

process_filter_10nm

June 6, 2025

Laden der notwendigen Bibliotheken

```
[1]: import os
import sys
import matplotlib.pyplot as plt
from matplotlib.image import AxesImage, NonUniformImage
import numpy as np
import math
from astropy.io import fits
from pathlib import Path
from scipy.signal import find_peaks
```

Festlegung der Dispersionsbeziehung aus determine_dispersion.ipynb als Polynom 1. Grades (Grade)

```
[2]: z = [5.16220206e-01, 3.93665555e+03]
p = np.poly1d(z)
positions = range(0,4944,1)
waves = p(positions)
```

Definition des Pfades der Filterkurve

```
[3]: fits_name = 'Light_FILTER-10NM_1sec_Bin1_23.
↳ 2C_gain0_2025-06-04_211719_frame0001.fit'
filter_name = fits_name.split('_')[1]
homepath = Path('.')
fits_path = homepath / '..' / 'data' / filter_name
fits_file = fits_path / fits_name
```

Einlesen der Fits Datei

```
[4]: with fits.open(str(fits_file)) as hdul:
    hdul.info()
    data = hdul[0].data
```

Filename: ../data/FILTER-10NM/Light_FILTER-10NM_1sec_Bin1_23.2C_gain0_2025-06-04_211719_frame0001.fit

No.	Name	Ver	Type	Cards	Dimensions	Format
-----	------	-----	------	-------	------------	--------

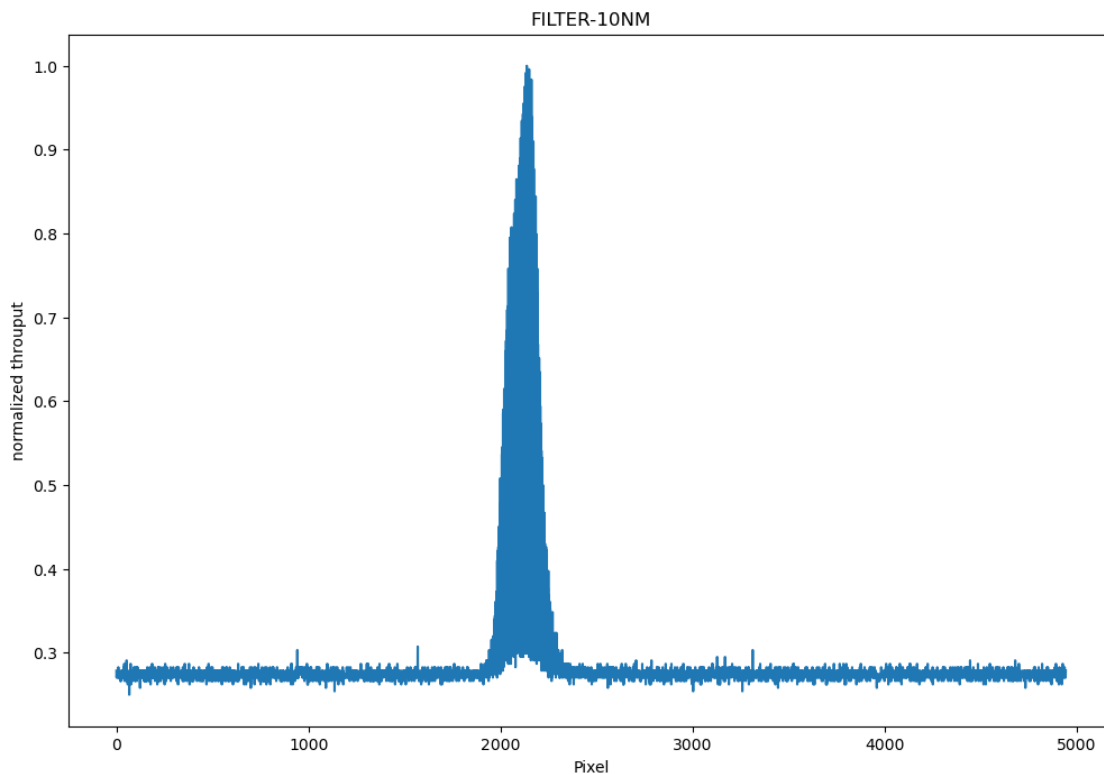
```
0 PRIMARY      1 PrimaryHDU      40  (4944, 3284)  int16 (rescales to
uint16)
```

Auswahl eines Schnittes durch das Bild entlang der Dispersionsachse, bei $y=1500$

```
[5]: trace = data[1500,:]
     trace_n = trace/max(trace)
```

Plot des Schnittes

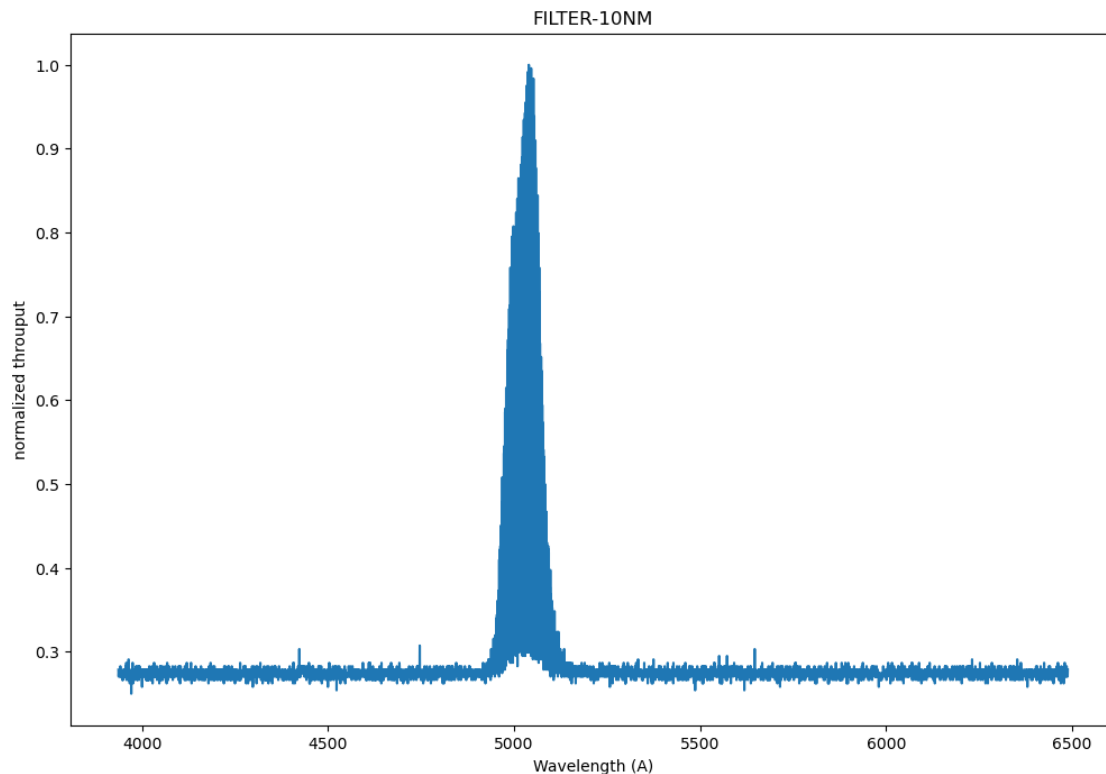
```
[6]: plt.rcParams['figure.figsize'] = (12,8)
     fig, axes = plt.subplots()
     plt.plot(trace_n)
     plt.xlabel('Pixel')
     plt.ylabel('normalized throuput')
     plt.title(filter_name)
     plt.show()
```



Plot des Schnittes, mit Wellenlängenzuordnung

```
[7]: plt.rcParams['figure.figsize'] = (12,8)
     fig, axes = plt.subplots()
     plt.plot(waves, trace_n)
     plt.xlabel('Wavelength (A)')
```

```
plt.ylabel('normalized throuput')
plt.title(filter_name)
plt.show()
```



Gesucht ist die Halbwertsbreite des Filters

Es wird ein Abschnitt aus dem Schnitt gemittelt: dieser Wert entspricht in etwa dem Untergrund ('dark'). Für die Bestimmung der Halbwertsbreite wird die Breite des Filterprofiles gemessen, die der halben Intensität - dark entspricht.

```
[8]: dark = np.mean(trace_n[100:500])
     half_intensity = (1-dark)/2+dark
```

numerische Bestimmung der Halbwertsbreite

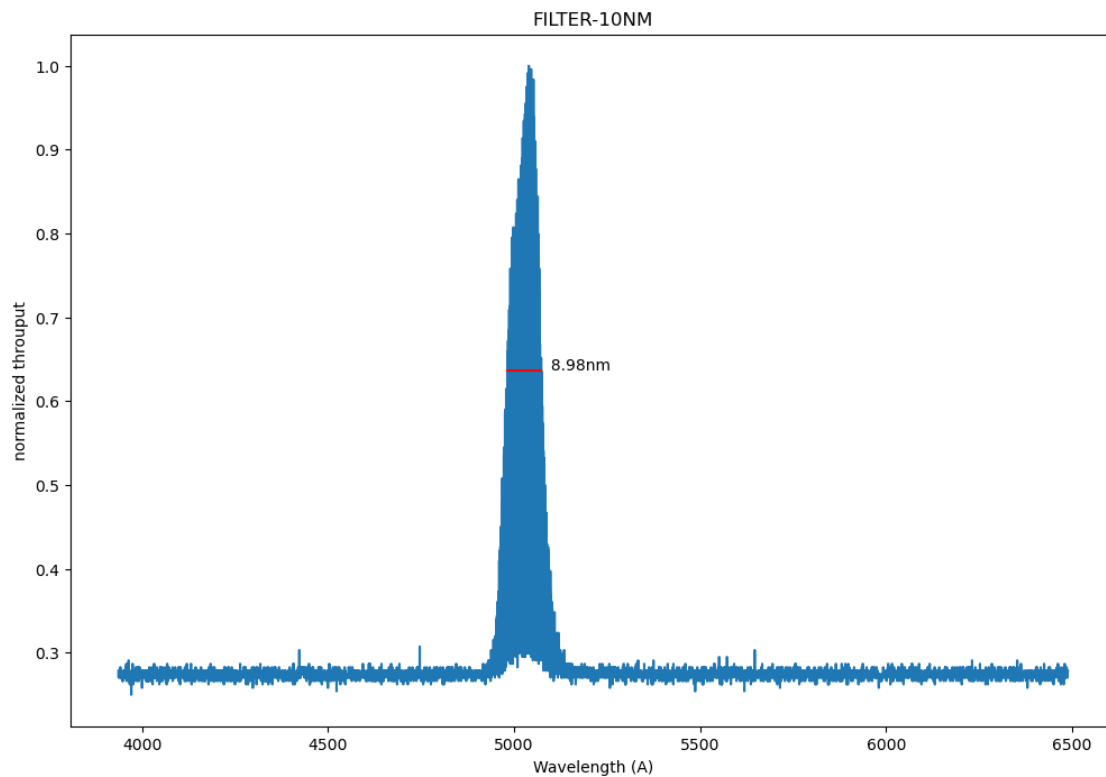
Finde alle Positionen des Schnitts, deren Wert größer als die "half_intensity" ist. Die Halbwertsbreite ist dann die Differenz zwischen der gefundenen maximalen und minimalen Position

```
[9]: positions_above_half = np.argwhere(trace_n > half_intensity)
     fwhm_pos_min = positions_above_half[0][0]
     fwhm_pos_max = positions_above_half[-1][0]
     fwhm_pos_min, fwhm_pos_max
     fwhm_wave_min = p(fwhm_pos_min)
     fwhm_wave_max = p(fwhm_pos_max)
```

```
width = fwhm_wave_max - fwhm_wave_min
width_nm = width/10.0
```

Darstellung des Filterprofiles mit Angabe der Halbwertsbreite

```
[10]: plt.rcParams['figure.figsize'] = (12,8)
fig, axes = plt.subplots()
plt.plot(waves,trace_n)
plt.
    ↳plot([fwhm_wave_min,fwhm_wave_max],[half_intensity,half_intensity],color='red')
plt.xlabel('Wavelength (A)')
plt.ylabel('normalized throuput')
plt.title(filter_name)
plt.xlim = [waves[0], waves[-1]]
plt.ylim = [0,1]
plt.text(fwhm_wave_max,half_intensity,f'{width_nm:6.3}nm')
plt.show()
```



```
[ ]:
```