# LiberTEM: Enhancement of pipeline to allow reshaping of n-dimensional datasets in the GUI

## About me


1. **Name:** Anand Baburajan (GitHub: @AnandBaburajan)
2. **University:** Government Engineering College, Palakkad
   **Program:** Bachelor of Technology in Computer Science and Engineering
   **Year:** 3rd year
   **Expected graduation date:** August 2021
3. **Email:** anandbaburajan@gmail.com
4. **Time zone:** Indian Standard Time (GMT+5:30)
5. **Resume:** resume.pdf


I've decided to apply at LiberTEM for GSoC this year because utilizing my skills in this project is a great opportunity for me to learn how a real-world project dealing with distributed processing of huge data is developed with good coding etiquettes and maintained with version control.

I also found contributing to the open-source scientific computing platform extremely fulfilling and gratifying. Working on the proposed project would give me a cross-sectional experience of LiberTEM's Python-based application side, NumPy-based data processing and TypeScript-React based GUI.

I look forward to contributing to LiberTEM for the foreseeable future.

# Code contribution

### Merged PRs

- Error handling for unsupported datasets in GUI (PR: 666)
- Initially set file's basename as dataset name (PR: 642)
- Exception for Direct I/O for RAW files on Windows (PR: 659)
- Updated marker to show picked pixel (PR: 639)
- Fixed buggy RAW Direct I/O checkbox (PR: 696)

### Issues opened

- Allow ds_path to be chosen from a dropdown in the HDF5 form in GUI (#692)
- RAW: Dataset with Direct I/O enabled fails to load on Windows (#658)

# Project information

- **Sub-org name**
  LiberTEM

- **Project abstract**
  4D STEM data in datasets can be stored with different shapes as different representations may be possible depending on the application. For example, 4D data is sometimes stored with a 3D shape as some software for crystallography expects data stored in 3D and currently, LiberTEM's GUI is limited to 4D datasets. Some datasets with synchronisation/acquisition problems also require specifying the offset. The proposed project aims to enhance LiberTEM's data pipeline by supporting reshaping of datasets in the GUI which would provide its users with more flexibility to load and perform analyses of multidimensional data.

- **Detailed description**

  **Goal**

After a dataset is loaded into the LiberTEM GUI, for each analysis, the user would have an option to specify the optional offset and the navigation and signal dimensions of the dataset.

**Implementation**

The following implementation plan is for the initial proposed design and as mentioned in the weekly timeline, I'll try to get a better understanding of the RAW format, find better design options and write prototype code for them. After discussing with the mentors, the most appropriate design would be implemented into LiberTEM.

Currently, only the RAW format allows specifying the scanning and detector dimensions for RAW files. But the dataset implementation of RAW format reads data from a memory-mapped file object and reshapes it in-place, unlike implementations of formats like HDF5, which read data directly from the dataset only while tile generation when the partitioning takes place. The RAW format also allows specifying invalid shapes which is useful in case of synchronization problems.

For specifying application-specific shapes to datasets, the reshaping must be done before partitioning. LiberTEM supports a variety of dataset formats and except HDF5 and K2IS, which use their own partition classes, all other formats depend on the Partition3D class for partitioning.

For prototyping, I tried to load 3D data from a HDF5 dataset into a Dask array and reshaped it into 4D before partitioning and it worked out fine. A Dask array was used as it allows parallel processing on data larger than RAM, but after discussing with mentors, I understood that LiberTEM's I/O layer is different from Dask arrays and isn't easily compatible, even though it works as a prototype.

For a general solution, the partition classes would be modified to generate slice objects to read reshaped data from the original datasets. Datasets will be initialized with the detected shape as usual. Some datasets also allow specifying a hard-coded 4D tileshape which would need to be changed to allow n-dimensional tileshapes.

On the GUI side, three textboxes for specifying offset, scanning and detector dimensions will be provided alongside the 'Add Analysis' button. Appropriate APIs

would be designed to send the new shape, tileshape and offset to the get_partitions function for reshaping before partitioning.

- **Weekly timeline**

  - **Before GSoC begins**
    - Gain better understanding of LiberTEM's codebase including the GUI and Python Dataset API
    - Understand and solve a number of smaller issues in LiberTEM
    - Learn basic STEM concepts and become familiar with Python libraries and modules used in the project for dataset handling
    - Get familiar with LiberTEM's UDFs and the structure and implementation of file formats supported by LiberTEM

  - **Community bonding, Week 1 and Week 2 (May 4 - June 15)**
    - Read various issues raised in the past related to data loading and processing to understand different user requirements and LiberTEM's use cases
    - Get better understanding of the dataset implementation of RAW format
    - Check issue #553 (not implemented yet) and discuss with mentors about implementing reshaping in the UDF interface
    - Find better system design options for implementing reshaping
    - Discuss the plan, design and feasibility of the options with mentors and write prototype code for each of them

  - **Week 3 and Week 4 (June 15 - June 29)**
    - Submit a report with the implementation plan and architecture of the most suitable reshaping design option
    - Begin implementing reshaping into LiberTEM

  - **Week 5, Week 6, Week 7 and Week 8 (June 29 - July 27)**
    - Submit 1st evaluation
    - Work on the GUI and design appropriate APIs
    - Continue working on implementation of reshaping

    - Write tests
    - Write documentation

- **Week 9 and Week 10 (July 27 - August 10)**
    - Submit 2nd evaluation
    - Complete any unfinished tasks
    - Debugging of the GUI and new features added
    - Write tests
    - Write documentation

- **Week 11 and Week 12 (August 10 - August 24)**
    - Implement reshaping in the UDF interface for #553
    - Complete any unfinished tasks
    - Write tests
    - Write documentation

- **Final week (August 24 - August 31)**
    - Submit the code, project summary and final evaluation

# Other commitments

- I will take my finals between May 11 and Jun 5. During that time, I'll put in not more than 30 hours per week on the proposed project. To compensate for the time spent for my finals, I'm spending time on the project before the official GSoC period begins. Except for my finals, I've no other obligations during the GSoC period and I'm willing to spend 48+ hours per week on the project.
- LiberTEM is the only organisation I'm applying to.