PhotonFit

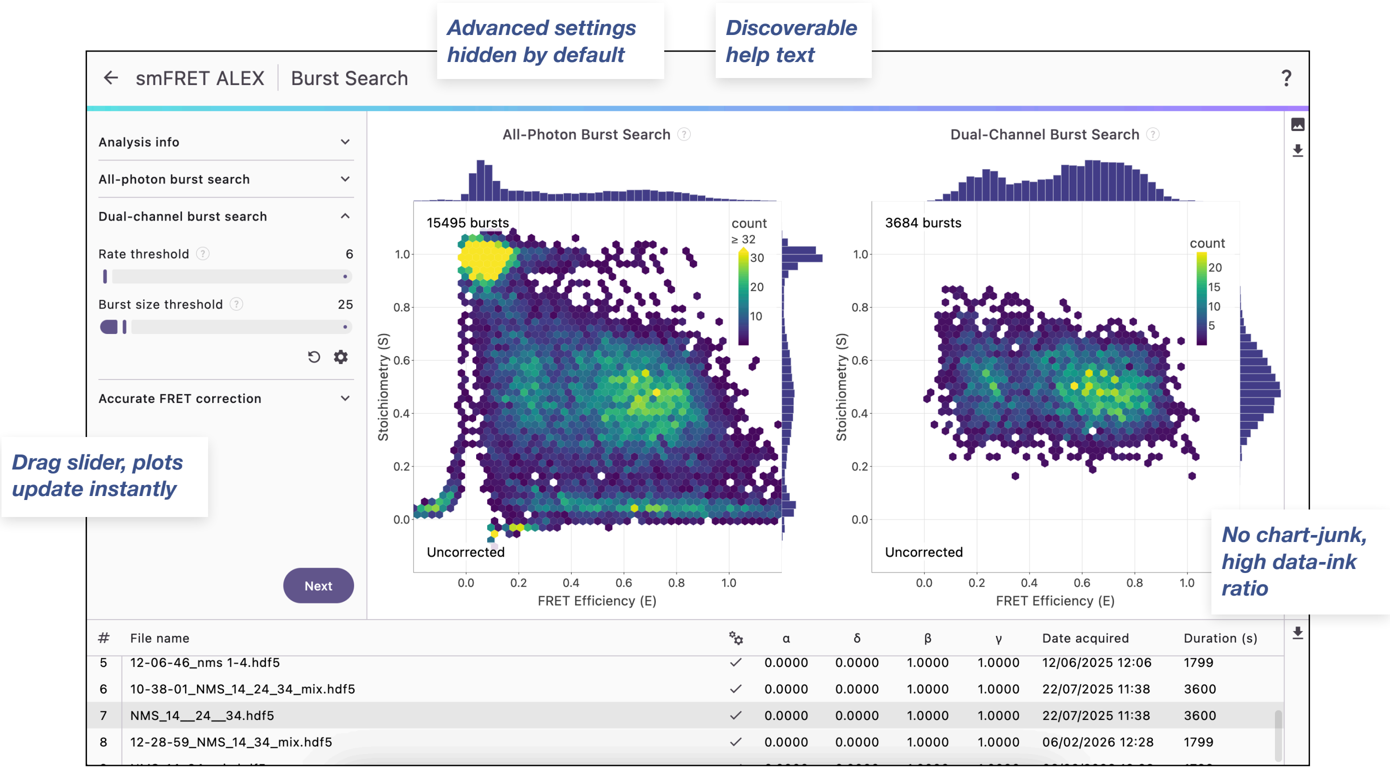
Exciting Instruments 2025
Stack: Kotlin, C++
Find out more: [website]

A biophysics data analysis app, for single-molecule fluorescence spectroscopy in academia and industry. End-to-end solo build.

When I joined Exciting Instruments (EI) we were selling cutting-edge single-molecule microscopes, operated via an app, but our customers had to analyse their data manually using scripts. I built a desktop app that lets them explore and analyse their data without writing any code.

Design

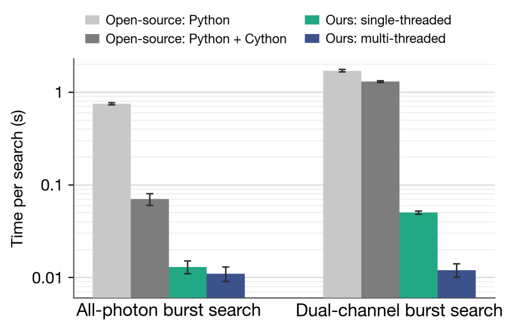
From EI's experience, most customers follow well-defined, linear workflows. The idea was to serve the majority by making these workflows smooth and intuitive and to serve the rest by providing data export throughout. I chose Kotlin and Compose Multiplatform for the frontend, for write-once multi-OS support, and chose performant C++ for the backend.



An example of a results screen in PhotonFit

Performance

'Burst search' is a core algorithm in PhotonFit. It involves scanning a photon stream for bursts of increased intensity, which mean that a fluorescent molecule is being observed. An open-source Python implementation took ~1s for a typical dataset — too slow for a responsive UI. My C++ implementation used data-oriented design and lock-free parallelism to run up to 100x faster (see right), so the settings can be smoothly driven by a slider.



Delivery

I worked with the science team to learn the domain and scope. Validation and stability came from automated testing, comparison against existing tools, and 'dogfooding' by colleagues. I shipped a release within 6 months, complete with 5 workflows, login, publication-quality figure and data export, a versioned file format with automatic migrations, and crash/usage telemetry.

"it's intuitive, easy-to-use, very quick … and gives me access to the data I need, without the complexity of python scripts and MATLAB code" — Senior Lecturer in biophysics, University of York

Multi-Wavelength Imaging (MWI)

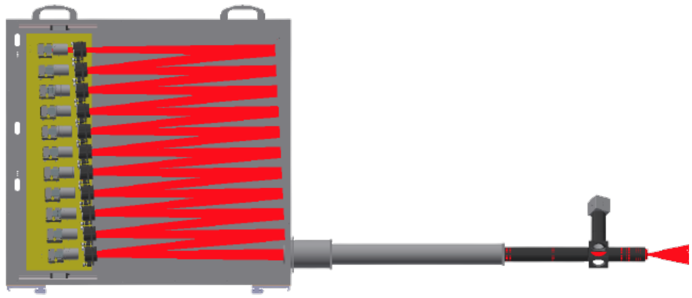
UKAEA 2020 - 2022
Stack: Python, C++, cameras, filters
Find out more: news release

A multi-camera scientific imaging system on the MAST-U tokamak. I installed and commissioned the MWI, wrote its control software, and operated it through the first experimental campaign.

The MWI began as a ~£1M collaboration between two UK universities and UKAEA — the UK's national fusion lab. It was designed to image the plasma simultaneously over many narrow wavelength bands, mapping atomic and molecular processes in 2-D to help solve the fusion heat exhaust problem. As the Responsible Officer at UKAEA, I took the MWI from a bench-top build to a workhorse instrument on a world-leading experiment.

Hardware

Turning raw video into scientific data requires calibration. I measured filter transmission and camera response, and performed radiometric and view calibrations. I integrated the MWI and its acquisition PC into MAST-U: power cables, data-link fibres, network fibres and trigger fibres — along with the paperwork that comes with engineering in a tightly-regulated environment.

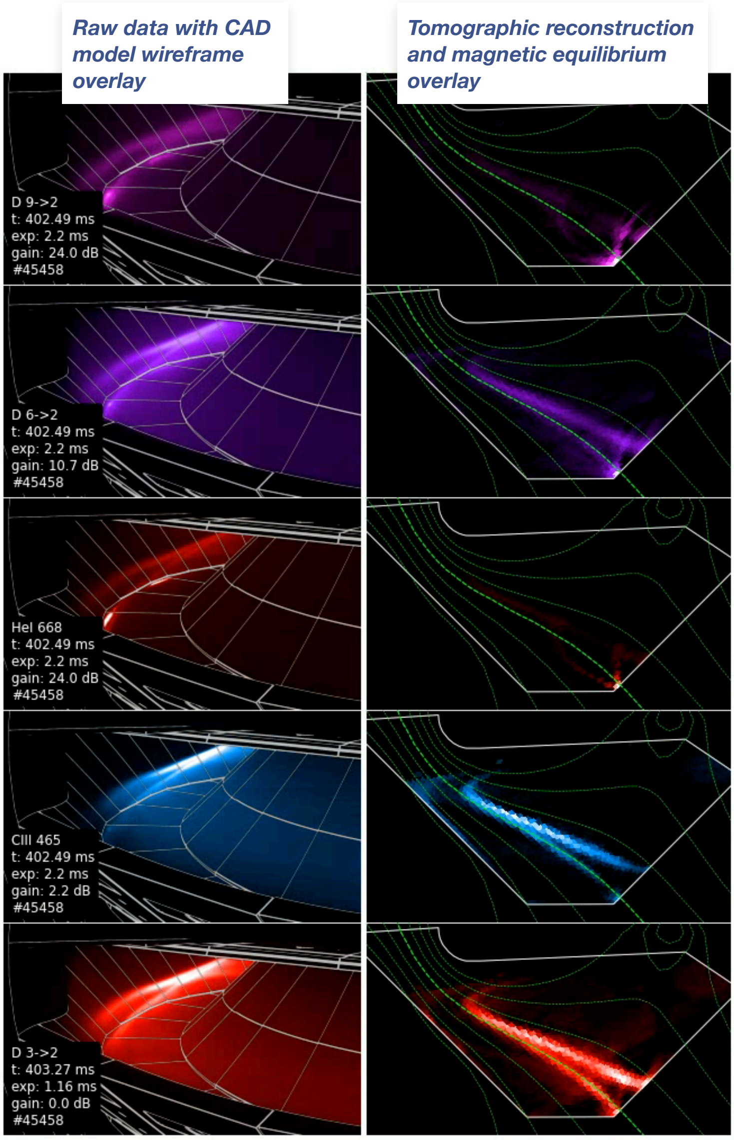


A schematic diagram of the MWI. Light path is shown in red.

Software

The MWI's core acquisition software came from a previous instrument. It used multithreaded C++ to drive ten cameras, their data aggregated over a PCIe switch and streamed to memory by DMA. I wrote a Python control layer on top for integration with MAST-U — reading experiment state, arming the FPGA trigger that fired the cameras on each plasma pulse, moving ~100 GB/day to storage, and generating the data visualisations shown right. I also wrote a suite of pre- and post-shot checks that flagged connection and frame drops.

The 11th channel was a polarised camera that the core acquisition software couldn't drive. I wrote independent C++ acquisition logic which I brought under the same Python control layer, and configured this camera to trigger off the same FPGA as the other cameras.



A single frame of visualised MWI data, showing the output of 5 cameras.

Impact

I operated MWI through the first MAST-U campaign. Its data has since fed into a range of physics investigations, including a 2-D characterisation of the MAST-U exhaust that I co-authored. It was later used as the real-time sensor behind the first demonstration of exhaust control in a novel, reactor-relevant tokamak configuration, published in Nature Energy.