holoaverage Documentation

Release 1.1.8

Tore Niermann

Jun 22, 2023

Contents:

CHAPTER 1

About

Holoaverage is a Python program for the reconstruction and averaging of series of off-axis electron holograms recorded in a transmission electron microscope. The averaging is performed iteratively, such that instabilities of the microscope, like specimen and biprism drifts, can be tracked and corrected between consecutive exposures.

The program is written and maintained by Tore Niermann (email: tore.niermann@tu-berlin.de).

Details on the usage can be found in the documentation. The documentation can be found at

https://holoaverage.readthedocs.io

1.1 Citation

When you use the program in your research work, please cite the paper describing the details of the averaging method. The details can be found in

T. Niermann and M. Lehmann
Averaging scheme for atomic resolution off-axis electron holograms
Micron 63 (2014) 28-34
doi: 10.1016/j.micron.2014.01.008

The BibTeX entry for the paper is:

```
@article{Niermann2014,
    title = "Averaging scheme for atomic resolution off-axis electron holograms",
    journal = "Micron",
    volume = "63",
    pages = "28 - 34",
    year = "2014",
    doi = "https://doi.org/10.1016/j.micron.2014.01.008",
    author = "T. Niermann and M. Lehmann",
    keywords = "Off-axis electron holography, High-resolution transmission electron
→microscopy, Iterative reconstruction"
}
```

1.2 Installation

The program is a Python program. Thus, a working Python 3.X distribution is required for running it (support for Python 2.X has been dropped since version 1.1.8). Beside the Python interpreter it requires the following packages:

- numpy (see www.numpy.org)
- scipy (see www.scipy.org)
- h5py (see www.hypy.org)
- pyFFTW is optional; speeds up the averaging (see https://pypi.python.org/pypi/pyFFTW)

The package is tested with following Python versions:

- Python 3.6, numpy 1.11, scipy 0.19, h5py 2.7, PyFFTW 0.10
- Python 3.8, numpy 1.23, scipy 1.9, h5py 3.7, PyFFTW 0.13

The package can be most conveniently installed using the `pip` package manager. Make sure you have Python installed and the Python interpreter is in your path. Go to the command line and execute:

```
python3 -m pip install --upgrade holoaverage
```

Holoaverage leverages the pyFFTW package for speed. If pyfftw can not be installed you can still use holoaverage without problems. You can install pyFFTW by

```
python3 -m pip install --upgrade pyfftw
```

Up to date source versions can be found on the GitHub site: https://github.com/niermann/holoaverage

1.3 Bug reporting

When reporting a bug please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

1.4 License

```
Holoaverage, program for reconstruction and averaging of electron holograms
Copyright (C) 2018-2022 Tore Niermann

This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.
```

```
You should have received a copy of the GNU General Public License
along with this program.  If not, see <https://www.gnu.org/licenses/>.
```

CHAPTER 2

Overview

The *holoaverage* program is controlled by a parameter file, which is supplied to the program as argument. Section *Parameters* explains the individual parameters and the format of the file itself. The general flow of the program is sketched in the following figure, the steps of the program are explained in more detail below.
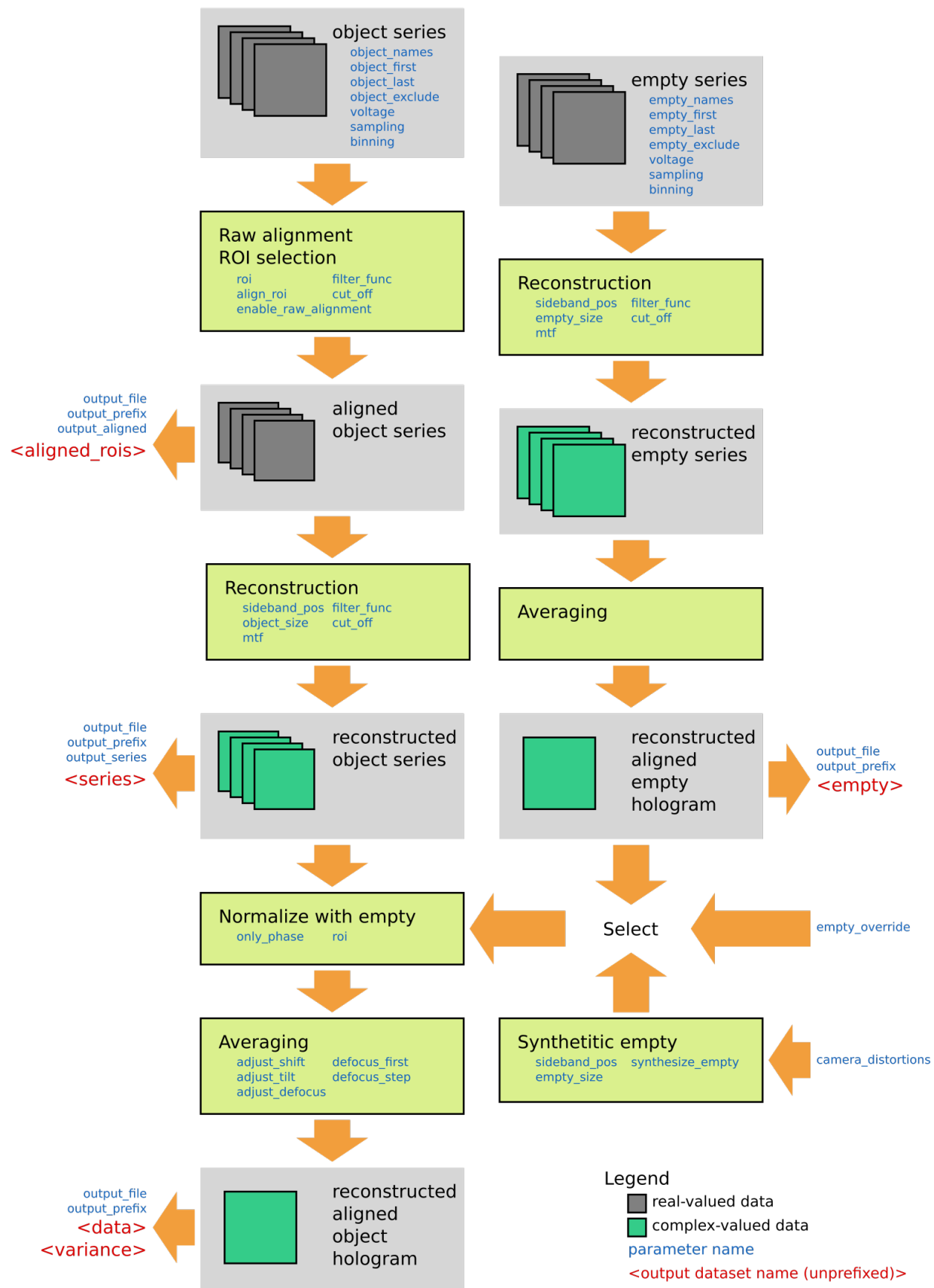
2.1 Object series

The main input to the program is a series of object holograms. Off-axis holograms interfere two regions of the specimen. In object holograms one of these regions is the area of interest, the other is the reference region (typically a vacuum area). This is in contrast to empty holograms (see *Empty series*), where typically both regions only contain vacuum.

The program loads the object holograms from a set of numbered files. The parameter *object_names* gives these filenames and also describes how the file number is encoded within the filename.

For example, if the filenames are `object_001.dm3`, `object_002.dm3`, `object_003.dm3`, ... the filenames are constructed by combing `object_` with a zero padded, three digit number and the extension `.dm3`. This would be encoded by the value `object_%03d.dm3` for the parameter *object_names*. The `%03d` within the parameter describes a zero padded, three digit decimal number.

If the filenames would be `object.1.dm3`, `object.2.dm3`, `object.3.dm3`, ... the value `object.%d.dm3` for the parameter *object_names* should be used.. `%d` in that case describes a simple decimal number without any padding.

A extensive description on this number enconding is given in the old-style formating rules of Python.

The interval of numbers constituting the series is given by the parameter values of *object_first* and *object_last*, which give the first and last number of range (inclusive). Additionally, it is possible to exclude images with specific numbers from the series, for instance if one image is very noises or otherwise unisuitable. This can be achieved by specifying these unwanted image numbers in the list *object_exclude*.

For a list of supported image formats see Section *File paths / File formats*. That section also describes, how to specify which specific dataset of multi-dataset data formats should be used. The program itself has to know certain metadata for the input series. These metadata are the size of one pixel, i.e. the spatial sampling rate, and the used acceleration voltage and the camera binning (the later is needed for correction from the modulation transfer function). The program will take these metadata from the input files (if the input file format provides these information). Otherwise, these informations must be explicitly provided by the parameters *sampling*, *voltage*, and *binning*. If these parameters are provided, the parameter values will used even if the input files also provide these data. This can be used to override the input file information, e.g. for the case where the sampling rate in the input file is incorrect.

2.2 Raw alignment and ROI selection

During the acquisition of the series it may happen that the specimen drifts. These specimen drifts are detected in the raw alignment step. The phase correlation function (see [Meyer02]) of the central band between consecutive images of the series is used for the detection of these drifts. For selection of the central band a low-pass filter is needed. The cut-off frequency for this low-pass can be specified by the parameter *align_cut_off* (in reciprocal nanometers). If no value is for *align_cut_off* is provided, the same cut off frequency as used for the hologram reconstruction is used, as given by parameter *cut_off*.

A region of interest (ROI) from which eventually the holograms are reconstructed can be specified by the *roi* parameters. The region of interest is specified by giving the `left`, `top`, `right`, and `bottom` pixel coordinates of the selected rectangle of the first image in the object series. The detected specimen drift over the series is used to move the ROI rectangular in such a way, that always the same specimen region is selected. If the ROI drifts out of the area covered by the image, the missing values are filled up by the average intensity. The program crops each hologram of the object series to this ROI.

The raw alignment detection by default is performed in the same ROI, which is later also used for holographic reconstruction. It is also possible, to select a different region for alignment than the region that is later used for reconstruction, by specfying an alignment ROI using the parameter *align_roi*. It is also possible to disable the raw alignment by setting the *enable_raw_alignment* parameter to false.

Within the program the aligned ROIs of the object series are internally forwarded to the reconstruction step. By default the aligned ROIs are not saved into the output file. When the *output_aligned* parameter is `true`, the aligned ROIs are also saved as dataset `aligned_rois`.

2.3 Holographic Reconstruction

All (aligned) holograms are reconstructed individually. The reconstruction is performed as described in [Lehmann02] for each hologram:

1. The hologram is Fourier transformed.

2. The modulation transfer function (MTF) is deconvoluted from the hologram. Basically this is done by dividing the Fourier transformed hologram by the MTF. The MTF is passed to the program by the *mtf* parameter (see *Modulation Transfer Function* for a detailed description of the parameter).

3. The transformed data is recentered in Fourier space, such that the sideband is moved to the origin. For this position of the sideband (corresponding to the 2D carrier frequency) must be specified by the parameter *sideband_pos*. Please note, that this sideband position is specified as pixel position in the Fourier transform of the input object images (before cropping it to the specified region of interest).

4. A low pass filter is applied to only select the (centered) side band. For this the spatial frequency of the filter must be specified using the parameter *cut_off*. The type of filter (hard edge, Butterworth, . . .) can be specifyed by the parameter *filter_func* (see description of the parameter).

5. The filtered side band is cropped to a square region of *object_size* pixels size (*empty_size* for reconstruction of the *Empty series*). Basically the region specified by the *roi* parameter (see *Raw alignment and ROI selection*) is rescaled to this size in the reconstructed holograms.

6. The inverse Fourier transform of the cropped region is calculated.

Within the program the reconstruced object holograms are forwarded to the averaging step. By default the individual holograms are not saved into the output file. When the *output_series* parameter is `true`, the aligned ROIs are also saved as dataset `series`.

2.4 Normalization

The holograms are very sensitive to distortions. These occur when the lateral positions of the microscope's object plane (or intermediate image plane) are not uniformly mapped to the camera pixels positions. This happens due to electron optical lens aberrations and due to non-perfect fiber optics in the camera. The distortions typically are visible as large-scale phase variations (lens aberrations) and hexagonal patterns (fiber optics) in the holograms.

It is common to correct these distortions by obtaining a second hologram (series) under the same microscope conditions, but with object removed from the field of view. Within the *holoaverage* program we refer to these holograms as "empty" holograms (see *Empty series*). The normalization is achieved by a complex-number division of the reconstructed object hologram by the reconstructed empty hologram. This removes the effects of these distortions and also normalizes the amplitude of the holograms (see the discussion on "reference holograms" in [Lehmann02] for details).

For the normalization step the *holoaverage* program needs a reconstructed empty hologram covering the whole camera area. This reconstructed empty hologram must not have the same pixel size (i.e. *empty_size* when the empty series is reconstructed) as the input holograms, the program will interpolate the provided reconstructed empty hologram by Fourier padding to the area of the (uncropped) holograms. The actual region of interest as given by parameter *roi* might change due to specimen drift for the indiviudal holograms over the object series. This drift is registered in the *Raw alignment and ROI selection* step. The program will normalize the cropped object holograms, by taking the associated raw aligned region from the reconstructed empty hologram.

When the interference pattern does not cover the full ROI of the holograms (especially in the empty hologram), strong artifacts might occur in this normalization step, due to singular values in the division. This artifacts are reduced when during the normalization only the reconstructed phases are normalized, but the reconstructed amplitudes are left as they are. This normalization of only the phases is selected by setting the parameter *only_phase* to `true`.

The *holoaverage* program allows different data to be used for normalization:

- One possiblity is to use an additional hologram series for this (see *Empty series* for details).

- A reconstructed empty hologram can be explicitly provided. This option is selected, by providing a file name containing a reconstructed empty hologram via the *empty_override* parameter.

- When the camera distortions are known, they can be passed to the program using the *camera_distortions* parameter. When *synthesize_empty* is set to `true` the program will calculate a suitable reconstructed empty hologram for correction of these distortions instead of using experimental data.

- When none of the above possibilities is used, no normalization is performed.

2.5 Averaging

While in principle the holograms in a series are obtained under the same experimental conditions, these experimental conditions are not stable over the time needed to obtain the series. In the averaging step some of these experimental fluctuations are registered and the effects of the fluctuations adjusted, such that the holograms after correction can be averaged. Obviously this a-posteriori alignment of the data is the reasoning behind recording a series instead of a single hologram with prolonged exposure.

The tracked and adjusted instabilities are:

- Biprism drift (corresponds to a change of the global phase of the individual hologram of the series).

- Hologram contrast (corresponds to a change of the global amplitude of the individual hologram of the series).

- Specimen drift (adjustment is controlled by the parameter *adjust_shift*).

- Defocus drift (adjustment is controlled by the parameter *adjust_defocus*).

- Biprism voltage drift (adjustment is controlled by the parameter *adjust_tilt*).

Most of these fluctuations are adjusted to their average value over the series and eventually all holograms are averaged. For the object series also the variance for each reconstructed pixel is estimaged. For further details on the averaging procedure see [Niermann14].

Additionally, the reconstructed holograms are propagated to the Gaussian defocus (even if *adjust_defocus* is set to `false`). The defocus values for the holograms of the object series are given by the parameters *defocus_first* (defocus of first hologram in the series, underfocus is negative) and *defocus_step* (defocus increment/decrement between consecutive holograms of the series). If this propagation should not be performed set both parameters to zero, which is also the default value for these parameters.

The averaged reconstructed object series is written to the output file as dataset `data`, its variance as dataset `variance`. The averaged reconstructed empty series (if calculated) is stored as dataset `empty`.

2.6 Empty series

For the normalization step a reconstructed empty hologram is needed. The program can reconstruct and average this hologram from a series of empty holograms along side the object-series reconstruction. As this series does not contain an object, no alignment for specimen drift is needed. Thus, no raw alignment step is performed on this series. As the reconstructed empty hologram is expected to cover the whole camera area, also no cropping is performed. In the averaging step, no defocus and drift alignment nor a propagation to a certain focus is performed.

The parameters describing the empty series, *empty_names*, *empty_first*, *empty_last*, *empty_exclude*, and *empty_size*, are equivalant to their counterpart for the object series. The averaged reconstructed empty hologram is stored as dataset `empty` into the output file.

Usage

3.1 Command Line

A `holoaverage` command line interface is provided (make sure you're python distribution's script directory is in the path).

The `holoaverage` command line program has the following syntax:

```
holoaverage [-vV] parameter-file

The parameter file is a JSON file. See section :ref:`sec-parameters` for details.
If "-" is passed as parameter file name, the parameters are read from stdin.

Options:
    -v Verbose
    -V Print version number and exit

Examples for parameter file are giben in the :ref:`sec-tutorial`.
```

3.2 Python API

Instead of calling the program on the command line, it can be also directly invoked within python. The parameters are passed as a dictionary object to the following function:

`holoaverage.`**`holoaverage`**(*param[, basepath=""][, verbose=0]*)

   Reconstruct averaged holograms. See documentation for parameter description.

   Parameters

   - **param** (*dict*) – Dictionary with parameters

   - **basepath** (*str*) – All filenames are taken relative to this path (defaults to current directory)

- **verbose** (*int*) – Verbosity level (defaults to 0)

<div align="right">

CHAPTER 4

</div>

<div align="right">

Tutorial

</div>

This tutorial is intended as a quick-start guide to show you, how you can average and reconstruct your hologram series. It assumes you are familiar with the general hologram reconstruction routine. A more thorough description of the general reconstruction can be found in textbooks. For instance, the chapter on electron holography in "Carter and Williams (Eds.), Transmission Electron Microscopy, Diffraction, Imaging, and Spectrometry, Springer" (the companion volume) covers these aspects.

4.1 Prerequisites

Obviously, *holoaverage* must be properly installed on your local computer (see *Installation*).

This tutorial shows how the required parameters for the reconstruction and averaging of a hologram series can be obtained in Digital Micrograph 2.30 by Gatan Inc. (also known as Gatan Microscopy Suite). Other microscopy software provides similiar capabilities to evaluate your data and can be used accordingly.

Furthermore a text editor is needed for writing and editing the parameter files provided to *holoaverage*. You can use your favorite text editor for this (this is not the same as word processing software), just make sure you save your files in "UTF-8" encoding (e.g. when you use Windows' notepad for this, select "UTF-8" in the encoding field in the save dialog; see the article on Wikipedia, if you want to learn more about encoding).

4.2 Data files

This tutorial uses a high-resolution holographic focal series of a GaN as example. This focal series is available as zip-archive from the DOI 10.14279/depositonce-6674.

Unzip the contents into a directory. You will see that actually two hologram series are contained, one with the GaN crystal in one partial wave. These are the files `a1_001.dm3, a1_002.dm3, a1_003.dm3,...`.
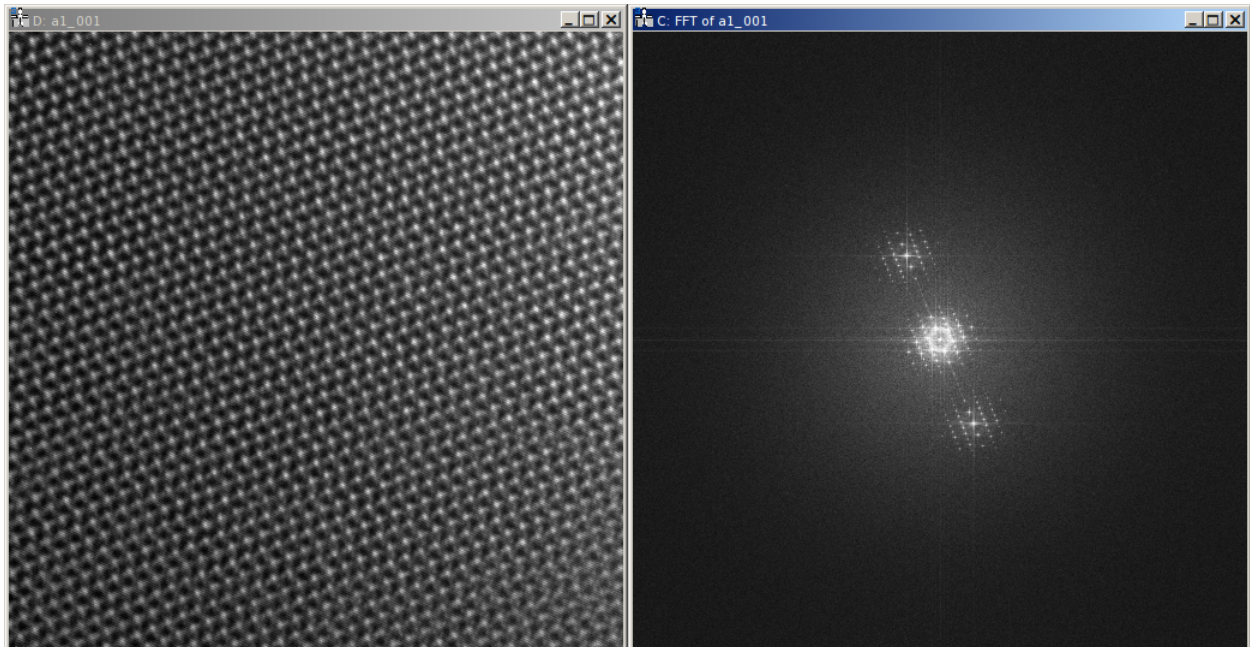
The other hologram series was obtained without an object in the beam and is used for normalization (flat-field correction). These are the files `empty_001.dm3`, `empty_002.dm3`, ...
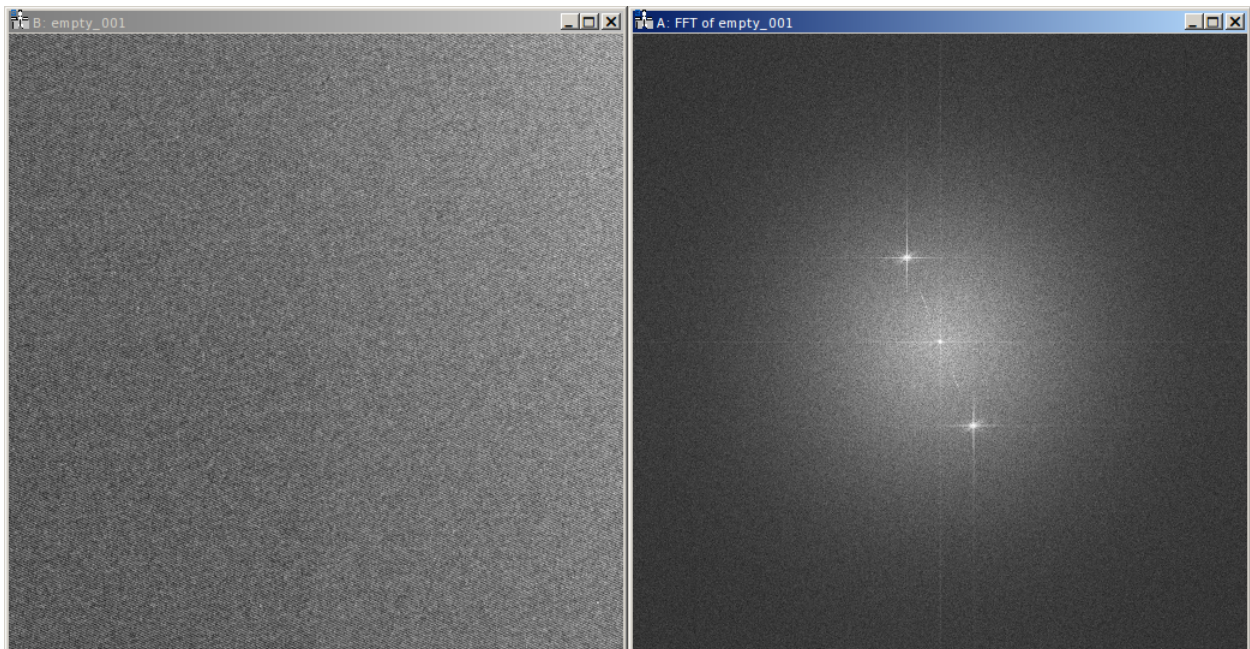
4.3 Overview

The main input to the program is a series of object holograms. Off-axis holograms interfere two regions of the specimen. In object holograms one of these regions is the area of interest, the other is the reference region (typically a vacuum area). This is in contrast to empty holograms, where typically both regions only contain vacuum.

The program performs roughly these steps:

1. Reconstruction of empty hologram series

2. Alignment and averaging of the reconstructed empty holograms

3. Raw alignment of the object hologram series

4. Reconstruction of the object hologram series

5. Flat field correction of the reconstructed object holograms by the averaged empty hologram

6. Alignment and averaging of the reconstructed object holograms

The alignment and averaging steps of both hologram series are done iteratively. Details on the program flow are given in the section *Overview*. Details on the whole alignment procedure can be found in

> T. Niermann and M. Lehmann
>
> Averaging scheme for atomic resolution off-axis electron holograms
>
> Micron 63 (2014) 28-34
>
> doi: 10.1016/j.micron.2014.01.008

The empty hologram series is only aligned for a global amplitude and phase factor (effects of filament instability). Four iterations of the averaging loop are performed. If enabled (see *adjust_tilt*) it is also aligned for the drift of biprism voltage.

The object hologram series is additionally aligned for specimen drift (which can be disabled by *adjust_shift*), and focal drift (if enabled by *adjust_defocus*). Seven iterations of the averaging loop are performed, the focal drift alignment is only done in the last three iterations.

Despite the name of the above mentioned paper, the averaging scheme also works for other kind of holographic reconstructions, e.g. medium resolution holography, or geometric phase analysis.

4.4 Creating the parameter file

Create a text file in the directory, where you unzipped your data files. The text file will contain the parameters used for the reconstruction and averaging. Copy the following text into this file and save it under the name `holoaverage-a1.json`. You can choose the name of the parameter file freely, however we will assume you called it `holoaverage-a1.json` in the following. We will discuss the meaning of the most important parameters in the following. A thorough reference for all parameters can be found in the Section *Parameters* of the documentation. Lines beginning with double slashes (i.e. `//`) are comments and are ignored by the program.

```
{
    // Path of hologram files (using printf format, integer argument). Required.
    "object_names" : "a1_%03d.dm3",

    // Index of first hologram.
    "object_first" : 1,

    // Index of last hologram. Required.
    "object_last" : 20,

    // Same (name, first index, last index) for "empty" holograms.
    "empty_names" : "empty_%03d.dm3",
    "empty_first" : 1,
    "empty_last" : 20,

    // Sampling (nm/px) of object holograms.
    // Defaults to value in data files.
    "sampling" : 0.00519824,
```

(continues on next page)

```
    // Voltage in kV. Default to value recorded in data files.
    "voltage": 300,

    // Defocus of first hologram (i.e. object_first) in nm. Defaults to 0.
    "defocus_first" : 20.0,

    // Defocus step size in nm. Defaults to 0.
    "defocus_step" : -2.0,

    // Size (in px) used for reconstruction of "empty" holograms. Required.
    "empty_size" : 512,

    // Size (in px) used for reconstruction of "object" holograms Required.
    "object_size" : 384,

    // X, Y Position of sideband in FFT pixels (origin is in center). Required.
    "sideband_pos" : [1136, 1304],

    // Reconstruction region in pixels (L, T, R, B). Defaults to full region.
    "roi" : [128, 128, 1920, 1920],

    // Region for raw alignment in pixels (L, T, R, B). Defaults to roi.
    //"align_roi" : [256, 256, 1536, 1536],

    // Disable raw alignment. Raw alignment is enabled by default.
    //"enable_raw_alignment" : false,

    // Output file name (will be HDF5 file). Required.
    "output_name" : "a1.hdf5",

    // Mask type (see FilterFunction for details). Defaults to "EDGE"
    // "filter_func" : "EDGE",
    "filter_func" : ["BUTTERWORTH", 14],

    // cut off frequency in 1/nm (q_max). Required.
    "cut_off" : 14.5,

    // Parameterization for MTF
    "mtf" : [["CONSTANT", -2.25536738e-02],
             ["LORENTZIAN", 1.02543658e-05, 1.15367655e-04],
             ["LORENTZIAN", 2.49224357e-02, 5.35262063e-02],
             ["GAUSSIAN", 4.60461599e-01, 4.36842560e+02]],

    // Correct phase by empty phase only (true), or full complex reconstruction
→(false).
    // Defaults to false.
    "only_phase": false,

    // Optimize defocus. Default: false
    "adjust_defocus" : true,

    // Optimize shift. Default: true
    "adjust_shift" : true
}
```

4.5 Setting the input files

At first the file names for the input files must be provided. Here we have two series, the series of the object holograms, i.e. `a1_XXX.dm3`, and the series of the empty holograms, i.e. `empty_XXX.dm3`. The first parameter *object_names* describes, how the object holograms are named:

```
"object_names" : "a1_%03d.dm3",
```

Our holograms have filenames starting with `a1_`, followed by a three digit number, and have the extension `.dm3`. The `%03d` in the filename describes, how the number is encoded in the filename (here 3 digits, zero padded). If our filenames, for instance, would be called `a1.1.dm3`, `a1.2.dm3`, `a1.3.dm3` ..., we would set the *object_names* parameter to `a1.%d.dm3`, where the `%d` would mean just a decimal number with as many digits as needed. If you interested, how to encode other numbers, look at the old-style formating rules of python.

The next two parameters give the range of numbers contained within the series:

```
"object_first" : 1,
"object_last" : 20,
```

Here, the series starts at index `1` and ends at index `20` (inclusive).

The next parameters describe, how the empty hologram series is named. These parameters follow the same conventions as the parameters described above.

```
"empty_names" : "empty_%03d.dm3",
"empty_first" : 1,
"empty_last" : 20,
```

4.6 The microscope parameters

The example holograms provided here are not correctly calibrated. The holograms were recorded using a non standard setup of the microscope, which is not correctly identified by the recording software. For this reason it is needed to provide the correct calibration to the *holoaverage* program. The correct calibration is specified by the parameter *sampling* and gives the size of one pixel in nanometers. When the *sampling* parameter is present, it overrides any calibrations from the image files. If the parameter is not present in the parameter file, the calibrations are taken from the image files.

Here, the material recorded in the holograms is well known, thus the reflections in the Fourier transforms of the holograms were used for calibration. The following setting makes the correct calibration known to the program

```
"sampling" : 0.00519824,
```

If you are used to Digital Micrograph, you can find the sampling of an image under "Calibrations" in the "ImageDisplay..." dialog available, when you right-click an image.

The *holoaverage* program also has to know the acceleration voltage of the microscope. In this case, it could also read it from the image files itself (i.e. omitting this parameter). Nevertheless, we provide it here in kilovolts:

```
"voltage": 300,
```

In this case the object series is a holographic focal series, which means the defocus is changing from hologram to hologram. The *holoaverage* program will propagate all holograms to the Gaussian focus. The parameter *defocus_first* gives the defocus of the first hologram (given by *object_first*) in nanometers (overfocus is positive). The parameter *defocus_step* gives the increment / decrement of the defocus between two consecutive images of the series. Here

the first hologram is taken with a defocus of (estimated) 20 nanometers, the second with a defocus of (estimated) 18 nanometers. So the defocus is decremented by 2 nanometers, between two consecutive acquisitions:

```
"defocus_first": 20,
"defocus_step": -2,
```

Please note, that the reconstructions are propagated to the Gaussian focus (defocus 0 nm), as given by the defocus parameters above. When wrong parameters are provided, this the target focus is not the Gaussian focus. The resulting holograms can nevertheless propagated to a different focus afterwards. If no defocus parameters are provided, no propagation is performed, and the program averages all reconstructed holograms as they are.

The final parameter describing the microscope, is the modulation transfer function (MTF) of the camera. This MTF must be provided to the *holoaverage* program. As it is practice in most labs to describe the MTF as a parameterization of some simple functions, this parameterization can be directly provided to the program. This description is documented in Section *Modulation Transfer Function*. For this tutorial we simply pass the provided MTF to the program. If you don't know the MTF of your detector, you can omit this parameter in the parameter file. Obviously no MTF correction is performed in this case.

```
"mtf" : [["CONSTANT", -2.25536738e-02],
         ["LORENTZIAN", 1.02543658e-05, 1.15367655e-04],
         ["LORENTZIAN", 2.49224357e-02, 5.35262063e-02],
         ["GAUSSIAN", 4.60461599e-01, 4.36842560e+02]],
```

4.7 Reconstruction parameters

The *holoaverage* program has to know the carrier frequency of your hologram. This carrier frequency is most conveniently identified as the position of the sideband in one of the Fourier transformed empty holograms. In Digital Micrograph you can use the *point ROI tool*. This is the central tool button in the *ROI toolbar* looking like a cross-hair (if this toolbar is not visible, enable it in the "Toolbars" tab of the "Window/Customize..." dialog):
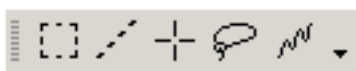
Selecting the center of the desired sideband allows to read out the coordinates of the sideband. Which of the two sidebands is the correct one, depends on the convention you use for your phase and on which side of the biprism your object was located. We use the convention that the phase shift increases with increasing specimen thickness. In this case, the correct sideband is the one, which is located on the side of the biprism filament, where the **reference** wave passed. In this GaN example, the biprism filament was oriented in 8 to 2 o'clock orientation. The object partial wave passed the top side of the filament, the reference partial wave the bottom side. Thus we select the **lower** side band here, located in the 5 o'clock position.
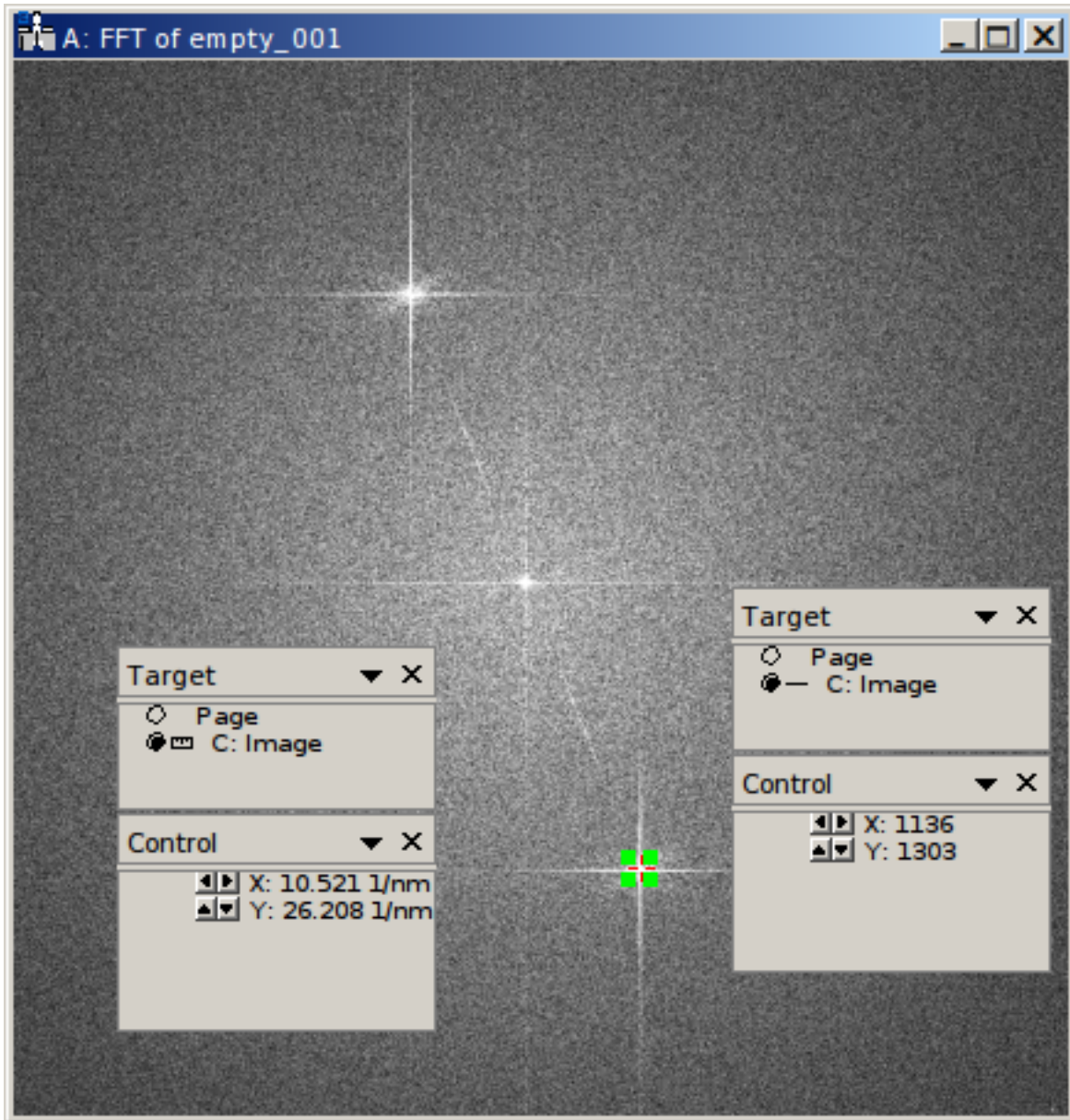
With the marker positioned on the sideband, the coordinates of the sideband can be read in the "Control" panel (if not visible, enable it in the "Window/Floating Window" drop down menu). The "Control" panel actually has two modes, either showing the position in calibrated units (see left part of the above figure) or as pixel position (right part of the above figure). You can toggle the mode by clicking on the little scalebar in the "Target" panel (in the figure above directly left to "C:Image"). For the parameter file we need this position **in pixels**. This position is given by the *sideband_pos* parameter as X and Y coordinates:

```
"sideband_pos" : [1136, 1304],
```

When the sideband is masked out in the reconstruction step, the size and form of the aperture used must be specified. For this the radius of the mask and the type of the mask are needed. The radius of the mask is chosen, such that the sideband is well separated from the central band, and all reflections are included within the radius.
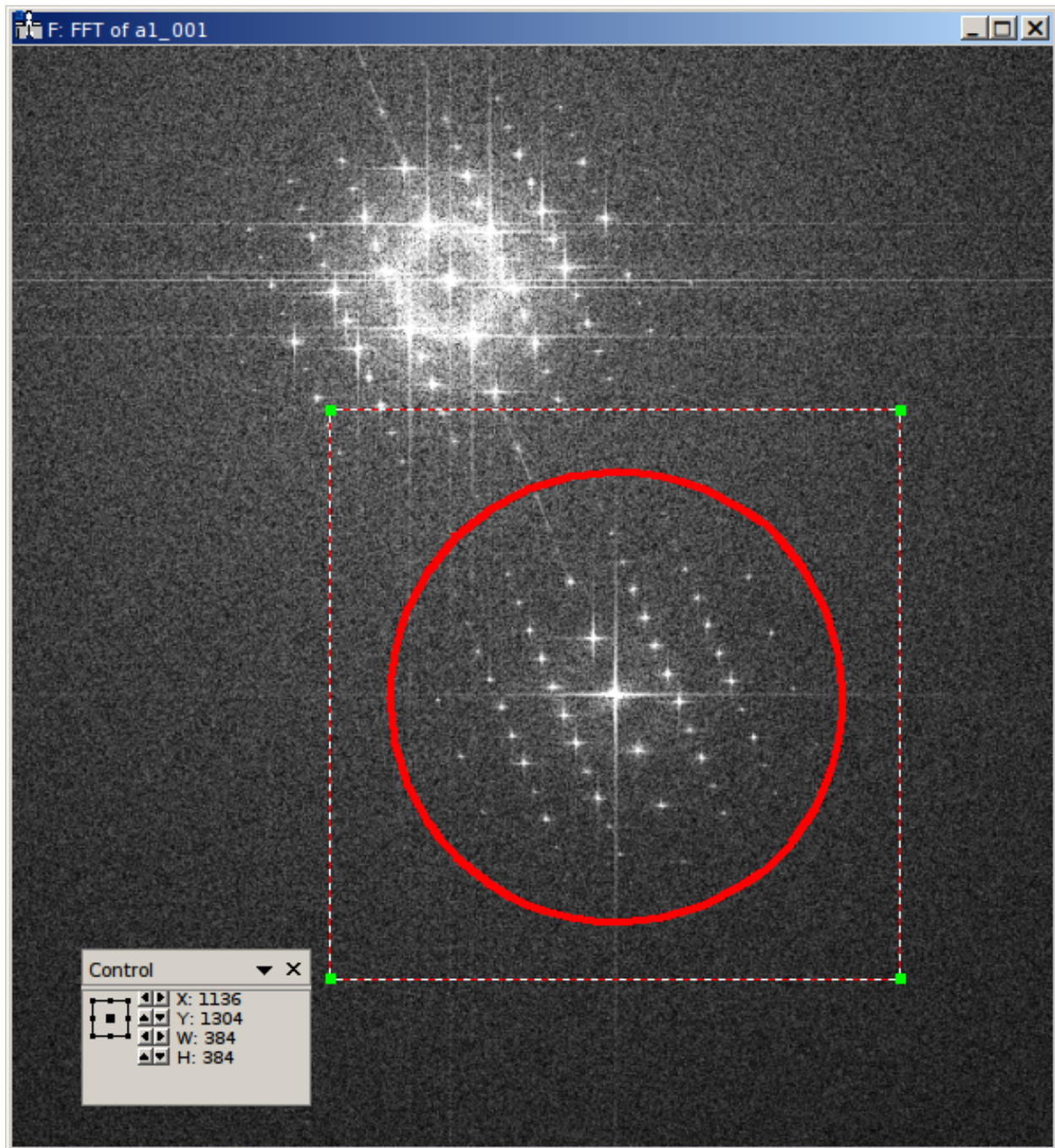
In the figure above, the radius of the red circle is 14.5 1/nm (measured using the Fourier transformed image, after re-calibration; see `sampling` parameter above). This radius in reciprocal nanometers is given by the *cut_off* parameter. Here, we use an aperture with a soft edge, which is specified by providing the string `"BUTTERWORTH"` and the order of the Butterworth function (here `14`) as *filter_func* parameter. For a hard edge, simply provide the string `"EDGE"` instead of the squared parenthesis.

```
"cut_off" : 14.5,
"filter_func" : ["BUTTERWORTH", 14],
```

Due to this masking in Fourier space, the reconstructed holograms will have a lower resolution than the original holograms. Thus the resulting data arrays can made smaller, which also speeds up the averaging process. It is sufficient

to make the result arrays only as large, that the mask defined above is fully contained within them. As we use a soft edge we give here a little leeway and make the result array 384 pixels large (for performance reasons you should prefer sizes with small prime factors here):

```
"object_size" : 384,
```

Another parameter needed for the reconstruction is the region of the object holograms, which should be reconstructed. If the specimen is drifting strongly, not the whole region of the holograms might be present in all acquisitions of the series. In this cases it might be meaningful to only reconstruct a sub region of the hologram:
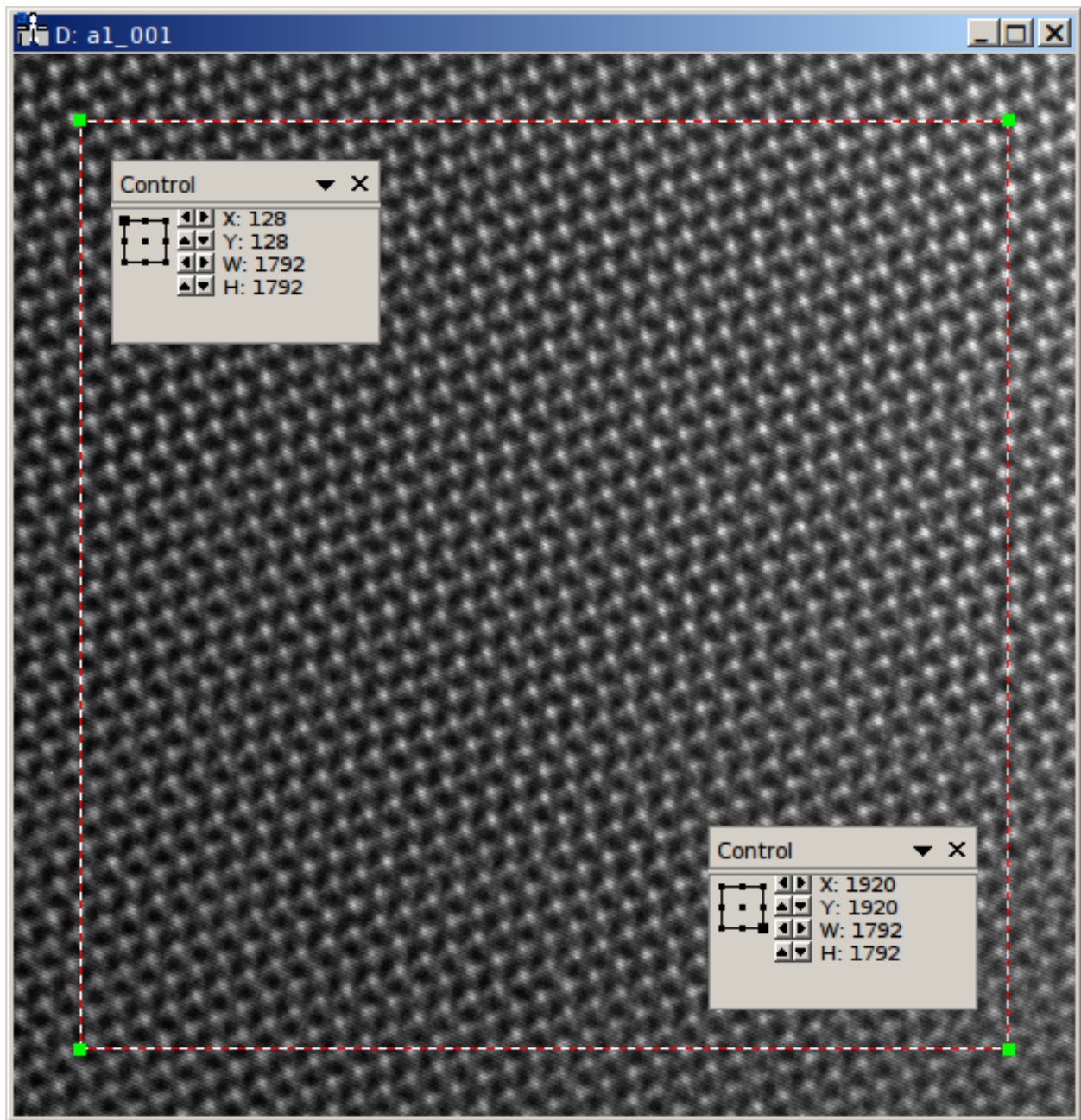
This region is provided to the image program as *roi* parameter. This region of interest (ROI) is defined as rectangular area in the first object hologram. During the raw-alignment step of the reconstruction, this region is tracked and the same sub-region is extracted from the consecutive object holograms. The left, top, bottom, and right borders

are provided to the program in pixel coordinates. As shown in the above figure, the "Control" panel can be used to determine these coordinates, by clicking on the respective corners of the rectangle in the panel. Here only the central 7/8 of the holograms are used, the coordinates are given as left (128), top (128), right (1920), and bottom (1920):

```
"roi" : [128, 128, 1920, 1920],
```

If this parameter is omitted, the whole hologram area is used for reconstruction. Please note, that also the region used for raw alignment can be specified explicitly using the *align_roi* parameter. Alternatively, the raw alignment can be disabled by setting the *enable_raw_alignment* parameter to `false`.

The empty hologram series is always reconstructed using the full image region. This is done, for two reasons. For one, there is no specimen drift in empty holograms. For another, the reconstructed empty hologram is used to remove some distortions of the camera and the projection system of the microscope. This are fixed to the camera pixels. The normalization of the object holograms is made with the corresponding sub-region (tracked for specimen drift) of the reconstructed empty hologram. This requires the empty hologram reconstruction to cover the whole camera area.

The reconstruction size of the empty holograms can be chosen independently, and is made a little bit larger here:

```
"empty_size" : 512,
```

If the whole hologram area is reconstructed in the object series, the `empty_size` parameter should be chosen identically to the `object_size` parameter.

4.8 Output and averaging

One has to select which drifts are tracked in the averaging step. By default only specimen drift is tracked and adjusted. Here we also want to track and adjust for focal variations. These adjustments are selected by the following parameters.

```
"adjust_defocus" : true,
"adjust_shift" : true
```

When the interference pattern is smaller than the area of the hologram, the amplitude normalization of the flat-field correction might produce strong artifacts. In these cases, it might be beneficial to only normalize the phases of the reconstructions. This can be adjusted by the *only_phase* parameter. You should also consider to use the *align_roi* parameter in cases of smaller interference patterns.

```
"only_phase": false,
```

Eventually, the filename of the output file must be provided by the *output_name* parameter. The output will be always HDF5 files, the contents of these files is described in Section *Outputs*.

```
"output" : "a1.hdf5",
```

4.9 Starting the program

When correctly installed, the program should be executable from the console. Change into the directory, where your tutorial data files are located, using the `cd` command of the console. Below it is assumed this directory is called `directory-with-data-files`, so adjust the path `directory-with-data-files` below accordingly. Note, you may have to enter the absolute path to the directory. Windows user might also have to switch the current drive they are working with; for that use the desired drive letter, followed by a colon as command (e.g. `D:`, if the data is located on drive `D:`) before using the `cd` command.

```
cd directory-with-data-files
```

Now call the holoaverage program and pass the name of the parameter file `holoaverage-a1.json` to it as argument. The switch `-v` is optional and enables verbose output.

```
holoaverage -v holoaverage-a1.json
```

The program now should start with the reconstruction and averaging and should output something like:

```
Loading parameters from
        /home/niermann/PycharmProjects/holoaverage/example/holoaverage-a1.json
Loading...
        20 datasets, shape=(2048, 2048), dtype=int16
        [00] GaN_holographic_focal_series/empty_001.dm3
        [01] GaN_holographic_focal_series/empty_002.dm3
        [02] GaN_holographic_focal_series/empty_003.dm3
        [03] GaN_holographic_focal_series/empty_004.dm3
        [04] GaN_holographic_focal_series/empty_005.dm3
        [05] GaN_holographic_focal_series/empty_006.dm3
        [06] GaN_holographic_focal_series/empty_007.dm3
        [07] GaN_holographic_focal_series/empty_008.dm3
        [08] GaN_holographic_focal_series/empty_009.dm3
        [09] GaN_holographic_focal_series/empty_010.dm3
        [10] GaN_holographic_focal_series/empty_011.dm3
        [11] GaN_holographic_focal_series/empty_012.dm3
        [12] GaN_holographic_focal_series/empty_013.dm3
        [13] GaN_holographic_focal_series/empty_014.dm3
        [14] GaN_holographic_focal_series/empty_015.dm3
        [15] GaN_holographic_focal_series/empty_016.dm3
        [16] GaN_holographic_focal_series/empty_017.dm3
        [17] GaN_holographic_focal_series/empty_018.dm3
        [18] GaN_holographic_focal_series/empty_019.dm3
        [19] GaN_holographic_focal_series/empty_020.dm3
Reconstructing...
        . . . . . . . . . . . . . . . . . . . .
Optimizing after iteration 0
        [NN] sx[px] sy[px] tx[1/px] ty[1/px] def[nm] Ampl   Phase  Error
        [00]  0.000  0.000  0.00000  0.00000   0.000 1.0000 +0.802 2.403186e+09
        [01]  0.000  0.000  0.00000  0.00000   0.000 1.0000 -0.672 2.365460e+09
        [02]  0.000  0.000  0.00000  0.00000   0.000 1.0000 -1.478 2.599819e+09
        [03]  0.000  0.000  0.00000  0.00000   0.000 1.0000 -2.849 2.534287e+09
        [04]  0.000  0.000  0.00000  0.00000   0.000 1.0000 -1.102 2.451658e+09
        [05]  0.000  0.000  0.00000  0.00000   0.000 1.0000 +0.163 2.365165e+09
        [06]  0.000  0.000  0.00000  0.00000   0.000 1.0000 +0.439 2.400905e+09
        [07]  0.000  0.000  0.00000  0.00000   0.000 1.0000 -0.745 2.470458e+09
        [08]  0.000  0.000  0.00000  0.00000   0.000 1.0000 +0.303 2.364581e+09
        [09]  0.000  0.000  0.00000  0.00000   0.000 1.0000 +1.830 2.513006e+09
        [10]  0.000  0.000  0.00000  0.00000   0.000 1.0000 +1.208 2.377789e+09
        [11]  0.000  0.000  0.00000  0.00000   0.000 1.0000 -0.781 2.302518e+09
...
```

and so on. Eventually it should output something like:

```
Iteration   7: total error=7.501946e+07
```

This error number should go down and converge to a stable value within the last iterations.

4.10 Results

After the program succeeded the outputs can be found in the output file as specified by the *output_name* parameter. In the present example the outputs are in the file `a1.hdf5` in the same directory as the parameter file.

If you are using Digital Micrograph, there is also a plugin to read HDF5 files, which can be found at https://github.com/niermann/gms_plugin_hdf5/releases. These outputs should be checked for artifacts.

The averaged empty reconstruction is contained in the HDF5 file as dataset `empty`. It should be more-or-less homogeneous. If the interference only covers the camera partially, only the area of the interference pattern should be homogeneous obviously. Also Fresnel fringes might cause small deviations, the central area of the interference region should be homogeneous nevertheless. Here the phase (left) and amplitude (right) of the empty example series are shown.
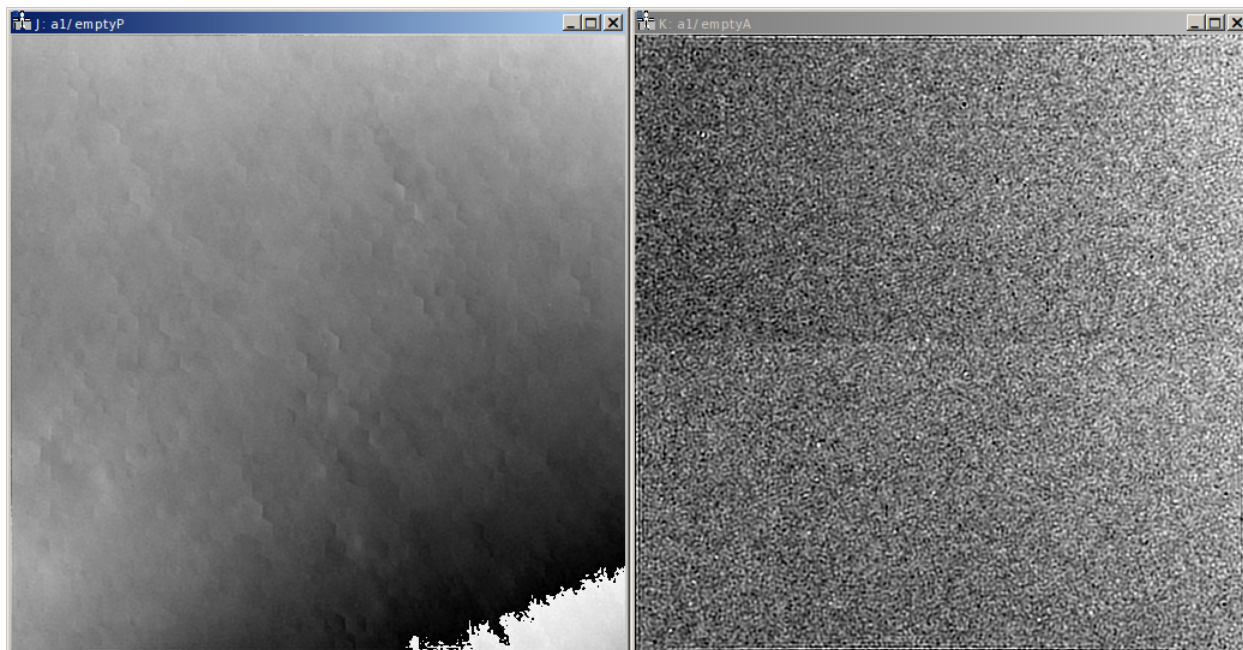
The patterns present in the phase are the aforementioned camera and projection distortions.

The reconstructed object series can be found in the dataset `data` of the HDF5 file. Here the amplitude (left), phase (center), and Fourier transform (right) of the averaged reconstruction are shown:
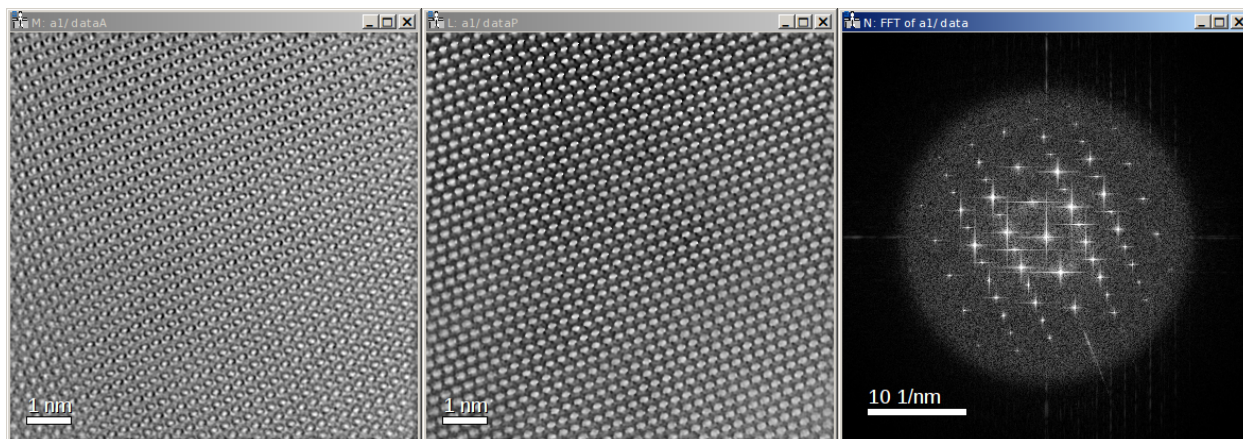
Please note, that this object hologram doesn't show the GaN structure perfectly, due to residual aberrations. But since

you now have reconstructed the full complex information you can now correct for aberrations a-posteriori.

Finally, the variance estimation of the averaging step should be checked. The reconstruction shown above represents the average of the object series. When the object series is averaged, also the variance of the object series (drift corrected and propagated to the Gaussian focus) is checked. This variance for each pixel of the reconstructed object hologram is provided in the dataset `variance` of the HDF5 file. Small variances (compared to the amplitude values) are to be expected especially at positions with strong contrasts (specimen edges etc.). Also hot pixels of single holograms might pop up in the variance array. However, strong variance indicate a problem with either the data or the averaging process.
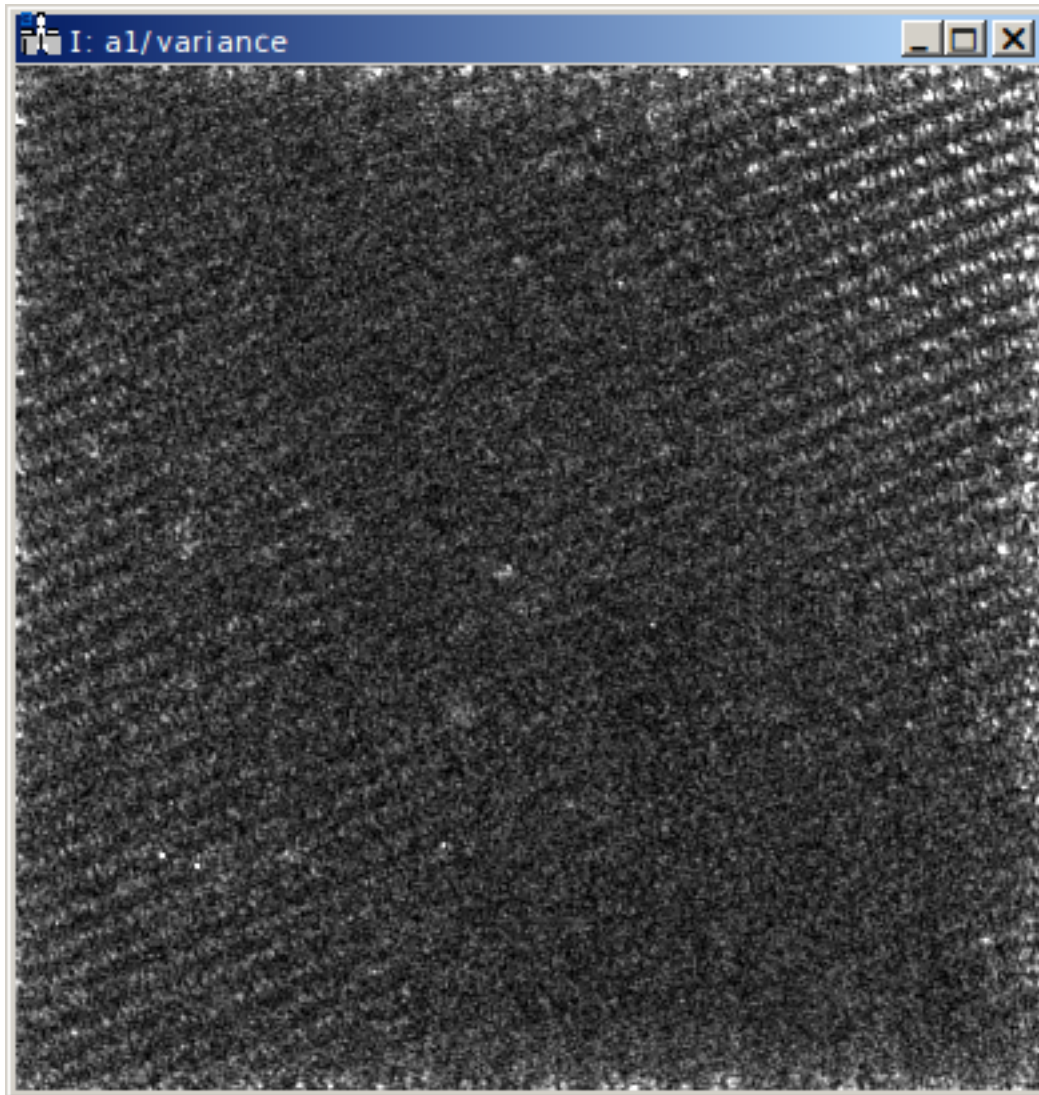
Parameters

5.1 File format

Parameter files for *holoaverage* are plain text files with a JSON (http://www.json.org) compatible syntax. Such files can be edited by any text editor. The whole parameter set must be a JSON object. Beside pure JSON the parser also allows Javascript-like comments (line and block comments), as well as trailing commas. See the *Tutorial* for example files.

When a parameter file is provided, *holoaverage* expects it to be in UTF-8 encoding (see Wikipedia article for details). For parameters from *stdin* it uses the standard Python encoding (you can use the PYTHONIOENCODING environment variable to override this behavior).

5.2 File pathes / File formats

Some parameters are file pathes. All file pathes are relative to the directory, where the parameter file is located. If the parameters are read from *stdin*, the pathes are relative to the current directory. This behavior can be changed by setting the *path* parameter (see description of parameter for details).

Within a path forward slashes (/) as well as backslashes (\\) both are treated as path separators. Internally all pathes are normalized to the platform's separator (e.g. backslashes on Windows platforms).

In JSON strings the backslash character is used as escape sequences with special meaning. For this reason, if you wan't to write a single backslash (like in a Windows path separator) into a JSON string you always have to put in a double backslash. For example the JSON string `"this\\is\\my\\path"` becomes the path `this\is\my\path`.

Some file types allow to pass additional parameters to the file reader. These parameters are separated from the file name by a question mark (`?`). The parameters itself are separated by an ampersand (`&`). Parameters have a parameter name and a value, both separated by a equal sign (=).

Changed in version 1.1.2: Separation of parameters by ampersand (instead of question marks), like in HTTP query strings.

For instance the path `some_file.raw?xsize=1024&ysize=1024&dtype=int32` is interpreted as file name `some_file.raw` with three parameters named `xsize` (value 1024), `ysize` (value 1024) and `dtype` (value `int32`).

The file type is recognized from the extension. You can manually select the file type by passing the parameter `type` with an extension described below as value (e.g. `file.hdf5?type=dm3` is read as DM3-file even if its has the extension of a different file type.

Supported file types are:

5.2.1 Digital Micrograph 3 Files

File Extension `dm3`

Description Sampling, Acceleration voltage and camera binning are read from file if possible.

Parameters None

5.2.2 Hierarchical Data Format 5

File Extension `hdf5`, `h5`

Description Sampling, Acceleration voltage and camera binning must be given in parameters.

Parameters

- *dataset* - The name of the dataset (required).

5.2.3 Raw binary files

File Extension `raw`

Description Sampling, Acceleration voltage and camera binning must be given in parameters.

Parameters

- *xsize* - Width of the image in pixels(required).

- *ysize* - Height of the image in pixels(required).

- *dtype* - Datatype (required; numpy compatible description: `int32`, `uint16`, `complex64`, `float32`, `F4`, ...)

- *offset* - Number of bytes to skip at file head (optional, defaults to 0)

- *swap_bytes* - Whether the bytes should be swapped (optional; 0 or 1; defaults to 0)

5.3 Parameter reference

This reference gives a description of the parameters. The format field gives the expected type(s) of the parameter. The actual types depend on whether the parameters are used in a JSON parameter file or as a Python object. The corresponding types are:

Format	JSON parameter file	Python
None/Null	`null`	`None`
Boolean	`false` or `true`	`False` or `True`
Integer	Number type	Type `int`
String	String type	Type `str`
List	Array type	Type `list`
Dictionary	Object type	Type `dict` with `str` keys

5.3.1 adjust_defocus

Parameter `adjust_defocus`

Type Optional (default is `false`)

Format Boolean

Description Switch, which determines whether the object reconstructions should be aligned for defocus variations. Due to instabilities of the microscope's stage or lens currents the defocus between the individual exposures of the series might drift. When this switch is set to `true`, the program tries to detect defocus deviations in the object hologram series.

5.3.2 adjust_shift

Parameter `adjust_shift`

Type Optional (default is `true`)

Format Boolean

Description Switch, which determines whether the object reconstructions should be aligned for specimen drift. When this switch is set to `true`, the program tries to shift all object holograms to a common position during the averaging step. This "fine" alignment is performed independently from the "raw" alignment, which is controlled by the parameter *enable_raw_alignment*.

5.3.3 adjust_tilt

Parameter `adjust_tilt`

Type Optional (default is `false`)

Format Boolean

Description Switch, which determines whether the object reconstructions should be aligned for drift of the sideband position. Such a drift might occur when the voltage supply of the biprism is not stable. Usually this alignment is not needed.

5.3.4 align_cut_off

Parameter `align_cut_off`

Type Optional (by default value of parameter *cut_off* is taken)

Format Floating point number

Unit Reciprocal nanometer (1/nm)

Description The value specified by this parameter is taken as cut-off frequency for the low pass used in the raw alignment step. For the raw alignment low pass, always a hard aperture (edge function) is taken. Please note, that if a wrong *sampling* is specified, the value of this parameter does not refer to the correct spatial frequency.

If this parameter is not given, the value of the parameter *cut_off* or *cut_off2* is used.

New in version 1.1.5.

5.3.5 align_roi

Parameter `align_roi`

Type Optional (by default region from parameter *roi* is taken)

Format List of four integers

Unit Pixels

Description `[left, top, right, bottom]` pixel coordinates of the region used for raw alignment of the object holograms. This region can be specified independently from the reconstruction region (as given by *roi*).

If this parameter is not given the reconstruction region *roi* is also used for raw alignment.

5.3.6 binning

Parameter `binning`

Type Optional (taken from input files by default).

Format Integer

Description Binning used for recording of the holograms. This parameter affects, how the parameterization of the MTF (see *mtf*) is interpreted. If this parameter is not given, the binning is taken from the image files. If the image files provide no binning, it is assumed to be one.

5.3.7 camera_distortions

Parameter `camera_distortions`

Type Optional

Format List of two Strings

Description Per pixel displacements due to camera distortions. The optics of the camera itself produce small displacements. This parameter contains two filenames. The first filename contains an array with the X-displacement of each pixel. The second filename contains the Y-displacements. The referenced arrays must have the same dimensions as the holograms. The displacements are given in units of pixels. These displacements are only used, if the parameter *synthesize_empty* is set.

5.3.8 cut_off

Parameter `cut_off`

Type Mandatory

Format Floating point number

Unit Reciprocal nanometer (1/nm)

Description This parameter defines in combination with the parameter *filter_func*, how the masking of the sideband in Fourier space is done. This is typically the radius of the mask used. The smaller this is chosen, the lower the resolution of the reconstructions will be. However, smaller values will spatially average the reconstructions more, thus decreasing the noise present in the holograms (at the cost of larger spatial correlations). Please note, that if a wrong *sampling* is specified, the value of this parameter does not refer to the correct spatial frequency. Instead of this parameter the parameter *cut_off2* can be specified.

5.3.9 cut_off2

Parameter `cut_off2`

Type Alternative to (*cut_off*)

Format 2x2 matrix of floating point numbers (list of two lists of two floats)

Unit Reciprocal nanometer squared (1/nm2)

Description This parameter extents the functionality of the parameter *cut_off* for non-isotropic masking. For a general description of the overall parameter see *cut_off*. For a masking with radius *a* along the major axis with an angle of *alpha* to the x-axis and a radius of *b* along the minor axis, specify

$$\text{cut\_off2} = R^T \cdot \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix} \cdot R$$

with

$$R = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

If this parameter is specified, the parameter *cut_off* must not be present.

Raw alignment still uses isotropic filtering with the geometric mean of both radii as radius.

New in version 1.1.4.

5.3.10 defocus_first

Parameter `defocus_first`

Type Optional (default is 0.0 nm)

Format Floating point number

Unit Nanometers

Description Defocus of first object hologram (hologram with index given by *object_first*). Negative focus values refer to underfocus. The reconstructed (averaged) object hologram is propagated to the Gaussian focus (i.e. defocus of zero) during reconstruction. No propagation of the reconstructed hologram is performed, when the defocus of an hologram is given as zero. The empty holograms are never propagated. Please note, that if the sampling of the holograms (see *sampling*) or the acceleration voltage (see *voltage*) are wrongly specified, the propagation will be performed wrongly. Also note, that if the defocus is specified wrongly, the holograms will be be propagated to a different focus than the Gaussian one.

5.3.11 defocus_step

Parameter `defocus_step`

Type Optional (default is 0.0 nm)

Format Floating point number

Unit Nanometers

Description Step of defocus between consecutive object holograms in the series. This is intended for the case that the hologram series is also a focal series, where every hologram has a different defocus. Defaults to 0.0 nm (all object holograms were taken at same defocus).

5.3.12 empty_exclude

Parameter `empty_exclude`

Type Optional (default is empty list)

Format List of integers

Description A list of empty hologram indices, which should **not** be used for averaging. See *object_exclude* for the rationale of this parameter. By default this list is empty and all empty holograms in the given range are used.

5.3.13 empty_first

Parameter `empty_first`

Type Mandatory

Format Integer

Description Index of first hologram in the empty hologram series.

5.3.14 empty_last

Parameter `empty_last`

Type Mandatory

Format Integer

Description Index of last hologram (inclusive) in the empty hologram series.

5.3.15 empty_names

Parameter `empty_names`

Type (see Description)

Format String or list of string

Description If this parameter is a list of strings, the list compromises the filenames of the empty hologram series. In this case the parameters *empty_first*, *empty_last*, and *empty_exclude* are ignored.

If this parameter is a single string, it represents the pattern used to form the file name of the empty hologram series. See *object_names* for the description of the format of this pattern.

If the parameter `empty_names` is not present in the parameter file, no empty hologram series will be reconstructed and averaged. In this case, the parameters *empty_first*, *empty_last* are not needed.

Changed in version 1.1.6: Instead of specifying a filename pattern, also a list of filenames can be supplied.

5.3.16 empty_override

Parameter `empty_override`

Type Optional

Format String

Description File name of empty hologram used for normalization. If this parameter is present in the parameter files the empty hologram will be read from this file (see *File paths / File formats* for format) and the parameters *empty_names*, *empty_first*, *empty_last*, and *empty_size* are ignored.

5.3.17 empty_size

Parameter `empty_size`

Type Optional (default is given by parameter *object_size*)

Format Integer

Unit Pixels

Description Size of the reconstructed empty hologram. See *object_size* for details concerning this parameter. For normalization of the reconstructed object holograms the reconstructed empty hologram is interpolated to the size of the object holograms (before its cropped to the *roi* region) by zero-padding. If parameter *empty_size* is missing, it is substituted by *object_size*.

5.3.18 enable_raw_alignment

Parameter `enable_raw_alignment`

Type Optional (default is `true`)

Format Boolean

Description Enables the raw alignment. If the raw alignment is disabled, the region of interest is taken from the same area in each hologram of the object hologram series. Otherwise, the region of interest is tracked across the series.

New in version 1.1.

5.3.19 filter_func

Parameter `filter_func`

Type Optional (default is `"EDGE"`)

Format see below

Description This parameter gives the function that will be used in combination with the parameter *cut_off* for masking the sideband in Fourier space. The format of this parameter is either a string describing the filter function, or a list with the function name as first element and further parameters in the remaining list.

If `filter_func` is `"EDGE"`, an edge function is used. This corresponds to a hard mask at the `cut_off` spatial frequency. If the edge function is chosen, you might observe "ringing" artifacts in the reconstructions especially at the borders or at "hot pixels".

If `filter_func` is `"GAUSSIAN"`, a Gaussian function is used. The Gaussian is chosen such that a `1/e` fall-off is reached at the `cut_off` spatial frequency.

If `filter_func` is `["BUTTERWORTH", order]`, a Butterworth function of the given order is used. This corresponds to a soft mask at the `cut_off` spatial frequency. The lower the order of the Butterworth function is, the softer this filter becomes.

If this parameter is not given, the edge function is used.

5.3.20 mtf

Parameter `mtf`

Type Optional

Format List

Description Parameterization of the camera MTF. The reconstruction are corrected for the effects of MTF (by dividing the Fourier transformed holograms by the MTF). See *Modulation Transfer Function* for details on the specification of this parameter. If this parameter is not given, no MTF correction is performed.

5.3.21 object_exclude

Parameter `object_exclude`

Type Optional (default is empty list)

Format List of integers

Description A list of object hologram indices, which should **not** be used for averaging. Usually all holograms with indices between *object_first* and *object_last* (inclusive) are used for averaging. Any indices occurring in this list are not used. For example with `object_first` of `1`, `object_last` of `5`, and `object_exclude` set to `[3, 4]` only object holograms with indices `1`, `2`, and `5` are used, since indices `3` and `4` were explicitly excluded. By default, this list is empty and all object holograms in the given range are used.

5.3.22 object_first

Parameter `object_first`

Type Mandatory

Format Integer

Description Index of first hologram in the object hologram series.

5.3.23 object_last

Parameter `object_last`

Type Mandatory

Format Integer

Description Index of last hologram (inclusive) in the object hologram series.

5.3.24 object_names

Parameter `object_names`

Type (see Description)

Format String or list of string

Description If this parameter is a list of strings, the list compromises the filenames of the object hologram series. In this case the parameters *object_first*, *object_last*, and *object_exclude* are ignored.

If the parameter is a single string, this string describes a pattern how to create the filenames of the object hologram series. Typically a series hologram file names contain an increasing number. The number in this parameter is encoded with the *printf*-style format rules (old-style formating in python). For instance simple numbers can be expressed as `%d` and become 1, 2, 3, etc. If you want to have zero padded three digit numbers use `%03d`, which becomes 001, 002, 003, etc. Due to this formatting rules you have to write a double percent sign (i.e. `%%`) if you want a single `%` in your filename.

If the parameter `object_names` is not present in the parameter file, only the empty hologram series will be reconstructed and averaged. In this case, the parameters *object_first*, *object_last*, and *object_size* are not needed.

Changed in version 1.1.6: Instead of specifying a filename pattern, also a list of filenames can be supplied.

5.3.25 object_size

Parameter `object_size`

Type Mandatory

Format Integer

Unit Pixels

Description Size of the reconstructed object hologram. Reconstructed holograms always have same size in width and height. This size in pixels is given by this parameter. The *roi* of the object holograms is scaled to this size during the reconstruction (by cropping in Fourier space). This parameter should be larger than the diameter of filter used during the reconstruction (see *cut_off* parameter). For performance reasons a number with low prime factors should be chosen, e.g. prefer `384 = 3 * 2^7` over `383` (prime).

5.3.26 only_phase

Parameter `only_phase`

Type Optional (default is `false`)

Format Boolean

Description Switch, which determines how the object reconstructions are normalized. When this parameter is `true`, the normalization is performed by dividing the individual reconstructed object holograms by the reconstructed (and averaged) empty hologram. This normalizes the object holograms in amplitude in phase. However, if the reconstructed empty hologram contains regions, where the amplitude is very small, the normalization will cause artifacts. Such cases typically occur when the interference region, does not cover the whole image. When this parameter is `true`, only the phases of the reconstructed holograms are normalized.

5.3.27 output_aligned

Parameter `output_aligned`

Type Optional (default is `false`)

Format Boolean

Description When set to `true`, the region of interest of the individual object holograms (before reconstruction) are also stored in the output file.

5.3.28 output_name

Parameter `output_name`

Type Mandatory

Format String

Description Name of the output file. The output(s) will be always stored in HDF5 format.

Changed in version 1.1: The parameter was renamed from `output` to `output_name`.

5.3.29 output_prefix

Parameter `output_prefix`

Type Optional (Defaults to empty string)

Format String

Description Prefix to dataset names in output file. By using the prefix multiple outputs can be written to the same HDF5 file. Especially forward slashes can be used in *output_prefix* to create the outputs in sub-groups. As example, if the value of `output_prefix` would be `alpha_`, the dataset `data` is saved as `alpha_data` in the output file.

New in version 1.1.

5.3.30 output_series

Parameter `output_series`

Type Optional (default is `false`)

Format Boolean

Description When set to `true`, also the individual object hologram reconstructions are stored in the output file. The averaged hologram (and the variance estimation obtained during averaging) are always stored in the output file. The individual reconstructions of the empty hologram series are never stored.

5.3.31 path

Parameter `path`

Type Optional (default is none)

Format String

Description All (relative) file names are relative to this path. Absolute file names are still absolute. If this parameter itself is not an absolute path, the path is taken relative to the path of the parameter file (current directory, if the parameters are read from *stdin*). By default this path is left empty, which means all file names are relative to the parameter file path (or the current directory, when the parameters are read from *stdin*; see *File paths / File formats*).

5.3.32 roi

Parameter `roi`

Type Optional (default is full image region)

Format List of four integers.

Unit Pixels

Description `[left, top, right, bottom]` pixel coordinates of the region of interest (ROI) in the first object hologram (as given by parameter *object_first*). The ROI is always a rectangular region. In the raw alignment step (*Overview*) of the hologram series the position of this ROI is aligned to the drift of the object, such that always the same object region is taken from each hologram.

The *left* and *top* pixel positions given here refer to the top, left corner in this rectangular region (inclusive). The *right* and *bottom* positions refer to the bottom, right corner (exclusive), which means they refer the pixel coordinate adjacent to right (bottom) edge of the ROI. X coordinates are going from left to right, Y coordinates are going form top to bottom. For performance reasons, the size of the ROI, i.e. `right - left` and `bottom - top`, should have only low prime-factors, e.g. prefer `384 = 3 * 2^7` over `383` (prime).

If this value is not given, the whole object hologram region is taken as ROI.

5.3.33 sampling

Parameter `sampling`

Type Optional (taken from input files by default)

Format Floating point number

Unit Nanometer per pixel

Description Sampling of the object and empty holograms. The number given by this parameter corresponds to the size of a single pixel of the holograms. If this parameter is not given, the sampling from the image files is taken. Otherwise this parameter overrides the sampling given in the files.

Please note that all holograms, independently of being part of the object or empty series must have the same sampling. Also only image files with samplings given in nanometer per pixel are supported. If the sampling recorded in the image files is wrong (or the file format does not provides this metadata), the `sampling` parameter must be set explicitly.

5.3.34 sideband_pos

Parameter `sideband_pos`

Type Mandatory

Format List of two floating point numbers.

Unit Pixels

Description `[X,Y]` position of the sideband in the Fourier transformed image files. When the discrete Fourier transform of the holograms is calculated and the Fourier transform is shifted such that the Fourier space origin is in the center of the transformed images (like the numpy commands `np.fft.fftshift(np.fft.fft2(image))` would do), this parameter refers to the pixel position of the sideband to be reconstructed.

5.3.35 synthesize_empty

Parameter `synthesize_empty`

Type Optional (default is `false`)

Format Boolean

Description When set to `true`, the reconstructed object hologram series is normalized by a synthetic empty hologram instead of an experimental empty hologram. The synthesized empty hologram is calculated from the provided camera distortions. If `synthesize_empty` is set, the parameters *camera_distortions* and *empty_size* must be also given. If `synthesize_empty` is set, other emtpy holograms (provided either by *empty_names* or *empty_override*) are ignored.

5.3.36 voltage

Parameter `voltage`

Type Optional (taken from input files by default)

Format Floating point number

Unit Kilovolts

Description Acceleration voltage used during acquisition of the holograms. If this parameter is not given it is taken from the holograms files. This parameter must be given explicitly, if the acceleration voltage cannot be read from the hologram files.

5.4 Modulation Transfer Function

The modulation transfer function (MTF) of the camera used for acquisition of the individual holograms is specified in parameterized form.

In the following, it is assumed the MTF is a 2 dimensional function $M(q_x, q_y)$ of the two dimensional spatial frequency (q_x, q_y). A spatial frequency of +/-0.5 gives the Nyquist frequency of the detector. The MTF consists then of two parts, one due to the binning into pixels, and the other part due to the beam broadening within the detector/scintillator.

$$M(q_x, q_y) = \text{sinc}(q_x)\text{sinc}(q_y) \sum_n f_n(q)$$

The effect of the binning is described by the two *sinc* functions, here defined as

$$\text{sinc}(q) = \sin(\pi q)/(\pi q).$$

The beam broadening in the above parameterization is described by a sum over functions $f_n(q)$, where

$$q = \sqrt{q_x^2 + q_y^2}.$$

These functions are specified in the parameter file as a list of terms, where each term describes one function $f_n(q)$. The terms itself are again lists, where the first element always is a string describing the kind of function and the other elements are parameters to the function.

Possible terms are:

- `["CONSTANT", A]`

$$f(q) = A$$

- `["GAUSSIAN, A, B]`

$$f(q) = A\exp(-Bq^2)$$

- `["LORENTZIAN", A, B]`

$$f(q) = A/(B + q^2)$$

As example, if the MTF of the detector is given by:

$$M(q_x, q_y) = \text{sinc}(q_x)\text{sinc}(q_y)\left[0.8\exp(-0.03q^2) + 0.2\right]$$

the parameterization as specified by the *mtf* parameter is

```
mtf = [["GAUSSIAN", 0.8, 0.03], ["CONSTANT", 0.2]]
```

Outputs

Within the parameter file the output filename is specified by the *output_name* parameter. This output file will always be a HDF5 file. The dataset names described below are prefixed by the value of the *output_prefix* parameter (which is empty by default). As example, if the value of output_prefix would be alpha_, the dataset data is saved as alpha_data in the output file. By using the prefix multiple outputs can be written to the same HDF5 file.

The output file typically holds three datasets:

The dataset data contains the averaged reconstruction of the normalized and drift aligned object holograms. If the defocus value for the individual holograms (as specified by the parameters *defocus_first* and *defocus_step*) is not zero, the holograms are propagated to the zero defocus. The dataset empty contains the averaged reconstruction of the empty holograms.

As HDF5 files do not support datasets of complex numbers, the datasets are stored as a compound data type of 8 bytes length. The compound has two members, both 32 bit floats. The first member r at byte offset 0, contains the real part of the complex numbers, the second member i at byte offset 4 the imaginary part. This convention is recognized by HDF5 python library (http://www.h5py.org) and the HDF5 Digital Micrograph plugin (https://github.com/niermann/gms_plugin_hdf5).

The dataset variance contains the estimate of the per-pixel variance of the object hologram series (drift aligned and propagated to zero defocus), this is stored as 32 bit float.

The datasets data and variance are not present, when the object hologram reconstruction is disabled by omitting the *object_names* parameter. The dataset empty is not present, when the empty hologram reconstruction is disabled by omitting the *empty_names* parameter.

If the parameter *output_series* is set, additionally a group series is present in the output file, which contains datasets 000, 001, 002, ... These contain the reconstructions of the individual holograms in the object series. The dataset 000 refers to the first hologram (as specified by the parameter *object_names* and *object_first*). The consecutive numbers refer to the consecutive holograms in the series. These are also stored as complex valued datasets.

If the parameter *output_aligned* is set, additional a group aligned_rois is present in the output file, which contains datasets 000, 001, 002, etc. These contain the region of interest (parameter *roi*) as tracked across the object series in the raw-alignment step. The datatype of the dataset is typically the datatype used in the image files.

Additional the datasets/groups have attributes, which contain further metadata. Not all the following attributes are present in all datasets. This list is incomplete.

Name	Type	Description
holoaverage_version	String	Version number of holoaverage used for averaging.
holoaverage_param	String	Parameter string passed to holoaverage (JSON).
align_roi	List of 4 Ints	Region of object holograms used for raw alignment (see *align_roi*)
binning	List of 2 Ints	(X, Y) Detector binning of series
carrier(nm-1)	List of 2 Floats	Spatial frequency of the reconstructed side band in 1/nm
convergence	List of Floats	Total squared residual for each iteration of averaging procedure
defocus(nm)	List of N Floats	Defocus of the individual holograms (after alignment) in nm
detector	String	Name of detector (according to image files)
dim_offset	List of 2 Floats	(X, Y) offset (according to image files)
dim_scale	List of 2 Floats	(X, Y) sampling
dim_unit	List of 2 Strings	(X, Y) units for `dim_scale` and `dim_offset`
error	List of N Floats	Squared residual between individual reconstruction and average
factor	List of Nx2 Floats	(real, imaginary) global amplitudes for individual holograms
microscope	String	Name of microscope (according to image files)
raw_shift	List of Nx2 Ints	(X, Y) shift of individual holograms in pixels after raw-alignment
reconstructionCut-Off2(nm2)	2x2 Floats	Squared cut-off frequency matrix (in 1/nm^2; see *cut_off2*)
reconstructionMaskType	. . .	Mask type as used for cutoff (see *filter_func*)
roi	List of 4 Ints	Reconstructed region of object holograms (see *roi*)
shift(nm)	List of Nx2 Floats	(X, Y) shift of individual holograms in nm (after fine-alignment)
tilt(1/nm)	List of Nx2 Floats	(X, Y) tilt of individual holograms in 1/nm (after alignment)
voltage(kV)	Float	Acceleration voltage in kV

Example parameter files

7.1 Only reconstruct empty hologram

This will reconstruct only the empty hologram. This can be used in for instance in subsequent reconstructions (see other examples):

```
{
    // Leave "object_names" unset

    // Usual empty hologram input
    "empty_names" : "empty.%d.dm3",
    "empty_first" : 1,
    "empty_last" : 10,

    // Size (in px) used for reconstruction of "empty" holograms. Required.
    "empty_size" : 1536,

    // X, Y Position of side band in FFT pixels (origin is in center). Required.
    "sideband_pos" : [1011, 1091],

    // Output file name (will be HDF5 file). Empty hologram will be in dataset "empty"
    "output_name" : "my_empty_reco.hdf5",

    // Mask type (see FilterFunction for details). Defaults to "EDGE"
    "filter_func" : ["BUTTERWORTH", 14],

    // Reconstruction cutoff in 1/nm
    "cut_off" : 3.0,

    // Parameterization for MTF
    // "mtf" : ...
}
```

7.2 Use pre-reconstructed empty hologram for normalization

This example uses a prereconstructed empty hologram for normalization (also see *Normalization*):

```
{
    // Object holograms (as usual). Here series of DM3 files
    "object_names" : "hologram.%d.dm3",       // Filename format
    "object_first" : 1,                       // Index of first
    "object_last" : 5,                        // Index of last (inclusive)

    // Use pre-reconstructed empty hologram for normalization, here dataset from HDF5␣
↪file.
    "empty_override" : "somefile.hdf5?dataset=empty",

    // Reconstruction parameters
    "object_size" : 1536,                     // Reconstruction size in px
    "sideband_pos" : [1011, 1091],            // X, Y Position of side band in FFT␣
↪pixels (origin is in center).
    "cut_off" : 3.0,                          // Reconstruction cut off in 1/nm
    "filter_func" : ["BUTTERWORTH", 14],      // Mask type

    // Optional reconstruction region (L, T, R, B). Defaults to full region.
    //"roi" : [166, 388, 1701, 1923],

    // Output file name (will be HDF5 file). Required.
    "output_name" : "my_reco.hdf5",

    // Parameterization for MTF
    //"mtf" : ...
}
```

7.3 Use predetermined camera distortions for normalization

This example creates an synthetic empty hologram for normalization. The synthetic empty hologram is creacted from predetermined camera distortions (also see *Normalization*):

```
{
    // Object holograms (as usual). Here series of DM3 files
    "object_names" : "hologram.%d.dm3",       // Filename format
    "object_first" : 1,                       // Index of first
    "object_last" : 5,                        // Index of last (inclusive)

    // Enable synthetic empty holograms
    "synthesize_empty": true,

    // Two datasets, same size as holograms, with displacements in px for each pixel
    // First in X direction, Second in Y direction
    "camera_distortions": ["camera.hdf5?dataset=dx", "camera.hdf5?dataset=dy"],

    // Reconstruction parameters (as usual)
    "object_size" : 1536,                     // Reconstruction size in px
    "sideband_pos" : [1011, 1091],            // X, Y Position of side band in FFT␣
↪pixels (origin is in center).
    "cut_off" : 3.0,                          // Reconstruction cut off in 1/nm
```

```
    "filter_func" : ["BUTTERWORTH", 14],      // Mask type

    // Optional reconstruction region (L, T, R, B). Defaults to full region.
    //"roi" : [166, 388, 1701, 1923],

    // Output file name (will be HDF5 file). Required.
    "output_name" : "my_reco.hdf5",

    // Parameterization for MTF
    //"mtf" : ...
}
```

7.4 Only reconstruct single hologram (no averaging)

This example reconstructs a single (object) hologram, and normalizes with a single empty hologram:

```
{
    // Output file name (will be HDF5 file). Required.
    "output_name" : "output.hdf5",

    // List with just a single filename
    "object_names" : ["my_hologram.dm3"],

    // Empty hologram (list with single filename)
    "object_names" : ["empty.dm3"],

    // Reconstruction parameters
    "object_size" : 512,
    "sideband_pos" : [749, 1019],
    "filter_func" : ["BUTTERWORTH", 14],
    "cut_off" : 1.0
}
```

Changelog

8.1 Version 1.1.8

- Dropped Python 2.X support
- Fix deprecated functions
- Added CI/CD support for GitLab

8.2 Version 1.1.7

- Output file attribute 'reconstructionCutOff2(nm2)' was set erroneously to isotropic cut off. Now squared anisotropic cut off is output as stated in the documentation.

8.3 Version 1.1.6

- Parameters 'object_names' and 'empty_names' now also allow file name lists instead of the printf-based syntax

8.4 Version 1.1.5 (unreleased)

- New parameter 'align_cut_off' allows specification of a dedicated raw alignment cut_off

8.5 Version 1.1.4

- Fixed bug, which caused the 'path' parameter to be ignored for some input files
- Fixed bug in configurable file importer, which occured when the deprecated syntas was used

- New parameter 'cut_off2' introduced, which allows nonisotropic reconstruction masks

8.6 Version 1.1.3

- Fixed a bug, which might have caused slight phase gradients over the image, if a ROI was used
- Display hint for use of PyFFTW
- Acceleration voltage for defocus propagation now taken from parameter file instead of image files, when present

8.7 Version 1.1.2

- Single hologram reconstructions do not require printf-format codes in input file paramters
- Include parameters in output file
- Include holoaverage version in output file
- Display global holograms amplitudes instead of intensities during reconstruction
- Made syntax of configurable file importer more URL-like.

8.8 Version 1.1.1

- Deprecation warnings are displayed by default
- Parameter 'output' was renamed to 'output_name'
- New parameter 'output_prefix' allows control over dataset names in output
- Added program overview to documentation
- Configurable image file importer
- Raw input image files
- New parameter 'enable_raw_align' to allow control over raw alignment
- Commandline option to print version number
- Improved handling of DM3 tag names encoding

Bibliography

[Meyer02]   R.R. Meyer, A. I. Kirkland, O. W. Saxton, A new method for the determination of the wave aberration function for high resolution TEM, Ultramicroscopy 92 (2002) 89

[Lehmann02]   M. Lehmann, H. Lichte, Tutorial on Off-Axis Electron Holography, Microsc. Microanal. 8 (2002) 447

[Niermann14]   T. Niermann, M. Lehmann, Averaging scheme for atomic resolution off-axis electron holograms, Micron 63 (2014) 28

Index

H

holoaverage.holoaverage() (*built-in function*),