

# The data processing pipeline for the MUSE instrument

Peter M. Weilbacher<sup>1</sup>, Ralf Palsa<sup>2</sup>, Ole Streicher<sup>1</sup>, Roland Bacon<sup>3</sup>, Tanya Urrutia<sup>1</sup>, Lutz Wisotzki<sup>1</sup>, Simon Conseil<sup>3,4</sup>, Bernd Husemann<sup>5</sup>, Aurélien Jarno<sup>3</sup>, Andreas Kelz<sup>1</sup>, Arlette Pécontal-Rousset<sup>3</sup>, Johan Richard<sup>3</sup>, Martin M. Roth<sup>1</sup>, Fernando Selman<sup>6</sup>, and Joël Vernet<sup>2</sup>

<sup>1</sup> Leibniz-Institut für Astrophysik Potsdam (AIP), An der Sternwarte 16, 14482 Potsdam, Germany  
e-mail: [pweilbacher@aip.de](mailto:pweilbacher@aip.de)

<sup>2</sup> ESO, European Southern Observatory, Karl-Schwarzschild-Str. 2, 85748 Garching bei München, Germany

<sup>3</sup> Univ. Lyon, Univ. Lyon1, Ens de Lyon, CNRS, Centre de Recherche Astrophysique de Lyon UMR5574, 69230 Saint-Genis-Laval, France

<sup>4</sup> Gemini Observatory/NSF's OIR Lab, Casilla 603, La Serena, Chile

<sup>5</sup> Max-Planck-Institut für Astronomie, Königstuhl 17, 69117 Heidelberg, Germany

<sup>6</sup> European Southern Observatory, Ave. Alonso de Córdova 3107, Vitacura, Santiago, Chile

Received 29 February 2020 / Accepted 13 June 2020

## ABSTRACT

The processing of raw data from modern astronomical instruments is often carried out nowadays using dedicated software, known as pipelines, largely run in automated operation. In this paper we describe the data reduction pipeline of the Multi Unit Spectroscopic Explorer (MUSE) integral field spectrograph operated at the ESO Paranal Observatory. This spectrograph is a complex machine: it records data of 1152 separate spatial elements on detectors in its 24 integral field units. Efficiently handling such data requires sophisticated software with a high degree of automation and parallelization. We describe the algorithms of all processing steps that operate on calibrations and science data in detail, and explain how the raw science data is transformed into calibrated datacubes. We finally check the quality of selected procedures and output data products, and demonstrate that the pipeline provides datacubes ready for scientific analysis.

**Key words.** instrumentation: spectrographs – techniques: imaging spectroscopy – methods: observational – methods: data analysis

## 1. Introduction

The Multi Unit Spectroscopic Explorer (MUSE; [Bacon et al. 2010, 2014](#)) is a large-field, medium-resolution integral field spectrograph operating at the European Southern Observatory (ESO) Very Large Telescope (VLT) since October 2014.

*Historical background.* MUSE was developed as one of the second-generation instruments at the ESO Paranal Observatory. At its inception the landscape of optical integral field spectrographs was led by a few instruments at 4 m class telescopes, like SAURON ([Bacon et al. 2001](#)) and PMAS ([Roth et al. 2005](#)), as well as fiber-based units like VIMOS ([Le Fèvre et al. 2003](#)) and GMOS ([Allington-Smith et al. 2002](#)) on 8 m telescopes. While the Euro3D network ([Walsh 2004](#)) was created to develop software for such instruments ([Roth 2006](#)), data reduction remained cumbersome (e.g., [Monreal-Ibero et al. 2005](#)). Depending on the observatory and instrument in question, only basic procedures were widely available. While there were exceptions ([Wisotzki et al. 2003](#); [Zanichelli et al. 2005](#), among others), researchers often struggled to subtract the sky and to combine multiple exposures, and many homegrown procedures were developed to do that and produce datacubes ready for analysis. The results were frequently suboptimal (e.g., [van Breukelen et al. 2005](#)).

In this environment, the specifications for MUSE were developed by ESO based mostly on the experience with the FLAMES/GIRAFFE spectrograph ([Pasquini et al. 2002](#)). Of the performance requirements, six were relevant for the develop-

ment of the pipeline. They included (i) the capability of reconstructing images with a precision of better than 1/4 pixel, (ii) a flux calibration accurate to  $\pm 20\%$ , (iii) the ability to support offsets, (iv) sky subtraction to better than 5% sky intensity outside the emission lines, with a goal of 2%, (v) the capability of combining up to 20 dithered exposures with a signal-to-noise ratio (S/N) of at least 90% of the theoretical value, and (vi) a wavelength calibration accuracy of better than 1/20th of a resolution element. Overall, the goal was to deliver software that would generate datacubes ready for scientific use, with the best possible S/N to detect faint sources, with only minimal user interaction. More details of the history of 3D spectroscopic instruments, their properties, and the development of MUSE can be found in [Bacon & Monnet \(2017\)](#).

*Instrument properties.* In its wide-field mode (WFM), the instrument samples the sky at approximately  $0''.2 \times 0''.2$  spatial elements and in wavelength bins of about  $1.25 \text{ \AA pixel}^{-1}$  at a spectral resolution of  $R \sim 3000$  over a  $1' \times 1'$  field of view (FOV) with a wavelength coverage of at least 4800–9300 Å (nominal) and 4650–9300 Å (extended mode). Since 2017, MUSE can operate with adaptive optics (AO) support ([Ströbele et al. 2012](#)). In WFM it is operated in a seeing-enhancing mode, which only corrects the ground layer (see [Kamann et al. 2018a](#), for a first science result), the pixel scale and wavelength range stay the same.

A high-order laser tomography AO correction ([Oberti et al. 2018](#)) has been available in the narrow-field mode (NFM)

since 2018 (the first science publications were Knapen et al. 2019; Irwin et al. 2019). In this mode the scale is changed to  $25 \text{ mas pixel}^{-1}$  to better sample the AO-corrected point-spread function (PSF). The resulting field is then  $8\times$  smaller (about  $7''.5 \times 7''.5$ ). The wavelength range is the same as in nominal mode. In all these cases, the data is recorded on a fixed-format array of 24 charge-coupled devices (CCDs), each of which is read out to deliver raw images of  $4224 \times 4240$  pixels in size. We summarize the instrument modes in Table A.1 and show a sketch of its operation in Fig. B.1.

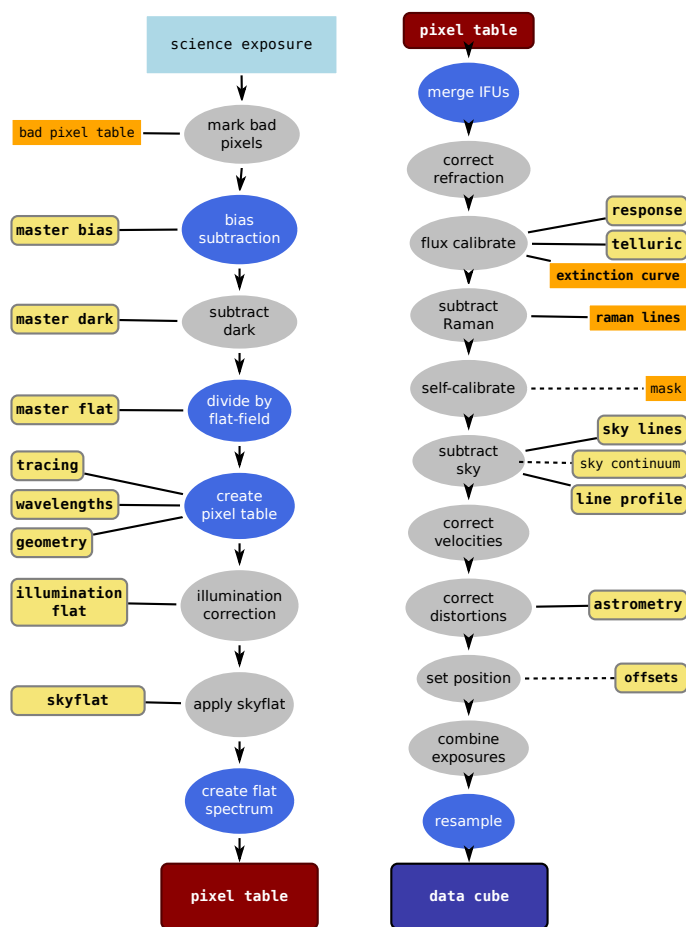
For an instrument with such complexity, combined with the size of the raw data (about 800 MB uncompressed), a dedicated processing environment is a necessity. This pipeline was therefore planned early on during the instrument development to be an essential part of MUSE. While some design choices of the instrument were mainly driven by high-redshift Universe science cases, many other observations were already envisioned before the start of MUSE observations. By now MUSE actually evolved into a general purpose instrument, as documented by recent publications on the topics of (exo-) planets (Irwin et al. 2018; Haffert et al. 2019), Galactic targets (Weilbacher et al. 2015; McLeod et al. 2015), and resolved stellar populations (Husser et al. 2016; Kamann et al. 2018b), via nearby galaxies (Krajnović et al. 2015; Monreal-Ibero et al. 2015) and galaxy clusters (Richard et al. 2015; Patrício et al. 2018) to high-redshift Lyman- $\alpha$  emitters (Bacon et al. 2015; Wisotzki et al. 2016, 2018), to name only a few science applications. In contrast to some other instruments with a dominating scientific application (see Strassmeier et al. 2015; Scott et al. 2018), the MUSE pipeline thus cannot be concerned with astrophysical analysis tasks. Its role is confined to the transformation from the raw CCD-based data to fully calibrated datacubes. Depending on the science case in question, other tools were then created to handle the datacubes (MUSE Python Data Analysis Framework, MPDAF, Bacon et al. 2016; Piqueras et al. 2019), to extract stellar (PampelMUSE, Kamann et al. 2013) or galaxy (TDOSE, Schmidt et al. 2019) spectra, or to detect emission lines sources (LSDCat, Herenz & Wisotzki 2017), among others. Describing these external tools is not the purpose of this paper.

*Paper structure.* The data processing of MUSE at different stages of implementation was previously described in Weilbacher et al. (2009, 2012, 2014) and Streicher et al. (2011). These papers still reflect much of the final pipeline software, and explain some of the design choices in more detail. The present paper aims to first describe the science processing steps on a high level (Sect. 2) to let the user get an idea of the steps involved in the above-mentioned transformation. Afterwards, in Sect. 3 we give a detailed description of all the steps involved in generating master calibrations and science products. Some algorithms that are used in multiple steps are then presented in Sect. 4, while Sect. 5 briefly discusses key parameters of the implementation. Section 6 investigates the data quality delivered by the pipeline processing. We conclude with a brief outlook in Sect. 7.

This paper is based on MUSE pipeline version 2.8.3 as publicly released in June 2020<sup>1</sup>. Version 2.8 was the first version to support all modes of the instrument, in particular the NFM<sup>2</sup>.

<sup>1</sup> <https://www.eso.org/sci/software/pipelines/muse/muse-pipe-recipes.html>

<sup>2</sup> Milestones of earlier versions were v1.0 in Dec. 2014 to support the first observing runs with only seeing-limited WFM, and v2.2 which first supported WFM AO data in Oct. 2017.



**Fig. 1.** *Left:* basic processing from raw science data to the intermediate pixel table. *Right:* post-processing from pixel table to the final datacube. Optional steps are in gray, mandatory ones in blue. Manually created input files have an orange background; calibrations are highlighted in yellow. Inputs that are needed are connected with a solid line; dotted lines signify inputs that are not required.

Where applicable, we note in which version a new feature was introduced.

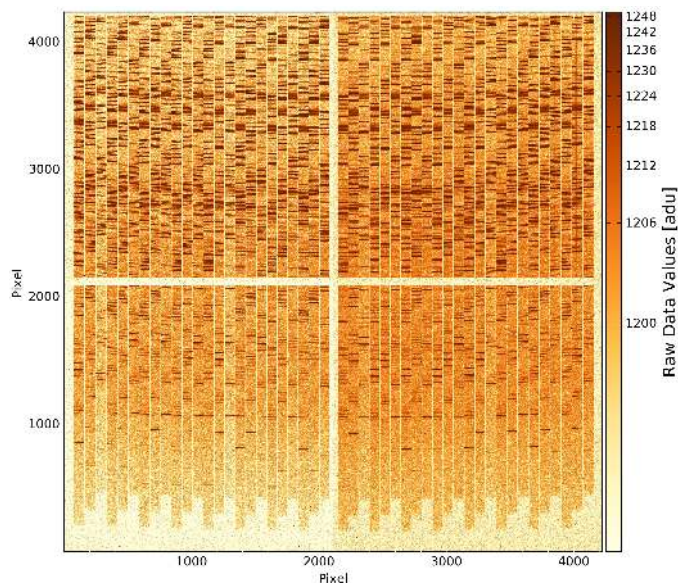
## 2. Science processing overview

The main processing steps to calibrate the data and transform it from the image-based format of the raw data via a table-based intermediate format during processing to the output cube are visualized in Fig. 1. The computations are split into two parts, the basic processing, which calibrates and corrects data on the basis of single CCDs, and the post-processing, which carries out the on-sky calibrations and construction of the final datacube. The intermediate data, the “pixel tables”, are the files that connect the two processing levels.

In this section we only briefly mention the processing steps, and point to later sections where they are described in more detail. In a few cases a processing step is not connected to a calibration file, and hence is not described further elsewhere. Then we describe this step here in greater depth.

### 2.1. Raw data

The MUSE raw data comes in multi-extension Flexible Image Transport System (FITS) files, where each of the 24 CCD



**Fig. 2.** Raw science data of one of the 24 CCDs, displayed in negative arcsinh scaling. This shows the data of IFU 10 of the exposure started at 2014-06-26T01:24:23 (during MUSE Science Verification). The 48 slices of the IFU are the stripes oriented almost vertically, which appear dark in this representation. The blue end of the MUSE wavelength range is located at the bottom, the red limit near the top; the step-pattern is created by the geometry of the image slicer. Since this was a 600 s exposure, the sky emission and continuum dominate over the relatively faint object signal in this part of the cube. The overscan regions of the CCDs are creating the cross in the center of the image; the pre-scan regions are the empty borders. This exposure was taken in nominal mode (WFM-NOAO-N), the second-order blocking filter removed the blue light so that the bottom part of the image appears empty.

images is stored in one extension. Each of the images is  $4224 \times 4240$  pixels in size, and stored as unsigned integers of 16 bit. The CCD is read out on four ports, so that the CCD has four regions of equal size, called “quadrants”. These quadrants have a data section of  $2048 \times 2056$  pixels, and pre- and overscan regions of 32 pixels in width. The images are accessible in the FITS files via extension names, formed of the IFU number prefixed by CHAN, for example, CHAN01 for the first IFU. A typical raw science image of CHAN10 is displayed in Fig. 2.

Several additional FITS extensions may be present for on-sky data, depending on the instrument mode used for an exposure. These are concerned with ambient conditions, performance of the auto-guiding system of the VLT, the slow-guiding system of the MUSE instrument, and the atmospheric turbulence parameters used by the AO system. These extra extensions are not used by the MUSE pipeline.

## 2.2. Basic science processing

The first step of the science processing is done with the module `muse_scibasic`. The inputs to this recipe are one or more raw science exposures, optionally one corresponding illumination flat-field exposure, and a number of calibration files. Outputs are pre-reduced pixel tables.

Processing is internally performed on each individual CCD, so that all of the following is done 24 times. The raw CCD image is loaded (DATA image) from the raw data (the corresponding CHANnn extension), and two images of the same size are added, one for the data quality (DQ, see Sect. 4.2), one for the variance

(STAT, see Sect. 4.1). Both new images are empty at the start. Next, if the optional bad pixel table was given, these pixels are marked in the DQ image. In any case, saturated pixels are detected and marked, if the data allow such a determination to be made (raw values of zero or above 65 500).

Next, the overscan regions of the MUSE CCDs are analyzed to determine the corrective factor and slopes to apply to the bias (see Sect. 4.3 for more details about this step) before the master bias image of the corresponding CCD is subtracted. Then the CCD gain is used to transform the input data from analog-to-digital units (adu) to electrons (internally called count). The gain value is taken from the FITS headers of the raw exposure. If the optional master dark was given, the pipeline subtracts the dark current image given by that calibration file. An optional step in the CCD-level calibration is to detect and mark cosmic rays (using the DCR algorithm, Pych 2004). However, this is usually not necessary at this stage (as explained in detail in Sect. 4.5.1). The science image is then divided by the master lamp flat-field image provided to the processing routine. The last step in the direct CCD-level processing propagates the relative IFU flux level from the twilight sky cube, if this optional calibration was given as input calibration file.

The mandatory input calibrations, trace table, wavelength calibration table, and geometry table (for their content and purpose, see Sects. 3.4–3.6) are used to assign coordinates – two spatial components in pseudo pixel units relative to the MUSE FOV, and one wavelength component – to each CCD-based pixel. Thereby a pixel table is created for each input science (on-sky) exposure.

All these steps are also applied in the same way to the optional raw illumination flat-field exposure if one was supplied<sup>3</sup>. The following steps are exclusively applied to exposures taken on-sky at night.

The wavelength zero point is corrected using sky emission lines, if applicable (see Sect. 3.5.2). Afterwards, the pixel table is usually cropped to the useful wavelength range, depending on the mode used for the observations. The useful wavelength is defined by the range for which the MUSE FOV is fully sampled. It extends from 4750–9350 Å for the nominal and 4600–9350 Å for the extended mode<sup>4</sup>.

If the optional raw illumination flat-field exposure was given as input, it is then used to correct the relative illumination between all slices of one IFU. For this the data of each slice is multiplied by the normalized median flux (over the wavelength range 6500–7500 Å to use the highest S/N data in the middle of the wavelength range) of that slice in that special flat-field exposure. Since the illumination of the image slicer changes with time and temperature during the night, this correction removes these achromatic variations of the illumination and thus significantly improves the flux uniformity across the field.

The last step in the basic science processing interpolates the master twilight sky cube (if it was given as input; see Sect. 3.7) to the coordinate of each pixel in the pixel table. Spatially, the nearest neighbor is taken; in wavelength, a linear interpolation between adjacent planes is carried out. The data values in the pixel table are then divided by the interpolated twilight sky correction.

<sup>3</sup> This illumination flat-field is a lamp-flat exposure taken by the observatory at least once per hour, or if the ambient temperature changes significantly.

<sup>4</sup> The exact ranges are slightly different for the AO modes, see Table A.1. For these modes the region affected by the NaD narrow-band blocking filter is also specially marked at this stage.



At this stage the pre-reduced pixel table for each on-sky exposure is saved to the disk in separate files for each IFU, including the corresponding averaged lamp-flat spectrum in one of the file extensions.

The module discussed in this section, `muse_scibasic`, is also used to process other non-science on-sky exposures taken for calibration purposes. Specifically, standard star fields, sky fields, and astrometric exposures of globular clusters are handled by `muse_scibasic`, but then are further processed by specialized routines.

### 2.3. Science post-processing

The input to the post-processing are the pre-reduced pixel tables; the main output is the fully reduced datacube. The first step is to merge the pixel tables from all IFUs into a common table. This step has to take into account the relative efficiency of each IFU as measured from twilight sky exposures. It is applied as a scaling factor relative to the first channel. When merging the (science) data, all lamp-flat spectra of the IFUs are averaged as well. Since removing the large-scale flat-field spectrum from the (science) data is desirable, without reintroducing the small-scale variations corrected for by flat-fielding, this mean lamp-flat spectrum is smoothed over scales larger than any small-scale features, such as telluric absorption or interference filter fringes. The on-sky data is then divided by this spectrum<sup>5</sup>.

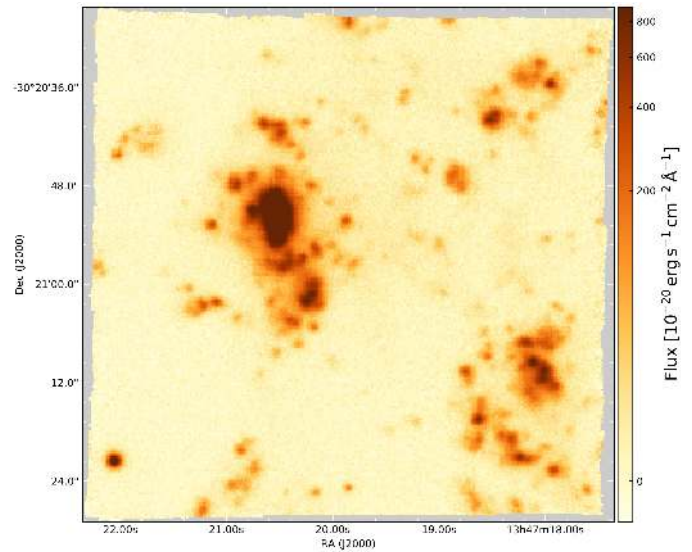
Then the merged pixel table is put through several corrections. The atmospheric refraction is corrected (for WFM data, the NFM uses an optical corrector) relative to a reference wavelength (see Sect. 4.8). When a response curve is available, the flux calibration is carried out next. It converts the pixel table data (and variance) columns into flux units. This uses an atmospheric extinction curve that has to be passed as input table. If a telluric correction spectrum was provided, this is applied as well (Sect. 4.9).

For exposures taken with AO in WFM, a correction of atmospheric emission lines caused by Raman scattering of the laser light can be carried out (see Sect. 3.10.1). A per-slice self-calibration can be run next to improve background uniformity across the FOV of MUSE (for details see Sect. 3.10.2).

Typically, sky subtraction is carried out next. This step has multiple ways of deriving the sky contribution, which also depends on the user input and the type of field being processed (sky subtraction is not needed for all science cases and can be skipped). In the case of a filled science field, an offset sky field has to be used to characterize the sky background (see Sects. 3.9.2 and 3.9.3); sky lines and continuum are then needed as inputs. The procedure is the same for a largely empty science field, just that the sky spectrum decomposition does not need extra inputs. Since the sky lines change on short timescales, they usually have to be re-fitted using a spectrum created from a region of the science exposure devoid of objects. This is the default behavior, but it is possible to skip the refit. The continuum, however, only changes slowly and is subtracted directly. In all cases, the user usually has to tell the pipeline which spatial fraction of an exposure is sky-dominated so that the software can use that portion of the data to reconstruct the sky spectrum.

The science data is then corrected for the motion of the telescope. This radial velocity correction is done by default relative to the barycenter of the solar system, but for special purposes heliocentric and geocentric corrections are available. Algorithms from G. Torres (bcv) and the FLAMES pipeline (Mulas et al.

<sup>5</sup> The correction for the lamp flat-field spectrum has been done since v2.0 of the pipeline; smoothing has been applied since v2.6.



**Fig. 3.** Reduced science data. A combination of three science exposures taken on 2014-06-26 between 1:00 and 2:00 UTC, including the image displayed in Fig. 2. This image shows a cut of the datacube at the wavelength of H $\alpha$  (redshifted to 6653.6 Å) displayed in negative arcsinh scaling. Regions at the edge that were not covered by the MUSE data are displayed in light gray.

2002) and transformations from Simon et al. (1994) are used to compute the values.

The spatial coordinate correction is applied in two steps and makes use of the astrometric calibration (Sect. 3.12). First, linear transformation and rotation are applied and the spherical projection is carried out. This transforms the pixel table spatial coordinates into native spherical coordinates, following Calabretta & Greisen (2002). The second step is the spherical coordinate rotation onto the celestial coordinates of the observed center of the field. This step can be influenced to improve both absolute and relative positioning of the exposure by supplying coordinate offsets. Such offsets can be computed manually by the user or automatically by correlating object detections in overlapping exposures (Sect. 3.13).

Once all these corrections and calibrations are applied on the pixel table level, the data are ready to be combined over multiple exposures. This can be a very simple concatenation of the individual pixel tables or involve relative weighting. By default, only linear exposure time weighting is carried out, such that exposures that are twice as long are weighted twice as strongly. Another possibility is seeing-weighted exposure combination, which is implemented to primarily use the FWHM measured by the VLT auto-guiding system during each (non-AO) exposure. More complex weighting schemes are possible, but require the users to determine the weights themselves, depending on exposure content and science topic.

Finally, the data of one or more exposures are resampled into a datacube. The process is described in detail in Sect. 4.5, by default it uses a drizzle-like algorithm (Fruchter et al. 2009) to conserve the flux (e.g., of emission lines over a spatially finite object). This cube is normally computed such that north is up and east is left, and the blue end of the wavelength range is in the first plane of the cube, and so that all data contained in the pixel table is encompassed by the grid of the output cube. The example in Fig. 3 shows a single wavelength plane from a reduced cube (this data of NGC 5291 N was published by Fensch et al. 2016). Both the extent of the cube and the sampling of the output data can be

adjusted by the user for a given science project. A logarithmic wavelength axis can be defined as well, and the user can choose to have it saved in vacuum instead of air wavelengths. The cube is the basis for the computation of reconstructed images of the FOV (see Sect. 4.7), which are integrated over a built-in “white” or any other filter function.

These post-processing routines for the science data are offered in the pipeline by `muse_scipost`, the combination of multiple exposures is also available separately as `muse_exp_combine`. The module `muse_exp_align` (Sect. 3.13) can be used to compute the offset corrections, as mentioned above, between multiple (overlapping) exposures in an automated fashion.

### 3. Full processing details

Calibration of an instrument usually includes creation of master calibration data which is then applied to subsequent calibration steps and the science data itself. This is no different for MUSE, where the usual calibration exposures are done during daytime, with the help of the calibration unit (Kelz et al. 2010, 2012). They are supplemented by on-sky calibrations done with the telescope during twilight and during the course of the night. The full details of how these specific calibrations are processed and then applied are provided in this section. The purpose and frequency of the different calibrations are further described in the ESO MUSE User Manual (Richard et al. 2019a). At the end of each subsection we also note within which pipeline module the described steps are implemented, and how the significant parameters are named.

#### 3.1. Bias level determination

The first step for removing the instrumental pattern from CCD exposures is always to subtract the bias level. To this end, daily sequences of typically 11 bias frames with zero exposure time and closed shutter are recorded. In the case of MUSE the CCDs are read out in four quadrants, and so the bias level already exhibits four different values in the middle of the read-out image. In addition to this quadrant pattern, the bias shows horizontal and vertical gradients so that the bias images have to be subtracted from the science and calibration data as 2D images. Finally, a variation in the bias level with time means that before subtraction, the bias needs to be offset to the actual bias determined from the other exposure, using values from the overscan (Sect. 4.3).

The bias images are also used to measure the read-out noise (RON) of each CCD. This is computed on difference images ( $B_1 - B_2$ ) of one whole bias sequence. On each difference image, 400 boxes (100 in each CCD quadrant) of  $9 \times 9$  pixels are distributed, within which the standard deviation of the pixel values is recorded. The median of these standard deviations is taken as the  $\sigma_{B_1-B_2}$  value for each difference image; the average of all these values is  $\overline{\sigma_{B_1-B_2}}$  for each CCD quadrant. To estimate the error the standard deviation of the standard deviations of all boxes is taken. If the error on  $\overline{\sigma_{B_1-B_2}}$  is found to be larger than 10% of  $\overline{\sigma_{B_1-B_2}}$ , the procedure is repeated, with a differently scattered set of boxes. The value of  $\overline{\sigma_{B_1-B_2}}/\sqrt{2}$  is then used as initial variance of the individual bias images (see Sect. 4.1). To compute a meaningful final read-out noise value and its error, the CCD gain<sup>6</sup>  $g$  is used to convert it to units of electrons (see Howell 2006):

$$\text{RON} = \frac{g \overline{\sigma_{B_1-B_2}}}{\sqrt{2}}.$$

The only other processing involved in creating the master bias image is to combine the individual images using the algorithm described in Sect. 4.4. By default, a  $3\sigma$  clipped average is computed.

Some CCD defects show up as columns of different values already on bias images. To find them, column statistics of median and average absolute deviation are used to set thresholds above the typical column level on the master bias image. Normally, a  $3\sigma$  level is used to find bright columns. High pixels within these columns get flagged in the master image. Since finding dark pixels is not possible on bias images, flagging of low-valued pixels is set to  $30\sigma$  so that this does not happen.

Application of the master bias image to a higher level exposure involves the overscan-correction described in Sect. 4.3, after checking that the same overscan handling was used for both. This is then followed by the subtraction of the master bias image which includes propagation of any flagged pixels to the resulting image.

The routine producing the master bias is available as `muse_bias` in the pipeline.

#### 3.2. Dark current

Estimating the dark current, the electronic current that depends on exposure time, is a typical aspect of the characterization of CCDs, which means this procedure is available in the MUSE pipeline as well. Darks, which are long exposures with the shutter remaining closed and external light sources switched off, can also be used to search for hot pixels and bright columns on the resulting images. Darks for MUSE are usually recorded as sequences of five frames of 30 min once a month.

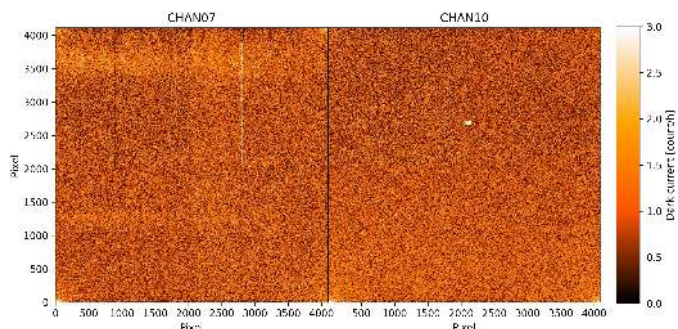
The processing of dark exposures is performed as follows: from each of a sequence of darks the bias is subtracted using the procedures outlined in Sects. 4.3 and 3.1. All exposures are then converted to units of electrons, scaled to the exposure time of the first one, and combined using the routines described in Sect. 4.4, by default using the  $\pm 3\sigma$  clipped average. If enough exposures were taken, these output images are free of cosmic rays. The resulting images, one for each CCD, are then normalized to a one-hour exposure, so that the master dark images are in units of  $e^- \text{h}^{-1} \text{pixel}^{-1}$ . Hot pixels are then located using statistics of the absolute median deviation above the median of each of the four data sections on the CCD. Typically, a  $5\sigma$  limit is used. These hot pixels are marked in the DQ extension to be propagated to all following steps that use the master dark. The master dark image that was thus created is shown in Fig. 4 for two example IFUs.

As an optional last step, a smooth model of the dark can be created<sup>7</sup>. Since the MUSE CCDs have very low dark current (typically measured to be about  $1 e^- \text{h}^{-1} \text{pixel}^{-1}$ ) averaged across their data regions, a model is necessary to subtract the dark current from science exposures at the pixel level to avoid adding additional noise. The model consists of several steps. Some CCDs show evidence of low-level light leaks. These can be modeled first, over horizontal regions 280–340 pixels high, using 2D polynomials of order 5 in both directions. After subtracting them, it is possible to refine the detection of bad pixels. To represent the large-scale dark current, a bilinear polynomial is used. The fit for this ignores 500 pixels at the edge of the CCD, so that uneven borders and the corners heated by the read-out

<sup>6</sup> The correct gain value has to be set up in the FITS header of the raw images in order for this to happen in practice.

<sup>7</sup> This feature has been available since v2.8.





**Fig. 4.** Master dark images for two MUSE CCDs, both combined from 149 raw dark images taken between 2018-06-25T11:20:47 and 2018-08-21T06:56:51. The CCD of channel 7 shows broad horizontal stripes, while the CCD of channel 10 shows a noticeable block of hot pixels. The location of the hot corners is different for both CCDs, and while channel 10 shows a vertical gradient seen in most MUSE CCDs, this is less apparent for the CCD in channel 7.

ports do not contribute. Finally, these read-out ports are modeled separately in a region of 750 pixel radius around each of the four corners. Here, a fifth-order polynomial is used again. To make the fit stable against small noise fluctuations, a 100 pixel annulus is used to tie the corner model to the global fit. The sum of all three polynomial models is the final dark model. Searching for bad pixels can then be done again, using statistics of the differences between the master dark and this smooth model.

If neither the master dark nor the dark model are used, one can transfer the hot pixels found by this procedure to separate bad pixel tables (see Sect. 4.2) instead.

The routine producing the master dark is available as `muse_dark` in the pipeline. The sigma-clipping level to search for hot pixels is adjustable by the parameter `hotsigma`, the smooth modeling activated with `model`.

### 3.3. Flat-fielding

Flat-field correction in the MUSE pipeline has the same purpose as in classical CCD imaging: to correct pixel-to-pixel sensitivity variations, and to locate dark pixels. The process is simple. From each of a sequence of exposures taken with the continuum lamps switched on, the bias is subtracted using the procedures outlined in Sects. 4.3 and 3.1. A dark image can optionally be subtracted; this is typically not done since the exposure times are short. The main purpose of subtracting a dark here would be to propagate its hot pixels map. The data is then converted from units of adu to electrons, using the gain value provided with the raw data. If different exposure times are used, the images are then scaled relative to the first one. Then all images are combined, using one of the algorithms described in Sect. 4.4, using a  $3\sigma$  level clipped mean by default. The final master flat image is then normalized to 1 over the whole data section of the CCD.

Once the slice tracing (Sect. 3.4) is done (in the implementation it is done in the same pipeline module as the flat-fielding) and the slice edges on the CCD are known, the master flat-field can also be used to search for dark pixels. This is done for each CCD row, in the region between the edges of each slice. The dark pixels are determined as outliers in the sigma-clipped statistics of normally  $5\times$  the absolute median deviation below the median. This effectively marks all dark columns. Since for MUSE CCDs some of the electrons lost in the dark columns appear as bright pixels on their borders, we also flag pixels  $5\sigma$  above the median. Both limits can be adjusted.

In the case of flat-field exposures in nominal mode, the blue (lower) end of each CCD image contains many pixels that are not significantly illuminated. Due to noise, some of these pixels are below the bias level and hence are negative in the master flat-field image. These pixels are flagged as well, to exclude them from further processing in case the science data is not automatically clipped at the blue end (the default).

Application of the master flat-field correction to any higher level exposure simply involves division by the master-flat image of the respective CCD. Pixel flags are thereby propagated from the master flat image to the other image, the pixel variances are propagated as well.

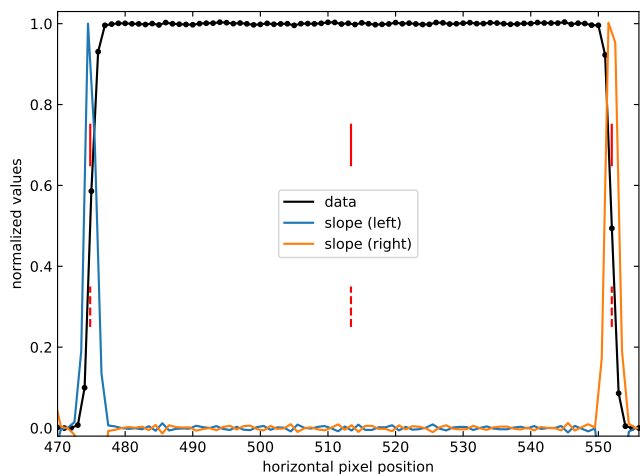
The routine producing the master flat-field is available as `muse_flat` in the pipeline. The bad pixel detection threshold is set with the parameters `losigmabadpix` and `hisigmabadpix`.

### 3.4. Slice tracing

The tracing algorithm is the part of the MUSE pipeline that determines the location of those areas on the CCDs that are illuminated by the light from the 48 slices of each image slicer in each IFU.

This step uses the flat-field exposures to create well-illuminated images for most of the wavelength range. They are prepared into a master-flat image for each CCD, as described in Sect. 3.3. Tracing starts by extracting an averaged horizontal cut of the  $\pm 15$  CCD rows around the center of the data area. By averaging the data from upper and lower quadrants, the influence of a possible dark column in one of the quadrants is reduced. Pixels that fall below a fraction of 0.5 (by default) of the median value of this cut determine the first-guess edges of each slice. The first slice (on the left side of the CCD image) is the most critical one. If it is not detected within the first 100 pixels of the cut, or if it is narrower than 72.2 or wider than 82.2 pixels, then the detection process is stopped and the tracing fails, otherwise the rest of the slices are detected the same way. If more or less than 48 slices were detected this way, or if some of the slices were too narrow, the process is iteratively repeated with edge levels that are  $1.2\times$  smaller until a proper number of slices is found. The horizontal centers (the mean position of both edges) of these slices are subsequently used to trace the curvature and tilt of all slices on the image. To this end, horizontal cuts are placed along a sufficient number of rows along the whole vertical extent of every slice. The number of cuts is determined by the number of rows  $n_{\text{sum}}$  that are averaged with each such trace point. By default,  $n_{\text{sum}} = 32$ , so there are 128 trace points over the vertical extent of each slice. Starting at the center of each cut, which is at the first-guess position of the center of the slice, pixel values are compared to the median value across the slice in both directions. If a pixel falls below a given fraction (typically 0.5, as above), the slice edge is found and linearly interpolated to a fractional pixel position of the given limit. This determines two initial slice edge positions (left and right) and an implicit slice center (the average of the two). Since the slice edge can be robustly defined by the position where the flux falls to 50% of the flux inside the slice, both edge positions are refined. This is done using the slope of all pixel values along the cut that contains a peak at the 50% position (see Fig. 5). This peak is then fitted with a Gaussian to give a very robust edge position, which is more accurate than a linearly interpolated fractional edge<sup>8</sup>.

<sup>8</sup> An exception are the slices in the bottom-right corner in the MUSE FOV, where the field is affected by unfocused vignetting. The relevant slices are numbers 37 to 48 on the CCD in IFU 24, where the Gaussian edge refinement is switched off for the affected edge.



**Fig. 5.** Tracing procedure and edge refinement. Shown is a cut through a MUSE slice at the CCD level, illuminated by a flat-field lamp. The data itself is displayed normalized to the constant region in between the edges (black). The slope of the data for the left edge (blue) and the right edge (orange) are displayed as well. The original trace position (vertical dashed red lines) and the refined edges (solid red lines) are also shown. In this case the refinement shifted the positions by less than 0.1 pixels.

After determining the edges at all vertical positions, those with unlikely results are filtered, using the range of expected slice widths (again, 72.2 to 82.2 pixels) as the criterion. This effectively removes CCD columns where the illuminated area of a slice is strongly affected by a dark column or some other artifact. Then a polynomial (typically of fifth order) is iteratively fitted to the remaining good trace points, using a  $5\sigma$  limit for rejection. The final tracing information then includes three polynomials for each slice, marking the slice center and its left and right edges on the CCD.

In the pipeline, this routine for computing the master trace table is part of the `muse_flat` module, for efficiency reasons, and is executed if the `trace` parameter is set to true. Parameters to change the edge detection fraction (`edgefrac`), the number of lines over which to average vertically (`nsum`), and the polynomial order of the fitted solution (`order`) can be adjusted.

### 3.5. Wavelength calibration

The wavelength calibration is essential for a spectrographic instrument. In the MUSE pipeline, a dedicated module computes the wavelength solution for every slice on the CCD. This solution is a 2D polynomial, with a horizontal order (2 by default) describing the curvature of the arc lines in each slice on the CCD, and a vertical order (6 by default) describing the dispersion relation with wavelength<sup>9</sup>.

Because the MUSE spectrographs do not operate in vacuum the wavelength calibration is based on arc line wavelengths in standard air. However, if convenient for the scientific analysis, the final cube can be constructed by the pipeline at vacuum wavelengths at a later stage.

#### 3.5.1. Computing the wavelength solution

This module is expected to process a series of exposures in which a number of images exist for each of the three arc lamps

<sup>9</sup> This is similar to the FITCOORDS task in the LONGSLIT package of the IRAF environment.

built into MUSE (HgCd, Ne, and Xe). The use of different lamps ensures a reasonable coverage of the wavelength range of the instrument. Typically five exposures per lamp are used to maximize the S/N for fainter lines without saturating the brightest ones. All images of the sequence are bias subtracted as discussed before. Optionally, they can be dark corrected and flat-fielded as well, but these calibrations are typically not used for the wavelength calibration. The units are then converted from adu to electron, normally using the gain value given in the raw data. Contrary to other modules, the exposures are not all combined. Instead they are sorted into the sub-sequences of exposures illuminated by each of the three arc lamps which are then combined, so that the following analysis is done separately on three images. This “lamp-wise” handling has the advantage that the images contain fewer blends of emission lines, and so the lines are easier to identify and measure. The combination method used by default is the  $3\sigma$  clipped average (see Sect. 4.4).

The actual analysis works separately for each slice. From the reference list of arc lines, only the lines for the relevant lamp are selected. To detect the corresponding lines on the real exposure an S/N spectrum is created from the central CCD column of each slice by dividing the DATA image by the square root of the STAT image. This lets the detection work equally well whether or not the arc exposures were flat-fielded. After subtracting a median-smoothed version to remove any constant background, lines are detected using  $1\sigma$  peaks (by default) in terms of mean of the absolute median deviation above the residual median of the full spectrum. The initial line center in CCD pixels is then determined using Gaussian fits to each detection. Artifacts that are not arc lines in these detections are filtered out by rejecting single-pixel peaks and those with FWHM outside the range 1.0–5.0 pixel, a flux below  $50 e^-$ , and with an initial centering error  $>1.25$  pixel. The detections then need to be associated with known arc lines. This is done using an algorithm based on 1D pattern matching (Izzo et al. 2008, 2016). This only assumes that the known lines are part of the detected peaks and that the dispersion is locally linear inside a range of  $1.19$ – $1.31 \text{ \AA pixel}^{-1}$ . A tolerance of 10% is assumed by default when associating distances in the peaks with distances in the known arc lines. For WFM data, this process typically detects between 100 and 150 peaks which are then associated with 90 to 120 known arc lines, all arc lamps taken together. For NFM, where the arc lines do not reach the same illumination level due to the  $64\times$  lower surface brightness levels, 85–110 detections turn into 70–100 identified lines. The analysis on each of the per-lamp images continues by fitting each of the identified lines with a 1D Gaussian in each CCD column, over the full width of the slice as determined from the trace table, to determine the line center. To reject deviant values among these fits, which might be due to hot pixels or dark columns (not all of which are detectable on previous calibration exposures), this near-horizontal sequence of the center of a line is iteratively fit with a 1D polynomial of the horizontal order. The iteration by default uses a  $2.5\sigma$  rejection. After all the individual lines and the multiplets of all arc lamps have been treated in this way, the final solution is computed by an iterative 2D polynomial fit to all measured arc line centers and their respective reference arc wavelengths. This iteration uses a  $3\sigma$  clipping level. This fit is typically weighted by the errors on all line centroids added in quadrature with the scatter of each line around the previous 1D fit. The coefficients of the fit are then saved into a table.

In the pipeline the calibration routine is implemented in `muse_wavec`. The parameters for the polynomial orders are `xorder` and `yorder`. The line detection level can be tuned with the `sigma` parameter, and the pattern matching with `dres` and

tolerance. The iteration  $\sigma$  level for the individual line fits is called `linesigma`, the one for the final fit `fitsigma`, and the final fit weighting scheme can be adapted using `fitweighting`.

### 3.5.2. Applying the wavelength solution

Applying the wavelength solution simply evaluates the 2D polynomial for the slice in question at the respective CCD position. This provides high enough accuracy for other daytime calibration exposures.

When applying this to night-time (science) data, however, it becomes important to note that these data are usually taken a few hours apart from the arc exposures, and that the ambient temperature might have significantly changed in that time. This results in a non-negligible zero-point shift of the wavelength of all features on the CCDs of up to 1/10th of the MUSE spectral resolution or more.

The procedure to correct this was already briefly mentioned in Sect. 2.2. After applying the wavelength solution to the night-time data, they are stored in one pixel table for each IFU. From this table, a spectrum is created simply by averaging all pixel table values whose central wavelength fall inside a given bin. By default, the bins are 0.1 Å wide, which oversamples the MUSE spectral resolution about 25×. This results in high S/N spectra of the sky background since approximately 3600 spectra are used to find the average (all illuminated CCD columns). This requires the objects in the cube to be faint compared to the sky lines. Since this is not always the case, the spectrum is reconstructed iteratively so that pixels more than  $\pm 15\sigma$  from the reconstructed value in each wavelength bin are rejected once. Here, the sigma level is used in terms of standard deviation around the average value. For cases with very bright science sources in the field, this does not remove the sources well enough from the sky spectrum, and iterative parameters may have to be tuned to  $2\sigma$  for the upper level and ten iterations. The spectrum is only reconstructed in regions around the brightest sky lines (by default,  $\pm 5$  Å around the [O I] lines at 5577.339 and 6300.304 Å). Any shift from the tabulated central wavelengths of these sky lines (Kramida et al. 2014) in the real data is then corrected by just adding or subtracting the difference from the wavelength column of the pixel table. Because the reconstructed spectrum contains data and variance information, the fitted Gaussian centroids are computed together with error estimates. The final shift is computed as the error-weighted mean centroid offset of all lines given. Since these [O I] lines are the brightest lines in the MUSE wavelength range, and the only bright lines that are isolated enough for this purpose, they are selected by default. Only if the science data contains a similarly strong feature at the same wavelength that covers much of an IFU should the user select a different line.

The method described here is implemented in the MUSE pipeline in the `muse_scibasic` module. The parameter `skylines` gives the zero-point wavelengths of the sky emission lines to use; `skyhalfwidth` determines the extraction window for the fit; `skybinsize` tunes the binning; and `skyreject` sets the rejection parameters for the iterative spectrum resampling.

### 3.6. Geometric calibration

One central MUSE-specific calibration is to determine where within the FOV the 1152 slices of the instrument are located. This is measured in the “geometric” calibration. The “astrometric” calibration then goes one step further and aims to remove

global rotation and distortion of the whole MUSE field. The instrument layout underlying this procedure is displayed in Fig. B.1.

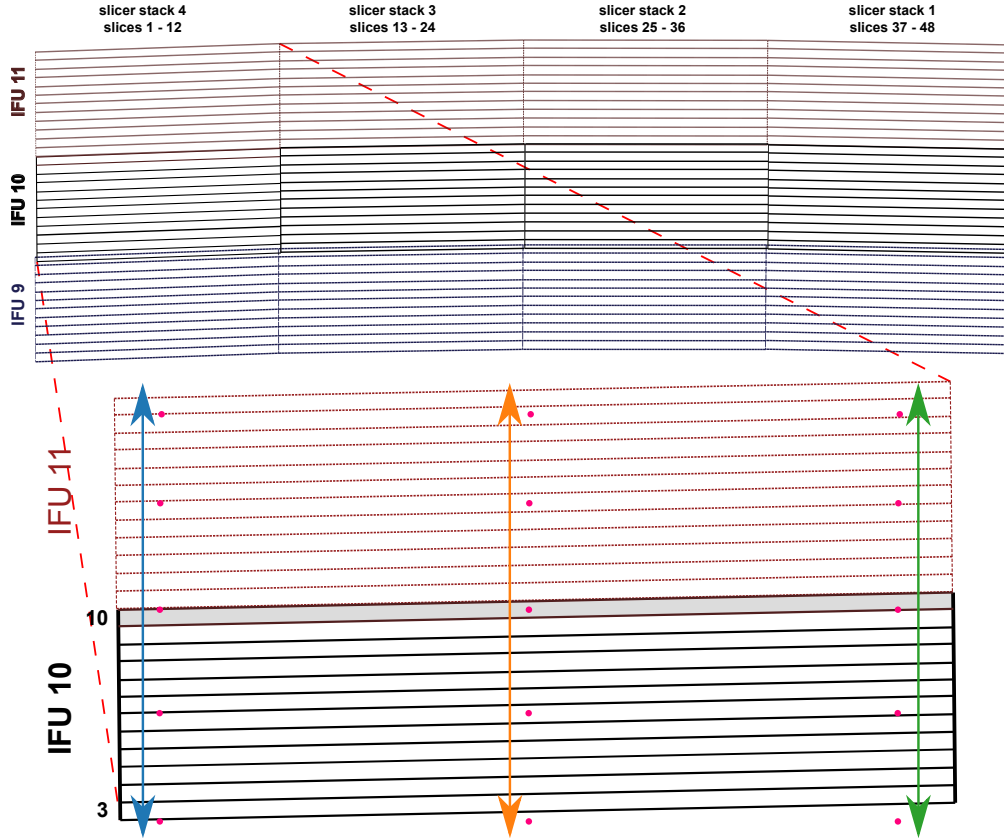
To measure this geometry, and determine the  $x$  and  $y$  positions, width, and angle for each slice, a pinhole mask is used. This mask contains 684 holes, distributed in 57 rows of 12 holes, with a horizontal distance of 2.9450 mm and a vertical distance of 0.6135 mm between adjacent holes; the covered field is therefore approximately equal to the  $35 \times 35$  mm<sup>2</sup> field that corresponds to the MUSE field in the VLT focal plane<sup>10</sup>. The mask gaps are chosen so that a given slice in its place in the image slicer is simultaneously illuminated by three pinholes, and every fifth slice is illuminated in vertical direction. A partial visualization of this setup is shown in Fig. 6. This mask is then vertically moved across the field, in 60–80 steps of 9 μm, while being illuminated by the Ne and HgCd arc lamps<sup>11</sup>. If the light from a given pinhole illuminates a slice in one MUSE IFU, the corresponding location on the CCD is illuminated as well, in the form of a bright spot whose size is dominated by the instrumental PSF. The expected position of the spot can be easily determined from the mask layout together with the trace table (Sect. 3.4) and the wavelength calibration table (Sect. 3.5). As the pinholes are moved and the pinhole disappears from the position in the MUSE field that this slice records, this spot gets less bright or completely disappears. The outcome of such a sequence is plotted in Fig. 7, which shows the illumination (flux distribution) of three pinholes observed by one slice through the sequence of 60–80 exposures. We note that slices on the edge of the FOV are only illuminated two or three times. Together with the known properties of the pinhole mask and the instrument, as well as other calibrations, the position, width, and tilt of each slice in the FOV can be determined as follows.

The processing has two stages: the first separately handles the data from all IFUs (in parallel) and the second then derives a global solution using the data from all IFUs. In the IFU-specific part, the raw data is first handled as other raw data so that it is bias-subtracted and trimmed, converted to units of electrons, and optionally dark subtracted and flat-fielded. Next, all input mask exposures of each IFU are averaged, and this combined image, together with the trace table and wavelength calibration as well as the line catalog, is used to detect the spots of the arc lines. For this purpose, the line catalog of the wavelength calibration is taken, but reduced to the 13 brightest isolated lines of Ne, Hg, and Cd in the wavelength range 5085–8378 Å. Based on tracing and wavelength calibration, rectangular windows are constructed for each slice and arc line, over the full width of the slice on the CCD and  $\pm 7$  pixels in wavelength direction. In this window a simple threshold-based source detection is run, by default using  $2.2\sigma$  in terms of absolute median deviation above the median value. Only if exactly three spots are found does the routine continue, otherwise artifacts may have been detected. Since the detection is run for all arc lines, and the geometry is not wavelength dependent, a failed detection for a single line is not critical. The flux of all spots is then measured at the detected position, in all 60–80 exposures. This is done using simple integration in a rectangular window of  $\pm 5$  pixels in all directions, subtracting a background in an aperture of  $\pm 7$  pixels. The centroid and FWHM of each spot are measured as well, by

<sup>10</sup> The focal plane scale for VLT UT4 with MUSE is 1:705 mm<sup>-1</sup>.

<sup>11</sup> Contrary to the wavelength calibration procedure, both arc lamps simultaneously illuminate the same exposure.

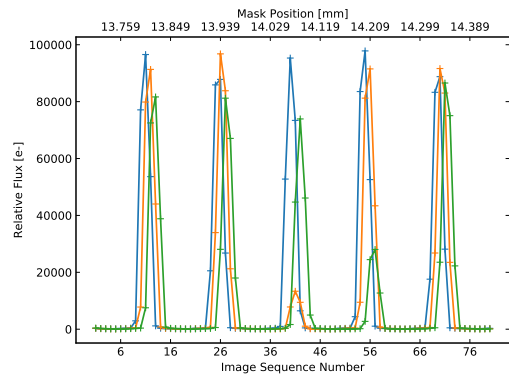




**Fig. 6.** Sketch of the geometry of selected IFUs. The four stacks of slices are indicated in the top figure which shows three selected IFUs of MUSE. The *upper part* shows that IFUs 9 and 10 partially overlap in projection to the VLT focal plane; these IFUs are also significantly offset horizontally. The *lower part* displays the approximate location of the pinholes (the pink dots) relative to the slices of the leftmost slicer stack in two of those IFUs. Slice 10 of IFU 10 is highlighted with a gray background. During the exposure sequence, the pinholes are moved vertically, the arrows represent the motion that resulted in the flux distribution depicted in Fig. 7, where the curves are displayed in the same color for the same pinhole.

using a 2D Gaussian fit<sup>12</sup>. The corresponding exposure properties (especially the mask position) is taken from the FITS header of the exposure. The result of these measurements is that for each of the three spots in each slice, for each arc line, and each exposure we have a set of CCD and spatial mask positions as well as the flux. Altogether these are a maximum of 149 760 values, but in practice about 140 000 of them are valid and are used for the further analysis.

While the vertical distance of the pinholes is known to high enough precision, the motion of the mask is not calibrated sufficiently since it includes an uncertainty about the angle of the mask inside the mask wheel and the relative motion. The effective pinhole distance therefore needs to be self-calibrated from the actual data. To do this, the centroid of all flux peaks (visible in Fig. 7) is measured in terms of the vertical position and the distance between all peaks on the scale of the vertical position. The difference between all peaks is then tracked and, after rejecting the outliers with unphysical distances, the average difference is the effective vertical pinhole distance; this is then converted to a scale factor  $f_{dy}$ . Its standard deviation indicates an accuracy of  $5 \mu\text{m}$  or better, about 4% of the slice height.



**Fig. 7.** Relative illumination of slice 10 of IFU 10 from the geometrical calibration sequence taken on 2015-12-03. The measurements are from the arc line Ne I 6383. The three colors represent the fluxes measured for the three different pinholes (see Fig. 6) that illuminate the slice. The slice is illuminated by the pinholes five times. Since the peaks occur at different positions for the different pinholes, it is already possible to determine that the slice is tilted in the MUSE FOV, by  $0.773^\circ$  in this case. Two of the pinholes are dirty, so that the orange peak near exposure 40 and the green peak at exposure 56 reach lower flux levels.

<sup>12</sup> A simpler barycenter measurement together with a direct determination of the FWHM from the pixel values was initially used, but the resulting precision was not sufficient. Since the spots in the wavelengths used for this calibration are very compact, the Gaussian is a good representation. The barycenter method can still be switched on by setting the centroid parameter to barycenter.

To start determining the slice position in the FOV, the central flux peak of the exposure series is taken. From this, the weighted mean CCD position in  $x$  and  $y$  is computed, taking into account the flux of the spot in each exposure of the peak. The mask position of this peak is recorded as well.

Using all three spots of a given slice, one can compute the scale  $s$  of a slice, using the average distance  $\langle\delta x\rangle$  between the pairs of pinholes of a slice:

$$\langle\delta x\rangle = (\delta x_1 + \delta x_2)/2 \text{ pix},$$

$$s = 2.9450 \text{ mm}/\langle\delta x\rangle \text{ pix}.$$

The width  $w$  of a slice in the FOV then follows from its width  $w_{\text{CCD}}$  as measured on the CCD:

$$w = 1''.705 \text{ mm}^{-1}/0''.2 \text{ pix}^{-1} s w_{\text{CCD}}. \quad (1)$$

Here  $w_{\text{CCD}}$  can be taken directly from the width of the slice as determined by the trace function (Sect. 3.4) at the vertical position of the detected spots. A simple error estimate  $\sigma_w$  is propagated from the standard deviation of the two measured distances  $\delta x_1$  and  $\delta x_2$  within each slice. If the width of a slice for a given arc line was determined to be outside pre-defined limits (72.2 to 82.2 pixels), the measurement is discarded.

The angle  $\varphi$  of each slice can be computed using the known distance of the pinholes in the mask and the distance between the mask positions of the maxima  $p$  between two pinholes:

$$\varphi = \arctan(\Delta p/2.9450 \text{ mm}). \quad (2)$$

Since it contains three spot measurements, each slice has two independent angle measurements. These are averaged to give one angle value per slice and arc line. Any angles above  $5^\circ$  are discarded at this stage since they are unphysical. An error estimate  $\sigma_\varphi$  is computed using the standard deviation of the two individual measurements.

The next step is to combine the measurements from all arc lines into one. This is done by an error-weighted average of all measurements of  $w$  and  $\varphi$ , after rejecting outliers using sigma-clipping in terms of average absolute deviation around the median.

Since any horizontal gap between the adjacent slices together with the width  $w$  of the involved slices determines the central position of these slices, we compute the gap based on the positions of the spots within the involved slices. To do this we take the distance  $\delta x_1$  of the left-most spot to the left edge of the slice on the CCD and  $\delta x_r$  of the right-most spot to the right edge of the slice, in CCD pixels. Of these we again compute a sigma-clipped mean over all arc lines and then compute the (central) gap in the  $x$ -direction by subtracting the measured distances from the total width of the slice as given by the mask design. With the scale factors involved we get

$$x_{\text{gap}} = 2.9450 \text{ mm} \cdot 1''.705 \text{ mm}^{-1}/0''.2 \text{ pix}^{-1} \left( 1 - \frac{\delta x_1}{\langle\delta x\rangle} - \frac{\delta x_r}{\langle\delta x\rangle} \right)$$

for the width of the gap in units of pixels<sup>13</sup>. This is illustrated in Fig. 8. For the two central slices, the initial  $x$  positions are then

$$x_{\text{init}} = \pm(x_{\text{gap}}/2 + w/2) \quad (3)$$

with  $w$  from Eq. (1). The error estimates of these are propagated from the individual measurements. The positions of the outer slices are computed in a similar way, but then depend on the positions of the inner slices.

This initial estimate of the horizontal position is already accurate for the relative positions within each IFU, but needs

<sup>13</sup> When this gap estimate is determined as negative, an average gap of 0.04 pix is taken for an inner gap and 0.07 pix for an outer gap. These values were determined as typical values for the gaps during instrument commissioning.

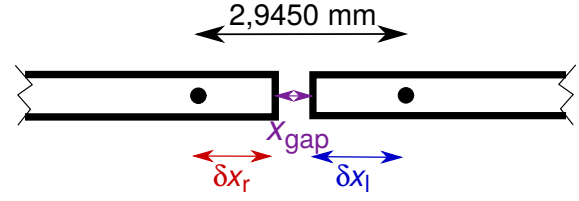


Fig. 8. Illustration of the computation of the gap between slices.

to be refined to give a correct global estimate. For this the relative positions of central spots that pass through the slices in adjacent IFUs (e.g., the top slice in IFU 10 and the bottom slice in IFU 11, as shown in Fig. 6) are compared. If the slices were centered at the same horizontal position, the spots would have the same position relative to the slice center. Any deviation  $x_{\text{diff}}$  from this can therefore be corrected by shifting all slices in all following IFUs. Since after this correction the FOV is no longer centered, the middle of the central gap of the top row of slices of IFU 12 is taken as a horizontal reference point  $x_{\text{ref}}$ , which is shifted to zero position. The fully corrected horizontal position  $x$  of the slices is then

$$x = x_{\text{init}} - \langle x_{\text{diff}} \rangle - x_{\text{ref}}, \quad (4)$$

where  $\langle x_{\text{diff}} \rangle$  is the weighted average value, after rejecting outliers, of the horizontal shifts determined for spots of all arc lines.

While deriving the weighted mean width and angle for each slice, the weighted mean mask position for the central flux peak is computed as well, which in turn is converted into a sigma-clipped weighted mean position using the data from all arc lines. This is now used to determine the last missing entry of the geometrical calibration, namely the vertical center,  $y$ . After subtracting an offset in order to center the field at the top row of IFU 12, we loop through all four slicer stacks (see Fig. 6) of all IFUs. Within each slicer stack, we go through all slices in vertical direction. Starting at the central reference slice and going in both directions, we convert the central mask position into a relative position. While doing this, we can detect whether a flux peak originates from a different pinhole in the mask since we know the approximate vertical size of the slices ( $\sim 120 \mu\text{m}$ ). This procedure results in monotonically increasing vertical positions in units of mm  $y_{\text{mm}}$ , which are then converted to the final  $y$ -position in pixels using

$$y = 1''.705 \text{ mm}^{-1}/288 f_{\text{dy}} y_{\text{mm}}, \quad (5)$$

where 288 is the number of vertical slices representing the nominal number of pixels, and  $f_{\text{dy}}$  is the scale factor that relates the nominal and effective vertical pinhole distance of the mask. An error estimate is propagated from the initial measurement of the mask position of the flux peak. One last trick in the computation of  $y$  concerns the top and bottom slices of each IFU. Since by design these may only partially be illuminated by light coming from the relay-optics, their vertical position can appear to be offset so that they seem to overlap with adjacent slices from the same IFU. Since this is unphysical, it is corrected by assuming that these edge slices will have the same vertical distance as all the other (non-edge) slices in the respective IFU.

For a few slices the computed four coordinates may be different from the surrounding slices. This occurs if artifacts like hot columns influence detection or centroid computation of the spots. In early versions of the pipeline (before v1.2) these few slices were corrected by hand in the resulting table. Since then, a new procedure was implemented to automate this. As the image slicer in each MUSE IFU is built of rigidly mounted stacks of

slices, one can take advantage of this and demand that all properties should change smoothly across the 12 vertical slices in each stack. By fitting a linear relation to each property with the number of each slice in vertical direction as the abscissa used in the fit, one can easily find the outliers among those 12 slices. The values for these are then set to the value of the fit at that slice number. This non-iterative replacement was found to perform best when used at a  $1.5\sigma$  level. Then less than 20% of the slices are changed for each of the four properties, and the resulting geometrical calibration is very smooth and no outliers are left.

It should be mentioned that with this procedure, and using Eqs. (1), (4), and (5) we arrive at a geometrical calibration that is in units of pixels. However, these are really “pseudo-pixels” that are constructed such that each horizontal and vertical element is defined to be  $0''.2$  on the sky. In reality, the sampling of MUSE is slightly different and varies across the field. The astrometric calibration (Sect. 3.12) is the step to correct this before producing the final cube that will be analyzed for scientific purposes. The geometric calibration changes very little over time. Unless instrument interventions or earthquakes affect the relative positions, only a monthly or quarterly recomputation is necessary. The geometric calibration does not need to be run by normal users, but is typically provided with the science data by ESO.

The pipeline module that carries out the procedure described in this section is called `muse_geometry`. The parameter that determines the spot detection sigma level is called `sigma` and the sigma-clipping parameter for the final smoothing is called `smooth`.

### 3.7. Twilight sky-flat fielding

The MUSE calibration unit (Kelz et al. 2012) provides an illumination that closely but not perfectly resembles the illumination of objects on the sky. To remove any residual gradients, the pipeline makes use of the bright sky background exposures, with the sequences started before sunset to reach high illumination levels in short exposures. They should very closely resemble the illumination of the MUSE instrument with a constant background. These twilight skyflats are bias-subtracted, converted from ADU to electrons, optionally dark-subtracted, and flat-fielded; then all the exposures are combined, using the chosen combination parameters (usually a sigma-clipped average). Using wavelength calibration, tracing solution, and the geometry table, each pixel in these combined images is then assigned 3D coordinates (spatial and wavelength), creating a pixel table for each IFU. Since the red part of the skyflats can be strongly affected by the spectral second order depending on the instrument mode, the pixel tables are cut in wavelength. In AO modes, the light in the wavelength region of the NaD laser light are blocked using notch filters, and pixels around this wavelength contain only noise. They are therefore excluded from further processing as well. If the optional raw illumination flat-field exposure was given as input, it is then used to correct the relative illumination between all slices of one IFU. The sum of the values in these pixel tables are computed and saved, to later be used for the relative scaling among all IFUs. The pixel tables of all IFUs are then merged as already described in Sect. 2.3 for the science data. A cube is then resampled from the merged data, using a sampling of  $250 \text{ \AA pixel}^{-1}$  in wavelength direction, and a white-light image gets created. This skyflat cube is then saved, together with the image.

This cube and image contain the residual gradients that need to be corrected in science data but also small-scale artifacts, especially strong variations at the edges of the slices and slicer

stacks. The cube is therefore post-processed to produce a correction that only contains the large-scale gradient. To this end, a mask of the illuminated area is created. If a vignetting mask was provided<sup>14</sup>, its area is not part of the illuminated region. The illuminated area is then smoothed by a median filter ( $5 \times 7$  pixels in size) to remove very small-scale outliers, normalized to 1, and fit with a 2D polynomial (by default, with order 2 in both directions), and normalized again. This is repeated for all wavelengths in the cube.

If a vignetting mask was provided or NFM data is processed, a small area close to the edge of the MUSE field is used to compute a 2D correction for the vignetted area. The original unsmoothed white-light image is corrected for large-scale gradients by dividing it with the smooth white image. The residuals in the edge area (as defined by the vignetting mask or using the top 22 pixels of the field for NFM) are then smoothed using input parameters. By default a  $4 \times 4$  order polynomial is used for this, but Gaussian or median filtering can be used instead. This smoothed vignetting correction is then multiplied onto each plane of the smooth cube before normalizing each wavelength plane of the cube again. The smoothed cube and white-light image are then saved to the disk.

The pipeline module for this procedure is called `muse_twilight`. The parameters that determine wavelength range and sampling are `lambdamin`, `lambdamax`, and `dlambda`, the polynomial orders can be changed with `xorder` and `yorder`, the vignetting model smoothing can be changed with `vignsmooth`, and the size of the built-in mask in case of NFM data can be adjusted with `vignnfm_mask`.

### 3.8. Line-spread function measurement

Measuring the line-spread function (LSF) is important within the MUSE pipeline to get good sky subtraction and to model the laser-induced Raman lines, but can also be used for scientific analysis on the reduced cube. Since the LSF is determined on data recorded by the MUSE CCDs, it implicitly includes the slit width (of the MUSE slices) and the bin width of the detector.

To derive the LSF, this module requires the same inputs as the wavelength calibration (see Sect. 3.5), as well as the final wavelength solution derived from them. The LSF is improved and better sampled if long sequences of exposures of all three arc lamps are used, so that the arc lines are detected with high S/N over the full MUSE wavelength range. As before, the raw exposures are overscan-corrected, bias-subtracted, and optionally corrected for dark current (not by default) and flat-fielded. Further processing depends on the type of LSF that is used. The MUSE pipeline mainly uses an interpolated 2D image which represents the line shape as a function of wavelength, for each IFU and slice of the instrument. An alternative implementation uses a Gauss-Hermite function to model the line shape as a function of wavelength. In this case, the coefficients are tracked in a table structure instead of an image. Since using the LSF for sky subtraction is computationally expensive and the former method is much faster and more robust, it was chosen as default. Both formats are further described in Sect. 4.10.

If the interpolated image approach is used the images of all arc exposures are converted into a special pixel table that contains only the data around the brightest and most isolated arc lines from the line list within a certain range of the known peak

<sup>14</sup> The lower right corner of the MUSE WFM field was strongly vignetted before mid-March 2017 in calibration exposures, causing higher values in the corner of flat-fielded twilight exposures.



of the line. No combination of images is carried out to avoid introducing biases at this stage. This arc-line pixel table is then divided up into the individual slices to fit the individual LSFs. The computation is then carried out as a 2D regression, where the  $x$ -direction is along the LSF (i.e., across the arc lines) and the  $y$ -direction along the wavelength (i.e., the central wavelength of the arc lines). For each step in LSF direction, all pixels of all arc lines within a certain distance range from the line peak are fit with a 2D polynomial, with order 2 in the  $x$ -direction and 3 in the  $y$ -direction. The resulting image is created by evaluating all these polynomials at their nearest output image pixel. The LSFs of all slices of one IFU are then stored together as a datacube, and saved to the disk.

If, on the other hand, the Gauss-Hermite function is used, the individual images are first combined, using given combination parameters. The resulting image is converted into a pixel table, on which the actual fit is run. The fit runs over all slices of the IFU for which the pixel table was created. In a first fit, the LSF width and the fluxes of all arc lines are minimized. The second step then keeps the fluxes constant, but fits all other Gauss-Hermite coefficients. A spectrum can then be simulated from the LSF parameters and the arc line list, and subtracted from the data for debugging. The Gauss-Hermite coefficients are finally stored into a table.

The module that implements this algorithm in the pipeline is called `muse_lsf`. The parameter method selects either the image-based interpolation or the Hermite-based table algorithm. One can adapt the interpolation method by selecting the half-size of the wavelength window around each arc line (`lsf_range`, 7.5 Å by default), the size of the interpolated image in 2D (`lsf_size` for the LSF direction, by default 150 pixels, and `lambda_size` for the wavelength direction, by default 30 pixels), and the size `lsf_regression_window` in LSF direction used for the interpolating fit.

### 3.9. Sky subtraction

As mentioned already in Sect. 2.3, the two ways to operate the sky subtraction are [a] determining and subtracting the sky in the science exposure itself and [b] observing an offset sky field to characterize the sky background to then subtracting it from the actual science exposure. The actual algorithm is the same for both; the only difference is how and when the sky spectrum is created and decomposed.

In case [a] the sky only needs to be determined and decomposed once. The sky line fluxes have not changed and can be directly subtracted from the science data. We describe the full procedure in Sect. 3.9.3 below. In case [b], however, the sky lines are expected to have changed between the exposure of the offset sky and the science field. The characterization of the sky is the same, but it needs to be stored in a way that can be applied to the science exposure. In the Sect. 3.9.2 we describe in detail how this is done.

The sky subtraction algorithm used in the MUSE pipeline was already described by Streicher et al. (2011) and the basic ideas have not changed. However, the handling of the LSF has evolved after on-sky data showed the original implementation to be insufficient (see Sect. 4.10). The MUSE pipeline tries to combine the ideas of Kelson (2003, working on unresampled data and using knowledge of sub-pixel information) and Davies (2007, employing groups of sky-lines) to derive a good description of the sky background. Since the sky continuum changes slowly (on sub-hour timescales), while most of the telluric emission lines change rapidly (within minutes or less), the

sky spectrum is decomposed. We model the sky emission lines (as described in Sect. 3.9.1) and adapt their fluxes to each exposure, and propagate the sky continuum as the residuals of a spectrum of empty sky regions<sup>15</sup>.

In the way that our algorithm decomposes the sky spectrum into emission lines and continuum, it is very similar to the approach taken in the ESO SKYCORR tool (Noll et al. 2014), developed at around the same time. However, SKYCORR is set up for single-object spectra, and hence is not well adapted to be applied to datacubes and cannot handle our unresampled pixel tables.

#### 3.9.1. Fitting telluric emission lines

The list of sky emission lines used as input to the procedure is of great importance. For MUSE we chose to divide all possible lines (Cosby et al. 2006) in the wavelength range 3129–11 000 Å into 52 groups. Most of the lines and groups are transitions of the OH molecule (7800 lines with 5100 of them within the MUSE wavelength range; see van der Loo & Groenboom 2007), and we form groups for lines that have the same upper level, which approximately vary in flux together. Other line groups originate from O<sub>2</sub> (4400 lines in the MUSE range), [O I] (71 lines, including the three brightest), H I (the two Balmer lines), N I 5198,5200, Na I D 5890,96, K I 7665,99, and He I 5015, where the single lines and doublets form separate groups. Of the many lines in the full list, those outside the wavelength range and those with a flux of less than 1/10 000th of the strongest lines are removed during the fitting process, so that typically 4000 to 4500 lines in 40 groups are used. The ESO SKYCORR tool uses a slightly different grouping of emission lines. The advantage of using groups of lines is that the fit to the line fluxes is more robust, and that many of the lines are unresolved at MUSE resolution. A drawback is that the flux calibration has to be accurate in a relative sense over the wavelength of each group. Since the sky line list is a FITS table distributed to users, the file can be edited to optimize the sky subtraction residuals, for example by removing lines that are on top of features in object spectra.

Parameters in the fitting process are flux factors for each line group and a linear correction for the wavelengths of all sky lines. The first guess uses the pixel value at the wavelength position of the brightest line of each group. The actual minimization then compares the modeled sky spectrum to the data in a differential manner, where the intensity difference of neighboring pixels forms the spectral residual  $s(p)$

$$s^2(p) = \sum_i \frac{[\Delta I_m(\lambda_i, p) - \Delta I(\lambda_i)]^2}{(\sigma_i^2 + \sigma_{i+1}^2) \Delta \lambda^2},$$

where  $\Delta I_m(\lambda_i, p)$  is the modeled intensity difference between two neighboring spectral bins  $i$  and  $i + 1$ , taking into account the fit parameters  $p$ ;  $\Delta I(\lambda_i)$  is the measured intensity difference;  $\Delta \lambda$  is the bin width of the spectrum; and  $\sigma_i^2$  is the estimated variance at each spectral position. When creating the simulated spectrum at each iteration, the LSF (Sect. 4.10) and the fluxes of all relevant sky emission lines are taken into account.

The fit is run using the global sky spectrum, averaged over all sky regions of an exposure. To allow interpolation with higher

<sup>15</sup> Another way that only works on fields with a significant part of blank sky background is to directly subtract the full sky spectrum instead of decomposing it into lines and continuum. This is implemented in the MUSE pipeline for the special case where flux calibration is not available. It can be activated in `muse_scipost` by switching the `skymethod` parameter to `simple`.

precision for the final subtraction, the sky spectrum is created with an oversampling factor of 4 to a binsize of  $0.3125 \text{ \AA}$ . The LSF for the fit is the weighted average LSF of all spectra, where the weighting is determined by how many pixels contribute from which IFU and slice to the sky, folded with the rectangular function of the sampling. After the sky line fit converged, the sky continuum is created by subtracting the line fit, again using the same weighted average LSF, from the average sky spectrum. By default, the continuum is created with the same oversampling as the spectrum. This approach has the drawback that any residuals of the sky line fit directly affect the continuum, but tests showed that any post-processing of the continuum did not result in any significant improvement. Nevertheless, the pipeline user can provide the sky continuum as input to the relevant pipeline modules to override the computed continuum.

This emission line fit is run in the module `muse_create_sky` and during processing of the on-target science data in `muse_scipost`.

### 3.9.2. Handling offset sky fields

To be able to subtract the sky background from exposures of large objects that fully cover the MUSE FOV, observations of an offset sky field are necessary (this was called case [b] above). The main reason for this is to characterize the sky continuum which changes on sub-hour timescales during the night and due to moon illumination; however, the sky emission lines change on shorter timescales. These, therefore, usually need to be characterized or refined in the actual science exposures.

Offset sky fields are processed in the same basic way as science data or standard star exposures, so the module that handles the sky exposure starts by loading and merging the pixel tables of the individual IFUs. It divides the data by the smoothed average flat-field spectrum and flux calibrates the data, using the given response and extinction curves. The correction for telluric absorption is carried out using the provided correction spectrum. Any kind of bad pixel is then removed from the table and the remaining pixels are corrected for differential atmospheric refraction (for WFM data). The next step is to create the sky spectrum, assumed to be constant across the small MUSE FOV. To this end, a cube is reconstructed using nominal instrument sampling, and a white-light image integrated over all wavelengths is created from it. Thresholding is used to first remove bad pixels (on the edge or in between image slicers in the MUSE field) to then select the chosen lowest percentile of pixels. In most cases, the resulting mask should cover a large fraction of the MUSE field, for example 75%. This mask is then used to select the corresponding pixels in the pixel table so that further work is carried out on unresampled data. The selected pixels are resampled into a 1D sky spectrum, normally using four times the nominal sampling (i.e.,  $0.3125 \text{ \AA pix}^{-1}$ ). To remove any remaining outliers, this is done iteratively, rejecting pixels deviating more than  $15\sigma$  on the second pass.

Next, the initial set of sky emission lines is read in from the specified table, and the LSF for all slices is loaded as well. The LSF is averaged using the relative contributions of the slices to the sky background area and folded with the rectangular function to match the resampled sky spectrum. This average LSF is then used to fit the fluxes of all sky emission lines, using Levenberg-Marquardt minimization. To make the fit robust and efficient, the underlying assumption is that most sky emission lines are part of groups, which vary in flux together. The solution allows for linear variation of the wavelengths against the wavelength

calibration present in the pixel table. A telluric emission spectrum is simulated using this fit and the averaged LSF and subsequently subtracted from the sky spectrum to compute the sky continuum. The table of sky lines with the updated line fluxes is saved to disk, as is the resulting sky continuum. The auxiliary files (the white-light image, the sky mask, and the sky spectrum) are also stored.

Tests showed that only short exposures are necessary for the offset sky field. Since the sky spectrum is averaged over a large part of the MUSE field, the  $S/N$  of the spectrum is extremely high already after 2 min. We estimate statistical  $S/N \gtrsim 250$  for the continuum and  $S/N > 1000$  for even moderately bright emission lines (e.g., the  $\text{NI } 5197.92, 5200.28$  doublet) for a 240 s sky exposure. When modeling the sky spectrum, systematic errors like line blends and limiting accuracy of the LSF therefore start to dominate with sky exposures  $> 60$  s.

The module that handles offset sky fields is `muse_create_sky`. The parameter `fraction` controls the area of the FOV to be regarded as sky background and (default: 0.75, since we expect the offset sky field to be almost empty), `ignore` (default: 0.05) is the fraction of the field to be ignored. Further parameters control the wavelength range used by the processing (`lambdamin`, `lambdamax`; default: all wavelengths) and set the reference wavelength for the atmospheric refraction correction (`lambdaref`, default:  $7000 \text{ \AA}$ ). Expert users can also adapt the sampling used for the sky spectrum and continuum; the relevant parameters are called `sampling` and `csampling`.

### 3.9.3. Science sky subtraction

The sky subtraction on the science data within the post-processing module starts from a pixel table that is merged from all IFUs, corrected for the lamp-flat spectrum and for atmospheric refraction. Optionally, the data are corrected for Raman signatures (see Sect. 3.10.1) and autocalibrated (Sect. 3.10.2). The data also have to be flux-calibrated. The user has to decide if the science field is empty enough to be used for the construction of a sky spectrum ([a]), in which case only the list of sky emission lines is required, or if it is filled by an object ([b]), in which case a pre-fitted sky-line list and a sky continuum is needed.

In either case a datacube is reconstructed from the pixel table and a white-light image is created from it. The image is thresholded, assuming that the darkest pixels (by default 5% of the FOV) are artifacts to be ignored, the faintest pixels (by default, the next 10% but this can be adapted for largely empty fields) are taken to be the actual sky background. All pixels from the pixel table located at these sky positions are used to reconstruct a spectrum of the sky background in the exposure.

In case [b] the choice is usually to re-fit the emission lines in the science exposure, since they are likely to have changed in flux since taking the offset sky field. Should the sky spectrum reconstructed from the science exposure be strongly affected by object emission lines, or if processing time is an issue, the refit can be skipped. In this case, the reconstruction of the cube and sky spectrum will not be done either. In case [a], the input sky line list is usually the default list, and an emission line fit is done from the sky spectrum as described in Sect. 3.9.1.

The continuum is linearly interpolated to the wavelength of each entry in the pixel table, and subtracted. The oversampling mentioned above helps to reduce artifacts when interpolating it onto each bin. The spectrum is modeled and subtracted separately from the data of each IFU and slice to remove the sky emission lines. The line fluxes are folded with the corresponding

LSF that was interpolated to the wavelength of the line and the position in the MUSE field, and then subtracted from each entry in the pixel table. This minimizes residuals due to changes of the LSF across the field.

The sky subtraction is run as part of the `muse_scipost` module. The method can be selected with the `skymethod` parameter, with the possible values `model` (the default) and `model-subtract` (no line refitting) for the algorithm described here. The `skymodel_ignore` parameter adjusts the fraction of darkest pixels to ignore as artifacts; `skymodel_fraction` defines the fraction of the field to take as sky. Expert users can adapt the sampling used for the sky spectrum and continuum. The relevant parameters are called `skymodel_sampling` and `skymodel_csampling`.

### 3.10. Internal calibrations of science exposures

Two calibration steps that can be applied to the science data itself are complex enough to be described in more detail than possible in Sect. 2.3. They are presented here.

#### 3.10.1. Correction of Raman-scattered laser light

After commissioning of the adaptive optics (AO) module with the MUSE instrument in 2017, new emission lines of telluric origin were found in the spectra. These turned out to be Raman-scattered light of the laser guide stars used to stabilize the field (Vogt et al. 2017). They appeared as features originating from  $O_2$  at around 6484 Å and  $N_2$  at 6827 Å and consist of bright peaks, unresolved at the spectral resolution of MUSE, with a band of faint secondary peaks extending to about 50 Å from the main lines. Since they vary spatially around the laser beams, they have to be modeled spatially, as described in the following, when observing in the WFM AO modes. For NFM and crowded WFM observations, objects in the field usually preclude such an approach, and a constant is subtracted as part of the sky continuum.

For the modeling, it is assumed that the flux ratio of the peaks to the secondary lines of each feature is constant. The relative fluxes of each single Raman line are computed from molecular physics (following the prescription of Vogt et al. 2019), and are given to the pipeline together with the exact wavelength as input. To model the absolute fluxes across the field, we extract the data of the affected wavelength ranges (by default,  $\pm 10$  Å around the main peaks) and reconstruct an image for each range. Sky regions are selected in these images, and the corresponding object pixels as well as pixels marked as bad (cosmic rays or CCD effects) are removed from the pixel tables. The remaining sky pixels are fit, using the estimations of the LSF at this wavelength, with the Raman lines. The minimization adapts the absolute fluxes of all Raman lines in a given feature, using a second-order 2D polynomial as model for the spatial domain. This fit is run separately for the  $O_2$  and  $N_2$  features, and removed from the science data. The corresponding spatial flux distribution can be output, for both features, for further checks.

In the pipeline implementation<sup>16</sup>, the Raman correction is done as part of the `muse_scipost` module. Besides the input table of Raman lines, the only parameter is the extraction width, this is called `raman_width`.

<sup>16</sup> This was first implemented in v2.4 of the pipeline, and updated with a new line list in v2.8.

#### 3.10.2. Self-calibration of fluxes within each slice

Flat-fielding removes spatial structure from MUSE exposures to about 1–1.5% accuracy. When integrating the datacube over many wavelength bins, for example when creating integrated images over broad-band filters or the whole wavelength range, a pattern is left that contains the four stacks of slices within each IFU and even to the level of single slices that have slightly different illumination than neighboring pixels. An autocalibration on the slice level can be activated when much of the MUSE field is blank sky background to facilitate object detection in deep surveys.

This procedure uses the sky background signal to compute reference fluxes, and is hence applied before sky subtraction. It is assumed that the background is intrinsically flat across the MUSE field<sup>17</sup> and that the science exposure itself contains sky in at least a few ( $\sim 1/3$ ) spectral pixels (spaxels, a spectrum in the datacube) of each slice, meaning that no large objects are present. By default, the MUSE pipeline then constructs a white-light image from the pixel table by resampling to a cube and then integrating it over the full MUSE wavelength range. Using this image, a sky mask is created by thresholding it to  $\pm 15\sigma$  (in terms of median absolute deviation around the median). A morphological opening of the resulting mask with a kernel of  $3 \times 3$  pixels ensures that contiguous regions are marked as sky. Then all pixels coinciding with the spatial position of the sky regions are selected in the pixel table. The actual algorithm to determine the flux correction factors for each slice then divides the data into 20 wavelength ranges (19 for AO data where the NaD region is masked). Correction factors are derived for all slices of all IFUs and all wavelength ranges, labeled “segments”, so 23040 (21888) individual factors are computed<sup>18</sup>. These ranges are hardcoded and were chosen to end in between groups of sky-lines to minimize the influence of the sampling that would be critical if a strong telluric emission line were on the edge.

Within each segment the reference level is computed first, if data of at least 50 pixels within 20 spaxels exist (i.e., are not marked as object). The reference is determined in a multi-step process as the MAD-clipped mean and deviation (where MAD-clipping refers to values computed after rejecting outliers outside the  $3\sigma$  median absolute deviation around the median) for all spaxels, then again averaging the spaxels with  $3\sigma$  MAD rejection on both spatial halves for all IFUs, and then again averaging those with plain mean and standard deviation rejection. In addition to the reference flux for a particular wavelength range across the whole sky portion of the FOV, reference values for each IFU and the spatial left and right half of each IFU are tracked. The correction factors in each segment are then computed as the ratio of the reference flux to the flux in the segment. Should the correction factor in one segment exceed the MAD-clipped ( $15\sigma$ ) mean value in the same half of the IFU, the correction factor is taken as the mean correction for this half of the IFU instead. The correction factors for the adjacent wavelength ranges are checked for large deviations as well, so that if the deviation is larger than the maximum of 3% or  $3 \times$  the difference between

<sup>17</sup> For exposures taken in WFM with the AO laser system active, this needs the Raman correction described in Sect. 3.10.1 so that it is also true in the wavelengths around the  $O_2$  and  $N_2$  features. This was changed in v2.8.3 of the MUSE pipeline; previous versions ran the self-calibration first.

<sup>18</sup> A similar function was available in the MPDAF package (Piqueras et al. 2019) before it was ported to the MUSE pipeline v2.4 and subsequently phased out from MPDAF.



these adjacent bins, the mean of the corrections for those two ranges is taken instead.

The correction factors computed in this way are then applied to the pixel table data, depending on the segment (slice and wavelength range) in question. The corrections are applied in quadrature to the pixel variances.

Two special cases can be handled as well. First, if the positions of the objects in the field are already known from ancillary data, the user can provide an external sky mask. If it contains a world coordinate system with sky coordinates, it is used to align the mask to the MUSE data. The computationally expensive step of creating the white-light image can then be skipped. Providing such an optimized mask can significantly improve the results. Second, in some cases a large object is present in the FOV, but several exposures of the same field were taken, so that different parts of the instrument were illuminated by this object. In that case, the resulting tables with correction factors could be combined with appropriate algorithms for clipping, for better results in a second iteration of this reduction step. The MUSE pipeline supports this by allowing the user to input a table with the corrections.

This function was developed to improve final data quality of deep fields observed with MUSE, in particular the HDF-S (Bacon et al. 2015) and UDF (Bacon et al. 2017) fields, and the wavelength ranges, clipping methods and sigma levels were adapted to give optimal results for these projects after thorough testing. It is possible that for some other datasets individual segments are assigned deviant correction factors. In this case, a discontinuity in the spectra at the edge between the wavelength ranges would show up<sup>19</sup>. Optimizing the sky mask by hand is often a way to improve the results.

This functionality is available in the `muse_scipost` module; it is switched on if the parameter `autocalib` is set to `deepfield`. The user-provided table is read if it is set to `user` instead. By default, this self-calibration is switched off (parameter set to `none`).

### 3.11. Standard star handling

In MUSE data processing, spectrophotometric standard stars are used to compute the instrument throughput and to derive a spectrum of normalized telluric absorption. The throughput is determined in the form of a response curve that is used to flux-calibrate the science data.

Standard stars get the same basic processing applied as science data, so this module also starts with merging the pixel tables, dividing by the smoothed average flat-field spectrum, and correcting for differential atmospheric refraction (in WFM). A cube of the stellar field is then reconstructed, using the nominal MUSE sampling of  $1.25 \text{ \AA pixel}^{-1}$ . Object detection using simple thresholding (with levels of  $50-5\sigma$ , until at least one object is found) on the central wavelength plane of the cube determines initial positions for all possible stars in the field.

Then the flux of all stars is integrated on each wavelength plane of the cube. For WFM (both with and without AO) the point-spread function on the sky is represented very well by a Moffat (1969) function (Husser et al. 2016; Kamann et al. 2018a). When directly fitting this function to extract the fluxes of the detected stars, changes due to noise can result in unphysical differences even in adjacent wavelength planes. The consequences would be increased noise in the output spectrum and

wavy continua in extracted spectra. Hence, we used the idea of PampelMUSE (Kamann et al. 2013) and first fit a free elliptical Moffat in all wavelength planes in the reconstructed cube. Then we fit second-order polynomials to the central position and all Moffat parameters, iteratively rejecting wavelengths where a parameter deviates by more than  $3\sigma$  from the polynomial. Then we re-fit the Moffat at every wavelength plane, fixing the parameters to the polynomial parameter solutions at every plane so that only the flux and the background level remain free parameters. This “smoothed Moffat” results in high S/N spectra for all stars in the field, and is used as the default extraction mechanism for WFM standard star exposures. For NFM, the profile is more complex and cannot be modeled fully analytically<sup>20</sup>. So a simple circular aperture with a sky annulus is taken. Depending on the mode, these are the automatically chosen defaults<sup>21</sup>. The extraction window at each wavelength is determined from the spatial FWHM of the exposure, given by metadata about ambient conditions (the observatory seeing) or by measuring it on the central wavelength. For the aperture extraction  $4 \times FWHM$  is used, and the (Moffat) fit is carried out over  $3 \times FWHM$ . Once the total flux of each object over all wavelengths is known, the pipeline selects the star to use (either the brightest one or the one closest to the field center). For most standard star fields, only a single star is detected.

Then the measured fluxes of the selected star are compared to the interpolated fluxes from the reference flux table of the target field, taking into account the airmass of the target and the extinction curve that was provided by the user<sup>22</sup>. The relation

$$s(\lambda) = 2.5 \log_{10} \left( \frac{d_{\text{ct}}(\lambda)}{t_{\text{exp}} \Delta \lambda f_{\text{ref}}(\lambda)} \right) + f_{\text{ext}}(\lambda) A \quad (6)$$

describes the sensitivity  $s$  computed at each wavelength  $\lambda$ , with the recorded flux in counts (electrons)  $d_{\text{ct}}$ , the effective airmass  $A$ , the exposure time  $t_{\text{exp}}$ , and using the reference flux  $f_{\text{ref}}$ . This curve is post-processed as follows. Wavelength ranges known to be affected by telluric absorption are marked and interpolated across with second-order polynomials. The fractional difference between the fit and the original data is taken as the telluric absorption factor  $f_{\text{tell}}$ ; between the telluric regions it is set to 1<sup>23</sup>. The final response curve  $f_{\text{resp}}$  is obtained from  $s(\lambda)$  by extrapolation to the largest possible MUSE wavelength range and then smoothing it. Smoothing can be done using a median filter, but piecewise cubic polynomials followed by a sliding average is usually more effective to reduce noise and reject outliers in single wavelengths. Finally, using the known effective area of the telescope, a throughput spectrum is computed from the smoothed response curve.

The whole procedure is repeated for each exposure given as input, but no attempt is made to combine the resulting response curves or compute an improved extinction curve. For each

<sup>20</sup> The function described by Fétick et al. (2019) in principle fits the AO-correction NFM PSF in Fourier-space. However, tests show that it is not robust enough to be used in an automated pipeline environment, where small-scale artifacts might be present.

<sup>21</sup> This automatic selection and the “smoothed” Moffat fit are new in v2.8; older versions always defaulted to non-iterative Moffat fits.

<sup>22</sup> This process was modeled following the widely used implementation in IRAF’s ONEDSPEC package, especially the SENSFUNC and CALIBRATE tasks.

<sup>23</sup> This procedure assumes that the sensitivity is smooth within the telluric absorption bands. This is true for all approved spectrophotometric standard stars used by the MUSE calibration plan, but excludes cooler stars often used as telluric stars in other projects.

<sup>19</sup> This can be checked visually using the MPDAF function `plot_autocal_factors()`.

exposure, the response and the table of telluric correction factors are saved to disk.

The pipeline recipe for this procedure is `muse_standard`. The parameter that controls the method is called `profile` and with `select` the way to choose among multiple detected sources can be changed. The smoothing behavior can be influenced with the `smooth` parameter.

### 3.12. Astrometric calibration

Since the geometric solution (Sect. 3.6) only corrects per-slice offsets, a calibration of the overall distortion and pixel scale of MUSE has to be done on sky, using astrometric fields. These are located in Milky Way Globular Clusters with existing *Hubble* Space Telescope imaging, so that reference catalogs of a few hundred stars over a MUSE field exists. Fields in the outer parts of the clusters were chosen for WFM, while positions closer to the centers and with a bright star in the field were selected for NFM.

Astrometric fields again get the same basic processing applied as all other on-sky data, so this module again starts with merging the pixel tables, dividing by the average flat-field spectrum and correcting for differential atmospheric refraction (in WFM). Here, optionally, the data can be flux calibrated if the related input data (response, telluric, and extinction curves) were given on input, but this is usually not necessary. A cube of the medium-dense stellar field is then reconstructed. Since the distortions are achromatic, a large sampling in wavelength is used ( $50 \text{ \AA pixel}^{-1}$ ) to improve S/N. The central three wavelength planes of this cube are combined using the median to remove any last cosmic rays and other artifacts. Thresholding at a given sigma limit is then applied to detect the reference stars in the exposure before their position is determined more accurately using (Moffat) profile fits. Two-dimensional pattern matching is used to identify the detected objects against the sky positions from the reference catalog. Here, we based our implementation on the `kd-match` library of Heyl (2013)<sup>24</sup>, based on the search of quadrilaterals<sup>25</sup>. We improved on their code by optimizing it for the case of the actual MUSE astrometric fields. For a given transformation judged to be valid, 80% of the detections have to match a catalog entry within the given radius. All matched objects (typically around 100) are then used to fit the astrometric solution using a six-parameter world coordinate representation (with zero point position, two scales, rotation, and shear) in the gnomonic projection of the tangent plane. The fit is typically iterated twice with a given sigma-clipping rejection. This reduces the effect of stars in the foreground of the cluster, which have different proper motions, on the final solution. Since the absolute astrometry of the astrometric field is not of interest, the zero point of the fit is ignored when the calibration is applied to science data.

The corresponding pipeline module is called `muse_astrometry`. The most important parameters are `detsigma` the sigma level to use for object detection; `centroid` is the centroiding method (Gaussian or Moffat fits); `radius` is the match-

ing radius; `niter` is the number of iterations of the final fit; and `rejsigma` sets the sigma level for the iterative rejection of the fit.

### 3.13. Exposure offset calculation

When combining multiple MUSE exposures it is usually necessary to correct for relative coordinate offsets. To apply an offset correction a list of the measured relative offsets for each of the exposures has to be prepared with respect to a given reference.

The automatic calculation of offsets uses the reconstructed FOV images of all exposures involved. It then measures the individual, relative coordinate offsets with respect to the first exposure. In the first step, the algorithm creates a list of detected sources, one for each input image. An implementation of the DAOPHOT (Stetson 1987) detection algorithm is used to find sources in the MUSE field, and is thus targeted on the detection of point sources. The parameters used to fine-tune the DAOPHOT star-finding algorithm, especially the FWHM of the Gaussian convolution filter, and the roundness and the sharpness criteria can be adjusted.

For each of the input images the source detection is iterated adjusting the detection threshold until the number of detected sources falls within pre-defined limits given by the recipe options or the maximum number of iterations is exceeded. Starting with an initial detection threshold, the detection threshold is adjusted in subsequent iterations by a given step size. The initial detection threshold is either a given absolute value or it is calculated from a given multiplier and an estimate of the background level of the current input image and its uncertainty. In the latter case, the initial detection threshold value is calculated as uncertainty of the background above the background estimate, where the background level and its uncertainty are estimated from a fraction of the pixel values at the low end of the pixel value distribution of the image. Finally, all sources located closer to a bad pixel than a given minimum distance are discarded.

Once the source lists for all input images are created, the field offsets are computed iteratively for a sequence of decreasing search radii. For a given search radius and for every possible pairwise combination of the fields, the relative offsets between the objects that were detected in each of the two exposures are calculated. The offsets in right ascension and declination are computed for every possible combination of the detected objects in the two fields.

The offsets measured for each pair of objects are then used to calculate the relative offset of two fields. The initial estimate of the relative field offset, in other words the estimate for the largest search radius, is determined as the mode of the 2D histogram of the relative offsets of the object pairs. Object pairs whose distance is larger than the search radius are excluded. In subsequent iterations, smaller search radii are used to refine the estimates of the field offsets. Here, the object pairs used to calculate the field offset are again selected by applying the previously mentioned distance criterion, taking into account the field offset calculated in the previous iteration step. The median of the relative offsets of the selected object pairs is then used as the measured relative offset of two fields.

Once the field offsets for all pairwise combinations of fields have been measured, the final field offsets for a given search radius are determined by a least-squares fit of all the measured field offsets. If weighting is enabled, the weights for the fit are initially the peak value of the histogram, and, in subsequent iterations, the variance of the relative offsets of the selected object pairs.

<sup>24</sup> <https://ubc-astrophysics.github.io/kd-match/>

<sup>25</sup> Originally, the triangle-based matching algorithm implemented in the FORS pipeline (Izzo et al. 2016) was used, but it was not robust enough on the scales of MUSE data and especially at NFM resolution. So with v2.8 it was changed to `kd-match` as default, but the older algorithm is still available. It also needed inputs regarding accuracy and matching radius, where the radius was automatically decreased  $1.25\times$ , until only unique matches were found.

The relative field offsets, which are eventually written to the offset list, are the field offsets computed for the smallest search radius. If one of the input exposures is found to not overlap with any of the other exposures, its relative field offsets are set to zero. During the combination of the exposures these relative offsets are used to correct the initial field center given by the FITS header keywords RA and Dec of the exposures. For input exposures that do not overlap with any other input exposure this means that no correction is applied, and the exposure is combined with the other input exposures using the header entries as they are.

Together with the offset list an approximate preview image of the combined FOV is generated. This allows the visual assessment of the computed offsets. It is also possible, optionally, to create an exposure map for the FOV.

The pipeline module that implements this algorithm is `muse_exp_align`<sup>26</sup>. The parameter that set roundness and sharpness are `roundmin/roundmax` and `sharpmin/sharpmax`, the FWHM of the convolution is `fwhm`, the number of sources is given with `srcmin/srcmax`, and the maximum number of iterations is controlled by `iterations`. One can modify the initial threshold and step-size using `threshold` and `step`. The initial threshold is taken as an absolute pixel value if `threshold` is larger than zero, and otherwise its absolute value is used as multiplier when calculating the initial threshold from the background estimate. The pixel values used for the background estimate can be chosen using `bkgignore` and `bkgfraction`, to exclude and select a fraction of the pixel values. The minimum distance between a detected source and the closest bad pixel is given using `bpixdistance`. The search radii can be adjusted using `rsearch` and weights are taken into account for the final fit when `weight` is set. The number of histogram bins is `nbins`.

## 4. Common algorithms

Some processing steps are used in several of the modules of the MUSE pipeline or represent central design choices. These procedures or algorithms are described in detail in this section.

### 4.1. Error propagation

The initial estimate of the noise of each pixel is done from the data itself. For this we assume that the errors on the individual pixels are independent. The variance of pixels in an exposed frame then consists of the photon noise, the read-out noise, and the error on the read-out noise estimate:

$$\sigma_{\text{initial}}^2 = \frac{v-b}{g} + \left(1 + \frac{1}{n_b}\right) \sigma_b^2, \quad (7)$$

(following Gössl & Riffeser 2002) where  $\sigma_b^2$  is the variance of a bias frame (derived from the RON, see Sect. 3.1) in units of  $\text{adu}^2$ ,  $n_b$  is the number of pixels used to determine the RON,  $v$  is the value of a pixel in  $\text{adu}$ ,  $b$  is the bias level of the exposure in  $\text{adu}$ , and  $g$  is the detector gain or conversion factor in electrons per  $\text{adu}$ . Computed values of  $\sigma_{\text{initial}}^2 < 0$  (in regions of low signal) are set to zero. This is a noisy estimator, as discussed by Bacon et al. (2017), and if used directly (i.e., for inverse variance weighting) can lead to incorrect results if a source is detected as low S/N.

For any subsequent operation on a pixel, Gaussian error propagation ensures that the variance after the operation is computed correctly. For any function  $f$  that affects two images  $a$  and

$b$ , the variance is therefore computed as follows:

$$\sigma_{f(a,b)}^2 = \left(\frac{\partial f}{\partial a}\right)^2 \sigma_a^2 + \left(\frac{\partial f}{\partial b}\right)^2 \sigma_b^2.$$

A simple, non-weighted average of a sequence of  $n$  images then takes the form

$$\sigma_{\text{average}}^2 = \frac{1}{n^2} \sum_{i=1}^n \sigma_i^2. \quad (8)$$

It should be pointed out that the treatment of the science data in the MUSE pipeline does not depend on the variance estimate, which is only propagated through the multiple processing steps.

### 4.2. Bad pixel processing

The MUSE pipeline tries to suppress CCD defects, cosmic rays, and other artifacts so that they do not contaminate the science data in the final datacube. Such artifacts are found in a multi-stage process. Some hot columns can be detected on bias images, some as dark columns on flat-fields, for example. These are marked, first on the image level in the DQ extension of the master calibration products, then in the data quality column of the MUSE pixel table. To encode the nature of the artifact, we use the data quality convention invented for the Euro3D format (Kissler-Patig et al. 2004). The possible quality values are documented as bitwise flags in Kissler-Patig et al. (2003, their Sect. 4.6.1). From these flags we use only a subset in the MUSE pipeline which we list in Table 1.

Any process that uses bad pixels (those with flags greater than zero) propagates the bits of the flag on to subsequent processing, using bitwise logical OR. Since not all bad pixels can be found using automated means, an extra table can be given to all pipeline modules that load raw data. Pixel positions and flags stored in that table are then propagated to the data in the DQ extension and the pixel table in the same manner.

In the final cube (see Sect. 4.5) written by the pipeline, the voxels (volume pixel, one element of the datacube) flagged in the DQ extension are replaced by NAN values in both the DATA and the STAT extension to save 1/3 of the data volume. Since the cube is resampled using 3D information, most of the bad pixels in the middle of the data have been interpolated over from neighboring good pixels, especially if the cube was reconstructed from multiple overlapping exposures, and only voxels on the edges of the cube are usually left as flagged.

### 4.3. CCD overscans and trimming

The CCD overscan regions<sup>27</sup> play an important role in the MUSE instrument. Since the bias level in the data section<sup>28</sup> of the CCDs has gradients that change with time, these regions can be analyzed to correct for this.

Pixels with (strongly) deviant values sometimes appear in the analysis of the overscans. These are caused either by cosmic ray hits or by hot columns or pixels whose read-out spills into the overscan regions. When computing statistics on the overscans,

<sup>27</sup> In the FITS images of the MUSE raw data, the ‘‘overscans’’ are located in a cross in the inside, between the CCD quadrants. They result from just continuing to read the CCD beyond the physically available pixels. The ‘‘pre-scan’’ regions that represent actual pixels on the outskirts of the CCD, which do not get illuminated, are present as well, but they do not sufficiently represent the bias level in the data section. Both are visible in Fig. 2.

<sup>28</sup> The ‘‘data section’’ corresponds to the illuminated part of the CCDs.

<sup>26</sup> First introduced into the MUSE pipeline v1.2.



**Table 1.** Bad pixel flags used in the MUSE pipeline.

Flag value	Bit-shift	Data quality condition
0	0	Good pixel – no flaw detected
1	1	Affected by telluric feature (corrected) <sup>(a)</sup>
2	1 $\ll$ 1	Affected by telluric feature (uncorrected) <sup>(a)</sup>
32	1 $\ll$ 5	Cosmic ray (unremoved) <sup>(b)</sup>
64	1 $\ll$ 6	Low QE pixel (<20% of the average sensitivity; e.g., defective CCD coating, vignetting)
256	1 $\ll$ 8	Hot pixel ( $>5\sigma$ median dark)
512	1 $\ll$ 9	Dark pixel (permanent CCD charge trap)
4096	1 $\ll$ 12	A/D converter saturation (signal irrecoverable, but known to exceed the A/D full-scale signal)
8192	1 $\ll$ 13	Permanent camera defect (such as blocked columns, dead pixels)
16384	1 $\ll$ 14	Bad pixel not fitting into any other category <sup>(c)</sup>
2 <sup>30</sup>	1 $\ll$ 30	Missing data (pixel was lost)
2 <sup>31</sup>	1 $\ll$ 31	Outside data range (e.g., outside of spectral range, inactive detector area, mosaic gap)

**Notes.** <sup>(a)</sup>In the MUSE pipeline telluric features are only marked in the standard star processing, but never propagated to science data. <sup>(b)</sup>The corresponding “removed” cosmic ray from the Euro3D specifications is not used. <sup>(c)</sup>In the MUSE pipeline this is used for non-positive pixels in flat-field images.

these should therefore be removed. In the MUSE pipeline this can be done using the DCR cosmic ray rejection routine (Pyech 2004) that was tuned to do optimal rejection in the overscan regions. Other possible options are only useful for testing, and iteratively fit a constant to the whole overscan region or to not do any value rejection. When computing statistics of the overscans, it can also be useful to ignore a few pixels that are located next to the illuminated data section; in this way, flux that may spill over due to less-than-optimal charge transfer efficiency of the CCD electronics does not get included. Typically, discarding bands of three pixels in width is good enough to get clean statistics, even if an exposure is illuminated close to saturation.

The first possibility is to ignore the overscan region. In this case the user is still warned if the overscans are found to be very different from the bias level in the data section (if this can be determined, i.e., for bias images). The sigma level of this warning can then be tuned. The second possibility is to compute the mean value in the overscans belonging to each quadrant. When combining multiple exposures or subtracting the master bias from other images, the offset between the mean values of the images in question is taken out. The last possibility, and the one that is typically used because it gives the best bias correction for MUSE data, is to model the slope of the vertical overscan with a polynomial. This polynomial is computed iteratively, by default with a  $30\sigma$  clipping. The order of the polynomial is increased until a good match is found, depending on the root mean square (rms) of the residuals and the  $\chi^2$  of the fit. Testing against deep dark frames (see Sect. 6.1) showed that most CCD overscans can be nicely modeled with a fifth-order polynomial, but one or two quadrants even need up to 15th order for a good fit. To not use such high polynomial orders for every vertical overscan, the next higher-order polynomial is chosen if it decreases the rms by more than a factor of 1.00001, and the fit quality  $\chi^2$  decreases by more than 1.00001<sup>29</sup>. In practice, 83% of the CCD overscans are fit with a polynomial order of five or lower. The polynomial is then subtracted from the data of the whole quadrant before combining it with other exposures or before removing a master bias which was already treated in the same way.

Once the overscan-based processing was executed, only the data section of the raw data remains relevant, and the regions of

overscan and pre-scan are cut off. Depending on the processing stage, statistics of the overscans are kept internally, for example, to facilitate combination with other exposures.

#### 4.4. Image combination methods

Several calibration processes require a sequence of exposures to be combined on the CCD level. This is done using image combination. While this is widely available in data processing packages (e.g., IRAF<sup>30</sup>), the specialty in the MUSE pipeline is that these procedure are aware of bad pixels and pixel variances. In the simplest case, the average, this means that any pixels flagged as bad at the same pixel position are discarded when computing the mean value. The propagated variance is then given by Eq. (8). If all pixels at one position are flagged as bad, the pixel with the least severe flaw and its variance are taken, and its flag is propagated.

In case of an image sum, any flagged pixels are discarded and the total sum is then computed by scaling the partial sum back to the total sum of the number of images involved. The variance is computed the same way, with the factor in quadrature.

An image median is computed by sorting the input values and taking either the middle value (for an odd number of input images) or the average of the two central values (for even inputs). Then the variance corresponding to either the middle value (again, for odd numbers) or the propagated average of the two middle values (for even numbers) is taken as output variance. This again uses only unflagged input pixels and only falls back to the least severe flaw in case all inputs were flagged.

Since these simple methods are either affected by noise peaks or cosmic rays or do not deliver the optimal S/N, two more options using value rejection are implemented for MUSE. These sort the values at each position, and then discard the outliers, either using simple minmax or sigclip rejection. The first uses a fixed number of user-selected outliers at each end of the pixel distribution, while the latter computes median and median deviation<sup>31</sup> and discards values outside the boundaries given by

<sup>29</sup> Until v2.6, a maximum fifth-order polynomial was used, with a 1.01 rms decrease and  $\chi^2 < 1.04$ .

<sup>30</sup> IRAF was written and supported until 2013 by the National Optical Astronomy Observatories (NOAO) in Tucson, Arizona, see Tody (1993). A [community edition is now available](#) for legacy applications.

<sup>31</sup> This might be better described by average absolute deviation against the median.

this deviation and two factors. Pixel selection for the initial histogram only uses unflagged pixels. The average of the remaining data is then computed with variance propagation as explained above.

For the MUSE pipeline modules that carry out such image combinations from raw data (e.g., `muse_bias`, `muse_wavcal`), the default is always the `sigclip` option. In all cases, the sigma-clipping factor was optimized for the data typically handled by the module, but can be adjusted (parameters `lsigma` and `hsigma`). For special cases, other combination methods can be chosen by the user with the parameter `combine`.

#### 4.5. Cube reconstruction

Since two of the high-level goals in the design of the MUSE pipeline were to propagate the variance from the raw data to the final product and to create optimal data quality, only a single resampling step is used in the science reduction. This is the last step, the reconstruction of the output cube. As elsewhere in this paper, we call the elements entering this process pixels<sup>32</sup>, while the output elements of the cube are called voxels.

To make resampling computationally tractable, the input pixels are sorted into 3D grid cells, each representing one output voxel. Each grid cell can contain no pixels, one pixel, or many pixels, and the pixels are assigned to them in a nearest-neighbor fashion depending on their 3D coordinates with respect to the coordinates of the grid cells. To then compute the data value of the output voxel, each input pixel is assigned a weight. This weight depends on the resampling method chosen. Flagged pixels are ignored.

The cube reconstruction works in the same way for a single and multiple exposures. In the latter case, however, multiple pixel tables enter the process, and so multiple pixels may be assigned to each grid cell, where exposures overlap. The process assumes that all exposures were taken under similar conditions. Strong deviations in spatial FWHM (seeing) and transparency (cloud cover) may lead to artifacts.

It should be noted that this resampling process causes variance to be transferred into covariances. Since we cannot store those covariances due to the size of the data involved, they are lost after reconstructing the cube. However, as the MUSE pipeline only resamples the data once, this means that the variances stored in the cube are accurate per voxel to the level where we can derive them from the data by the procedure given in Sect. 4.1. Only when further analyzing the data by integrating voxels in any of the three dimensions will the variance of the integrated data be underestimated.

The actual methods used to compute the output value of each voxel using the pixel grid are discussed below. In the pipeline modules that allow this, the method can be chosen by setting the `resample` parameter.

##### 4.5.1. Cosmic ray rejection

Cosmic ray detection is an issue for all astronomical observations that record the data on CCDs, and MUSE is no exception. Most of the time the solution is to do statistics of pixels in small apertures (e.g., Pych 2004) in 2D or to filter the CCD image (e.g., van Dokkum 2001; Husemann et al. 2012). With 3D data

there are three advantages. First, each pixel has 26 instead of 8 neighbors, so the statistics can detect outliers to much lower levels. Second, additional power comes from the fact that data contributing to voxels that are adjacent in the 3D grid partly originate from regions in the raw CCD-based images that are very far from each other. This reduces the number of pixels in the statistics to about 18, but makes it highly unlikely that the reference pixels are contaminated by a cosmic ray hit as well. Third, if there are multiple ( $n$ ) exposures that overlap on the sky and hence at least partially in the 3D grid, one can compare the statistics of  $18 \times n$  (or  $26 \times n$ ) pixels, and even more efficiently detect cosmic rays.

The implementation in the MUSE pipeline follows the approach of running the cosmic ray rejection once the grid-cells (that define the output voxels, as mentioned above) are set up and linked with all pixels whose centers are located within them in 3D. Basically, the pipeline then loops through all grid cells, and then through all pixels assigned to the grid cell itself and the directly surrounding grid cells and computes statistics of all pixel values, and their variances. By default, the variances are ignored, and only the median of all surrounding pixels as well as the median of the absolute median deviation are computed. All pixels in the central grid cell that are above a given sigma level with respect to the median are then marked as cosmic rays. Alternative statistics can be selected instead, then either mean and standard deviation are used to compute the lower limit for cosmic rays. Or an estimator using the estimated pixel variances can be used, which was inspired by the AVSIGCLIP rejection algorithm available in several IRAF tasks. In the latter case the mean is compared to the local average noise. Both alternatives are faster to compute the statistics, but much less effective at similar sigma levels, and much more likely to clip real peaks with correspondingly tighter constraints. They are therefore only useful for special purposes.

If the angle between the input data and the output grid is a multiple of  $90^\circ$  (or within  $5^\circ$ ), a simple operation can determine which pixels originate from the same slice on the CCD. If this is the case, the pipeline will ignore those close neighbors when computing the statistics and make cosmic ray rejection more effective.

In the pipeline modules that resample data into a cube, the parameter `crtype` can be changed to set the type of statistics to use while `crsigma` allows to change the sigma level for the rejection. Tests have shown that for single exposures a  $15\sigma$  cutoff works well with median statistics, while for multiple exposures a tighter  $10\sigma$  cutoff is more effective without destroying data. These are set as defaults. In the typical range of exposure times used with MUSE, the pipeline detects cosmic ray hits that affect between about 0.06% (for 600 s) and 0.11% (for 1500 s) of the pixels of a MUSE exposure.

##### 4.5.2. Drizzle-like resampling

The primary method for this step uses a technique inspired by the Drizzle algorithm used for HST imaging data (Fruchter et al. 2009). In this case the weight for each pixel is computed using the geometrical overlap between pixel and voxel.

The calibrations applied to the data ensure that the coordinates of the center of each pixel within the output grid are known to high accuracy. However, they do not form a contiguous image as in the case of the original drizzle algorithm, but are irregularly spaced within the output grid. Due to the size of the data, the orientation and individual input size of the pixels cannot be tracked. Hence, we have to assume that all pixels are the same size and

<sup>32</sup> These pixels are internally represented as rows in the pixel table(s), and still in a one-to-one relation to the original CCD pixels of the one or several exposures involved in this process.

that the rotation is negligible. Then, the formula to compute the weights is

$$t = \begin{cases} \max [t_{\text{out}}, 0], & \text{if } t_{\text{out}} + dt \leq t_{\text{in}} \\ \max [(t_{\text{in}} + t_{\text{out}})/2 - dt, 0], & \text{otherwise} \end{cases},$$

where  $t$  is one of the three coordinates ( $x, y, z$ ),  $t_{\text{in}}$  is the input pixel size,  $t_{\text{out}}$  is the output voxel size, and  $dt$  is the distance between the centers of input pixel and output voxel. The total drizzled weight  $w$  for an output pixel can then be computed as

$$w = \prod_{x,y,z} \frac{\max[t, 0]}{t_{\text{in}}}.$$

This algorithm conserves the flux of the objects within the FOV, and was therefore chosen to be the default method in the MUSE pipeline. It can be set by selecting the `drizzle` option of the `resample` parameter. Scaling of the input pixel size to give  $t_{\text{in}}$  is set using `pixfrac` which can be different in all three dimensions.

#### 4.5.3. Nearest neighbor

Another much simpler approach to compute the output voxel values is to just use the value of the pixel lying closest to the center of each grid cell. This is the fastest method, but if one is resampling more than one exposure into a cube, the improvement in S/N is lost. It is therefore only useful to provide a quick look.

This method can be set by selecting the nearest option of the `resample` parameter.

#### 4.5.4. Other weighted resampling types

The pipeline uses several other weighted resampling schemes. Common to all is that for each input pixel a weight is computed that is derived from the distance between the center of the grid cell defining each voxel and the 3D location of the center of the pixel. It should be pointed out that these methods do not conserve the flux as well as the Drizzle method.

In case of `linear`, the weight is computed as the inverse linear distance, while for `quadratic` the squared inverse distance is taken. The option `renka` is following the approach of Renka (1988) and uses the relation

$$w = \left[ \frac{r_c - r}{r_c r} \right]^2,$$

where  $r_c$  is the critical distance after which the weight is set to zero.

The normalized sinus cardinal function  $\text{sinc}(r) = \sin(\pi r)/(\pi r)$  is the only function that represents the Fourier-space box filter in real space and therefore does not add additional correlated noise to resampled data (see, e.g., Devillard 2000). Many algorithms have therefore been developed to find a computationally reasonable implementation of this function that does not have to be integrated over the whole dataset. The Lanczos functions, that cut off for distances larger than  $k$ , have been used in many applications:

$$L_k(r) = \begin{cases} \text{sinc}(r)\text{sinc}(r/k), & \text{if } |r| < k \\ 0, & \text{otherwise} \end{cases}.$$

This can be selected with the `lanczos` option, and  $k$  can be set with `ld`.

#### 4.6. Variances of the resampled datacubes

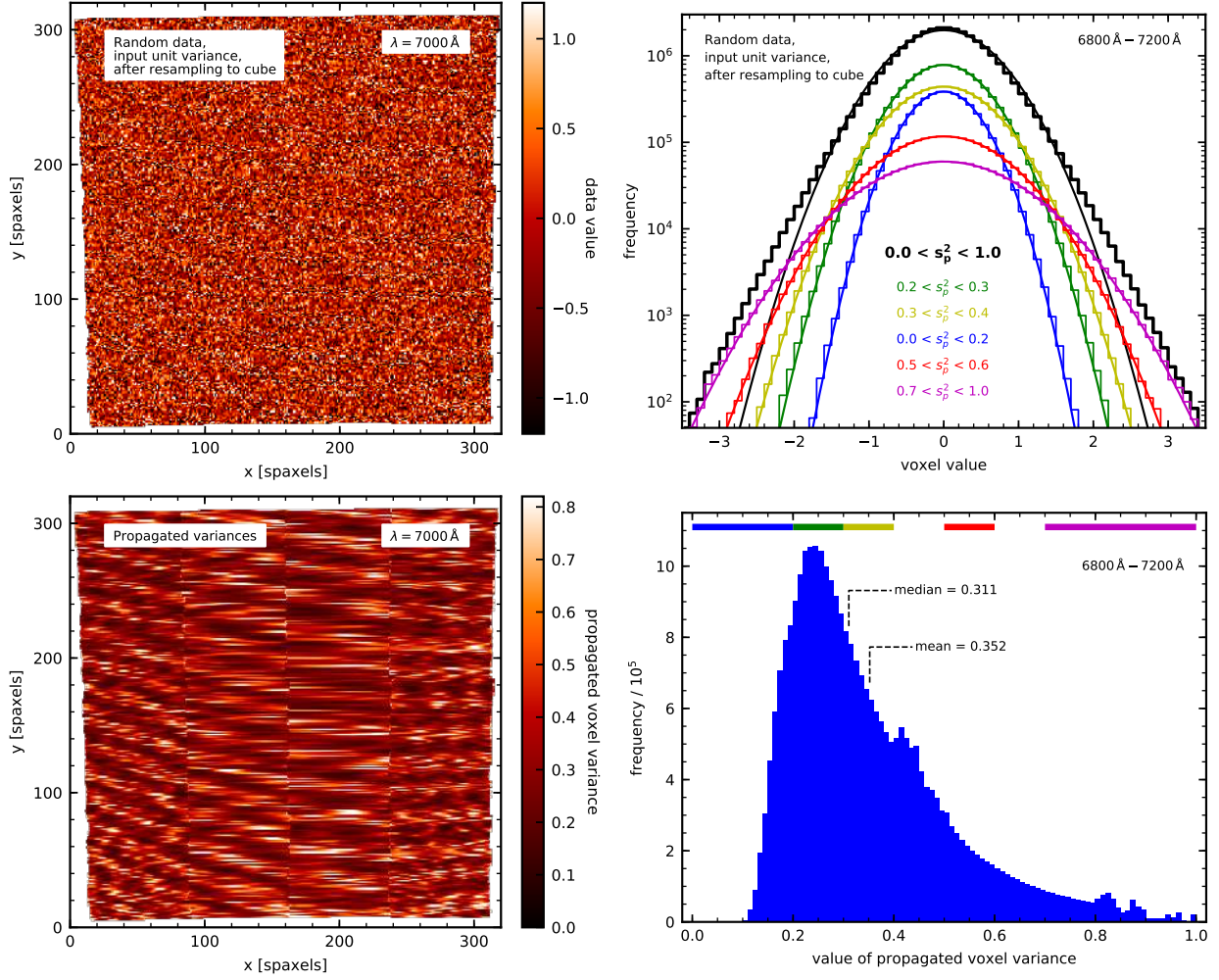
We already mentioned in Sect. 4.5 that the resampling step invariably introduces cross-talk between adjacent voxels (with the details depending on the actually adopted resampling algorithm). While the MUSE pipeline formally propagates the individual pixel errors through the resampling process, it is unavoidable that the real uncertainties of the data get partly diffused into off-diagonal covariance terms which are not recorded by the pipeline. This has three principal consequences: (i) the propagated variances  $s_p^2$  are systematically lower than the true variances  $\sigma^2$ ; (ii) the propagated variances show substantial but unphysical spatial variations; and (iii) the noise distribution of voxels becomes apparently non-Gaussian. In the following we discuss these quantitatively.

To isolate the effects on the variances caused by the resampling of a pixel table into a datacube we performed the following simple experiment: We created a MUSE pixel table filled with pure random numbers of zero mean and unit variance. We converted this pixel table into a datacube using the drizzle-like resampling option with default parameters (`pixfrac` of 0.8 in all directions). The top left panel of Fig. 9 shows one arbitrarily selected wavelength plane of this cube. A quick visual inspection of this image already reveals a pattern. In some regions the pixels appear to be spatially smoothed; in others they are largely unaffected. This variable smoothing pattern is a direct consequence of the non-trivial geometric mapping of the detector plane into a regularly gridded datacube. Voxels with projected locations falling between detector pixels inherit similar amounts of flux from multiple pixels implying a large covariance term, whereas voxels nearly congruent with pixels suffer from little such cross-talk. Figure 9 also shows that there is spatial coherence in these patterns, with strongly and weakly smoothed horizontal bands alternating in vertical direction.

The same structures are evident in the propagated variances  $s_p^2$  displayed in the bottom left panel of Fig. 9. Lower values of  $s_p^2$  correspond to heavier local smoothing of the voxels. Around each of the three vertical slicer stack transitions the patterns changes abruptly, but there are also drastic changes between slices within one stack, and even across a single slice, typically resulting from a non-zero tilt between detector rows and the datacube grid. For this reason the spatial coherence length is much shorter in the two outer slicer stacks which are more tilted. Additional variations are affected by the resampling in wavelength together with the non-linear dispersion relation. The amplitude of these variations are substantial, up to a factor  $\sim 6$  in the value of  $s_p^2$  even between adjacent voxels; we recall that all input data to this datacube have a variance of exactly  $\sigma^2 = 1$ . But a similar underlying pattern will be present in all real science data, only less visible because of the noisy nature of the variance estimates when using Eq. (7). These variations are purely geometric in origin and obviously imply no changes in the level of trustworthiness of the data (e.g., as an S/N). Because of the horizontal striping in  $s_p^2$  one cannot even define a small aperture over which the variations get averaged out.

The bottom right panel of Fig. 9 presents the actual histogram of propagated variances, revealing a very skewed distribution with a most frequent value of 0.24, implying an artificial increase in the apparent S/N value by a factor of 2; the median and mean are correspondingly higher. Only a tiny fraction of the voxels has  $s_p^2$  values close to 1. The smallest values of  $s_p^2$  correspond to the voxels suffering from the heaviest cross-talk. Since for resampling with the given setup each voxel can be inheriting flux from at most eight pixels, the propagated variance can be





**Fig. 9.** Behavior of the propagated variances  $s_p^2$  after resampling a pixel table with pure random data of unit variance into a datacube. *Upper left panel:* single wavelength plane of this cube; the same plane of the corresponding variance cube is shown below. *Upper right panel:* histograms of the resampled random data evaluated over a subcube of 320 planes ( $400 \text{ \AA}$ ) bandwidth, together with Gaussian fits to these histograms. Each of the colored curves represent a different subset of voxels selected by a specific range of the propagated variances in these voxels, as indicated by the labels. *Lower right panel:* histogram of the propagated variances, for the same subcube. The colored horizontal bar in the top indicates the selected variance ranges.

reduced by up to that number,  $s_p^2 \geq 0.125 \times \sigma^2$ , corresponding precisely to the lower bound in the histogram.

One further consequence of the variable scale of spatial (and spectral) smoothing is that the histogram of actual voxel values at any given wavelength deviates from a Gaussian even if the input data follow a perfect normal distribution, as can be seen in the upper right panel of Fig. 9. Not only is the histogram (thick black line; standard deviation of  $s_p = 0.593$ ) much narrower than the input distribution, it also shows extended wings that in real data might lead the user to suspect significant non-Gaussian errors. However, Fig. 9 demonstrates that these wings can be decomposed into a superposition of essentially perfect Gaussians when performing the histogram analysis on subsets of voxels within a narrow range of  $s_p^2$  values; each of these histograms corresponds to a different degree of local smoothing and therefore reduced variance.

We note that this does not mean that the propagated variances contain no useful information. In fact they describe the actual noise level per voxel accurately (apart from the limitation of Eq. (7)). In other words, any repeat experiment with identical resampling geometry would reveal random fluctuations in each voxel in accordance with the corresponding value of  $s_p^2$ . But as

the variations in adjacent voxels are correlated, the amplitude of any aperture-based quantity would be larger than formally expected if interpreting  $s_p^2$  as white noise variance. For this reason it also would not help to self-calibrate the variance level by measuring the actual voxel-to-voxel fluctuations in an observed datacube; these fluctuations will always be significantly lower than the real noise.

A relatively simple recipe to obtain realistic variances for background-limited observations would involve three steps:

1. Run a similar simulation as described in this subsection, but specifically for the chosen resampling setup. This needs to be done only once per setup.
2. At each wavelength, obtain the histogram of propagated variances in the observed target datacube and estimate, for example, its mean or median.
3. Scale that value by the inverse of the corresponding mean or median of  $s_p^2$  in the random numbers datacube, and replace the fluctuating variance plane by the resulting number as a constant.

This approach implicitly also solves the problem of the initially noisy variances. It is clearly limited to cases where the objects of interest are faint enough that their photons can be neglected

as contribution to the shot noise per pixel. An object model with defined simulated noise would have to be used for the case of a field where objects contribute significantly.

#### 4.7. Image reconstruction

Once a datacube is reconstructed, it is possible to integrate it over the wavelength direction to create an image of the FOV. By default, a white-light image is created, where each voxel of the cube is weighted equally, and only data outside the wavelength range 4650–9300 Å is discarded. If a filter function was given, for example for the Johnson *V* filter, then each voxel in the cube is weighted according to the filter transmission  $w_\lambda$  at its wavelength:

$$f_{\text{pixel}} = \frac{\sum_\lambda w_\lambda \Delta\lambda f_\lambda}{\sum_\lambda w_\lambda \Delta\lambda}. \quad (9)$$

For cubes with constant wavelength sampling this is simply

$$f_{\text{pixel}} = \frac{\sum_\lambda w_\lambda f_\lambda}{\sum_\lambda w_\lambda}. \quad (10)$$

#### 4.8. Correction of atmospheric refraction

The correction of atmospheric refraction is important for all spectroscopic data, especially for an instrument with a wavelength range as long as the one of MUSE. In WFM, the red and blue ends are shifted significantly (more than a spatial resolution element). In NFM, the instrument has a built-in atmospheric dispersion corrector (ADC), and in most cases no software correction appears to be necessary. However, the ADC was built to allow for up to 2 pixels of residual shift for zenith distances of 50°, so an empirical correction may be necessary (see below).

The algorithm in the MUSE pipeline was originally developed by Sandin et al. (2008) and implemented in the P3D software (Sandin et al. 2010). Using known atmospheric parameters for humidity, temperature, and pressure at the observatory, the refractive index of air is computed for a reference wavelength and all other wavelengths of the MUSE data. The relative shift between the data in both spatial directions is then computed depending on parallactic and position angle of the exposure. This shift is then applied to the coordinates of all pixels in the pixel table.

Computation of the refractive index uses the formulae from Filippenko (1982) by default. Following the refinement of Sandin et al. (2012), the MUSE pipeline also implemented different more accurate methods to calculate the refractive index of air based on the abovementioned atmospheric conditions at the time of the observation. These were taken from Owens (1967), Edlén (1966), Birch & Downs (1993), and Ciddor (1996). However, since it was shown that the Filippenko (1982) approach gives satisfactory results with MUSE data (see Weilbacher et al. 2015), it remains the default. For the typical case of an observation at airmass 1.5, shifts of about 1.2 occur at the blue end of the MUSE wavelength range. All four methods correct this shift equally well to below 0.04, far smaller than the MUSE sampling on the sky.

An additional step is implemented in the MUSE pipeline (parameter `darcheck`). It reconstructs a cube with nominal spatial but long (10 Å) wavelength pixels to improve the S/N per wavelength plane. It then runs a simple threshold-based object detection on the central plane of the cube. All objects detected are then centroided in each plane. A polynomial is then fitted to these measurements. If the cube was already corrected for atmospheric refraction effects, a quadratic polynomial is used, otherwise a fourth-order fit is done. This fit can

be used (setting `darcheck=correct`) either to correct the data for atmospheric refraction effects, if none of the above analytical methods was applied before, or to correct residual shifts after an analytical correction. Alternatively, one can use it (setting `darcheck=check`) to quantify any residual effects in the dataset in terms of maximum deviation from the reference wavelength. This empirical procedure is not activated by default since it takes a significant amount of extra time, and the accuracy of its operation depends on the science field in question. In the case of a single bright source in the field, correction to sub-pixel accuracy should be possible. This is the typical case for NFM observations, and can be used to correct for residuals caused by the ADC. In a more complex case or in fields without any directly visible continuum sources, a separate characterization outside the pipeline will be necessary.

#### 4.9. Flux calibration

When applying the response curve and the telluric corrections that were computed from the standard star (see Sect. 3.11) to science data, the calibration data (the response  $f_{\text{resp}}$  and the telluric correction  $f_{\text{tell}}$ , as well as the atmospheric extinction  $f_{\text{ext}}$ ) are linearly interpolated to the wavelength  $\lambda$  of each pixel and applied to the data in counts  $d_{\text{ct}}$  using

$$d_{\text{cal}}(\lambda) = \frac{10^{-0.4f_{\text{resp}}(\lambda)} 10^{0.4f_{\text{ext}}(\lambda)A} f_{\text{tell}}(\lambda)}{t_{\text{exp}}\Delta\lambda} d_{\text{ct}}(\lambda) \quad (11)$$

to yield the calibrated flux  $d_{\text{cal}}$  in each pixel in units of  $10^{-20} \text{ erg s}^{-1} \text{ cm}^{-2} \text{ \AA}^{-1}$ , where  $A$  is the effective airmass of the given exposure.

This procedure is used for science exposures, and implemented in the `muse_scipost` module. It is also used in the same way for offset sky exposures (in `muse_create_sky`) and can be applied to astrometric exposures as well (from `muse_astrometry`).

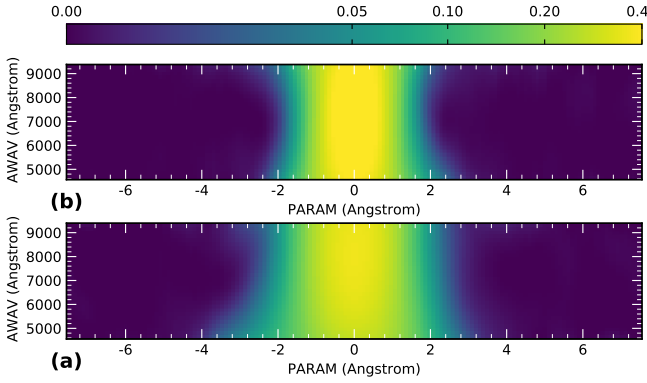
#### 4.10. Line-spread function

The LSF is used in the MUSE pipeline for the description of sky emission lines and their subtraction (see Sect. 3.9.3), and in the modeling of the laser-induced Raman lines (Sect. 3.10.1). Section 3.8 described how it is determined, here we provide a description of the two possible representations that are used in the various pipeline modules.

The interpolated LSF is saved in image form, where each image stores the LSF for a given slice in a given IFU of MUSE. The line profile includes the slit width of the instrument (determined by the height of the image slicer in the spectrographs) and the bin width (the CCD pixel size). The LSF from each IFU is stored in one file or FITS extension, thereby using the FITS cube format, with 48 pixels in the third axis. In Fig. 10 we show two representative slices, one with a narrow LSF (a slice in the middle of the FOV, also located in the middle of the CCD), and a wider LSF (a slice at the left edge of the CCD, and near the bottom of the MUSE field). Since this interpolated, empirical description provides a superior and much faster sky subtraction, it is used by default.

The alternative LSF description is a parameterization using damped Gauss-Hermite function of the form<sup>33</sup>

<sup>33</sup> Our definition was inspired by Zhao & Prada (1996), who used a modified Gauss-Hermite polynomial to parameterize line-of-sight velocity distributions.



**Fig. 10.** Interpolated image of the LSF for two MUSE slices, computed from the arc exposure set started on 2016-05-12T11:40:31: (a) LSF for slice 1 of IFU 1; (b) result for slice 22 in IFU 12. The value range is the same and uses arcsinh scaling. The horizontal axis shows the LSF direction, whereas the vertical axis is the MUSE wavelength range.

$$L(x) = e^{-\frac{x^2}{2w^2}} + e^{-\frac{x^2}{w^2}} \sum_{i=3}^6 k_i H_i\left(\frac{x}{w}\right),$$

with the coefficients  $k_i$  of the Hermitian polynomials  $H_i(x)$  and the LSF width  $w$  as slice and wavelength dependent fit parameters ( $x = \lambda - \lambda_0$  with  $\lambda_0$  as the wavelength of the emission line). This LSF is then analytically convolved with rectangular functions representing the slit width and the CCD binning. The LSF width  $w$  is parameterized with a quadratic dependency of the wavelength, the Hermitean coefficients  $k_i$  with linear function. This alternative LSF description was mainly used before MUSE was first tested on sky. Commissioning data then showed this method not to be robust enough and leave too strong residuals of the telluric emission, so the interpolating LSF above was developed, and since then used by default.

Within the MUSE pipeline, the LSF is only used for background modeling (sky and Raman lines). However, it can be used later in other tools for analysis if the sampling of the output cube is taken into account. For example, [Weilbacher et al. \(2018\)](#) use the FWHM of the averaged LSF as determined by the pipeline in their pPXF fits, using the log-spaced pixel sizes in wavelength direction to convolve the unbinned LSF. It should also be noted that the LSF is very similar between the four wide-field modes of MUSE, but has different shape and width for the narrow-field mode.

## 5. Implementation

Since the pipeline discussed in the present paper is written to serve as data reduction environment for an ESO instrument, it is implemented in the ESO software framework. It is written in pure C and uses the ESO Common Pipeline Library (CPL; [Banse et al. 2004](#); [ESO CPL Development Team 2014](#)) as its base library for all internal data structures. CPL in turn uses CFITSIO ([Pence 2010](#)), wcslib ([Calabretta 2011](#)), and FFTW ([Frigo & Johnson 2005](#)) for parts of its functionality.

This means that the data reduction modules are available as shared libraries that can be used as plugins with esorex ([ESO CPL Development Team 2015](#)) alone or in the Reflex environment ([Freudling et al. 2013](#)). While the latter makes MUSE reduction very easy and hides a lot of the complexity and data association behind a graphical user interface, the former forces the user to think about the relation between science data and calibrations, and allows scripting. A Python interface (`python-cpl`,

[Streicher & Weilbacher 2012](#)) can be used to start the modules as well<sup>34</sup>.

Each individual reduction module (also called a “recipe”) calls into the shared library to carry out its task. The user-facing side of the recipes and its basic C code, on the other hand, is generated from XML descriptions, which provide the possibility to create documentation as well. The XML interface was finally also used to integrate the MUSE pipeline into the MUSE-WISE data processing system ([Pizagno et al. 2012](#); [Vriend 2015](#)) that is based on the AstroWISE concept ([Valentijn et al. 2007](#)). This is used to process part of the data collected by the MUSE collaboration in its guaranteed observing time.

To allow the MUSE pipeline to efficiently handle the data processing on modern hardware, it was planned to be parallelized from the start. The basic processing part operates in parallel for the data from the 24 IFUs, and can therefore only make use of 24 cores. This can be run either as internal parallelization using OpenMP or parallelized externally using scripting to run multiple processes. The post-processing steps are always parallelized internally, using OpenMP loops, and can employ all the cores that the computing hardware offers. However, testing showed that not all steps benefit from parallel operation, so that operations that read data from disk or save files are serial. The same is true for operations involving concatenation of large data buffers in memory, for example when merging multiple exposures.

The pipeline code is Open Source (GPL v2) and distributed through the ESO pipeline website<sup>35</sup>. The releases of the MUSE consortium are code-identical, but up to v2.6.2 used a slightly different packaging<sup>36</sup>. The full data reduction cascade of the implementation, complete with the tags of the input and output files, and all relevant data reduction recipes described in this paper is shown in [Fig. 11](#).

## 6. Testing of the data quality

Testing the quality of the data reduction was central throughout the development of the MUSE pipeline. Before on-sky data became available in 2014, testing relied heavily on the comprehensive simulation of raw data provided by the Instrument Numerical Model (INM, [Jarno et al. 2010, 2012](#)). Especially developing complex procedures like the geometrical calibration and testing cosmic ray rejection or the combination of multiple exposures with offsets benefitted strongly from the availability of such data. A multitude of tests, too many to be reported here, were run for each data processing module using this simulated data to ensure readiness for on-sky operation. Since the data format, especially the metadata recorded in the FITS headers of the raw exposures that the MUSE pipeline now requires to be present, has changed since then, we cannot use these data any more for verification of the current software version.

Instead, we show a few cases here that are reproducible with the current publicly available datasets to show the performance of the pipeline processing.

### 6.1. Bias subtraction accuracy

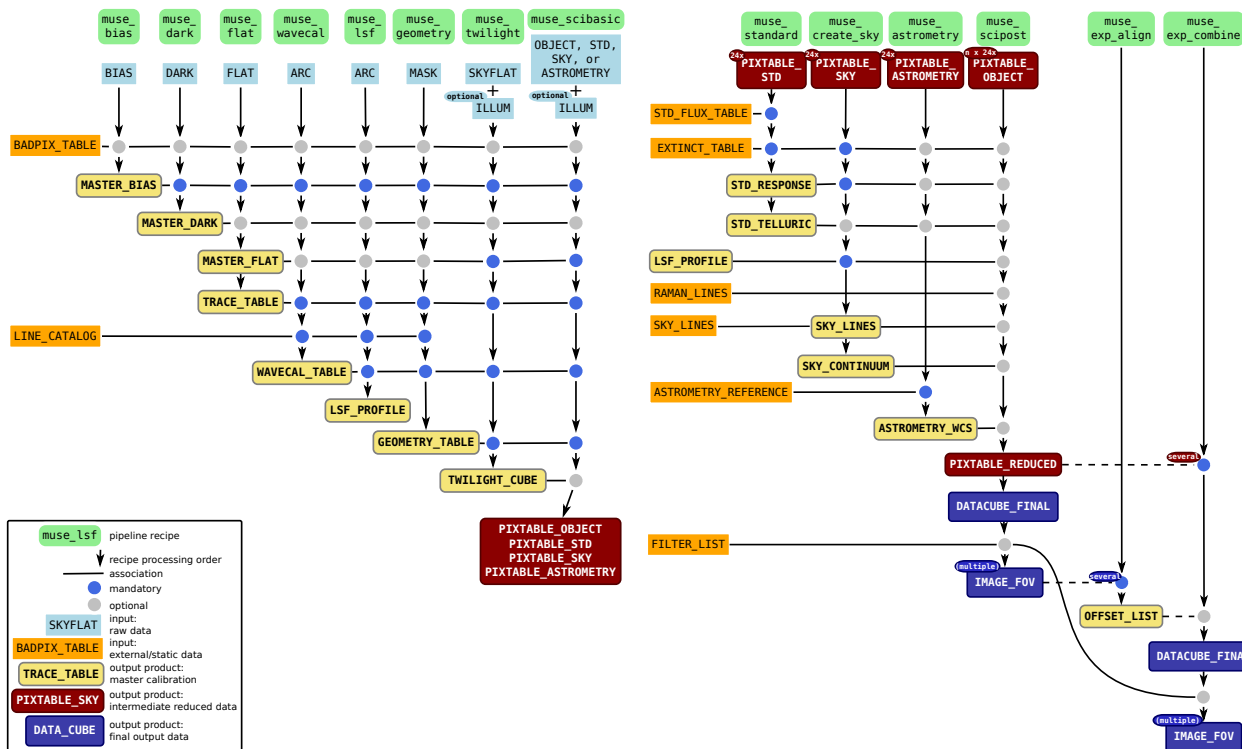
The quality of the bias subtraction, including overscan modeling, can be best judged by checking the flatness of darks. To

<sup>34</sup> It actually works for all ESO pipelines built around the plugin concept of the CPL.

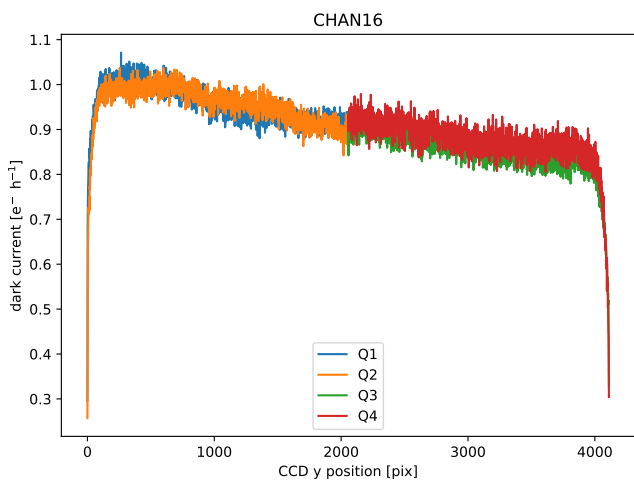
<sup>35</sup> <https://www.eso.org/sci/software/pipelines/muse/muse-pipe-recipes.html>

<sup>36</sup> <https://data.aip.de/projects/musepipeline.html>





**Fig. 11.** *Left*: basic processing recipes and products of the MUSE pipeline. *Right*: post-processing steps in the MUSE pipeline. Compared to Fig. 1 this shows all input and output files and includes all calibration modules.



**Fig. 12.** Dark current distribution in CCD “pallas” of MUSE channel 16, measured after bias subtraction, averaged over 1000 CCD columns.

be able to see differences in small fractions of an ADU, one needs to check the combination of many dark frames, meaning the combined master dark of a long series. Such a series was taken in 2018, between June 25 and Aug. 21, comprising 149 darks of 30 min exposure time each. All bias frames that were taken on the same days as the dark exposures (418 in total) were used to create master bias images for bias subtraction. The process used the default parameters of pipeline v2.8. A vertical cut through a typical combined master dark image is shown in Fig. 12. The vertical cut was computed as the average over 1000 CCD columns in the dark image, ignoring flagged pixels. The vertical gradient of the dark current (about  $0.15 e^- h^{-1}$  over 4000 pixels) seems to be a property of the e2v CCDs as used

in the MUSE instrument together with the ESO NGC electronics. The strong gradients in the first and last few pixels are edge effects that appear differently in bias and dark frames; these are not of relevance to science data, since these extreme CCD pixels are at wavelengths that are not used for normal analyses. In the case of the displayed dark of channel 16, a small jump (about  $0.05 e^- h^{-1}$ ) between the four CCD quadrants is visible. While the displayed jump is typical, for some of the other 23 CCDs it is not visible at all and for a few others the discontinuity is stronger. However, this discontinuity is undetectable without large-scale binning and it could also be caused by a horizontal gradient in the affected CCDs, an error in the determination of the CCD gain in a quadrant, or undetected bad pixels that deviate from the surrounding pixels in a subtle manner.

Any residual features due to bias subtraction seem to be  $\lesssim 0.1 e^- h^{-1}$ , so are below  $1.5 \times 10^{-21} \text{ erg s}^{-1} \text{ cm}^{-2} \text{ \AA}^{-1}$  for typical atmospheric transparency.

## 6.2. Wavelength solution

To check the wavelength solution, we use a sequence of interleaved arc sequences taken in the different instrument modes on 21 April 2018 between 18:06:05 and 20:37:01 UTC. Five exposures in each mode were taken after each other before changing the arc lamp, so that each full sequence of 15 exposures (three lamps, with five exposures per lamp, in five modes) was as close in time as possible. This was done to exclude any changes in flexure in the spectrographs between the modes, which might otherwise happen due to temperature changes.

For the test we derive the wavelength solution using the 15 arc exposures taken in non-AO extended mode (WFM-NOAO-E). We then use the arcs taken in AO extended mode (WFM-AO-E) as comparison. We average them at the CCD level (to save processing time), but reduce them as if they were science

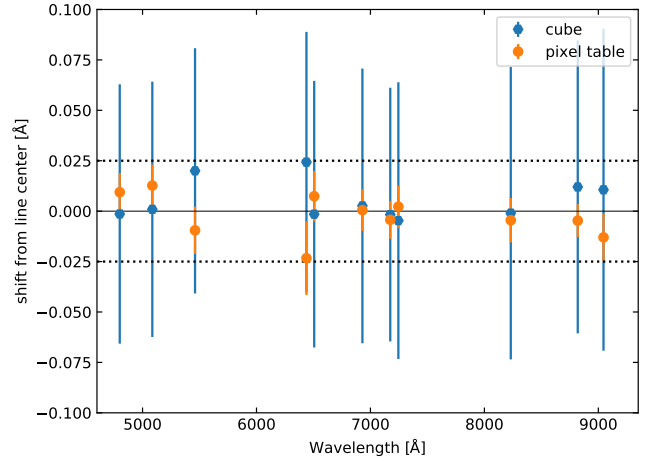
**Table 2.** Arc line wavelengths recovered from the pixel table.

$\lambda$ Å	Ion	Mean Å	Median Å	Std. dev. Å
4799.912	Cd I	0.0094	0.0083	0.0095
5085.822	Cd I	0.0127	0.0086	0.0104
5460.750	Hg I	-0.0095	-0.0110	0.0117
6438.470	Cd I	-0.0233	-0.0211	0.0183
6506.528	Ne I	0.0074	0.0033	0.0125
6929.467	Ne I	0.0005	-0.0013	0.0103
7173.938	Ne I	-0.0043	-0.0037	0.0094
7245.167	Ne I	0.0022	-0.0011	0.0105
8231.634	Xe I	-0.0045	-0.0082	0.0110
8819.410	Xe I	-0.0047	-0.0062	0.0082
9045.446	Xe I	-0.0130	-0.0131	0.0117

exposures, and let the pipeline create a combined pixel table and also reconstruct a cube. We ensure that corrections for atmospheric refraction, sky subtraction, astrometric calibration, and radial velocity are not carried out. We then measure the centers of the brightest and most isolated 11 arc emission lines using Gaussian fits, with spectra reconstructed directly from the pixel table with 0.1 Å bins for each slice of the instrument and for the whole FOV, and for each spaxel in the cube.

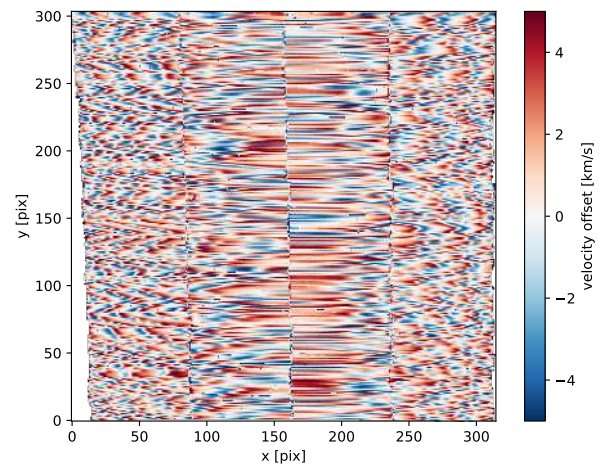
The measured line positions for the pixel table (see Table 2) represent the intrinsic wavelength calibration accuracy that can be achieved by the pipeline process. This is not degraded by the resampling to (typically) 1.25 Å bins of the cube. Similar levels of accuracy can be recovered during data analysis when using full-spectrum fitting to overcome the sampling problem of individual lines. The table lists the arc line wavelength in air, as given in the NIST database (Kramida et al. 2014) and used as wavelength reference in the MUSE pipeline; the ion it originated from; and the offset from the reference in the mean across the whole field, in the median across all slices, and the standard deviation measured over all 1152 MUSE slices. Given the width of the MUSE instrumental profile of about 2.5 Å, both the absolute and the relative accuracy reached is 100× better than the instrumental resolution for all arc lines, namely below 0.024 Å. This corresponds to a velocity accuracy of  $\sim 1 \text{ km s}^{-1}$ . In the central part of the MUSE wavelength range (6500–8500 Å) where the density of high S/N arc lines is higher and hence the polynomial fit is better constrained, even 0.01 Å (equivalent to  $\sim 0.4 \text{ km s}^{-1}$ ) are reached. This is in line with the velocity accuracy that was reached by fitting high S/N stellar spectra with solar metallicity, as was demonstrated by Kamann et al. (2018b). The orange data points in Fig. 13 represent these measurements based on the pixel table.

The results for the fits of individual lines at the cube level (see Table 3) are what a naive user can recover using fits to single (emission) lines. Since this folds the intrinsic wavelength calibration with the binning of the cube, the recovered line center depends on the sampling of the original pixel on the CCD with respect to the final sampling of the LSF in the cube. The values show that the average and median line centers can be measured with the same precision as for the pixel table. This is visually presented as the blue data points in Fig. 13. The relative offsets for all pixel positions (the standard deviation) are affected by the resampling, and only allow us to reconstruct the per-spaxel wavelength to 0.06–0.08 Å accuracy, corresponding to a  $1\sigma$  velocity precision of 2.5–4.0  $\text{km s}^{-1}$ . The spatial distribution

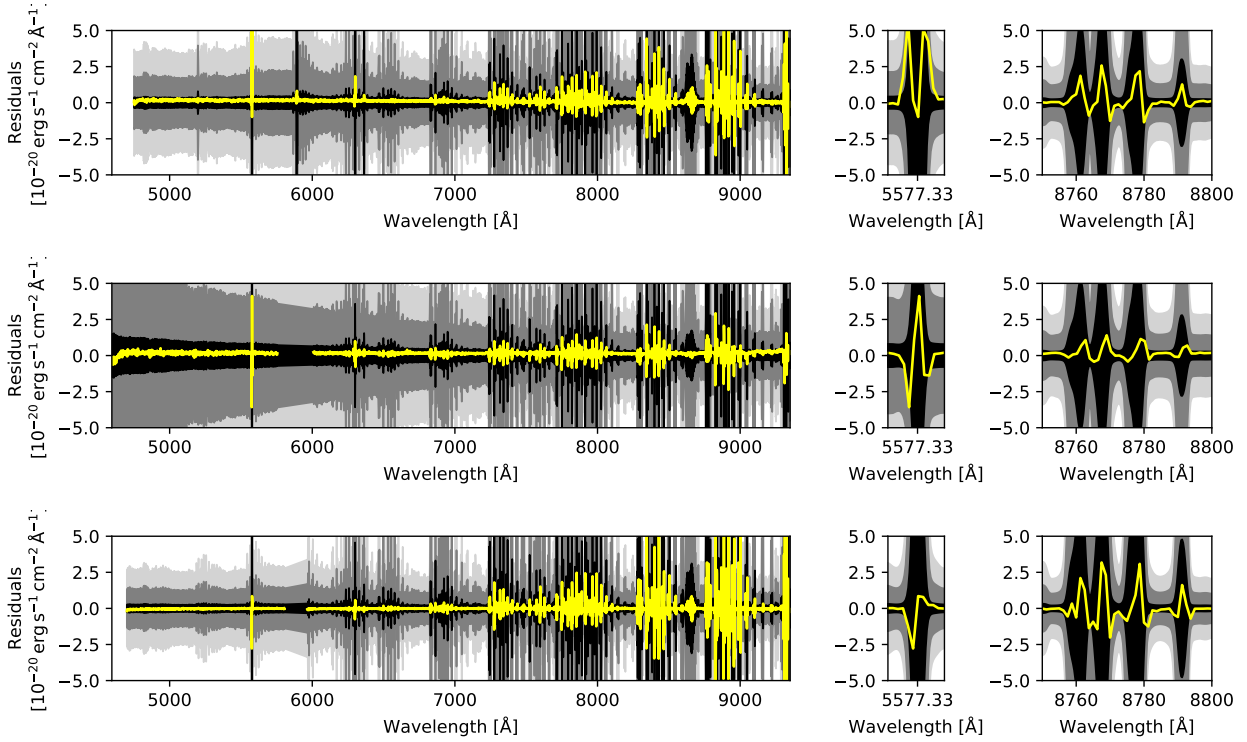

**Fig. 13.** Mean arc line positions recovered from the pixel table (orange) and datacube (blue). The error bars represent  $1\sigma$  standard deviations. The dotted horizontal lines mark the range of  $1/100$ th of the instrumental resolution. This is the visual representation of Tables 2 and 3.

**Table 3.** Arc line wavelengths recovered from the datacube.

$\lambda$ Å	Ion	Mean Å	Median Å	Std. dev. Å
4799.912	Cd I	-0.0014	-0.0032	0.0643
5085.822	Cd I	0.0009	-0.0005	0.0633
5460.750	Hg I	0.0200	0.0195	0.0608
6438.470	Cd I	0.0243	0.0243	0.0646
6506.528	Ne I	-0.0015	0.0006	0.0661
6929.467	Ne I	0.0026	0.0034	0.0681
7173.938	Ne I	-0.0017	-0.0001	0.0629
7245.167	Ne I	-0.0047	0.0019	0.0686
8231.634	Xe I	-0.0010	0.0025	0.0725
8819.410	Xe I	0.0120	0.0164	0.0726
9045.446	Xe I	0.0106	0.0131	0.0798


**Fig. 14.** Velocity offsets across the MUSE FOV as measured from the arc line Xe I 8819 on a reconstructed cube.

of this pattern is shown in Fig. 14 for the example of the Xe I line at 8819.410 Å. This effect can be mitigated in the analysis of astronomical objects, if the measurement of multiple lines of similar S/N can be combined.



**Fig. 15.** Sky background percentages (light gray: 10%, gray: 5%, black: 1%) and median residuals (yellow) in the three MUSE exposures: *top*: galaxy AM 1353-272 in WFM-NOAO-N mode on 2014-04-29T04:22:00 UTC, *middle*: tidal dwarf NGC 7252 NW in WFM-AO-E mode on 2017-07-16T09:13:11 UTC, *bottom*: a QSO in WFM-AO-N mode on 2018-02-15T05:08:06 UTC.

### 6.3. Sky subtraction residuals

To assess the quality of the sky subtraction performed by the MUSE pipeline we take publicly available MUSE data of three different instrument modes as examples, and reduce them in the standard way. The science post-processing module outputs the cube, a white-light image, and files to characterize the sky subtraction. We use the high S/N sky spectrum of the exposure (averaged over the sky regions of the exposure) to assess the original sky level at each wavelength. The residuals still present in the corresponding exposure are taken as the integrated spectrum of the cube. To exclude any object features, we used the median over the FOV (for targets that cover a small fraction of the field), or the median after masking the brightest 75% of the white-light image (for nearby galaxies).

We plot the median residuals of the cube as yellow lines in Fig. 15, where  $\pm 10\%$ ,  $\pm 5\%$ , and  $\pm 1\%$  level deviation from the original sky background are highlighted as areas shaded in light gray, dark gray, and black. We show the whole MUSE wavelength range in the left panels, zoomed-in on the small region around the strong line [O I]5577 in the middle panel, and an enlarged display of the range around the blue end of the OH 7–3 band. The MUSE pipeline was designed to reach a sky subtraction accuracy of at least 5% with a goal of 2% for wavelengths not affected by strong sky emission lines. These goals were clearly reached; the yellow lines are always within the dark gray 5% region, and in the continuum within the 1% range as well. Only for a few lines was the LSF not determined with enough precision. In those cases, the wings of the line residuals extend beyond this limit, to about 2% of the original sky level. This is visible in the [O I]5577 line in the middle panels in the two top rows in Fig. 15. While residuals for other lines are apparent, they do not extend beyond the 1% range.

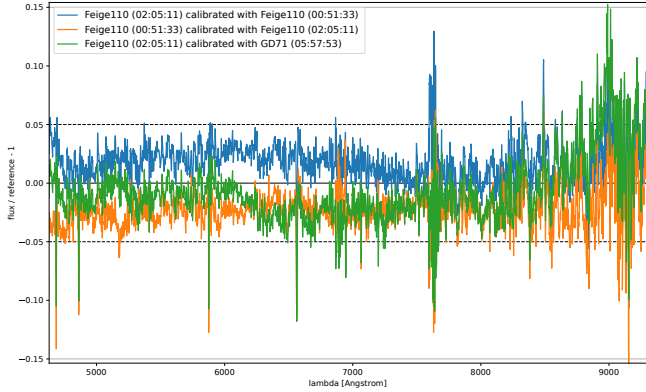
### 6.4. Flux calibration accuracy

The flux calibration accuracy of the pipeline reduction was already shown to be within 4–7% of fluxes measured by other instruments, depending on the emission line in question, and to vary by about 5% across the FOV of the instrument (Weilbacher et al. 2015). Here, we investigate how the accuracy of the flux calibration depends on the wavelength.

We reduced standard stars taken during the night of 22 November 2018 with the flux-calibration module (`muse_standard`), resulting in a response curve and the telluric correction spectrum. Then we treated these exposures as if they were science data, using the science module (`muse_scipost`), including the response and telluric correction derived from another standard star exposure. Since the night was classified photometric between 00:58 and 06:55 UTC, we did not expect strong atmospheric changes during the observing sequence. Then we used the same extraction method (the smoothed Moffat profile described in Sect. 3.11) to measure the resulting fluxes again, and computed the ratio of the resulting spectrum with respect to the reference spectrum. The residuals can be seen in Fig. 16 for the two exposures of the star Feige 110 calibrated with the other exposure of the same star and an exposure of GD 71 observed during the same night. In these datasets the residuals are  $\lesssim 2\%$  on average, with a standard deviation of 2.4% to 3.7%.

The instrument mode used for these exposures was the extended wide-field mode (WFM-NOAO-E) that incurs a second-order overlap in the red part. Since the white dwarfs used for flux calibration are very blue, this effect is particularly strong. Therefore, the deviations at wavelengths beyond  $\sim 8300$  Å can be larger. It is also obvious, that during the night the atmospheric





**Fig. 16.** Residuals in the flux calibration accuracy as a function of wavelength. Three different exposures of the standard star Feige 110 observed on 22 November 2018 are shown. The calibrations were done with another exposure of the same star and with GD 71 at different time of the same night. The UTC times of the individual exposures are given as labels.

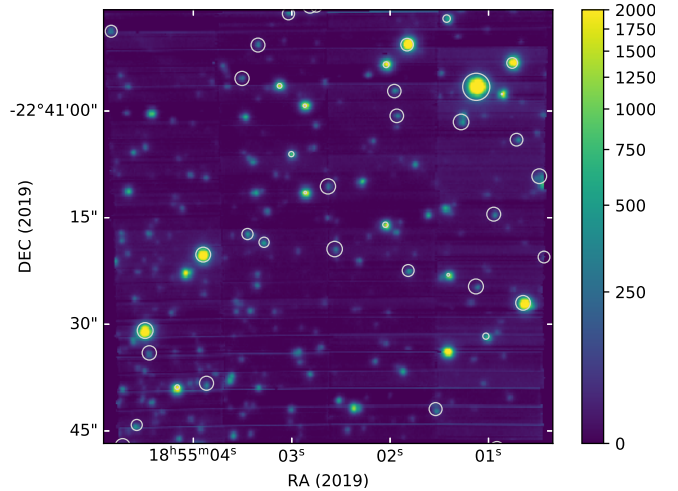
absorption in the telluric *A*- and *B*-bands changed, so that strong outliers and enhanced noise are visible around 6900 and 7650 Å.

To summarize, the flux calibration of MUSE data by the pipeline should be accurate to within 3–5% across the wavelength range of the instrument.

### 6.5. Astrometric precision

The combination of geometrical (see Sect. 3.6) and astrometric (Sect. 3.12) calibration is not meant to give an absolute world coordinate system (WCS), but should give a high-precision relative coordinate solution within the spatial extent of a MUSE cube. The astrometric calibration, derived from typically 50–100 stars within the MUSE field and repeated about once a month, provides a measure of the median residuals of the final WCS solution in both axes. The average over 36 such calibrations is  $0''.048 \pm 0''.018$  in the horizontal direction of a cube (in right ascension), and  $0''.027 \pm 0''.006$  vertically (in declination). The calibration is therefore thought to be better than 1/4 of a MUSE sampling element in WFM. In the high-resolution NFM that started operating only in 2019, the five existing calibrations show average residuals of 9 mas, so about 1/3 pixel, in both directions.

For the WFM fields, enough stars are available in the *Gaia* DR2 catalog (Lindegren et al. 2018) to carry out an independent check. We used the astrometric calibration created using a field in the outskirts of the globular cluster NGC 3201 (observed at 2019-10-29T08:38:37.91 UTC) together with the geometry sequence created from calibrations taken on 2019-10-26T09:20:36. We then reduced the data of a globular cluster field in NGC 6717 (observed on 2019-10-25T23:40:52.126 UTC) as a science exposure, using this calibration. The resulting image, integrated over a wavelength range with little sky contribution (7700–8100 Å), is shown in Fig. 17. There are 15 stars from the *Gaia* DR2 catalog within the MUSE field with magnitudes  $G < 17.65$  mag that have high-quality proper motions, and that can therefore be projected to the epoch of observations (2019.81640). We match these positions with those measured as Gaussian centroids using the ESO SKYCAT tool. When correcting for the zero-point offset, the 15 stars show a separation of  $0''.036 \pm 0''.018$  ( $\leq 1/5$  MUSE pixels) with standard deviations of  $0''.031$  in RA and  $0''.024$  in Dec, between the epoch-corrected *Gaia* positions and their centers in the MUSE data. Unfortunately, *Gaia* DR2 does not provide stars



**Fig. 17.** Image of a stellar field in the globular cluster NGC 6717. Positions from the *Gaia* DR2 catalog are shown as circles.

with high enough density to do the same check for small fields taken in the NFM.

This suggests that the quality of the astrometric solution computed by the pipeline gives a realistic representation of the real quality on sky, which corresponds to 1/4 of a spatial element of MUSE or better.

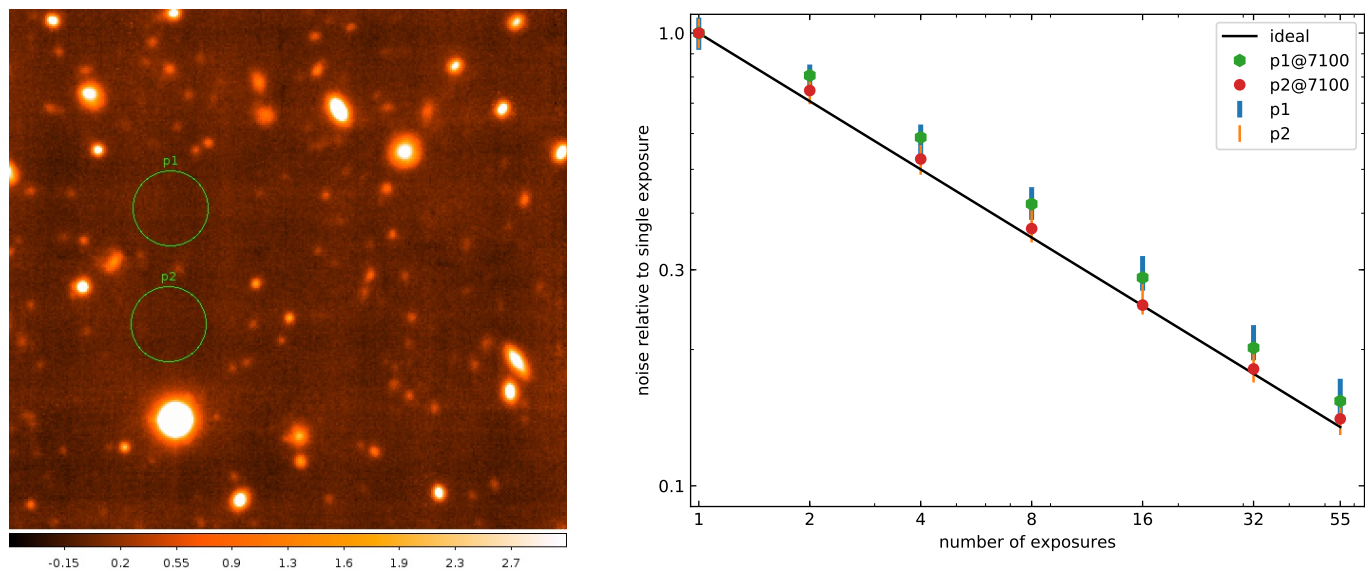
### 6.6. S/N behavior

In the ideal case a data reduction system should remove any systematics from the data so that the combination of a number of  $n$  exposures results in an S/N improvement of  $\sqrt{n}$ . For a deep MUSE dataset this was already tested in Bacon et al. (2015) to show that the combination of deep datasets reaches this ideal case to within a factor of 1.2. In that work, however, the authors used heavy post-processing of the individual cubes, and the exposures were not combined using the MUSE pipeline but externally at the cube level.

We therefore re-run that experiment with the MUSE dataset of the *Hubble* Deep Field South, using only the pipeline recipes and combining 1, 2, 4, 8, 16, 32, and 55 exposures at the pixel table level, without any exposure weighting using the `muse_exp_combine` module of the pipeline. Since we do not intend to use the data for science, we restrict the experiment to the wavelength range 7000–7200 Å, where a number of strong sky residuals and a clean wavelength range are present. We do not use the slice autocalibration (Sect. 3.10.2).

To check the S/N behavior with the number of exposures, we measure the noise in two circular apertures of 20 pixel radius (as displayed in Fig. 18 left), in regions where the white-light image (of Bacon et al. 2015) does not show any object to be present. We measure the standard deviation over these areas for all seven cubes and for each wavelength plane of the cubes. The mean and standard deviations of all wavelengths for a given cube are plotted in Fig. 18 (right) for both regions. The results for the wavelength plane at 7100 Å is plotted without error bars to show a particularly good case since this wavelength plane does not contain any significant residual of telluric emission.

Overall, the results show that the S/N improves almost as expected from the ideal curve, but in region p1 the systematics are slightly stronger. The final datapoint, for the cube that was combined using all 55 exposures of the field, is only a factor of 1.05 (1.17) higher than the theoretical expectation for region p2 (p1).



**Fig. 18.** *Left:* MUSE image of a region in the *Hubble* Deep Field South integrated over the wavelength range 7000–7200 Å. The two circular apertures that were used for measurements of the noise are shown as green circles. The displayed version shows the full depth, 55 exposures corresponding to 27.4 h. We selected the apertures to be located in a region without any detectable objects. *Right:* improvement of the noise measured in the final cube as a function of the number of exposures, measured as the standard deviation across two circular areas without any objects. See text for more details.

## 7. Conclusions and outlook

At the time of writing more than 200 publications have produced new science results using the MUSE instrument. All of them used the MUSE pipeline to process the raw data, or directly used datacubes made available by ESO through their Phase 3 process<sup>37</sup>. While most publications used the pipeline as described above (Krajnović et al. 2015; Guérou et al. 2016; Husser et al. 2016; Roth et al. 2018), others tried to improve sky subtraction or background uniformity in different ways (e.g., Bacon et al. 2017; Borisova et al. 2016; Urrutia et al. 2019; Fossati et al. 2019), often using the external ZAP-tool (Zurich Atmosphere Purge, Soto et al. 2016) to improve sky residuals. This tool was integrated into the ESO MUSE Reflex workflow to improve accessibility for users.

In hindsight we can say that there was no single key to success, but several ingredients helped to write the pipeline described in this paper: (i) We started prototyping early, shortly after the final properties of the instrument were fixed. (ii) In parallel to the reduction software a simulation of the raw data was created as well, which was finally used for a complete dry-run. (iii) The pipeline was actively used to process test data during the hardware tests by multiple people at several institutes. (iv) The main pipeline developer had a vested interest in making the instrument work, even after commissioning, as a member of the team exploiting the data collected in the guaranteed time. (v) We used multiple communication channels during different stages of the project to collect ideas for features and algorithms, to keep track of the implementation, and to track bugs. (vi) The software was published completely, without holding back consortium-internal parts, once testing of new features was completed.

While the pipeline works well to produce a plethora of science results on many different topics, some ideas exist that could improve its performance. Prototype code to compute and correct non-linearities of the data exists, and implementation in the public pipeline releases is planned and has the potential to improve

data accuracy for low and high levels of illumination. To improve the removal of telluric absorption, coupling with the MOLECFIT tool (Smette et al. 2015) could be possible. Likewise, sky emission lines could be treated by taking line groups from SKYCORR (Noll et al. 2014) instead of those distributed with the MUSE pipeline. To improve the accuracy with which the subsequent data analysis can be carried out, propagation of the LSF and limited propagation of covariances are further possibilities.

Some of the algorithms described in this paper would also be applicable for future integral-field instruments, especially for the ESO project HARMONI for the Extremely Large Telescope (Piqueras et al. 2016) and the possible BlueMUSE spectrograph for the VLT (Richard et al. 2019b).

*Acknowledgements.* The authors thank Joris Gerssen for contributions to the pipeline code at early stages of the development. We also thank all members of the MUSE collaboration for feedback on the pipeline during all stages of development, and especially Sebastian Kamann for preparing the astrometric reference tables. ESO staff, most notably Danuta Dobrzycka and Lodovico Coccato, helped to improve usage, capabilities, and algorithms since commissioning. We are also grateful for the anonymous referee who provided a very helpful and timely report on this long paper in difficult times. PMW, OS, and TU received support through BMBF Verbundforschung (project MUSE-AO, grant 05A14BAC) to develop the pipeline, PMW and OS were further supported by the MUSE-NFM project (grant 05A17BAA). We thank the developers of the Image Reduction and Analysis Facility (IRAF, Tody 1986), as ideas from some of its tasks were used in the design of the MUSE pipeline.

## References

- Allington-Smith, J., Murray, G., Content, R., et al. 2002, *PASP*, **114**, 892
- Bacon, R., & Monnet, G. 2017, *Optical 3D-Spectroscopy for Astronomy* (Weinheim: Wiley-VCH)
- Bacon, R., Copin, Y., Monnet, G., et al. 2001, *MNRAS*, **326**, 23
- Bacon, R., Accardo, M., Adjali, L., et al. 2010, in *Ground-based and Airborne Instrumentation for Astronomy III*, Proc. SPIE, 7735
- Bacon, R., Vernet, J., Borisova, E., et al. 2014, *The Messenger*, **157**, 13
- Bacon, R., Brinchmann, J., Richard, J., et al. 2015, *A&A*, **575**, A75
- Bacon, R., Piqueras, L., Conseil, S., Richard, J., & Shepherd, M. 2016, *Astrophysics Source Code Library* [record ascl:1611.003]
- Bacon, R., Conseil, S., Mary, D., et al. 2017, *A&A*, **608**, A1

<sup>37</sup> <https://www.eso.org/sci/observing/phase3.html>

- Banase, K., Ballester, P., Izzo, C., et al. 2004, in *Astronomical Data Analysis Software and Systems XIII*, eds. F. Ochsenbein, M. G. Allen, & D. Egret, *ASP Conf. Ser.*, 314, 392
- Birch, K. P., & Downs, M. J. 1993, *Metrologia*, 30, 155
- Borisova, E., Cantalupo, S., Lilly, S. J., et al. 2016, *ApJ*, 831, 39
- Calabretta, M. R. 2011, *Astrophysics Source Code Library* [record ascl:1108.003]
- Calabretta, M. R., & Greisen, E. W. 2002, *A&A*, 395, 1077
- Ciddor, P. E. 1996, *Appl. Opt.*, 35, 1566
- Cosby, P. C., Sharpee, B. D., Slinger, T. G., Huestis, D. L., & Hanuschik, R. W. 2006, *J. Geophys. Res. (Space Phys.)*, 111, A12307
- Davies, R. I. 2007, *MNRAS*, 375, 1099
- Devillard, N. 2000, *The Messenger*, 100, 48
- Edlén, B. 1966, *Metrologia*, 2, 71
- ESO CPL Development Team 2014, *Astrophysics Source Code Library* [record ascl:1402.010]
- ESO CPL Development Team 2015, *Astrophysics Source Code Library* [record ascl:1504.003]
- Fensch, J., Duc, P.-A., Weibacher, P. M., Boquien, M., & Zackrisson, E. 2016, *A&A*, 585, A79
- Fétick, R. J. L., Fusco, T., Neichel, B., et al. 2019, *A&A*, 628, A99
- Filippenko, A. V. 1982, *PASP*, 94, 715
- Fossati, M., Fumagalli, M., Lofthouse, E. K., et al. 2019, *MNRAS*, 490, 1451
- Freudling, W., Romaniello, M., Bramich, D. M., et al. 2013, *A&A*, 559, A96
- Frigo, M., & Johnson, S. G. 2005, *Proc. IEEE*, 93, 216
- Fruchter, A., Sosey, M., Hack, W., et al. 2009, *The MultiDrizzle Handbook, Version 3.0* (Baltimore: STScI)
- Gössl, C. A., & Riffeser, A. 2002, *A&A*, 381, 1095
- Guérou, A., Emsellem, E., Krajnović, D., et al. 2016, *A&A*, 591, A143
- Haffert, S. Y., Bohn, A. J., de Boer, J., et al. 2019, *Nat. Astron.*, 3, 749
- Herenz, E. C., & Wisotzki, L. 2017, *A&A*, 602, A111
- Heyl, J. S. 2013, *MNRAS*, 433, 935
- Howell, S. B. 2006, *Handbook of CCD Astronomy* (Cambridge: Cambridge University Press), 5
- Husemann, B., Kamann, S., Sandin, C., et al. 2012, *A&A*, 545, A137
- Husser, T.-O., Kamann, S., Dreizler, S., et al. 2016, *A&A*, 588, A148
- Irwin, P. G. J., Bowles, N., Braude, A. S., Garland, R., & Calcutt, S. 2018, *Icarus*, 302, 426
- Irwin, P. G. J., Toledo, D., Braude, A. S., et al. 2019, *Icarus*, 331, 69
- Izzo, C., Jung, Y., & Ballester, P. 2008, in *2007 ESO Instrument Calibration Workshop*, eds. A. Kaufer, & F. Kerber, 191
- Izzo, C., de Bilbao, L., & Larsen, J. M. 2016, *FORS Pipeline User Manual, Issue 5.3* (ESO)
- Jarno, A., Bacon, R., Ferruit, P., et al. 2010, in *Modeling, Systems Engineering, and Project Management for Astronomy IV*, Proc. SPIE, 7738
- Jarno, A., Bacon, R., Pécontal-Rousset, A., Streicher, O., & Weibacher, P. 2012, in *Software and Cyberinfrastructure for Astronomy II*, Proc. SPIE, 8451
- Kamann, S., Wisotzki, L., & Roth, M. M. 2013, *A&A*, 549, A71
- Kamann, S., Bastian, N., Husser, T. O., et al. 2018a, *MNRAS*, 480, 1689
- Kamann, S., Husser, T. O., Dreizler, S., et al. 2018b, *MNRAS*, 473, 5591
- Kelson, D. D. 2003, *PASP*, 115, 688
- Kelz, A., Bauer, S. M., Biswas, I., et al. 2010, in *Ground-based and Airborne Instrumentation for Astronomy III*, Proc. SPIE, 7735
- Kelz, A., Bauer, S. M., Hahn, T., et al. 2012, in *Ground-based and Airborne Instrumentation for Astronomy IV*, Proc. SPIE, 8446
- Kissler-Patig, M., Copin, Y., Ferruit, R., Pécontal-Rousset, A., & Roth, M. M. 2003, *Euro3D Data Format – Format Definition, Technical Report, Issue 1.2* (AIP)
- Kissler-Patig, M., Copin, Y., Ferruit, P., Pécontal-Rousset, A., & Roth, M. M. 2004, *Astron. Nachr.*, 325, 159
- Knapen, J. H., Comerón, S., & Seidel, M. K. 2019, *A&A*, 621, L5
- Krajnović, D., Weibacher, P. M., Urrutia, T., et al. 2015, *MNRAS*, 452, 2
- Kramida, A., Ralchenko, Yu., Reader, J., & and NIST ASD Team 2014, *NIST Atomic Spectra Database (v5.2)*, <http://physics.nist.gov/asd>
- Le Fèvre, O., Saisse, M., Mancini, D., et al. 2003, in *Instrument Design and Performance for Optical/Infrared Ground-based Telescopes*, eds. M. Iye, & A. F. M. Moorwood, *Proc. SPIE*, 4841, 1670
- Lindgren, L., Hernández, J., Bombrun, A., et al. 2018, *A&A*, 616, A2
- McLeod, A. F., Dale, J. E., Ginsburg, A., et al. 2015, *MNRAS*, 450, 1057
- Moffat, A. F. J. 1969, *A&A*, 3, 455
- Monreal-Ibero, A., Roth, M. M., Schönberner, D., Steffen, M., & Böhm, P. 2005, *ApJ*, 628, L139
- Monreal-Ibero, A., Weibacher, P. M., Wendt, M., et al. 2015, *A&A*, 576, L3
- Mulas, G., Modigliani, A., Porceddu, I., & Damiani, F. 2002, in *Observatory Operations to Optimize Scientific Return III*, ed. P. J. Quinn, *Proc. SPIE*, 4844, 310
- Noll, S., Kausch, W., Kimeswenger, S., et al. 2014, *A&A*, 567, A25
- Oberti, S., Kolb, J., Madec, P.-Y., et al. 2018, *Proc. SPIE*, 10703, 107031G
- Owens, J. C. 1967, *Appl. Opt.*, 6, 51
- Pasquini, L., Avila, G., Blecha, A., et al. 2002, *The Messenger*, 110, 1
- Patrício, V., Richard, J., Carton, D., et al. 2018, *MNRAS*, 477, 18
- Pence, W. D. 2010, *Astrophysics Source Code Library* [record ascl:1010.001]
- Piqueras, L., Jarno, A., Pécontal-Rousset, A., et al. 2016, in *Modeling, Systems Engineering, and Project Management for Astronomy VI*, Proc. SPIE, 9911, 99111Z
- Piqueras, L., Conseil, S., Shepherd, M., et al. 2019, in *Astronomical Data Analysis Software and Systems XXVI*, eds. M. Molinaro, K. Shortridge, & F. Pasian, *ASP Conf. Ser.*, 521, 545
- Pizagno, J., Streicher, O., & Vriend, W. J. 2012, in *ADASS XXI*, eds. P. Ballester, D. Egret, & N. P. F. Lorente, *ASP Conf. Ser.*, 461, 557
- Psych, W. 2004, *PASP*, 116, 148
- Renka, R. 1988, *ACM Trans. Math. Softw.*, 14, 139
- Richard, J., Patrício, V., Martínez, J., et al. 2015, *MNRAS*, 446, L16
- Richard, J., Bacon, R., Vernet, J., et al. 2019a, *MUSE User Manual, ESO-261650, v10.4* (ESO)
- Richard, J., Bacon, R., Blaizot, J., et al. 2019b, *ArXiv e-prints* [arXiv:1906.01657]
- Roth, M. M. 2006, *New Astron. Rev.*, 50, 252
- Roth, M. M., Kelz, A., Fechner, T., et al. 2005, *PASP*, 117, 620
- Roth, M. M., Sandin, C., Kamann, S., et al. 2018, *A&A*, 618, A3
- Sandin, C., Schönberner, D., Roth, M. M., et al. 2008, *A&A*, 486, 545
- Sandin, C., Becker, T., Roth, M. M., et al. 2010, *A&A*, 515, A35
- Sandin, C., Weibacher, P., Tabataba-Vakili, F., Kamann, S., & Streicher, O. 2012, in *Software and Cyberinfrastructure for Astronomy II*, Proc. SPIE, 8451
- Schmidt, K. B., Wisotzki, L., Urrutia, T., et al. 2019, *A&A*, 628, A91
- Scott, N., van de Sande, J., Croom, S. M., et al. 2018, *MNRAS*, 481, 2299
- Simon, J. L., Bretagnon, P., Chapront, J., et al. 1994, *A&A*, 282, 663
- Smette, A., Sana, H., Noll, S., et al. 2015, *A&A*, 576, A77
- Soto, K. T., Lilly, S. J., Bacon, R., Richard, J., & Conseil, S. 2016, *MNRAS*, 458, 3210
- Stetson, P. B. 1987, *PASP*, 99, 191
- Strassmeier, K. G., Ilyin, I., Järvinen, A., et al. 2015, *Astron. Nachr.*, 336, 324
- Streicher, O., & Weibacher, P. M. 2012, in *ADASS XXI*, eds. P. Ballester, D. Egret, & N. P. F. Lorente, *ASP Conf. Ser.*, 461, 853
- Streicher, O., Weibacher, P. M., Bacon, R., & Jarno, A. 2011, in *ADASS XX*, eds. I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots, *ASP Conf. Ser.*, 442, 257
- Ströbele, S., La Penna, P., Arsenault, R., et al. 2012, in *Adaptive Optics Systems III*, Proc. SPIE, 8447
- Tody, D. 1986, in *Proc. SPIE*, ed. D. L. Crawford, 627, 733
- Tody, D. 1993, in *ADASS II*, eds. R. J. Hanisch, R. J. V. Brissenden, & J. Barnes, *ASP Conf. Ser.*, 52, 173
- Urrutia, T., Wisotzki, L., Kerutt, J., et al. 2019, *A&A*, 624, A141
- Valentijn, E. A., McFarland, J. P., Snigula, J., et al. 2007, in *Astronomical Data Analysis Software and Systems XVI*, eds. R. A. Shaw, F. Hill, & D. J. Bell, *ASP Conf. Ser.*, 376, 491
- van Breukelen, C., Jarvis, M. J., & Venemans, B. P. 2005, *MNRAS*, 359, 895
- van der Loo, M. P. J., & Groenboom, G. C. 2007, *J. Comp. Phys.*, 126, 114314
- van Dokkum, P. G. 2001, *PASP*, 113, 1420
- Vogt, F. P. A., Bonaccini Calia, D., Hackenberg, W., et al. 2017, *Phys. Rev. X*, 7, 021044
- Vogt, F. P. A., Kerber, F., Mehner, A., et al. 2019, *Phys. Rev. Lett.*, 123, 061101
- Vriend, W. J. 2015, *Science Operations 2015: Science Data Management – An ESO/ESA Workshop*, 1
- Walsh, J. R. 2004, *Astron. Nachr.*, 325, 83
- Weibacher, P. M., Gerssen, J., Roth, M. M., Böhm, P., & Pécontal-Rousset, A. 2009, in *ADASS XVIII*, eds. D. A. Bohlender, D. Durand, & P. Dowler, *ASP Conf. Ser.*, 411, 159
- Weibacher, P. M., Streicher, O., Urrutia, T., et al. 2012, in *Software and Cyberinfrastructure for Astronomy II*, Proc. SPIE, 8451
- Weibacher, P. M., Streicher, O., Urrutia, T., et al. 2014, in *Astronomical Data Analysis Software and Systems XXIII*, eds. N. Manset, & P. Forshay, *ASP Conf. Ser.*, 485, 451
- Weibacher, P. M., Monreal-Ibero, A., Kollatschny, W., et al. 2015, *A&A*, 582, A114
- Weibacher, P. M., Monreal-Ibero, A., Verhamme, A., et al. 2018, *A&A*, 611, A95, Paper I
- Wisotzki, L., Becker, T., Christensen, L., et al. 2003, *A&A*, 408, 455
- Wisotzki, L., Bacon, R., Blaizot, J., et al. 2016, *A&A*, 587, A98
- Wisotzki, L., Bacon, R., Brinchmann, J., et al. 2018, *Nature*, 562, 229
- Zanichelli, A., Garilli, B., Scodreggio, M., et al. 2005, *PASP*, 117, 1271
- Zhao, H., & Prada, F. 1996, *MNRAS*, 282, 1223



## Appendix A: Instrument modes

Table A.1 lists all five modes of the MUSE instrument and their properties.

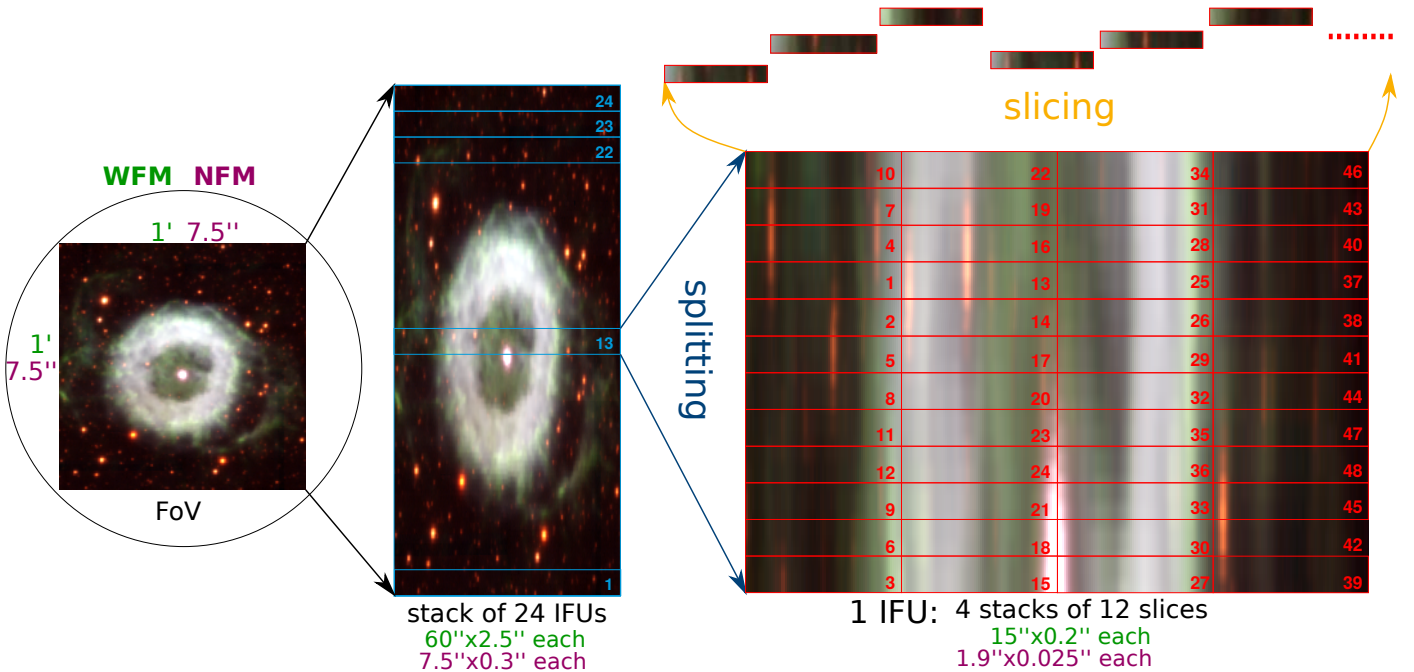
**Table A.1.** MUSE instrument modes and nominal properties.

Mode <sup>(a)</sup>	FOV	Sampling	Wavelength <sup>(b)</sup>	AO	Filter <sup>(b)</sup>
WFM-NOAO-N	1' × 1'	0":2	4750–9350 Å	No	2nd order (<4750 Å)
WFM-NOAO-E	1' × 1'	0":2	4600–9350 Å	No	None
WFM-AO-N	1' × 1'	0":2	4750–9350 Å	GLAO	Blocking (5800–5970 Å), 2nd order (<4750 Å)
WFM-AO-E	1' × 1'	0":2	4600–9350 Å	GLAO	Blocking (5750–6010 Å)
NFM-AO-N	7":5 × 7":5	0":025	4750–9350 Å	LTAO	Blocking <sup>(b)</sup> (5780–6050 Å), 2nd order (<4750 Å)

**Notes.** <sup>(a)</sup>-N stands for nominal, -E for extended wavelength range. <sup>(b)</sup>NaD blocking filter in the GALACSI AO system.

## Appendix B: Instrument layout

Figure B.1 shows a sketch of the instrument layout, and how the field is distributed over the 24 IFUs and the 48 slices per IFU.



**Fig. B.1.** Layout of the MUSE instrument, from the focal plane of the VLT (*left*) to the individual slices (*top right*). The light from the slices is subsequently dispersed and recorded on the CCD, and there forms the step-pattern visible in Fig. 2. The sizes of the different elements are given as nominal values (green: for WFM, violet: for NFM), the actual sizes are slightly different and change across the field. Overlaid on the images are the channel numbers (on the IFU stack) and the numbers of the slices as counted left to right on the raw data images (on the slicer stack). The example image is a color picture of the planetary nebula NGC 6369, as observed with MUSE in WFM-AO-N on 15 July 2017.