

Darrell Bird

Seattle, WA • Bird.Darrell@gmail.com • (206) 588-5623 • DarrellBird.com • <https://github.com/driblerrad>

CAREER SUMMARY

- Principal Scientist, hardware and software engineer with over 13 years of experience in IT product design, production, and distribution focusing on color science and display calibration.
- Laboratory director with 6 years experience managing a state of the art optical calibration lab.
- GC Chemist/Analyst with 5 years experience in petroleum hydrocarbon assay and identification.
- Highly adaptable to new environments and quick to master the latest hardware and software technologies.

EXPERIENCE

Principal Scientist & Software Engineer

Portrait Displays

January 2017 - February 2023, Seattle, WA

- As C++ Software Engineer, I've worked on a number of display calibration hardware and software packages, most notably Calman and the Aurora Color Engine, as well as customized calibration packages for specific clients such as HP and Dell.
- As Principal Scientist I created mathematical algorithms for color science and display metrology.
- I designed, built, and managed our state of the art calibration and display metrology lab.
- Inventor of United States Patent No 2020/0143767 A1 - System and Method for Color Calibration. The focus of this was to create a mathematical method for the calibration of a unique subset of typical tristimulus colorimetric sensors which included one or more additional active sensors such as infrared or physical temperature. The data from the additional sensor along with the patented smoothing functions allowed us to make inexpensive sensors like the AMS TCS3430, AS7261, AS73211 incredibly accurate and highly capable of display calibration/characterization. Patent Link: <https://patents.google.com/patent/US20200143767A1>
- Created Portrait's first prototype HDR Pattern Generator on an embedded Linux system. The company later developed this into a commercial product, the VideoForge PRO Pattern Generator supporting 4K DolbyVision, HDR-10, and HLG.
- macOS pattern generation, color management, and KVM using various Apple APIs including ColorSync, CoreGraphics, CoreFoundation, IOKit, and Metal rendering.
- Linux pattern generation, color management, and KVM using Xlib.

Laboratory Director

SpectraCal

January 2010 - January 2017, Seattle, WA

- I designed, built, and managed our state of the art NIST traceable optical calibration and display metrology lab.
- Created a revenue generating and highly profitable business model for the recalibration and enhancement of color sensors already sold and in the field. This allowed us to partner with companies such as BestBuy and FutureShop to land contracts for the enhancement and yearly recalibration of all GeekSquad hardware.
- Created customized solutions for several independent R&D companies with unique optical testing needs. A notable example was to build up an optical characterization of newly developed transparent solar panels for window installation.

Gas Chromatography Chemist / Analyst

TestAmerica - (North Creek Analytical)

January 2001 - December 2006, Seattle, WA

- Analysis and identification of petroleum hydrocarbons, VOCs, PCBs, Mercury, Pesticides and Herbicides in solvent extracted soil and water samples.

EDUCATION

Physics BS / Astronomy BS / Philosophy BA

University of Washington

January 2006 - August 2010, Seattle, WA

- Emphasis on optics, large scale gravitation, and epistemology related to the philosophy of science.

SKILLS

- C++ [most emphasis on the C++17 standard and cross-platform conformance for projects deployed to multiple systems]
- CMake [cross-platform build-system management for production software deployed to Windows, macOS, Linux]
- Linux / Darwin / Unix / WSL environments [I live on a terminal]
- Bash (sh, zsh, fish) / Powershell / CMD / Batch [I love bash and linux but Powershell is surprisingly powerful and I have grown to love it for Windows only tasks]
- AI / Large Language Models / Machine Learning -
 - I possess foundational skills in Python programming and have hands-on experience utilizing various open source multi-modal large language models from huggingface for basic text processing, image recognition (computer vision), image generation, voice recognition (speech-to-text), and voice generation (text-to-speech) to interact with language models similar to ChatGTP."
 - Created various AI apps and agents in Python with local database ingestion that implemented text-splitting, vector embedding, and database creation through both external API services and local LLMS running on a local OLLAMA server.
 - Utilized various AI frameworks (LangChain, OpenAI, CrewAI) to implement RAG algorithms for data retrieval and accuracy checking.
 - Proficiency in multiple RAG architectures including single shot RAG, Corrective RAG (CRAG), and iterative RAG.
 - Developed AI unit tests using pytest for both human and self evaluation of their accuracy.
 - Experience utilizing various open source multi-modal large language models from huggingface for basic text processing, image recognition (computer vision), image generation, voice recognition (speech-to-text), and voice generation (text-to-speech) to interact with language models similar to ChatGTP.
 - Basic language model training and tensor creation on small data sets using Pytorch using the c++ API.

Projects - Open Source

- [Chess Agent](#) - AI Chess Chatbot written in python configured to use local llms and an updatable local RAG database referencing a customizable set of chess documentation.
- [temporary file handler](#) - C++ library that provides a cross-platform way to create obfuscated temporary files that are automatically removed when the class object goes out of scope or the parent executable closes.
- [Rec_em](#) (Wreck'em) - Executable that creates a c++ header from an input file as an unsigned char array which can be used as an embedded resource in another binary.
- [libCMakeConfig](#) - A CMake script that generates a pre-configured CMakeList.txt and accompanying project files to generate a Modern CMake library project that can automatically be included in a super-project.