

CrystalGrowthTracker: A Python package to analyse crystal face advancement rates from time lapse synchrotron radiography

Joanna Leng ^{*}1, Jonathan H. Pickering [†]1, Sven L. M. Schroeder^{2,3,4,5}, and Gunjan Das^{2,3,4,5}

1 School of Computer Science, University of Leeds, Leeds, United Kingdom 2 School of Chemical and Process Engineering, University of Leeds, Leeds, United Kingdom 3 Diamond Light Source Ltd, Harwell Science and Innovation Campus, Harwell, United Kingdom 4 Research Complex at Harwell (RCaH), Rutherford Appleton Laboratory, Harwell, United Kingdom 5 EPSRC Centre for Innovative Manufacturing in Continuous Manufacturing and Advanced Crystallisation, University of Strathclyde, Glasgow, United Kingdom

DOI: [10.21105/joss.04333](https://doi.org/10.21105/joss.04333)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Lucy Whalley](#) 

Reviewers:

- [@CFGrote](#)
- [@KedoKudo](#)

Submitted: 28 January 2022

Published: 15 November 2022

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

X-ray synchrotron radiation allows the investigation of many physical processes in unprecedented ways. One application, important to the fine chemicals industry, is characterizing the early stages of crystallization. To aid chemical engineers working in this area we developed the CrystalGrowthTracker package. It allows crystals to be found in videos and their growth rates measured, see [Figure 1](#). We hope that this work can provide the basis for fully automated and 3D systems.

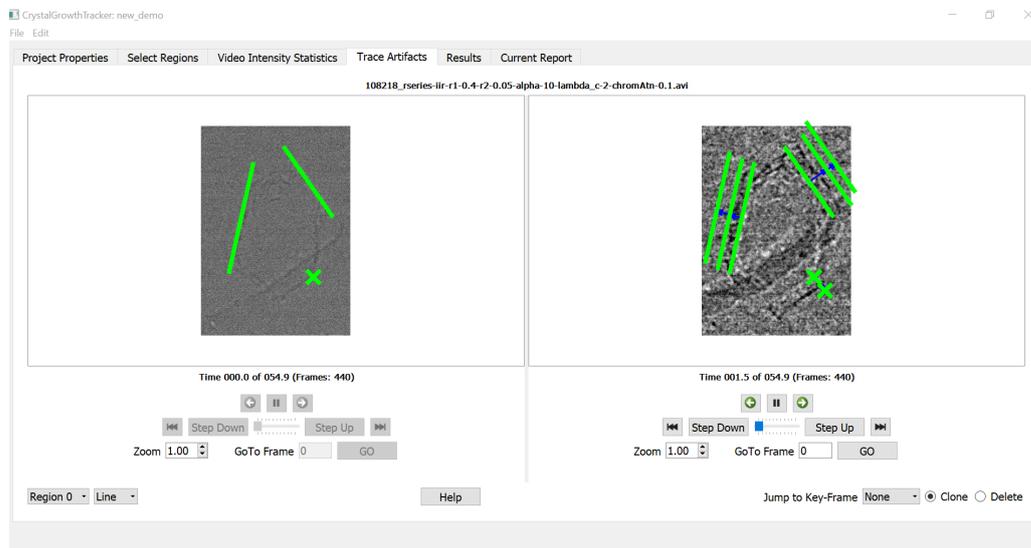


Figure 1: Screenshot showing annotation of a video. More screenshots are available in the project readme file.

*co-first author

†co-first author

Statement of Need

Much of the output of global fine chemicals industries consists of crystalline powders produced by precipitation from solution. Since the geometry of a crystal is the result of differential growth rates on different crystal planes, there is great commercial interest in studying the growth rates of crystal faces in the early stages of crystallization. One approach is to use the X-rays of synchrotron radiation ([Baruchel et al., 2013](#)) to produce 2D shadowgraphs of crystals precipitating onto a substrate.

Existing image analysis packages such as ImageJ, popular with scientific users, are designed for single images and does not handle video data well. For example, one of the authors, Gunjan Das, originally took eighteen months to annotate one video using ImageJ. Alternative video display and editing software is designed for entertainment and does not allow annotation and measurement of features, and some packages will not work with high resolution video.

Videos of the shadow graphs first need enhancement using packages such as Euler's Magnifier ([Wu et al., 2012](#)) and some statistical analysis of the raw video is desirable to find regions of interest. CrystalGrowthTracker has been developed to assist in the analysis of videos of the raw and enhanced videos. The package was developed by Leng and Pickering, with specifications provided by Schroeder. Das beta tested and presented on the project ([Das, 2021a](#)), ([Das, 2021b](#)). Details of the experiments can be found in ([Marathe et al., 2019](#)), the resulting videos were shared with the developers so this data analysis software could be developed.

Design

The obvious approach would be to use image analysis and machine learning. However the relatively noisy data, the limited number of data sets, and the need for verifiable results lead to a manual approach. User analysis of video requires the package to have a graphical user interface (GUI). The best way to co-ordinate a GUI with the computational engine that runs in the background is through the model-view-controller (MVC) software architecture (Krasner & Pope, 1988). Finally, users need to be able to download and run the package on any machine, so the package is developed using Python within the conda virtual environment and in the pip package management system.

The PyQt5 toolkit, from Riverbank Computing Limited ([Riverbank Computing Limited, n.d.](#)), is chosen for the GUI component as it runs on any machine, is feature rich, and implements cross thread message passing using slots and signals. The latter is necessary for the complex functionality needed to play the video forwards, backwards and jump to any frame/time with and without markers and zoom.

However the Qt video player does not allow the low level access to the video stream required for this analysis, so the ffmpeg package ([Bellard, n.d.](#)) is used to access the video files in the computational engine. A custom viewer is constructed with Qt which allows the video to be viewed in high resolution, with zoom, so markers and annotations can be viewed across the video and display how features change over time. No effort has been made to play at the correct number of frames per second, as preserving the highest image resolution is considered paramount.

The user is able to add a marker to an image feature in a single key-frame of the video, then advance to a new key-frame and drag a clone of the marker to the feature's new locations. The markers are graphics items augmented to hold their frame numbers and the identity of their parent. Calculating the speeds of motion consists of ordering the chains of cloned graphics items by frame number, finding the displacements in pixel coordinates, converting the pixel distances to real distances using the scaling factor. The pixels are assumed to be square as individual x-ray sensors in the array are square. Frame number intervals are converted to times

and speeds calculated.

Functionality

Videos documenting the [install](#), [uninstall](#) and [operation](#) of the package are available on YouTube. Sample video data is available via [Zenodo](#).

The package allows the user to load two videos, raw and enhanced, and provides the following functionality:

1. Easy install and uninstall plus availability of a translation package.
2. Analyse and display intensity statistics of the raw video, to assist selection of times of interest.
3. Play enhanced video and allow the selection of regions of interest.
4. In regions of interest allow features to be marked up on a given frame. The markers being, lines for crystal faces, or points for other features.
5. Change the frame of a region of interest and move the markers to follow the underlying feature.
6. Use time and space calibration data to calculate the true speed of the marker motions.
7. Store the above data in an open human readable format and present a report in HTML.

Project data and results are saved in comma-separated value (CSV) files using the Python csv module. CSV is a text format, which can be viewed and edited in text editors or spreadsheets, such as Excel. The results of a project can be saved in a HTML or PDF report, which included tables, graphs and annotated video frames.

For Developers

Some readers may wish to extend this package, for them we provide some further information on testing, source code style and documentation.

A test suite is provided, because the Python unittest module cannot run the ffmpeg subprocess, a separate test was developed for these interactions. The Qt QTest object is used to simulate IO events, and capture the resultant signals for testing the GUI. The tests can be executed by adding the '-t' flag to the main cgt command line. The Python script run_test executes all tests and saves the results to a CSV file.

Doxygen is used to generate documentation from source code comments. The code has been developed using the Pylint static code analysis tool, with a custom-made runner script. The script runs Pylint on all files, with the output displayed in the shell tool window or saved to CSV file.

Availability

The software can be obtained from [GitHub](#) under an Apache License.

Acknowledgement

The authors acknowledge I13-2 beamline at DLS for the beamtime (MT20984-1) and Research Complex at Harwell for generating the test data. Funding was provided by: the Future Continuous Manufacturing and Advanced Crystallization (CMAC) Hub (EPSRC Grant EP/P006965/1); G. Das's PhD studentship from the University of Leeds; and Joanna Leng's EPSRC funded RSE Fellowship (EP/R025819/1).

References

- Baruchel, J., Di Michiel, M., Lafford, T., Lhuissier, P., Meyssonnier, J., Nguyen-Thi, H., Philip, A., Pernot, P., Salvo, L., & Scheel, M. (2013). Synchrotron x-ray imaging for crystal growth studies. *Comptes Rendus Physique*, 14, 208–220. <https://doi.org/10.1016/j.crhy.2012.10.010>
- Bellard, F. (n.d.). *FFmpeg*. Opensource project web site, last visited 20 June 2022. <https://www.ffmpeg.org/>
- Das, G. (2021a). *Time-resolved x-ray phase contrast imaging of continuous anti-solvent crystallisation*. BCA-BACG Joint Spring Meeting 2021; Online event. <https://www.crystallography.org.uk/spring-meetings/>
- Das, G. (2021b). *Time-resolved x-ray phase-contrast imaging of continuous anti-solvent crystallisation*. ISIC 21 - 21st International Symposium on Industrial Crystallization; Virtual Event.
- Marathe, S., Storm, M., Kuppilli, V. S. C., Harrison, R., Das, G., Schroeder, S. L. M., Cipiccia, S., Döring, F., David, C., & Rau, C. (2019). Development of synchrotron pink beam x-ray grating interferometer at the Diamond Light source I13-2 beamline. In B. Müller & G. Wang (Eds.), *Developments in x-ray tomography XII* (Vol. 11113, pp. 269–275). International Society for Optics; Photonics; SPIE. <https://doi.org/10.1117/12.2530698>
- Riverbank Computing Limited. (n.d.). *Riverbank computing limited*. Corporate web site, last visited 20 June 2022. <https://riverbankcomputing.com/>
- Wu, H.-Y., Rubinstein, M., Shih, E., Guttag, J., Durand, F., & Freeman, W. T. (2012). Eulerian video magnification for revealing subtle changes in the world. *ACM Transactions on Graphics (Proc. SIGGRAPH 2012)*, 31(4). <https://doi.org/10.1145/2185520.2185561>