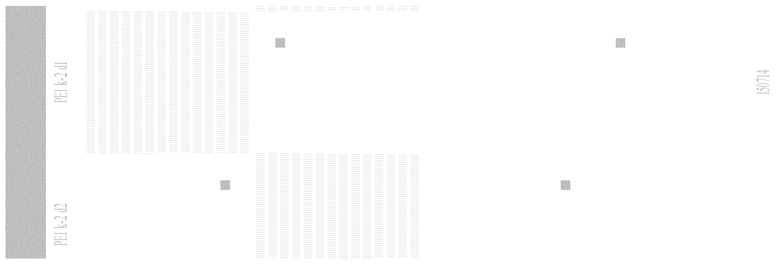
Another technical challenge was writing a method to draw large rectangles efficiently. The existing method filled in the rectangle pixel-by-pixel, which was quite slow when scaled up. The new version I wrote used `memcpy` to fill in entire rows at once. The challenge was that, due to compression in how the image was stored in memory, rows didn't necessarily match up with byte boundaries, so I had to write some tricky bit masking code to handle those cases correctly.

Lessons Learned

- The really valuable programming skill in industry is software design and architecturing, not writing super-optimized code. The most respected programmers on my team were the ones who designed the interfaces.
- My work is valuable, and I should be more confident about my skills and abilities! In the past I felt like any job offer I received was more than I deserved, but now I know that I can be picky.
- I want to do more "whiteboard work" than the typical software engineering job allows. After talking to my coworkers, especially the chemical engineers, I've decided to get an MS in robotics after I graduate in the hope of finding more satisfying work.
- I read an interesting quote about English pedagogy that said before 4th grade, you're learning to read, and then after, you're reading to learn. Working side-by-side with pros who had been coding for 30, 40 years made me realize that I've reached a similar point in programming, where I'm now coding to learn instead of learning to code.
- Interpersonally, I learned to relax more around my boss and mentors. When I started the internship I was very anxious about spending every moment I was on the clock being productive. As time went on, I realized that taking breaks to chat with coworkers or get a snack made me feel better, and without much impact on my output.

Ethics and Society

My work focused on speeding up the press's startup routine, which represents time that the printer (and its energy-intensive dryers) has to be powered on but not printing anything. By reducing wasted time, I also reduced wasted energy, helping the planet.



UCSD Connections

The most important UCSD course for this internship was CSE 110, Software Engineering. My team at HP used Agile processes just like the ones we learned about in class, and my project involved a lot of working with dependency injection, another 110 topic. Other helpful courses included:

- **CSE 120:** The speedups I got were mostly achieved by replacing filesystem reads with in-memory copies. CSE 120's overview of how hard drives and file systems work, and how they can be slow relative to CPU speeds, gave me the framework I needed to understand the work I was doing.
- **CSE 5A:** This class taught pointers very well, and just overall gave me the C++ skills I needed.
- **CSE 30:** Another C/C++ class - this one especially helped with memory management.
- **CSE 176E:** Working on quadcopters was a good introduction to embedded development, and it gave me practice working on the bit-level logic I needed for optimizing rectangle draws.

The Little Tips

- Git is really much better than SVN. My team used SVN but was trying to switch to git, and I could see why - the lack of local branches and commits held me back multiple times throughout the internship.
- My classes didn't really prepare me for working on large-scale codebases like the one at HP, but thankfully my extracurricular experience with high school robotics and contributing to various open-source projects did. If you're an underclassman and haven't worked on a bigger-than-class project, I'd highly recommend finding some cool open-source project on github and contributing!
- Before my internship, I was too focused on learning as many common libraries/frameworks as I could. When I got to work, I realized I didn't need to be, because I had to learn a ton of internal libraries anyways, so needing to learn one or two external ones wouldn't have mattered.
- Diversity in coding knowledge is important! Although my internship was mostly C++, I also wrote some python and bash scripts for myself that spend up my work enormously, and would have been way more of a pain to write in C++.
- Check out github copilot and `tmux`, great tools!