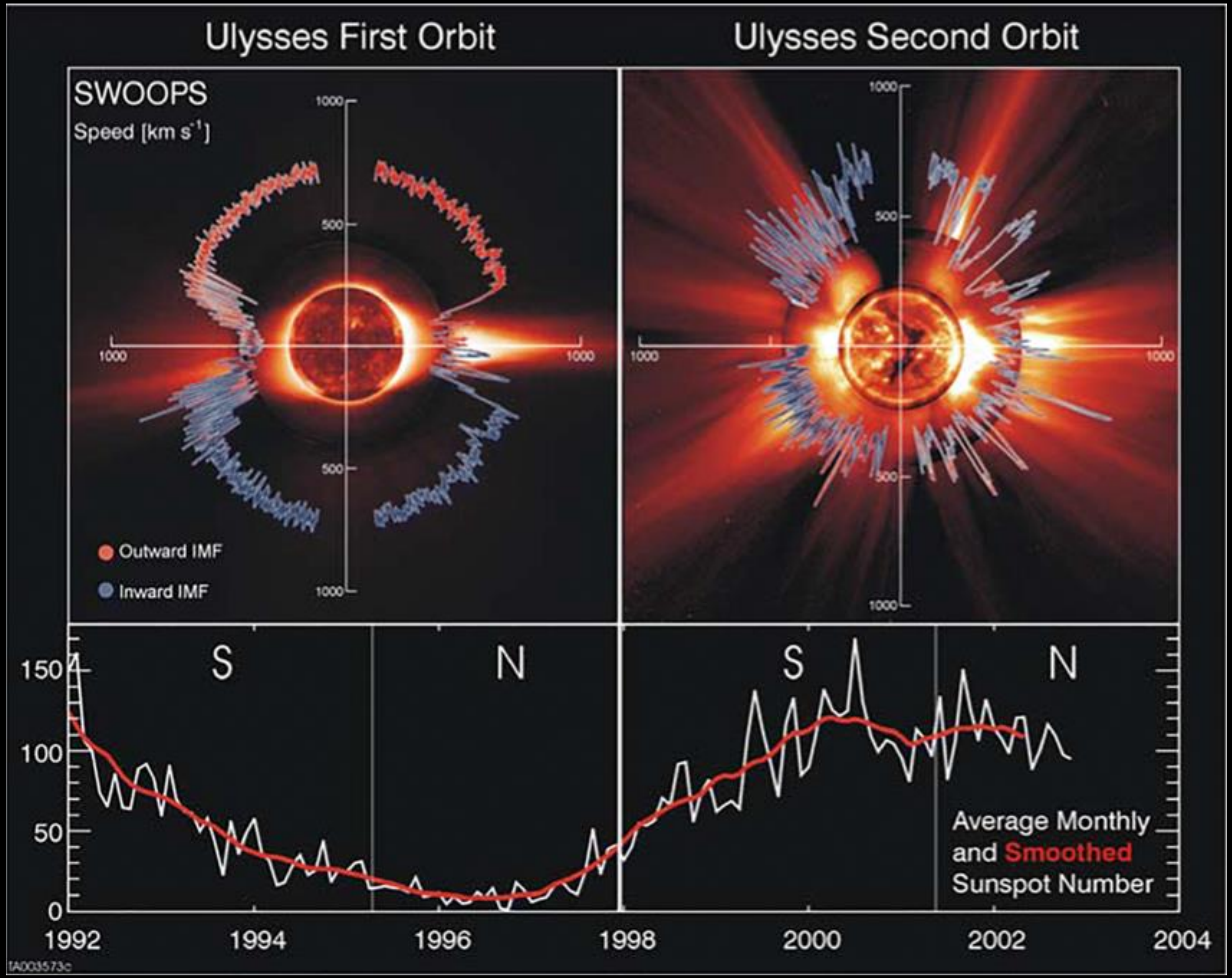
CHRIS LOWDER, CRAIG DEFOREST, MARCUS HUGHES, DEREK LAMB,
RITESH PATEL, JILLIAN REDFERN, DAN SEATON, SAM VAN KOOTEN
SOUTHWEST RESEARCH INSTITUTE

GO WITH THE FLOW:
SOLAR WIND TRACKING WITH PUNCH

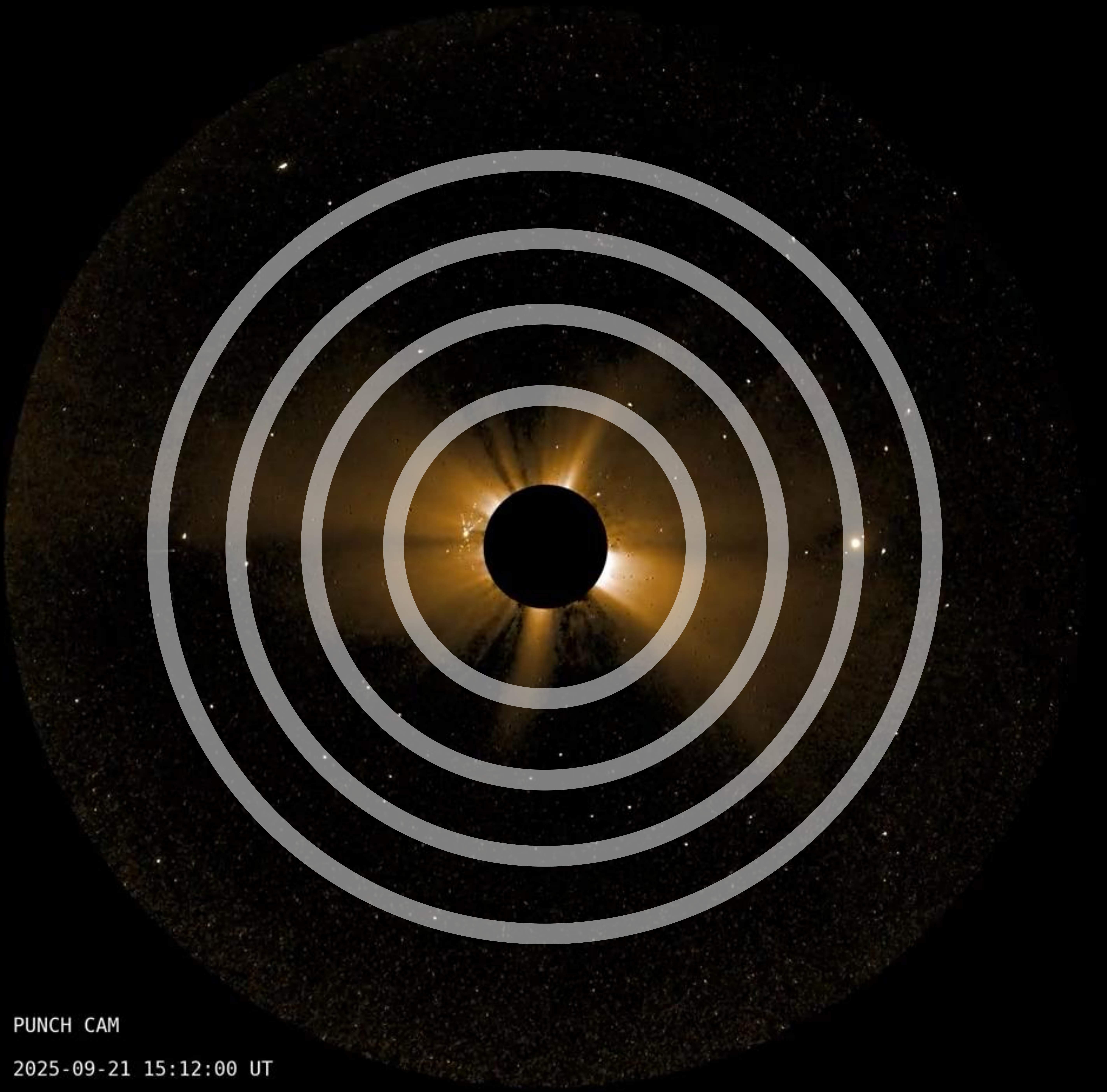
SOLAR WIND MAPS

- Historical wind maps out of the ecliptic from Ulysses
  - Amazing classic observations
  - Limited by cadence



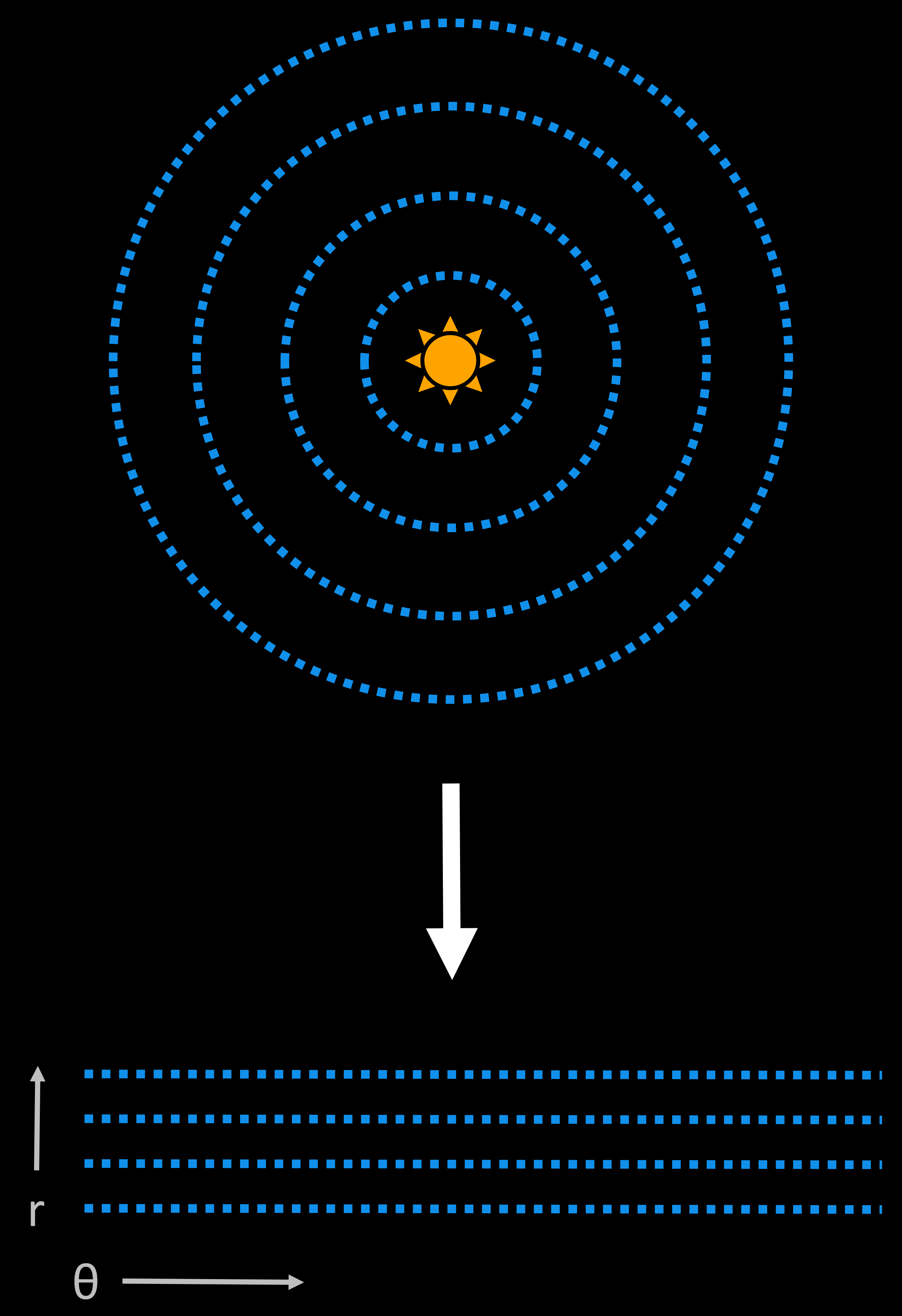
SOLAR WIND MAPS

- PUNCH can use flow tracking to measure solar wind velocities
- Data are reprojected to a polar coordinate frame
- Features tracked using cross-correlation



MAP STRUCTURE

- Generated as plots and saved as FITS files
- VAM product code:
  velocity map in the mosaic frame
- 2D array with corresponding WCS describing coordinates in radius and azimuth



WORKING WITH MAPS

Load a map as an object

```
wmap = load_ndcube_from_fits("file.fits")
```

Plot a full wind map

```
plot_flow_map("wind_map.pdf", wmap)
```

Plot a slice of the wind map

```
wind_slice = wmap.data[0,:]


fig, ax = plt.subplots()
azimuths = np.linspace(0, 360, len(wind_slice)
                  endpoint=False)
ax.plot(azimuths, wind_slice)
```

MAP CUSTOMIZATION

- Located in punchbowl.level3.velocity
- Assemble a list of files paths for input
- Adjust parameters, including the radial band number, spacing and size.
- Returns an NDCube object with measurements. Can be saved to a FITS file with punch_io.write_ndcube_to_fits()

```
def track_velocity(files: list[str],
               delta_t: int = 12,
               sparsity: int = 2,
               n_ofs: int = 151,
               delta_px: int = 2,
               expected_kps_windspeed: int = 300,
               r_band_half_width: float = 0.5,
               max_radius_deg: int = 45,
               num_azimuth_bins: int = 1440*8,
               az_bin: int = 4,
               velocity_azimuth_bins: int = 36,
               ycens: np.ndarray | None = None,
               rbands: list[int] | None = None) -> NDCube:
    """
    Generate velocity map using flow tracking.

    Parameters
    ----------
    files : list[str]
        List of file paths for input data

    delta_t : int, optional
        Time offset in frames between images

    sparsity : int, optional
        Frame skip interval for averaging

    n_ofs : int, optional
        Number of spatial offsets for cross-correlation

    delta_px : int, optional
        Pixel offset increment per sample

    expected_kps_windspeed : int, optional
        Expected wind speed in km/s

    r_band_half_width : float, optional
        Half-width of each radial band in solar radii

    max_radius_deg : int, optional
        The maximum radius in degrees

    num_azimuth_bins : int, optional
        Number of azimuthal bins in the polar remapped images

    az_bin : int, optional
        Binning factor for binning the polar remapped image over the azimuth

    velocity_azimuth_bins : int, optional
        Number of azimuthal bins in the output flow maps

    ycens : numpy.ndarray, optional
        Radial band centers in solar radii

    rbands : list[int], optional
        Indices of radial bands to visualize

    Returns
    -------
    ndcube.NDCube
        The generated velocity map

    """
```

Feel free to ask us any questions about working with these data!

The GitHub discussions page is a useful hub for community questions and answers.

We’re happy to help!