A Real-Time Data Reduction Pipeline for the Goodman Spectrograph

Simón Torres-Robledo,[1] and César Briceño[1,2]

[1]*SOAR Telescope, La Serena 1700000, Chile;* storres@ctio.noao.edu

[2]*Cerro Tololo Interamerican Observatory, Casilla 603, La Serena 1700000, Chile*

Abstract. The Goodman Spectroscopic Pipeline is reaching some maturity and behaving in a stable manner. Though its improvement continues, we have started a parallel effort to develop a real-time version, with the goal of obtaining fully reduced spectra seconds after the data has been obtained at the telescope. Most of the required structure, algorithms and processes already exist with the offline version. The real-time version differs in its requirements for flow control, calibration files, image combination, reprocessing, observing logging assistance, etc. Here we present an outline of the route for implementation of a real time online version of the Goodman spectroscopic pipeline.

1. Introduction

The 4-m Southern Astrophysical Research telescope *SOAR* telescope, located on Cerro Pachón, northern Chile, currently has as it most used instrument the *Goodman High Throughput Spectrograph* (GHTS), an imaging spectrograph developed by the University of North Carolina Clemens et al. (2004). SOAR operations are run in classical mode, with approved proposals scheduled on specific dates. Though observers can go up to the summit, most often they observe from a remote location via an Internet connection, through a VPN tunnel, accessing the instrument software with a VNC client. With the advent of the new generation of survey telescopes such as LSST, SOAR is aiming at becoming a prime follow up facility for transients and Time Domain events. To this effect, a project has been set up in collaboration with the National Optical Astronomical Observatory (NOAO) and Las Cumbres Observatory (LCO), to automate various processes in order to make observations more efficient, allow the telescope to respond faster to incoming alerts, and interface smoothly with the existing robotic scheduling technology developed by LCO.

The Goodman Spectroscopic Pipeline (GSP) is part of this effor, with an online version envisioned, working automatically and capable of delivering science-ready spectra seconds after the shutter has closed. This is particularly important when scientists need to make real time decisions like whether to obtain additional spectra of a given object that may be on the rise, or fading. However, at the moment of writing GSP exists only as an offline version, a one-line command that can be run once the observing night has finished. The user connects via VPN and VNC to a dedicated data reduction machine on which we have the latest version of the GSP. In this contribution we describe the requirements for an online implementation of the GSP using web technology with secure authentication and responsive layout, using modern, well proven technology.

2. Goodman Spectroscopic Pipeline: offline version status and performance

The GSP is based on a two-step process. `redccd` does the basic CCD data reduction, common for imaging and spectroscopy, with some small differences for spectroscopy. `redspec` does spectroscopic-specific data reduction Massey & Hanson (2013), including automatic wavelength calibration. Processing a typical full night takes between three and five minutes.

From the start of this project, we aimed for an automated, unsupervised, wavelength calibration of the spectra. After experimenting obtaining wavelength calibrations with several methods, including trying out an interactive module, we found that the best solution was to use a catalog of templates, a collection of comparison lamp spectra taken with our own hollow cathode lamps. This approach provides good, reliable solutions without the need for any manual interaction from the user. The GHTS is a highly configurable instrument, which means that the camera and grating angle can be adjusted to almost any combination of values, yielding a wide range of possible wavelength coverage options. This flexibility leads to practical problems when trying to setup a library of comparison lamps for various modes, because of the large number of possible wavelength ranges. Therefore, we decided to setup the standard spectral lamp library limited to the most often used modes, for the more frequently requested gratings. It is still possible to use the instrument in *Custom* mode (in which the user defines the central wavelength for the Littrow mode) but such custom modes do not have a corresponding template in the library, for obvious reasons. There are seven gratings available at present for the Goodman spectrograph: 400, 600, 930, 1200, 1800, 2100, and 2400 l/mm; for the last three there is no fixed mode defined, but rather they are normally used in Littrow mode. We built standard lamp spectra for the two most used modes of the 400 line grating, and for several modes of the 600, 930 and 1200 line gratings.

Table 1. Sample of spectroscopic modes definition for the 930 l/mm grating

Grating (lines/mm)	Dispersion (Å/pixel)	Coverage (Å)	Max R @ 5500nm (3 pix with 0.46" slit)	Blocking Filter
930	0.42	M1: 300-470	4450	–
		M2: 385-555		–
		M3: 470-640		GG-385
		M4: 555-725		GG-495
		M5: 640-810		GG-495
		M6: 725-895		OG-570

The lamps in the library are not linearized because the raw spectra coming out of the GHTS are non-linear in wavelength space, therefore reference wavelength solutions need to be non-linear. All the emission lines detected in the lamp spectrum are recorded in the header, together with their corresponding wavelength value as obtained from the fit of the mathematical model used to describe the solution.

Performance-wise the results have been quite satisfactory. Overall, we obtain root-mean square (RMS) values similar to those obtained using IRAF. For instance using the 930 l/mm grating in the M2 mode the RMS error of the wavelength solution was 0.281Å.

3. Goodman Spectroscopic Pipeline Real-Time version: Design Constraints

For the live data reduction pipeline we want to do away with the VNC system and move to a web-based service, that would allow secure authentication. Also, the offline version of the GSP exists as a single Python package; for the live version we will need several other components.

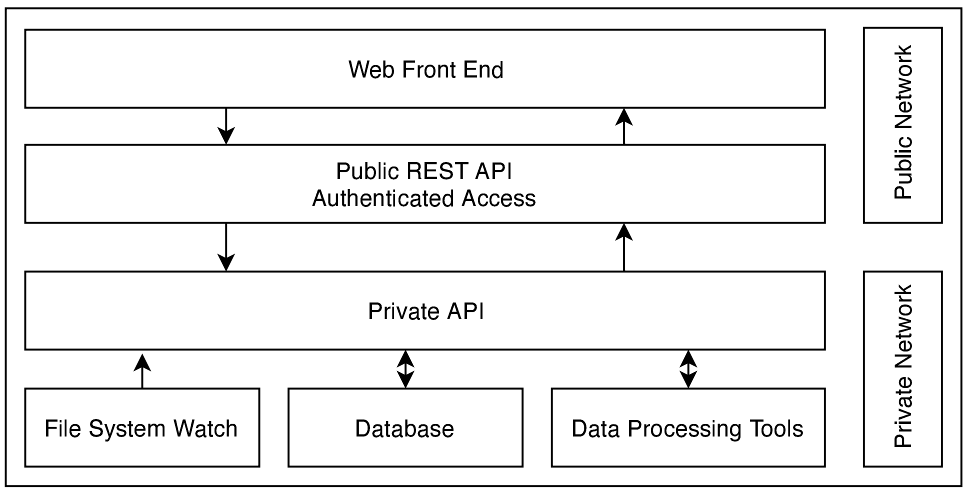

Figure 1. Simplified schematic representing the hierachical structure and relation between components for the proposed design. Several components were intentionally omited due to spacing.

3.1. Data Reduction Package

This is based on the offline GSP Python package. The modifications required for the real-time version are intended to enable asynchronous data processing as well as control, this could be achieved by adding a private REST API that would enable modifying the pipeline's settings needed to operate under different observing strategies. It has to be relatively simple and able to work independently and automatically, allowing intervention by request from the Public API. The allowed control hooks in the private API should be minimal so that authentication is not required, though the access should be constrained to the Observatory's private network. Some of the controls that need to be included are: turn on or off file system event watching routines, alter settings, report errors. It should also handle a light database for redundancy in case the web server fails. In terms of data processing most of the routines are already implemented in the offline version, but there are a couple of things still missing, such as optimal extraction Marsh (1989) and Horne (1986) flux calibration, deblending of multiple sources, low signal-to-noise extraction.

As is explained in Torres-Robledo et al. (2017) our philosophy is to rely as much as possible on Astropy's code Astropy Collaboration & Astropy Contributors (2013) and Astropy Collaboration & Astropy Contributors (2018) therefore, some of the code we had to develop ourselves because there was nothing equivalent implemented yet in Astropy. We plan to add these pieces of code as our contribution to the appropriate Astropy Package.

3.2. Public REST API

Publishing a web application is not a simple task, with security been the biggest point of concern, but also that the application be stable and reliable. Fortunately, there are plenty of tools that allow us to simplify these tasks, in fact, most of them, which is important because we don't have the resources to hire an entire team of developers experts on these very specific tools. The scope of the services provided by the public API should be end-user oriented only, such as, secure authentication and channeling communications with the private API.

3.3. Web Front End

A highly responsive website is favored over a local GUI for one simple reason: most of the GHTS users are working remotely, while local users will still benefit from it. Though more difficult to implement, there are several benefits that come with a web font end, for instance: adaptive layout, user experience less dependant on connection quality, less bandwidth usage and of course taking advantage of the interactive experience that web technology allows.

We have not yet decided what are the tools (stack) that we will use, but the criteria for selecting them are: being well documented and easily testable. By *easily* we mean that there have to be good tools available and a good testing philosophy behind its development. Testability is highlighted here but it applies to all the code of our project.

Acknowledgments. The authors would like to acknowledge the important contribution from Bruno Quint, David Sanmartim and Tina Armond. This research made use of Astropy, a community-developed core Python package for Astronomy Astropy Collaboration & Astropy Contributors (2013) and Astropy Collaboration & Astropy Contributors (2018) This work has been developed at the Southern Astrophysical Research (SOAR) telescope, which is a joint project of the Ministério da Ciência, Tecnologia, Inovação e Comunicações (MCTIC) do Brasil, the U.S. National Optical Astronomy Observatory (NOAO), the University of North Carolina at Chapel Hill (UNC), and Michigan State University (MSU).

References

Astropy Collaboration, & Astropy Contributors 2013, A&A, 558, A33. 1307.6212
— 2018, AJ, 156, 123. 1801.02634
Clemens, J. C., Crain, J. A., & Anderson, R. 2004, in Ground-based Instrumentation for Astronomy, edited by A. F. M. Moorwood, & M. Iye, vol. 5492 of SPIE, 331
Horne, K. 1986, PASP, 98, 609
Marsh, T. R. 1989, PASP, 101, 1032
Massey, P., & Hanson, M. M. 2013, Astronomical Spectroscopy, 35
Torres-Robledo, S., Briceno, C., Quint, B., & Sanmartim, D. 2017, in ADASS XXVIII, edited by TBD (San Francisco: ASP), vol. TBD of ASP Conf. Ser., TBD