MEMO 2 - Summary of Astrometry.net Investigation

25-Jan-2021

Nic Erasmus

Background:

For the IO to successfully automate observations on small to moderate FoV telescopes (e.g. SHOC on the 74-in @ 1.3×1.3 arcmin FoV, SHOC on the 40-inch @ 2.9×2.9 arcmin FoV, and SHOC on Lesedi @ 5.7×5.7 arcmin FoV), an unsupervised method needs to be implemented to ensure the telescope is pointing at the requested target.

Currently this is done manually by the observer comparing the image of the observed field to that of a catalogue image and making pointing adjustments in the TCS if there is any offset in the telescope pointing. In principle, this can be automated through services like that offered by Astrometry.net[1] that allows a user to upload an image (or a table of sources with their pixel position and flux) to a website and it returns the WCS[2]. Astrometry.net is also available as installable software where the astrometric catalogues can also be downloaded and stored locally, i.e., no need to upload images to a website which can significantly slow-down the process and is reliant on an internet connection.

A. Installation of Astrometry.net

The approach is to install asrometry.net software on a dedicated local server or VM that can be accessed by any user (or LCU):

Instructions followed to do that here:

http://astrometry.net/doc/readme.html

NB - From installation instruction link on readme:
"As of April 2019, the script doc/install_astrometry_on_linux.sh will install all dependencies along with astrometry.net on Linux, and download 4200/ index files."

Did not use the bash script directly but used it as a guide because it shows all the commands to do a complete instruction on a new (>2019) Ubuntu machine with Python3 which the website instructions does not show.

So, installation instructions on a clean install of Ubuntu 20.04 LTS:

---

[1] http://astrometry.net/
[2] WCS is an acronym for "world coordinate system" and is a standards-based description of the transformation between image coordinates in pixels and sky coordinates in (usually) RA and DEC.

Download astrometry.net software:

```
wget
http://astrometry.net/downloads/astrometry.net-latest.tar.gz
```

Untar the file:

```
tar xvzf astrometry.net-latest.tar.gz
```

Install prerequisites: ~~(NB. I added "swig" which was not in the bash file version but a prerequisite)~~

~~sudo apt install -y make build-essential python3 python3-pip netpbm libnetpbm10-dev zlib1g-dev libcairo2-dev libjpeg-dev libcfitsio-dev libbz2-dev wget wcslib-dev swig~~

```
sudo apt install build-essential curl git file pkg-config swig
libcairo2-dev libnetpbm10-dev netpbm libpng-dev libjpeg-dev
zlib1g-dev libbz2-dev libcfitsio-dev wcslib-dev python3
python3-pip python3-distutils python3-dev python3-numpy
python3-scipy python3-pil
```

~~Install required Python modules:~~

~~pip3 install numpy scipy fitsio~~

Do the install: (NB. sudo in last command because you need permissions to create directories)

```
cd astrometry.net-*
make
make py
make extra
sudo make install
```

Download index files: ~~(NB. I only download the "5000-series", i.e. GAIA based builds, and also only the "00" fields for our small FoV (see Section B below)~~

~~sudo wget -r -nd -np -P /usr/local/astrometry/data/ "data.astrometry.net/5000/" -A "index-5000-*.fits"~~

```
sudo wget -r -nd -np -P /usr/local/astrometry/data/
"https://portal.nersc.gov/project/cosmo/temp/dstn/index-5200/LI
TE/" -A "index-5200-*.fits"
```

Details where installed for IO project:

~~Server: astrometrynet.suth.saao.ac.za~~
~~Username: astro~~
~~Pass: Saaoastro~~

Server: `IOServer1.suth.saao.ac.za`
Username: `io`
Pass: usual SAAO convention of password

B. Downloading of Index Files

The downloading of the index files is included in the installation procedure followed (see Section A). More can be downloaded and added later if needed.

The index files are what astrometry.net uses to calculate the WCS. In the readme they have some description text and a table to describe which files you should grab for your FoV size. From the table it seems for the 74- and 40-inch, and probably Lesedi too, we only require the index-4200-*.fits with the * the “healpix” number that indicates the patch of sky covered. The “healpix” numbers and their sky coverage is shown in Figure 1. The “4200-series” index files are built from the 2MASS catalog is the one suggested on the readme and available here:

http://data.astrometry.net/4200/

But a link on the forum to index files built from GAIA which is the “5000-series” available here:

http://data.astrometry.net/5000/

For the IO project these rather than the default “4200-series” were downloaded because we think we have a better chance with GAIA to calibrate our small FoV.

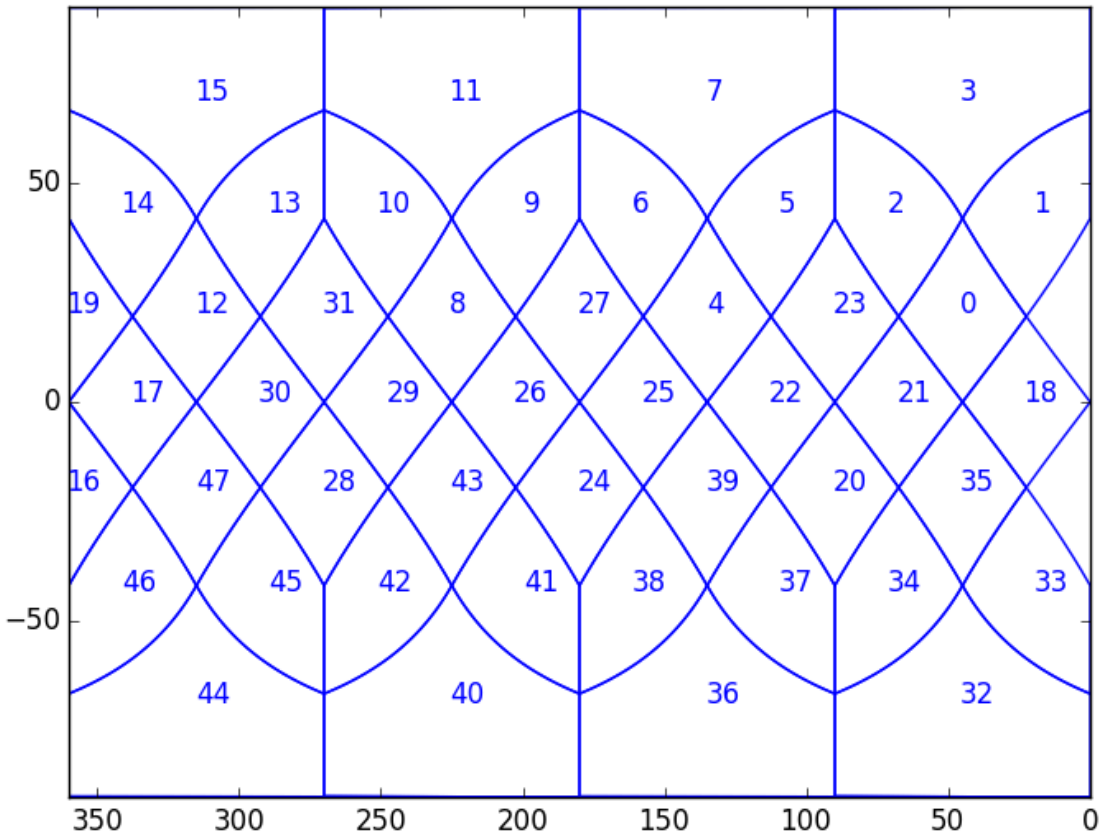

Figure 1: "Healpix" numbers for determining which index files you require depending on where your observations are located on sky. For the SAAO telescopes we probably don't need 15, 11, 7, 3, 14, 13, 10, 9, 6, 5, 2, and 1 because they cover the northern sky. The index files for the "4200-series" are ~100-400MB each so that is ~4GB that we don't need to download. For then"5000-series" the files are double to triple the size so that can be a significant amount of data we don't need to download but all were downloaded anyway for the installation (because it was just easier ;-)).

C. Preparing images and/or source tables

Currently the approach is to copy the fits image across to the local server and to feed the image directly to the astrometry.net software. This seems fine since the fits images from SHOC are max ~2MB so a scp takes a second or two. The SHOC fits images as is seems fine (i.e. no preparation to the images or headers) because all that is grabbed is RA and Dec from the header to help astrometry.net solve for WCS. The other parameters that help astrometry.net as input are things like FoV and search radius which can be hard coded.

In future we might want to send across a source table rather (e.g. for Sibonise) because then the image size will become cumbersome to scp across every time.

D. Getting WCS of an observation

This is done with a Python wrapper that builds the terminal command from user inputs (eventually the LCU) and executes the astrometry.net command via ssh on the locale server and returns the output wcs of the image to a file called "wcs_info.txt" (that can be scraped by the LCU at a later stage).

Requirements: **Python3**, **astropy** (needed to get header info from fits), **sshpass** (needed to copy files and send commands to local astrometry.ner server)

To run:

```
python3 get_wcs_SHOC.py --image /folder/relative/to/where/get_wcs_SHOC.py/is/located/SHA_20200214.0290.fits
```

Or if the fits image is in same folder as get_wcs_SHOC.py then just:

```
python3 get_wcs_SHOC.py --image SHA_20200214.0290.fits
```

Then also indicate the telescope with the `--telescope` option (`40_inch`, `74_inch` or `Lesedi`):

```
python3 get_wcs_SHOC.py -image SHA_20200214.0290.fits -telescope 40_inch
```

This picks the best input criteria for astrometry.net for that telescope like the FoV (and maybe others to add later) but for now the default will be to try search between 1arcmin and 5 arcmin for the FoV solution if no telescope is indicated and that seems to work for the test image I tried for 40 inch telescope images.

The python code will also grab the RA and DEC from the fits header and feed that to astrometry.net as the initial guess for the pointing.

We are entirely flexible now and there are MANY input options still that we can feed astrometry.net if needed by scraping info from the header or asking for additional input parameters from the user or LCU.

To do1: add a cron job to server to delete the copied files every midday.
To do2: implement option to scp from a remote location the server instead of from local location
To do3: ask Jean to create another user that only has write access to the tmp_images folder. Or to only give current "astro" user those rights and create another super user for us IO people.