

Index EBSDs from the NORDIF software using AstroEBSD

Department of Materials Science and Engineering, NTNU

PhD Candidate Håkon Wiik Ånes

March 14, 2019

Description

A tutorial on how to index electron backscatter diffraction patterns (EBSDs) using the open-source indexing package AstroEBSD [1]. The software is developed by Britton and co-workers at the Experimental Micromechanical Characterisation Research Group at Imperial College London¹, is implemented in Matlab, and is available on GitHub [2]. As of March 2019, the version of AstroEBSD in the main repository can only read patterns from the Bruker eSprit software. However, I wrote a ‘plugin’ for reading patterns from the NORDIF .dat binary file, available in my fork of AstroEBSD [3]. For questions regarding this tutorial or the NORDIF plugin, send me an email at hakon.w.anes@ntnu.no or open an issue in my fork respectively, while questions regarding AstroEBSD should be discussed in an issue in the main repo.

Contents

1 Prerequisites and setup	1
2 Overview of example data set	1
3 Index example data set	2
3.1 Pattern processing	2
3.2 Radon transformation and peak localisation	2
3.3 Pattern centre	3
4 Inspect results with MTEX	3

1 Prerequisites and setup

Download Matlab and the necessary toolboxes as described here in the AstroEBSD wiki². Then, download the branch named ‘nordif-reader’ from my fork [3] and place the unzipped directory in Matlab’s initial working directory (can be set by going to Preferences > General > Initial working folder). Now, instead of launching the main version of AstroEBSD, we need to launch the version from my fork with the NORDIF plugin. Check that everything is as expected by opening Matlab and enter in the command window:

```
>> start_AstroEBSD
>> help Input_Deck_Nordif
```

Do some troubleshooting on your own if errors occur, and if you cannot fix the problem, contact me (email above).

2 Overview of example data set

The example data set of 400 patterns is from a homogenised, 90% cold rolled aluminium alloy containing 0.5wt.%Fe and 0.15wt.%Si and annealed at 325 °C to full recrystallisation. It was acquired on a Hitachi SU-6600 with the NORDIF UF1100 detector using an exposure time of 1350 µs and a binning of 4, resulting in patterns with a (120 × 120) pixel resolution. An example pattern is shown in Fig. 1.

¹Micromechanical Characterisation Research Group webpage: <https://www.expmicromech.com> (visited 03/14/2019).

²AstroEBSD wiki: <https://github.com/benjaminbritton/AstroEBSD/wiki> (visited 03/14/2019).

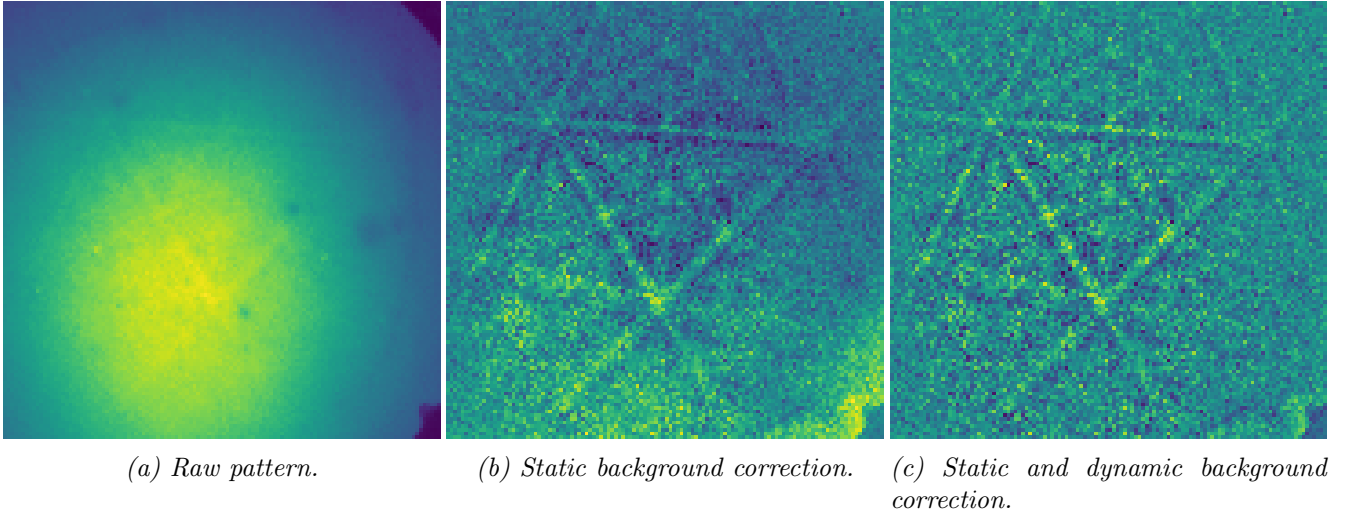


Figure 1: Example pattern from the data set showing the image preprocessing done by AstroEBSD before Radon transformation (and subsequent Radon peak hunt and indexing).

3 Index example data set

All interaction with AstroEBSD is done using the input file ‘decks/Input_Deck_Nordif.m’. We need to tell AstroEBSD the pattern and settings file locations by setting the variables:

- `Astro_FP`: Full path to AstroEBSD.
- `data_FP`: Full path to data, i.e. ‘Pattern.dat’ and ‘Setting.txt’.
- `InputUser.EBSD_File`: Name of file with patterns, in our case ‘Pattern.dat’.
- `InputUser.SettingFile`: Name of settings file, in our case ‘Setting.txt’.
- `bg_file`: Full path to background acquisition pattern.

Before indexing all patterns, we need to find the optimal parameters for (i) pattern processing, (ii) the search for peaks in the Radon Transform and (iii) pattern centre. A screenshot of AstroEBSD’s graphical user interface (GUI) for optimisation of these parameters before indexing is shown in Fig. 2. You should try to vary these parameters in turn to see the effect on the indexing result. Indexing of the full data set starts after the GUI is closed.

3.1. Pattern processing. In the case of the patterns in the example data set (see Fig. 1a) both a static and dynamic background correction is needed for AstroEBSD to find peaks in the radon transform. We therefore subtract both a background pattern, acquired on the microscope (`bg_file`), and a Gaussian blurred pattern, unique to each pattern. The standard deviation of the Gaussian blurring kernel is set in the `gfilt_s` variable. A radius mask should be applied before radon transforming (see [1] for more details), while the patterns contain no hot (dead) pixels and a resize is unnecessary due to the small pattern size.

3.2. Radon transformation and peak localisation. To locate the bands within the EBSD, *bands* of raised intensity are transformed into *points* of raised intensity. A parallel projection of integration of the pattern perpendicular to a sampling line, namely the Radon transformation, is used. It is directly related to the Hough transform used in the commercial EDAX TexSem Laboratory (TSL) Data Collection software, so the transform parameters are similar to TSL’s parameters. The number of sampling lines, i.e. the ρ steps, is equal to the greatest pattern dimension (here $\rho = 120$), while the number of sampling angles is typically in steps of $\Delta\theta = 1^\circ$. See [1] for more details.

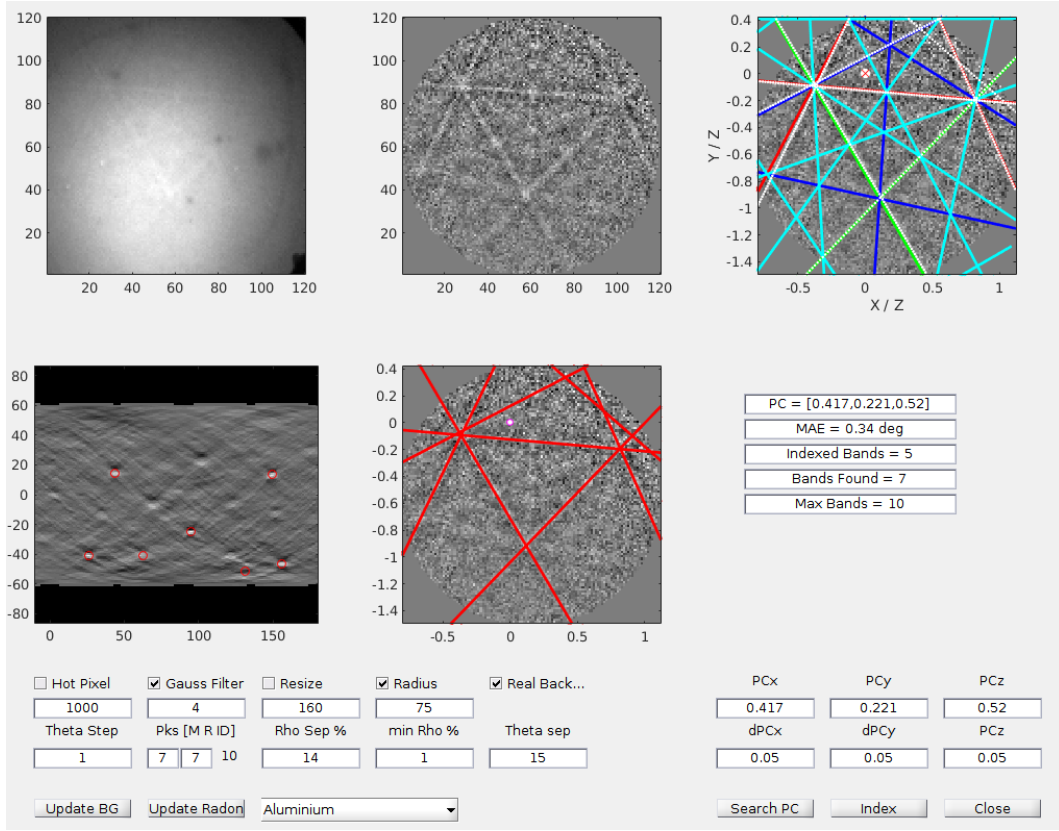


Figure 2: AstroEBSD's graphical user interface for pattern analysis settings optimisation. Images from upper left to bottom right: (i) raw pattern, (ii) pattern after background correction, (iii) overlay of (striped lines) detected bands and (full coloured lines) bands suggested from best matching crystal orientation, (iv) radon transform of background corrected pattern and (v) overlay of detected bands.

Peaks in the radon transform are localised by a filter which detects edges confined within a window. The window dimensions is set by us in `theta_search_pix` (width) and `rho_search_per` (height). The number of peaks to search for is set in `num_peak` and the maximum number of peaks to use in indexing is set in `max_peaks`. Note that in TSL the *minimum* number of peaks to use in indexing is set. The minimum separation of each peak's highest and lowest point, i.e. the band edges, can be set in `min_peak_width`.

3.3. Pattern centre. A correct pattern centre is needed for the apparent interplanar angles in the pattern to be correct. A suitable range within which the pattern centre optimisation algorithm is allowed to change the suggested pattern centre can also be set.

4 Inspect results with MTEX

After indexing AstroEBSD displays a summary of the results, showing e.g. indexing quality metrics and inverse pole figures (IPFs), examples of which are shown in Fig. 3.

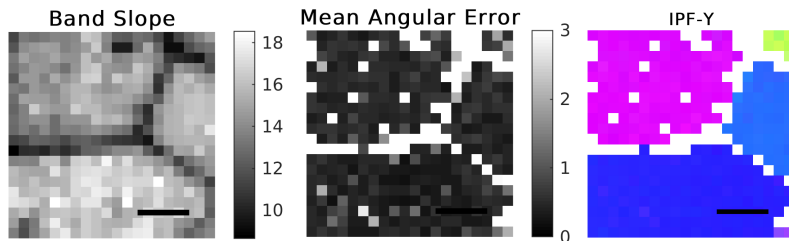


Figure 3: Parts of the indexing output from AstroEBSD for our example data set. The scale bar is 5 μm .

All output data from AstroEBSD is stored in an output .mat file found in the output directory. I've made a script `astroebdsd2mtex.m`³ to export sample position (x, y), Euler angles, pattern quality, pattern slope, mean angular error, band number and phase per sample position to a .dat file which can be read by the open source and powerful Matlab toolbox MTEX⁴. With MTEX you can analyse and model crystallographic textures by means of EBSD or pole figure data. In the same GitHub repository as `astroebdsd2mtex.m` I've added a script showing an example usage of it together with MTEX. The workflow in the script is as follows:

1. Import data.
2. Filter orientations based on mean angular error.
3. Reconstruct grains, remove smallest grains, reconstruct grains again and finally lightly smooth the grain boundaries.
4. Lightly smooth orientation data and fill in filtered out orientations.
5. Plot band slope, mean angular error and inverse pole figure map, all with grain boundaries overlaid.

The results from MTEX is shown in Fig. 4.

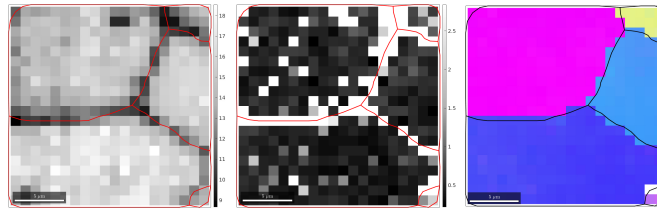


Figure 4: Same output from MTEX as from AstroEBSD in Fig. 3, with grain boundaries overlaid. The orientation data processing is explained in the text above.

References

- [1] Thomas Benjamin Britton et al. “AstroEBSD: exploring new space in pattern indexing with methods launched from an astronomical approach”. In: *Journal of Applied Crystallography* 51 (2018), pp. 1–10. ISSN: 1600-5767. DOI: [10.1107/S1600576718010373](https://doi.org/10.1107/S1600576718010373). arXiv: [1804.02602](https://arxiv.org/abs/1804.02602).
- [2] Thomas Benjamin Britton. *AstroEBSD*. 2019. URL: <https://github.com/benjaminbritton/AstroEBSD> (visited on 01/21/2019).
- [3] Håkon Wiik Ånes. *Fork of AstroEBSD*. 2019. URL: <https://github.com/hwagit/AstroEBSD/tree/nordif-reader> (visited on 03/13/2019).

³`astroebdsd2mtex` GitHub repository: <https://github.com/hwagit/mtex-snippets/tree/master/astroebdsd2mtex>.

⁴MTEX homepage: <https://mtex-toolbox.github.io/documentation.html> (visited on 03/14/2019).