Handbook of Data Reduction School for Subaru/MOIRCS spectroscopy

Tomohiro Yoshikawa
Koyama Astronomical Observatory
Kyoto Sangyo University
3/22/2011

Translated by Miki Ishii
Subaru Telescope
National Astronomical Observatory of Japan
9/18/2013

Revised by Ken-ichi Tadaki
National Astronomical Observatory of Japan
2/13/2014

Contents

Chapter 1

Introduction

This handbook shows an example of how to reduce and analyze the data obtained with MOIRCS Multi-Object Spectroscopy (MOS), using IRAF (Image Reduction and Analysis Facility) and its python interface PyRAF. I assume that the readers have already learned basic UNIX commands, PyRAF/IRAF, and how to show the command help.

As the sample data, I adopted the MOS data acquired for the observations of emission lines from multiple high-redshift galaxies. Then the reduction/analysis contains (1) extract the spectrum of the target object from a frame with multiple spectra, (2) sum the extracted target frames, and (3) measure its emission lines. I expect that many of the above processes can also be applicable in reducing low-resolution ($R \sim 500 - 2000$) spectra with a slit spectroscopy in the near-infrared.

The reduction processes described in the main text are automated by MCSMDP (MOIRCS MOS Data Pipelines), which was developed with PyRAF. The pipelined processes with MCSMDP is summarized in the Appendix. I hope that the main text part is also useful to understand the processes in the pipeline reduction.

This book was written as a handbook of "Subaru Autumn School 2010" by the lecturer Tomohiro Yoshikawa (Kyoto Sangyo University, Koyama Astronomical Observatory, research specialist). The sample data in the handbook are the data from the Subaru Telescope open use observations S07A-083 (archived in SMOKA), and the mask design file (MDP file) used with the observations, which Dr. Masayuki Akiyama (Tohoku University Graduate School of Science, Astronomy Department, Associate Professor), the PI of the observation, kindly provided. Part of the sample data were published in Yoshikawa et al. 2010 [5].

I acknowledge the coordinators of "Subaru Autumn School 2010", who are the members of the National Astronomical Observatory of Japan (Hawaii Observatory, Division of the optical and infrared astronomy, and the Astronomy Data Center).

Tomohiro Yoshikawa
research specialist, Koyama Astronomical Observatory,
Kyoto Sangyo University
12/14/2010

Chapter 2

MOIRCS MOS spectroscopic data

2.1 MOIRCS

MOIRCS (Multi-Object InfraRed Camera and Spectrograph), which is installed at the cassegrain focus of the Subaru Telescope, provides imaging and spectroscopic capabilities in the near-infrared (0.9–2.5 μm). The imaging mode enables a $4' \times 7'$ field of view, with the $YJHK_s$ broad band filters and some narrow-band filters. The spectroscopic mode provides low-resolution ($R \sim 700$), medium-resolution ($R \sim 1500$), and high resolution ($R \sim 3000$) spectroscopy with grisms and VPH grisms. Table 2.1 summarizes the MOIRCS grisms with the wavelength coverage and the resolution. The spectroscopic mode also enables multi-object spectroscopy of about 40 objects, by placing a slit mask (with multiple slits) on the focal plain. This function is similar to that of FOCAS (Faint Object Camera and Spectrograph), one of the instruments of the Subaru Telescope for the visible light. On the other hand, the slit mask of MOIRCS is made of thin aluminum plate, which is cooled down to ∼100 K, same as other part of the instrument, to suppress thermal emission such as from the instrument (dominant at >2μm).

Table 2.1: MOIRCS grisms [a]

Grism	Wavelength [μm]	Resolution [R] [b]	Dispersion [Å/pixel]	Notes
$zJ500$	0.9–1.78	700	5.57	
$HK500$	1.3–2.5	640	7.72	H band
		820		K band
$R1300$	1.16–1.34	1500	1.91	J band, 4th order light [c]
	1.45–1.80	1600	2.61	H band, 3rd order light [c]
	2.00–2.40	1500	3.88	K band, 2nd order light [c]
$VPH-J$	peak at 1.23	3050	0.96	with wavelength shift [d]
$VPH-H$	peak at 1.65	2940	1.31	with wavelength shift [d]

[a] When you write a proposal, please review the Subaru web site carefully, and consult the support scientist if necessary.

[b] Resolution with the slit width of 0.″5.

[c] J, H, and K band filters are used as the order sort filter to put the 4th-, 3rd-, and 2nd-order light, respectively. The efficiency decreases as the order increases. Efficiency at J is about half of that of the K band.

[d] The peak wavelength will shift depending the spatial position of the slit. The shift differs among the grisms and the channels. Please read the Subaru web page for the detail.

MOIRCS is equipped with two Hawaii-2 detectors to obtain the data. The Hawaii-2 is HgCdTe array with the format of 2048 pixel × 2048 pixel. The pixel scale at the focal plain of the MOIRCS is 0.″117/pixel both for imaging and spectroscopic modes. The incoming light to MOIRCS is divided into two field of views at the telescope focal plain, and then re-focused onto the two HAWAII-2 arrays through two sets of identical optics. The two HAWAII-2 arrays are placed a little inward past the center, to pick the spectra of sources near the edge of the FOV with the spectroscopic mode. As a result, with the imaging mode, the light has not come into the central ∼0.′5 area (∼ 300 pixel), as shown in figure 2.1. Although the two

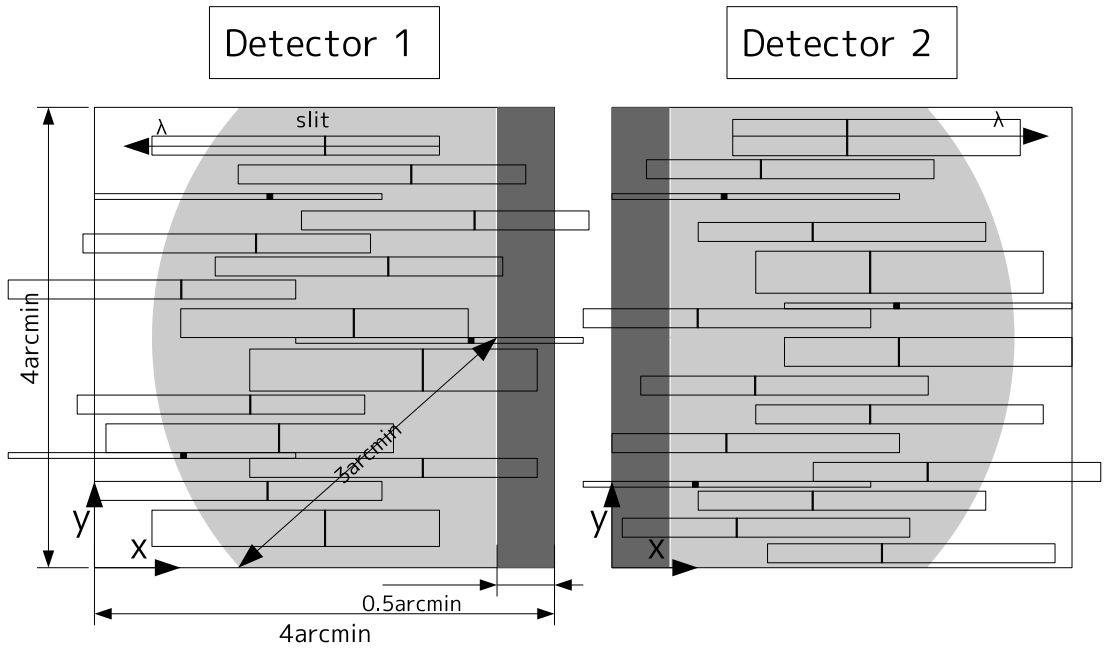

Figure 2.1: Schematic view of the MOIRCS two detectors, with an example of the image with the spectroscopic mode. The black rectangles represent slits, while white (outlined) rectangles represent the spectra. The slits are allowed to place in the light-gray areas. The spectra are dispersed along the X-axis — the wavelength increases as the x-coordinate increases on the Detector 1, which is opposite in the Detector 2. (The dispersion directions are opposite for $VPH - J$ and $VPH - H$ grisms.) White and light-gray areas represent the FOV available with the imaging mode. The spectra are allowed to spread over the entire detector, but some part of a spectrum may run over the detector, depending on the position of the slit.

arrays are illustrated in figure 2.1 with a space in between, it does NOT mean that there is unobservable area at the center of FOV — because the FOV is split onto two at the telescope focal plain, the $4' \times 3'.5$ FOV with each of the two arrays is nearly continuous.

The two arrays are called Detector 1 and Detector 2, as shown in figure 2.1. When the position angle (PA) is zero, the Detector 1 and the Detector 2 covers the southern and northern part of the FOV, respectively. With the spectroscopic mode, a spectrum is dispersed along the X-axis, so that the slit is placed along the Y-axis. The dispersion direction is different between the two detectors: the wavelength increases as the x-coordinate increases in the Detector 1, and the reverse in the Detector 2. Note that the dispersion directions of $VPH - J$ and $VPH - H$ grisms are opposite to that of other grisms.

For the details of the instrument, please refer to the instrument paper (Suzuki et al. 2008 [4]). For the details of observations modes available, please refer to the Subaru web

page.[1]

As is the case with the other instruments of the Subaru Telescope, MOIRCS data are stored to FITS (Flexible Image Transport System) files. Each of the MOIRCS two detectors creates individual data file, so that two FITS files are created in regard to one data acquisition. The data files are named like as "frame-ID.fits", and it is uniquely-numbered by the Subaru Observation Control System. The frame-ID consists of alphabets and numerics, e.g., MCSA00057429. The "MCS" represents that the data were obtained with MOIRCS, and the "A" (after "MCS") represents that it is a raw frame, directly created from the instrument. The eight-digit number after "MCSA" is a sequential serial number — there is no identical number since the MOIRCS first light. In principle, frame-IDs for the Detector 1 and Detector 2 have odd numbers and even numbers, respectively.

FITS files have header part to contain meta-data such as the observation conditions, in addition to the data part to record the value of each pixel. The FITS header consists of lines of a card image, where a card image consists of three parts, i.e., keyword, value, and comment. For MOIRCS, each of the observation modes (imaging, long-slit spectroscopy, and multi-object spectroscopy) has its own set of card images with keywords, although some keywords are common among the observation modes. For example, independent of the observation modes, the card image with keyword "FRAMEID" records the frame-ID, so that one can know the original frame-ID even if the file name has been changed. Table 2.2 shows important keywords in the MOIRCS FITS header (especially for the MOS mode).

Because FITS style is widely used among astronomical researchers and amateurs, there are many softwares to read/write the FITS files. In this text, how to read/write FITS files using IRAF/PyRAF is explained in a later section. For the details about FITS data format, corresponding softwares, and the FITS headers added to the Subaru Telescope's data, please refer to the "handbook of FITS" by the NAOJ/Astronomy Data Center. The handbook is also available online at http://www.fukuoka-edu.ac.jp/~kanamitu/fits/index.html.

[1] http://subarutelescope.org/Observing/Instruments/MOIRCS/index.html

Table 2.2: Major keywords in the MOIRCS FITS header

Keyword	Example	Note
FRAMEID	MCSA00057430	Frame-ID: same as the file name of the raw frame.
EXP-ID	MCSA00057429	Exposure-ID: same as the frame-ID of the data taken with the Detector 1.
DET-ID	2	Detector-ID: either 1 or 2.
OBS-MOD	SPEC_MOS	Observation mode: "IMAG" for imaging and "SPEC" for long slit spectroscopy.
DATA-TYP	OBJECT	Data type: other examples are DARK, DOMEFLAT, INSTFLAT.
OBJECT	CDFN_MASK01	Target name.
SLIT	CDFN1	Slit mask, named by the support scientist.
DISPERSR	HK500	Grism name.
EXPTIME	900.050	Exposure time.
K_DITWID	3.000	Nodding length.
K_DITCNT	1	Nth number of nodding: either 1 or 2 for spectroscopy.
DATE-OBS	2010-12-14	Starting date of the observation (UT).
HST-STR	22:25:30.234	Starting time of the observation (HST).
UT-STR	08:25:30.234	Starting time of the observation (UT).

2.2 Near-infrared spectroscopy

Spectroscopy in the near-infrared[2] is basically same as that in the visible light. Therefore, the data reduction method used for the optical low-resolution spectroscopy (such as for FOCAS) can also be applied to the near-infrared spectroscopic data in many aspects. However, there are some differences arising from the characteristics of the near-infrared. I will first explain the characteristics of the near-infrared observations with MOIRCS, by pointing differences from other instruments at the Subaru Telescope. It would be better to refer to handbooks for the other instruments for your better understanding.

Because the silicon CCD (Charge Coupled Device), the detector used for the optical observations such as for FOCAS and Suprime-Cam, does not have sensitivity at wavelength longer than 1 μm, near-infrared observations adopt semiconductor crystals which show sensitivity at the longer wavelength. For MOIRCS, HgCdTe arrays named HAWAII-2 are used as the detectors. Because the manufacturing technique has not yet been well developed compared to the popular silicon CCDs, and especially it is the early large-format near-infrared array, the HAWAII-2 arrays show some imperfections compared to CCDs — such as remarkably inhomogeneous sensitivity, non-sensible pixels (bad pixels), and notable differences among the arrays. Also, unlike in the case of silicon CCD, the charge-transfer method can not be used for such crystals — instead charge in each pixel is read by using multiplexer. This results in another characteristic that the exposure time can be set without a mechanical shutter, by using the double sampling method (readout before and after the exposure).

[2]In this text we use the term "near-infrared" to represent the wavelength between 0.9–2.5 μm which is observable with MOIRCS.

In the near-infrared, radiation from space is absorbed through the Earth's atmosphere, such by as water vapor (H_2O) and carbon dioxide (CO_2). As a result, observable wavelength from the ground is limited within a certain widths of bands. Figure 2.2 shows the atmospheric transmission at the summit of Mauna Kea along with the transmission curve of the broad band filters used for MOIRCS. While the atmospheric transmission bands give a restriction in designing near-infrared broad band filters for imaging (figure 2.2), they should be considered also in case of spectroscopic observations. For example, although the $HK500$ grism enables the simultaneous observation between 1.3–2.5 μm, the spectra at the wavelength around 1.9 μm (boundary between H and K) is almost absorbed by the Earth's atmosphere.

Figure 2.2 also shows a plot of background emission observed by MOIRCS. Primary source of the background emission is night airglow lines, such as from OH and O_3 in the atmosphere. Along with the atmospheric transmission bands, night airglow lines hamper the observations; a target spectrum can not be observed if a strong airglow emission line presents there and the background noise becomes too large. Unlike FMOS, MOIRCS is not equipped with a mechanism to suppress the night airglow lines such as OHS (OH Suppressor), then the emission from night airglow should be removed by software at the time of data reduction.

On the other hand, because the wavelength of OH airglow is well-known, the airglow lines can be used as a wavelength calibrator in case of near-infrared spectroscopy. Then additional lamp frames for the wavelength calibration are not necessarily required, unlike optical spectroscopy. To use the OH lines as the wavelength calibrator, one should adjust the exposure time so that the signals from the OH lines are below the saturation level. A typical exposure time with a low-resolution grism will be about 10–15 minutes. However, a little bit shorter exposure time would be better to remove the background emission to some extent with the "A−B subtraction" (described later), because intensity of the OH lines changes largely in time.

While the airglow emission lines strongly affect the data, continuum component of the background emission is relatively small for MOIRCS. The thermal background (continuum) emission from atmosphere and telescope becomes larger at wavelength longer than 3 μm (such as the case for IRCS or COMICS). For MOIRCS, the thermal emission is suppressed, especially at wavelength shorter than 2.5 μm, by putting a cold stop at the pupil plain. Even so, the residual thermal emission and unresolved airglow lines affect the spectroscopic data obtained with the low-resolution grisms, resulting in background limited data, in practice.

Compared to the observations in the visible wavelengths, near-infrared observations have a short history, and catalogs of standard stars for flux calibration are partial. In particular, there is no catalog of faint spectroscopic standard stars which is adequate for the use of 8-meter class telescopes, to date. In regarding this point, I will discuss how to calibrate the flux level in a later section.

For the details of near-infrared observation technique and its detectors, please refer to "Electronic Imaging in Astronomy; Detectors and Instrumentation" [2] by Ian McLean, if you are interested.

2.3 MOIRCS MOS observations

The procedure of the MOIRCS MOS observation is basically same as that of the FOCAS, optical multi-object spectrograph with slit mask. In ahead of observation, the observer defines

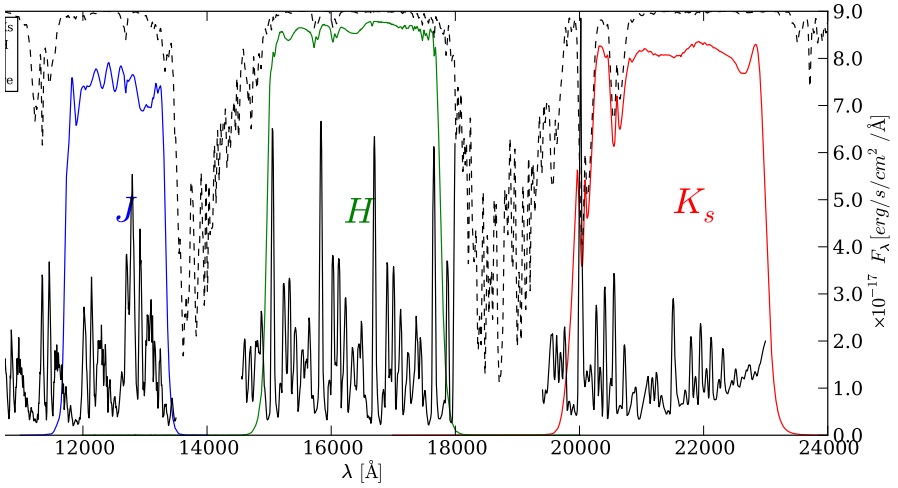

Figure 2.2: Left axis: atmospheric transmission at the summit of Mauna Kea (dashed line) and the transparency of the broadband filters for MOIRCS (J: blue solid line, H: green solid line, K_s: red solid line). Right axis: background emission observed with MOIRCS (black solid line, data with strong atmospheric absorption are omitted).

the slit positions in the field of view (mask design), then the observatory makes the slit mask accordingly. At the time of observation, the telescope will be pointed to the designed position, then the observation will be started by inserting the slit mask at the focal plain. In this section, I will summarize the preparation and the flow of the MOIRCS MOS observation, from the view points that should be known in reducing the data.

2.3.1 Preparation of the observation

Designing a slit mask needs the imaging data (pre-image) with a sufficient S/N to detect alignment stars to mark the target field. It is necessary that the pre-imaging data accurately reproduce the image at the Subaru Telescope's focal plain, so that the target objects can be accurately placed on the slits. The most reliable way for successful observation is the use of imaging data taken with MOIRCS, which minimizes the uncertainty of distortion correction and the pixel scale.

When an open use proposal for MOIRCS MOS observation is accepted, the support scientist will contact the PI of the proposal, asking the PI to define the coordinate and the PA of the pre-image. With the coordinate and the PA from the PI, the support scientist will take the pre-imaging data in a spare time in other open use observations, simply reduce the data, then send the data to PI for the mask design.[3]

[3]Because time for taking the pre-image is provided from other open use programs, all MOIRCS MOS observers are required to give a part of their time for pre-imaging of subsequent MOS observers.

The slit mask is designed with a text file named mdp file. The mdp file defines the coordinate, size, and angle of the slits, the same format as is used for FOCAS MOS observations. To create the mdp file, it is convenient to use a software named wmdp_moircs, which is released at the Hawaii Observatory. The wmdp_moircs is also used to transfer the file into another format (sbr file) which is usable for the observatory's machine (laser cutter) to make the slits. Shrinkage of the slit mask under cooling is also considered at this time, based on the empirical value.

Once the observer sends the mask design, the observatory will make a slit mask on a aluminium plate according to the mask design. Then the slit mask will be installed on a slot of the instrument before the observations, cooled to 100K at the time of the observation.

2.3.2 Observation flow

To place all targets precisely on the slits, it is necessary to point the telescope accurately to the designated position. For this purpose, more than 6 holes are defined on the slit mask, to place alignment stars (point sources) on them. In most cases, well-experienced support scientist or an instrument operator will do the acquisition procedure.

Once alignment stars are centered on the holes, spectroscopic observation will start by inserting a grism. Accurate and longtime pointing of the telescope is particularly important in case of spectroscopy. Auto guiding with a guide star is used for this purpose. In addition, the telescope will be moved back and forth for each exposure so that the targets will move along and within the slits (nodding). The nodding is to remove the time-varying night airglow lines by making subtraction between the neighboring frames. By taking the neighboring frames at different positions on a slit, the subtraction does not remove the targets but remove the airglow lines. The nodding also reduces systematic errors from the bad pixels and inhomogeneous sensitivity, by placing the targets at different positions on the detector, as is the case of imaging observations. However this is not so effective, because usually only two nodding positions will be taken for the spectroscopy.[4]

Because the spectroscopic observation will take long time, the target objects may drop off from the slits during the observation, due to the deflection of the telescope and the instrument, even with the auto-guiding. Therefore, pointing will be adjusted every 1–2 hours by removing the grism and switching to the imaging mode, centering the alignment stars on the holes.

2.3.3 Calibration data

Below is a list of calibration data, which should be taken at the time of MOIRCS observations.

Dark Usually dark frames are not used for the MOIRCS data reduction, because the background noise is much more dominant than the dark. However, the dark frames are taken at the beginning of the observations for the purpose of health check of the detectors.

Flat Dome flat frames are taken in the evening or in the morning, with the slit mask placed at the focal plain. The flat frames are important not only for correcting unevenness of the detector sensitivity but also for correcting the machining error such as slit width.

[4] Although there is another mode to take 4 nodding positions for MOIRCS spectroscopy, this mode is rarely used because the time for taking one sequence becomes so long.

The mask insert/retract movements and the telescope's position would cause a shift of about 6 pixels in the mask position between the target and the flat. To avoid the shift, flat frames can be taken just after the target observations (with the same telescope position) by inserting the cal probe (although this method is rarely used because it takes so much time).

Wavelength calibration Because the wavelength calibrations can be done using the OH airglow lines, as previously explained, calibration data obtained with the comparison lamp are rarely used for the MOIRCS data reduction. Use of OH lines results in more accurate wavelength calibration, because there is no shift in the slit position. However, in case of the data with little OH airglow lines, comparison frames will be taken in the evening or in the morning — for example, if the spectra at the wavelength between 2.2–2.5μm are important (taken with the $R1300$ grism plus K filter), comparison frames with the ThAr lamp frames should be taken for the wavelength calibration, because there are little OH lines at wavelength longer than 2.25μm.

Standard stars Stars which can be easily modeled with smaller absorption lines, such as A0V stars, are observed as standard stars. The flux calibration is made by using a model spectrum, because there is no well-organized near-infrared catalog of the spectroscopic standard stars, as used in the optical observations. Standard stars will be taken in the evening/morning twilights. If you care about the changes of the atmospheric absorption, it is better to take a standard star in the mid night as well. To save the time, a standard star will be taken by at one of the slits in the mask.

Chapter 3

Data Reduction (Overview)

3.1 Procedure of the MOIRCS MOS data reduction

There are various ways for reducing the spectroscopic data, depending on the details and purposes of the data and the analysis. The purpose of the MOS data reduction here is to detect the Hα emission line from the ionized region associated with a distant galaxy, derive the redshift from its wavelength, and measure the Hα flux. For this purpose, it is necessary to increase the S/N ratio of the spectrum by combining multiple frames. First, the data should be reduced to a two-dimensional spectrum where the X- and Y-axis corresponds to wavelength, and the spatial extent, respectively, and the unit of the data is specific flux (F_λ [erg s^{-1} cm^{-2} Å^{-1}]). Then the emission line will be detected from the two-dimensional spectrum. In this section I will explain the steps to reduce the observation data to the two-dimensional spectrum. In the next section, an exercise of the reduction using IRAF will be shown.

The MOS data reduction is basically same as that of the single slit spectroscopic data. However, because there are multiple slits in a single frame for the MOS data, the reduction is divided into two types of the processes — one is the steps applied commonly to an entire frame (common processing) and the other is the steps after extracting individual slits (individual processing). Figure 3.1 shows the flow chart of the data reduction.

The common processing is the following:

Bad pixels and cosmic rays reduction While the OH airglow lines are easily seen in the raw data, you will additionally find some pixels with very high signals with a careful inspection. These pixels will be a source of noise when combining the frames, so they should be masked beforehand. The causes of such pixels are bad pixels intrinsic to a detector (pixels with extremely poor linearity, and pixels with zero or extremely high counts for all time) and cosmic rays in individual frames. The cosmic rays detected in each frame are combined into the known bad pixel mask; then the masked pixels are interpolated with the nearby pixel values.

A−B sky subtraction Because the night airglow lines, the dominant source of the near-infrared background emission, vary with time, the airglows are removed by subtracting a frame taken with a short interval, as I already mentioned. In this step, night airglows are roughly removed with the subtraction of a pair of A-B frames.

Flat fielding Inhomogeneous sensitivity arising from the detectors and the instrument optics, as well as the cutting error of the slit width, will be corrected by dividing the data with a flat frame. As described in the calibration data section, the flat frame is made by the data obtained with a flat light source, such as dome flat and flat probe.

Distortion correction Generally, distortion will be seen in the images focused on a detector, by passing through the optics of the telescope/instrument. This step corrects the distortion to some extent by using the distortion database which has been prepared for the imaging data. (The distortion will be further corrected in a latter step for straightening the OH lines.)

After the common processing, spectra of individual objects are extracted. Individual processing for the extracted spectra is the following:

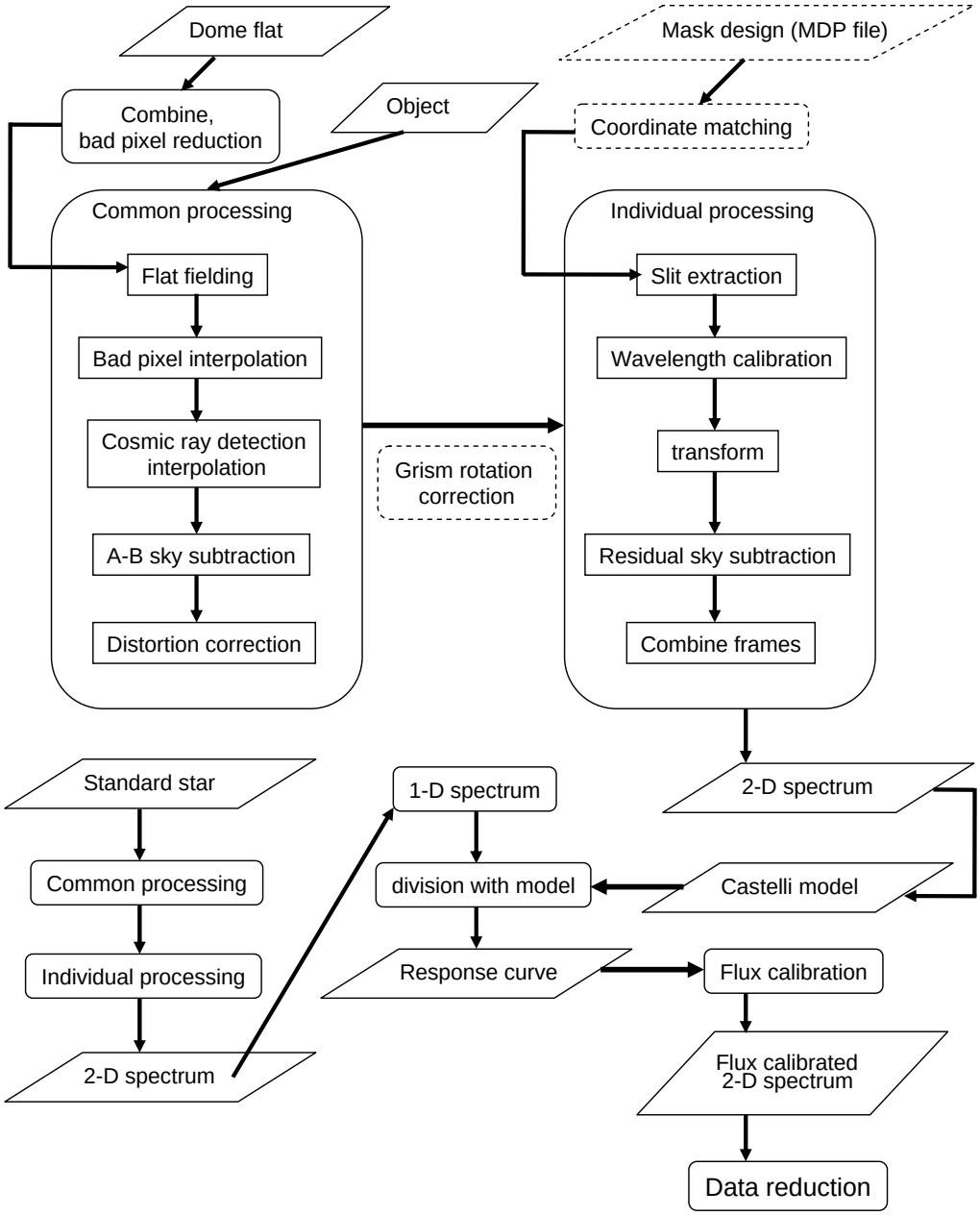

Figure 3.1: Flow chart of the MOIRCS MOS data reduction

Wavelength calibration By defining the first-order relation between the wavelength and the coordinate of the detectors, the images are transformed into a form where the X- and Y-axis corresponds to wavelength and spatial extent along the slit, respectively. OH lines are used for this purpose, because the wavelengths of the lines are well known.

Sky subtraction Although the background emission is roughly removed by the A−B subtraction in the common processing, there are some residual OH lines. These residuals are due to the time variability of the OH lines, which can not completely be removed by the A−B subtraction. In this step, the residual lines are removed by fitting along the spatial extent.

Combine All frames which were taken at the same field of view are combined in this step.

Flux calibration With a standard star data, counts of a two-dimensional spectrum are converted into the flux per wavelength (F_λ [erg sec^{-1} cm^{-2} Å^{-1}]).

3.2 MCSMDP

MCSMDP (MOIRCS MOS Data Pipelines) is a PyRAF script package for the MOIRCS MOS data reduction, developed by Tomohiro Yoshikawa (Kyoto Sangyo University, Koyama Astronomical Observatory), one of the MOIRCS developers. The package can be used for your research; it can be downloaded from the server of the Kyoto Sangyo University [1] or from the link on the Subaru Telescope's web site. Although the package was developed basically for a personal research and does not always result in appropriate reductions for all types of the data, it has been developed also for common use to some extent. If you use the package for your research and find some troubles or ideas for improvements, please let me know.

3.2.1 IRAF and PyRAF

IRAF (Image Reduction and Analysis Facility) is an astronomical data reduction/analysis software, which has been developed by NOAO (National Optical Astronomical Observatories) in the United States. It can be downloaded for free.[2] IRAF is equipped with many commands (tasks) running on its interactive command interface (cl), and has been used to develop data reduction scripts not only for the instruments on NOAO but also for instruments on worldwide telescopes.

PyRAF is a command language which enables IRAF tasks to run on Python, a general-purpose script language. It has been developed and distributed for free by NASA STScI (Space Telescope Science Institute).[3] PyRAF offers an user-friendly command line which is alternative to cl. Basically, all that can be done with IRAF are also available on PyRAF. Moreover, because PyRAF tasks can be described as the Python scripts, PyRAF can easily extend the utility of IRAF data reductions, by using generous Python libraries for scientific calculation.

[1] http://www.cc.kyoto-su.ac.jp/~tomohiro/MCSMDP/
[2] http://iraf.nao.ac.jp/iraf/web/
[3] http://www.stsci.edu/resources/software_hardware/pyraf

MCSMDP consists of many PyRAF tasks, and runs on the PyRAF command line. Current developing environment for MCSMDP are IRAF version 2.14.1 (Linux version) and PyRAF version 1.9.[4] Especially because PyRAF is updated frequently, I have been developing MCSMDP so that it can run on the latest version as possible. While MCSMDP has been developed under Ubuntu Linux 10.4, it is supposed to work on any environment where IRAF/PyRAF are enabled, including the data analysis environment at Mitaka.

3.2.2 Required software

Following softwares are required for the use of MCSMDP. Some developing environment packages may also be required, depending on your environment (e.g., *-dev packages for Ubuntu Linux); you should appropriately install the packages by using the package management system on your environment (e.g., apt, yum).

IRAF, Python You need to install the versions which are required for stsci_python.

stsci_python It is a group of python modules. It can be downloaded from the STScI web site. Please refer to the web site for the instruction of the installation. Including IRAF and Python, softwares necessary for PyRAF need to be installed.

ds9, xpa They are distributed at the web site[5] of Harvard-Smithsonian Center for Astrophysics. ximtools is not supported.

RO Python Package It is a Python module developed and distributed[6] by Dr.Russel Owen (University of Washington). It can be installed using easy_install, if Python setuptools are already installed.

Although the following softwares are not necessarily for the data reductions, they will be necessarily to use tasks for analysing emission lines.

matplotlib It is a python module for graph drawings. It is distributed at the official web site.[7] It is also included in many Linux distributions.

scipy It is a python module for scientific computing. It is distributed at the official web site.[8] It is also included in many Linux distributions.

ipython It is a program to provide an interactive shell for python. It can be used as a PyRAF front-end, providing richer functions than the PyRAF command line. It is distributed at the official web site.[9] It is also included in many Linux distributions.

R, rpy They are a free statistical analysis software and its python interface. They are distributed at their official web sites.[10][11] They are also included in many Linux distributions.

[4] Although IRAF version 2.15 was released on November 2011, its performance on PyRAF has not yet been officially supported.
[5] http://hea-www.harvard.edu/RD/ds9/
[6] http://www.astro.washington.edu/users/rowen/ROPackage/Overview.html
[7] http://matplotlib.sourceforge.net/
[8] http://www.scipy.org
[9] http://ipython.scipy.org/moin
[10] http://www.r-project.org/
[11] http://rpy.sourceforge.net/

3.2.3 Installation

First, MCSMDP (main script) and MDPDB (calibration database) are downloaded from the MCSMDP web site, then extracted to any directory.[12] Here I assume that the files are extracted just under the home directory.

```
$ wget http://www.cc.kyoto-su.ac.jp/~tomohiro/MCSMDP/MCSMDP_v1_1_2.tgz
$ wget http://www.cc.kyoto-su.ac.jp/~tomohiro/MCSMDP/MDPDB_v1_0_1.tgz
$ tar xvfz MCSMDP_v1_1_2.tgz
$ tar xvfz MDPDB_v1_0_1.tgz
```

Move to the MCSMDP directory, then run setup.sh.

```
$ cd MCSMDP_v1_1_2
$ ./setup.sh
```

In case the login shell is bash, ksh, sh, or zsh[13]

Add the following line at the end of the .bash_profile (in case of bash) or .profile (in case of ksh, sh, or zsh) file under the home directory:

```
source ${HOME}/MCSMDP_v1_1_2/etc/mcsmdp.sh
```

In case the login shell is tcsh, or csh

Add the following line at the end of the .cshrc file under the home directory:

```
source ${HOME}/MCSMDP_v1_1_2/etc/mcsmdp.csh
```

3.2.4 Preparation for the data analysis

Prepare a working directory, then launch the mcsmdp. On a directory where mcsmdp is launched for the first time, setting for the IRAF environment will be done on the directory. When a message asking if it is OK to run mkiraf appears, press the return key.[14] After a moment, PyRAF shell (-->) will start.

```
$ mcsmdp
mkiraf? (yes):
...
-->
```

After the PyRAF shell is started, load the mcsmdp tasks by inputting mcsmdp.[15]

[12] Hereafter a line starting with "$" represents a command input on the bash prompt.

[13] Here I assume the data analysis environment at the Astronomical Data Center. Please adopt an adequate setting depending on your actual environment.

[14] Please refer to an iraf document regarding mkiraf. Only if you don't want the automatic launching of mkiraf, reply with "no".

[15] With IRAF, a task will be automatically executed when IRAF is started, if the task is listed in the loginuser.cl. This function seems to be in fail in case of the PyRAF tasks.

```
--> mcsmdp
...
-->
```

Like cl in IRAF, PyRAF shell is operated by inputting a command interactively. While detailed explanation on how to work with PyRAF is beside the purpose of this text, Here I briefly summarize the basic operation which will be necessary for the exercise in the next section. For the details, please refer to "The PyRAF Tutrial"[16] distributed by STScI.

Before a task is executed, a list of parameters of the task will be shown in this text. For example, to display the parameter list of mdpdisplay task,

```
--> lpar mdpdisplay
         file = "HK500_MODS11-0390fl.fits" Image (list) to be loaded
       (frame = 1)                 Display frame to be loaded
  (multiframe = no)                Increment frame number for image list?
       (scale = "linear")          scale type
       (smode = "zscale")          scale mode
          (z1 = INDEF)             minimum greylevel to be displayed
          (z2 = INDEF)             maximum greylevel to be displayed
        (cmap = "BB")              color map
      (invert = no)                invert color map?
        (mode = "al")
```

The parameters should be edited so that their values are same as those shown in the text. To edit the parameters:

```
--> epar mdpdisplay
```

then, the parameter editor (figure 3.2) will be launched. Parameters whose names are in parentheses "()" should be edited. Like IRAF, parameters in parentheses are hidden parameters whose values are not necessarily specified explicitly at the time of the task execution from the PyRAF prompt and the value edited/saved with epar will be used. After the parameter edition, clicking the "Save & Quit" button on the upper menu of the editor will save the value and close the editor to come back to the PyRAF prompt.

Here I will summarize the functions of other buttons on the parameter editor, although they are not used in this text. "Execute" will directly execute the task using the parameter value listed in the editor. "Unlearn" will restore the edited values to the default values. "Cancel" will close the editor without saving the edited values. Although "Mdpdisplay Help" in the rightmost is to show the explanatory text of the task, the button does not work because the help document has not been prepared. In case of a usual IRAF task, the help document will be shown on a pop-up window.

A task should be executed on the PyRAF prompt. For those parameters which are not in parentheses in the parameter editor, their values need to be specified in the order of the list. Values of hidden parameters (i.e., those in parentheses) can also be explicitly specified as tentative ones (i.e., not saved) with the style "parameter=value":

```
--> mdpdisplay MCSA00057147.fits frame=2
```

[16] http://stsdas.stsci.edu/stsci_python_epydoc/docs/pyraf_tutorial.pdf

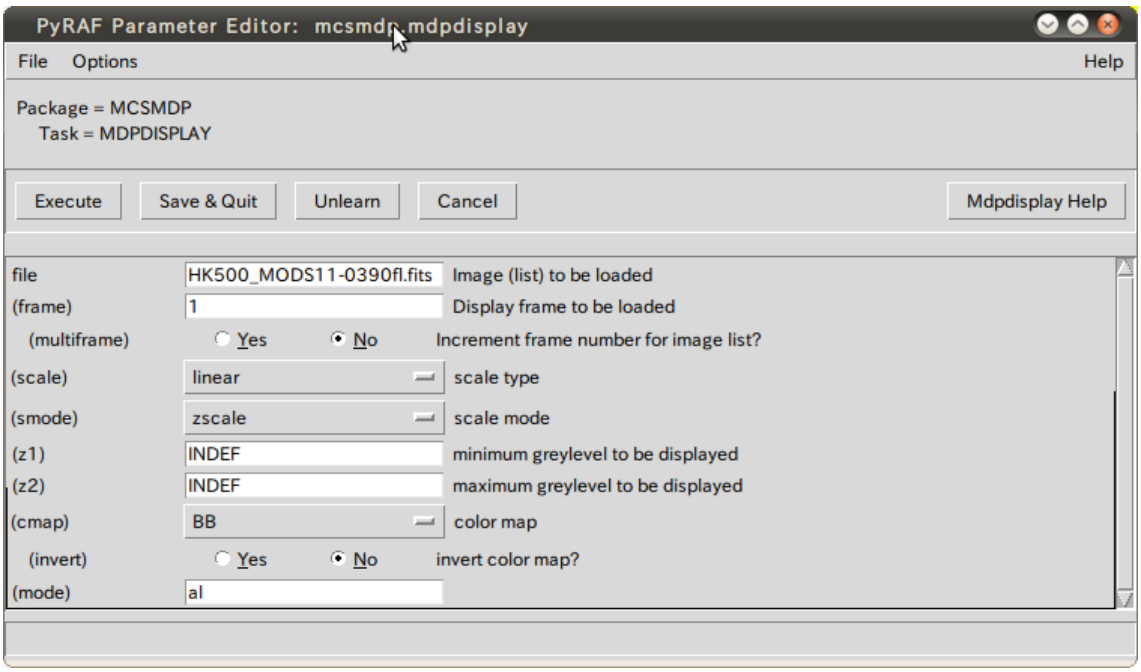

Figure 3.2: The parameter editor for mdpdisplay.

To quit mcsmdp, type ".exit"

```
--> .exit
$
```

Supplement: Launching with ipython mode

Adding the --ipython option to the mcsmdp task will launch the ipython shell instead of the PyRAF shell:

```
$ mcsmdp --ipython
...
In [1]:
```

If you choose to use the ipython shell mode, please add the following lines in the ~/.ipython/ipythonrc-pyraf file:

```
import_mod pylab
execfile pylabinit.py
```

If this file does not exist, please launch the mcsmdp with the ipython mode first with the above command, quit it with "Ctrl+D", then the file will be automatically created.

The ipython shell enables a use of python drawing tools such as matplotlib. However, the PyRAF shell is mainly used in this text, because the operation is similar to the original IRAF cl.

Chapter 4

Data Reduction (Practice)

4.1 Preparation of the data

Let's prepare a directory to store the data, and copy the sample data as follows: [1]

```
$ mkdir -p /home/subaru_tutor02/scratch/yourname_work
          # The working directory should be named by yourself.
$ cd /home/subaru_tutor02/scratch/yourname_work
$ tar jxf /home/subaru_tutor02/scratch/subaru_sample_data/MCSMDP_sample.tbz
$ cp MCSMDP_sample/*.* .
```

If you can not copy the sample data, download it from the following site.

```
$ wget http://www.cc.kyoto-su.ac.jp/~tomohiro/MCSMDP/MCSMDP_sample.tbz
```

The sample data consists of two types of files whose extensions are mdp and fits. The file whose extension is mdp is a text file called MDP file. A slit mask design is described in this file. The observer made the file and sent it to the observatory before the observation to make the slit mask. This file is also used to extract a spectrum from each slit at the time of the data reduction.

The file whose extension is fits is the data file called FITS file. Let's check the data by launching mcsmdp. It is convenient to use hselect task for this purpose:

```
$ mcsmdp
--> mcsmdp
--> hselect *.fits $I,OBS_MOD,DATA_TYP,OBJECT,DISPERSR yes
MCSA00057015.fits SPEC_MOS OBJECT DOMEFLAT HK500
MCSA00057016.fits SPEC_MOS OBJECT DOMEFLAT HK500
MCSA00057017.fits SPEC_MOS OBJECT DOMEFLAT HK500
...
MCSA00057114.fits SPEC OBJECT M53735(A0V:J8.9:H8.9:K8.9) HK500
MCSA00057116.fits SPEC OBJECT M53735(A0V:J8.9:H8.9:K8.9) HK500
MCSA00057147.fits SPEC_MOS OBJECT CDFN_MASK02 HK500
MCSA00057148.fits SPEC_MOS OBJECT CDFN_MASK02 HK500
MCSA00057149.fits SPEC_MOS OBJECT CDFN_MASK02 HK500
...
```

OBS-MODE means the observation mode. For the sample data, the observation mode is multi-object spectroscopic mode (SPEC_MOS) for most of the data. Data with single slit spectroscopic mode (SPEC) are those for a standard star. DATA-TYP means the type of the data, where DOMEFLAT represents the dome flat data. OBJECT means the object name. DISPERSR means disperser, where $HK500$ grism was used for the sample data.

Before entering into the reduction steps, let's make lists of the files to classify the data. The hselect task is also used for this purpose:[2]

```
--> hselect MCSA*.fits $I
 "OBJECT = 'DOMEFLAT' & @'DET_ID' = 1" > flat1.lst
```

[1] The sample data can also be downloaded from http://www.cc.kyoto-su.ac.jp/~tomohiro/MCSMDP/MCSMDP_sample.tbz.

[2] Although a command line is folded into multiple lines in the following example due to limitations of space, please type the command in a single line in reality. This rule will be adopted in the following text.

```
--> hselect MCSA*.fits $I
 "OBJECT = 'DOMEFLAT' & @'DET-ID' = 2" > flat2.lst
--> hselect MCSA*.fits $I
 "OBJECT = 'CDFN_MASK02' & @'DET-ID' = 1 & K_DITCNT = 1" > obj1a.lst
--> hselect MCSA*.fits $I
 "OBJECT = 'CDFN_MASK02' & @'DET-ID' = 1 & K_DITCNT = 2" > obj1b.lst
--> hselect MCSA*.fits $I
 "OBJECT = 'CDFN_MASK02' & @'DET-ID' = 2 & K_DITCNT = 1" > obj2a.lst
--> hselect MCSA*.fits $I
 "OBJECT = 'CDFN_MASK02' & @'DET-ID' = 2 & K_DITCNT = 2" > obj2b.lst
```

The word "flat1" and "flat2" represents the dome flats taken with the Detector 1 and the Detector 2, respectively; "obj1a" and "obj1b" represents the object frames taken with the Detector 1, at position A, and the position B, respectively. In the same way, "obj2a" and "obj2b" represent the object frames taken with the Detector 2. A list of standard star frames is left out, because they are only two frames in the sample data. These lists are used for the data reduction below. In this exercise, data of the Detector 1 will be reduced, because the target object is on the Detector 1. You can try the reduction of the Detector 2 data same way.

4.2 Common processing

4.2.1 Making the dome flat and the flat fielding

First, let's make a dome flat from the dome flat data.

```
--> epar imcombine
```

Parameter of imcombine should be edited as follows:

```
      input =              List of images to combine
     output =              List of output images
   (headers = "")          List of header files (optional)
   (bpmasks = "")          List of bad pixel masks (optional)
  (rejmasks = "")          List of rejection masks (optional)
 (nrejmasks = "")          List of number rejected masks (optional)
  (expmasks = "")          List of exposure masks (optional)
    (sigmas = "")          List of sigma images (optional)
     (imcmb = "$I")        Keyword for IMCMB keywords
   (logfile = "STDOUT")    Log file

   (combine = "median")    Type of combine operation
    (reject = "sigclip")   Type of rejection
   (project = no)          Project highest dimension of input images?
   (outtype = "real")      Output image pixel datatype
 (outlimits = "")          Output limits (x1 x2 y1 y2 ...)
   (offsets = "none")      Input image offsets
  (masktype = "none")      Mask type
```

```
 (maskvalue = "0")          Mask value
     (blank = 0.0)          Value if there are no pixels

     (scale = "exposure") Image scaling
      (zero = "none")       Image zero point offset
    (weight = "none")       Image weights
   (statsec = "")           Image section for computing statistics
   (expname = "EXPTIME")    Image header exposure time keyword

(lthreshold = INDEF)        Lower threshold
(hthreshold = INDEF)        Upper threshold
      (nlow = 1)            minmax: Number of low pixels to reject
     (nhigh = 1)            minmax: Number of high pixels to reject
     (nkeep = 1)            Minimum to keep (pos) or maximum to reject (neg)
     (mclip = yes)          Use median in sigma clipping algorithms?
    (lsigma = 3.0)          Lower sigma clipping factor
    (hsigma = 3.0)          Upper sigma clipping factor
   (rdnoise = "0.")         ccdclip: CCD readout noise (electrons)
      (gain = "1.")         ccdclip: CCD gain (electrons/DN)
    (snoise = "0.")         ccdclip: Sensitivity noise (fraction)
  (sigscale = 0.1)          Tolerance for sigma clipping scaling corrections
     (pclip = -0.5)         pclip: Percentile clipping parameter
      (grow = 0.0)          Radius (pixels) for neighbor rejection
```

then, run the imcombine task:

```
--> imcombine @flat1.lst HK500_CDFN2_Domeflat1.fits
```

Although the resulting file could be used for the flat fielding as it is, the signals of the divided object frames would have been smaller values. To avoid this, the flat frame is divided by a number so that its signals become nearly one:

```
--> imarith HK500_CDFN2_Domeflat1.fits / 10000.
 HK500_CDFN2_Domeflat1.fits
```

Let's divide the object frames by the flat frame. I choose to add "fl" for the prefix of the resulting files here:

```
--> !sed 's/\(.*\)/fl\1/' obj1a.lst > flobj1a.lst
--> imarith @obj1a.lst / HK500_CDFN2_Domeflat1.fits @flobj1a.lst
```

4.2.2 Bad pixels and cosmic rays detection/interpolation

To detect cosmic rays, a task craverage is used. This task will detect cosmic rays as pixels whose signals are higher than those of the surrounding pixels, then make a new mask file by adding the detected pixels to an existing bad pixel mask. A bad pixel mask is an image file whose signals are either 1 or 2, where 1 represents a bad pixel. Please use the help command for the details of the task.

First, let's make lists of masks of bad pixels plus cosmic rays. Although any names should work, here I choose to use the same object frame names whose extension is changed to ".pl", and the directory name to save the lists is "BPM". To make the lists, I use a UNIX command, sed. The BPM directory is also created:

```
—> !sed 's/\(.*\)\.fits/BPM\/\1\.pl/' obj1a.lst > bpm1a.lst
—> !sed 's/\(.*\)\.fits/BPM\/\1\.pl/' obj1b.lst > bpm1b.lst
—> mkdir BPM
```

Known bad pixel masks have been prepared by the observatory, which will be used in this exercise. The bad pixel masks for this sample data is also included in the MCSMDP package — mdpdb$bpm/nlbpm1_FF64r.fits and mdpdb$bpm/nlbpm2_FF64r.fits for the Detector 1 and the Detector 2, respectively. The bad pixel masks are copied with the file names prepared above. Using a bad pixel mask, the craverage task detects cosmic rays from the pixels other than the known bad pixels, then add the detected pixels to the original bad pixel mask.

```
—> imcopy mdpdb$bpm/nlbpm1_FF64r.fits ,mdpdb$bpm/nlbpm1_FF64r.fits ,
mdpdb$bpm/nlbpm1_FF64r.fits ,mdpdb$bpm/nlbpm1_FF64r.fits @bpm1a.lst
—> imcopy mdpdb$bpm/nlbpm1_FF64r.fits ,mdpdb$bpm/nlbpm1_FF64r.fits ,
mdpdb$bpm/nlbpm1_FF64r.fits ,mdpdb$bpm/nlbpm1_FF64r.fits @bpm1b.lst

—> epar craverage
```

Below are the parameters of craverage:

```
    input = ""      List of input images
   output = ""      List of output images
  (crmask = "")     List of output cosmic ray and object masks
 (average = "")     List of output block average filtered images
   (sigma = "")     List of output sigma images

    (navg = 5)     Block average box size
    (nrej = 30)    Number of high pixels to reject from the average
    (nbkg = 5)      Background annulus width
    (nsig = 10)     Box size for sigma calculation
    (var0 = 0.0)    Variance coefficient for DN^0 term
    (var1 = 0.0)    Variance coefficient for DN^1 term
    (var2 = 0.0)    Variance coefficient for DN^2 term

   (crval = 1)      Mask value for cosmic rays
  (lcrsig = 100.0) Low cosmic ray sigma outside object
  (hcrsig = 10.0)  High cosmic ray sigma outside object
  (crgrow = 0.0)    Cosmic ray grow radius

  (objval = 0)      Mask value for objects
 (lobjsig = 10.0)  Low object detection sigma
 (hobjsig = 5.0)    High object detection sigma
 (objgrow = 0.0)    Object grow radius
```

then, run the craverage:

```
--> craverage @flobj1a.lst "" crmask=@bpm1a.lst average="" sigma=""
--> craverage @flobj1b.lst "" crmask=@bpm1b.lst average="" sigma=""
```

Next, bad pixels listed in the new bad pixel masks are interpolated with fixpix task. In this exercise, the linear-interpolation is only along the Y-axis, i.e., the spatial extent along which the spectrum will be binned to measure the flux. Interpolation along the X-axis is avoided so that the information along the wavelength should be kept.

Let's make lists of the input files for fixpix in the same way described above. Here I choose to add "cr" for the prefix of the files:

```
--> !sed 's/\(.*\)/cr\1/' flobj1a.lst > crobj1a.lst
--> !sed 's/\(.*\)/cr\1/' flobj1b.lst > crobj1b.lst
```

Because fixpix will overwrite the input files, the flatfielded files should be copied to the input files in advance:

```
--> imcopy @flobj1a.lst  @crobj1a.lst
--> imcopy @flobj1b.lst  @crobj1b.lst

--> epar fixpix
```

Let's edit the parameters of fixpix using epar:

```
      images =                 List of images to be fixed
       masks =                 List of bad pixel masks
(linterp = "INDEF")            Mask values for line interpolation
(cinterp = "1")                Mask values for column interpolation
(verbose = no)                 Verbose output?
 (pixels = no)                 List pixels?
```

then run the fixpix task:

```
--> fixpix @crobj1a.lst @bpm1a.lst
--> fixpix @crobj1b.lst @bpm1b.lst
```

Supplement: Cosmic rays and the bad pixel reduction with MCSMDP

Because reduction procedures of cosmic rays and bad pixels are complicated, I have explained a simple way in the above exercise. In MCSMDP, a task named crrejection is prepared to reduce cosmic rays and bad pixels in the following way:

1. Detect cosmic rays from the data (Cosmicray1)
2. Detect cosmic rays from the data divided by its A-B pair (i.e., A/B in case of A frame) (Cosmicray2)
3. Make a mask of the bad pixels plus cosmic rays, by making sum-set (OR) operation of Cosmicray1, Cosmicray2, and the bad pixel mask.
4. Interpolate the masked pixels along the spatial extent.

5. Interpolate the pixels in the quadrant boundary of the detectors along the wavelength.

The Cosmicray2 frames are additionally created because cosmic rays which are buried in strong night airglow emission lines are difficult to be detected from the raw frames. The division by a neighbouring frame will suppress the airglow lines, enabling easier detection of cosmic rays in the emission lines. Although the number of detected cosmic rays actually increases with the addition of Cosmicray2, it is not yet sufficient detections. Improving the method and tuning the parameters of cosmic ray detection is one of the challenges to be solved for the MOIRCS data reduction.

There are gaps of 1 pixel width at the boundary of the detector quadrants and the vertical gap can not be interpolated along the Y-axis. So, only these pixels are interpolated along the wavelength.

The parameters for crrejection are set as follows:

```
 inimage1 = ""          input frame at A position
 inimage2 = ""          input frame at B position
outimage1 = ""          output frame at A position
outimage2 = ""          output frame at B position
      bpm = ""          badpixel mask
    (navg = 15)         Block average box size
    (nrej = 100)        Number of high pixels to reject from the average
    (nbkg = 5)          Background annulus width
    (nsig = 10)         Box size for sigma calculation
    (var0 = 0.0)        Variance coefficient for DN^0 term
    (var1 = 0.0)        Variance coefficient for DN^1 term
    (var2 = 0.0)        Variance coefficient for DN^2 term
  (lcrsig = 100.0)      Low cosmic ray sigma outside object
  (hcrsig = 10.0)       High cosmic ray sigma outside object
  (crgrow = 0.0)        Cosmic ray grow radius
  (bpmdir = "BPM")      directory name to store bad pixel masks
```

then, crrejection is executed in the follwoing (Here, the output file names are "cr2*" so that the previous files are not overwritten. Bad pixel masks will be saved below "BPM2" directory.):

```
--> !sed 's/\(.*\)/cr2\1/' flobj1a.lst > cr2obj1a.lst
--> !sed 's/\(.*\)/cr2\1/' flobj1b.lst > cr2obj1b.lst
--> crrejection @flobj1a.lst @flobj1b.lst @cr2obj1a.lst @cr2obj1b.lst
 mdpdb$bpm/nlbpm1_FF64r.fits bpmdir="BPM2"
```

4.2.3 A−B Sky subtraction

Next, let's make the subtraction between A-B pairs to remove the sky emission. A task imarith will be used for this purpose. Here I choose to add "ab" for the prefix of the resulting files:

```
--> !sed 's/\(.*\)/ab\1/' crobj1a.lst > abobj1a.lst
--> imarith @crobj1a.lst - @crobj1b.lst @abobj1a.lst
```

4.2.4 Distortion correction

Let's correct the distortion by adopting a distortion database prepared for imaging data.

—> epar geotran

The parameters for geotran task should be set as follows:

```
        input =                 Input data
       output =                 Output data
     database =                 Name of GEOMAP database file
   transforms =                 Names of coordinate transforms in
                                database file
   (geometry = "geometric")     Transformation type (linear,geometric)
        (xin = INDEF)           X origin of input frame in pixels
        (yin = INDEF)           Y origin of input frame in pixels
     (xshift = INDEF)           X origin shift in pixels
     (yshift = INDEF)           Y origin shift in pixels
       (xout = INDEF)           X origin of output frame in
                                reference units
       (yout = INDEF)           Y origin of output frame in
                                reference units
       (xmag = INDEF)           X scale of input picture in pixels per
                                reference unit
       (ymag = INDEF)           Y scale of input picture in pixels per
                                reference unit
   (xrotation = INDEF)          X axis rotation in degrees
   (yrotation = INDEF)          Y axis rotation in degrees
       (xmin = INDEF)           Minimum reference x value of output
                                picture
       (xmax = INDEF)           Maximum reference x value of output
                                picture
       (ymin = INDEF)           Minimum reference y value of output
                                picture
       (ymax = INDEF)           Maximum reference y value of output
                                picture
      (xscale = 1.0)            X scale of output picture in reference
                                units per pixel
      (yscale = 1.0)            Y scale of output picture in reference
                                units per pixel
       (ncols = INDEF)          Number of columns in the output picture
      (nlines = INDEF)          Number of lines in the output picture
     (xsample = 1.0)            Coordinate surface sampling interval
                                in x
     (ysample = 1.0)            Coordinate surface sampling interval
                                in y
 (interpolant = "linear")       Interpolant
    (boundary = "constant")     Boundary extension (nearest,constant,
```

```
                                reflect,wrap)
    (constant = 0.0)            Constant boundary extension
(fluxconserve = yes)            Preserve image flux?
     (nxblock = 512)            X dimension of working block size in
                                pixels
     (nyblock = 512)            Y dimension of working block size in
                                pixels
     (verbose = yes)            Print messages about the progress of
                                the task
```

then, run geotran. Here I choose to add "gc" for the prefix of the resulting files:

```
--> !sed 's/\(.*\)/gc\1/' abobj1a.lst > gcobj1a.lst
--> geotran @abobj1a.lst @gcobj1a.lst
 mdpdb$geomap/mcsdistcrr1_feb07new.dbs mcsdistcrr1_feb07new.gmp
```

Here, distortion database included in MDPDB are used for the geotran task. mdpdb$geomap/mcsdistcrr1_feb07new.dbs, and mcsdistcrr1_feb07new.gmp are for the Detector 1, and mdpdb$geomap/mcsdistcrr2_feb07new.dbs, and mcsdistcrr2_feb07new.gmp are for the Detector 2, where *.dbs and *.gmp is the database and the transform, respectively. If you need a distortion database for the data obtained in other observation period, please use the database included in MCSRED.

Supplement: Correction of a grism rotation

Even with the wavelength calibrated data, the continuum spectra may not be exactly aligned with the X-axis. This is because the dispersion of the grism was not accurately aligned with the X-axis. As the MOIRCS grisms are installed on a rotatable filter turret, its dispersion direction may change depending on the reproducibility of the turret position. Practically, it is known that there is an error of about 0.3° for the turret position. The angle should be different between the Detector 1 and the Detector 2, because each detector has its own filter turrets for grisms. The grism rotation is not a big problem in measuring the flux of emission lines as in this exercise. In case of measuring a faint continuum, however, it is better to correct the grism rotation.

Determination of the rotation angle is very difficult in a precise sense, because the rotation angle could change every time when the grism is exchanged. If there is a detectable continuum emission (such as from a bright star) on the slit mask, this should be used to measure the rotation angle. A standard star frame which was obtained close in time to the target could work to some extent, although the grism turret moved between the standard and the target observations.

If a grism rotation is measured, the entire frame will be rotated with a task mdprotate (below is an example of 0.3° rotation):

```
--> !sed 's/\(.*\)/r\1/' gcobj1a.lst > rgcobj1a.lst
--> mdprotate @gcobj1a.lst @rgcobj1a.lst 0.3
```

4.3 Extraction of a slit

In the following sections, the reduction will be done for each target. A slit data for an individual target will be extracted to a separate file.

Let's plot the MDP file on a distortion corrected image, using a MCSMDP task maskplot :

```
--> maskplot CDFN_MASK02.mdp image=gcabcrflMCSA00057147.fits raw+
```

Because the coordinate system is different between the mask design file and the observation data, maskplot transforms the coordinate so that it matches roughly. Although the plot can not reproduce the slit positions precisely, it is still convenient to tell which spectrum corresponds to which target.

In this exercise, data of an object MODS11-0390 will be reduced. After making input/output file lists, the target images will be copied to the output files. Please check the x-, and y-coordinates of the target by yourself. On this occasion, please take care neither to cut the slit image too much nor to include extraneous data in the image.

```
--> !sed 's/\(.*\)\.fits/\1_MODS11-0390\.fits/' gcobj1a.lst >
 gcMODS11-0390.lst
--> !sed 's/\(.*\)/\1[*,1755:1840]/' gcobj1a.lst > cut.lst
--> imcopy @cut.lst @gcMODS11-0390.lst
```

4.4 Individual processing

4.4.1 Wavelength calibration

Here, relation between the x-coordinate and the wavelength will be determined. OH night airglow lines are used for this purpose, because they are simultaneously observed with the target over the near-infrared wavelengths. Because OH lines are removed by the A−B subtraction in the reduced target frames, the sky spectra are created in the following way:

```
--> !sed 's/\(.*\)/gcsky\1/' crobj1a.lst > gcsky1a.lst
--> geotran @crobj1a.lst @gcsky1a.lst
 mdpdb$geomap/mcsdistcrr2_feb07new.dbs mcsdistcrr2_feb07new.gmp
--> !sed 's/\(.*\)\.fits/\1_MODS11-0390\.fits/' gcsky1a.lst
 > gcskyMODS11-0390.lst
--> !sed 's/\(.*\)/\1[*,1755:1840]/' gcsky1a.lst > cut.lst
--> imcopy @cut.lst @gcskyMODS11-0390.lst
```

To identify the OH emissions, a task identify will be used. The task will plot a spectrum at a certain line, find emission lines and identify their wavelengths using a list of wavelengths of the emission lines. Once several emission lines are identified manually, rest of the emission lines will be automatically detected and identified with a fitting algorithm.

```
--> epar identify
```

The parameters of identify are following:

```
       images =                    Images containing features to be
                                   identified
        crval =                    Approximate coordinate (at reference
                                   pixel)
        cdelt =                    Approximate dispersion
    (section = "middle line")      Section to apply to two dimensional
                                   images
   (database = "database")         Database in which to record feature
                                   data
(coordlist = "mdpdb$ohlist/list_NS_HK500") User coordinate list
      (units = "")                 Coordinate units
       (nsum = "20")               Number of lines/columns/bands to
                                   sum
                                   in 2D images
      (match = -3.0)               Coordinate list matching limit
(maxfeatures = 50)                 Maximum number of features for
                                   automatic identification
     (zwidth = 100.0)              Zoom graph width in user units
      (ftype = "emission")         Feature type
     (fwidth = 8.0)                Feature width in pixels
    (cradius = 5.0)                Centering radius in pixels
  (threshold = 0.0)                Feature threshold for centering
     (minsep = 2.0)                Minimum pixel separation
   (function = "chebyshev")        Coordinate function
      (order = 4)                  Order of coordinate function
     (sample = "*")                Coordinate sample regions
    (niterate = 10)                Rejection iterations
 (low_reject = 3.0)                Lower rejection sigma
(high_reject = 3.0)                Upper rejection sigma
       (grow = 0.0)                Rejection growing radius
  (autowrite = no)                 Automatically write to database
   (graphics = "stdgraph")         Graphics output device
     (cursor = "")                 Graphics cursor input
    (aidpars = "")                 Automatic identification algorithm
                                   parameters
```

The parameter "section" represents the line to be plotted, and "nsum" represents the number of the sum along the column. The "coordlist" parameter represents the list of OH emissions to be used for the wavelength identification. The MCSMDP package include a list of OH emission which were selected from the airglow atlas of Rousselot et al. (2000) [3] (figure 4.1) for the calibration of $HK500$ grism data.

Let's run the identify task:

```
--> identify gcflskycrMCSA00057147_MODS11-0390.fits
```

A graphic window (figure 4.2) appears to process the data interactively. The horizontal axis represents the x-coordinate, and the vertical axis represents the signals near the image

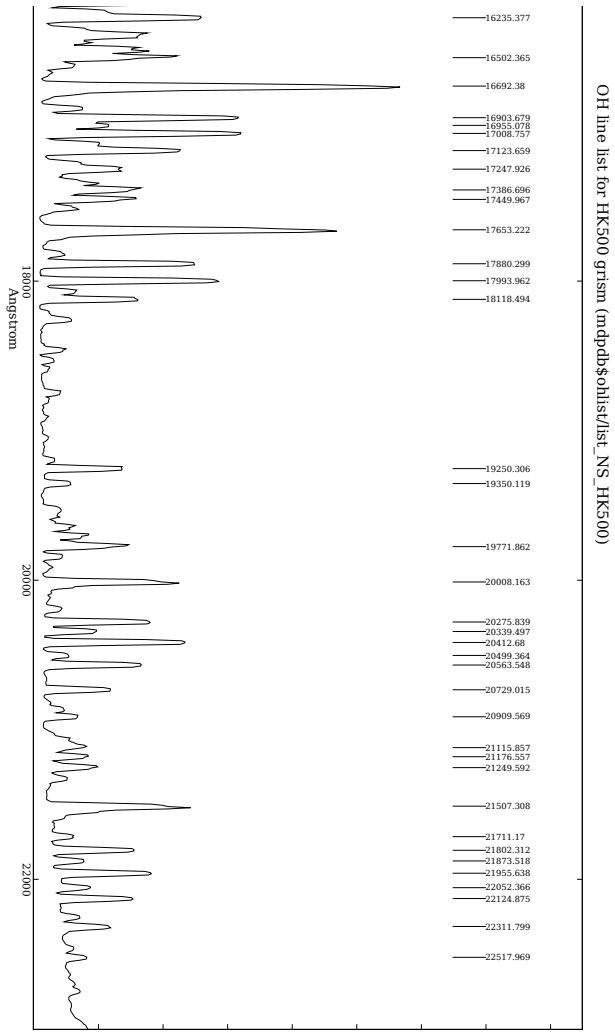

Figure 4.1: Atlas of OH airglow lines for the $HK500$ grism data.

center summed up along the column of 20 pixel width. Table 4.1 summarizes major functions executed from a key command.

Table 4.1: Major command keys for the identify task

Key	Action
m	Find a emission line near the cursor and identify its wavelength.
d	Delete an identified line near the cursor.
c	Show the wavelength of an identified line near the cursor.
w	Input a window command (see below) after the space key.
f	Enter into the fitting mode (see text).
l	Automatically find and identify emission lines from the line list.
q	Quit the identify task.
?	Show the command list in the terminal.
	Window command
b	Specify the lower edge of the plot with the cursor position.
t	Specify the upper edge of the plot with the cursor position.
x	Zoom along the X-axis.
y	Zoom along the Y-axis.
?	Show the list of window commands in the terminal.

First, remarkable OH emission lines should be identified by selecting a line with "m" key; about five OH lines should be selected from figure 4.1. The brightest four OH lines in the H band should be avoided because their peak counts are usually saturated. Note that the wavelength decreases as the x-coordinate increases in the Detector 1, which is reduced in this exercise. Press "m" key near one of the selected emission lines, then you will be asked to input the wavelength of the emission line. You don't have to input the number after the decimal point for the wavelength, because the task will select the closest wavelength from the line list.

After repeating the procedure for about five OH emission lines, press "f" key to enter into the coordinate fitting mode (figure 4.3). In figure 4.3, the horizontal axis and the vertical axis represents the wavelength, and the residuals from the fitting function, where 3rd-order Chebyshev polynomial is adopted here.[3] The order and the fitting function can be changed here interactively. This coordinate fitting mode is commonly used in other IRAF tasks which involve the polynomial fitting. Table 4.2 summarizes major key commands used in the fitting mode.

[3]Although the *order* =4 in the parameter list, the mathematical order should be $order - 1$, because the IRAF *order* does not mean the order but the number of terms of the polynomial.

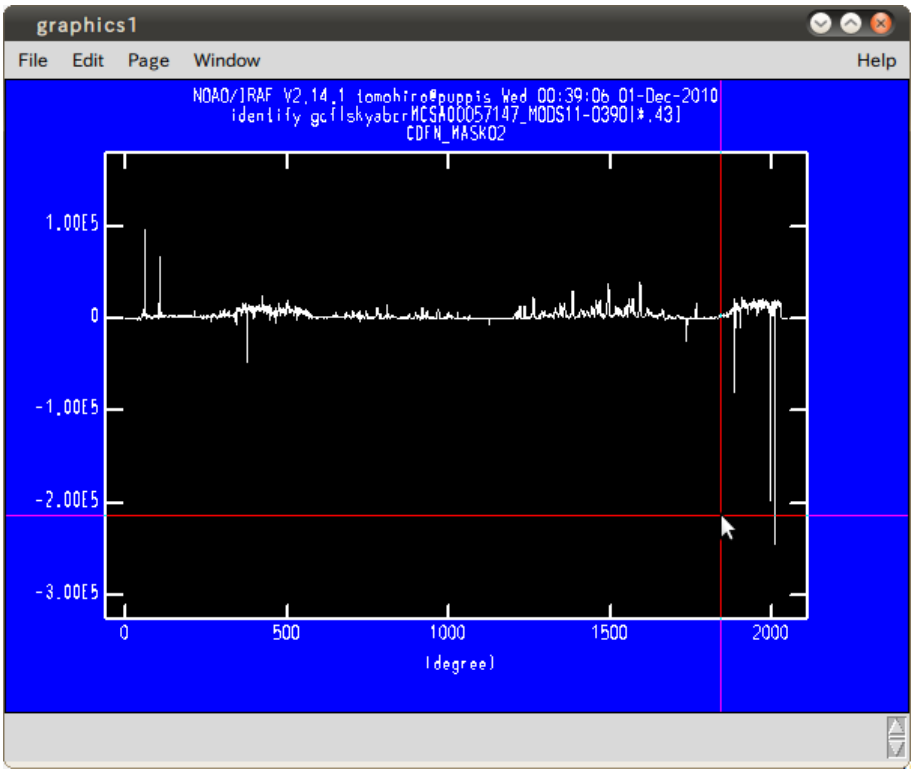

Figure 4.2: Graphic window of the identify task.

Table 4.2: Major command keys for the fitting mode of the identify task

Key	Action
h,i,j,k,l	Change the plot along the X-, Y-axis.。
f	Re-fit the data and display the fitting result.
d	Remove a point near the cursor from the fitting data.
u	Re-include a removed point to the fitting.
c	Show the coordinate of a point near the cursor.
s	Specify the sample region used for the fitting.
z	De-select the sample region.
:order n	Change the order of the fitting to $n-1$.
:function $functionname$	Change the polynomial function used for the fitting.
:niterate	Set the number of sigma rejection.
:low_reject	Set the lower threshold of the sigma rejection.
:high_reject	Set the higher threshold of the sigma rejection.
q	Exit from the fitting mode.
?	Show the list of fitting commands in the terminal.

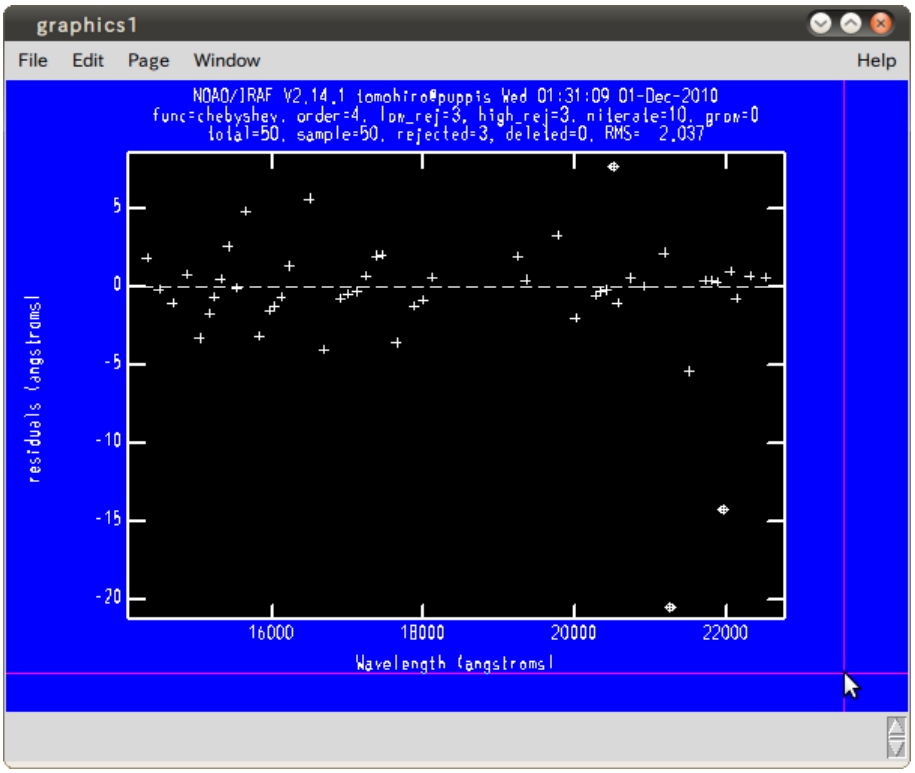

Figure 4.3: Fitting mode in the identify task.

At this time, pressing "q" key calls back the original graphical window, but its x-coordinate now represents the wavelength which was calculated from the fitting function. Then pressing "l" key will automatically identify rest of all emission lines in the line list.

Let's enter into the fitting mode again with the "f" key, and check if the fitting is good. In case of the $HK500$ grism, rms of a few Å is sufficient, because the dispersion is 7.72 [Å/pixel]. Exit the fitting mode with "q", then press "q" again to quit the identify task. At this point, you will be asked:

```
Write feature data to the database (yes)?
```

then, press the return key. This will save the information on the relation between the identified emissions and the coordinate to the file under the database directory (database/idgcskycr-flMCSA00057147_MODS11-0390).

With the above procedure, the relation between the pixel coordinate and the wavelength has been determined for the lines near the image center. However, it is not always true that the relation can also be applied to other lines of the spectral image. The spectrum is curved due to the distortion from the optics as well as to the relative rotation between the slit and the detector.

A task reidentify is used to automatically identify the OH emissions in the upper/lower lines of the spectral image, based on the fitting database at the center lines created by the identify task.

—> epar reidentify

Let's set the parameters for reidentify, then run the task:

```
    reference =                  Reference image
       images =                  Images to be reidentified
       answer = "yes"            Fit dispersion function
                                 interactively?
        crval =                  Approximate coordinate (at reference
                                 pixel)
        cdelt =                  Approximate dispersion
(interactive = "no")             Interactive fitting?
    (section = "middle line")    Section to apply to two dimensional
                                 images
     (newaps = yes)              Reidentify apertures in images not in
                                 reference?
   (override = yes)              Override previous solutions?
      (refit = yes)              Refit coordinate function?

      (trace = yes)              Trace reference image?
       (step = "20")             Step in lines/columns/bands for
                                 tracing an image
       (nsum = "20")             Number of lines/columns/bands to sum
      (shift = "0.")             Shift to add to reference features
                                 (INDEF to search)
     (search = 0.0)              Search radius
      (nlost = 10)               Maximum number of features which may
                                 be lost

    (cradius = 5.0)              Centering radius
  (threshold = 0.0)              Feature threshold for centering
(addfeatures = no)               Add features from a line list?
  (coordlist = "mdpdb$ohlist/list_NS_HK500") User coordinate list
      (match = -3.0)             Coordinate list matching limit
(maxfeatures = 50)               Maximum number of features for
                                 automatic identification
     (minsep = 2.0)              Minimum pixel separation

   (database = "database")       Database
   (logfiles = "logfile")        List of log files
   (plotfile = "")               Plot file for residuals
    (verbose = yes)              Verbose output?
   (graphics = "stdgraph")       Graphics output device
```

```
      (cursor = "")              Graphics cursor input

     (aidpars = "")              Automatic identification algorithm
                                 parameters
```

```
--> reidentify gcflskycrMCSA00057147_MODS11-0390.fits
 gcflskycrMCSA00057147_MODS11-0390.fits
```

The reidentify task sums up the upper/lower lines from the center at an interval of 10 pixels, identifies the OH emissions in each summed line, then saves the results to the database.[4] Now the database keeps the relation between the wavelength and the x-, y-coordinates at the OH emission lines. A two-dimensional fitting to these points will relate wavelength to the X-axis and spatial extent along the slit to the Y-axis for the entire spectral image.

The fitting will be done with a task named fitcoords.

```
--> epar fitcoords
```

The parameters for fitcoords are following:

```
      images =                  Images whose coordinates are to be fit
    (fitname = "")              Name for coordinate fit in the
                                database
(interactive = yes)             Fit coordinates interactively?
    (combine = no)              Combine input coordinates for a
                                single fit?
   (database = "database")      Database
  (deletions = "deletions.db")  Deletion list file (not used if null)
   (function = "chebyshev")     Type of fitting function
     (xorder = 4)               X order of fitting function
     (yorder = 3)               Y order of fitting function
   (logfiles = "STDOUT,logfile") Log files
   (plotfile = "plotfile")      Plot log file
   (graphics = "stdgraph")      Graphics output device
     (cursor = "")              Graphics cursor input
```

For function and xorder parameters, those values which resulted in good fitting in identify/reidentify tasks are used. Let's run the fitcoords,

```
--> fitcoords gcflskycrMCSA00057147_MODS11-0390
```

then a fitting graph similar to the identify task appears, but some fitting commands are different from those used in the identify task. Table 4.3 summarizes the major commands used in the fitcoords task. In fitcoords, the fitting results are checked while switching axis of the data (X, Y, and residuals) on the plot, and fitting parameters can be modified if necessary.

[4] You can see each process by setting a parameter interactive=yes.

Table 4.3: Major command keys with fitcoords task

Key	Action
x*[key]*,y*[key]*	x, y axes of the plot are changed to the *key* values (see below).
r	Redraw the plot.
f	Fit with the given parameters.
:order n	Change the order of the fitting to $n-1$.
:function *functionname*	Change the polynomial function used for the fitting.
q	Quit fitcoords.
?	Show the list of fitting commands in the terminal.
	[key] value
x	X-coordinate.
y	Y-coordinate.
z	Wavelength of the identified line.
s	Wavelength from the fitting.
r	Residual of the fitting, (s-z).

For example, typing "xxyy" and following "r" will show a map of identified lines along the x- and y-coordinates. From the plot, you can see how the emission lines were identified with the identify/reidentify tasks. Please confirm that the identified lines are distributed throughout the entire image.

To check the fitting residuals as a function of x-coordinate, type "xxyr" and "r", then the Y-axis changes to the residuals. In the course of changing the xorder of the fitting, please check not only the rms of the residuals but also if there is a global pattern in the fitting. For example, if the global pattern is quadratic, the pattern can be erased by increasing the order of the fitting by two. You should select the lowest fitting order which does not produce a global pattern. In the same way, fitting residuals as a function of y-coordinate can be checked by typing "xyyr" and "r". When you finish checking the result, type q to quit the fitcoords, then the results are saved to the database.

So far, fitting function for transform has been derived using the night airglow emission lines in the first one frame (MCSA00057147). In case of MOIRCS, it is known that the mask position relative to the detectors can be shifted up to about 3 pixels during the observation due to the flexure of the instrument. So, in a strict sense, the airglow lines should be identified for all the frames to adjust the shift.[5] In this exercise, because number of frames are not large, distortion correction for all the frames will be done using the fitting result of the first one frame.

A task named transform is used to transform the images based on the database by the fitcoords task.

```
--> epar transform
```

Parameters for transform are set as follows:

```
input =                    Input images
```

[5]MCSMDP will automatically do this task.

```
      output =                  Output images
    fitnames =                  Names of coordinate fits in the
                                database
     (minput = "")              Input masks
    (moutput = "")              Output masks
   (database = "database")      Identify database
(interptype = "linear")         Interpolation type
         (x1 = INDEF)           Output starting x coordinate
         (x2 = INDEF)           Output ending x coordinate
         (dx = INDEF)           Output X pixel interval
         (nx = INDEF)           Number of output x pixels
       (xlog = no)              Logarithmic x coordinate?
         (y1 = INDEF)           Output starting y coordinate
         (y2 = INDEF)           Output ending y coordinate
         (dy = INDEF)           Output Y pixel interval
         (ny = INDEF)           Number of output y pixels
       (ylog = no)              Logarithmic y coordinate?
       (flux = yes)             Conserve flux per pixel?
      (blank = INDEF)           Value for out of range pixels
(logfiles = "STDOUT,logfile") List of log files
```

then, run the transform task (Here I choose to add "tr" for the prefix of the resulting files):

```
--> !sed 's/\(.*\)/tr\1/' gcMODS11-0390.lst > trMODS11-0390.lst
--> transform @gcMODS11-0390.lst @trMODS11-0390.lst
 gcflskycrMCSA00057147_MODS11-0390
```

Let's compare the images before and after the transform task by displaying them on ds9 (figure 4.4). You can see that while airglow lines in the original image are tilted, the airglow lines are straightened along the Y-axis in the corrected image. You may also note that the dispersion direction is reversed. Also, because the wavelength information are added to the fits header of the corrected image, wavelength at the cursor position is shown at the information panel of ds9.

```
--> mdpdisplay gcflabcrMCSA00057147_MODS11-0390.fits frame=1
--> mdpdisplay trgcflabcrMCSA00057147_MODS11-0390.fits frame=2
```

4.4.2 Removal of residual sky emission

Although the sky was subtracted between the A-B pairs before the target extraction, there are residual night airglow lines. This is because the intensity of the airglow lines was changing during the observation of A-B pairs. Here the residual sky emissions will be removed by fitting a polynomial function along the spatial direction. One of the purposes of the transform task in the previous section was to make this sky fitting easier: because the spatial direction was aligned to the Y-axis by the transform task, sky can be removed by only the fitting along the Y-axis.

A task named background will be used.

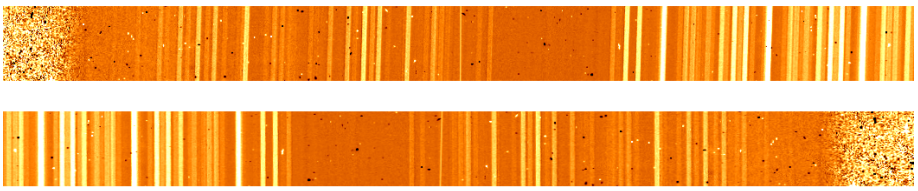

Figure 4.4: Two-dimensional spectrum before and after the distortion correction. (Upper: before the correction. Lower: after the correction.)

```
--> epar background
```

Let's set the parameters of background, then run the task (Here I choose to add "bg" for the prefix of the resulting files):

```
        input =                 Input images to be background
                                subtracted
       output =                 Output background subtracted images
        (axis = 2)              Axis along which background is fit
                                and subtracted
 (interactive = yes)            Set fitting parameters interactively?
      (sample = "*")            Sample of points to use in fit
    (naverage = 1)              Number of points in sample averaging
    (function = "chebyshev")    Fitting function
       (order = 3)              Order of fitting function
  (low_reject = 3.0)            Low rejection in sigma of fit
 (high_reject = 3.0)            High rejection in sigma of fit
    (niterate = 3)              Number of rejection iterations
        (grow = 0.0)            Rejection growing radius
    (graphics = "stdgraph")     Graphics output device
      (cursor = "")             Graphics cursor input
```

```
--> !sed 's/\(.*\)/bg\1/' trMODS11-0390.lst > bgMODS11-0390.lst
--> background @trMODS11-0390.lst @bgMODS11-0390.lst
```

then, a graphic window appears with a message to enter the column of the fitting. Then please input a x-coordinate to determine the fitting parameters; input a column at a bright residual sky line near the center of the image, by checking the spectral image on ds9.

Then signal counts at the specified column are plotted together with the fitting function given in the task parameters. By checking the plot, you can change the fitting parameters. Commands used in the window of background are almost same as those used in the fitting mode of identify.

Second-order polynomial (order=3) would be OK for the fitting, because residual sky emission is typically just tilted with a moderate curve in its profile. If there is a bright continuum line such as from a standard star or there are extraneous data at the edge of the

image, such regions should be omitted from the sample region of the sky fitting by using the "s" keys.

Including the sample region parameters, the fitting parameters specified at this point will be applied to all the columns. When you exit with "q" after setting the parameters at a column, you will be asked again to input a column for the fitting. Please check if the parameters are OK at other columns.

After you finish the background task, please display the resulting images on ds9 and confirm that the residual emission lines are removed.

4.5 Combine of the spectra

Here the spectra will be combined. The four frames currently processed were originally produced by the A−B subtractions. In other words, each of the four frames should show not only a spectrum from A frame but also a negative spectrum from B frame. By reversing the negative B spectrum to positive and shifting it to the A position, eight spectra will be combined in total.

First let's shift the B spectrum by the nodding length to match the position of the A spectrum. Nodding length can be checked by displaying the header keyword (K_DITWID) as follows:

```
--> hselect @bgMODS11-0390.lst "$I,K_DITWID" yes
bgtrgcflabcrMCSA00057147_MODS11-0390.fits        3.000
bgtrgcflabcrMCSA00057151_MODS11-0390.fits        3.000
bgtrgcflabcrMCSA00057159_MODS11-0390.fits        3.000
bgtrgcflabcrMCSA00057163_MODS11-0390.fits        3.000
```

The result shows that the nodding length is 3 arcsec for all frames. Then the shift should be $3.0/0.117 \sim 26$pixel, because the pixel scale of MOIRCS is 0.117 arcsec/pixel. Integer number is used for the shift, because a shift with a sub-pixel value involves unnecessary interpolation. A task imshift will be used:

```
--> !sed 's/\(.*\)/sh\1/' bgMODS11-0390.lst > shMODS11-0390.lst
--> imshift @bgMODS11-0390.lst @shMODS11-0390.lst 0 26
```

Next, let's reverse the sign of the shifted spectra using imarith:

```
--> !sed 's/\(.*\)/ng\1/' shMODS11-0390.lst > ngMODS11-0390.lst
--> imarith @shMODS11-0390.lst * -1 @ngMODS11-0390.lst
```

Finally, combine the images using imcombine. In this exercise, median combine is adopted, because the number of frames are not large, and there remains many cosmic rays.

```
--> epar imcombine

        input = ""              List of images to combine
       output = ""              List of output images
  (headers = "")                List of header files (optional)
  (bpmasks = "")                List of bad pixel masks (optional)
 (rejmasks = "")                List of rejection masks (optional)
```

```
   (nrejmasks = "")              List of number rejected masks

      (optional)
    (expmasks = "")             List of exposure masks (optional)
      (sigmas = "")             List of sigma images (optional)
       (imcmb = "$I")           Keyword for IMCMB keywords
     (logfile = "STDOUT")       Log file

     (combine = "median")       Type of combine operation
      (reject = "sigclip")      Type of rejection
     (project = no)             Project highest dimension of input
                                images?
     (outtype = "real")         Output image pixel datatype
   (outlimits = "")             Output limits (x1 x2 y1 y2 ...)
     (offsets = "none")         Input image offsets
    (masktype = "none")         Mask type
   (maskvalue = "0")            Mask value
       (blank = 0.0)            Value if there are no pixels

       (scale = "exposure")     Image scaling
        (zero = "none")         Image zero point offset
      (weight = "exposure")     Image weights
     (statsec = "")             Image section for computing statistics
     (expname = "EXPTIME")      Image header exposure time keyword

  (lthreshold = INDEF)          Lower threshold
  (hthreshold = INDEF)          Upper threshold
        (nlow = 1)              minmax: Number of low pixels to reject
       (nhigh = 1)              minmax: Number of high pixels to reject
       (nkeep = 1)              Minimum to keep (pos) or maximum to
                                reject (neg)
       (mclip = yes)            Use median in sigma clipping algorithms?
      (lsigma = 3.0)            Lower sigma clipping factor
      (hsigma = 3.0)            Upper sigma clipping factor
     (rdnoise = "0.")           ccdclip: CCD readout noise (electrons)
        (gain = "1.")           ccdclip: CCD gain (electrons/DN)
      (snoise = "0.")           ccdclip: Sensitivity noise (fraction)
    (sigscale = 0.1)            Tolerance for sigma clipping scaling
                                corrections
       (pclip = -0.5)           pclip: Percentile clipping parameter
        (grow = 0.0)            Radius (pixels) for neighbor rejection
        (mode = "al")
```

—> imcombine @bgMODS11-0390.lst ,@ngMODS11-0390.lst HK500_MODS11-0390

Other objects can be reduced same way from the extraction of a slit image to the sum of the spectra.

4.6 Flux calibration and telluric correction

Value of the spectral data corresponds to a count (ADU) per pixel per exposure. The value should be converted to flux per wavelength (F_λ [erg sec^{-1} cm^{-2} Å^{-1}]) to measure the flux from the target. For the purpose of flux calibration, standard stars with known brightness should be observed close in time to the target at an elevation similar to the target observation.

Due to the atmospheric absorption and the efficiency of the telescope and the instrument, not all photons from space reach the detectors. As these factors are thought to depend on the wavelength, the relation between the count and the flux should be considered as a function of wavelength, in particular for the spectroscopic observations. Let $N_{\rm obs}(\lambda)$, $F_{\lambda,{\rm int}}$, and $R(\lambda)$ represents the observed count, intrinsic flux from the target, and the efficiency due to atmosphere/telescope/instrument, respectively. Then the relation can be represented as:

$$N_{\rm obs}(\lambda) = R(\lambda) \times F_{\lambda,{\rm int}}. \tag{4.1}$$

While there are many libraries of spectroscopic standard stars with intrinsic flux for optical observations, it is not the case for the near-infrared observations. Thus stars with known spectral types are observed as standard stars, and theoretical model consistent with their spectral types is used to derive their flux. A-type stars are often used, because their spectra are relatively featureless except for hydrogen absorption lines. For this exercise, a model spectrum based on Castelli (2004) [1] has been prepared.[6]

First, data of the standard star should be reduced in the same way as the target. Please make a two-dimensional spectrum of the standard star by following the previous sections. For the residual sky subtraction, you should take care not to sample the bright continuum of the star as the background regions. The file name of the resulting two-dimensional spectrum is assumed to be HK500_M53735.fits.

A task named apall is used to extract the stellar continuum from the two-dimensional spectrum. There are two main processes in apall — one is to determine the aperture and background regions for the extraction, the other is to trace and extract the continuum light. Parameters used in each step can be determined interactively. So let's run the task with default parameters. Although you will be asked many questions, it is basically OK to reply with 'yes':

```
—> apall HK500_M53735.fits
Recenter apertures for HK500_M53735? ('yes'):
Resize apertures for HK500_M53735? ('yes'):
Edit apertures for HK500_M53735? ('yes'):
```

A graphic window to determine the aperture and background appears at the beginning (figure 4.5 upper left). Center of the aperture is detected automatically at this point. Placing "u" and "l" key sets the upper and lower limit of the aperture at the cursor position, respectively. A "b" key changes the graphic window to set the background parameters (figure 4.5 upper right). In this window, background regions are defined with "s" keys — select appropriate background regions on both sides of the aperture. Press "q" key to go back to the initial window, then "q" key again to exit from the aperture determination.

[6] If you do not mind the hydrogen absorptions, it is OK to adopt blackbody radiation. In case of synthetic spectra, the absorption lines should be broadened according to the instrument resolution, and I have created such a spectrum for this exercise.

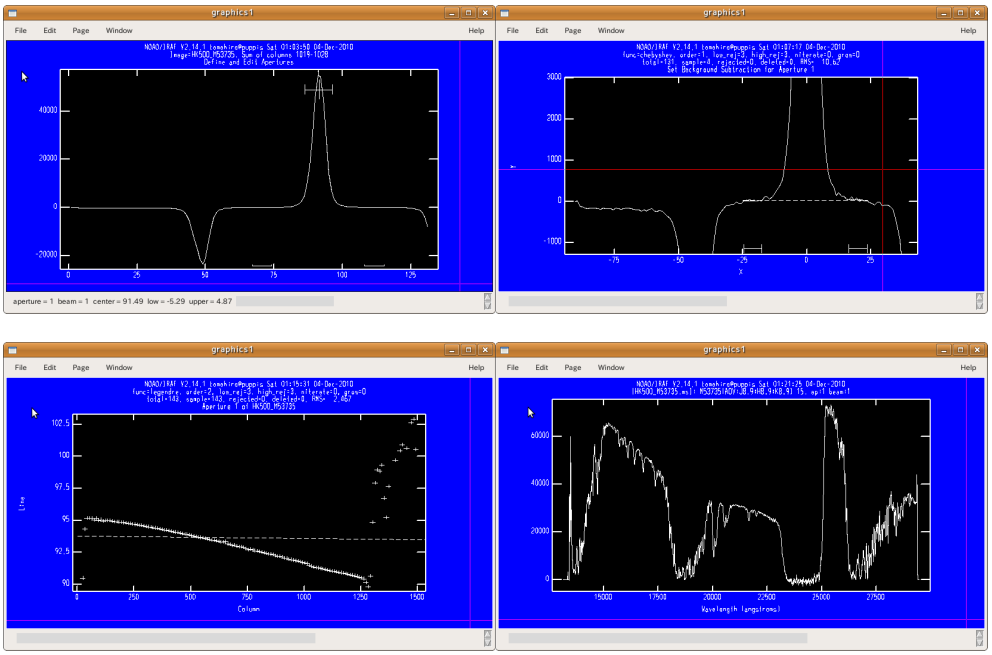

Figure 4.5: Graphic windows of apall. (Upper left: initial window to define the aperture and background. Upper right: window to determine the background. Lower left: window to determine the polynomial function to trace the spectrum. Lower right: finally obtained spectrum.)

```
Trace apertures for HK500_M53735?
Fit traced positions for HK500_M53735 interactively?
Fit curve to aperture 1 of HK500_M53735 interactively
```

A window to trace the spectrum appears next (figure 4.5 lower left). Peak of the detected aperture is traced and fitted with a polynomial function. A spectrum will be extracted according to the fitting. So let' make the fitting residuals small by using "s" keys (set the sample region for the fitting) and ":order" command (set the order of the fitting function).

```
Write apertures for HK500_M53735 to database?
Extract aperture spectra for HK500_M53735?
Review extracted spectra from HK500_M53735?
```

An extracted spectrum appears in the end (figure 4.5 lower right). The resulting file is named as like HK500_M53735.ms.fits, which corresponds to $N(\lambda)$ in the above relation.

This file is divided by the model spectrum to derive $R(\lambda)$. A task rescurve in MCSMDP is used. The task parameters are set as follows:

```
stddata = "" one-D standard star data
 stdmag =    Vega magnitude of the standard at magfilter
```

```
     output = "" output response curve
(magfilter = "mdpdb$filter/2mass_K.fits") transmission curve of filter for stdmag
  (moddata = "mdpdb$standard/A0V_g50.spc") standard star model spectrum
(vegamodel = "mdpdb$filter/alpha_lyr_stis_003.fits") Vega model spectrum
```

In addition to the model spectrum (A0V_g50.spc), Vega magnitude at K, filter transmission curve to determine the magnitude, and the model spectrum of Vega to convert the vega magnitude to the flux are specified here. All data are included in MDPDB.

An A0 star from the HIPPARCOS catalog, M53735 was observed this time. Its K magnitude is known to be 8.856 from the 2MASS catalog. Then the command is executed as follows:

```
--> rescurve HK500_M53735.ms.fits 8.856 resc_CDFN2_HK500.fits
```

The resulting file, resc_CDFN2_HK500.fits, corresponds to $R(\lambda)$.[7]

Finally, two-dimensional object spectrum is divided by $R(\lambda)$ to derive the flux calibrated two-dimensional spectrum. A task mdpfcalib in MCSMDP is used. This task divide the spectrum also by its exposure time to derive energy per second (=flux):

```
--> mdpfcalib HK500_MODS11-0390.fits resc_CDFN2_HK500.fits
HK500_MODS11-0390fl.fits
```

The flux calibrated frame, HK500_MODS11-0390fl.fits, is in units of [erg sec^{-1} cm^{-2} Å^{-1}].

[7] The file can not be displayed on ds9 because it is a binary table fits file.

Chapter 5

Data Analysis

5.1 Measurement of redshift

In this observation, target object was selected based on the colors which were derived from multi-band photometry and which are characteristic of continuum emission from stars in a star-forming galaxy at redshift $z \sim 2$ (Yoshikawa et al. 2010 [5]). The purpose of this observation is to investigate the emission lines from such a high redshift galaxy. The first step is to detect emission lines from the combined two-dimensional spectrum, and to derive the redshift from the lines.

First, let's search for emission lines by displaying the two-dimensional spectrum on ds9. Because the A−B subtraction was made, real emission lines appear also as negative lines below and above the positive line. If you find such an emission line, please make a note regarding its wavelength and the spatial coordinate (y-coordinate).

```
—> mdpdisplay HK500_MODS11-0390fl.fits frame=1
```

Next, wavelength at the center of the emission lines will be measured. A task named splot will be used to plot an one-dimensional spectrum at the spatial position of the emission lines, then to make various analysis on it. Let's run splot as follows:

```
—> splot HK500_MODS11-0390fl.fits （輝線を検出した空間方向の座標）
```

Major commands used in splot are summarized in table 5.1. In the initial window, the spectrum at the specified y-coordinate (1 pixel width) is plotted along the wavelength. So, let's sum up the spectrum with a certain width in its spatial direction by using ":nsum" command. The summed width should be determined by checking the two-dimensional spectrum displayed on ds9; about five to seven pixels would be adequate.

Then, let's expand the wavelength scale at an emission line using window command ("w") or "a" key. When the target line is appropriately displayed, use "k" to measure the line. Pressing "k" key 2 times with the cursor positions on the left and right sides of the emission line results in the Gaussian fitting to the line with the continuum level defined at the two cursor positions. The center, flux, equivalent width, and FWHM of the line fitting are displayed at the bottom of the window.

Because the $HK500$ grism covers the wavelength between 1.3 μm and 2.5 μm, emission lines at rest-frame wavelength $\sim$ 6000Å should be detected if the galaxy is really at redshift $z \sim 2$. In table 5.2, major optical emission lines detected from galaxies and AGNs are summarized.[1] At least two emission lines should be detected from the target. So, please identify the lines and estimate the redshift by comparing the measured wavelengths and the wavelengths listed in table 5.2.

[1] Scientific explanation for each line is beyond the scope of this text.

Table 5.1: Major command keys with splot task

Key	Action
a	Expand the wavelength scale. Put the cursor at the left and right sides of the region to be expanded and press "a" key at each position.
w	Input window command following the space key (see below).
k	Gaussian fitting to a line. Put the cursor at the left and right sides of the line and press "k" key at each position.
s	Smooth the spectrum. (Smoothing box size will be prompted.)
?	Show the list of fitting commands on the terminal.
:nsum n	Sum n pixels along the spatial direction.

Table 5.2: Major emission lines in the visible light

Emission line	Wavelength [Å]	Note
[SII]	6716, 6731	Ratio of 1:1
Hα	6563	Hydrogen Balmer series.
[NII]	6548, 6583	Seen at both sides of Hα, with ratio of 1:2.
[OIII]	4959, 5007	Ratio of 1:3

5.2 Estimation of luminosity from the flux and the redshift

Let's calculate the luminosity (total amount of energy per unit time) of the Hα emission line, using the wavelength and the flux of the line measured by splot task. The relation between the flux (F) and the luminosity (L) is given as:

$$L = F \times 4\pi d_L^2(z), \tag{5.1}$$

where $d_L(z)$ represents luminosity distance, which is derived as a function of redshift by assuming cosmology. For the details of luminosity distance, please refer to textbooks on cosmology. Because there is a task to calculate luminosity distance in MCSMDP, let's use the task as follows (in case of redshift $z \sim 2$):

```
--> cosmology 2.0
Luminosity Distance(m): 4.800e+26
Angular Diameter Distance(m): 5.333e+25
Look-Back Time(yr): 1.024e+10
```

The first line of the results represents the luminosity distance.

Supplement: Emission line luminosity

Although I have explained emission line luminosity in a very simple way, it is only halfway of the analysis. For example, when you estimate stellar birthrate of a galaxy from its Hα luminosity, at least following points should be taken into consideration.

Estimation of flux loss

In the above analysis, the flux was measured with an aperture of a certain spatial extent. You should consider which part of the galaxy corresponds to the aperture. A possibility that the flux was limited by the slit width should also be considered. To accurately derive the absolute flux from data obtained with a slit spectroscopy, it is necessary to calibrate the data such as by comparing its continuum flux with the data obtained with imaging observations.

If the target is a point source and is observed with the same seeing size as for the standard star, the flux loss due to the slit width will be naturally corrected. But this won't happen for galaxies whose profiles are not point-like. Even if the target is a point source, flux loss will happen if a very bright standard star who needs to be defocused is observed.

Interstellar extinction in the Galaxy

It is necessary to know how much interstellar extinction due to interstellar dust in our Galaxy occurs. The distribution and amount of dust in the Galaxy have been known from far-infrared observations. Extinction can be checked by the NED Extinction Calculator. [2] For deep survey fields, they were originally selected due in part to small interstellar extinction in many cases, so the near-infrared extinction (smaller than in the visible wavelengths) might be negligible. Extinction of the target field in this analysis is $E(B-V) = 0.011$.

Interstellar extinction in the target galaxy

It is possible that the observed flux suffers an extinction due to interstellar dust in the target galaxy. In particular, it is important to consider this effect for high redshift galaxies, because they are thought to be more dusty than local galaxies due to their intense star formation.

[2] http://nedwww.ipac.caltech.edu/forms/calculator.html

References

[1] Castelli, F., Kurucz, R. L. 2004, arXiv:astro-ph/0405087

[2] McLean, 1997, Electronic Imaging in Astronomy; Detectors and Instrumentation (John Wiley & Sons Ltd., 1997)

[3] Rousselot et al. 2000, A&A, 354, 1134

[4] Suzuki, R., et al. 2008, PASJ, 60, 1347

[5] Yoshikawa, T., et al. 2010, ApJ, 718, 112

Appendix A

Pipeline reduction with MCSMDP

A.1 Features and overall flow of MCSMDP

In this section, semi-automatic reduction of spectroscopic data using MCSMDP scripts will be explained. The basic flow is same as described in figure 3.1, and mdpproc and mdpcombine task is used for common processing and individual processing, respectively.

In addition to the reduction processes described in the main text, MCSMDP is equipped with the following functions for such purposes of improving the working efficiency, increasing the S/N of the emission line detection, and estimating the error.

Cosmicrays and bad pixel interpolation As explained in the [Supplement] of the main text, A-B pairs are used to detect cosmicrays. Also, pixels in the quadrant boundary of the detectors are interpolated.

Extraction of a slit Spectral image of a target is automatically extracted by referring to the coordinate in the MDP file.

Wavelength calibration: OH line identification in each frame After the OH emission lines are identified in the first one frame of a target, the identification in the other frames will be done automatically by referring to the result of the first frame. Relative shift between the mask and the detector (in the wavelength direction) during the observations is automatically corrected.

Wavelength calibration: Automatic OH line identification Once the OH lines are identified manually for one target, the identification in the other targets will be done automatically by referring to the result of the first one target.

Making sky frames In both common- and individual-processing, frames without background subtraction are created. The sky frame is added as the second extension of a multi-extension FITS (MEF) file.

Estimation of error In flux calibration, error is estimated from the Poisson noise of the sky (estimated from signal counts of the sky frame). The noise frame is added as the third extension of a MEF file.

Line fitting with SPECFIT specfit task in STSDAS is used for fitting with a multi-Gaussian profile, for lines such as [NII]-Hα which are not resolved with the MOIRCS $HK500$ grism. The sky Poisson noise is used to estimate the error of the measured quantities such as flux.

Here I supplementary explain the error estimation. The slit length of the MOIRCS MOS is limited to observe as much objects as possible, so it is not long enough to estimate the error. Thus a frame without the background subtraction is created and the Poisson error of the sky background is adopted for the error estimation.

The error is estimated from the noise frame with the following equation:

$$\sigma_i = \frac{\sqrt{g \times x_i}}{g} \times \frac{1}{r} = \frac{1}{r}\sqrt{\frac{x_i}{g}}, \tag{A.1}$$

where x_i, g, and r represents signal count of the sky frame, gain of the detector ($\mathrm{e^-ADU^{-1}}$), and the response curve ($\mathrm{ADU\,erg^{-1}\,sec\,cm^2\,\AA}$), respectively.

When a two-dimensional spectrum is summed up to an one-dimensional spectrum, the error is estimated with the following equation:

$$\sigma_{\rm obj} = \sqrt{\frac{2\Sigma_{\rm obj}\sigma_i^2}{t_{\rm exp}}}, \tag{A.2}$$

where t_{exp} represents the exposure time, and the factor of $\sqrt{2}$ is due to the A−B subtraction.

A.2 Actual processing steps

A.2.1 Flat frame

A task named mkdflat is used. The task parameters are the following:

infile Input file list

outfile Output file name

bpm Bad pixel mask (already included in MDPDB)

offfile Lamp off frames; can be skipped if nothing.

normilize A value given for the normalization — it can be given as a rough number, because doing imstat in entire frames results in various values depending on the number density of slits.

The task is executed as follows,

```
—> mkdflat @flat1.lst HK500_CDFN2_Domeflat1.fits
 mdpdb$bpm/nlbpm1_FF64r.fits
```

A.2.2 Template file

A template file for the reduction is created using a task mktemplate. Basically the template file is referred to in the PyRAF environment.

A config file for mktemplate is edited first: copy the mktemplate.conf under the install directory of MCSMDP to the current directory and edit it. Then run mktemplate with the config file as an argument, and save the result to the standard output with an appropriate file name:

```
—> copy MCSMDP$doc/mktemplate.conf .
—> !vi mktemplate.conf
—> mktemplate mktemplate.conf > log.py
```

Content of the mktemplate.conf is shown below. It is written in the same syntax as for bash parameters.

```
## parameters
coordlist="mdpdb\$ohlist/list_NS_HK500"
mdpfile="CDFN_MASK02.mdp"
prefix="HK500"
# calibration data for chip1
bpm1="mdpdb\$bpm/nlbpm1_FF64r.fits"
flat1="HK500_CDFN2_Domeflat1.fits"
gdb1="mdpdb\$geomap/mcsdistcrr1_feb07new.dbs"
gmp1="mcsdistcrr1_feb07new.gmp"
# calibration data for chip2
bpm2="mdpdb\$bpm/nlbpm2_FF64r.fits"
flat2="HK500_CDFN2_Domeflat2.fits"
gdb2="mdpdb\$geomap/mcsdistcrr2_feb07new.dbs"
gmp2="  mcsdistcrr2_feb07new.gmp"
```

The parameters are as follows:

coordlist Line list used for wavelength calibration

mdpfile Mask design file

prefix String characters prepended to the final file name of the two-dimensional spectrum; the file name becomes prefix + "_" + MDP file name + ".fits".

bpm[12] Name of the bad pixel mask

flat[12] Name of the flat file

A.2.3 Common processing

Process the data as in the file created by mktemplate.sh.

As in the following example, file lists for each detector and each nodding position are prepared. Combination of A-B pairs should be taken care of — in case only a A (or B) position was taken due to limitation of time, a spare B (or A) file taken close in time should be paired with the A (or B) file in the list (duplication of the B (or A) file is OK).

```
hselect @files.list $I "@'DET-ID' = 1 & K_DITCNT = 1" > files1a.list
hselect @files.list $I "@'DET-ID' = 1 & K_DITCNT = 2" > files1b.list
hselect @files.list $I "@'DET-ID' = 2 & K_DITCNT = 1" > files2a.list
hselect @files.list $I "@'DET-ID' = 2 & K_DITCNT = 2" > files2b.list
```

Run mdpproc:

```
unlearn mdpproc
iraf.mdpproc.bpmask="mdpdb$bpm/nlbpm1_FF64r.fits"
iraf.mdpproc.gdb="mdpdb$geomap/mcsdistcrr1_feb07new.dbs"
iraf.mdpproc.gmp="mcsdistcrr1_feb07new.gmp"
iraf.mdpproc.flat="HK500_CDFN2_Domeflat1.fits"
mdpproc @obj1a.lst @obj1b.lst
```

The resulting files are Multi Extension FITS (MEF) files, where 0th extension is the A−B subtracted data and 1st extension is the data without the background subtraction (for the estimation of noise, and for the wavelength calibration).

The mdprotate task for the correction of grism rotation can also be used for the MEF file. Details about the grism rotation correction are described in the section 4.

A.2.4 Extraction of a slit

A task mdptran is used to match the coordinate between the MDP file and the FITS files. Run the task as follows:

```
--> mdptran CDFN_MASK02.mdp CDFN_MASK02tr1.mdp
 gcflabcrMCSA00057147.fits
```

The first, second, and third argument represents the name of the original MDP file, resulting (converted) MDP file, and the reference FITS file, respectively.

Then, as for the maskplot task, the FITS file is displayed on ds9 and the approximate slit positions are over-plotted. Following message is shown in the terminal:

```
select shape then return or 'q' to quit:
```

then, select an appropriate slit with a mouse click, and press the return key. The next message on the terminal is:

```
click the object position:
```

then click the center of the brightest OH line on the selected slit displayed on ds9 (the line at 16692Å is the brightest in case of the $HK500$ grism). A slit selection ends and the former message appears again to select another slit. Repeat the same procedure to select a slit and the corresponding OH line. At this time, wavelength of the OH line must be same as that selected for the previous slit.

It is not necessary to select all slits; four to five slits or at least three slits are enough. The slits should be evenly selected over the entire image. After appropriate slits are selected, press 'q'+return key at the slit selection message. Then the coordinate of the MDP file is adjusted and the slits are redrawed on ds9. If there is no problem with the result, press 'q'+return key on the following message:

```
type q to write mdp database to file:
```

The result is saved on the MDP file. If a key other than 'q' was pressed, the result is discarded and the process starts from the beginning.

With the resulting MDP file and a task cutspec, a slit can be extracted by only specifying the target name. The target name should be described in the comment field of MDP file (after ",") and should not include characters which can not be used as file name (e.g., space or "."). The task is executed as:

```
--> cutspec @gcobj1a.lst CDFN_MASK02tr1.mdp name=ALE03_205
```

A.2.5 Individual processing

Wavelength calibration is made manually at first for one slit using a task mdpcombine. Select a slit near the center. The target name of the slit has been displayed on ds9 with the mdptran task, and it should be specified as the 4th argument of mdpcombine.

```
--> unlearn mdpcombine
--> iraf.mdpcombine.coordlist="mdpdb$ohlist/list_NS_HK500"
--> iraf.mdpcombine.ymin=42
--> iraf.mdpcombine.ymax=2003
--> mdpcombine @gcflab1.list CDFN_MASK02tr1.mdp HK500_MODS11-0390
 MODS11-0390 id_mode=manual
```

Once the wavelength calibration has been done for one slit with "id_mode=manual", OH lines in other slits are identified automatically by referring to the first result. In the output of mktemplate, there is a sed command like the following, which is to create a command to be executed as mdpcombine from the description in the MDP file :

```
!awk '{print $8}' CDFN_MASK02tr1.mdp | sed 's/\(.*\)/mdpcombine\
 \@gcflab1\.list\ 'CDFN_MASK02'tr1\.mdp\ 'HK500'_\1 \1/'
```

After setting the parameters of mdpcombine, mdpcombine for each object is executed sequentially as follows:

```
--> iraf.mdpcombine.id_mode='auto'
--> iraf.mdpcombine.refname='MODS11-0390'
--> iraf.mdpcombine.bg_inter=no
--> mdpcombine @gcflab1.list CDFN_MASK02tr1.mdp HK500_MODS11-0094
 MODS11-0094
```

A.2.6 Standard star reduction and flux calibration

Common processing for a standard star is done same way as normal MOS data using mdpproc.

The task mdpcombine is also used same way, except that a slit is extracted by specifying the upper and lower coordinates with ymin, ymax parameters (MDP file is not used).

```
--> unlearn mdpproc
--> iraf.mdpproc.bpmask="mdpdb$bpm/nlbpm2_FF64r.fits"
--> iraf.mdpproc.gdb="mdpdb$geomap/mcsdistcrr2_feb07new.dbs"
--> iraf.mdpproc.gmp="mcsdistcrr2_feb07new.gmp"
--> iraf.mdpproc.flat="HK500_CDFN2_Domeflat2.fits"
--> mdpproc MCSA00057114.fits MCSA00057116.fits
--> unlearn mdpcombine
--> iraf.mdpcombine.coordlist="mdpdb$ohlist/list_NS_HK500"
--> mdpcombine gcflabMCSA00057114.fits INDEF HK500_M53735 M53735
 id_mode=manual coordlist="mdpdb$ohlist/list_NS_HK500"
 calframe="INDEF" ymin=900 ymax=1030
```

Because spectra of standard stars are bright enough to be seen in the raw data, following points should be noticed.

- With the identify task, OH lines can not be detected if there is the standard star spectrum on the (detector) line. To avoid this, the detector line should be manually specified with id_section = "line #".
- With the background task, the sample regions for the background fitting should be specified manually with "s" keys.

The processes for making one-dimensional spectrum with apall, response-curve with rescurve, and flux-calibrated two-dimensional spectrum with mdpfcalib are same as described previously. However, because the apall task can not treat MEF files, the first extension of MEF file should be copied as the input of apall. If a MEF file is read in the mdpfcalib task, the error is also calculated as explained previously.

```
--> imcopy HK500_M53735.fits[0] HK500_M53735_0.fits
--> apall HK500_M53735_0.fits
--> rescurve HK500_M53735_0.ms.fits 8.856 resc_CDFN2_HK500.fits
--> mdpfcalib HK500_MODS11-0390.fits resc_CDFN2_HK500.fits
 HK500_MODS11-0390fl.fits
```

A.2.7 Fitting to emission lines in the one-dimensional spectrum

To use the graphic window of matplotlib, mcsmdp is launched with ipython mode as below:

```
$ mcsmdp --ipython
```

Two tasks named mdp2dplot and mdpextract will be used. The mdp2dplot task displays a one-dimensional spectrum by summing up nsum pixels of the input two-dimensional spectrum along the spatial direction, where the wavelength range is specified by its center and the width parameters.

The mdpextract task makes fitting to multiple emission lines, by automatically creating a database file which is necessary for line fitting using SPECFIT. Type of emission line is specified with the fittype parameter, where "Halpha" ([NII]-Hα), "OIII" ([OIII]-Hβ), and "emission" (single line) are currently supported. SPECFIT determines the fitting parameters with an interactive graphic window; for more information, see the help of SPECFIT task. To obtain a successful result, initial parameters needs to be set properly, which should be searched with several trials.

Finally, by specifying the resulting file of mdpextract as the input of mdp2plot, the spectrum is plotted along with the best fitting function.

Sample commands are as follows:

```
In [1]: mcsmdp
In [2]: unlearn mdp2dplot
In [3]: iraf.mdp2dplot.xlabel=r'$\lambda_{\rm obs} \left[\AA\right]$'
In [4]: iraf.mdp2dplot.ylabel=
 r'$F_\lambda\,[\times10^{-18}\,erg\,cm^{-2}s^{-1}\AA^{-1}]$'
In [5]: iraf.mdp2dplot.bottomplot=False
In [6]: iraf.mdp2dplot.yscale=1e18
In [7]: iraf.mdp2dplot.linelist="mdpdb$linelist/nebulae_tex.gaia"
```

```
In [8]: iraf.mdp2dplot.linewidth=1
In [9]: iraf.mdp2dplot.linealpha=1
In [10]: iraf.mdp2dplot.lstyle='k--'
In [11]: iraf.mdp2dplot.ymin=0
In [12]: mdp2dplot HK500_MODS11-0390fl 59 21990 width=800 nsum=5
In [13]: mdpextract HK500_MODS11-0390fl 59 21990
 HK500_MODS11-0390fl_Halpha width=800 nsum=5 sub_frac=0.5 sub_fwhm=42
In [14]: mdp2dplot HK500_MODS11-0390fl_Halpha 59 21990 plotfit+
 ymin=0 width=800
In [15]: pylab.savefig('HK500_MODS11-0390_fl_Halpha.png')
```

The final result is shown in figure A.1.

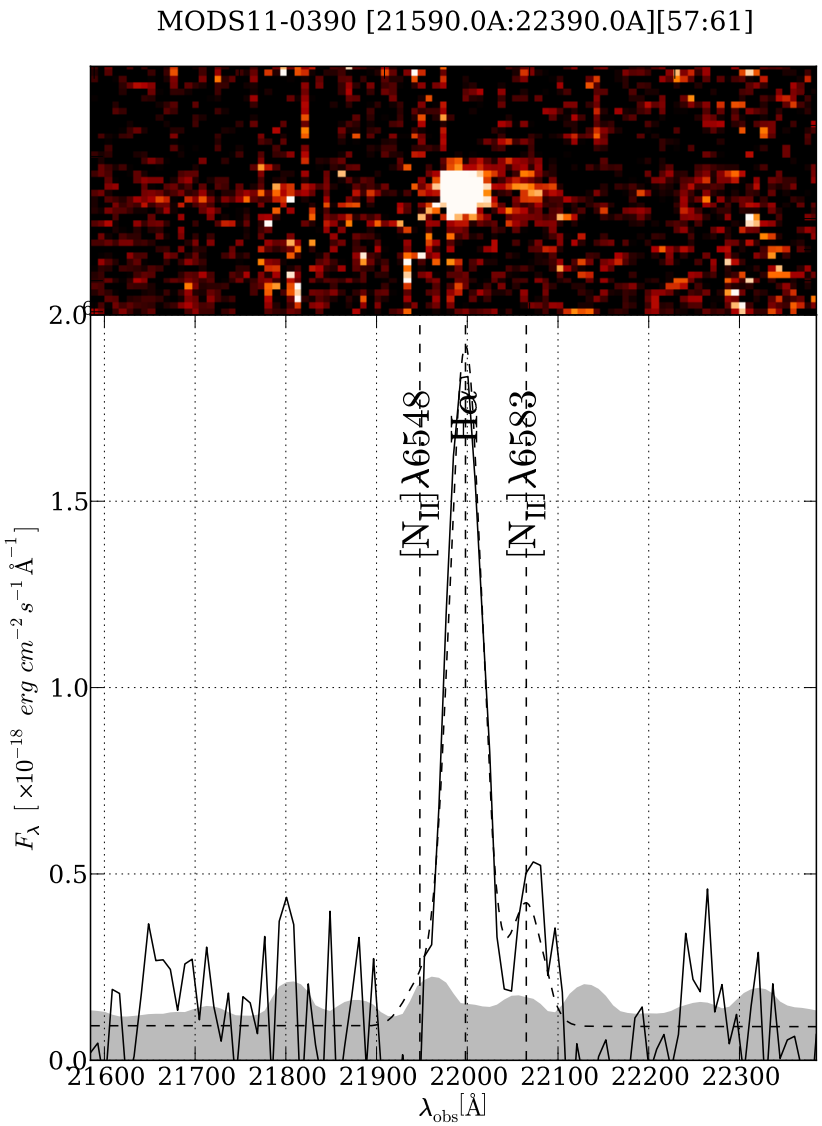

Figure A.1: One-dimensional spectrum with the fitting result plotted by mdp2dplot.