

Technical Report:

Finding chart tool for astronomical observations at LBT

Software Version: 1.0.6 (fov.sh)

Author: Andrea Rossi, Felice Cusano

Affiliation: INAF- Istituto Nazionale di Astrofisica - Osservatorio di Astrofisica e Scienza dello Spazio Bologna (OAS), via Gobetti 93/3 40129 Bologna - ITALY

Date: 2 February 2026

1. Introduction

This document describes a Bash-based tool ([fov.sh](#), version 1.0.6) designed to generate finding charts for astronomical observations, with a focus on the Large Binocular Telescope (LBT). The script automates image retrieval, visualization, and export using SAOImage DS9 and multiple sky surveys.

2. Overview

The tool is a command-line Bash script that:

- Accepts sky coordinates (RA, Dec) in sexagesimal format
- Retrieves imaging data from different surveys
- Produces a standardized PNG finding chart
- Overlays target position and field-of-view (FoV) information

The output file is named:

None

`fc_LBT_PA<angle>.png`

where `<angle>` is the parallactic angle (integer degrees).

3. Dependencies

The script relies on:

- **SAOImage DS9** (invoked via command line)

- **curl** (for Pan-STARRS queries)
- **dfits / fitsort** (for FITS header inspection, when using user images)
- Standard Unix tools (awk, bash)

4. Input Parameters

Required

- **-r** : Right Ascension (HH:MM:SS)
- **-d** : Declination (DD:MM:SS)

Optional

- **-f** : Field of view (arcmin, default = 5, max = 30)
- **-a** : Parallax angle (degrees, default = 0, max = 360)
- **-s** : Image scale parameter (default = 98)

Survey Selection

- Default: DSS2-red (ESO DSS)
- **-m** : Use 2MASS (NIR)
- **-p <filter>** : Use Pan-STARRS (g, r, i, z)

Custom Image

- **-i <file>** : Use a user-provided FITS image

5. Input Validation

The script performs validation checks:

- RA/Dec format via regular expressions
- Numeric constraints on FoV, PA, and scale
- File existence for user images
- Availability of Pan-STARRS data

6. Coordinate Handling

Coordinates are converted internally:

- RA → degrees (×15 conversion)

- Dec → decimal degrees

These are used for external queries (e.g., Pan-STARRS API).

7. Survey Handling

7.1 DSS2-red (Default)

Accessed via DS9 using:

```
None  
-dsseso survey DSS2-red
```

7.2 2MASS

Activated with `-m`, retrieved via DS9 native interface.

7.3 Pan-STARRS1

Activated with `-p <filter>`:

- Uses `curl` to query the PS1 filename service
- Extracts image metadata (position, filename)
- Builds a FITS cutout request via `fitscut.cgi`
- Uses DS9 `-url` to load the image

Pixel scale is fixed to 0.25 arcsec/pixel.

7.4 User-Provided Image

Activated with `-i`:

- Reads pixel scale from FITS headers (CD2_2 / CDELT2)
- Computes cropping region
- Determines image size and centering

8. Image and FoV Computation

Key parameters:

- FoV converted from arcmin to degrees

- Pixel size computed from FoV and pixel scale
- Download size increased by a factor (1.5) to allow margins
- Zoom level dynamically computed

Image dimensions are constrained to:

- Minimum: 200 pixels
- Maximum: scaled up to ~600 pixels

9. Region Overlays

Two DS9 regions are created:

- **FoV box:**
 - Centered on target
 - Size = FoV
 - Rotated by parallactic angle
 - Annotated with FoV and PA
- **Target marker:**
 - Red circular marker
 - Size scaled with FoV

10. DS9 Command Construction

The script builds a full DS9 command including:

- Image loading (survey or URL or FITS)
- Zoom and centering
- Rotation (negative PA for rotator convention)
- Scaling and colormap (grey, inverted)
- Region overlays
- Output export (PNG)

Example structure:

```
None
ds9 -height <pix> -width <pix> \
  <image source> \
  -zoom <z> \
  -pan to <RA> <Dec> wcs \
  -rotate <angle> \
```

```
-scale <scale> -cmap grey -invert yes \  
-region command <box> \  
-region command <target> \  
-saveimage png <output> \  
-exit
```

11. Execution Strategy

The DS9 command is:

- Written to a temporary script file
- Executed via `bash`
- Deleted after execution

This avoids quoting and parsing issues in complex command strings.

12. Output

- Format: PNG
- Filename: `fc_LBT_PA<angle>.png`
- Includes:
 - Sky image
 - Target marker
 - FoV box with annotation

13. Limitations

- Limited error handling typical of Bash
- Dependence on external tools (DS9, curl, FITS utilities)
- Pan-STARRS availability not guaranteed for all coordinates
- No support for negative declination sign validation
- Fixed output format (PNG only)

14. Possible Improvements

- Port to Python (Astropy, astroquery, matplotlib)
- Add logging and verbose/debug modes
- Improve input validation (full coordinate parsing)

- Add more surveys (Gaia, SDSS, Legacy Surveys)
- Support multiple output formats (PDF, FITS overlays)
- Remove temporary file usage (direct execution)

15. Conclusion

The `fov.sh` script is a compact and effective tool for generating LBT-compatible finding charts. It integrates DS9 with multiple survey backends and provides flexible input handling, making it suitable for rapid observational preparation workflows.

A. Appendix

A.1

Help: A simple call to [fov.sh](#) prints instruction from the help option

```
-----  
Create finding chart for LBT  
version 1.0.6  
-----
```

No positional arguments specified

Example: `bash fov.sh -r 00:00:01 -d 10:00:00 -f 2 -a 0`

options:

r Enter RA in sexagesimal format

d Enter DEC in sexagesimal format

f (optional) Enter size in arcmin

a (optional) Enter Parallactic angle in degrees

m (optional) use this option (no argument) for NIR (2MASS), otherwise it uses DSS2red

p (optional) use this option for PanSTARRS, e.g. -p z for z-band

i (optional) use this option to use your own fits image

s (optional) use this option to enter a different scale, e.g., 95

h Print this Help.

A.2

Example of finding chart: As example , the following creates a finding chart with a field of view of 2 arcmin (see the box region), position angle PA=0 and uses PanSTARRS (option -p) in the r-band. The target at the given coordinates is indicated with a red circle.

`bash fov.sh -r 00:00:01 -d 10:00:00 -f 2 -a 0 -p r`

