M_SPEC Python manual

Martin Dubs, FMA

Summary

This manual explains the use of a Python script for the analysis of video meteor spectra. The theory behind it has been described in (Dubs and Schlatter, 2015): meteor_spectroscopy_wgn43-4_2015.pdf).
Until now, the method had been applied successfully to analyse meteor spectra obtained with high sensitive video cams such as the Watec 902 H2 ultimate or colour cameras such as the Sony alpha 7S, which allow video recording. Unfortunately the method required the use of different software for the various steps of the spectra processing, since standard spectroscopy was not adapted to the special requirements of meteor spectroscopy with a fast moving spectral source and nonlinear spectra caused by oblique incidence of the light to be analysed. By combining the different steps of the analysis in a single program, using Python, the analysis should be easier, faster and more reliable. Processing meteor spectra requires the calibration of the camera – lens – grating combination, in order to obtain transformation parameters for the orthographic transformation. This has to be done once for each different camera setup and can be done also with this program.
The processing steps include extraction of the images from the video file, defining a background image, subtraction of background, transformation of the images to orthographic projection, resulting in linear spectra. These are superposed (registered) and converted to 1-dimensional spectra by summing over several rows. The intermediate images of the processing are stored as FITS-images. The advantage of this format is that it also contains information about the image creation and processing, important for spectra of professional quality. The last steps are wavelength calibration and plotting of the final spectra. The procedure follows the processing as described in (Dubs and Maeda, 2016: Calibration_of_Meteor_Spectra_Dubs_IMC2016.pdf).
At the moment some processing steps are not implemented yet, such as the flat correction and the instrument response. These will be added later if needed.
The program is still in a development stage, so the user is asked to report problems and errors, for correction in later versions. The present version 0.9.22 has changed from a pipeline based approach to a windows GUI, using PySimpleGUI (https://pysimplegui.readthedocs.io/en/latest/) for creating the windows interface. This allows more flexibility in solving the tasks of meteor spectra processing. The style is similar to ISIS, well-known and widely used astronomical spectroscopy software.

Introduction

For amateurs, the analysis of meteor spectra has been quite difficult. Standard spectroscopy software is not particularly suited for meteor spectra. Therefore different software had to be used for parts of the processing, which discouraged potential users and made the analysis quite complex (see the manual: processing-meteor-spectra-v151.pdf for the old method). Most of the processing steps are the same for different meteors, therefore a program which combines the different steps can simplify the meteor spectra processing. The program is written as a Python script with graphical user interface (GUI).
Python was chosen, because

- it contains all the necessary tools to do the analysis
- it finds widespread use in the astronomy community
- it is free
- it runs on different platforms

I was inspired to use Python by Giovanni Leidi, who was giving a talk about the use of Python for the analysis of spectra in the spectroscopy workshop at OHP 2018 (https://www.shelyak.com/ohp-spectro-star-party-2018/).
For the processing of meteor spectra it is necessary to calibrate the equipment beforehand. This is included in this version of the program.
Note of caution: I am new to Python, so the script presented here may not be the best solution. Some things have been done in a complicated way, copying examples from different sources and trying to make it work. I try to improve it for clarity and safety of operation. Therefore I hope you will suggest improvements.
The manual is divided into four main sections and additional features. First a general description of the processing steps is given, which are followed by any processing software. The next section describes the installation of Python, such that the specific Python scripts for analysing meteor spectra can be run. In the third section the determination of the parameters for the orthographic transformation is described. In the fourth section a detailed description how to process a video file into a calibrated meteor spectrum is given. At last some hints and tricks are given for efficient use of the script.
M_spec is intended for a first analysis of meteor spectra. For more detailed studies and reanalysis of processed meteor spectra standard spectroscopy such as ISIS or Vspec may be used. The registered images and spectra produced by the Python script are in a format, which is suitable for these spectroscopy programs.

Processing of meteor spectra, overview

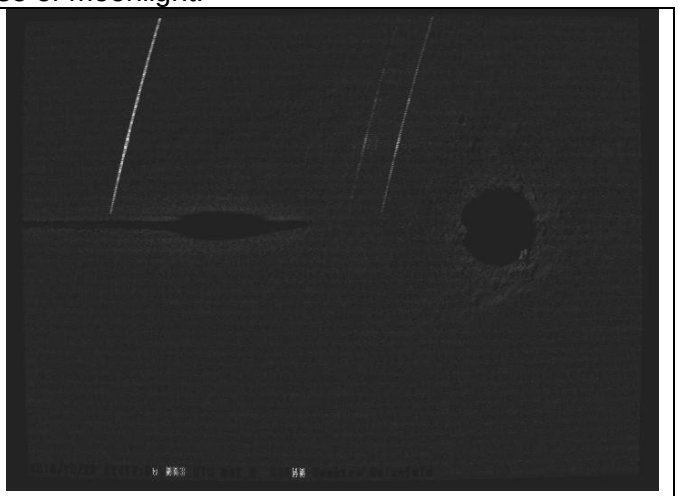
Image extraction from video sequence

The starting point for the analysis is a video file. Typically these are recorded with UFO capture, a program to detect meteors in real time. The program uses a pretrigger to record one second before the meteor appears until the end of the meteor. This video is separated into an image sequence which is stored as *.BMP images. In Python this done with a call to ffmpeg (https://www.ffmpeg.org/). For the Watec camera, the images are either single frames or the interlaced frames can be separated into two fields, each containing a half frame (even and odd scan lines). These have to be arranged in the correct order (bottom field first or top field first). The use of fields with double image rate is useful for fast meteors, having a large velocity component in the dispersion direction, which reduces the spectral resolution.
For colour videos, there are two processing methods, in colour or black and white. Processing in colour produces nice colour images of the spectra and makes the assignment of spectral lines easier. On the other hand it is three times slower and uses lots of disk space. For the extraction of 1d-spectra the colour images have to be converted to b/w images. Therefore it is usually more convenient to do the whole processing in b/w. Another possibility to speed up the processing is by binning the images in the video extraction in case of 4k videos (only 2x2 binning supported at present).

Background image

An important step in the processing of meteor spectra is background subtraction. The video records not only the meteor spectrum, but also other light sources such as sky background, starlight and light pollution from different sources. Most of this does not vary rapidly and can be subtracted before further processing of the spectra.
The background image is made by averaging a number of images without the meteor. For this, the first images of the video can be used. The pretrigger in UFO capture is adjusted normally to about one second. At a frame rate of 25 images per second this means that the first 20 images can be used for the background, leaving some safety margin if the detection algorithm misses the first few meteor images. Averaging of 20 images reduces the statistical noise in the images by a factor square root of 20 or about 4.5. In the case that the video is converted to fields the image rate is doubled and also the number of images for the background can be doubled.
This background image is subtracted from the remaining images containing the meteor spectra.
The effect of the background subtraction in the case of moonlight:

Peak image of meteor images as displayed by UFO capture (no background subtraction)



Peak image of meteor images after background subtraction and orthographic transformation (see next section)

Image transformation to the orthographic projection

As described in the above mentioned paper: meteor_spectroscopy_wgn43-4_2015.pdf, the recorded spectra have a nonlinear dispersion and are curved differently in different image areas. These spectra are linearized and made parallel by a transformation to an orthographic projection. This transformation can also correct lens distortions. In the same processing step the spectra can be rotated, so that the resulting linear spectra are parallel to the image x-axis. The transformation is described by an axial symmetric radial transformation of the form $r = r' * (1 + a3*r'^2 + a5*r'^4)$ (plus higher terms in the future), where r and r' are polar coordinates in the original and transformed images. The equation describes the inverse transformation, since for every image point in the resulting image the original coordinates have to be calculated. The polar coordinates are measured from the coordinates of the optical axis (x0, y00 and the polar angle is corrected by the spectrum rotation. The transformation is done in three steps, conversion to polar coordinates, radial and angular transformation, conversion back to Cartesian coordinates. For the interested reader the Python code is given here:

```
x, y = xy.T
x0, y0 = center
y0 = y0 * yscale # the center in the original image has to be scaled as well
# y has been scaled in a previous step with resize image
rp = np.sqrt((x - x0) ** 2 + (y - y0) ** 2)
phi =  np.arctan2(y - y0, x - x0) + rotation
r = rp*(1 + rp**2*(a3+a5*rp**2))
xy[..., 0] = x0 + r * np.cos(phi)
xy[..., 1] = y0 + r * np.sin(phi)
```

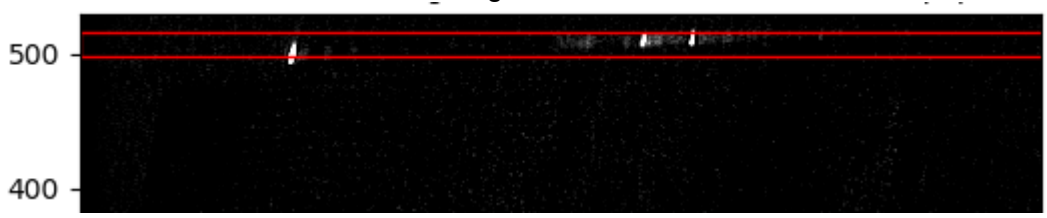
The parameters x0, y0, a3, a5 and rotation have to be determined beforehand in a one time calibration for each camera – lens – grating configuration. In addition, in the case of non-square pixels a transformation (stretching in y-direction) by a factor yscale has to be applied before the transformation to orthographic coordinates. When fields (half frames) were extracted from the interlaced video, the images have to be scaled by an additional factor 2 in y-direction, in order to compensate for the missing lines. For the laser calibration, the script M_calib.py has been integrated into m_spec (see tab “Laser Calibration”). The results of this calibration can be directly used for the processing of meteor spectra, under the condition that the camera setup has not been changed after the calibration.
The radial transformation conserves the dispersion or image scale in the center of the image (at x0, y0). The dispersion disp0 can be used to convert the spectra from pixel to wavelength (more about this see below).

Image registration

In stellar spectroscopy, the position of the star image can be kept in a fixed place, usually defined by a narrow slit, which allows a constant calibration of the spectrum. With meteor spectra this is hardly possible, the meteor appearing without warning and moving fast enough to make capture and tracking impossible, at least for the amateur. Therefore the spectra may be recorded anywhere in the image plane, in a different position from image to image. For the analysis they can be shifted in such a way that they overlap. This works only because they are all parallel and have the same linear dispersion. Aligning therefore the zero order will automatically align the whole spectrum. Summing these aligned (registered) spectra allows to improve signal to noise ratio. The aligning is done by measuring the position of a prominent line (the zero order if visible, or another prominent line if the zero order is outside the image area) in each image. This position is determined by fitting a Gaussian peak shape to the line and using the centre as line position. The line position in the first spectrum is used as a reference, to which all the following spectra are shifted.
In the next step a suitable number of images are added, which results in a 2-D spectrum image, which can be analysed further in the following steps.

Correction of tilt and slant

The deviation of the meteor spectrum from the horizontal axis is called tilt. For the extraction of the spectrum from the 2-dimensional image a number of rows corresponding to the width of the spectrum are added. This works correctly only if the spectrum is aligned parallel to the rows. During calibration this angle has been determined and the images were rotated so that the spectra should be aligned parallel. In the case that the grating has been rotated since the calibration, the spectra will not be parallel to the rows. This can be corrected by applying a "rotation" to the added spectrum. Mathematically a shear is applied; the columns are shifted vertically in proportion to the column number:
dy = tilt * (x – x0)
xo is selected in the centre of the image.



Spectrum before tilt correction

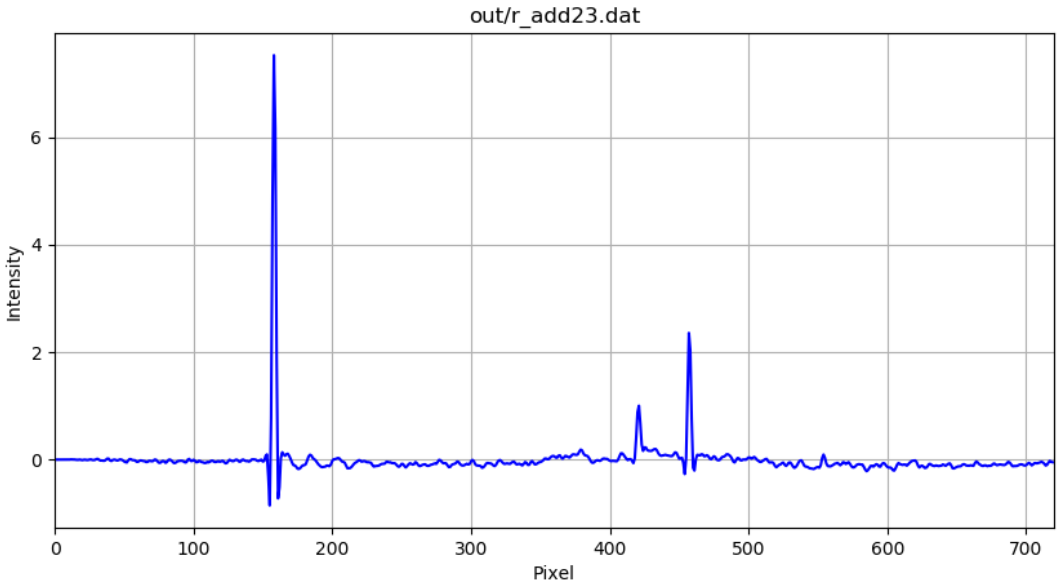
Spectrum after tilt correction, tilt = -0.04, spectrum "rotated" clockwise.
The same effect would be obtained by changing the rotation angle in the configuration file by increasing rot by 0.04 (angle measured in radians)

Notice that the spectral lines are not vertical. This is caused by the movement of the meteor. The deviation from the vertical is called slant and is defined as the slope
Slant = dx/dy
For the correction of the slant the rows are shifted horizontally in proportion to the row number. Dx = slant * (y – y0). Y0 is chosen as the center of the selected rows for adding the spectrum. Therefore the spectrum is shifted only minimally by the slant operation.

Spectrum after slant correction, slant = -0.2, spectral lines "rotated" anti-clockwise.
It is advisable to perform the operations in this order and adjust the row selection before the slant correction.
With these corrections, the processing may continue with the spectrum extraction.

Spectrum extraction

The sum image can be converted to a 1-d spectrum by adding a suitable number of rows of the image which contain the spectrum. The number of rows is determined by the width of the spectrum. Ideally one wants to add all the rows which contain the information but avoid the rows which contain only background noise. The choice is somewhat arbitrary and depends on the skill of the operator, but not too critical. There exist methods of optimal extraction of the spectrum by measuring the width of the spectrum and signal to noise ratio in comparison with the background. This has not been implemented yet.
The result of this summing over rows is a 1-d spectrum: intensity versus pixel.



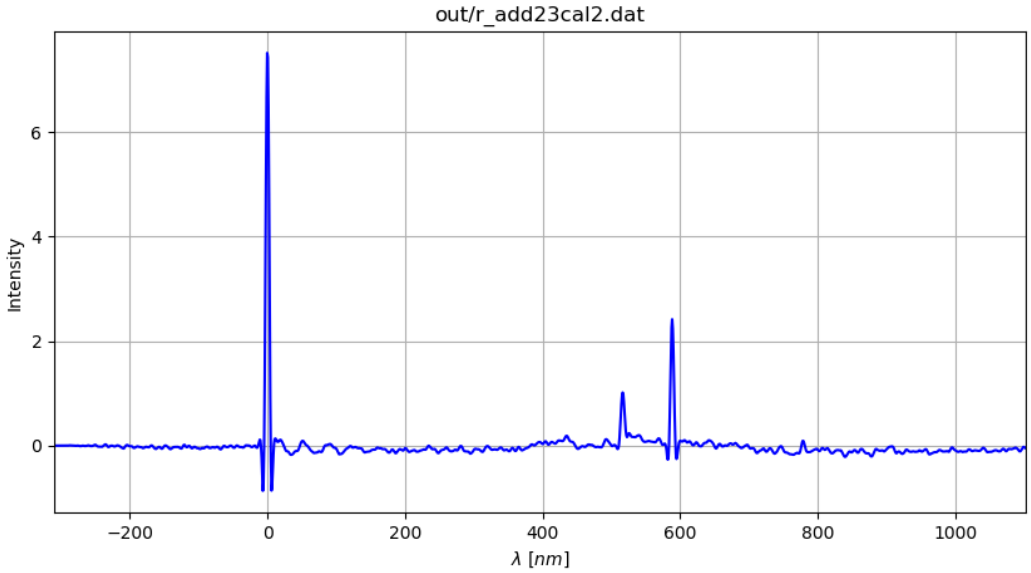
With the known dispersion disp0 [nm/pixel] wavelength intervals can be calculated by measuring separations in pixels of prominent features (zero order, Na-line, Mg-line etc.).This helps in a course assignment of spectral features.

Spectrum calibration

Apart from the instrument calibration, which corrects for lens distortion and transforms to the orthographic projection with constant dispersion, each spectrum has to be converted to a proper wavelength scale, since the meteor and its spectrum can appear anywhere in the image
In principle, the meteor spectrum can be calibrated in wavelength by setting the zero point at the zero order and convert the pixels to nm by multiplying the distance in pixels of a spectral line from the zero order by the factor disp0, the (inverse) dispersion. In practice, some changes in alignment of the grating, changes of focus and focal length (for zoom lenses) may occur and it is advisable to calibrate the spectrum with the help of known spectral lines of the meteor. Although in theory a linear function should fit the spectrum, in order to correct for small calibration errors a second order polynomial may give a better result. Before fitting a higher order polynomial it is advisable to check if a lower order polynomial works as well. Checking with more lines than required for a fit is always advisable in order to be able to eliminate measurement errors or assignment of the wrong lines.

Plotting the spectra

Once calibrated, the spectrum can be plotted and saved as *cal.dat file. This is a common data format, a text file with two columns for wavelength and intensity, each data point in a separate row. Some programs which read spectra expect a constant wavelength interval; therefore the spectra are resampled with a constant wavelength interval, about half as large as the average original sampling interval. Quadratic interpolation is used. This helps to prevent loss of information with the resampling.

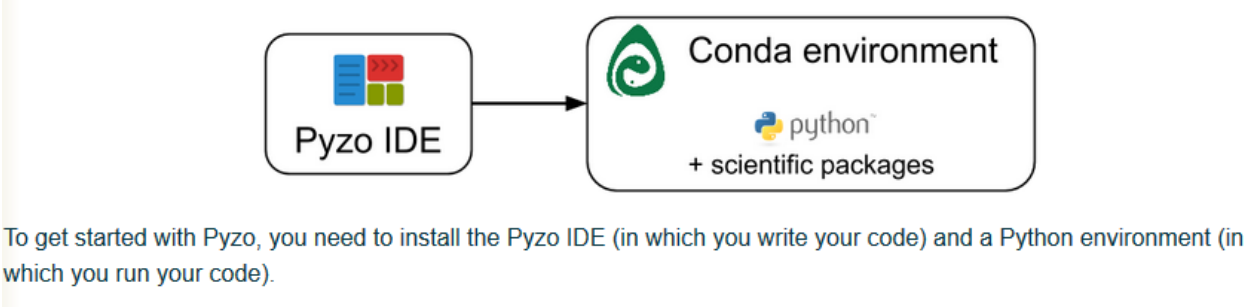


Calibrated spectrum, the zero order peak is at 0nm, the Mg line at 515.5 nm and the Na line at 589 nm are also clearly visible.

For the calibration in this example a linear fit of the zero order and the two spectral lines of Mg and Na were used.

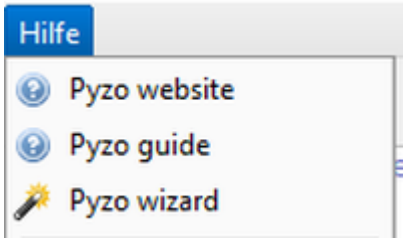
Python Installation

Python comes in different versions. Apart from the base version it requires additional packages, in particular:
Numpy
Astropy
Etc.
Some versions are not quite compatible. For that reason it is highly recommended to install a pre-packaged version, such as Anaconda, unless you are familiar with Python installing packages.
For running and editing Python scripts you need a development environment (IDE). I use PYZO, as recommended by Giovanni Leidi. It is installed from http://www.pyzo.org/start.html.



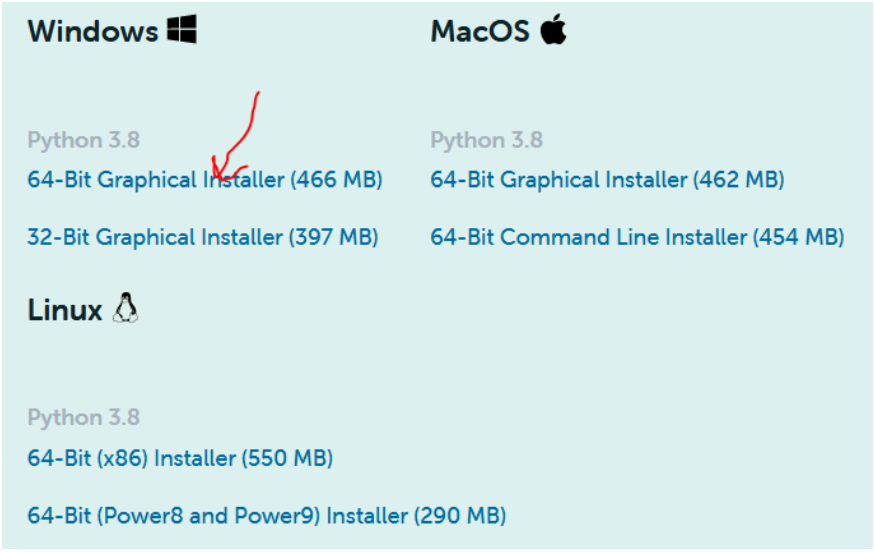
To get started with Pyzo, you need to install the Pyzo IDE (in which you write your code) and a Python environment (in which you run your code).

Download and install it. Pyzo recommends installing Miniconda afterwards. I had some problems with Miniconda and other packages; therefore I recommend using Anaconda instead. This is a large package, but it contains almost all the required libraries for running the script m_spec.
The installation of Anaconda is described next. You are free to use whatever version of Python you like, but do not complain if it does not work **(do not use Python 2.7; it does not work with my script). I recommend using Python 3.7**, which was used for the present version.
If you are not familiar with Pyzo, you may look at the Pyzo wizard. (There you can also change the language of the editor). Also the guide or website are quite instructive:



Installation of Anaconda

The advantage of using Anaconda is that it comes with almost all of the required packages needed for running the spectroscopy scripts. This saves a lot of trouble with different versions of Python libraries.
Go to https://www.anaconda.com/download/
Install the latest version (64-bit in my case). The 3.6 or 3.7 version also works, if already installed:



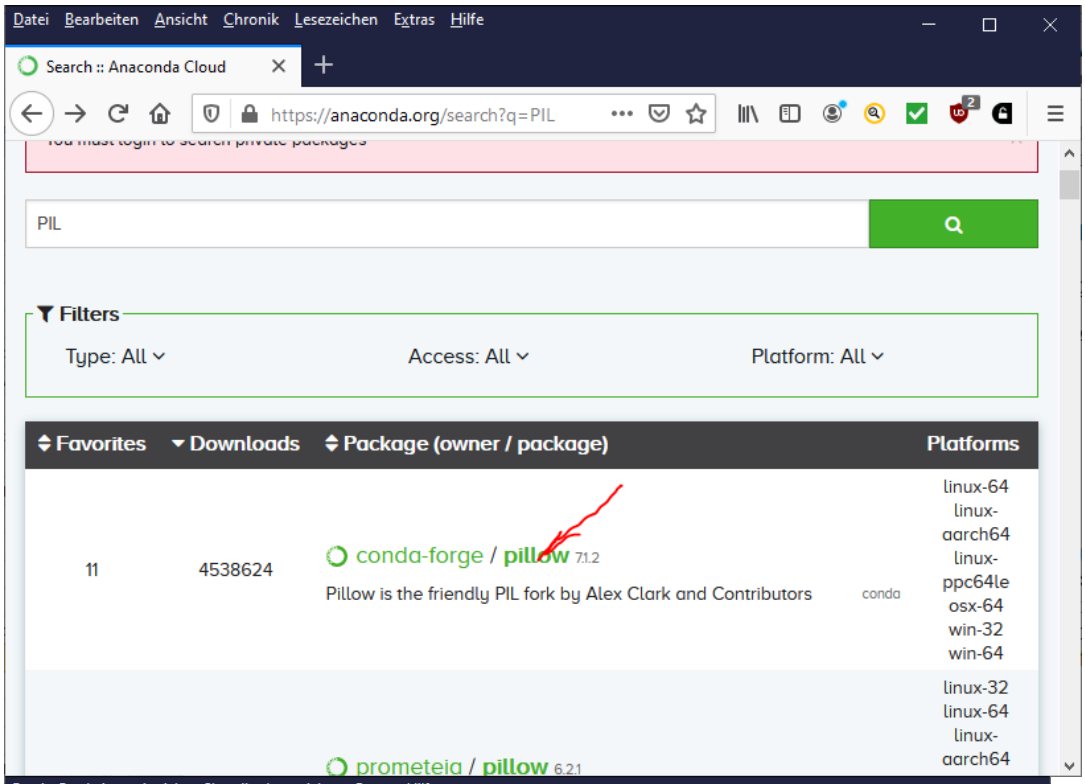
Follow the tips for installation
https://docs.anaconda.com/anaconda/install/windows

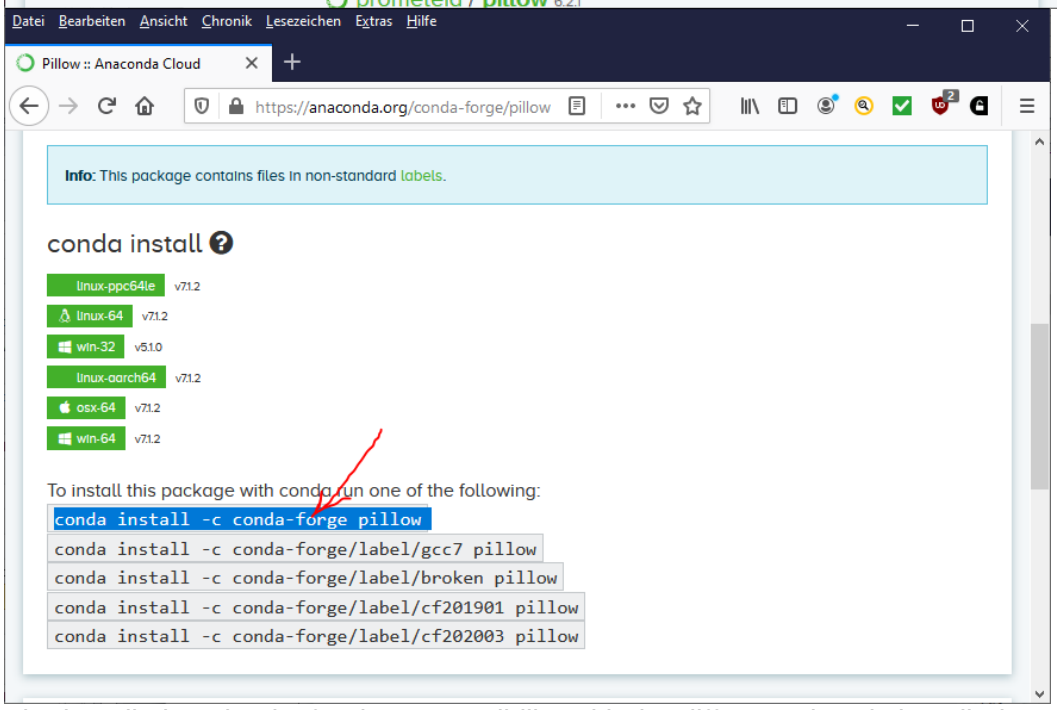
- Required system architecture: Windows- 64-bit x86, 32-bit x86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bit Power8/Power9.
- Install Anaconda in C:\, otherwise you may have problems finding it on your computer (makes it easier to write path to anaconda, but you are free to install it elsewhere).

Note: Install Anaconda to a directory path that does not contain spaces
(such as: Program Files) or Unicode characters.
This will take a while. At this point, if everything has gone well the Pyzo IDE should recognize without telling it the environment Anaconda you chose and you can start writing in python.

Load libraries

If the scripts do not run, it may be that a library is missing. You have to load libraries, which are not contained in your Python distribution.
With Anaconda the command is, e.g. for the installation of lmfit:
>>> conda install -c conda-forge lmfit
or
>>>conda install -c conda-forge pillow (for the image library PIL, included in Anaconda 3 2020.07)
You type it into the shell window on the right size of Pyzo.
You find the installation command by going to the Anaconda cloud https://anaconda.org/ and search for “PySimpleGUI” or “PIL”.



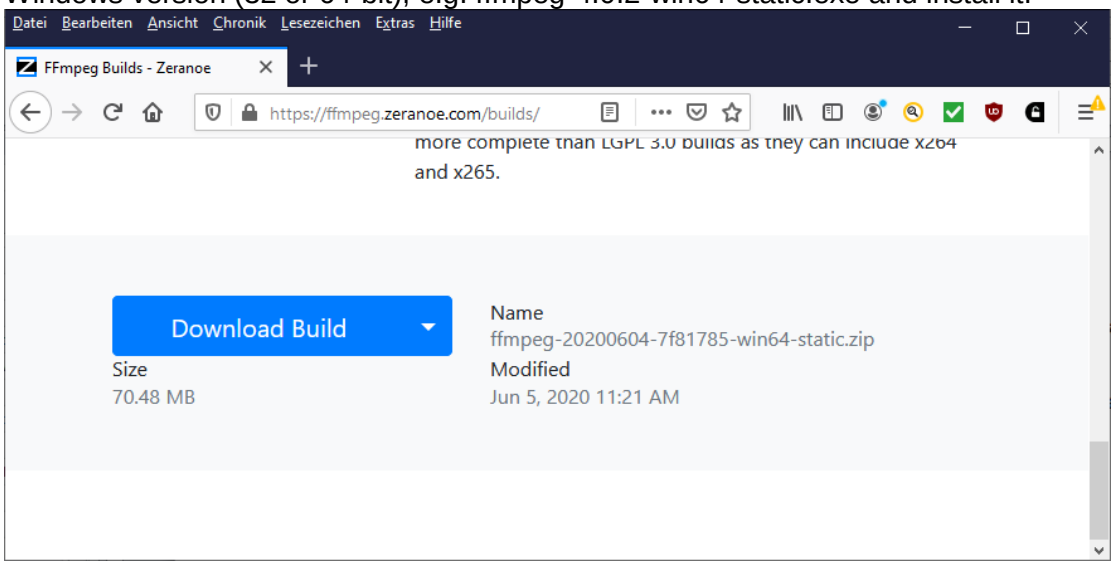


The installation checks for the compatibility with the different already installed packages and makes the necessary updates.
In addition you need to copy other scripts and files (m_specfun.py, m_set.ini etc.).
Fortunately most of the needed library packages are already contained in Anaconda, namely:

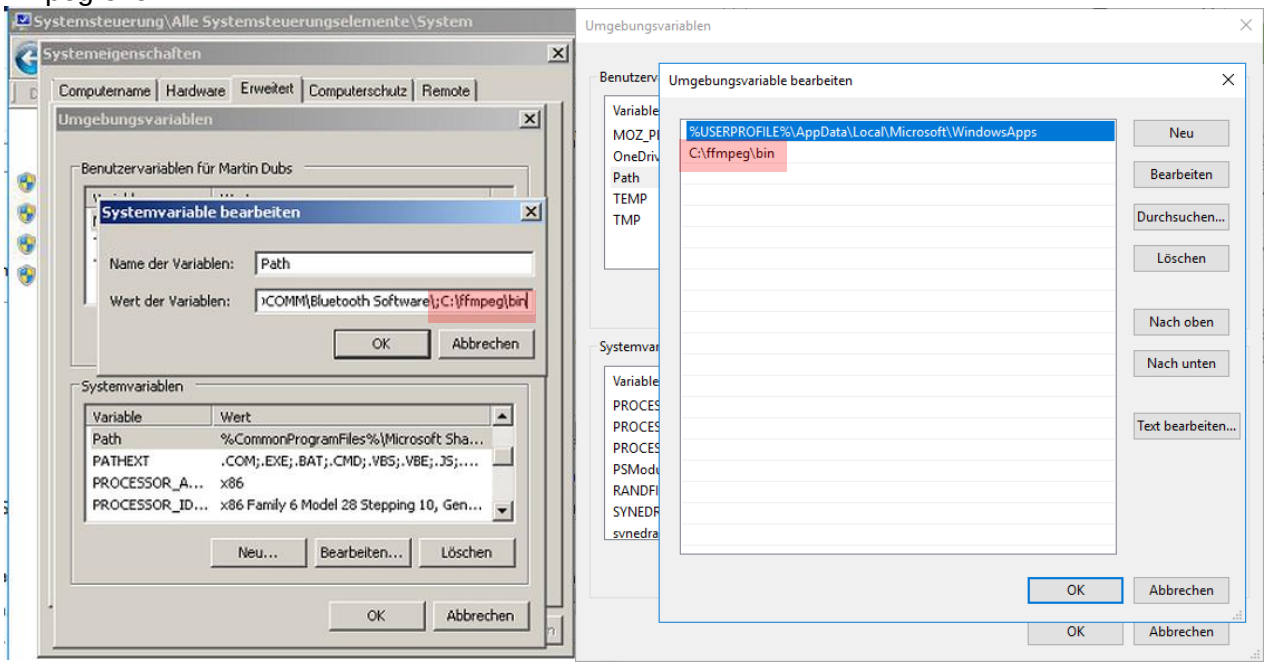
- numpy
- matplotlib
- astropy
- scipy
- skimage (is contained in scikit-image)

Install ffmpeg

The script uses ffmpeg.exe for the conversion of the AVI-files to BMP-images. Download ffmpeg.exe (e.g. from: https://ffmpeg.zeranoe.com/builds/) select the correct Windows version (32 or 64-bit), e.g. ffmpeg-4.0.2-win64-static.exe and install it.



Add the ffmpeg.exe directory to the Windows system path variable. This is slightly different in Windows7 and 10. In system variable, double click on the path and add the location of ffmpeg.exe:

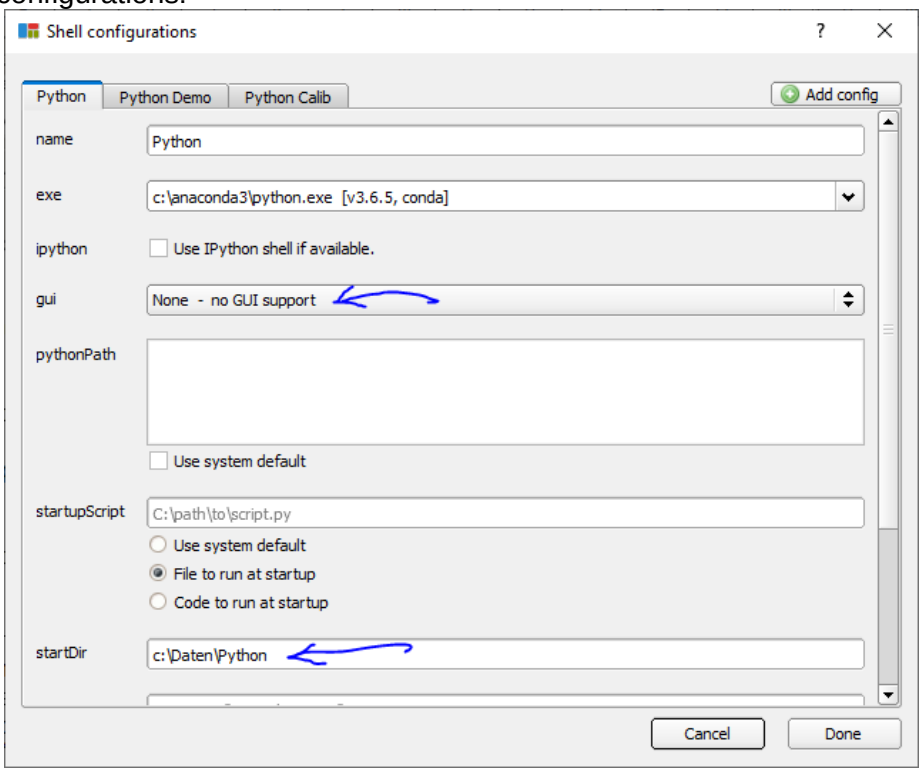


Setup Pyzo

The python script should run in different environments. I show how to run it with Pyzo, which is fairly simple but quite convenient. I am new to python, so I cannot tell how it compares with other environments.
(For writing and editing scripts I use PyCharm, another free tool https://www.jetbrains.com/pycharm/download, which has advanced features such as error correction, renaming variables, version control etc. I do not recommend it for beginners).
When starting Pyzo, one can select several windows. Of these the shell is the most important, there the script is run. Also useful is the editor window, where one can make changes to the

scripts and the workspace, where one sees the actual values of variables while running the script. Other windows can be selected under Tools:

When starting Pyzo, one should check the configuration, under Shells – Python… - edit shell configurations:



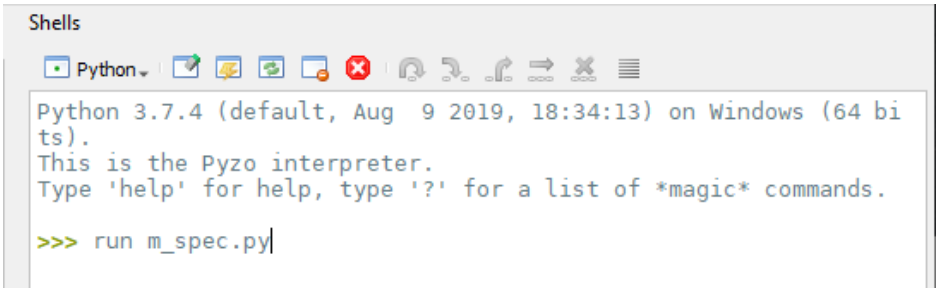
Important are the path to the python.exe file, which with correct installation is automatically set. Gui should be set to none, otherwise some strange things happen with the plots. The start directory should be set to the directory where the python scripts such as m_spec.py and m_specfun.py are located, C:\Daten\Python in my case. The data and results I put in subdirectories of this “root” directory. In this directory some other files are needed, i.e. mset.ini, the configuration created in the calibration script (See: Calib_Python_manual.pdf).
Important: Before running this script, run the laser calibration m_calib.py (or l_calib6.py) or edit a configuration file mset.ini with the correct parameters for your camera – lens - grating configuration (e.g. if you already made your calibration with EXCEL). It is good to start with the calibration first, until you are a bit familiar with Python.

Installing Python in LINUX

The installation under Linux (Ubuntu 18.04) without Anaconda works also:
This has not been tested yet for m_spec, but worked well for m_calib according to a Linux user of Python.

Starting the script

There are two methods to start a script. Either you open the script, at present m_pipe6.py, in the editor window, select it and start it with: Run – Run file as script (Ctrl shift E)
The second method is to type in the shell window after the prompt >>> run m_spec.py ↵ (enter)



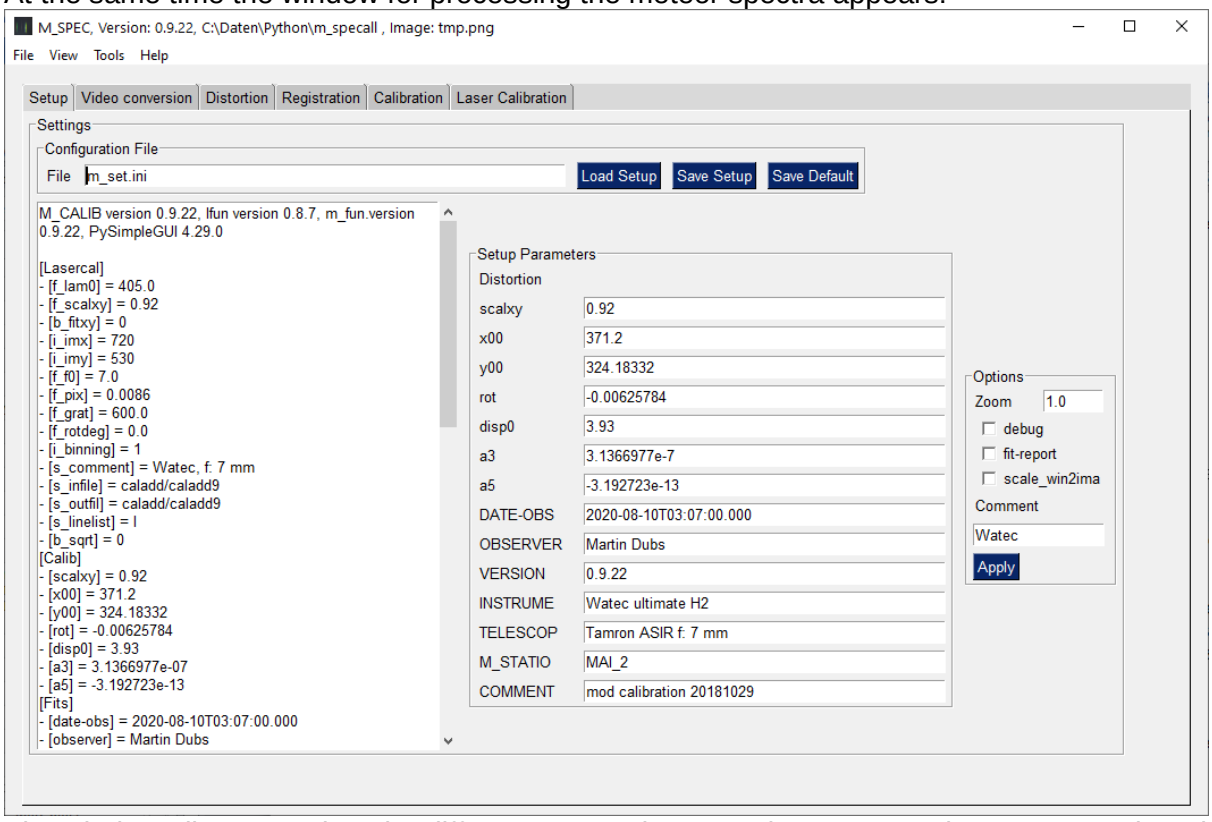
If you do not see the prompt, you are running something already, then it may be necessary to restart with: .
When you start the script for the first time you will probably see some error messages of the type … package not found.
Required packages are included in Anaconda:

- numpy
- matplotlib
- astropy
- skimage (contained in scikit-image)
- pathlib
- scipy

If I have forgotten something, look in Anaconda if you can find it and install it (send a little note to https://meteorspectroscopy.org/contact/ so that I can fix this manual).
Once all packages are installed, you should be able to start the script and the following message appears:

```
>>> run m_spec.py
PySimpleGUI 4.29.0
version m_spec, m_specfun, m_plot 0.9.22 0.9.22 0.9.21
```

At the same time the window for processing the meteor spectra appears:

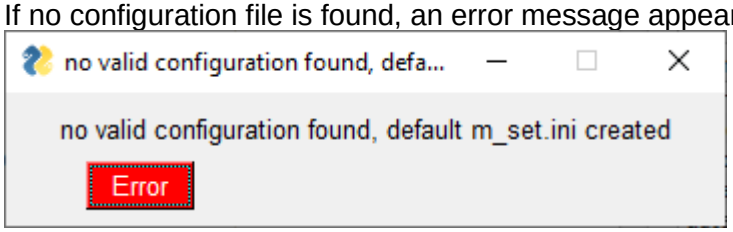


The window allows to select the different processing steps in separate tabs. At startup the tab Setup is shown, where the parameters from the configuration file are displayed. By default, the

configuration file m_set.ini from the work folder (where m_spec.py resides) is loaded. This file Is also stored when you leave the program with File – Exit in the menu or you can store changes to it with “Save Default”. This file contains many program settings and, most important, also the distortion parameters from a calibration with s_calib.py or m_calib.py. In the case you have different cameras or setups, it is advisable to store them in separate configuration files.
The content of the configuration file you see in the left memo. Since the same configuration file is used for the calibration and for the spectrum processing, it starts with parameters for the laser calibration, followed by the distortion parameters and fits-header values and finally the settings of m_spec under [Options].
The most important setup parameters are shown in the center frame. Most values can be edited, some values such as DATE_OBS, VERSION and M_STATIO are updated by the program in case you use UFO Capture for recording the meteor spectra (VERSION shows the actual version of the script as written to the fits-header)

Fits-header

A good practice in astronomical photography and spectroscopy is to record information about the image content in the Fits-header, in particular observation time and date, observer, location and equipment. In the case of the orthographic transformation also the used transformation parameters may be useful for a later check on the processing history.
DATE-OBS is a standard keyword; its value is derived from the UFO capture filename.
M_STATIO is also derived from this filename and gives the location of the camera.
D_DISP0 is the dispersion from the laser calibration. If you have a better value from actual meteor spectra, you may enter the correct value here (used for one line calibration).
Others, such as COMMENT you may edit to your taste.
If you save the changes in the default directory you can use them next time you start the script. It is good practice to save them also in the output directory, together with the other processed files, in case you would like to reprocess some files later on.
If no configuration file is found, an error message appears:



Load a valid configuration file or edit the default values with correct values.
If you use different cameras or lenses, make sure you are using the correct configuration. I use different directories for different cameras. Also make a backup of the configuration in a safe place, it is possible to overwrite or erase the file by accident, if something goes wrong.
(The script is not yet foolproof, not even without bugs).
On the right side is a frame for options. With zoom you can adjust the scale of the displayed images (works after video conversion or distortion. If it does not work correctly, save configuration and restart program. Another possibility is to deselect scale win2ima, then you can adjust the window size manually and the images should fit. If they do not fit, save the default configuration and restart the script. Check the other settings, then save the configuration under a meaningful name.
With this preparation, you are ready to use the script, either for laser calibration and the determination of the parameters for the orthographic transformations, or for the processing of meteor spectra using this transformation. We start with the laser calibration.

Laser calibration of meteor spectra

The laser calibration tab contains the calibration of meteor spectra with laser calibration images replacing the older method with IRIS for the determination of laser point coordinates and EXCEL for the fitting of parameters for the distortion model. These parameters are required for the change from image coordinates to the orthographic projection, which is used to linearize meteor spectra. It is derived from l_calib6.py, described in Calib_Python_manual_2.pdf. It can also be used as a standalone script: https://github.com/meteorspectroscopy/calibrate-spectrum.

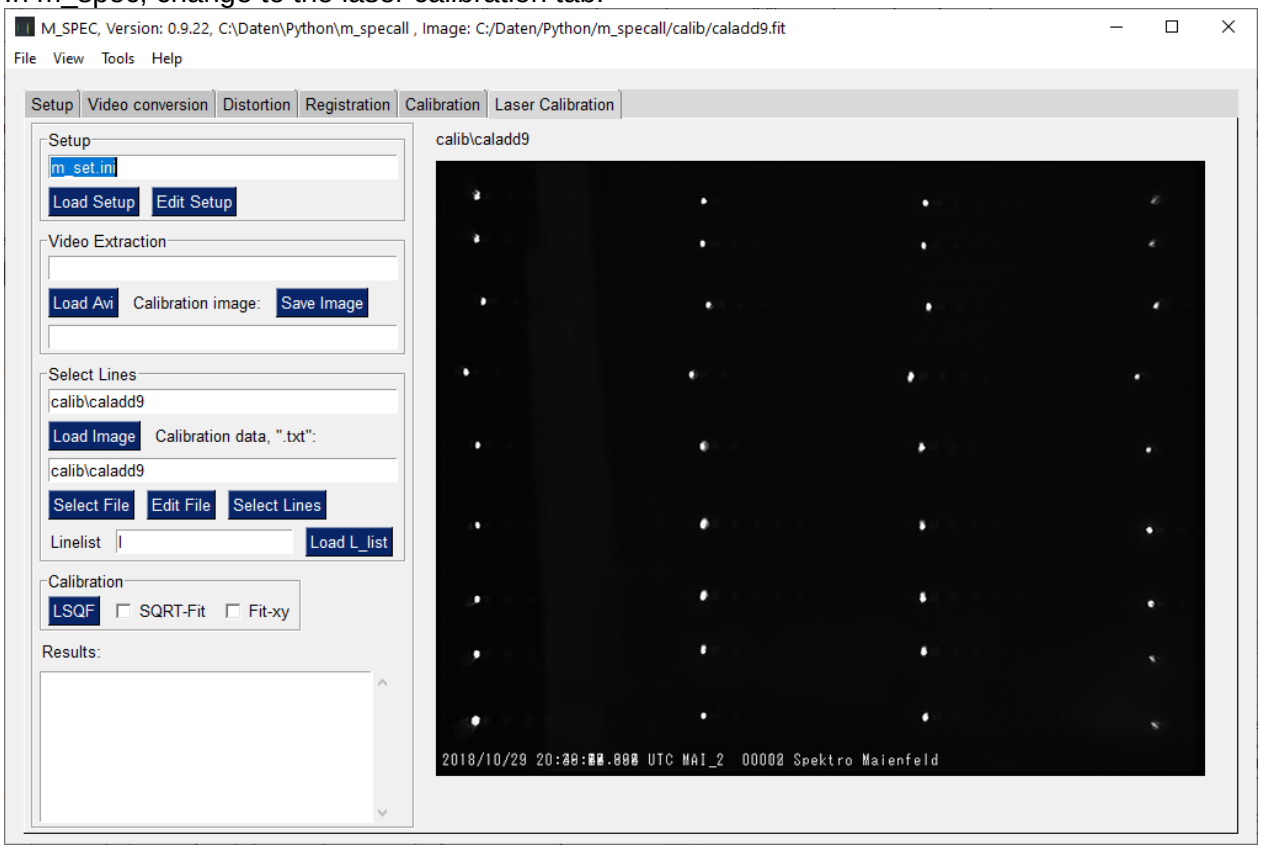
Laser calibration images

For the calibration of meteor spectra you need some calibration images, containing laser spectra with different orders in different positions of the image area. Ideally they cover most of the image area, in order to avoid extrapolation of the fitting functions. Details are described in the old manual
https://meteorspectroscopy.files.wordpress.com/2018/01/processing-meteor-spectra-v151.pdf
and are not repeated here. The theory is described in the following paper:
A practical method for the analysis of meteor spectra
Martin Dubs, Peter Schlatter (WGN Journal, Ausgabe 43-4, 2015):
Meteor_Spectroscopy_WGN43-4_2015
For simplicity, the spectra are in a single image, obtained earlier with ADD_MAX2 in IRIS (get peak values from several images). In M_CALIB, these images can be created directly from video files.
Calibration with a laser is particularly useful for wide angle lenses and/or low resolution spectra and is easier to interpret:

.\calib\caladd9.png
On each horizontal row 1 spectrum is displayed, with 4 laser lines in different orders, for 8 spectra in total. This was recorded with
WATEC 902 H2 ultimate image size: 720x576 pixels
(Tamron 12VG412ASIR ½", f: 4-12mm f/1.2, used at f: 8 mm)
Grating: 50x50 600 lines/mm Thorlabs
Calibration laser: 405 nm wavelength from eBay:
http://www.ebay.com/itm/Diode-Laser-405nm-20mW-Violet-Purple-Laser-Dot-Module-w-cable-13x42mm-3-5-5V-/191174143732?ssPageName=ADME:L:OC:CH:3160

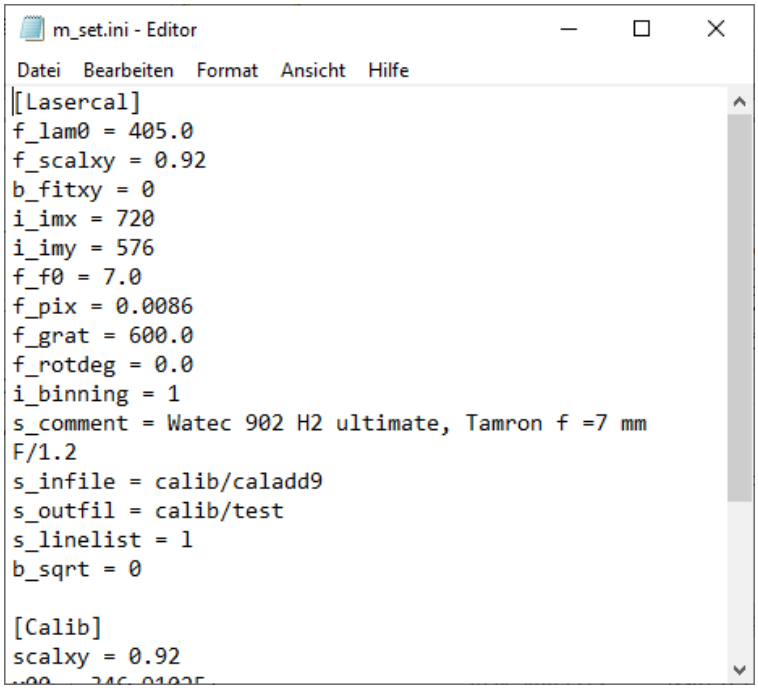
Run the test example

For simplicity we start with the laser calibration as shown in the above figure with the image caladd9.fit or caladd9.png. The creation of this image will be shown later. This avoids uploading large video files. The files are in a folder calib.
Download the test example from the demo zip file or obtain a copy of the files from the author.
In m_spec, change to the laser calibration tab:



Setup

The exact look may vary, depending on the content of the configuration file. At startup, it uses the configuration m_set.ini. Its content is updated at the end of each session with the button Exit, so you can close a session and resume the work later with the same configuration.
The configuration file is a text file like this:



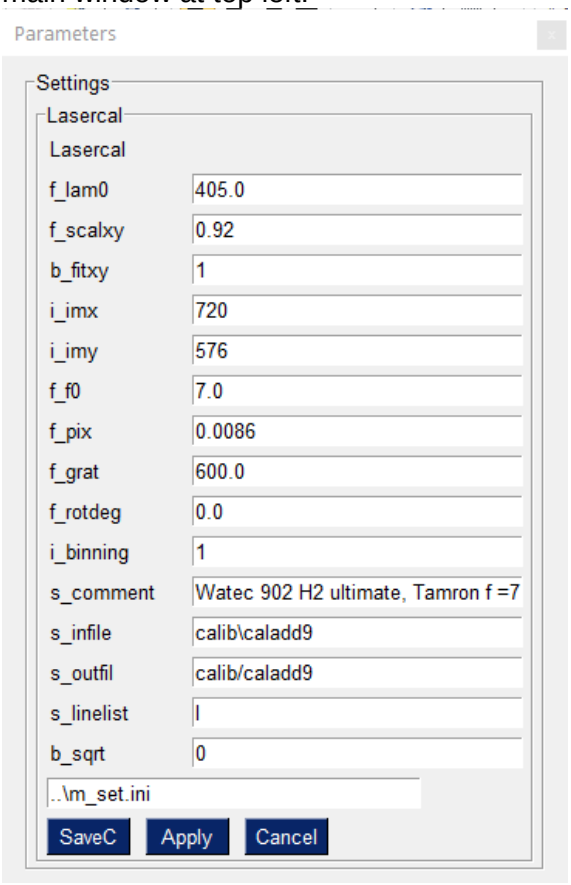
The meaning of the parameters is as follows:

- **F_lam0**: laser wavelength [nm] in case of a calibration with a laser, ignored for calibration lamp. The f_ at the beginning indicates a floating point value
- **F_scalxy** gives the ratio pixel height to pixel width of the camera (or the video capture device, 0.92 is the value determined from UFO analyzer.
- **B_fitxy** is a boolean value, 0 for False, do not use scalxy as a fit parameter, keep it fixed, 1 for True, obtain a better fit with scalxy as fit parameter. If scalxy deviates far from the expected value, set b_fitxy to 0
- The following parameters (imx, imy, f0, pix and grat) are image size, focal length [mm], pixel size [mm] and number of lines /mm of the grating [mm^{-1}]. Its approximate values are used for the start of the fit, but the least square fit works with a wide range of values, takes a bit longer maybe.
- F_rotdeg is the tilt of the spectrum (deviation angle from x-axis, measured in degrees [°].
- **F_binning** indicates if the calibraation image was binned. In that case the distortion parameters are converted to an unbinned image. The spectrum script works with the correct value if it given the correct value of binning (binning is useful for 4k videos, it speeds up the processing considerably, without loosing much precision).
- S_comment gives a short description of the equipment, in case you use different, each with its own configuration file.
- **S_infile** gives the path and filename of the calibration image (without extension). At present images with extension PNG and FIT can be read. PNG is used because it is compatible with the user interface based on PySimpleGUI and FIT is used for internal calculations and as a standard astronomical image format. It contains detailed information about the processing in its header, important for the documentation.
- **S_outfil** (s_for string) gives the path and name of the output of the line positions, as determined by the next step Select Lines (extension .txt added by the script).
- **S_linelist** (.txt) is used for the calibration of lamp or meteor spectra. It contains the (wavelength*order) of the selected lines and a comment for each line. A value ‘l’ (lower case ‘L’) for the linelist means a laser calibration with lam0.
- **B_sqrt** is a boolean flag: 0 for polynomial fit, 1 for SQRT fit. The latter has one parameter less to fit, is more stable and useful for low distortion lenses and/or longer focal length.
- Under the heading [Calib] follow the results of the calibration, to be discussed later.

All these parameters may be edited with a text editor or easier within the GUI

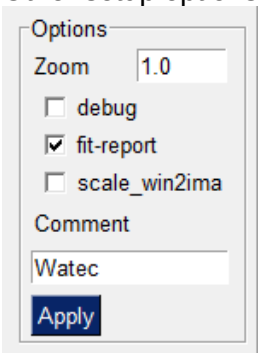
Select configuration

The actual configuration used by the script can be checked with the button ‘Edit Setup’ in the main window at top left.



Important in the setup is the field ‘s_outfil’, which gives the path and filename to the textfile, where the results of the line positions are stored. When starting a new calibration, it might be advisable to give it the same name or a similar name as the image file ‘infile’ or a name indicating specifics like date or camera setup. With ‘Apply’ the configuration is used. A different name for the configuration file can be saved with SaveC. With Cancel the configuration remains as before Edit Setup was chosen, changes are ignored. The files s_infile and s_outfil can also be changed in the laser calibration tab with “Load Image” and “Select File”. Best to try it out, easier than explaining!

Other setup options may be changed in the main setup tab of m_spec:



Debug is mostly for testing purpose. Fit-report gives detailed information about the least square fit in the shell window. You probably do not need it either.If you check scale_win2ima the image is scaled by the factor zoom and the window size changed that the image fits inside. Use a zoom value according to original image size and screen size. It works only after closing the program and restarting it (peculiar to PySimpleGUI). If this is not checked, the image fits the

available space in the window, the zoom factor is ignored (For the determination of line position, the original image is used, with the binning as given by i_binning).

Select points of the different orders

Once the configuration has been set, we can load a calibration image for analysing with “Load Image”.
If you would like to do another analysis, you can type the new filename in the input field Calibration data or select an existing file with “Select File”.
Start the calculation of the line positions with ‘Select Lines’. A new window appears:

- Start with the lowest order at the left and click with the mouse on the spot you would like to select:

  The program calculates a Gaussian fit of the spectrum and saves the result. A yellow circle shows the selected point. A label shows wavelength*order. Continue with the next point. Since all lines have the same spacing, it is not important if the first point is actually the zero order. This is different with a calibration lamp, where the lines have to be selected precisely as given in the list.
- If you have selected all orders of one spectrum, press “OK”, the coordinates and widths of the different orders will be saved to the text file.

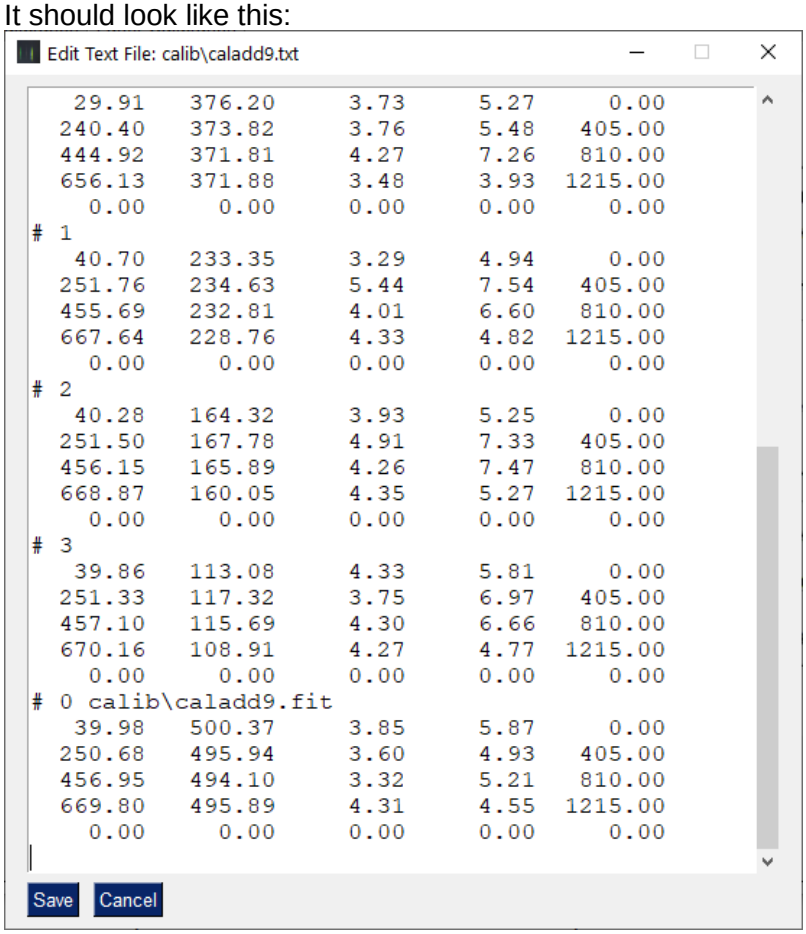
- If you made a mistake, press ‘Cancel’, to start with the same or another spectrum
- With ‘i(ncrease)’ and ‘d(ecrease)’ you can adjust the image contrast
- With ‘Skip Line’ you can skip a line (not needed here,but useful if you do not see a line corresponding to the linelist, or when it overlaps another line)
- Press “Finish” after the last spectrum
  In the Results window it should say ‘Finished, saved caladd/caladd9.txt’.

Note:
If you have problems with the selection of points, you may change the size of the rectangle, in which the Gaussian peak is searched and fitted (default is a rectangle with corners x0, y0 +/- 10 pixels. You may draw a rectangle in the image by selecting one corner and draw with the mouse down to the opposite corner and release the button there:

If the rectangle is too small, you may miss the point when selecting it, if the rectangle is too large, the Gaussian fit may fail. You will get some error messages in the shell, try with another size! Drawing the rectangle does not select the point. Type ‘r’, then start a new spectrum. Notice, this is still a test version.

Inspection of the calibration text file

With “Edit File” you can check the measurements.
It should look like this:



- The columns denote x-coordinate, y-coordinate, FWHM(x) and FWHM(y) of the spots, as determined by the Gaussian fit
  The last column is wavelength*order (added for analysis of calibration spectra)
- The lines with # are comments and ignored for the fit.

- Each spectrum ends with a line of zeros as separator
- Additional lines of zeros in between are ignored, also spectra containing only one line (there is nothing to fit there)
- If you forgot one or more spectra, you can run the script again and add the missing spectra
- If you recorded one spectrum twice or entered wrong data points, you may erase these in the text file manually and save the file again for the following step.
  This can be done directly in the Edit File window, followed by 'Save'

Least square fit

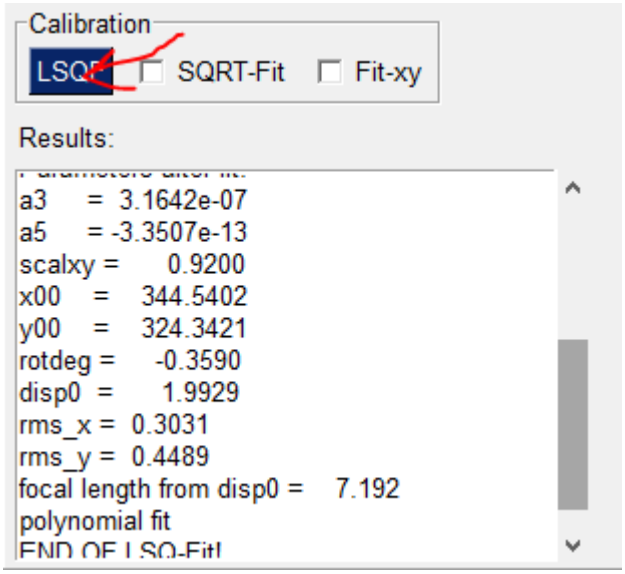
For the determination of the distortion parameters, a linear dispersion law is postulated in the transformed image
(x' = x0 + i*lam0*disp0, i = 0, 1, 2, … for laser calibration, used in old version
In present version:
x' = x0 + lam[i]*disp0, i = 0, 1, 2, … , where lam[i] = wavelength*order of line[i] from line list, both for laser calibration and spectral line calibration
y' = y0
from these coordinates the transformation to the image coordinates (as required in the coordinate transformation of the meteor images) is calculated by a transformation to polar coordinates around the image axis at coordinates (x00, y00), rotation of the image by an angle rotdeg (converted to rot, angle in radians, rot = rotdeg*Pi/180)), transforming the radial coordinate by
$r = r' * (1 + a3*r'^2 + a5*r'^4)$ (plus higher terms in the future),
transforming back to Cartesian coordinates with the axis remaining at (x00, y00)
For those interested in the details, here is the corresponding code from the Python function.
(x0, y0s) correspond to (x', y').
xyr +xy00 corresponds to (x, y) in the calibration image, to be fitted to the coordinates in the calibration file (testcal.txt in the example)

```
    xy00 = [x00,y00]
    xyl = [x0s+lam/disp0,y0s] - np.array(xy00) # coordinates of laser lines wrt, optical axis
    r = np.sqrt(xyl[0]**2+xyl[1]**2) # polar coordinates
    phi =np.arctan2(xyl[1],xyl[0])   # polar coordinates
    phi += rot                       # apply rotation
    r = r*(1 + a3*r**2 + a5*r**4)    # transform radial coordinate
    xyr = np.multiply(r,[np.cos(phi), np.sin(phi)/scalxy]) # return to cartesian coordinates
    return (xyr + xy00)
#-------------------------------------------------------------------------------
```

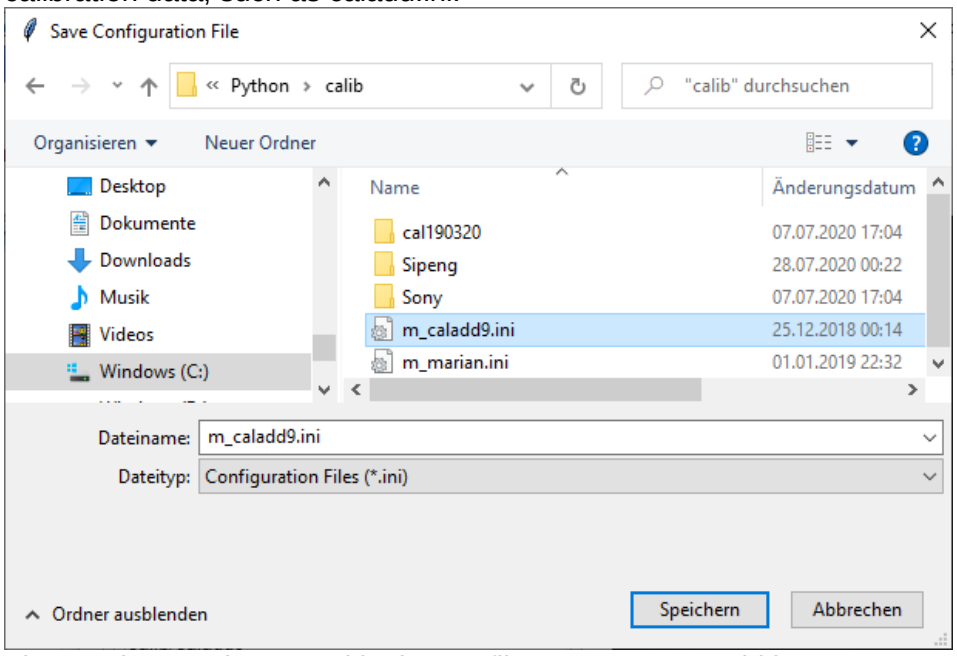
NB: For the sqrt-fit the following equation is used for r':
$r = r' * (1 + a3 * r'^2 + 1.5*a3^2 * r'^4)$

Run the fit for the example caladd9.txt with m_calib.py

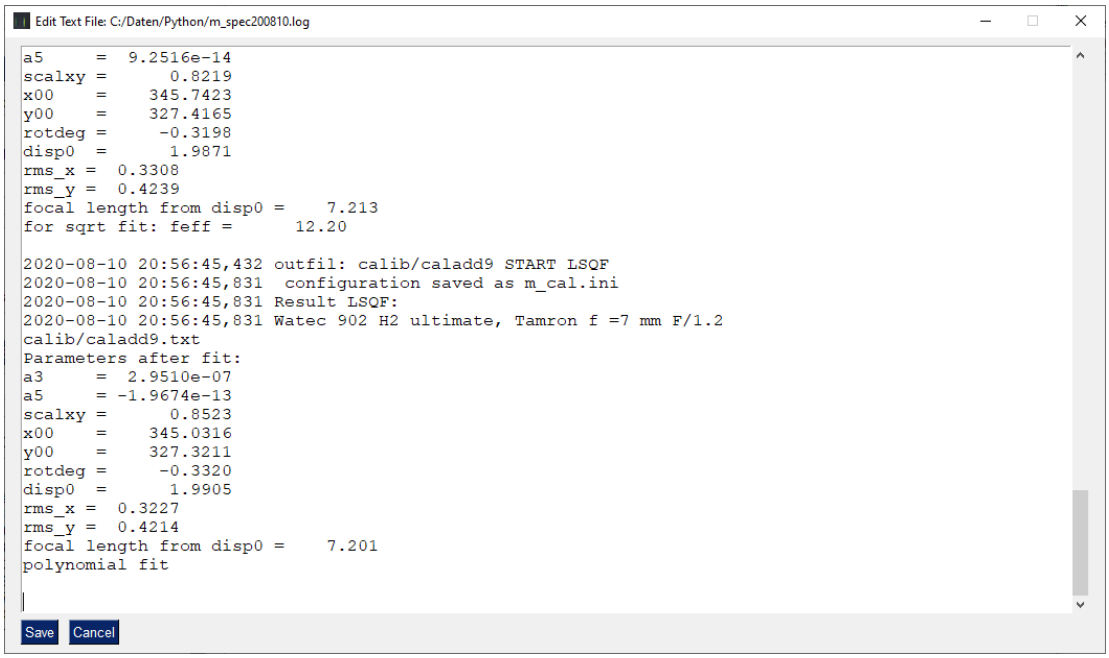
Let us assume that you have edited the file caladd9.txt and corrected all the errors (wrong points, wrong order, etc.) and you would like to start the least square fit with these data.
Select 'LSQF'.
After the fit the results of the least square fit appear in the log window:



At the same time they are stored in m_cal.ini in the working directory (in order not to overwrite m_set.ini, which may be used as a backup). If you like the result and want to use it with m_spec, you may save the new configuration under m_set .ini or another meaningful name (e.g. m_set_yymmdd.ini or m_set_sony24mm.ini etc.)
These parameters you may use to transform your meteor spectra to the orthographic projection. Notice that the fit runs quite fast, the Levenberg Marquard algorithm is very efficient for this kind of problem. In addition the results of the fit are shown in graphical form (helpful, if you have made any errors in the assignment of any lines).
I recommend to save the configuration under a suitable name, perhaps in the directory with the calibration data, such as caladd.ini:



The results are also stored in the Logfile m_spec_yymmdd.log
You can check the logfile with Menu – Tools – Edit log file:

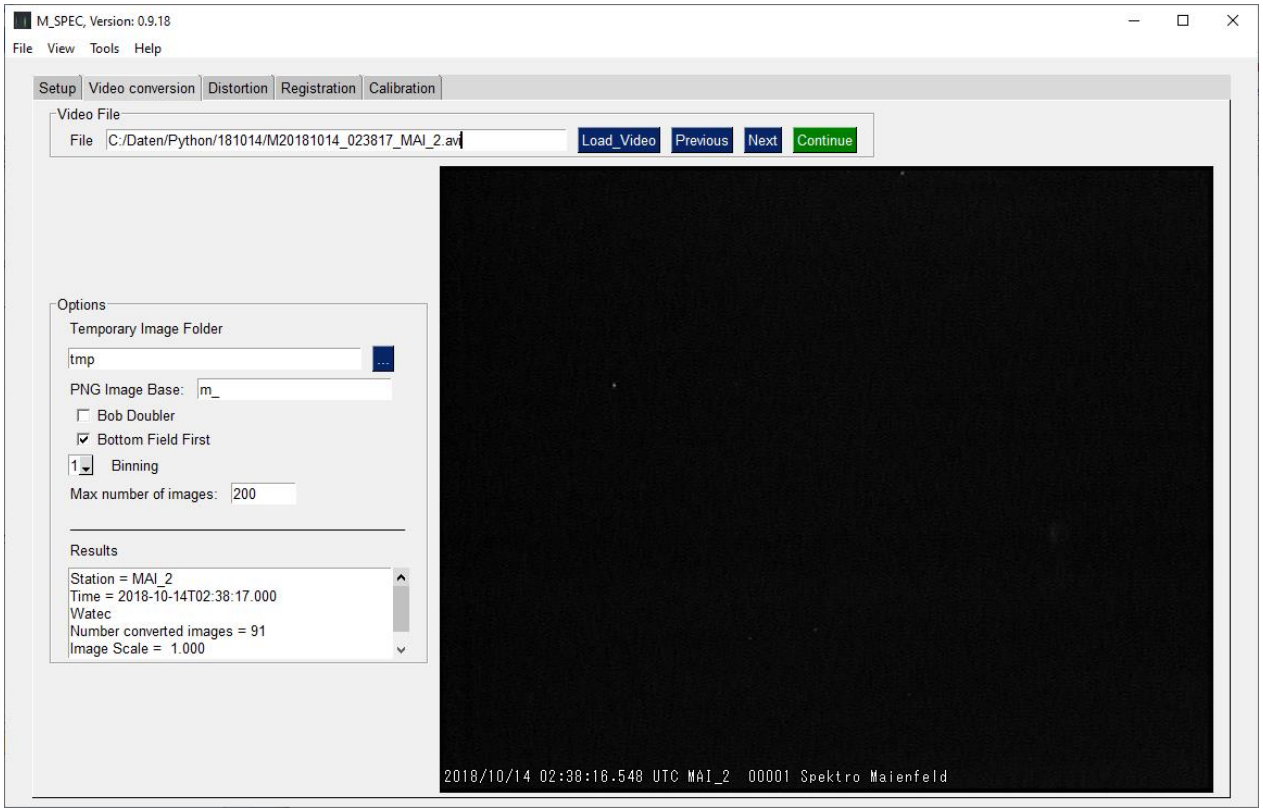


If the results look ok and the rms-errors are reasonably small, you are ready to process actual meteor spectra.

Processing of meteor spectra, step by step

Conversion of *.AVI to PNG

With the correct configuration we can start the meteor spectra processing, first converting the video file into a sequence of *.PNG images. This is done in the next tab, “Video conversion”:



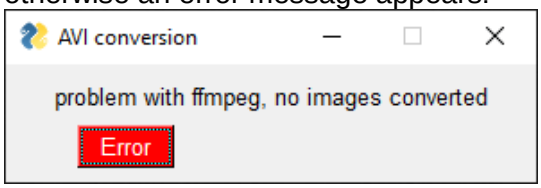
Before you load the video file you have to set the options!

- You can select a folder to store the images. Since they are used only once they are stored in a temporary folder tmp.
- You can also choose the filename of the images, the index starting from 1 is added: m_1.png, m_2.png, …
  This is the same naming convention as in IRIS, so you may use these files also in IRIS, for processing steps not included in this pipeline. You can change these names, but this is not really useful as these are only intermediate files you need once or a few times if some errrors occurs later in the processing.
  The images are erased before a new video conversion, so do not select a folder with images you want to keep.
- With Bob Doubler you can read half frames for interlaced videos, in this case you also have to choose which field is read first. For Watec cameras this is usually Bottom Field First. With the wrong selection, the meteor jumps around and it is difficult to register the images correctly. This increases time resolution and therefore spectral resolution if the meteor has a velocity component parallel to the dispersion direction (recommended for WATEC).
  Important: the calibration parameters are adjusted automatically; in particular the factor scalxy is multiplied by 2 to correct for the different aspect ratio.
- 2x2 binning is useful to reduce the file size for 4k images (for 4k color cameras the color pixels are often interpolated, so you loose little information by reducing the size to 2k).
  Important: The distortion parameters are not adjusted automatically, it is assumed that

you use the same binning as for the calibration (keep 2 sets of calibration parameters for 2k and 4k images!)

- It is also possible to limit the length of the video sequence. This is helpful in case of a corrupted file or when you want to save time or disk space.

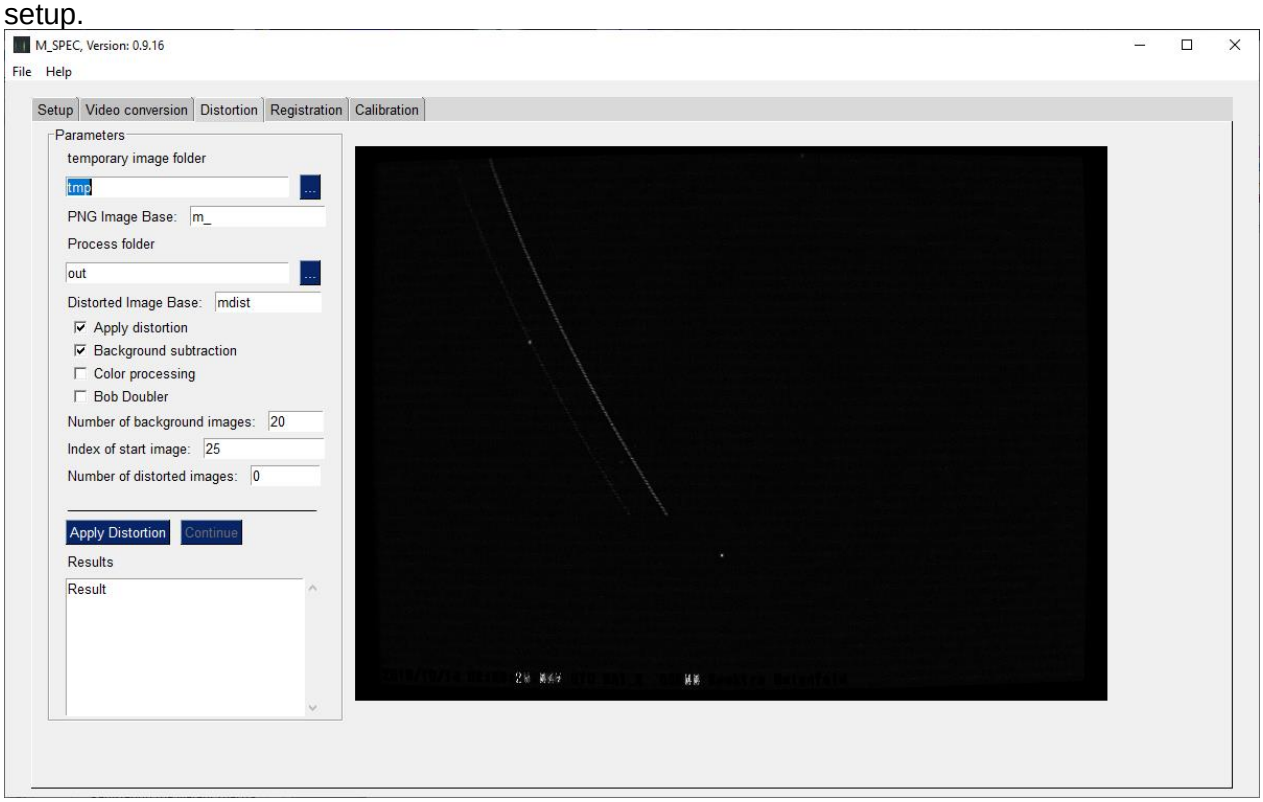
With the options set, you can load an AVI-file. The conversion starts automatically and the button Continue turns green when completed. For this step ffmpeg.exe has to be installed, otherwise an error message appears:



Notice that the path of the selected file is written with forward slash: /, so users of LINUX should not have a problem with it. In most cases also a backslash \ is accepted. I personally use a different directory for each meteor, with the image files having standard names.
In the memo at left some details of the conversion are given. With Continue you go to the next tab, “Distortion”. Before you do that you may look at the images with the very basic image browser. With the Next button you jump from frame 1 to 25 and then advance in steps of 1. With Previous you go backward in the sequence. This is because the first 25 images only contain background and the meteor appears around frame 25. With Bob Doubler selected, the Next button selects field 50 after the first field, corresponding to frame 25.

Distortion

In the distortion tab you can apply the transformation to an orthographic projection, which linearizes the spectra. For this to work, you have to set the correct distortion parameters in the setup.

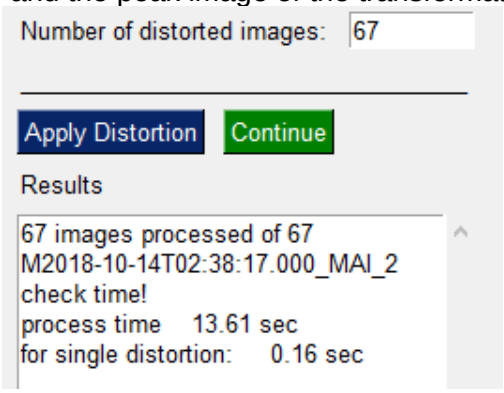


In addition you have to set some additional parameters in the frame at left.

- Temporary image folder, PNG image base: if you have just done the video conversion, these values are preset, otherwise you can choose them to convert other images. Note that they should conform to the naming convention, file base index .png, e.g. m_1.png, m_2.png, …

- Process folder and distorted image base: Folder and name for the processed images. Naming convention as above. Notice that files with the same name are removed from this folder before a new distortion is performed. This prevents a mix of different spectra. For the process folder I use a separate folder for each date, where I also store the AVI file(s), setup-file, processed spectra and logfile(s) for later reference.
- Apply distortion, background subtraction: These are normally checked, but you can do only one operation if desired.
- Color processing is mostly used for making nice images, the spectra are evaluated from b/w images. Use b/w, it is three times faster than colour processing and for the extraction of the spectra you have to convert the images anyway.
- Bob Doubler: This is set the same value as in Video Conversion. Be careful to set the right value when you do a distortion correction at a later time. It is advisable to do the distortion right after Video Conversion.
- Number of images: For background and start images use the default values. The number of distorted images is calculated from the number of available images after video conversion. Select a reasonable number. It will be adjusted to a lower value, if not enough images are found.
- Background images:
  As described in the general processing section, a background image is created from the first second of video. The default value of the number of images used for the averaging is given for a pretrigger of 1 sec and a frame rate of 25 images/sec. With bob doubling, the image rate (fields/second) is doubled, therefore also the double number of images can be used for the background. Make sure that the default value does not use any images with the meteor appearing early.

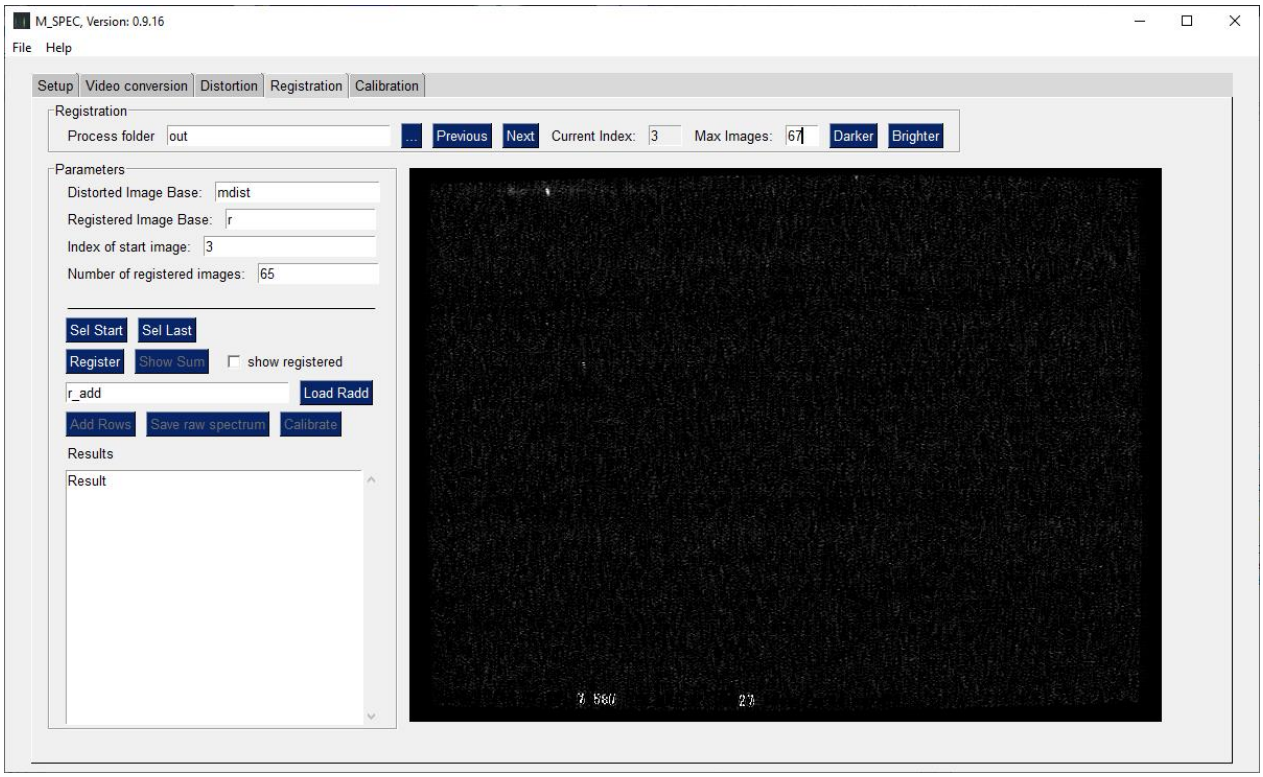
Once everything seems ok you press the button “Apply Distortion” and wait for the end of the processing. This is indicated by the button Continue turning green (means enabled) and some information is output in the result memo at left bottom:
and the peak image of the transformation is displayed:



The process time is the time between the start of the background image and the end of the transformation. The log shows also the average process time per image.
The contrast and brightness of the peak image has not been adjusted, so you should not be disappointed by this result. If you continue you can adjust the brightness of the transformed images in the registration tab.

Registering the meteor spectra

As explained above, the meteor spectra of the different images can be added after registration, because they are all aligned parallel to a common dispersion direction and they all have the same linear dispersion. This is because of the transformation in the previous step. The meteor movement causes a displacement of the spectra. This is removed by registration, which will be explained in detail.

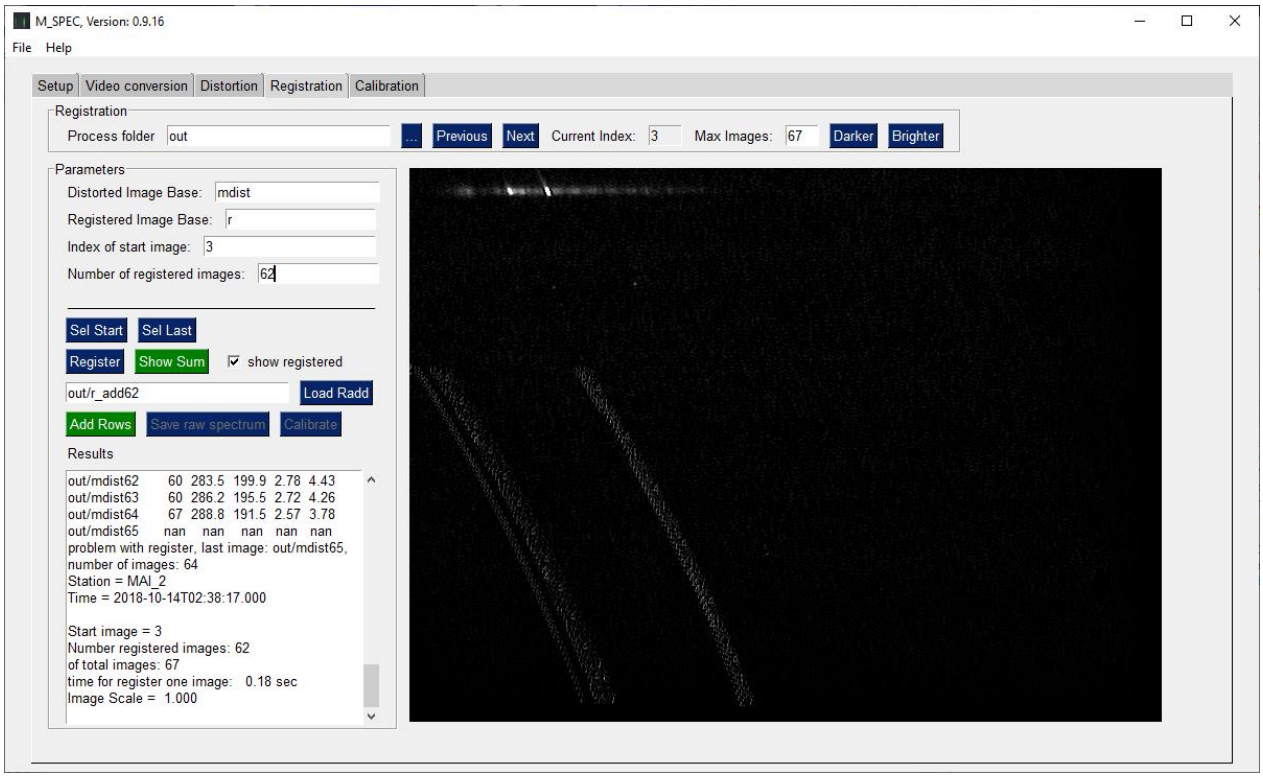


The idea is to identify a clearly visible line in the meteor spectrum and shift the following images so that this line is in the same position in all images. The position is determined with a Gaussian line fit, similar to the determination of line position in the calibration script. The problem is that this determination of line centre is not fool-proof, it may be disturbed by a missing line in one or several images, by saturation or overlap with something else. If the registration does not work the first time it is possible to repeat it with different parameters. Important is the selection of the first image used for registration. If the 1st image is not useable a different one may be chosen. The registration frame at the top is a simple image browser, which helps to select the correct images.

- The process folder and file base are preset from the distortion tab. If you want to register some other files you can choose the folder with the button "…" and change the base file name of the images you would to register (here mdist, for the files mdist1.fit, mdist2.fit, …, with the now familiar file naming convention)
- You can also change the name of the registered file base, r for r1.fit, r2.fit, …
- With the buttons "Next" and "Previous" you can click through the images to find a suitable start image, here mdist3 was selected
- With the buttons "Brighter, "Darker" the image contrast can be changed
- The current index is shown and the number of images can be set. If this is larger than the actual number of images it is adjusted automatically.
  Notice that the numbering of the meteor spectra mdist is offset from the image number in the original video file by: first meteor image – 1

If you have found the correct start image, press the button "Sel Start" or enter the image number (here 3) into the text box "Index of start image". Browse through the images with next until your last image to register and press "Select last" or enter the corresponding number of images (not the index) into the box "Number of registered images".
With all settings done you can start the registering procedure by pressing "Register".
A new window appears with the start image. Here you can draw with the mouse a rectangle enclosing the zero order or whatever spectral line you would like to use for the registering:

If you do not see the zero order either the image is too dark or you selected the wrong image. With Cancel you can try different parameters. With Ok, the registering starts. In case of success you see the sum spectrum of the registered images:

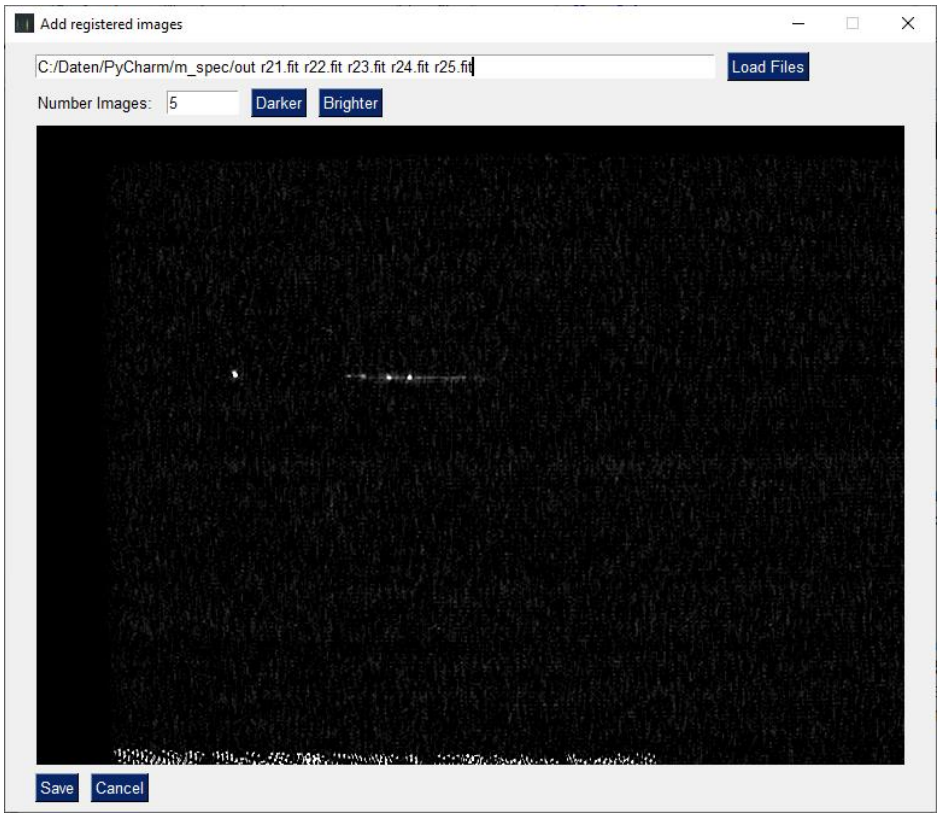


With Previous and Next you can look through the registered images, Show Sum shows the sum image again and by unchecking “show registered” you can browse through the original images. In the Results section at left you can check if the registering was successful. A list of the selected files, line intensities, positions and line widths of the Gaussian fit is given. Apparently there was a small problem with one of the last images. Try again with a different (smaller) rectangle size or leave it as it is. The size of the rectangle should be at least as large as the size of the line and the movement from frame to frame (whichever is larger), but not too large. This need a bit of practice, the algorithm to determine the line centre (x, y) could probably be improved. Notice that in this particular case not the zero order was chosen but the Na-line, often one of the strongest lines in the spectrum. The zero order was outside of the image area to the left.

New in version 0.9.18

If you have registered a series of images, but only want to add some of them (e.g. to use only unsaturated images or to create a time series r1-r5, r6-r10, r11-r15, …) you can select in the menu – Tools – Add images

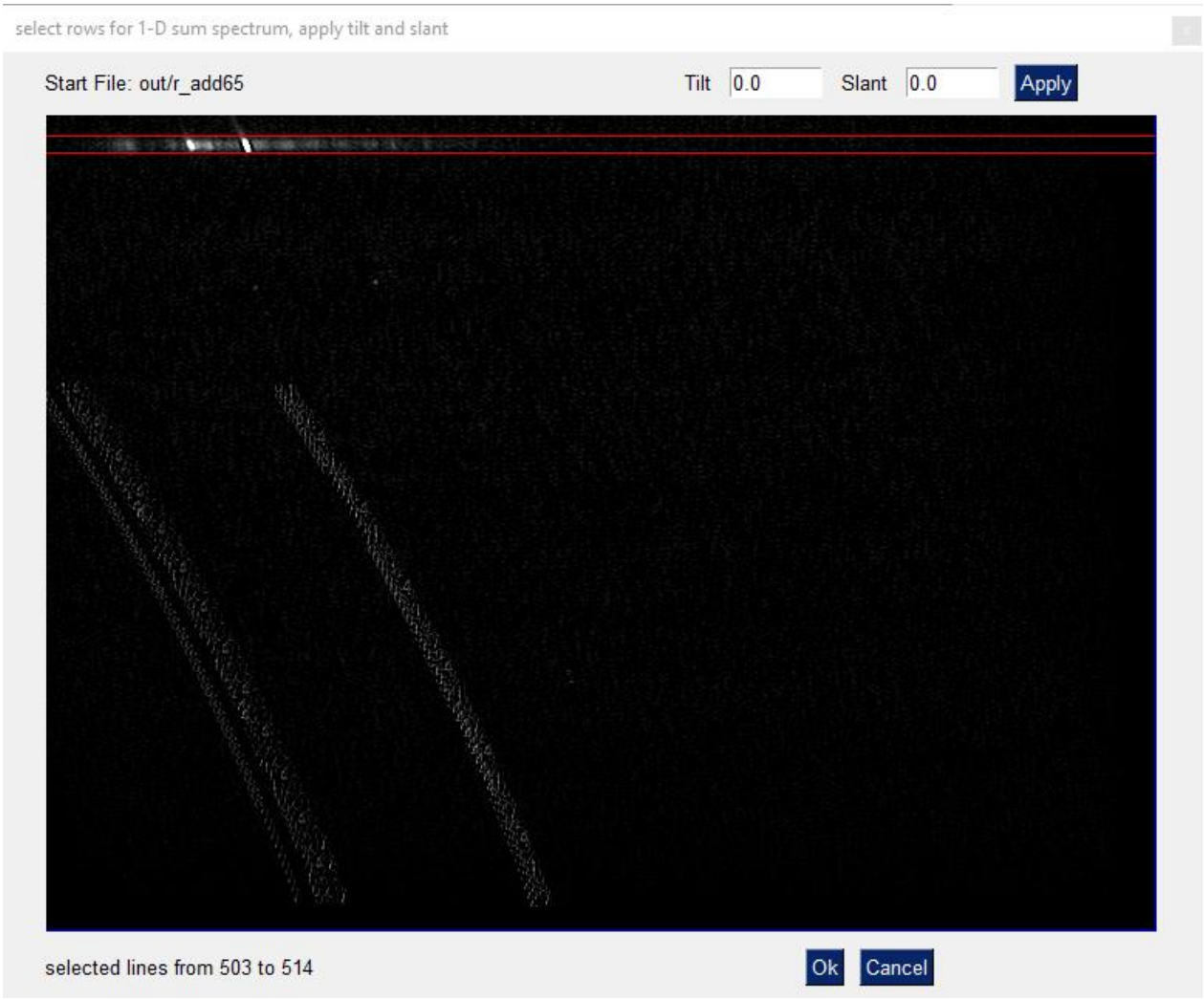
A new window opens, where you can choose the desired files to add with ‘Load Files’:



You can save the sum image (actually it is the average) under a suitable name (e.g. r21-25.fit)

Spectrum extraction

Now you are almost there. Adjust the contrast of the coadded spectra until you see the spectrum strongly but only the strongest lines saturated. Continue by pressing “Add Rows” or load a previously registered image by pressing “Load Radd”. A new window with the sum spectrum appears. Next you can select the rows you would like to add, by dragging the mouse over the width of the spectrum:

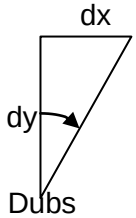


For the width of the selection it is important to add all the rows which contain the spectrum, but not the rows which contain only noise or other signals (The stripes in the lower half were produced by the text in the images, avoid these). If you do not like your choice, try again. In this case mdist3 to mdist67 were registered as r1 to r65 (notice the IRIS compatible naming) and added to r_add65.fit. In this particular spectrum you may notice that the spectral lines are not vertical because of the movemont of the meteor. You can improve this by processing fields with half exposure time (bobdoubler) or by correcting the slant (making the lines vertical). More about this later.

A look at the FITS-header (e.g. with IRIS: image info or Audela:) shows that the information about the time and observation as well as the distortion parameters are stored for documentation. Starting from version 0.9.18 this is also possible from within m_spec (see additional features)

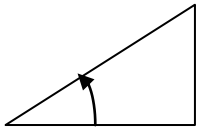
With Ok you could select the rows and the resulting 1-d spectrum is stored as r_add65.dat, with the summed intensities as a function of column number (starting at column 0 to 719) for further processing with any spectra software (ISIS, Vspec, EXCEL …). Before doing that you should read the next paragraph.

You can adjust tilt and slant by typing a suitable value into the text boxes for Tilt and Slant. For a clockwise change of slant you enter a positive value. The value is the tangens of the angle or the slope dx/dy applied to the image



You confirm the value with Apply. The modified image is displayed. If you do not like the result you may type r and start all over or you type a new value for the slant and ↵ until you are satisfied with the result:

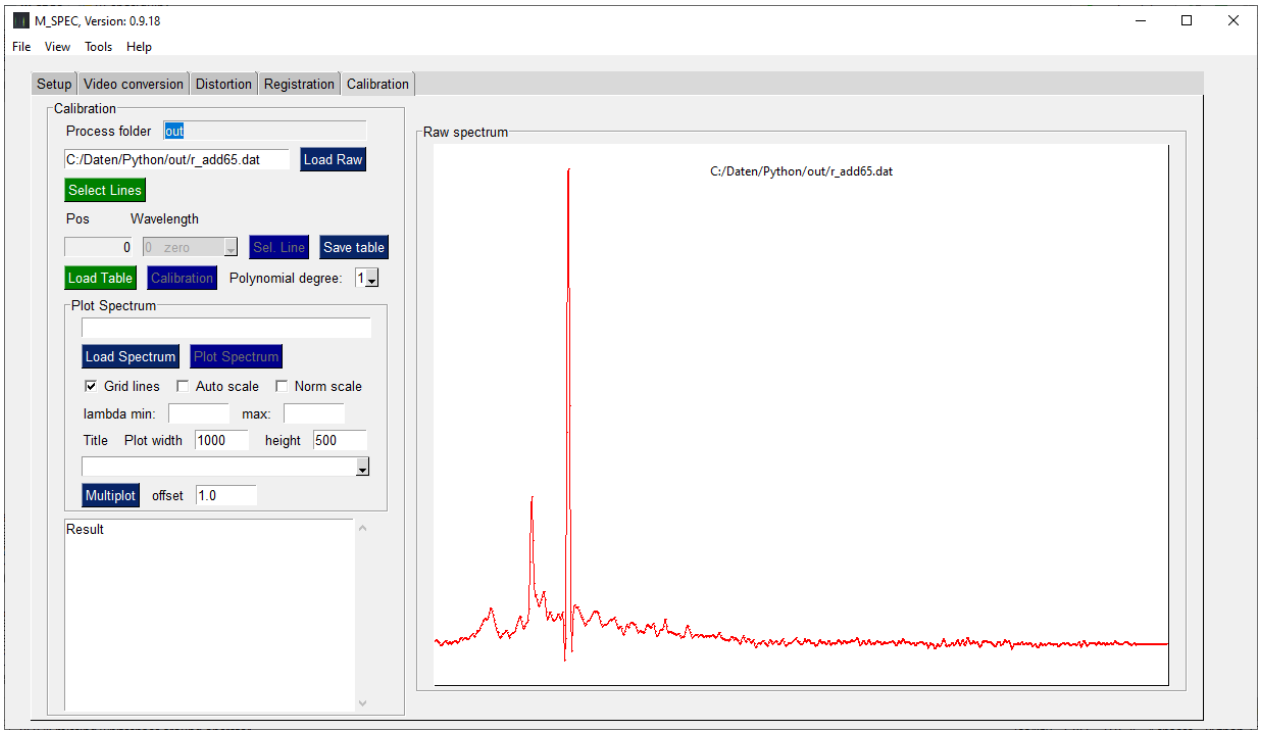
For the tilt the function is very similar. It is defined as the the tangens of the angle of the direction of the dispersion with respect to the x-axis or the slope dy/dx:



A positive value of the tilt rotates the spectrum anti-clockwise, as shown for demonstration in the next image:

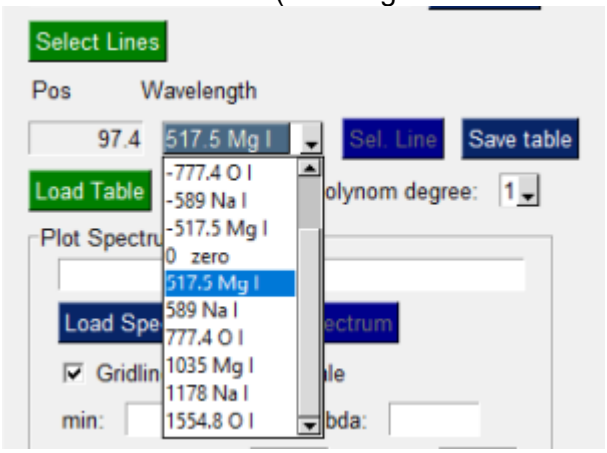
Of course, in this case a tilt of zero is the correct value, since the spectrum was aligned initially.. Notice that you can restore the original image by entering 0 for tilt and slant. When everything is adjusted, you type Ok to save the modified image and the 1-dimensional spectrum obtained by adding the rows column by column. The corrected image is stored as r_add65**st**.fit for processing with other software and the 1-dimensional spectrum obtained by adding the selected rows is stored as r_add65.dat (uncalibrated spectrum). You may also save it under a different name with “Save Raw Spectrum”
The slant is needed for correction of the meteor moving diagonally across the image field. Making the spectral lines vertical improves the spectral resolution. The tilt is used to correct for misalignment of the disperssion direction. This can also be corrected by adjusting the angle in the setup parameters. If the grating rotates slightly in front of the camera, a correction of tilt during the analysis is easier here, where the result is seen immediately. If a consistent tilt angle is observed in the spectra, it is advisable to correct the angle “rot” in m_set.ini. For small angles the tangens of the tilt is equal to the required correction of “rot”, measured in radians.
If you are satisfied with the result, you finish this process step with “Ok”. This saves the corrected image as r_add65st.fit. The image is normalized to a maximum of 32767 (2^15-1) in order to be viewed easily in IRIS. In the fits-header of this image the selected values of slant and tilt as well as the selected rows and the selected width (half width) are displayed:
The original image r_add65.fit is simply the average of the added images, with 255 ADU’s scaled to 32767, for readable display in any FITS-viewer.

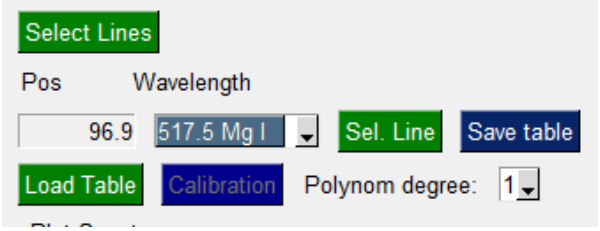
By addition of the selected rows for each column the 1-dimensional spectrum is calculated and displayed when you press “Calibrate”. This brings you to the last tab for wavelength calibration of the raw (Intensity vs. pixel) spectrum:



Wavelength calibration

In the graph the raw spectrum is shown. Start the calibration with “Select Lines”. This activates the mouse clicks in the graph and starts a new table with wavelengths and positions of calibration lines. Select a line in the raw spectrum by dragging the mouse with pressed left button across it. 5 points around the maximum intensity are used for a quadratic fit. The position of its maximum determines the line position in pixels. In the text window Pos is the center of the selected range given. Make sure the highest point of the line is within the range indicated by green lines. Select the correct wavelength from the table “Wavelength” or enter a wavelength not listed in the table (you have to enter the table with a mouse click). Confirm the choice with the button “Sel Line” (it turns green when a wavelength was chosen

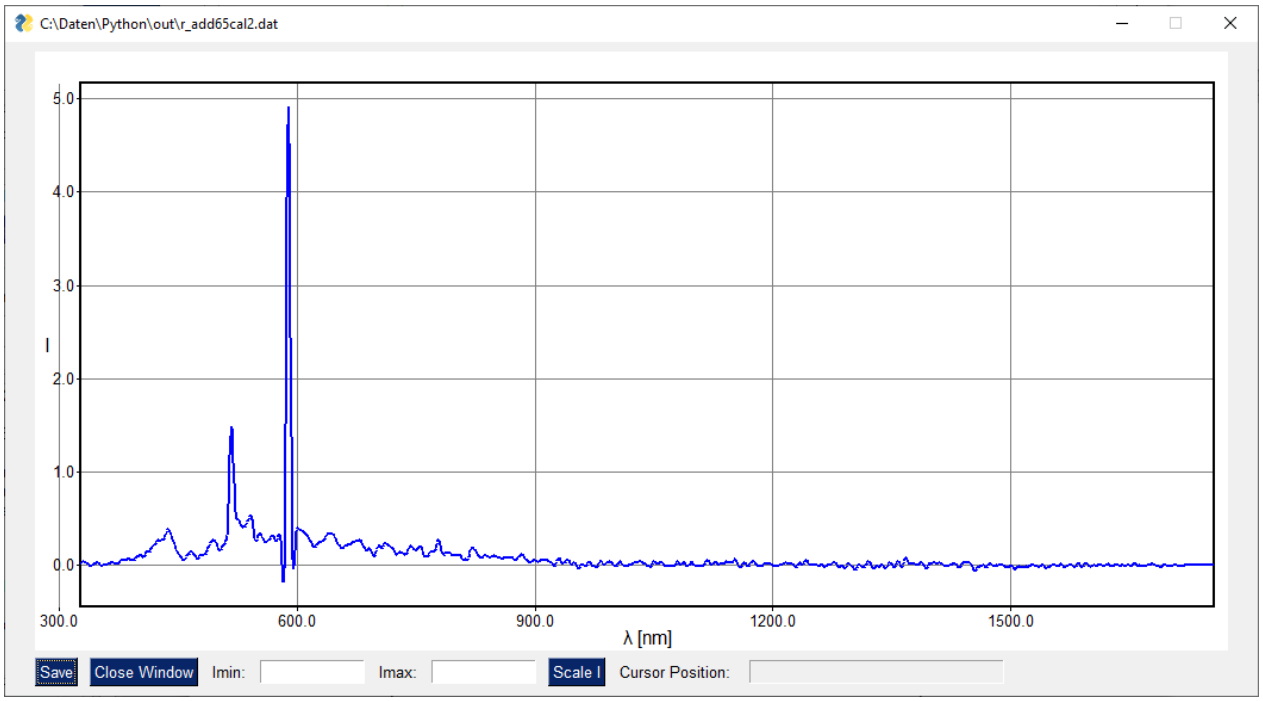




In the memo below the position, width (calculated from the parabolic fit and selected calibration wavelength with information about the line are displayed. Continue by selecting more lines (those already used are shown in blue). When finished, press “Save Table”. The list of positions and wavelengths is stored under the raw file name.TXT

```
#     x     lambda
   95.78   517.50
  131.67   589.00
```

You may edit this table with a text editor or the built-in text editor, in case you have entered wrong lines or wavelengths. Load the raw spectrum again if you would like to make a new calibration. The table is overwritten if you do a new line measurement with the same raw spectrum.
With two lines only a linear fit is possible. Select 1 for polynom degree (default). The polynom is of the form $a_n*x^n + a_{n-1}*x^{n-1} + \ldots + a_1*x + a_0$.
For the linear polynom: $a_1*x + a_0$, with dispersion a_1 and offset a_0.
With “Calibration” (green if you have saved or loaded a table) a least square fit of the polynom to the selected line positions is done.
The error of the fit is zero, because with 2 lines a linear fit is determined, there are no additiional lines for evaluating the error. The calibrated spectrum is saved as r_add65cal.dat, unless you give a different name to the raw spectrum. With “Plot Spectrum” the calibrated spectrum is shown:

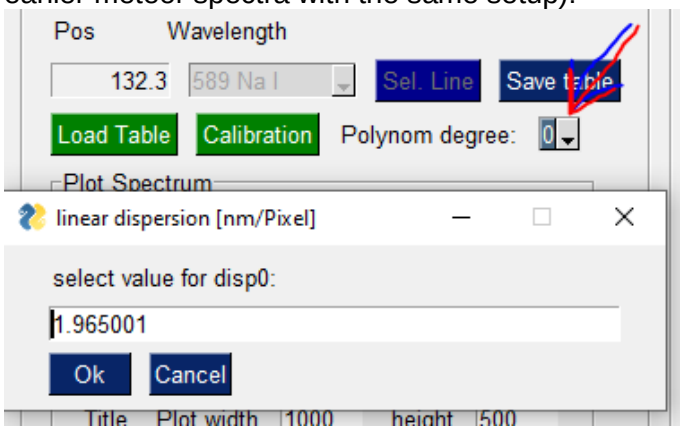


You also have the possibility to select the plot range. The intensity scale you set in the Plot window, with “Scale I” you plot the new intensities. By closing the window and replotting it you also set the correct scales. The wavelength range and the window title are set in the main window. The title can be selected from a list of the latest converted avi-files and the latest raw spectrum or typed into the list manually. The spectrum can be saved with “Save”. Choose a filename, a png image is saved. If no filename is entered, r_add65cal_plot.png or the corresponding raw file name + “_plot.png” is saved. Further possibilities to plot spectra are given below, in additional features.
Important:
The script requires that the wavelength increases to the right, to higher pixel values. Therefore orient the grating in front of your camera that the first order is on the right side with respect to the zero order. The orders to the left of the zero orders have negative values, therefore the corresponding wavelengths are negative. Always enter the line wavelengths as wavelength*order (0 for zero order, wavelength for first order, 2*wavelength for 2nd order, etc.).

Single line calibration

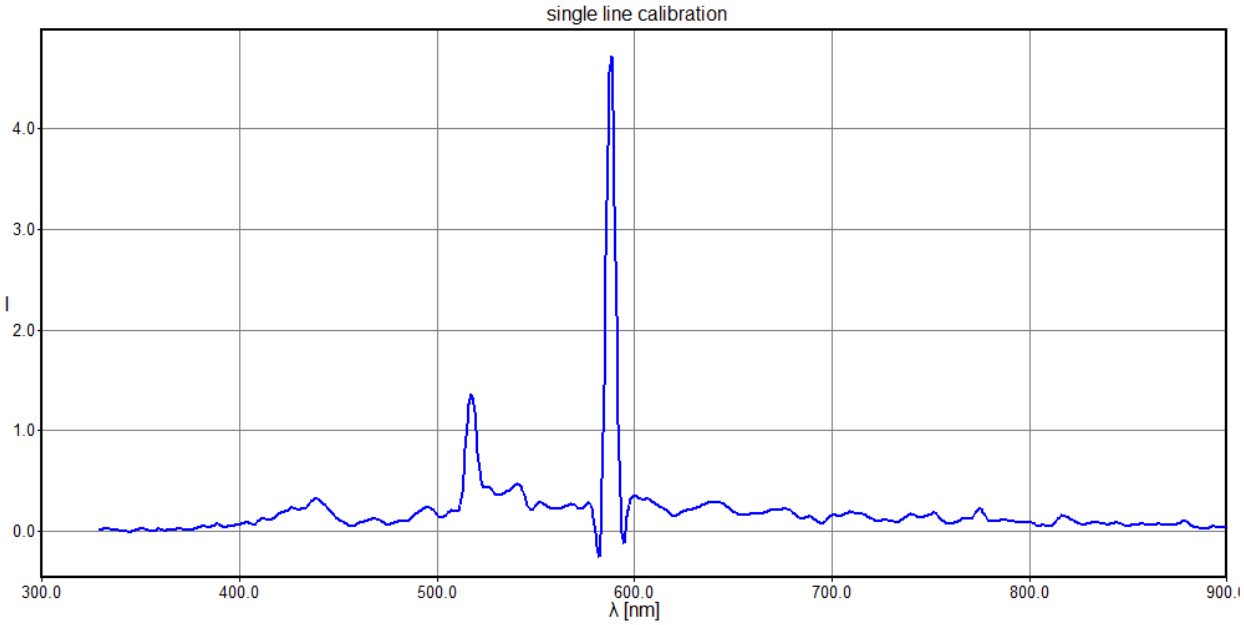
The dispersion calculated from these two lines is similar to the dispersion from the laser calibration: disp0 = 2.005. However, it was derived from two lines in a narrow region, therefore the value is not very precise. In that case it may be better to use the dispersion disp0 for the fit.
This is done by entering 0 for the polynom degree:

The value 2.005 from the setup file is suggested for disp0. In case you know the dispersion from other meteor spectra, you may enter here the correct value (1.965 was selected, based on earlier meteor spectra with the same setup):

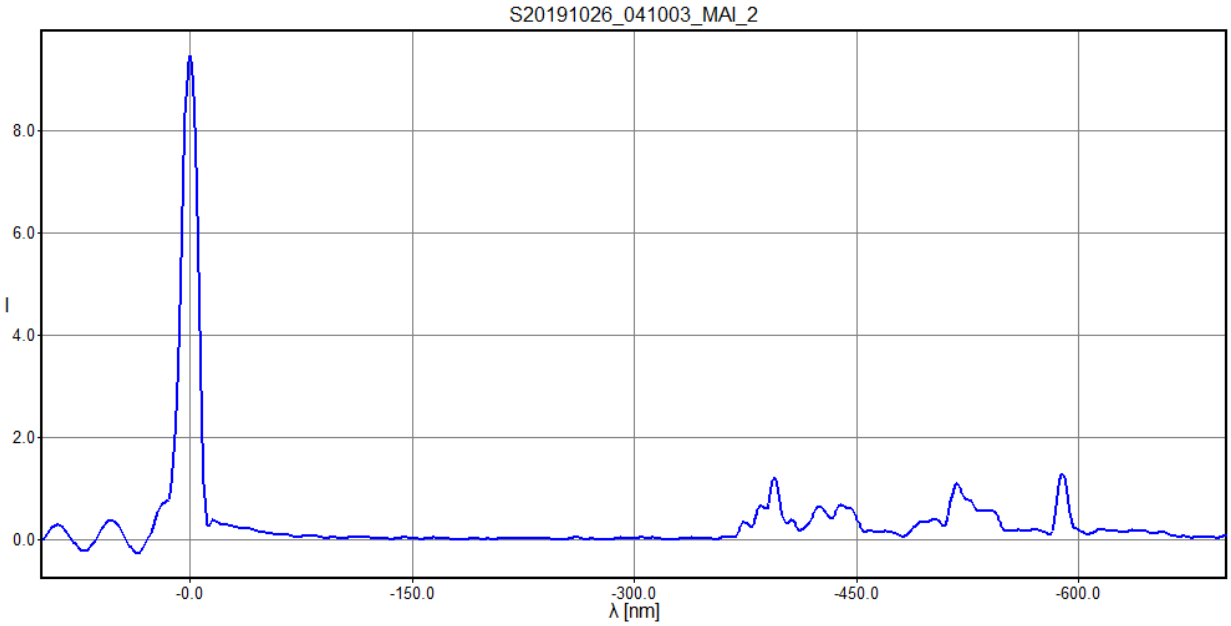



```
File C:/Daten/Python/out/r_add65.dat loaded
   Pixel     lambda        fit      error
   96.05     517.50     517.50       0.00
  131.95     589.00     588.04      -0.96
polynom degree: 1
polynom for fit lambda c: [1.965001,
328.76165395]
rms_x =   0.6763
spectrum C:\Daten\Python\out\r_add65cal.dat
saved
```

We use the same linelist, but only the first entry is used, therefore the following lines will have an error. I use the Mg-line for calibration



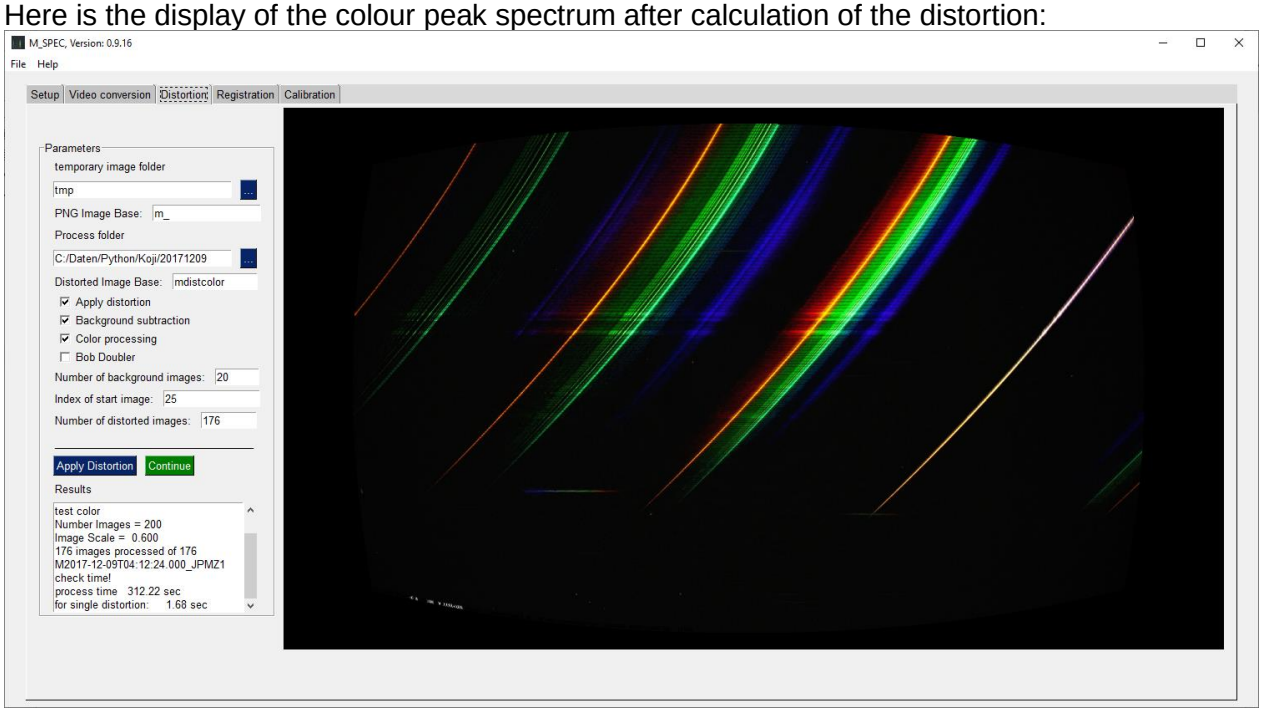
For plotting a negative order you may reverse the plot by plotting from e.g. +100 nm to -900 nm, plotting the zero order and negative first order:


S20191026_041003_MAI_2
8.0
6.0
I
4.0
2.0
0.0
-0.0
-150.0
-300.0
-450.0
-600.0
λ [nm]

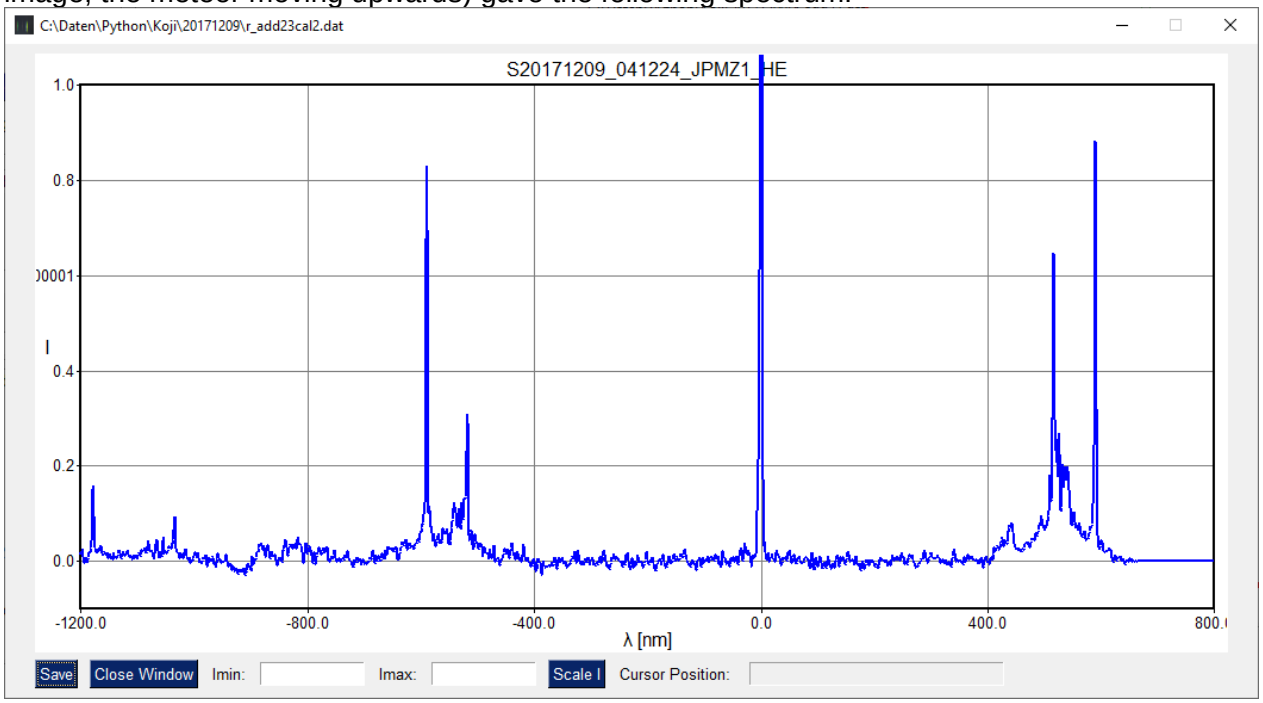
Processing of 4k meteor videos

For processing of meteor spectra recorded with large format video cameras such as Sony Alpha 2 a fast computer is advisable. For faster processing 2x2 binning is recommended. Both colour and b/w processing is possible. Colour processing takes about 3 times longer, for registering a small portion of the images is converted to b/w. For the summation of rows and conversion to a 1-D spectrum the different colour channels are added, so the final result is the same as with b/w processing. As an example a spectrum recorded by Koji Maeda was processed (see also https://meteorspectroscopy.org/2018/01/03/koji/)

Here is the display of the colour peak spectrum after calculation of the distortion:



Notice the time for calculating the distortion (1.68 sec for images of size 1920x1080, 2x2 binning). An analysis of the images 28 to 50 (approximately the first second, at the bottom of the image, the meteor moving upwards) gave the following spectrum:



Typical for an early spectrum are the strong Na I lines (-2nd to +1st order), which were also used for calibration with M_spec and for calibration of the actual spectrum. A linear fit was used:

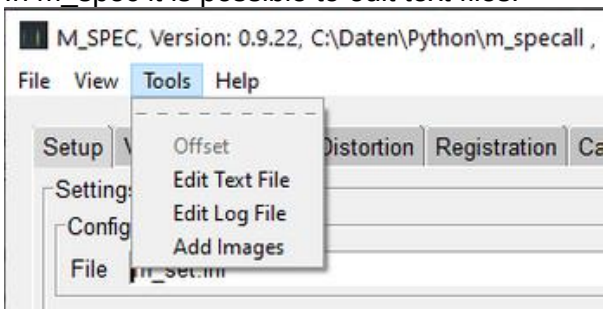
```
polynomial for fit lambda c: [ 1.259e+00 -1.503e+03]
    pixel     lambda      fit        error
    257.95,  -1178.00,  -1177.83,     0.1668
    725.21,   -589.00,   -589.48,    -0.4819
   1193.86,      0.00,      0.62,     0.6196
   1603.37,    517.50,    516.25,    -1.2451
   1661.89,    589.00,    589.94,     0.9405
rms_x =   0.7847
spectrum C:\Daten\Python\Koji\20171209\r_add23cal.dat saved
```

Additional features of the script

Tools

Edit text files

In m_spec it is possible to edit text files:



With Save the files are saved under the same name

Logfile

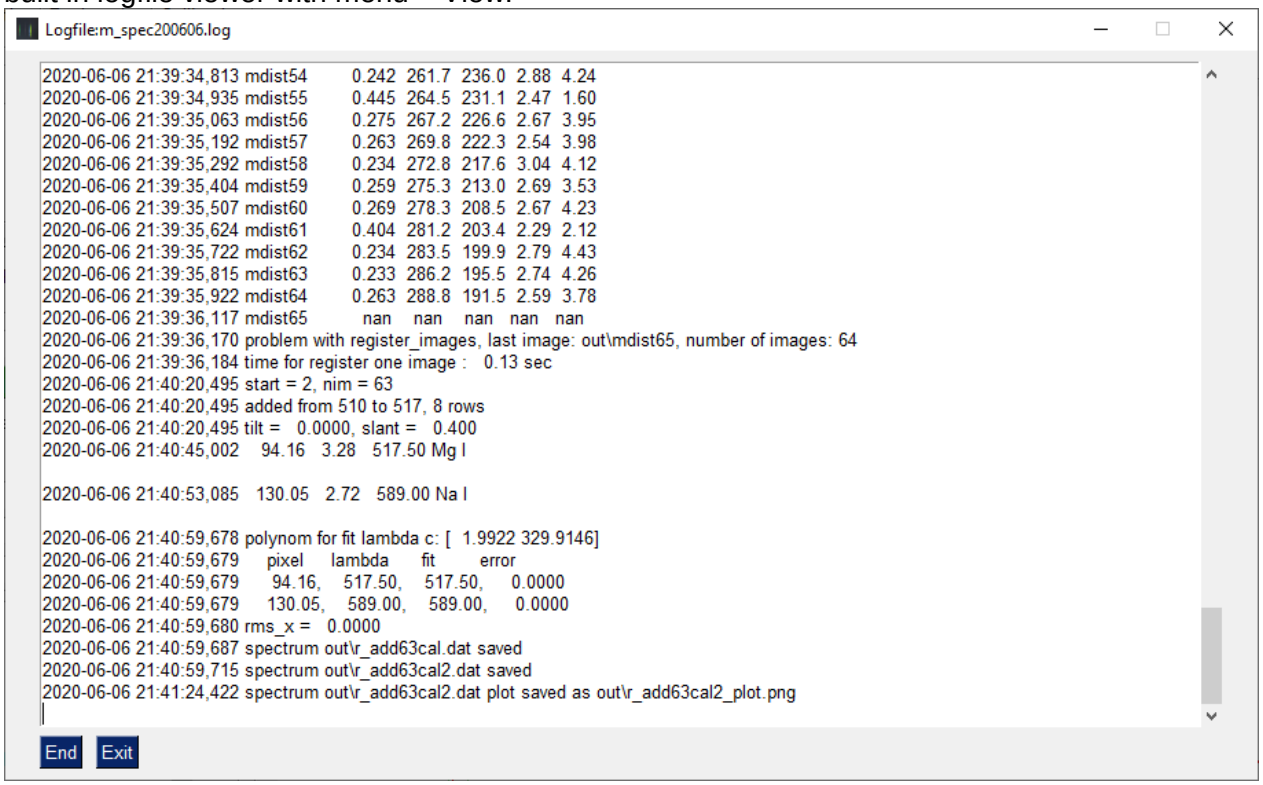
The main inputs and results of the script execution are stored in a logfile. You may review what you have done by reading the file m_spec(yymmdd).log, where (yymmdd) stands for the actual date in year, month day format. There you will find the main input and output for the complete processing from converting the AVI-file to the wavelength calibrated spectrum:

```
2020-04-03 15:59:51,525 M_SPEC START
++++++++++++++++++++++++++++++++++++++++++++++
2020-04-03 15:59:51,525 M_SPEC version 0.9.16, 0.9.16 ++++++++++++++++++++++++++++
2020-04-03 16:00:10,565 Platorm: Windows
2020-04-03 16:00:13,779 converted C:/Daten/Python/181014/M20181014_023817_MAI_2.avi
91 images
2020-04-03 16:00:13,779 Station = MAI_2 Time = 2018-10-14T02:38:17.000
2020-04-03 16:00:47,500 scalxy =     0.920
2020-04-03 16:00:47,500 x00 =   371.200
2020-04-03 16:00:47,500 y00 =   324.183
2020-04-03 16:00:47,500 rot =    -0.006
2020-04-03 16:00:47,500 disp0 =     1.965
2020-04-03 16:00:47,500 a3 = 3.137e-07
2020-04-03 16:00:47,500 a5 = -3.193e-13
2020-04-03 16:00:47,500 'DATE-OBS' = 2018-10-14T02:38:17.000
2020-04-03 16:00:47,500 'M-STATIO' = MAI_2
2020-04-03 16:00:57,735 67 images processed of 67
2020-04-03 16:00:57,735 process time for single distortion:     0.15 sec
2020-04-03 16:01:50,690 start x y, dx dy, file: 132 508,14 14, out/mdist
2020-04-03 16:01:50,700          out/mdist3  31.33 131.91 507.01   3.18   6.10
2020-04-03 16:01:50,776          out/mdist4  47.43 134.01 500.65   2.66   6.52
2020-04-03 16:01:50,998          out/mdist5  41.72 136.11 495.62   2.71   6.01
…
2020-04-03 16:01:55,362         out/mdist65  65.64 291.37 187.57   2.58   3.38
2020-04-03 16:01:55,414         out/mdist66  59.70 294.40 183.00   2.72   3.70
2020-04-03 16:01:55,488         out/mdist67  66.45 297.08 178.37   2.56   3.25
2020-04-03 16:01:55,532 time for register one image :   0.07 sec
2020-04-03 16:02:43,981 start = 3, nim = 65
2020-04-03 16:02:43,981 added from 503 to 514, 12 rows
2020-04-03 16:02:43,981 tilt =   0.0000, slant =   0.400
```

2020-04-03 16:04:31,615 Pixel: 95.75 FWHMp = 3.540 lambda = 517.500 Mg I

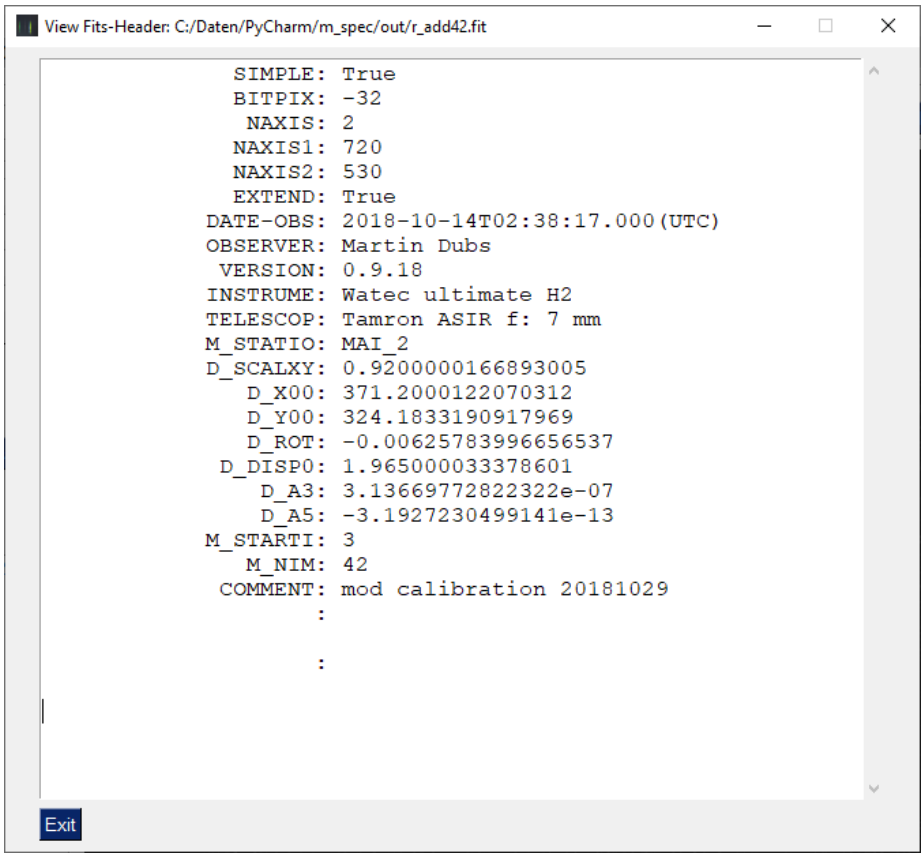
2020-04-03 16:04:41,126 Pixel: 131.44 FWHMp = 2.910 lambda = 589.000 Na I

2020-04-03 16:04:45,728 polynom for fit lambda c: [2.003 325.678]
2020-04-03 16:04:45,729 pixel lambda fit error
2020-04-03 16:04:45,729 95.75, 517.50, 517.50, 0.0000
2020-04-03 16:04:45,730 131.44, 589.00, 589.00, 0.0000
2020-04-03 16:04:45,730 rms_x = 0.0000
2020-04-03 16:04:45,738 spectrum C:\Daten\Python\out\r_add65cal.dat saved
2020-04-03 16:04:45,758 spectrum C:\Daten\Python\out\r_add65cal2.dat saved
2020-04-03 16:06:29,804 spectrum C:\Daten\Python\out\r_add65cal2.dat plot saved as 20181014_023817_MAI_2.png
You may edit the logfile with a text editor. For a quick view of the actual logfile you can use the built in logfile viewer with menu – View:

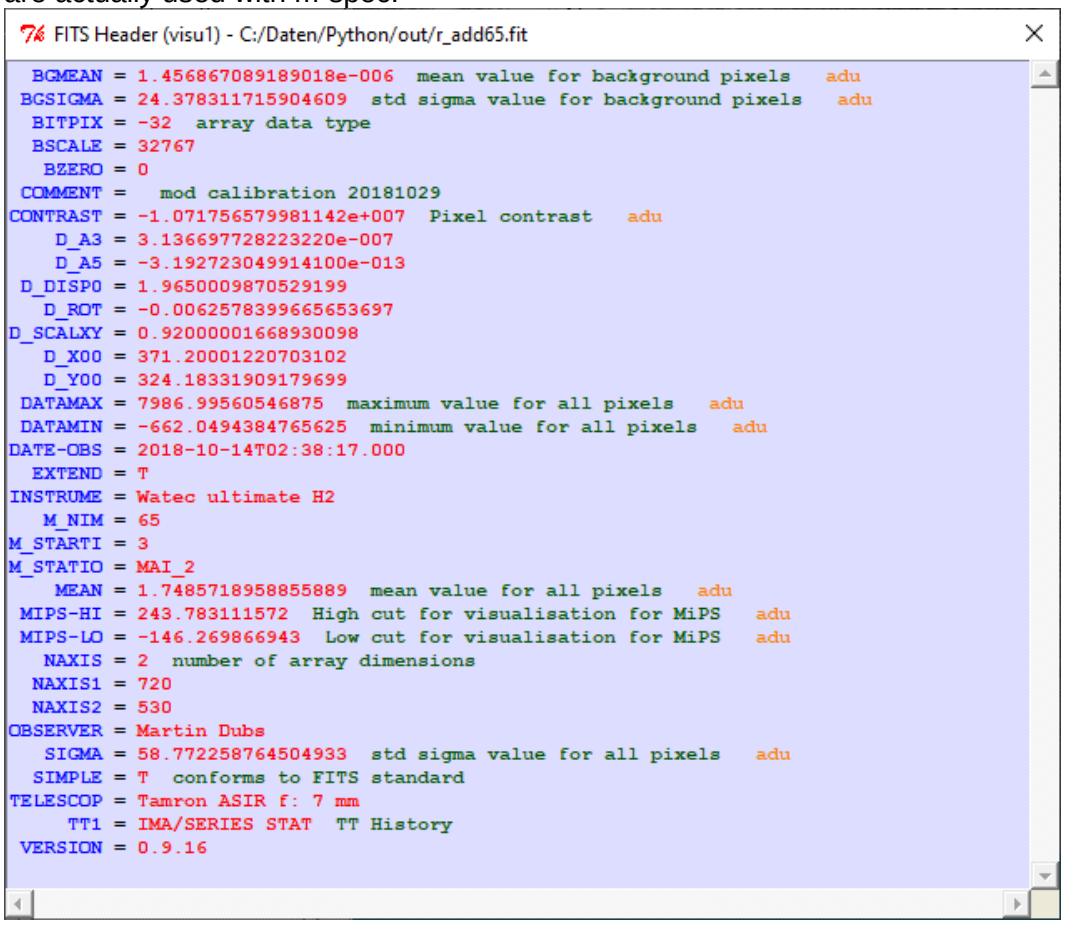


Built in Fits viewer

In the Menu – View – Fits header you can select any Fits File and look at its header information. This is quite useful, if you are not sure about the observation date or the used calibration:



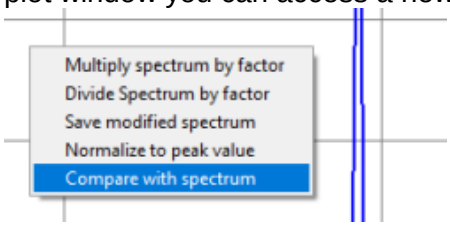
If you compare with another Fits-viewer, you will notice that you see only the keywords which are actually used with m-spec:



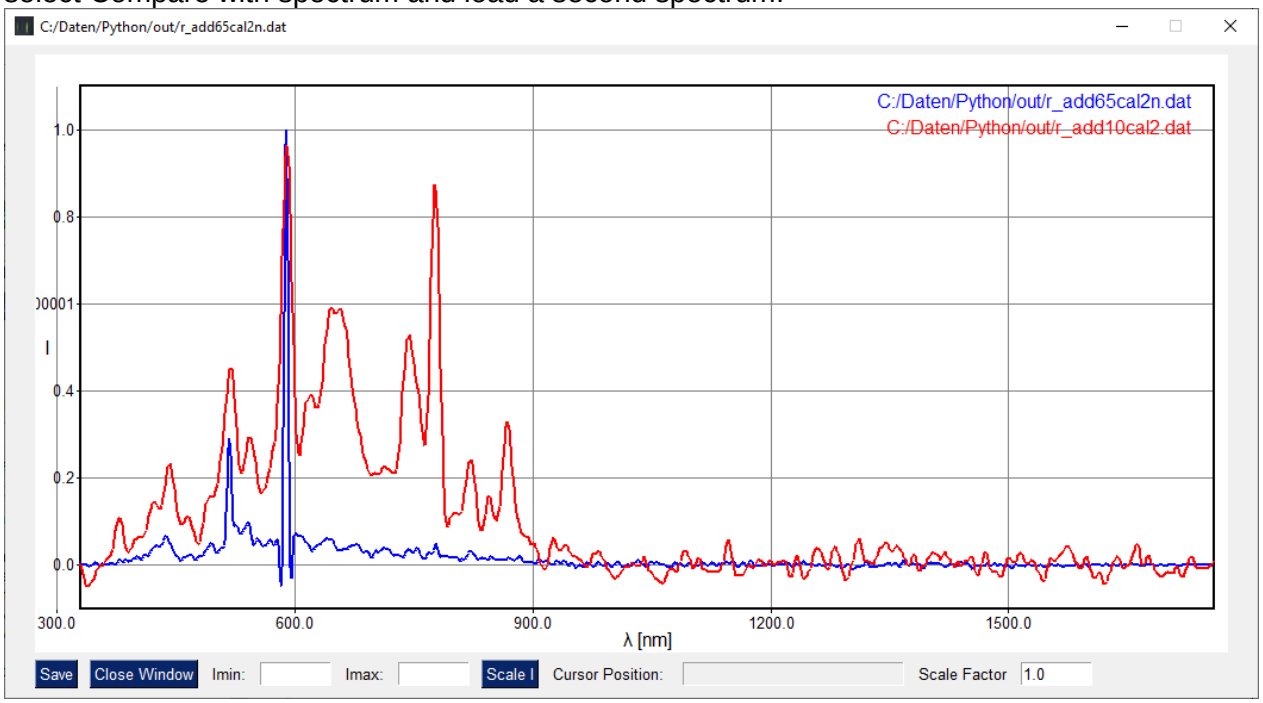
More plots

Compare spectra

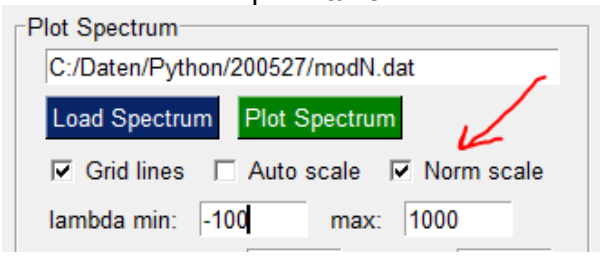
In addition to plot a single spectrum, you also have the possibility to compare two spectra. First select wavelength range, plot one spectrum and select intensity range. With a right click into the plot window you can access a new menu with different selections:

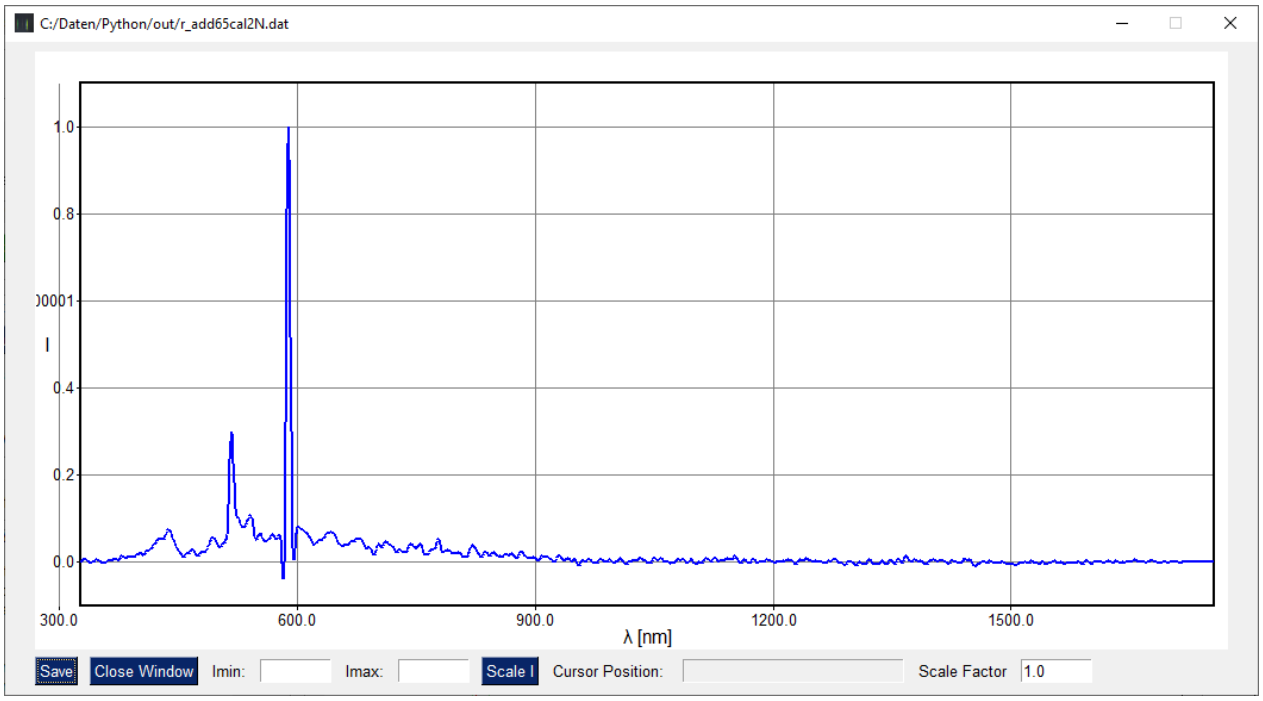


select Compare with spectrum and load a second spectrum:



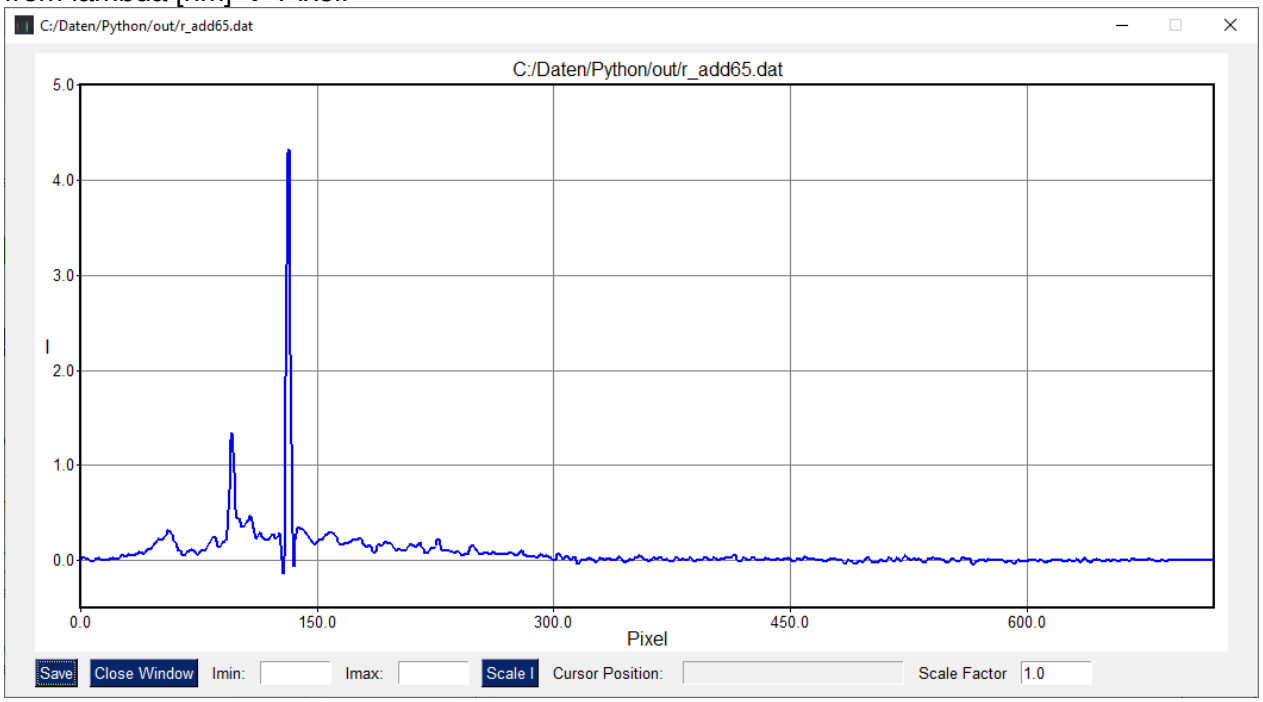
In order to get a nice view it may be necessary to scale the intensity of the spectra. This can be done with Multiply spectrum by a factor or Divide spectrum by a factor. The factor you select in the text input box at the bottom right of the plot window.
After the division or multiplication you have to store the scaled spectrum with right click – 'Save modified spectrum'
You also have the possibility to normalize the spectrum, so that the highest peak (often the zero order) has intensity one. Select right click – 'Normalize spectrum'. The file is stored automatically with 'N' added to the filename. r_add65cal2.dat → r_add65cal2N.dat. Close the window and plot again. In this case you can also use a normalized plot range by checking 'Norm scale' in the plot frame:





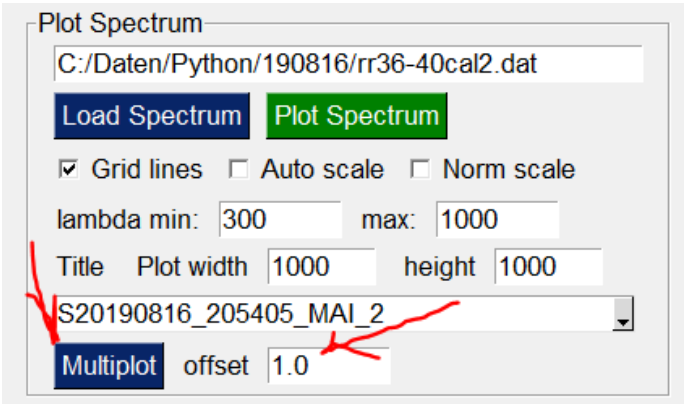
Plot raw spectrum

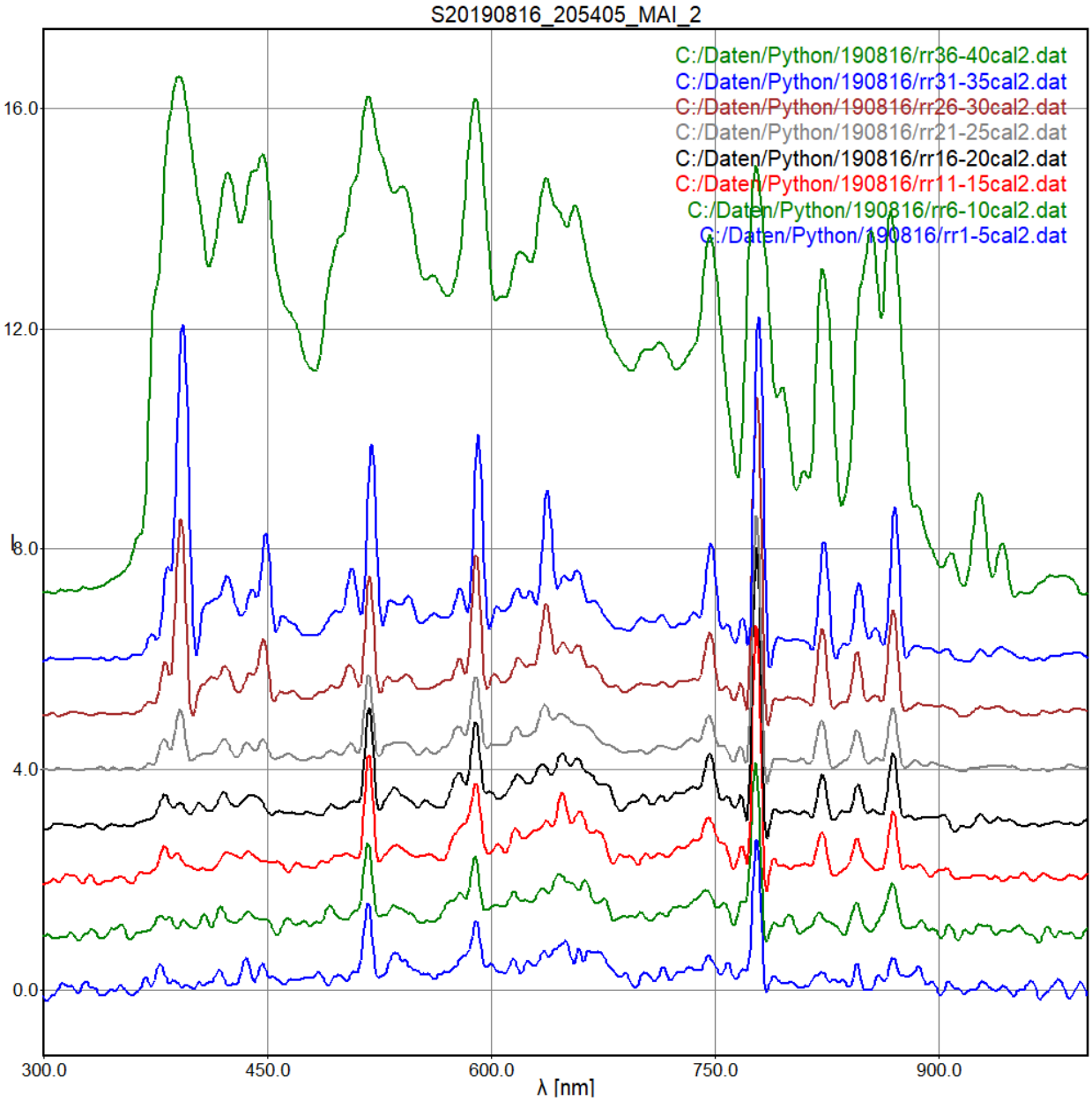
You may also plot the raw (uncalibrated) spectrum. In this case the label on the x-axis changes from lambda [nm] → Pixel:



Multiplot

If you would like to plot several spectra for comparison, as for a time series, it is also possible to plot several spectra in a single figure. Go to the calibration tab, select the wavelength range and the spacing of the different spectra with offset. A positive offset means that spectra are plotted from bottom to top, a negative offset plots them from top to bottom. The legends are sorted accordingly. With ‘Multiplot’ you can select the desired spectra and plot them:

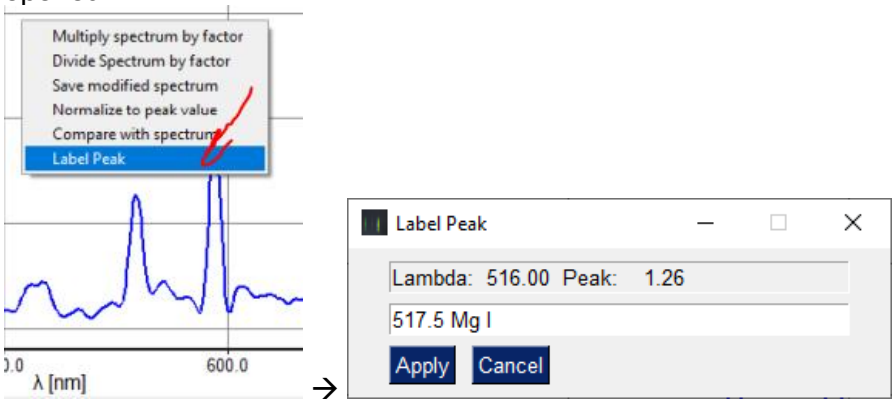




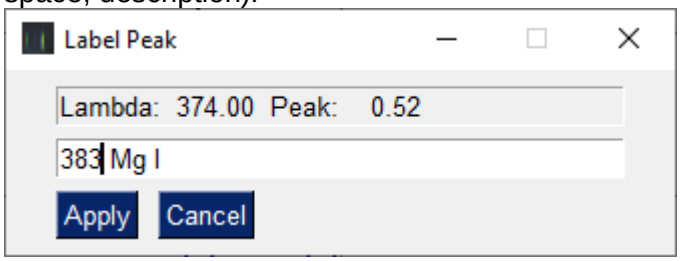
the last image (at the top) is quite strongly saturated! notable is also the increase of the Ca-Line.

Label spectra

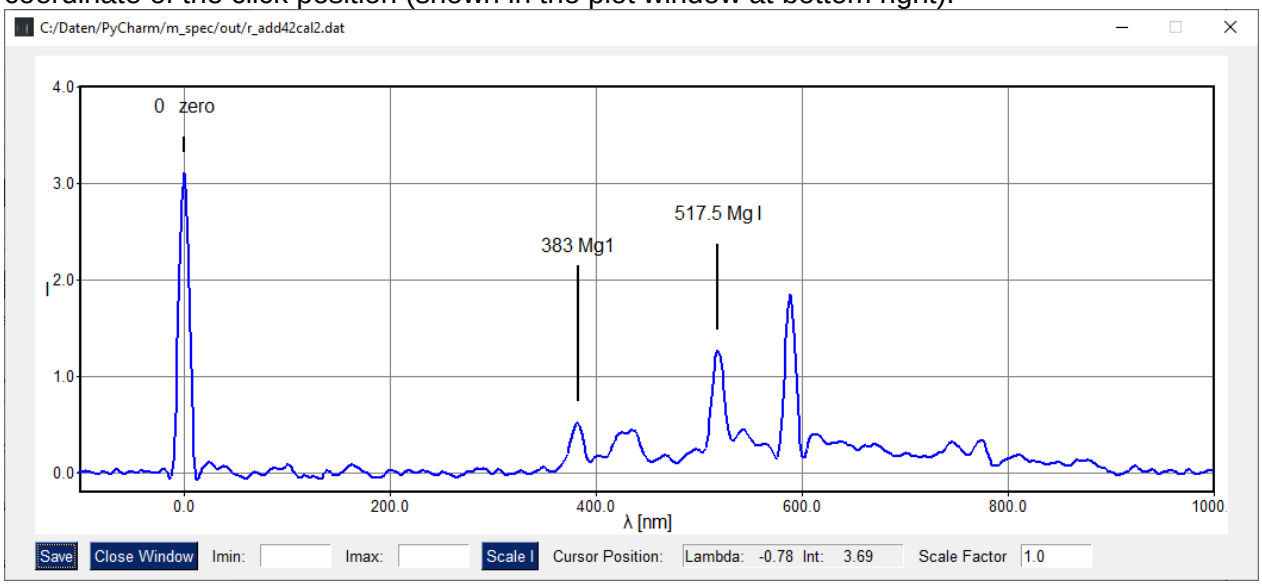
Sometimes it is useful to label peaks in the spectrum. This can be done by clicking above a peak in the spectrum. This records the click position. With a right click the label window can be opened:



In the upper field the click position of the wavelength and peak intensity at this wavelength is shown. The script searches in the file m_linelist.txt for the closest matching reference line. If the chosen peak is not in the linelist, you can edit this field (use the same format: wavelength, space, description):



The peak is marked at the indicated wavelenth. The y-coordinate of the label is given by the y-coordinate of the click position (shown in the plot window at bottom right).



If this feature is used often, the calibration line list can be edited. This may also be necessary if you work at higher resolution. Right now, the wavelength for close-spaced lines is given as a weighted average (for Mg and Na lines). If these lines are resolved in the spectrum, it is advisable to list the components separately, with wavelengths taken e.g. from NIST:
https://physics.nist.gov/PhysRefData/ASD/lines_form.html
Maybe the following list is helpful also, which gives also typical meteor line intensities (convert wavelengths to nm):

Table 3-7: List of spectral lines frequently found in meteor spectra and their relative intensities. Th tification of the lines (numbers) in our example is also given. Lines marked with an asterisk appear b spectra of fast meteors, such as the Perseids, but much fainter in spectra of slow meteors.

Laboratory data			ident.	Laboratory data			ident.
λ_{lab}, [Å]	atom/ion	intensity	number	λ_{lab}, [Å]	atom/ion	intensity	number
3719.9	Fe	10	2	4923.9	Fe^+	2*	
3734.9	Fe	8		4957.6	Fe	4	
3737.1	Fe	9	3	5012.1	Fe	1	
3745.6	Fe	8		5018.4	Fe^+	3*	
3749.5	Fe	8		5110.4	Fe	1	
3820.4	Fe	9		5167.3	Mg	17	
3825.9	Fe	8		5172.7	Mg	25	
3829.4	Mg	10		5183.6	Mg	28	
3832.3	Mg	11		5208.4	Cr	10	
3838.3	Mg	12		5227.2	Fe	5	
3859.9	Fe	11		5269.5	Fe	14	
3886.3	Fe	9		5328.0	Fe	12	
3933.7	Ca^+	40*	8	5371.5	Fe	9	
3968.5	Ca^+	35*	9	5397.1	Fe	5	
4030.8	Mn	10		5405.8	Fe	6	
4045.8	Fe	10		5429.7	Fe	6	
4063.6	Fe	9		5434.5	Fe	4	
4131.0	Si^+	1*		5446.9	Fe	4	
4226.7	Ca	11	12	5455.6	Fe	4	
4254.4	Cr	9		5528.4	Mg	2	
4271.8	Fe	10		5615.7	Fe	1	
4274.8	Cr	8		5890.0	Na	40	
4289.7	Cr	7		5895.9	Na	35	
4307.9	Fe	10		6156.8	O	1*	
4325.8	Fe	10		6162.2	Ca	1	
4383.5	Fe	14	15	6347.1	Si^+	6*	
4404.8	Fe	11		6371.4	Si^+	3*	
4481.2	Mg^+	15*		6495.0	Fe	1	
4920.5	Fe	3		6562.9	H	2*	

Different calibration methods

Use of spectrum lamp for calibration

For cameras with high dispersion, either with a long focal length or with a grating with large number of grooves/mm, not enough orders of the laser may fit on the detector size. Maybe a suitable laser is not at hand. In these cases it is possible to use a calibration lamp with several emission lines for calibration. Suitable lamps are low pressure Hg-lamps or Hg-Ar lamps. I have used successfully a lamp from Ocean Optics (http://oceanoptics.com/product/hg-1/) with several lines from UV to near-IR. An example is given in https://meteorspectroscopy.files.wordpress.com/2018/01/meteor_spectroscopy_wgn43-4_2015.pdf, although with low dispersion and somewhat problematic separation of spectral lines.

Therefore here an example with longer focal length is used (spectra recorded by Sipeng Yang). With a violet laser, only three orders fit on the image area, not enough for a good determination of the distortion parameters. With a green or red laser, only two orders fit in the image area, definitely not enough for a determination of the distortion parameters.
With the Hg-lamp, several spectral lines in addition to the zero order can be used for the fit:

Load image: calib/Sipeng/cal1_peak_5.png
5 spectra of a mercury argon lamp are shown, zero order at left, with first and order lines to the right. The spectra were recorded by Sipeng Yang[1], thanks for using these files. Equipment:
1. Watec 902H2 ULTIMATE 1/2' CCD, 720*576 Resolution
2. Lens: Hikvision HV0733D-6MP, F/0.95, f = 7-33mm zoom lens, operated at approx. f = 10mm
3. Grating: Edmund 600l/mm transmission grating
4. Camera FOV: 30.4 degree

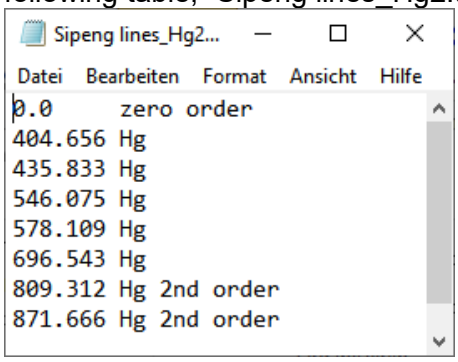
This is particularly useful for lenses with longer focal length and/or high resolution spectra and needs some knowledge about the spectrum of the lamp (which spectral line is where)
The bright line at left is the zero order, 5 Hg-lines 1st order are visible. Some second order lines are visible as well and can also be used for calibration.

Line list

When working with a spectral lamp, a slightly different procedure from the laser calibration will be used. Instead of a single laser wavelength, a list with the wavelengths of the spectral lamp in

[1] Beijing National Observatory for Space Environment
Institute of Geology and Geophysics, Chinese Academy of Sciences
NO.19 Beitucheng Xilu Street, Chaoyang District, Beijing, China 100029

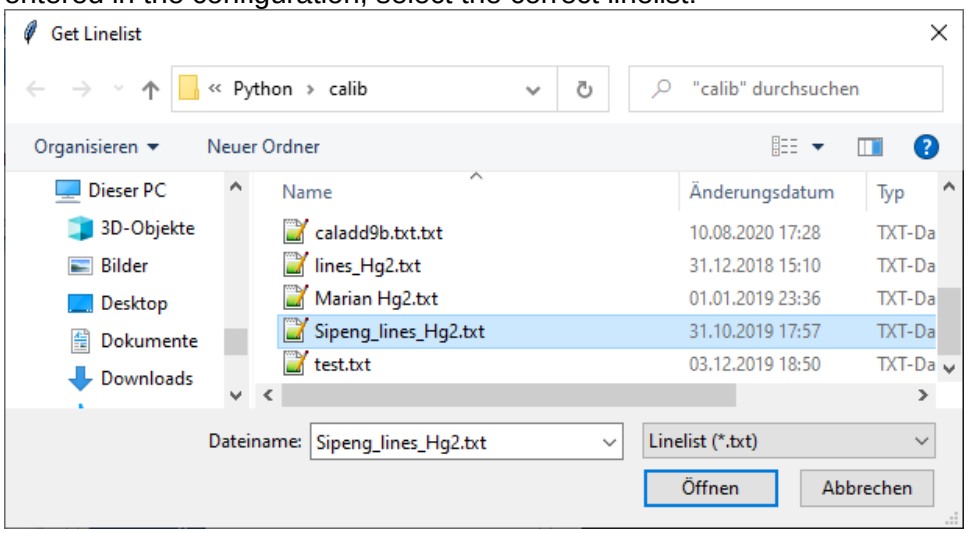
different orders is required. This is a textfile with two columns, the first column contains the wavelength*order (negative for negative orders, zero for zero order, wavelength for first order, 2*wavelength for 2nd order etc. as used here also for the plotting of spectra. The 2nd column must be separated by a tab and contains text information (element, order or whatever you like to add). The script actually reads only the first column, but make sure you end the column with a tab for compatibility with other scripts, which will use this file. It can be created with a text editor (Windows Editor or Notepad++ which I prefer). For the image of Hg spectra above I used the following table, “Sipeng lines_Hg2.txt”:



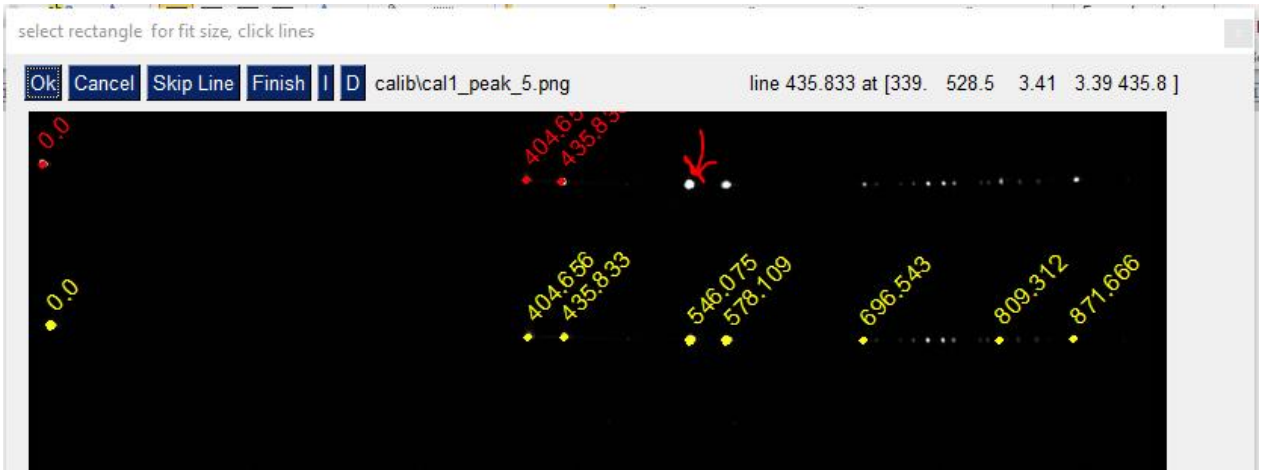
The line at 578.109 nm is actually the weighted average of two lines at 576.96 and 579.066 nm.

Run the script for spectral lamp calibration

Run the script as before, with the configuration ‘m_set_sipeng.ini’ and if the linelist has not been entered in the configuration, select the correct linelist:



You can also change the linelist manually in the setup.
Finish the setup with ‘Apply’.
Load the image and go to ‘Select Lines’.

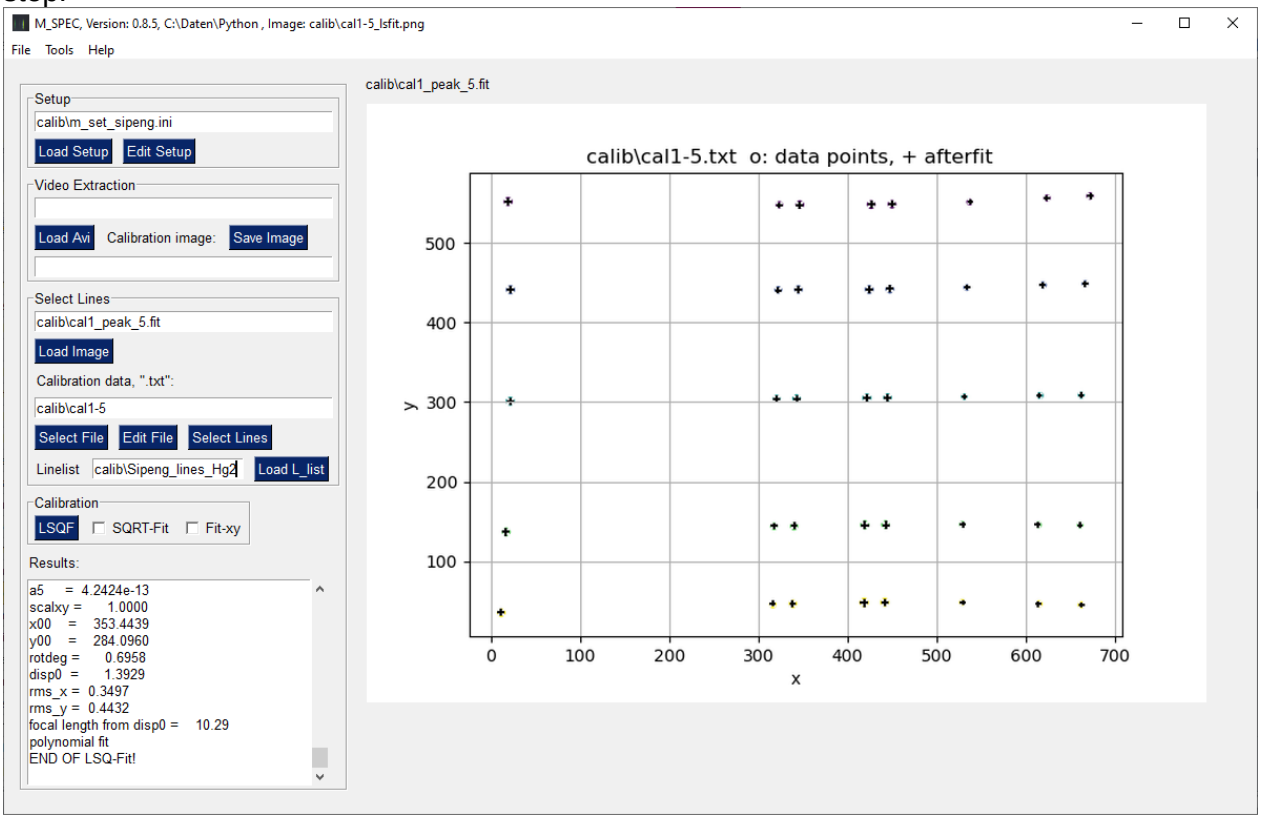


The selected lines are marked by a circle with wavelength*order indicated. They have been selected to cover the full width of the sensor with regular spacings. Some weak lines have been omitted. Finish each spectrum with **OK**

Continue with the other spectra, Skip missing lines with 'Skip Line' (e.g. if the leftmost line is out of the image area or if a line is too weak). Select only the lines which you can assign correctly. Finish the list with **OK** again.

It is important that you choose the correct lines! If you select the wrong lines, the errors will be large. Start over with a new calibration file and select the correct lines!

Proceed the same way as for the laser calibration (edit the list if necessary, then go to the LSQF step.



If you would like to run the least square fit with other parameters (a sqrt fit would be suitable here for the fairly long focal length) go to the Setup and edit the corresponding field(s).

The results are from the logfile:

Result LSQF: **polynomial fit**	Result LSQF: **square root fit**
==========	==========
Watec 902 H2 ultimate, f = 10mm	Watec 902 H2 ultimate, f = 8mm
Sipeng\lamp_video\cal1-5.txt	Sipeng\lamp_video\cal1-5.txt

Parameters after fit: a3 = 1.7052e-07 a5 = 6.5356e-13 scalxy = 0.9240 x00 = 352.6166 y00 = 283.4155 rotdeg = 0.6431 disp0 = 1.3906 rms_x = 0.3387 rms_y = 0.4269 focal length from disp0 = 10.31 polynomial fit END OF LSQ-Fit!	Parameters after fit: a3 = 2.7372e-07 a5 = 1.1239e-13 scalxy = 0.9671 x00 = 354.5030 y00 = 282.9148 rotdeg = 0.6752 disp0 = 1.3969 rms_x = 0.3248 rms_y = 0.4764 focal length from disp0 = 10.26 for sqrt fit: feff = 11.62 END OF LSQ-Fit!

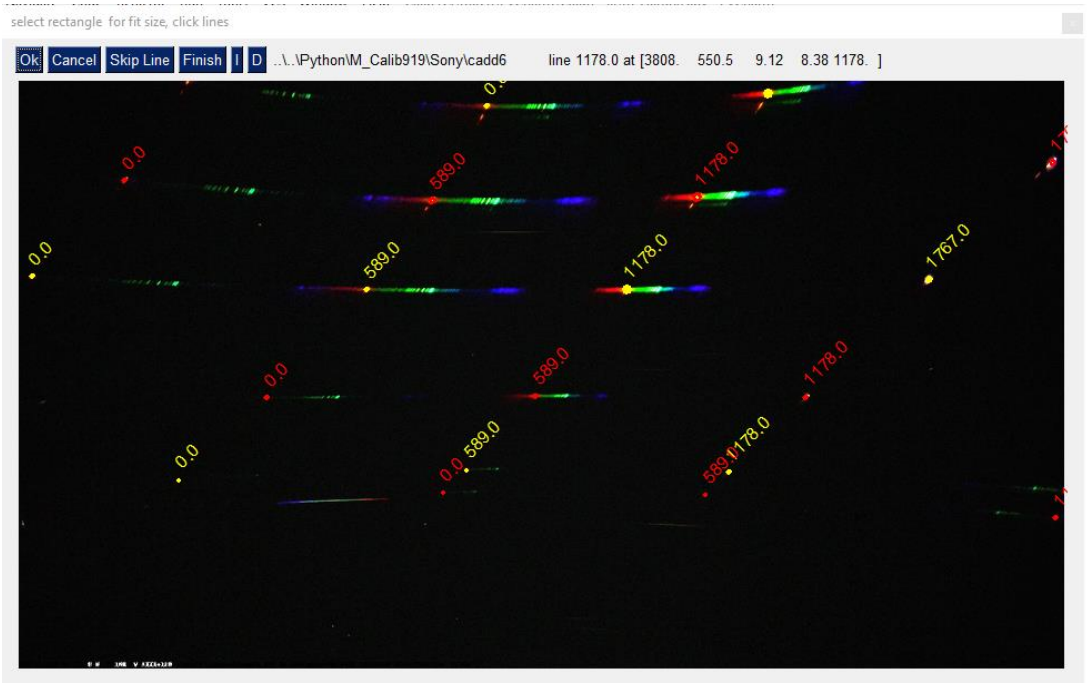
Since the errors of the fit are similar, I would prefer the sqrt fit, but this is a matter of taste. Another possibility would be to constrain the value of scalxy to a more realistic value, as determined by UFO capture or calculated from the image size and pixel size and detector size. An rms error of < 1 pixel is the acccuracy which can be expected, subpixel accuracy is achieved with determining the line positions with a Gaussian fit.

Use of meteor spectra for calibration

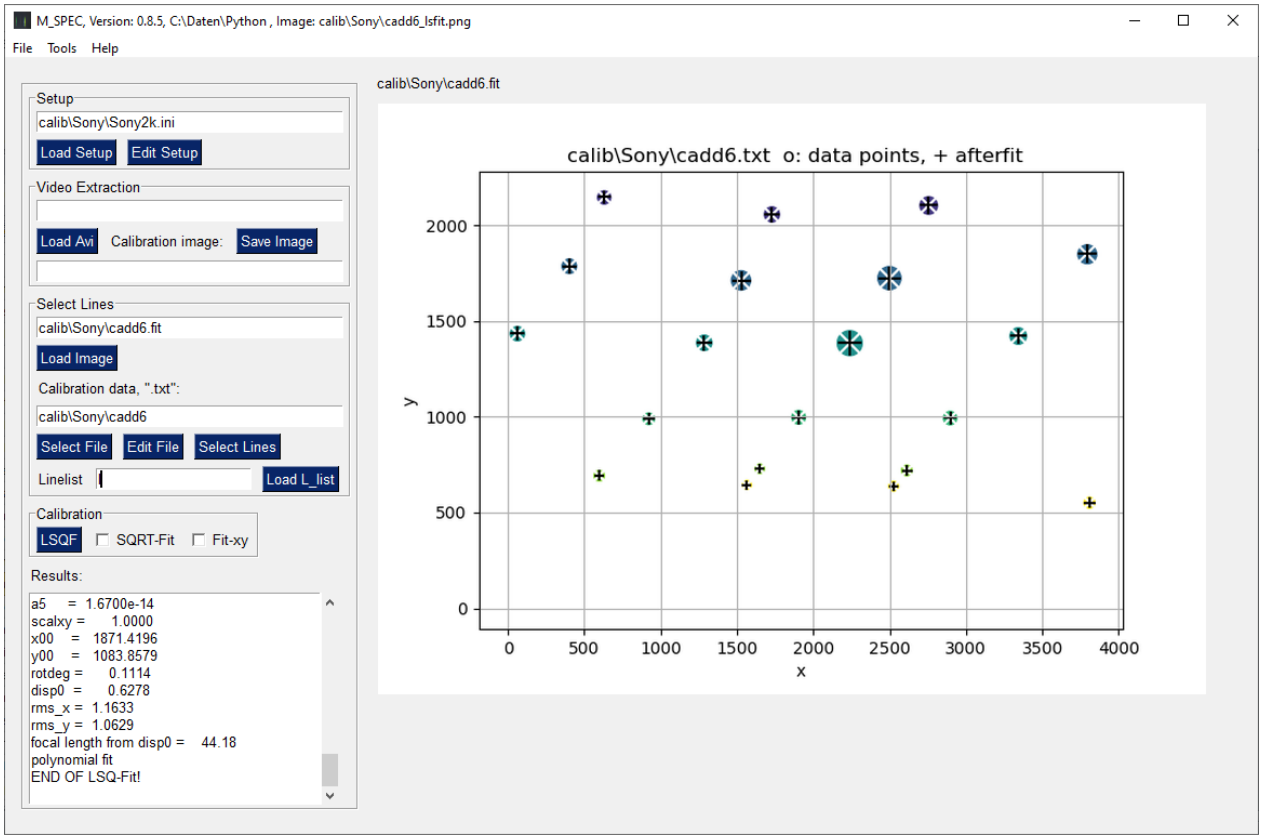
When a meteor spectrum with identified spectral lines covers a large portion of the image area, it can also be used for the determination of the distortion parameters. This is useful if no laser or spectral lamp spectra have been recorded or if the camera setup has been changed in the meantime. I present an example which has previously been analysed with the old method. The spectrum was recorded by Koji Maeda with a Sony Alpha 7S equipped with a Canon 24 mm F/1.4 lens (used at F/2) and a 600 L/mm grating. The meteor spectrum was captured as a 4K video (image size 3840 x 2160 pixels) at 30 images/sec with SonotaCo UFOCaptureHD. The meteor was observed for about 5 sec and reached magnitude -3.7m.

M20171209_041224_JPMZ1, Koji Maeda

The Na-lines are easily identified and can be used for calibration. The spectrum pre-processing was done with m_pipe6 including the AVI conversion and background subtraction.
Several colour fit-images were peak-added and saved as cadd6.fit.



For simplicity I only used the Na-lines, therefore no list of spectral lines was needed, I used the "laser" calibration. Notice that the orders are not correct, important is that the wavelength differences are constant between orders.
With the fit:



The size of the data points corresponds to the Gaussian width of the lines, increased by saturation, movement of the meteor and meteor train
The fit parameters agree with earlier calculations:
Result LSQF:
==========
Sony alpha, f 24 mm

sony/cadd6.txt
Parameters after fit:
a3 = 4.3292e-08
a5 = 1.6700e-14
scalxy = 1.0000
x00 = 1871.4196
y00 = 1083.8579
rotdeg = 0.1114
disp0 = 0.6278
rms_x = 1.1633
rms_y = 1.0629
focal length from disp0 = 22.09
polynomial fit
END OF LSQ-Fit!

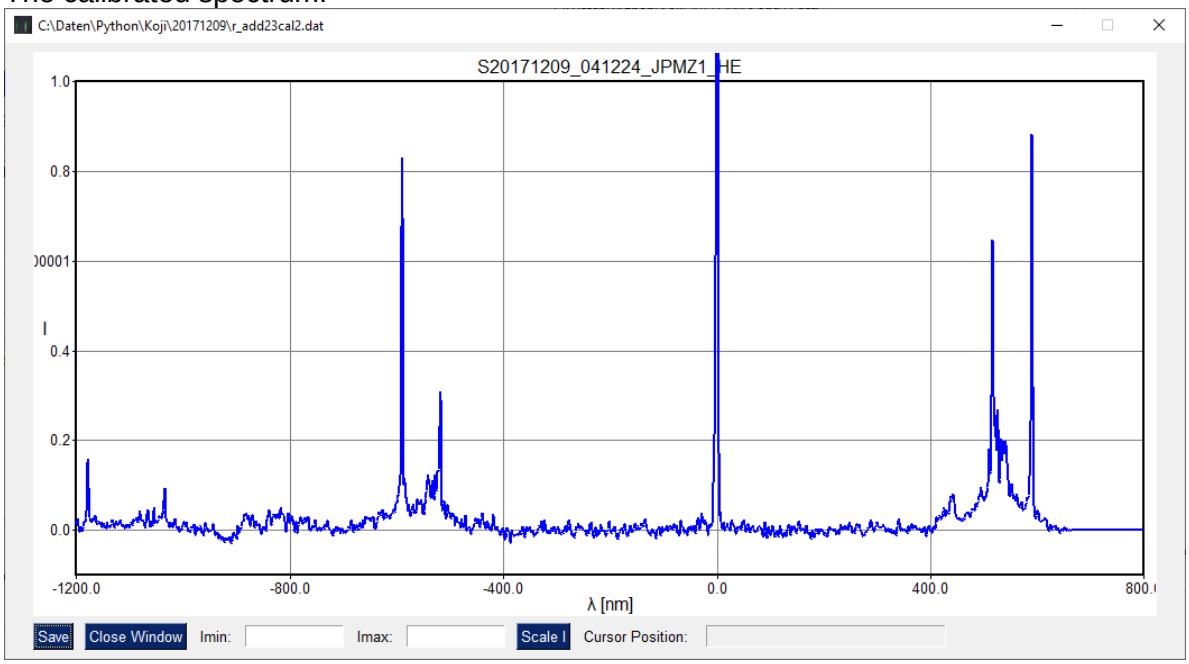
With these parameters, also saved as m_set.ini, the full spectrum can be analysed.

Analysis of spectrum M20171209_041224_JPMZ1

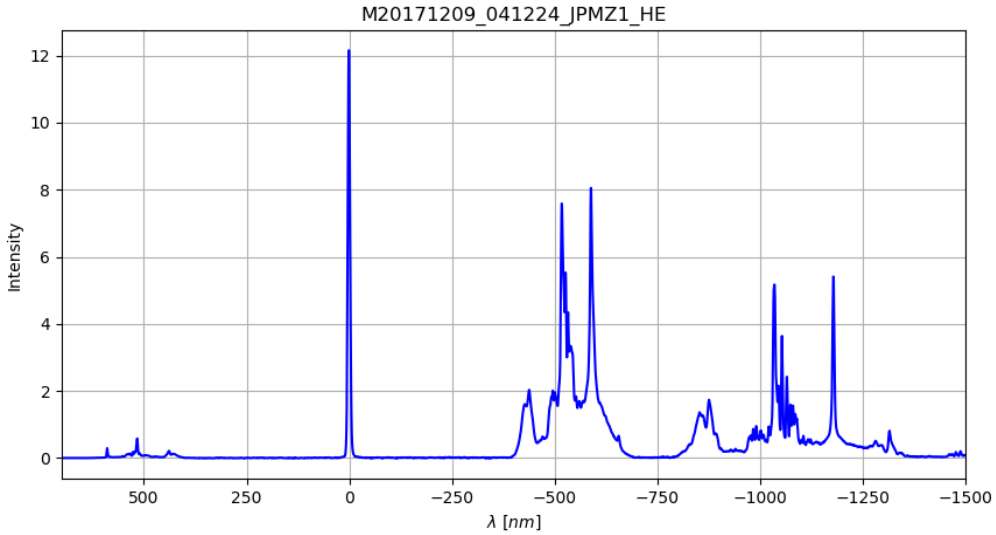
(see: http://sonotaco.jp/forum/viewtopic.php?p=47795#47795)
The distortion parameters obtained from the calibration with the Na-lines were used to calibrate the meteor spectrum, analysed with m_spec.
After adding 23 registered files (about the 1st second of the meteor (near the bottom of the image9 and calibrating with Mg- and Na-lines the following result was obtained:
A linear fit was used:
polynom for fit lambda c: [1.259e+00 -1.503e+03]

pixel	lambda	fit	error
257.95,	-1178.00,	-1177.83,	0.1668
725.21,	-589.00,	-589.48,	-0.4819
1193.86,	0.00,	0.62,	0.6196
1603.37,	517.50,	516.25,	-1.2451
1661.89,	589.00,	589.94,	0.9405

rms_x = 0.7847
spectrum C:\Daten\Python\Koji\20171209\r_add23cal.dat saved
Not surprisingly, a linear fit works well.
The calibrated spectrum:



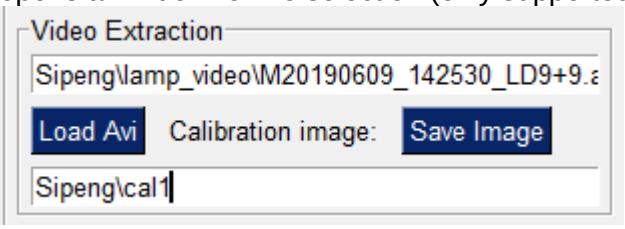
Below, the spectrum obtained from adding all image frames(123)



The spectrum was plotted with reverse wavelength scale, to show the strongest lines to the right of the zero order. The actual first order visible around 500 nm was very weak, because it was mostly outside the image area.

Creating calibration images from video files

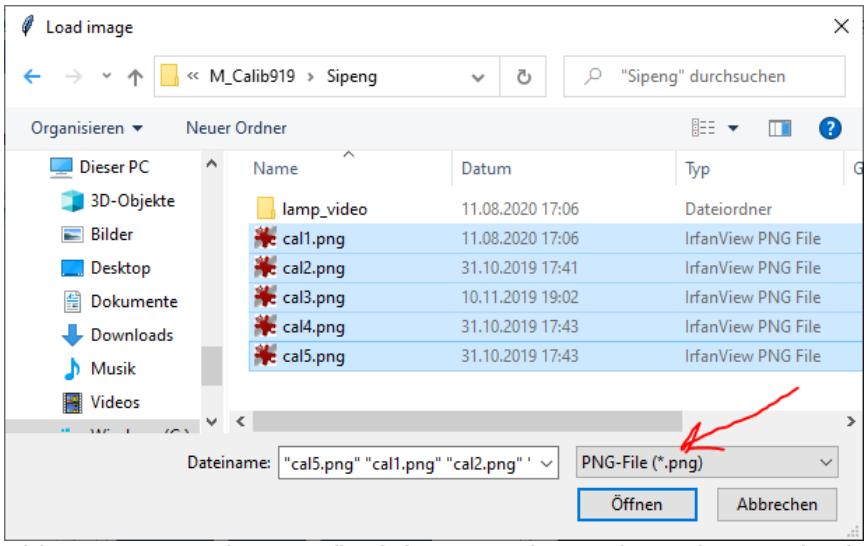
For the previous versions of this calibration script, the images used for calibration had to be created with different software (IRIS). In order to streamline the workflow, the creation of calibration images has been included in this script. For calibration images, short video clips of the laser or spectral lamp are recorded. These can be read with the button ‘Load Avi’, which opens a window for file selection (only supported format AVI, as produced by UFO capture):



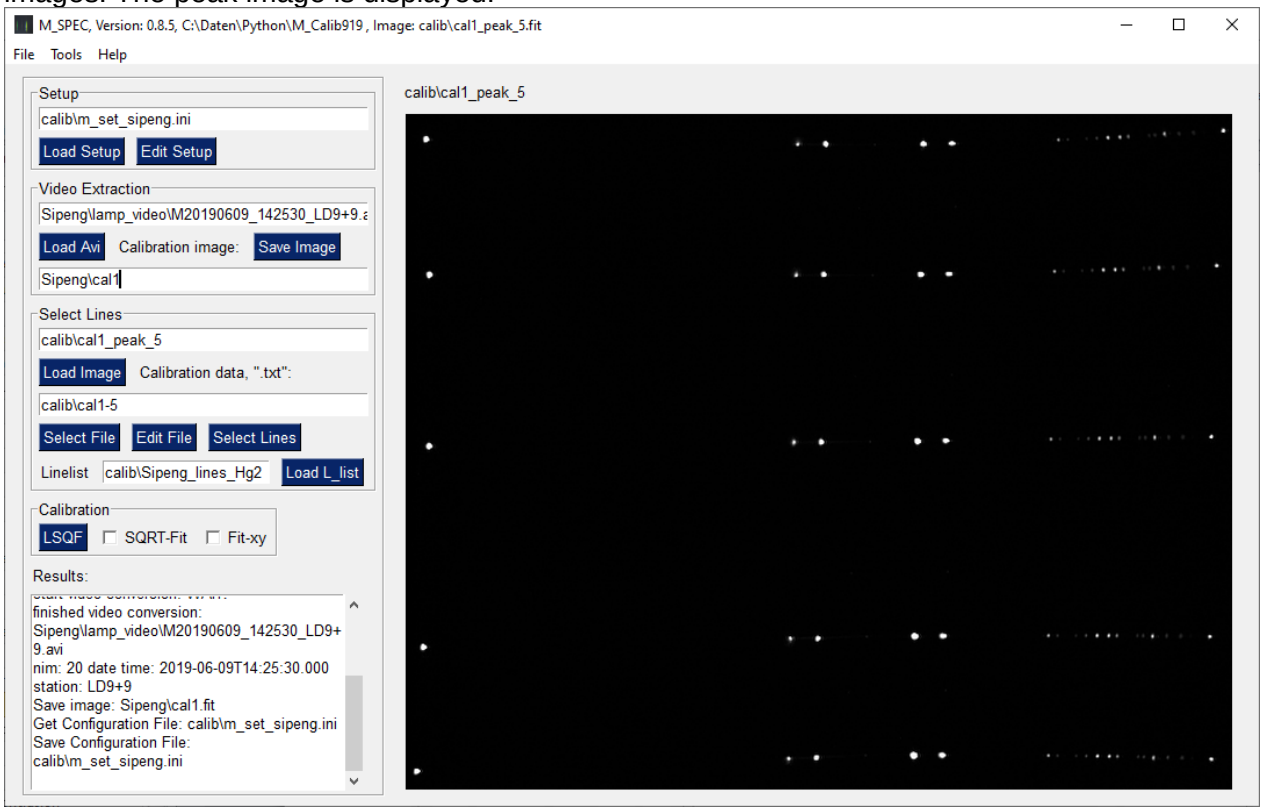
From the selected video file, a number of frames (20 at present) are converted to png images and from these a peak image and an average image are calculated with ‘Load Video’. After a short delay the image can be saved with an appropriate filename by selecting the directory with the browser button ‘…’ and adding the filename in the image window:
If the videos are not in the demo files (for saving file size), try with your own meteor video file (a background image will be produced).

Making peak images from single calibration images

You can load an image with a single spectrum, determine the line positions and then load another image, repeating these steps until all spectra are measured. However, it is more convenient to create a peak image from all the relevant calibration spectra. Previously this had to be done with IRIS or some other imaging software. In the present version of m_calib, this is simplified considerably. Instead of loading the image files one by one, the whole series can be selected at once in the browser ‘…’ for the actual image:

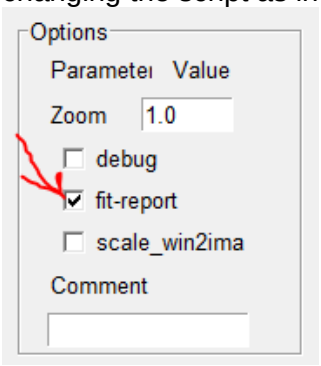


This creates two images (both in FIT and PNG format) named cal1_peak_5.png and cal1_ave_5.png(the filename is created from the name of the first image plus ‘,-peak-‘ plus number of images. Notice that the peak images can be produced from *.fit, *.png and *.bmp images. The peak image is displayed:



Detailed results of Least Square Fit

A detailed view of the results of the fit is given with the command "**report_fit(out)**", which shows statistical errors, correlations etc. in the shell window for the interested reader. Instead of changing the script as in a previous version, it can be switched on in the setup options:



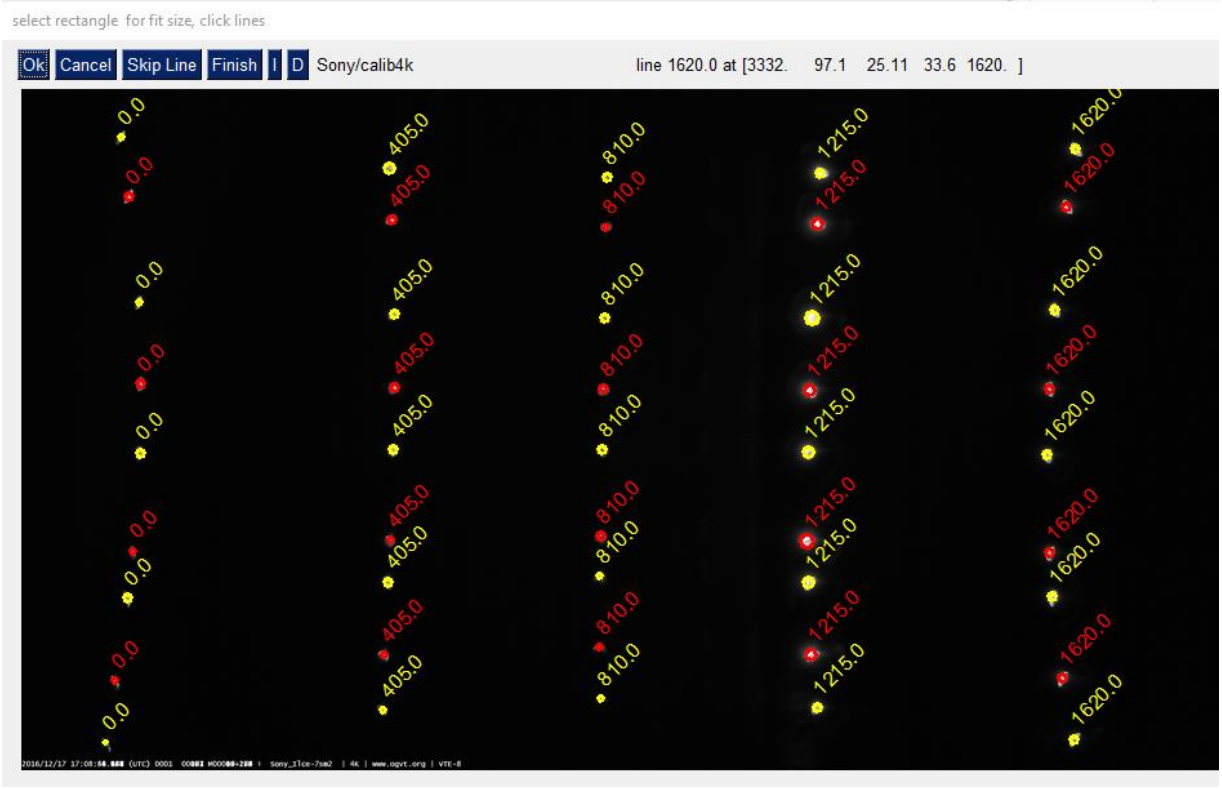
For practical purposes the average rms error of the fit in x- and y-coordinates rms_x and rms_y are shown. With a well corrected and focused lens these values should be < 1 pixel. With noise free and aberration free synthetic data produced with ImageTools, rms-errors were < 0.05 pixel.

```
>>> report_fit(out)
[[Fit Statistics]]
    # fitting method   = leastsq
    # function evals   = 210
    # data points      = 108
    # variables        = 25
    chi-square         = 10.1468068
    reduced chi-square = 0.12225068
    Akaike info crit   = -205.416995
    Bayesian info crit = -138.363714
[[Variables]]
    x0_0:     49.0181527 +/- 0.40095395 (0.82%) (init = 48.39159)
    y0_0:     501.714124 +/- 6.76732160 (1.35%) (init = 515.1632)
```

...

```
    scalxy:   0.85217823 +/- 0.03405730 (4.00%) (init = 0.92)
    x00:      344.900732 +/- 1.73905549 (0.50%) (init = 345.386)
    y00:      327.219815 +/- 3.81784835 (1.17%) (init = 322.3328)
    rot:     -0.00578993 +/- 3.1517e-04 (5.44%) (init = -0.006249105)
    disp0:    1.99038480 +/- 0.00260282 (0.13%) (init = 1.987447)
    a3:       2.9385e-07 +/- 3.1388e-08 (10.68%) (init = 2.402625e-07)
    a5:      -1.8900e-13 +/- 1.8562e-13 (98.21%) (init = 8.658912e-14)
[[Correlations]] (unreported correlations are < 0.100)
    C(y0_7, y0_8)     =  1.000
    C(y0_0, y0_1)     =  0.999
```

Notice, that the scalexy and the coefficient a5 are not well defined by the chosen data points (4 points in one spectrum cannot define the 5th power of a polynomial.



m_set.ini - Editor

Datei Bearbeiten Format Ansicht ?

```
[Lasercal]
f_lam0 = 405.0
f_scalxy = 1.0
b_fitxy = 0
i_imx = 3840
i_imy = 2160
f_f0 = 24.0
f_pix = 0.00832
f_grat = 600.0
f_rotdeg = 0.0
i_binning = 2
s_comment = Sony alpha, f 24 mm
s_infile = sony2k/calib2k
s_outfil = sony2k/calib2k

[Calib]
x00 = 953.3508684132236
y00 = 547.0873636640123
rot = -0.012352211369287523
disp0 = 1.2763356272969484
a3 = 1.8386668878523818e-07
a5 = 2.0314624623241l2e-13
```

```
Sony alpha, f 24 mm
sony2k/calib2k.txt
rms_x = 0.2861
rms_y = 0.4682
parameters after fit:
…
\sony2k\m_set.ini saved
```

m_cal.ini - Editor

Datei Bearbeiten Format Ansicht Hilfe

```
[Lasercal]
f_lam0 = 405.0
f_scalxy = 1.0
b_fitxy = 0
i_imx = 3840
i_imy = 2160
f_f0 = 24.0
f_pix = 0.00832
f_grat = 600.0
f_rotdeg = 0.0
i_binning = 1
s_comment = Sony alpha, f 24 mm
s_infile = Sony/calib4k
s_outfil = Sony/calib4k
s_linelist = 1
b_sqrt = 0

[Calib]
scalxy = 1.0
x00 = 1908.5919
y00 = 1099.8146
rot = -0.0051855845
disp0 = 0.6342405
a3 = 5.0031932e-08
a5 = 1.3980166e-14
```

100% Windows (CRLF) UTF-8

```
Sony alpha, f 24 mm
Sony/calib4k.txt
rms_x = 1.5168
rms_y = 1.8173☹
parameters after fit:
… Save config in directory
Sony\Sony4k.ini
```

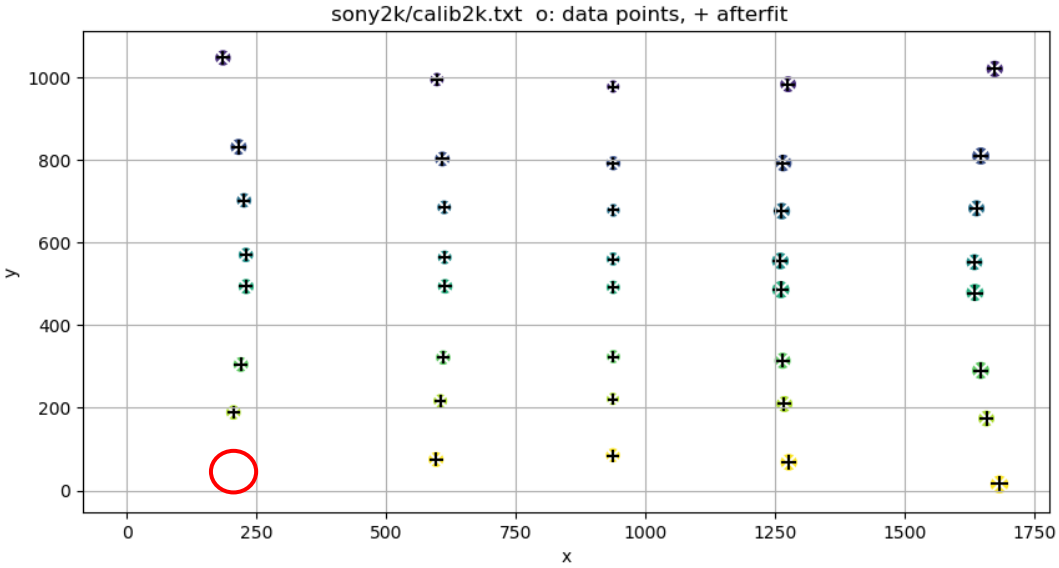
Note:

The parameter names in the configuration are preceded by a letter indicating the type of the variable. This simplifies the conversion. f_ is for floating point parameters, i_ for integers, s_ for strings. b_ is for boolean variables, **True** replaced by 1 and **False** by 0.

After finishing a calculation, the relevant parameters are stored in the file m_cal.ini. It corresponds to a Windows-style *.INI and contains the input parameters under the heading [lasercal] and the results of the least square fit under the header [Calib]. The input and output directories you can change during the running of the program l_calib.py. Other changes have to be done manually by editing the file m_set.ini with a text editor such as Notepad.
Next time you run l_calib6.py the new values will be used.

Binning, directory structure

If you have different cameras, lenses, image formats you may need more than one configuration. In this case you can use different subdirectories or different ini-filenames for each case, in this example there is a directory “sony4k” for 4k images and a directory “sony2k” for binned 2k images. The only difference in the configuration is the binning = 2 for the 2k images, as compared to binning = 1 for the 4k images. Of course the results for [Calib] are different for the different binning factors; the values of x00 and y00 are 2 times larger for the unbinned images, the values of a3 and a5 4x respectively 16x smaller for the unbinned images. Notice also that the errors are much smaller in the 2k images. This is because the first line in the lowest laser spectrum was omitted in the 2k analysis (this is permitted for the first or the last order, but you may not skip intermediate orders), since it overlapped partly with the image caption:



The results are included in the configuration file, because they will be used in the analysis of the meteor spectra, for the calculation of the distortion to transform to orthographic projection.

Content