ISSN 1990-3413, Astrophysical Bulletin, 2022, Vol. 77, No. 3, pp. 334–346. © Pleiades Publishing, Ltd., 2022.
Russian Text © The Author(s), 2022, published in Astrofizicheskii Byulleten', 2022, Vol. 77, No. 3, pp. 369–382.

Pipeline Reduction of Long-Slit Spectra Obtained with the SALT Telescope

A. V. Kniazev[a, b, c, *]

[a] *South African Astronomical Observatory, Cape Town, 7935 South Africa*
[b] *Southern African Large Telescope, Cape Town, 7935 South Africa*
[c] *Sternberg Astronomical Institute, Moscow State University, Moscow, 119234 Russia*
**e-mail: a.kniazev@saao.nrf.ac.za*

Received September 6, 2021; revised February 17, 2022; accepted February 18, 2022

Abstract—This paper presents a description of the pipeline reduction of low and medium resolution spectral data obtained in the long-slit mode of the multi-functional Robert Stobie Spectrograph (RSS), mounted on the Southern African Large Telescope (SALT). The paper describes features of RSS that result in the creation of the pipeline. The main ideas of the developed pipeline are presented together with a general description of the pipeline and detailed description of its steps. The flow diagrams of all the pipeline steps are shown as well.

Keywords: methods: data analysis, software: development

DOI: 10.1134/S1990341322030075

1. INTRODUCTION

In Kniazev et al. (2019) the author described the Southern African Large Telescope (hereafter—SALT; Buckley et al., 2006; O'Donoghue et al., 2006) and its peculiarities. Because of its design features, SALT is mainly a spectral telescope. It is currently equipped with two spectrographs:

(1) the Robert Stobie Spectrograph (RSS, R = 300–12000) (Burgh et al., 2003; Kobulnicky et al., 2003), which is a long-slit spectrograph of low to medium spectral resolution; and

(2) the High Resolution Echelle Spectrograph (HRS, R = 15000–65000) Barnes et al. (2008); Bramall et al. (2012, 2010); Crause et al. (2014).

During SALT observations it is possible to select either of these spectrographs and change the mode of observation within about one minute. All SALT observations are carried out in service mode by professional astronomers.

The RSS spectrograph has a large set of observational modes:

(1) long-slit spectroscopy;

(2) multi-object spectroscopy;

(3) imaging in narrow–band filters;

(4) Fabry–Perot imaging;

(5) long-slit spectropolarimetry;

(6) high time resolution spectroscopy.

Of these available modes, the most commonly used is that of long-slit spectroscopy, which is chosen for 90% of the observational time. A feature of this mode is its large number of possible spectral configurations: there is a set of six grisms that are always available for observation, and each grism can be used in several dozen positions, determining the spectral range used (with the exception of the grism of the lowest spectral resolution, which covers the entire range). Altogether, there are several hundred possible grism positions. In addition to this, there are six spectral lamps to obtain comparison spectra (Ar, CuAr, Xe, ThAr, Ne, HgAr), two read-out speeds of the CCD mosaic, which consists of three CCDs and six ADCs (Analog-to-digital converters; two for each CCD), and two ADU operating levels, as well as the ability to bin the entire mosaic from values of 1 × 1 to 9 × 9 and a set of seven standard spectral slits[1]. Thus, the number of possible spectral and detector configurations turns out to be very large. However, even this large number of variants is not the final one. Every major technical shutdown of a spectrograph and/or its detector, or a technical problem resulting in re-assembly of the detector results in small change in the geometry of the mosaic. Of course, after such a situation, the support teams necessarily conduct a set of standard tests and redefine the set of parameters embedded in the primary data reduction pipeline (see Section 4). The parameters of the two-dimensional function responsible for the wavelength calibration for each specific spectral configuration might vary slightly after that, so this needs to be somehow tracked as well.

[1] http://pysalt.salt.ac.za/proposal_calls/current/ProposalCall.html.

At the same time the spectral reduction of RSS long-slit data is a very standard procedure and could be done using techniques implemented either in MIDAS or IRAF/PyRAF, or by using other standard packages for processing long-slit spectra.

The author has been working at SALT since its official inauguration and has extensive experience with all the SALT spectral instruments. About fifteen hundred long-slit RSS spectra with different spectral configurations have been obtained for the author's fifty SALT RSS proposals during the last sixteen years. For that reason the problem of fast and uniform RSS data reduction appeared for the author long time ago. Of course, the pipeline did not always exist in the form as presented in this paper, but was modified and supplemented along with the author's increasing experience with the peculiarities of RSS spectral data (Kniazev, 2006)[2].

This article is organized as follows: Section 2 summarizes the basic principles used in the creation of the described RSS pipeline. Sections 3 and 4 describe the RSS pipeline itself. Section 5 describes the structure of the software packages used in this pipeline.

2. PRINCIPLE OF IMPLEMENTATION OF THE SPECTRAL PIPELINE

The current version of the RSS spectral data pipeline was created by the author using all the software packages and tools he was familiar with and that seemed for him the most efficient[3]:

(1) The bash shell language available with all UNIX systems was used for the high level scripts and for the programming of different scenarios.

(2) The lower level spectral processing itself is written mainly using either IRAF or PyRAF, depending on the need for interactive mode. Tasks from two packages were mainly used longslit and apextract. The author believes that the implementation of two-dimensional spectral processing in IRAF/PyRAF is the most successful, complete and covers all possible features for the processing of spectra obtained in long-slit mode.

(3) MCL (the MIDAS Command Language) has also been used if the implementation in IRAF/PyRAF seemed to the author not efficient enough in the sense defined above.

(4) Since the high level programming language is the bash shell, the parameters of all IRAF/PyRAF and MIDAS programs and procedures are defined in a text configuration file, which has a default name, but another name could be used when called. All parameters are defined in the shell command language variable format as "Parameter=value" and can also be overridden when called from the command line. All this makes it possible to form different scenarios for the pipeline processing.

(5) Many spectral processing steps have interactive and non-interactive (automatic) modes. The choice of one or the other mode is determined by a parameter in the configuration file, which is automatically recalled at the beginning of each procedure and also has the above format, "Parameter=value", which makes it easy to translate all the parameters into shell variables.

(6) The main program allows one to process the spectral data obtained with RSS for a particular observing program, or for all observing programs of a particular observation night, or for any consecutive time intervals (starting from a specific date and ending on a specific date).

3. THE INTERACTIVE AND NON-INTERACTIVE MODES OF PROCESSING AND THE SALT SPECTRAL DATABASE

The implementation of the fully automatic data reduction of all possible RSS spectral configurations is a very complex task. For that reason the author implemented a much simpler procedure wherein each newly encountered spectral configuration requires the interactive identification of the reference spectrum first, and this identification is then stored by the system in a general database of configurations (SSDB hereafter; SALT Spectral Database). These initial saved configurations are used for subsequent spectra if all configuration parameters are the same and non-interactive (automatic) processing mode is set. Some changes with time of some parameters in a particular configuration are compensated for by the availability of these configuration files for different dates, and the closest date for the specific spectral configuration is always selected.

Currently the SSDB includes the following information:

(1) the identified spectral lines in the reference spectrum for a particular spectral configuration;

(2) coefficients of the 2D transformation between pixel coordinates and wavelengths for a particular spectral configuration;

(3) fully prepared and normalized spectral flat fields for each spectral configuration (if previously observed and processed);

(4) gain-correction coefficients (see Section 3.2);

(5) spectral sensitivity curves for all previously observed spectrophotometric standards observed with different spectral configurations.

All of these types of data are tied to the dates of observation and the automatic pipeline selects calibra-

[2] All the SALT reports that the author references in this article are collected in https://www.saao.ac.za/akniazev/pub/RSS/. Some of them are also referenced in http://pysalt.salt.ac.za/proposal_calls/current/ProposalCall.html.

[3] In this particular case, the author interprets efficiency as the fastest way to create and debug working programs used by the author to reduce long-slit spectra.

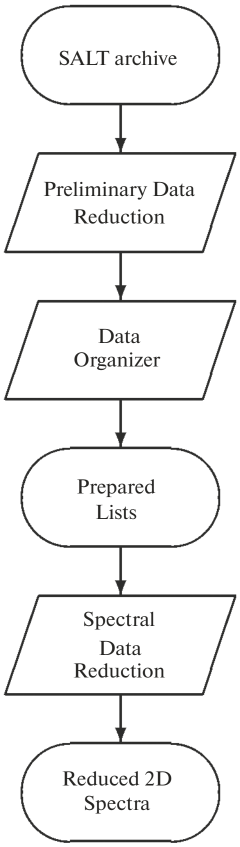

Fig. 1. Flow chart for the RSS spectral pipeline.

tions of a given spectral configuration that are closest by date.

4. DETAILED DESCRIPTION OF THE RSS SPECTRAL PIPELINE

he general flow chart of the RSS spectral pipeline is shown in Fig. 1. In this and the following flow diagrams the following notations are assumed:

- Ovals show input/output data;
- Rectangles show specific processing steps implemented using specific IRAF/MIDAS programs;
- Parallelograms show calls to fairly complex procedures that in turn consist of a set of several steps;
- Rhombuses show the transitions into different branches and the criteria used for them.

4.1. Primary Data Reduction of RSS Spectra

The primary processing of RSS spectra is not shown in the diagram of Fig. 1. This primary data reduction is carried out in automatic mode after the end of each night of observations and includes corrections for bias, overscan, cross-talk and gains. All these steps are done separately for each of the six ADCs and finally all data from these ADCs are combined into a common RSS mosaic (Crawford et al., 2010).

4.2. Preliminary Data Reduction of RSS Spectra

The flow diagram of the preliminary data reduction of RSS spectra is shown in Fig. 2. Preliminary data reduction includes a set of steps that, for various reasons, are not done during the primary data reduction and additionally, these steps are often only specific for the RSS data processing and also do not belong to spectral data reduction. Preliminary data reduction includes the following steps:

(1) *Correction of bad columns and lines on CCDs.*

(2) *Calculation and application of gain-correction coefficients.* Unfortunately, the RSS CCD mosaic has some problems with the electronics which result in changes of gains. This problem leads to the fact that after performing the primary data reduction using previously obtained gains, the levels, which should be the same for all six ADCs, differ by a few percent (in particularly bad cases, this difference was as much as 5—6%; Kniazev, 2020b). Once such a difference is detected, new gain values are calculated and are used after that (Kniazev, 2020a). However, there are cases when such jumps happened several times during the night, so the problem of random changes of gain values is important and needs to be controlled continuously. The solution to the problem is made easier by the fact that the SALT telescope design makes absolute photometric calibration practically impossible. Therefore, after bias subtraction, all values from each amplifier can be multiplied by some arbitrary coefficient, bringing identical readings of different ADCs to the same numerical level. The author has developed a simple algorithm for the automatic calculation of the gain-correction coefficients required to correct the level of each ADC (Kniazev, 2019) and this algorithm is used in this step of the preliminary reduction. In the case where a set of flat fields was observed during a block of observations of a science object, these flats are used for the calculation of gain-correction coefficients that apply at the time of observation and RSS mosaic readout parameters. These coefficients are applied to the science spectra of this block of observations and are written to the SSDB, for subsequent use in the processing of other observational blocks that are nearest in time but do not have observed flat fields. An example of using this algorithm is shown in Fig. 2. The algorithm aligns the levels with an accuracy no worse than 0.5%.

(3) *Construction of a spectral flat field for subsequent pixel sensitivity correction.* After applying the gain-correction coefficients, spectral flat fields, if they were observed for this particular observational block[4], are stacked with median averaging and normalized using the sliding median algorithm. These normalized flat fields are used to correct the pixel-to-pixel sensitivity

[4] The light sources for the flat fields are two continuum spectrum lamps mounted in the telescope's calibration system. There is a single location for all calibration sources of all scientific instruments, the light from which is fed into the telescope's beam by a special mirror that overlaps the signal path of the object.

variations, resulting in a 3–5% reduction in noise variance, which is an important improvement when obtaining spectra of faint objects and/or studying the outer extended parts of galaxies. Since the pixel sensitivity is a function of wavelength, different spectral configurations, binning parameters and CCD readout parameters result in slightly different spectral flat fields. Finally, normalized spectral flats are stored in the SSDB for later application to other observational blocks lacking spectral flats but with the same configuration and nearest in time.

(4) *Filling the gaps between different CCDs of the RSS mosaic.* This step is necessary because during the subsequent spectral processing data gaps between CCDs will become curved after 2D transformation to wavelengths. This curvature introduces large errors in the subsequent background interpolation if the gaps are not filled with interpolated values beforehand.

(5) Finding and removing cosmic ray events. In addition to the standard MIDAS algorithm for finding and removing cosmic ray events, the author implemented in MIDAS the algorithm described in van Dokkum (2001) that is based on the use of a Laplace filter to select strong intensity gradients. This algorithm is already implemented in IRAF and Python,[5] but is very slow. To improve the speed of the algorithm, as well as to improve the quality of its performance with spectral data, the author additionally used a MIDAS context AIP that makes possible (1) the creation of and topological operations with mask images and (2) performing background correction using such image masks (Kniazev, 1997; Kniazev et al., 2004; Lorenz et al., 1993; Vennik et al., 1996).

4.3. Data Organizer

For any pipeline reduction it is very important to organize the input data properly. Such data, comprising the actual scientific data and the set of relevant calibrations, should be grouped according to certain attributes. Very often this type of work is done with the use of a specific kind of software called a “Data Organizer”. A typical example of such a data organizer is the MIDAS package (a context in MIDAS terminology) DO (Grosbøl and Ponz, 1990). This context can be seen as a kind of prototype for the GASGANO software[6] currently used to organize data obtained with all instruments of the European Southern Observatory (ESO) for subsequent processing.

All data obtained with SALT are already organized into observational blocks that were constructed by astronomers who obtained observational time with the telescope[7]. For that reason a simple data organizer has been written with the use of the UNIX bash shell; this program considers a very limited number of standard situations:

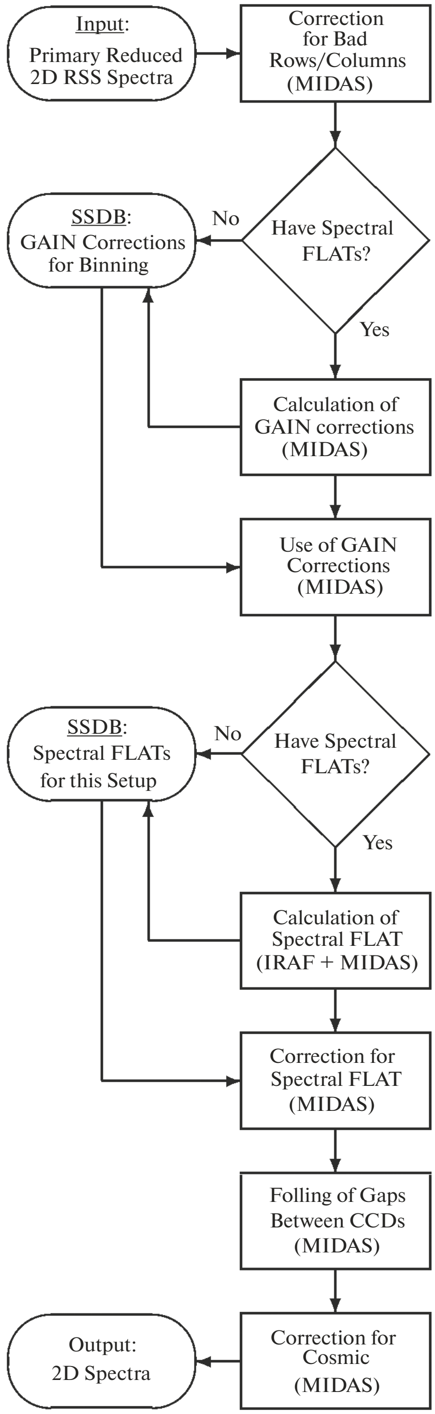

Fig. 2. Flow chart of the preliminary RSS spectra reduction.

(1) standard observation when a set of spectra of some object was obtained each with the same configuration and accompanied by a reference spectrum;

(2) standard observation when a single object was exposed several times during the same block with dif-

[5] http://www.astro.yale.edu/dokkum/lacosmic/.

[6] https://www.eso.org/sci/software/gasgano.html.

[7] https://astronomers.salt.ac.za/proposals/phase-ii.

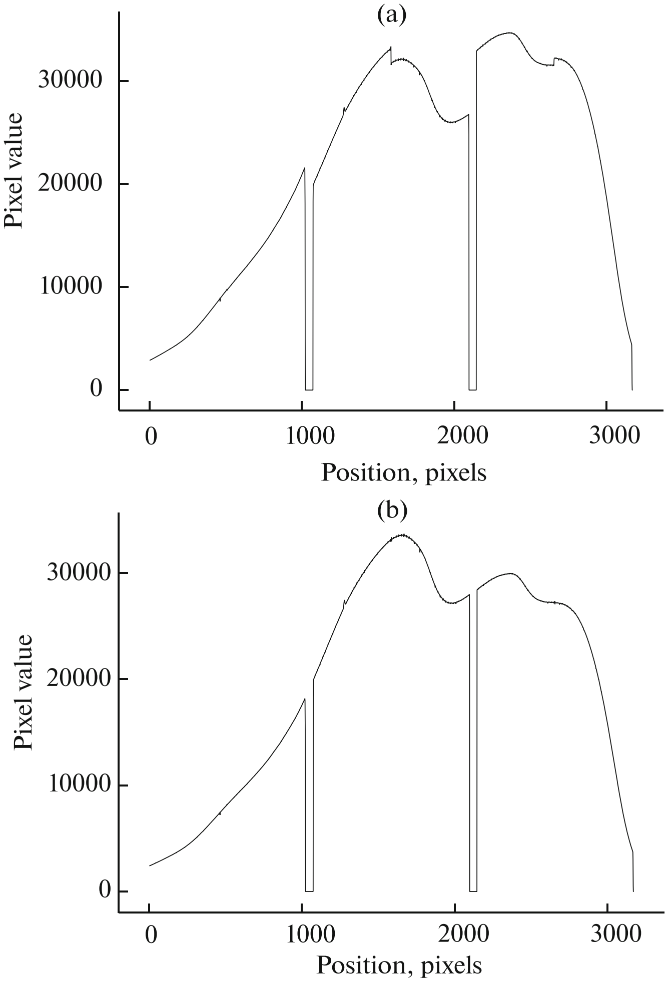

Fig. 3. Example of the spectral continuum distribution of a flat-field lamp before use of gain-correction coefficients (panel (a)) and after (panel (b)). Only the central line of the mosaic is shown. Each panel shows two dips to level zero, which are the gaps between the three CCDs of the mosaic. The panel (a) clearly shows the steps in the middle of each CCD (except the leftmost one), which show the boundaries between adjacent ADCs on each CCD. One can also see the jumps between levels on neighboring CCDs around the gaps. On the panel (b) one can see the smooth level line after calculating and applying the gain-correction coefficients. As the "true" level, the level of the third ADC was selected, which remains unchanged.

ferent spectral configurations and each configuration is accompanied by its own reference spectrum;

(3) an object was observed several times during the night, each time as a new block, each observation being accompanied by its own reference spectrum;

(4) any of the cases described above, but where a reference spectrum was not taken.

Based on an analysis of the above situations, the data organizer sorts the input data, organizes them into lists, and sends these lists to the input of the long-slit Spectral Data Reduction pipeline.

4.4. Long-Slit Spectral Data Reduction

The processing of spectral data for each specific spectral configuration can be broken down into three basic steps:

(1) reduction of reference spectrum;

(2) reduction of spectrophotometric standard;

(3) reduction of object.

In the absence of a reference spectrum or spectrophotometric standard observed on the same night, the required data are extracted from the SSDB based on the criteria: identical configuration, closest in time of observation.

As already noted in Section 2 and shown in the diagrams, all the lower-level programs used in the long-slit reduction steps are tasks from the longslit and apextract IRAF/PyRAF packages (Kniazev, 2008). It should be noted here that the process of long-slit spectral reduction with IRAF/PyRAF is described in detail in numerous documents (e.g., collected here)[8], and each task has numerous parameters that are described in detail there, or could be displayed on-

[8] https://www.saao.ac.za/~akniazev/pub/Manuals.

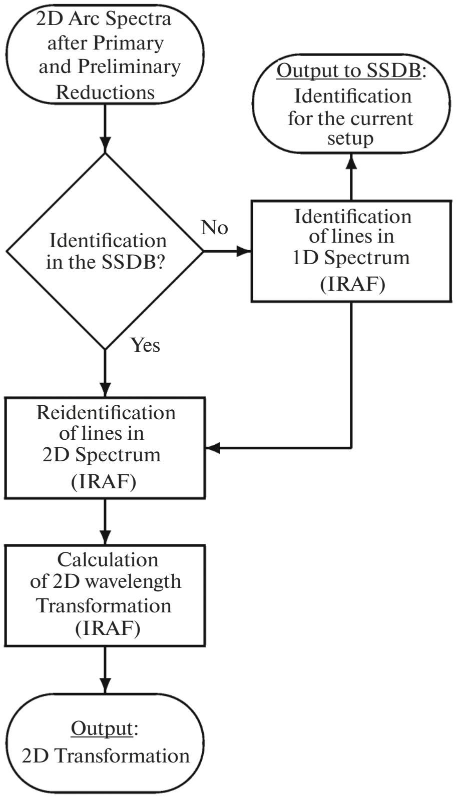
2D Arc Spectra after Primary and Preliminary Reductions
Output to SSDB: Identification for the current setup
Identification in the SSDB?
No
Identification of lines in 1D Spectrum (IRAF)
Yes
Reidentification of lines in 2D Spectrum (IRAF)
Calculation of 2D wavelength Transformation (IRAF)
Output: 2D Transformation

Fig. 4. Flow chart of the reference spectrum reduction.

screen within IRAF/PyRAF using `help task_name`.

4.4.1. Reduction of Reference Spectrum

A flow diagram for processing the comparison spectrum is shown in Fig. 4. Observations of wavelength reference spectra are part of the so-called standard SALT Calibration Plan; described as user-requested charged calibrations, (UCC[9]), they are therefore, in the vast majority of cases, an integral part of a particular observational block.

(1) The first step is to either retrieve the previous identification from the SSDB, or, if a similar configuration has not been used before, to perform the identification in interactive mode. At the start of the identification process the IRAF task extracts several central rows of the 2D reference spectrum (a configurable parameter). The identify task is used, where the standard call when processing data with the PG900 grating looks like this[10]:

```
> identify images = refspec.fits
coordlist = cuar.dat
function = chebyshev
order = 5 fwidth = 6. cradius = 6.,
```

where “refspec.fits” is the name of the input file, and “cuar.dat” is the ASCII file containing the list of lines for the CuAr lamp. When the RSS spectral pipeline is used, all input parameters are set automatically depending on the grating name and the name of the reference spectrum lamp, specified in the FITS header. Initial interactive identification of lines of the reference spectrum can be made using, for example, the line identifications given in the Kniazev (2009) or the identifications given on the WEB[11] page.

[9] For details see Sections 5 and 7.10 of the document http://pysalt. salt.ac.za/proposal_calls/current/ProposalCall.html.

[10] IRAF/PyRAF assumes that any task call is a single line. Breaking it up into separate lines here is done purely for the sake of convenient display. Also, it should be kept in mind that all parameters can be set interactively by calling epar task_name.

[11] http://pysalt.salt.ac.za/lineatlas/lineatlas.html.

Since the RSS spectrograph has six different gratings, the spectral resolution spans the range, $R = 150$–12000, and the final accuracy of the dispersion curve strongly depends on the grating and slit width used. Generally, it varies from 0.5 pixel for the smallest R to 0.05 pixel for the largest R. An absolute comparison of the resulting velocities after RSS data processing is presented, for example, in Kniazev (2012).

(2) The next step is to identify lines across the entire two-dimensional spectrum. The `reidentify` task is used, where the standard call to work with RSS data looks like this:

```
> reidentify reference = refspec.fits
images = refspec.fits interactive = no
newaps = yes override = no refit = yes
nlost coordlist = cuar.dat verbose = yes,
```

(3) The last step is to construct a 2D wavelength transformation. It is possible to perform this step in both automatic and interactive mode. The `fitcoord` task is used, where the standard call when processing data from RSS looks like this:

```
> fitcoord images = refspec
interactive = yes combine = no
function = legendre xorder = 5 yorder = 3.
```

4.4.2. Reduction of Spectrophotometric Standard

Observations of spectrophotometric standards are the part of the standard SALT Calibration Plan that does not count against the user's time (UNC—User-requested Non-charged Calibrations). Spectral configurations as well as CCD binning parameters are accumulated in the SALT database after science spectral observations requiring a spectrophotometric standard are obtained. The spectrophotometric UNC standards are observed during astronomical twilight. Observations are made with the largest slit, of width 4″, and therefore are not accompanied by reference spectra because they have poor spectral resolution and because spectra so obtained will be further rebinned with a window of a few tens of Ångstrom. It has been suggested that reference spectra from science observations could be used for the reduction of spectrophotometric standard of the same configuration. Nevertheless, formally speaking, the user can request such observations as a separate block belonging to a specific observational program and accompanied by a reference spectrum (UCC). Since absolute calibration is not achievable on the SALT telescope, spectrophotometric standard observations are not tied to a given observing night.

The flow chart for the reduction of a spectrophotometric standard is shown in Fig. 5. Steps are following:

(1) The first step is a 2D transformation to wavelengths using the coefficients of a previously constructed transformation for a given RSS spectral configuration and CCD binning. In this case, the `transform` task is used, which employs a transformation built during the `fitcoords` task. Using the names from the previous section, the command for RSS data reduction can be written as follows:

```
> transform input = stdstar.fits
output = stdstarw.fits fitnames = refspec
interptype = linear flux = yes blank = INDEF
x1 = INDEF x2 = INDEF dx = INDEF y1 = INDEF
y2 = INDEF dy = INDEF ,
```

where "stdstar.fits" is the filename of the spectrophotometric standard of a particular spectral configuration for which a 2D dispersion curve "refspec" was calculated. The pixel interpolation will be linear (for objects with strong emission lines this is the most stable option), and the flux will be conserved. The remaining parameters determine the transformation area and step sizes for both coordinates and are set equal to "as obtained". During this step the geometry of the 2D spectra (curvature of spectral lines, possible slope of the dispersion axis) should be fixed.

(2) The next step is to construct a 2D background in the direction perpendicular to the direction of dispersion. The `background` task is used. In the case of the spectrophotometric standard, it is natural to set non-interactive mode:

```
> background input = stdstarw.fits
output = stdstarws.fits axis = 2
interactive = no naverage = 1
function = chebyshev order = 4 low_rej = 1.5
high_rej = 1.5 niterate = 5
```

where "stdstarw.fits" is a 2D spectrum converted to wavelengths, and "stdstarws.fits" is the same spectrum after subtracting the constructed 2D background. The `axis` parameter sets the dispersion axis, and naverage sets the number of lines/columns to be averaged in the direction perpendicular to the dispersion direction. The `order` is the degree of the 1D polynomial used, `low_rej` and `high_rej` are levels of the signal that are discarded when each line/column is approached with a 1D polynomial of the `order` degree and `niterat` is the number of such iterations. In the case of interactive mode, the user can change all the above parameters and see the result of their use for the specific line/column before the procedure with the selected parameters will be executed for the whole spectrum. Also in interactive mode it is possible to select only those areas in the direction along the slit that will participate in the construction of the background.

(3) The next step uses the `apall` task from the apextract package, which is used to extract a one-dimensional (1D) spectrum from a 2D one. It assumes that the 2D spectrum is not aligned in the dispersion direction for a variety of reasons, so it looks for the location of the center of the two-dimensional spectrum with a certain step, fits all found centers with a given polynomial and either retrieves the 1D spectrum after that, or retrieves a given part of the 2D spectrum aligning it along the dispersion direction. In the current case, it extracts a 1D spectrum in non-interactive (automatic) mode:

```
> apall input = stdstarws.fits
format = onedspec interactive = no nfind = 1
llimit = -25 ulimit
```

where "stdstarws.fits" is the 2D spectrum after subtracting the 2D background, and the output is a 1D spectrum with name "stdstarws.0001.fits", `llimit` and `ulimit` define the limits of the summation, `t_order` is the degree of the 1D polynomial applied to the positions of all centers found, and the parameter `nfind` is the number of objects (apertures) found. In interactive mode, values of the input parameters can be changed interactively to get the best result.

(4) Then, the resultant spectral distribution of the extracted 1D spectrum is collapsed with the same step as the absolute energy distribution of a given spectrophotometric standard, taking into account the air-mass at which this standard was observed and using the extinction curve of the particular observatory. All of this is done using the `standard` task:

```
> standard input = stdstarws.0001.fits
output = sens.dat caldir = ../caldir
interact = no star_name = feige110
extinct = ../caldir/suzextinct2.dat,
```

where "sens.dat" is the output ASCII file, "caldir" is the directory where the files of energy distributions of different spectrophotometric standards are located, and "extinct" is the filename with the extinction curve. By default, it is assumed that the air-mass is written as a parameter in the FITS header.

(5) The last step is to create a spectral sensitivity curve, which is done using the sensfunc task:

```
> sensfunc standard = sens.dat
sensitiv = sens.fits interactive = yes,
```

where "sens.fits" is the output file with the spectral sensitivity curve.

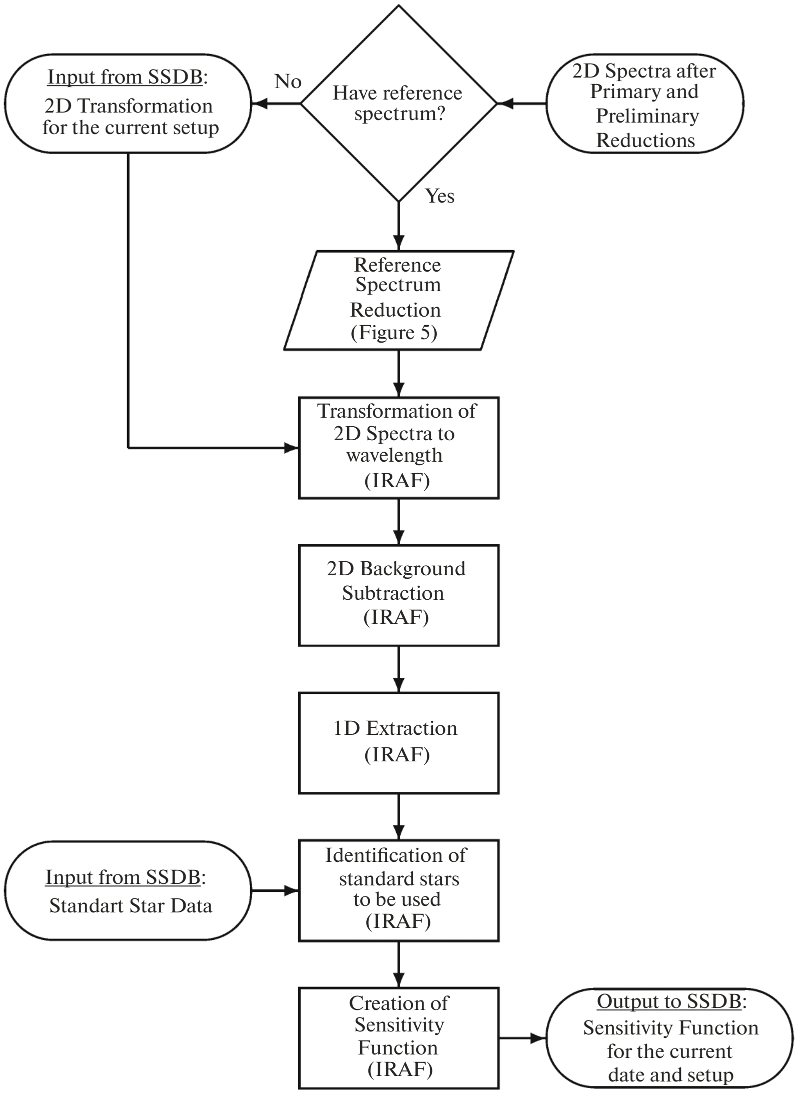
Input from SSDB:
2D Transformation for the current setup
No
Have reference spectrum?
2D Spectra after Primary and Preliminary Reductions
Yes
Reference Spectrum Reduction (Figure 5)
Transformation of 2D Spectra to wavelength (IRAF)
2D Background Subtraction (IRAF)
1D Extraction (IRAF)
Input from SSDB:
Standart Star Data
Identification of standard stars to be used (IRAF)
Creation of Sensitivity Function (IRAF)
Output to SSDB:
Sensitivity Function for the current date and setup

Fig. 5. Flow chart of the spectrophotometric standard reduction.

4.4.3. Reduction of Object

The flow diagram for an object reduction is shown in Fig. 6.

(1) The first step is a 2D transformation to wavelengths using the coefficients of the previously constructed transformation for a given RSS spectral configuration and CCD binning. In this case, the `transform` task is used with a transformation built during the `fitcoords` task. With the names from previous sections, the command for RSS data reduction can be written as follows:

```
> transform input = object.fits
output = objectw.fits fitnames = refspec
interptype = linear flux = yes
blank = INDEF x1 = INDEF x2 = INDEF
dx = INDEF y1 = INDEF y2 = INDEF dy = INDEF .
```

(2) The next step is to construct a 2D background in the direction perpendicular to the direction of the dispersion. Task `background` is used. In the case of an object spectrum, it is natural to set interactive mode to control the result:

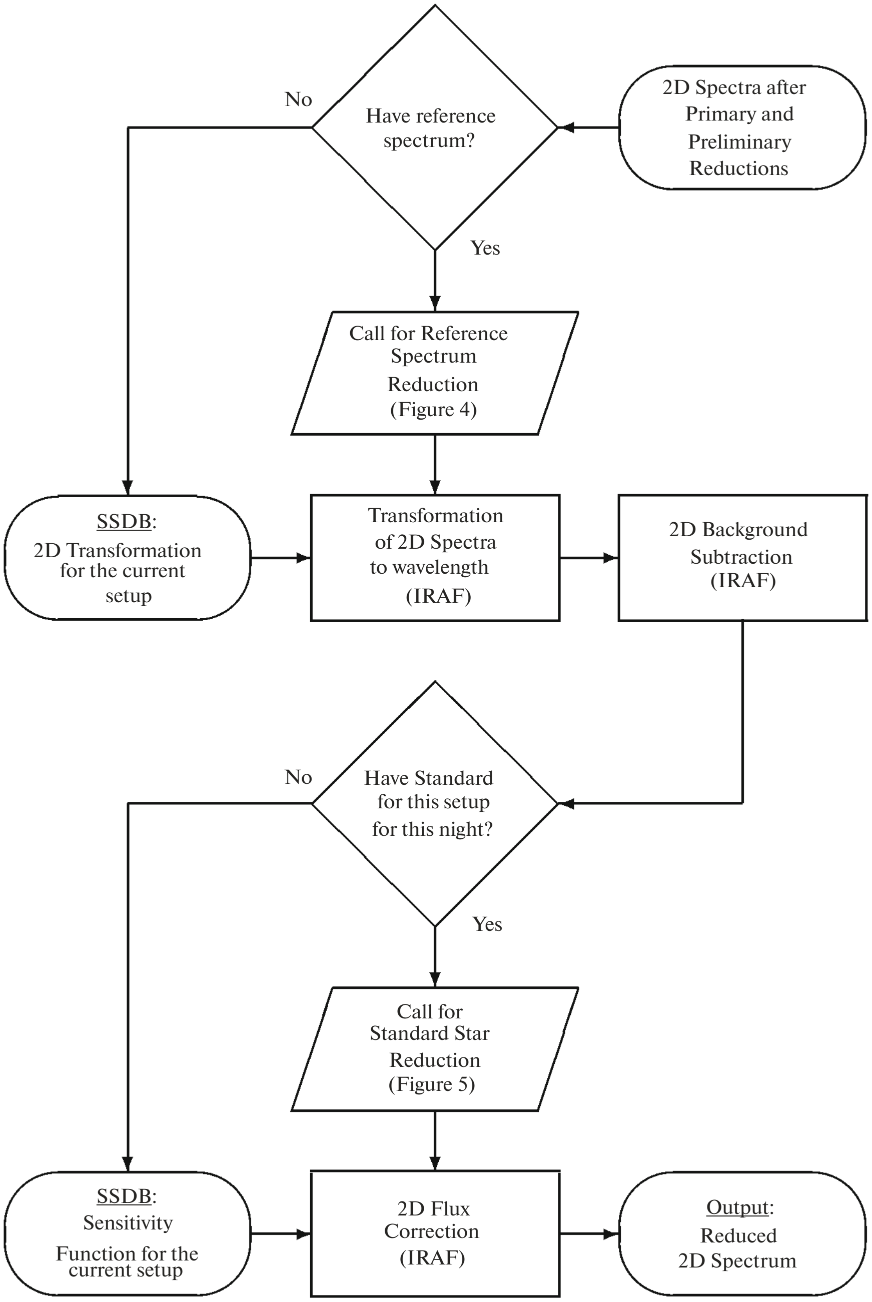
Have reference spectrum?
2D Spectra after Primary and Preliminary Reductions
No
Yes
Call for Reference Spectrum Reduction (Figure 4)
SSDB: 2D Transformation for the current setup
Transformation of 2D Spectra to wavelength (IRAF)
2D Background Subtraction (IRAF)
Have Standard for this setup for this night?
No
Yes
Call for Standard Star Reduction (Figure 5)
SSDB: Sensitivity Function for the current setup
2D Flux Correction (IRAF)
Output: Reduced 2D Spectrum

Fig. 6. Flow chart for object reduction.

```
> background input = objectw.fits
output = objectws.fits axis = 2
interactive = yes naverage = 1 niterate = 5
function = chebyshev order = 4 low_rej = 1.5
high_rej = 1.5 .
```

(3) The last step is to correct the 2D spectrum of an object with sky background removed for the spectral sensitivity curve, taking into account the value of the air-mass during observations. This is done using the `calibrate` task:

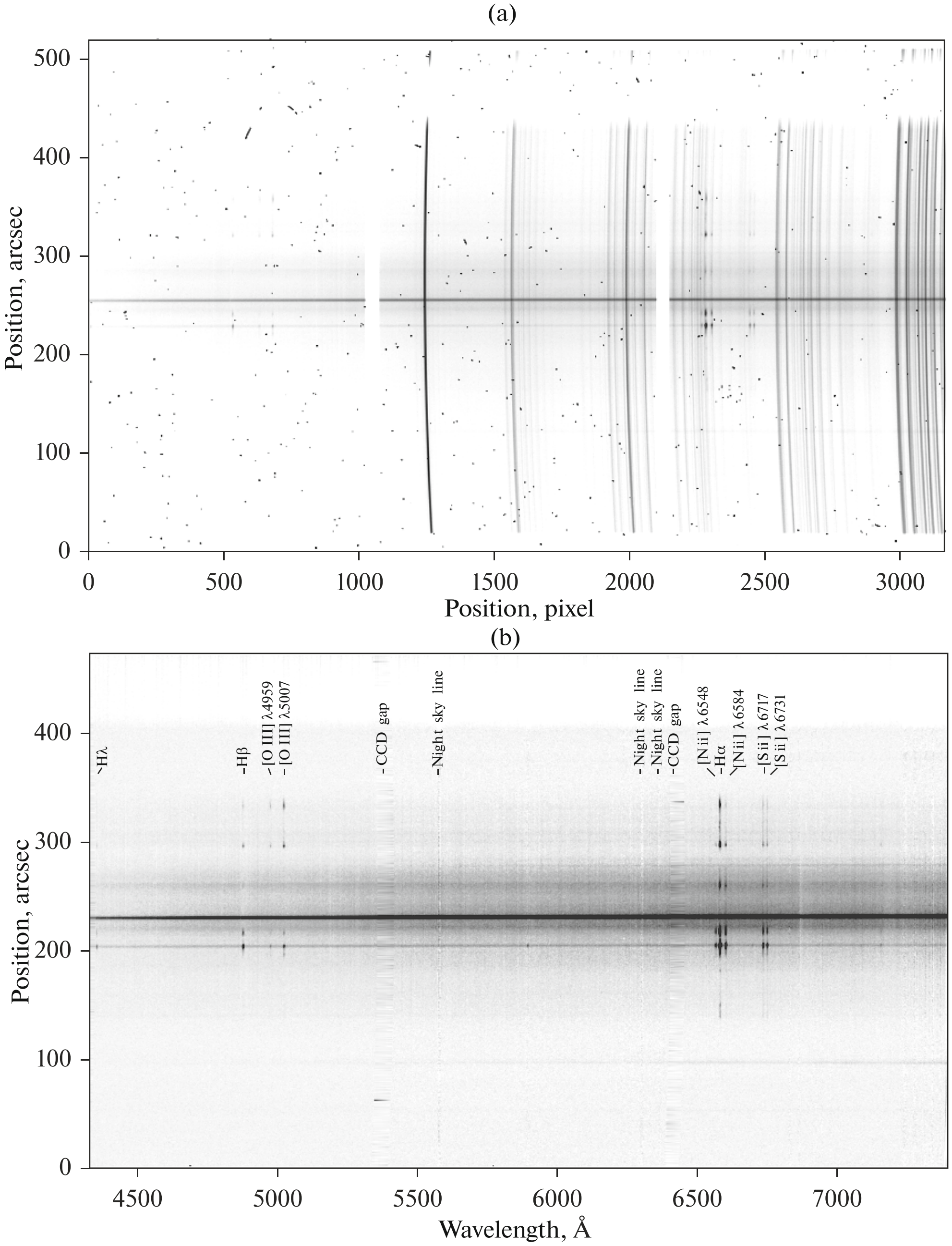

Fig. 7. Two-dimensional spectrum of H II regions of the galaxy NGC 2835. Exposure 300 s. Part of the upper part of the spectrum is overlapped by the guiding system that crosses the slit. Panel (a): unprocessed spectrum. Panel (b): the processed spectrum. The overlapped region was not removed during processing, but was simply excluded during backgrounding. The spectrum of the galaxy is visible over about 200. All major spectral lines are indicated altogether with the brightest lines of the night sky as well as the gaps between the CCD mosaics filled with the interpolated signal.

```
> calibrate input = objectws.fits
output = objectwsf.fits
sensitivity = sens.fits
extinction = ../caldir/suzextinct2.dat .
```

An example of the original RSS spectrum and the same fully reduced spectrum is shown in Fig. 7.

4.4.4. Final Remarks

I would like to point out here that the RSS spectral pipeline is designed for a standard spectral reduction. That is why it does not include a procedure for extracting 1D spectrum/spectra of the object(s), or averaging of the reduced spectra of one object and so on—this all belongs to the further analysis of the reduced data, which should be done by the astronomer who has obtained the fully reduced 2D spectra.

5. THE RSS SPECTRAL PIPELINE SOFTWARE STRUCTURE

The RSS spectral pipeline described above consists of two separate software packages.

The first package is called `RELSA`, the source code of which can be found at bitbucket[12]. The package is designed for long-slit spectral processing and is based on the standard astronomical data processing system PyRAF. The package requires the UNIX shell language for its operation, a preinstalled PyRAF, and either preinstalled standard astronomical data processing system MIDAS or programs `dfits` and `fitsort` for handling FITS file descriptors. A brief description of the package and its main ideas are given in Kniazev (2006), which can also be found in the source of the package and are presented in Section 3.4 of this article.

The current version of the package was also used by the author to reduce long-slit data from the spectrograph SCORPIO mounted on 6-m BTA telescope (Afanasiev and Moiseev 2005), data from the TWINS spectrograph mounted on the 3.5-m telescope at Calar Alto (Spain), and data from the long-slit spectrograph mounted on the 1.9-m telescope at Sutherland (South Africa).

However, it should be noted here that it is the responsibility of the user to create all input file lists as well as to divide them into logical groups (objects, comparison spectra, spectrophotometric standards).

The second package, called `RSS reduction`, is the core of the RSS spectral pipeline. This software produces a preliminary data reduction, collates the spectra with a Data Organizer and finally calls `RELSA`. The source code can be found at bitbucket[13]. A brief description of the package is given in this article. The package requires the UNIX bash shell language, a preinstalled standard astronomical data processing system, MIDAS, and the dfits and `fitsort` programs.

Both software packages are distributed without restriction and anyone who copies these packages may change all the programs as he/she likes. Installing and configuring the packages requires the knowledge of an advanced UNIX user, as well as the ability to work with the MIDAS and PyRAF systems. The author guarantees that both packages work on his computer, with the Linux openSUSE Tumbleweed distribution, but he is not responsible for their operation on other computers and in other environments.

CONCLUSIONS

An RSS spectral reduction pipeline has been developed and implemented. This pipeline is designed for the reduction of long-slit spectral data obtained with the RSS spectrograph mounted on the SALT telescope. The developed system allows a huge variety of RSS spectral configurations to be taken into account. This RSS pipeline includes the following steps:

- preliminary data reduction
- reduction of reference spectrum
- reduction of spectrophotometric standard
- reduction of object spectrum

Most pipeline steps can be done in either interactive or non-interactive (automatic) mode. This pipeline was used by the author for the reduction of about one and a half thousand RSS long-slit spectra obtained from fifty of the author's SALT observational programs. The results of these studies have been published in about forty scientific papers to date.

ACKNOWLEDGMENTS

I dedicate this work to the memory of my good friend, astronomer and science popularizer Oleg V. Verkhodanov. The author acknowledges support from the National Research Foundation (NRF) of South Africa. The author would also like to thank the reviewer for his careful reading of the text and advice on how to improve. Spectral observations that were used to verify and debug the RSS spectral pipeline were obtained on the author's (PI: Kniazev) SALT applications:

2010-1-RSA-004, 2010-1-RSA-007,
2010-1-RSA_OTH-001, 2011-3-RSA-003,
2010-1-RSA_OTH-002, 2010-1-RSA_OTH-012,
2010-1-RSA_OTH_013, 2011-3-RSA_OTH-001,
2011-3-RSA_OTH-002, 2012-1-RSA-001,
2012-1-RSA_OTH-001, 2012-1-RSA_OTH-002,
2012-2-RSA-001, 2012-2-RSA_OTH-002,
2012-2-RSA_OTH-003, 2012-2-RSA_OTH-004,
2013-1-RSA_OTH-008, 2013-1-RSA_OTH-014,
2013-2-RSA_OTH-003, 2013-2-RSA_OTH-005,
2014-1-RSA_OTH-005, 2014-2-MLT-001,
2015-1-SCI-017, 2015-1-SCI-030,
2015-2-MLT-002, 2015-2-MLT-004,
2015-2-SCI-022, 2015-2-SCI-043,
2016-1-MLT-002, 2016-1-MLT-003,
2016-1-MLT-005, 2016-1-SCI-012,
2016-1-SCI-019, 2017-1-MLT-001,

[12] https://bitbucket.org/akn_zdes_i_seichas/relsa/.

[13] https://bitbucket.org/akn_zdes_i_seichas/rss_reduction/.

2017-1-MLT-005, 2017-1-SCI-006,
2017-2-DDT-002, 2017-2-MLT-001,
2018-1-MLT-003, 2018-1-MLT-008,
2018-2-MLT-001, 2019-1-MLT-002,
2019-1-MLT-005, 2019-1-SCI-003,
2019-1-SCI-004, 2019-1-SCI-007,
2020-1-MLT-001, 2020-1-MLT-003,
2020-1-SCI-002, 2020-1-SCI-005.

CONFLICT OF INTERSTS

The author declares that there is no conflict of interest.

REFERENCES

1. V. L. Afanasiev and A. V. Moiseev, Astronomy Letters **31** (3), 194 (2005).
2. S. I. Barnes, P. L. Cottrell, M. D. Albrow, et al., SPIE Conf. Proc. **7014**, id. 70140K (2008).
3. D. G. Bramall, J. Schmoll, L. M. G. Tyas, et al., SPIE Conf. Proc. **8446**, id. 84460A (2012).
4. D. G. Bramall, R. Sharples, L. Tyas, et al., SPIE Conf. Proc. **7735**, id. 77354F (2010).
5. D. A. H. Buckley,G. P. Swart, and J. G.Meiring, SPIE Conf. Proc. **6267**, id. 62670Z (2006).
6. E. B. Burgh, K. H. Nordsieck, H. A. Kobulnicky, et al., SPIE Conf. Proc. **4841**, pp. 1463–1471 (2003).
7. L. A. Crause, R. M. Sharples, D. G. Bramall, et al., SPIE Conf. Proc. **9147**, id. 91476T (2014).
8. S.M. Crawford,M. Still, P. Schellart, et al., SPIE Conf. Proc. **7737**, id. 773725 (2010).
9. P. Grosbøl and D. Ponz, in *Proc. Int. Workshop on Acquisition, Processing and Archiving of Astronomical Images, Astronomical Observatory of Capodimonte, Naples, Italy, 1990*, Ed. by G. Longo and G. Sedmak (Officine-Grafiche Liguori, 1990), pp. 111–123 .
10. A. Y. Kniazev, Candidate's Dissertation in Mathematics and Physics (SAO RAS, Nizhnij Arkhyz, 1997).
11. A. Y. Kniazev, Long-slit reduction with SHyRAF package RELSA, SALT Report (2006).
12. A. Y. Kniazev, SALT Long-slit Data Reduction, SALT Report (2008).
13. A. Y. Kniazev, Atlas of Reference Spectra for RSS Observations, SALT Report (2009).
14. A. Y. Kniazev, RSS: Radial Velocity Accuracy using Emission Lines, SALT Report (2012).
15. A. Y. Kniazev, Automatic gain correction algorithm for the RSS mosaic CCD, SALT Report (2019).
16. A. Y. Kniazev, Gain calculation for the RSS CCD, SALT Report (2020a).
17. A. Y. Kniazev, RSS CCD mosaic gains up to now, SALT Report (2020b).
18. A. Y. Kniazev, E. K. Grebel, S. A. Pustilnik, et al., Astron. J. **127** (2), 704 (2004).
19. A. Y. Kniazev, I. A. Usenko, V. V. Kovtyukh, and L. N. Berdnikov, Astrophysical Bulletin **74** (2), 208 (2019).
20. H. A. Kobulnicky, K. H. Nordsieck, E. B. Burgh, et al., SPIE Conf. Proc. **4841**, pp. 1634–1644 (2003).
21. H. Lorenz, G. M. Richter, M. Capaccioli, and G. Longo, Astron. and Astrophys. **277**, 321 (1993).
22. D. O'Donoghue, D. A. H. Buckley, L. A. Balona, et al.,Monthly Notices Royal Astron. Soc. **372** (1), 151 (2006).
23. P. G. van Dokkum, Publ. Astron. Soc. Pacific **113** (789), 1420 (2001).
24. J. Vennik, U. Hopp, B. Kovachev, et al., Astron. and Astrophys. Suppl. **117**, 261 (1996).