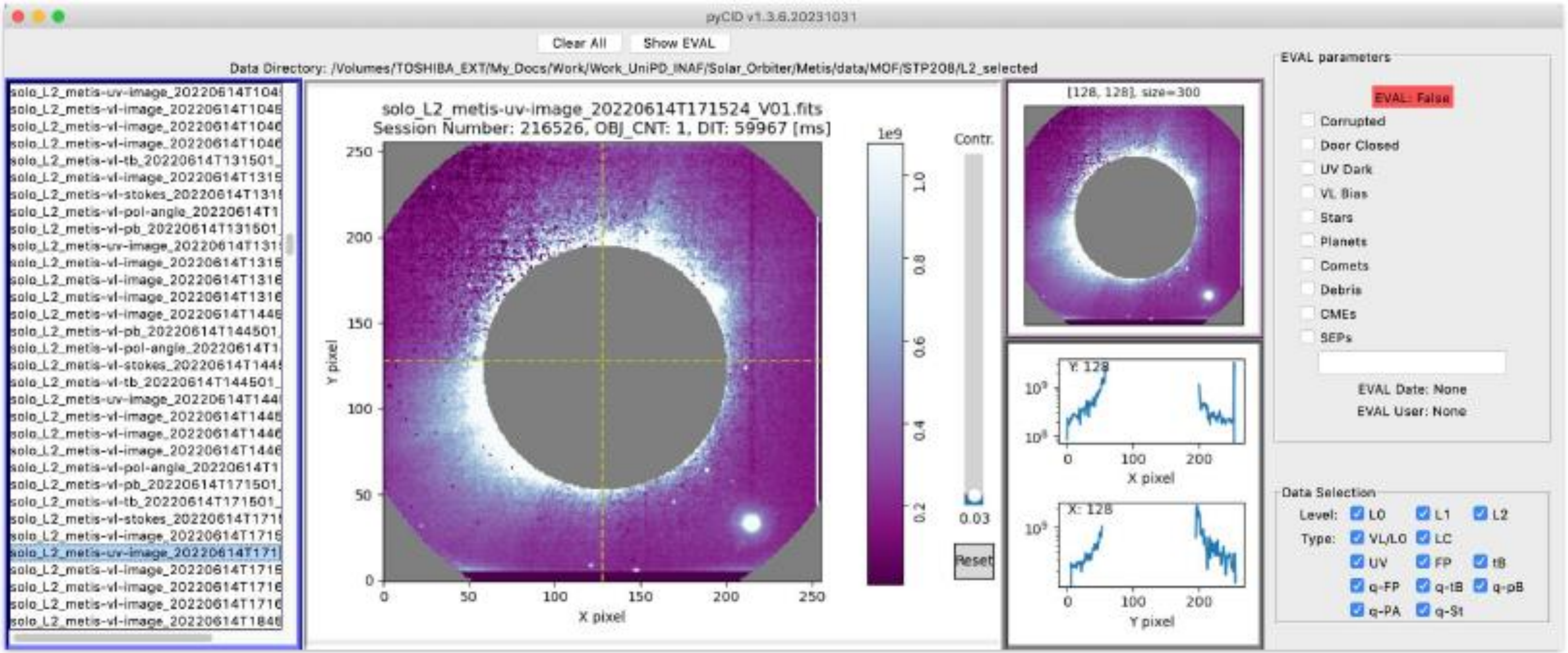
Contribution to the pyCID GUI: a semi-automatic CME detection tool

INAF
ISTITUTO NAZIONALE DI ASTROFISICA
metis
II Metis Science Meeting
January 27-29, 2025
Capodimonte Astronomical Observatory, Naples, Italy

Gianluca Napoletano, Università di Firenze

What is pyCID?

pyCID is a quick-look GUI tool for visualization and validation of Metis data of different types (images, light curves) and levels (L0, L1 and L2). It was inspired by the more sophisticated IDL tool iCID, it is written in Python and was presented during the previous METIS meeting by A. Burtovoi



Data Directory: /Volumes/TOSHIBA_EXT/My_Docs/Work/Work_UniPD_INAF/Solar_Orbiter/Metis/data/MOF/STP208/L2_selected

```
solo_L2_metis-uv-image_20220614T104
solo_L2_metis-vl-image_20220614T1045
solo_L2_metis-vl-image_20220614T1046
solo_L2_metis-vl-image_20220614T1046
solo_L2_metis-vl-tb_20220614T131501_
solo_L2_metis-vl-image_20220614T1315
solo_L2_metis-vl-stokes_20220614T131
solo_L2_metis-vl-pol-angle_20220614T1
solo_L2_metis-vl-pb_20220614T131501_
solo_L2_metis-uv-image_20220614T131
solo_L2_metis-vl-image_20220614T1315
solo_L2_metis-vl-image_20220614T1316
solo_L2_metis-vl-image_20220614T1316
solo_L2_metis-vl-image_20220614T1445
solo_L2_metis-vl-pb_20220614T144501_
solo_L2_metis-vl-pol-angle_20220614T1
solo_L2_metis-vl-stokes_20220614T144
solo_L2_metis-vl-tb_20220614T144501_
solo_L2_metis-uv-image_20220614T144
solo_L2_metis-vl-image_20220614T1445
solo_L2_metis-vl-image_20220614T1446
solo_L2_metis-vl-image_20220614T1446
solo_L2_metis-vl-pol-angle_20220614T1
solo_L2_metis-vl-pb_20220614T171501_
solo_L2_metis-vl-tb_20220614T171501_
solo_L2_metis-vl-stokes_20220614T171
solo_L2_metis-vl-image_20220614T1715
solo_L2_metis-uv-image_20220614T171
solo_L2_metis-vl-image_20220614T1715
solo_L2_metis-vl-image_20220614T1716
solo_L2_metis-vl-image_20220614T1716
solo_L2_metis-vl-image_20220614T1845
```

EVAL parameters

EVAL: False

- ☐ Corrupted
- ☐ Door Closed
- ☐ UV Dark
- ☐ VL Bias
- ☐ Stars
- ☐ Planets
- ☐ Comets
- ☐ Debris
- ☐ CMEs
- ☐ SEPs

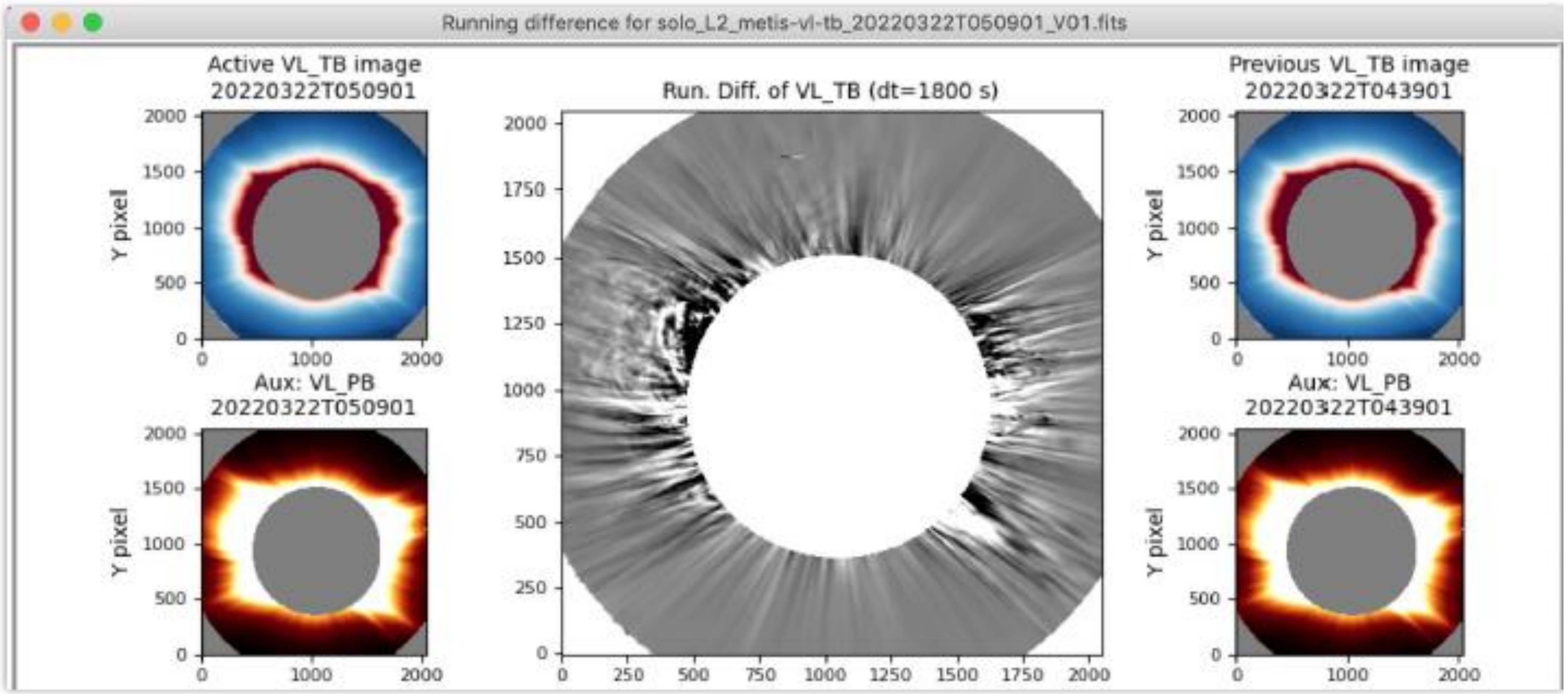
EVAL Date: None
EVAL User: None

Data Selection

Level: ☑ L0 ☑ L1 ☑ L2
Type: ☑ VL/L0 ☑ LC ☑ UV ☑ FP ☑ tB ☑ q-FP ☑ q-tB ☑ q-pB ☑ q-PA ☑ q-St

What is pyCID?

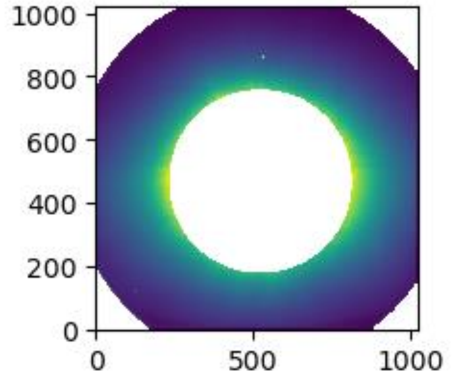
It makes use of the SunPy affiliated package “sunkit-image” to implement additional tools like coordinate transformation and filters for image enhancement and processing


Running difference for solo_L2_metis-vl-tb_20220322T050901_V01.fits
Active VL_TB image 20220322T050901
Run. Diff. of VL_TB (dt=1800 s)
Previous VL_TB image 20220322T043901
Aux: VL_PB 20220322T050901
Aux: VL_PB 20220322T043901
Y pixel

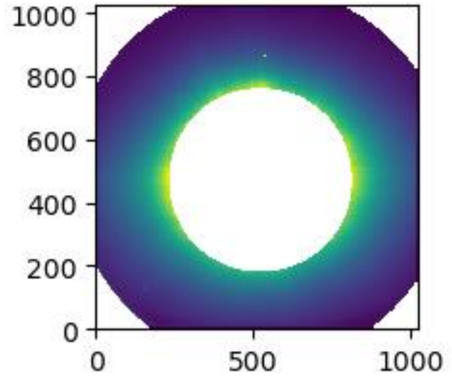
Running differences

- Running differences enhance moving features in the FOV such as Coronal Mass Ejections

L2_metis-vl-tb_20231214T193001

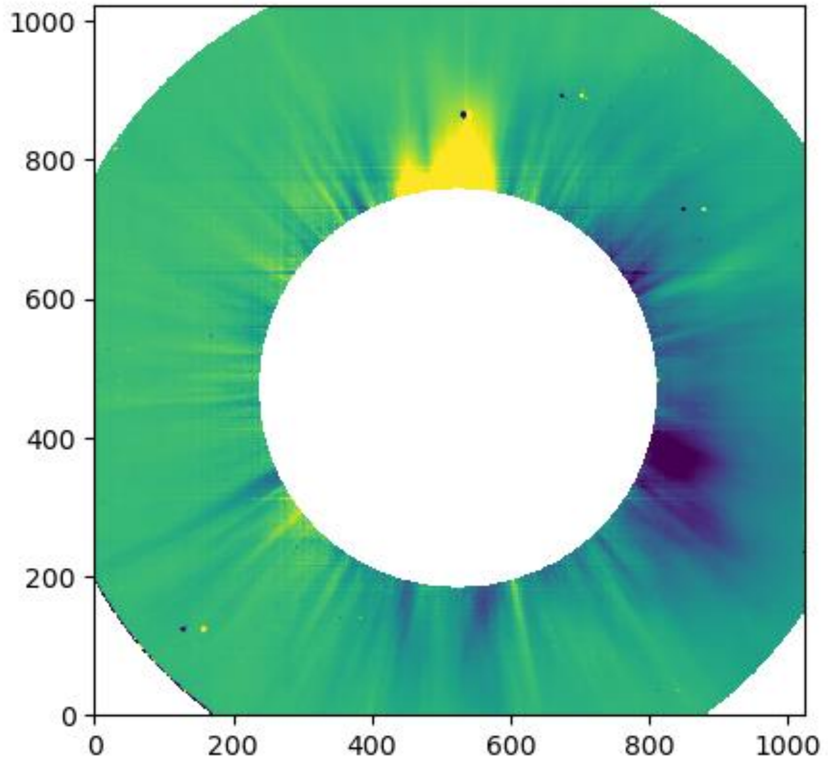


L2_metis-vl-tb_20231215T002957



zero-mean running difference

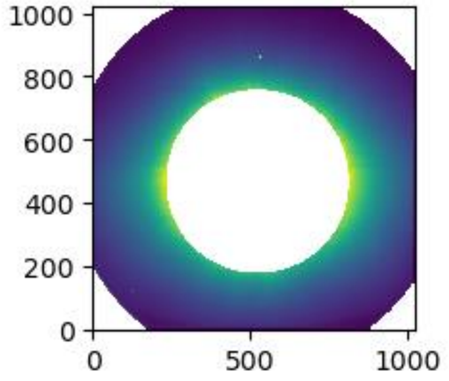
$$u = n_i - n_{i-1}\left(\frac{\overline{n_i}}{\overline{n_{i-1}}}\right)$$



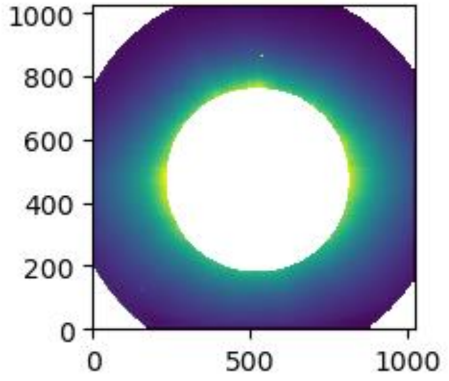
Running differences

- Running differences enhance moving features in the FOV such as Coronal Mass Ejections

L2_metis-vl-tb_20231214T193001



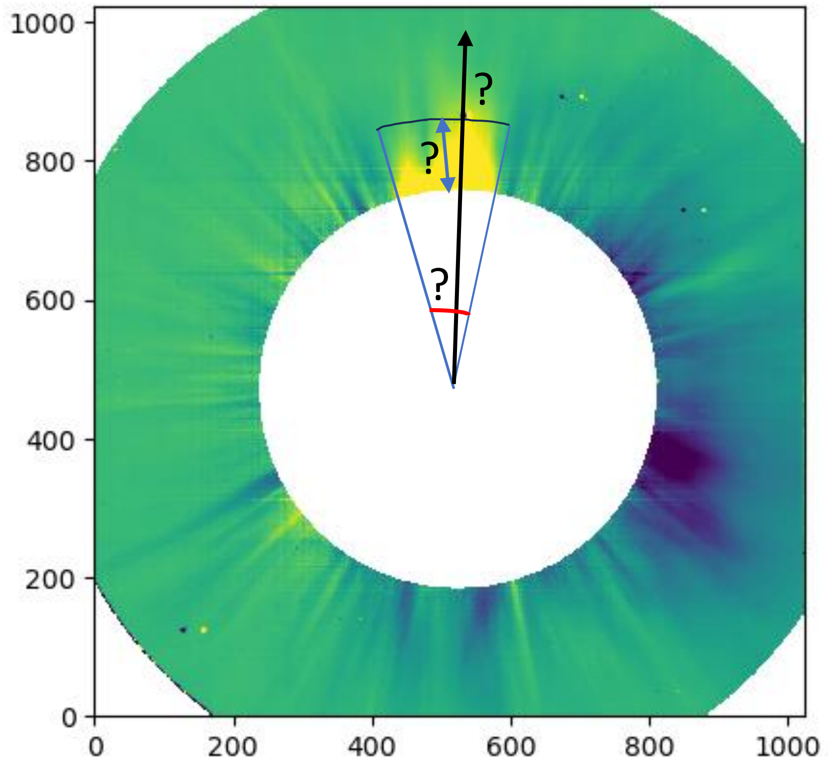
L2_metis-vl-tb_20231215T002957



zero-mean running difference

$$u = n_i - n_{i-1}\left(\frac{\overline{n_i}}{\overline{n_{i-1}}}\right)$$

<u>Goal</u>: provide an additional function to the pyCYD GUI to assist the CME hunters in assessing additional information, such as CME width, position angle, main direction and height;



Algorithm steps

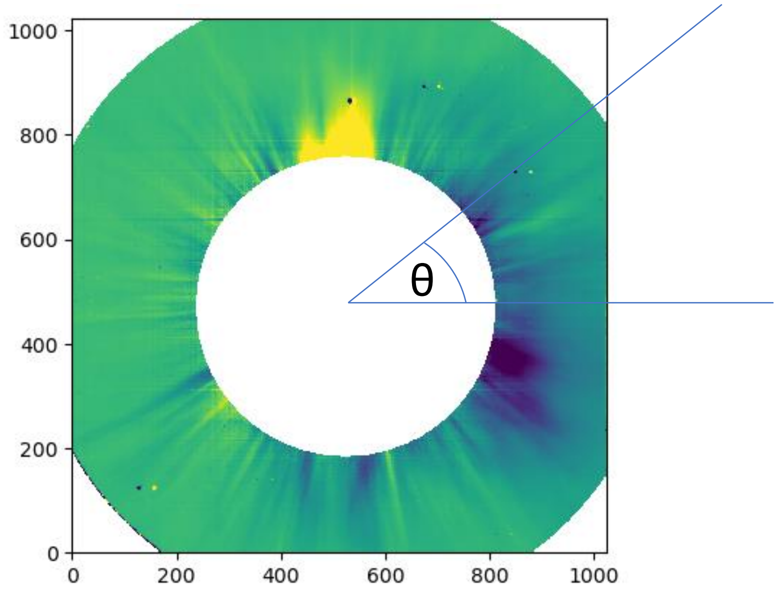
- The main idea is adapted from the CACTUS fully automatic CME detection algorithm employed to image sequences from LASCO (see Robbrecht et al. 2004, Olmedo et al. 2007, Robbrecht et al. 2009).
- We opted for a semi-automatic algorithm where two parameters are manually adjusted;

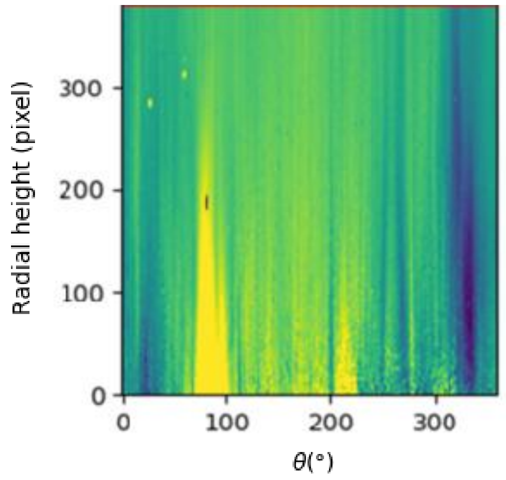
1. Conversion to polar coordinates

Algorithm steps

- The main idea is adapted from the CACTUS fully automatic CME detection algorithm employed to image sequences from LASCO (see Robbrecht et al. 2004, Olmedo et al. 2007, Robbrecht et al. 2009).
- We opted for a semi-automatic algorithm where two parameters are manually adjusted;

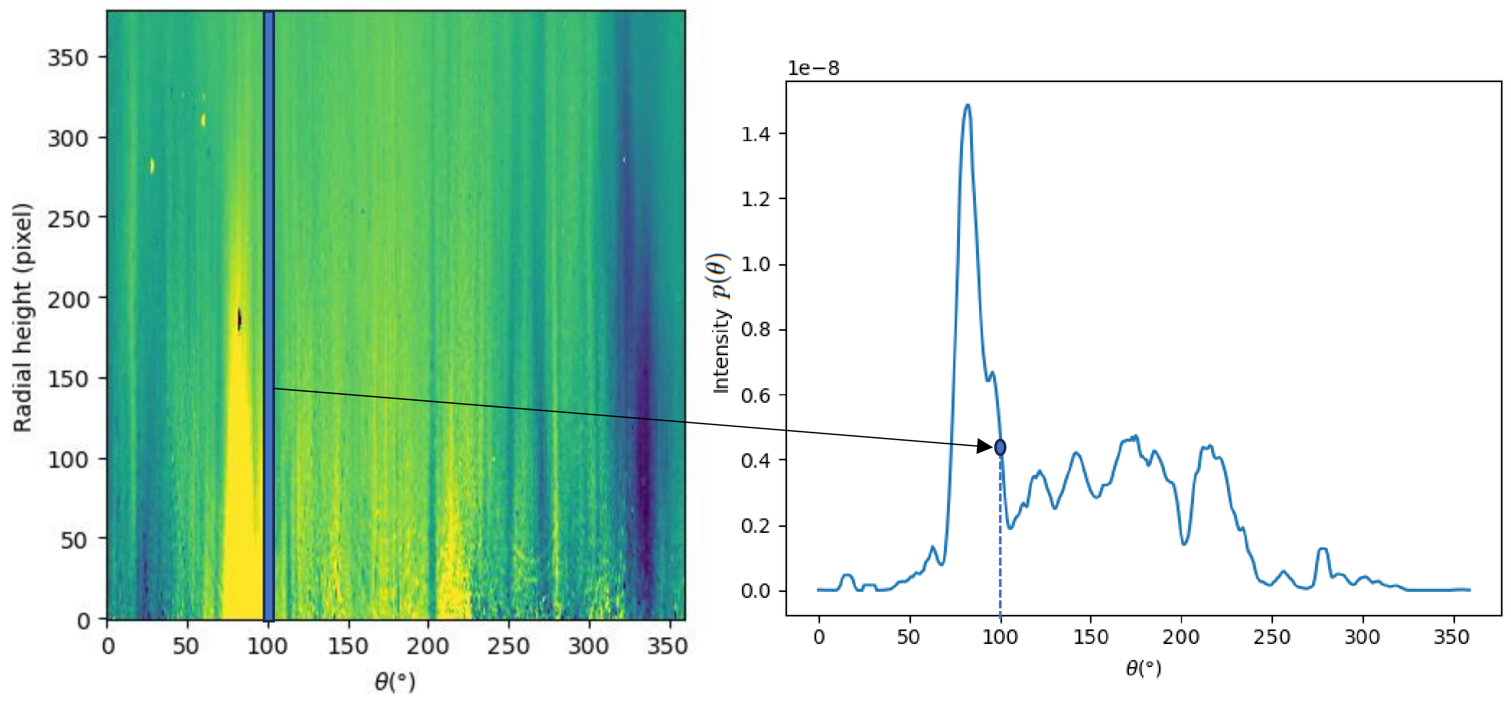
1. Conversion to polar coordinates





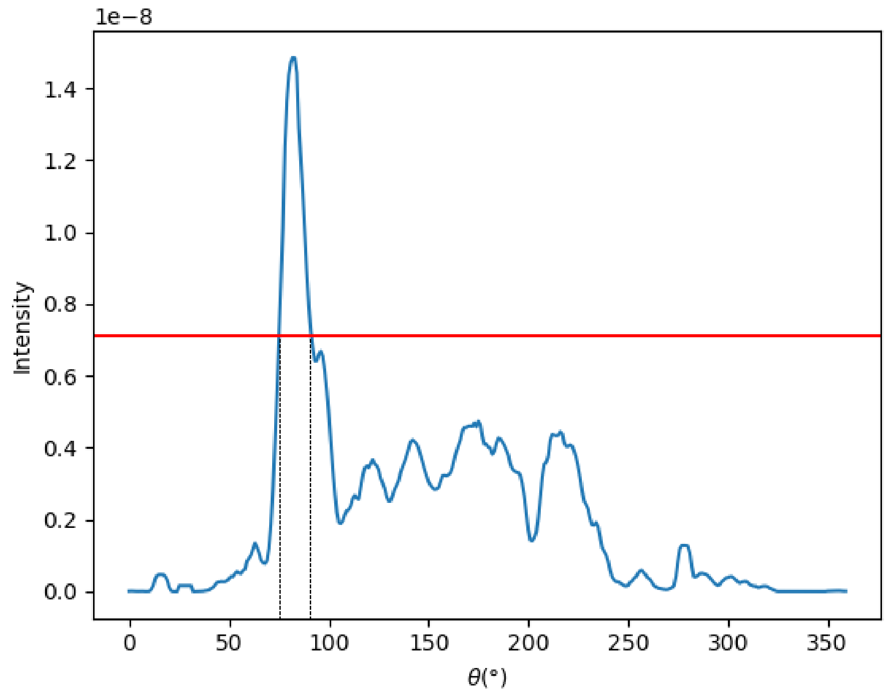
Algorithm steps

2. Compute angular intensity: for each angle (image column) positive pixels are added to construct an angular intensity map $p(\theta)$



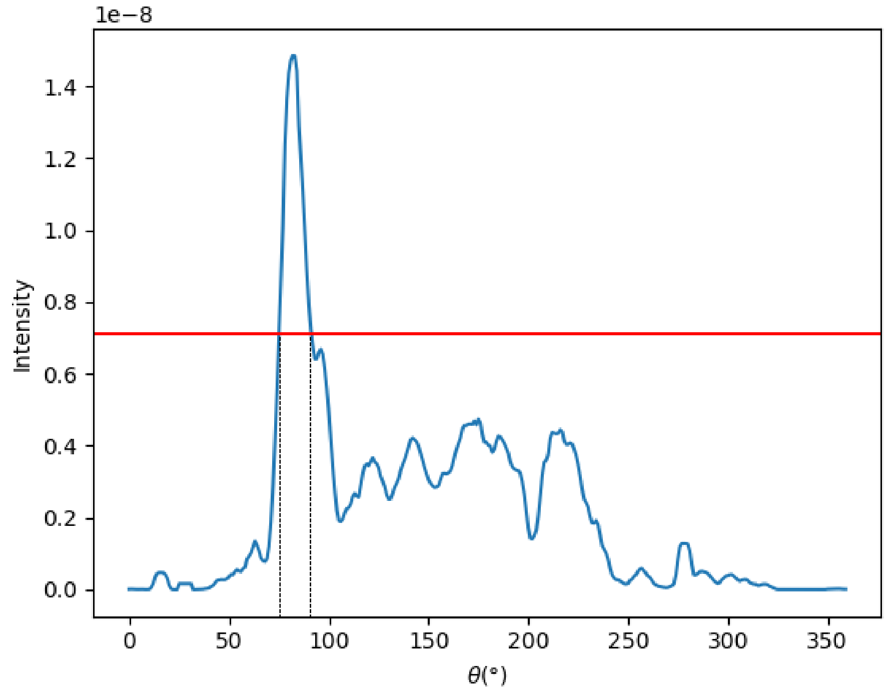
Algorithm steps

3. Define the threshold $T_1 = \overline{p} + N_1\sigma_p$ and look for intervals where $p(\theta)$ is above T1. This defines the first angular window detection where the CME is located (core region):



Algorithm steps

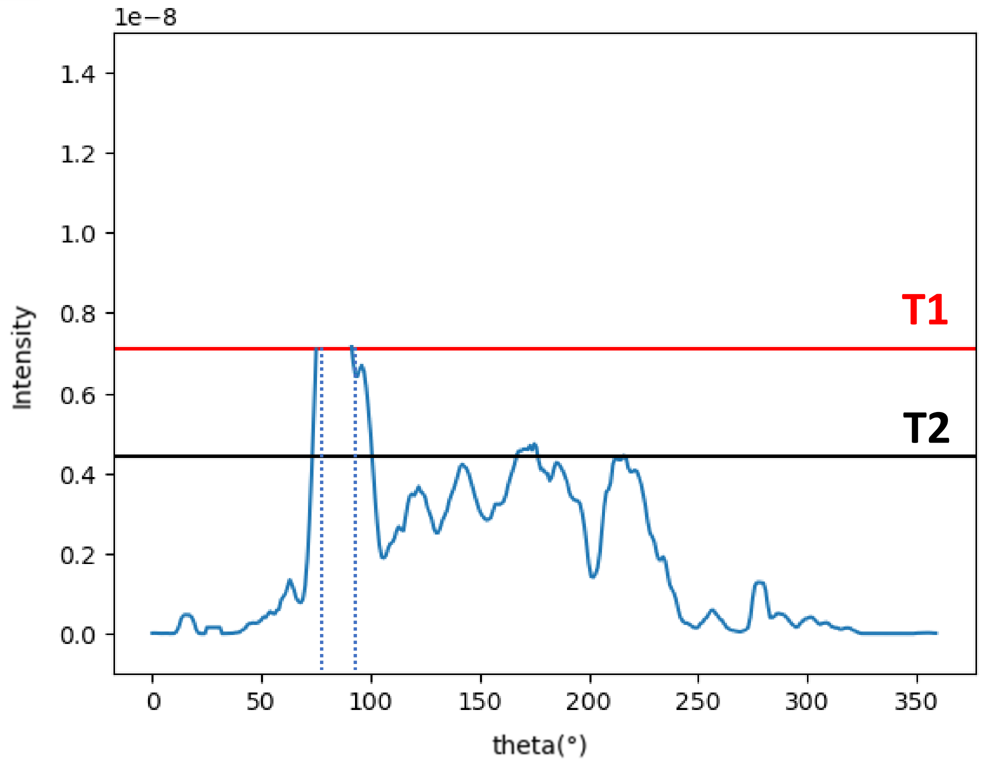
3. Define the threshold $T_1 = \overline{p} + \boxed{N_1}\sigma_p$ and look for intervals where $p(\theta)$ is above T1.
This defines the first angular window detection where the CME is located (core region):



N1 is the first user controlled parameter, defining where the brightest feature of the image will be.
A lower threshold will result in a wider core region.

Algorithm steps

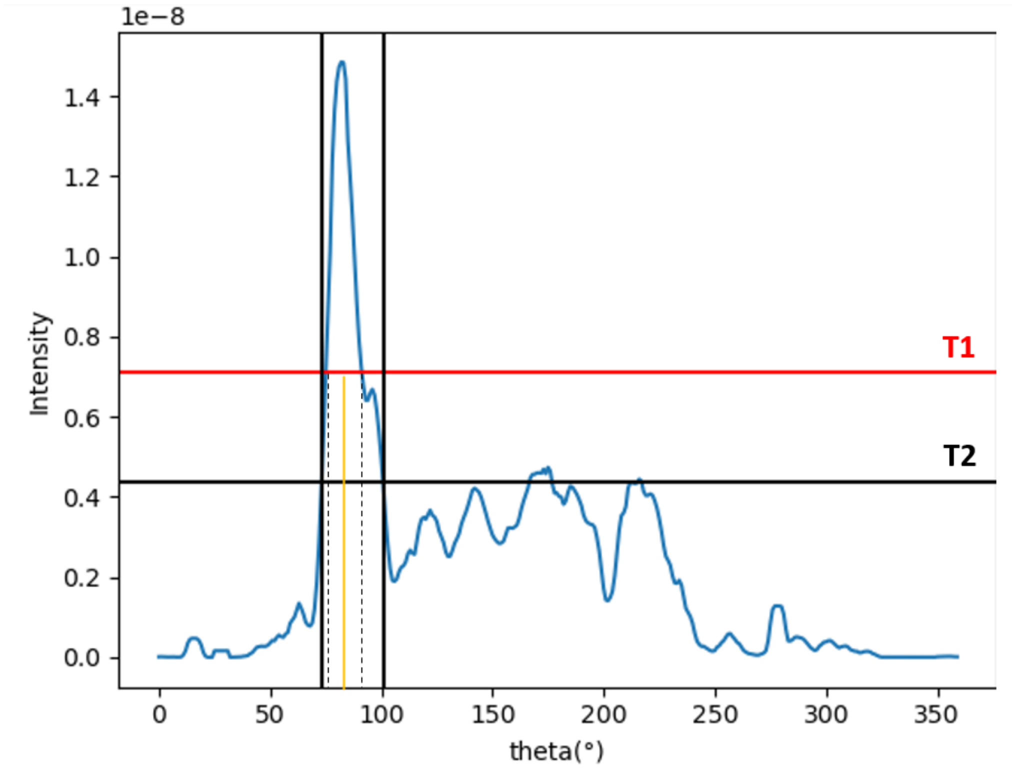
4. Increase the core region by setting a new threshold $T_2 = \overline{p^*} + N_2\,\sigma_{p^*}$ where p^* is a reduced distribution which omits values of $p(\theta)$ within the core region.



Algorithm steps

4. Increase the core region by setting a new threshold $T_2 = \overline{p^*} + N_2\,\sigma_{p^*}$ where p^* is a reduced distribution which omits values of $p(\theta)$ within the core region.

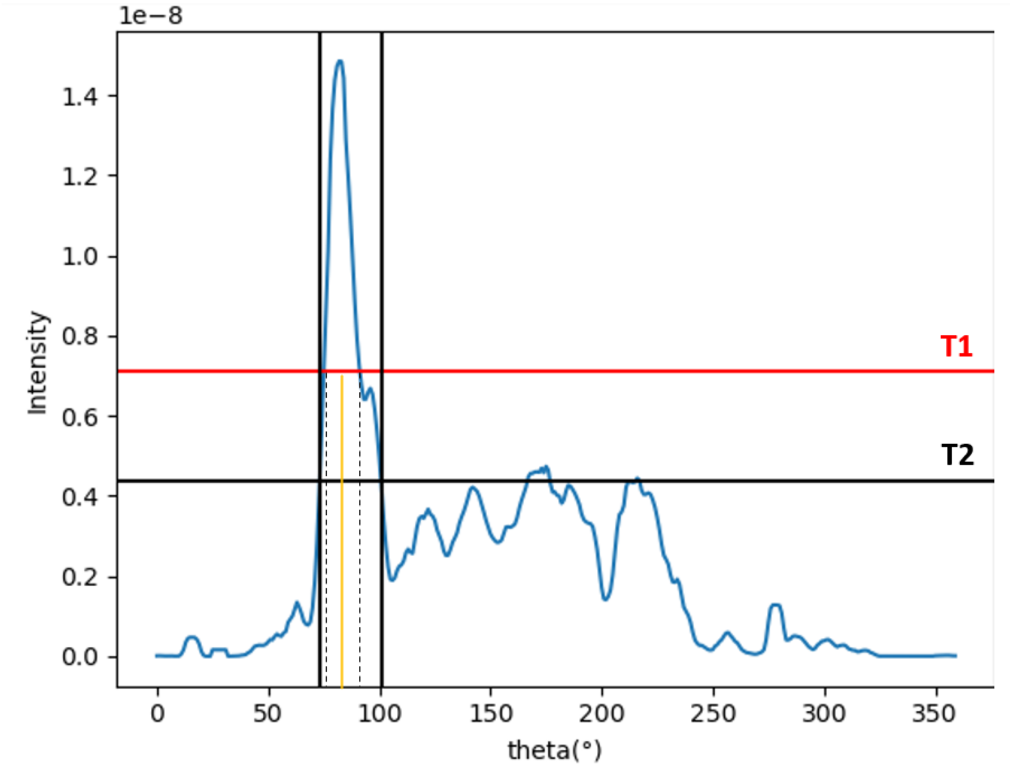
The core angle region is widened up to the angular positions where intensity falls below the new threshold T2.



Algorithm steps

4. Increase the core region by setting a new threshold $T_2 = \overline{p^*} + \boxed{N_2} \sigma_{p^*}$ where p^* is a reduced distribution which omits values of $p(\theta)$ within the core region.

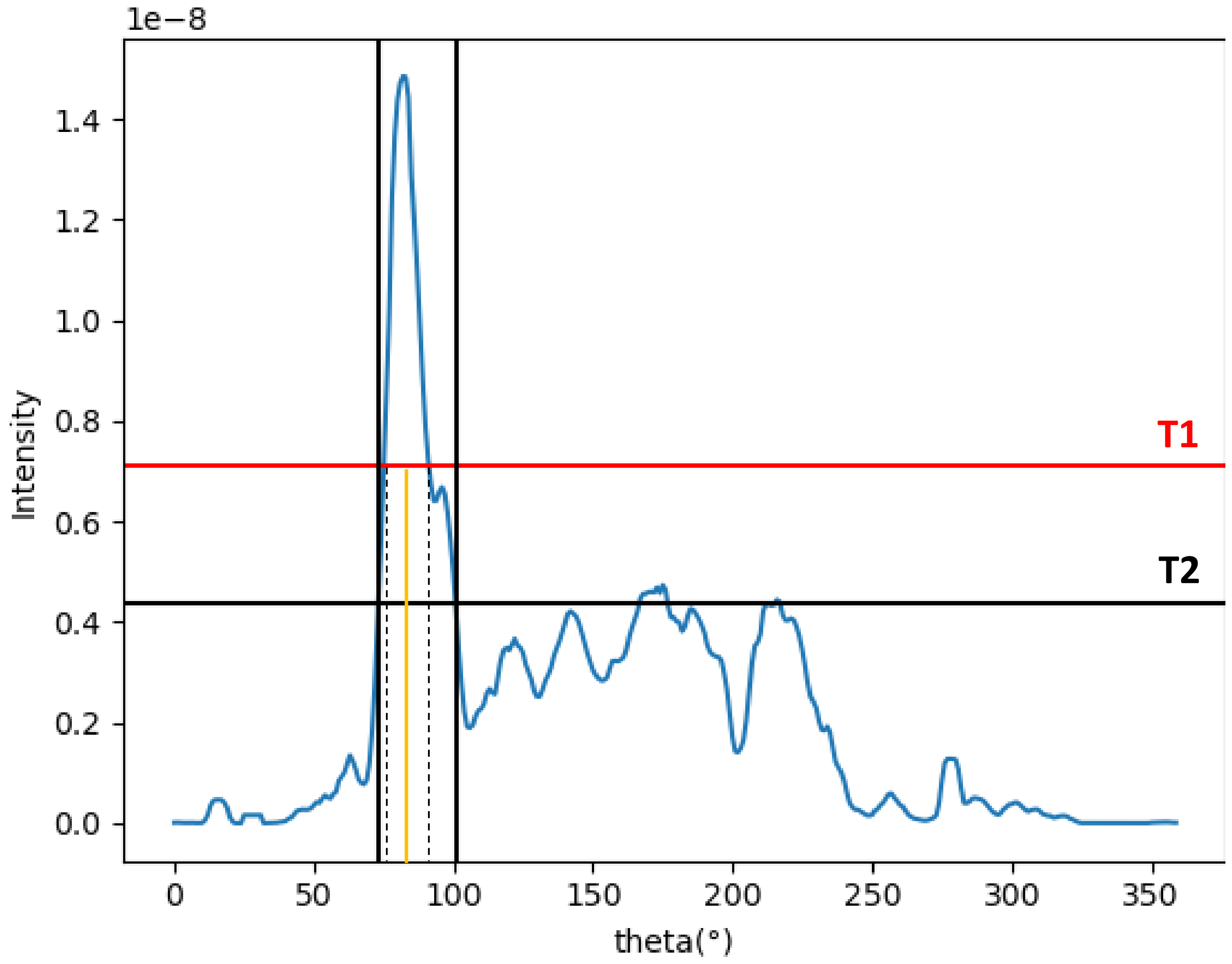
The core angle region is widened up to the angular positions where intensity falls below the new threshold T2.



N2 is the second user controlled parameter, with the effect of widening the core region to establish the final CME angular window.

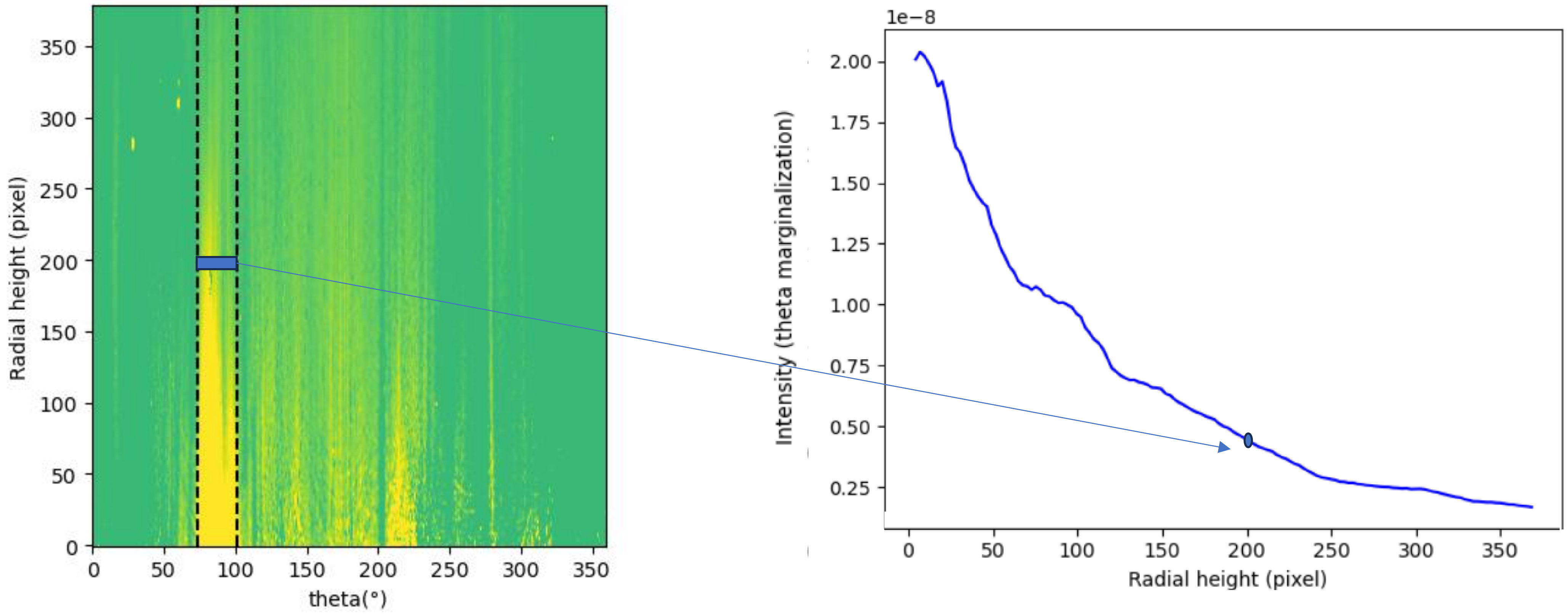
Algorithm steps

Core region center



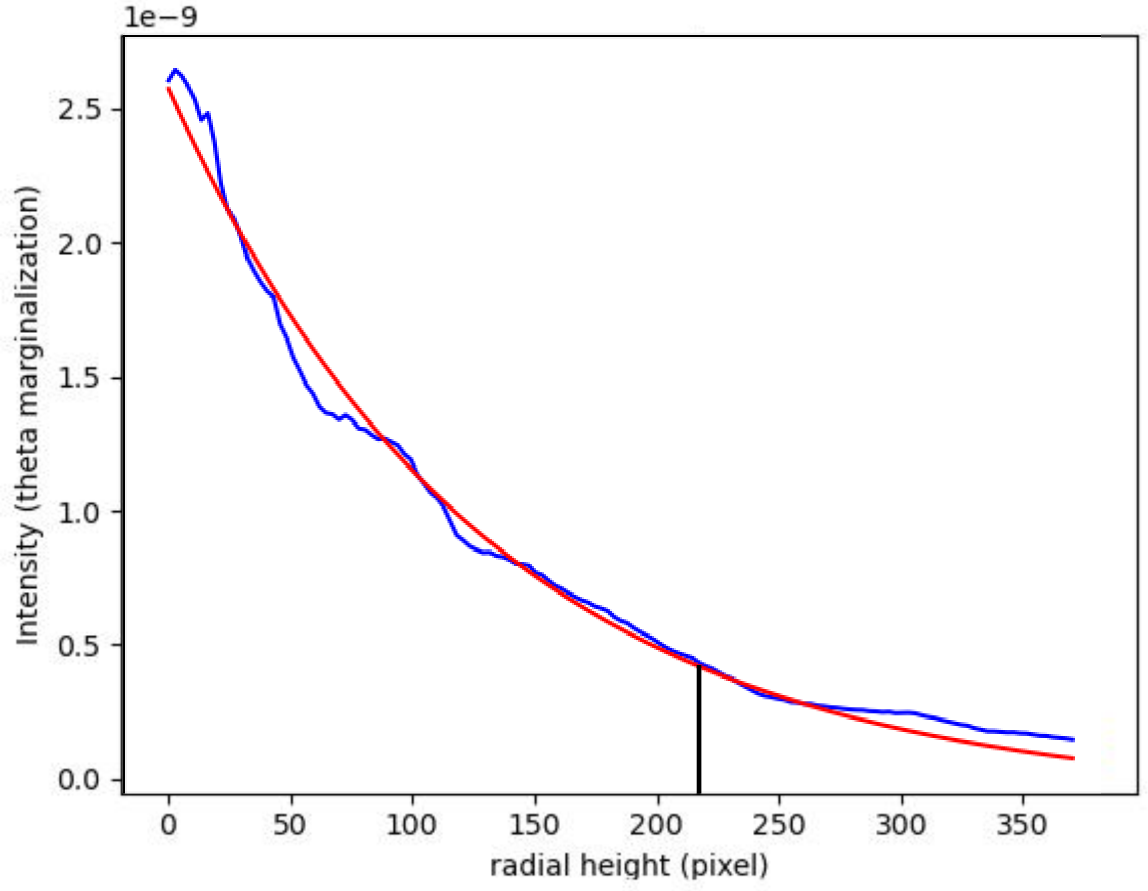
Algorithm steps

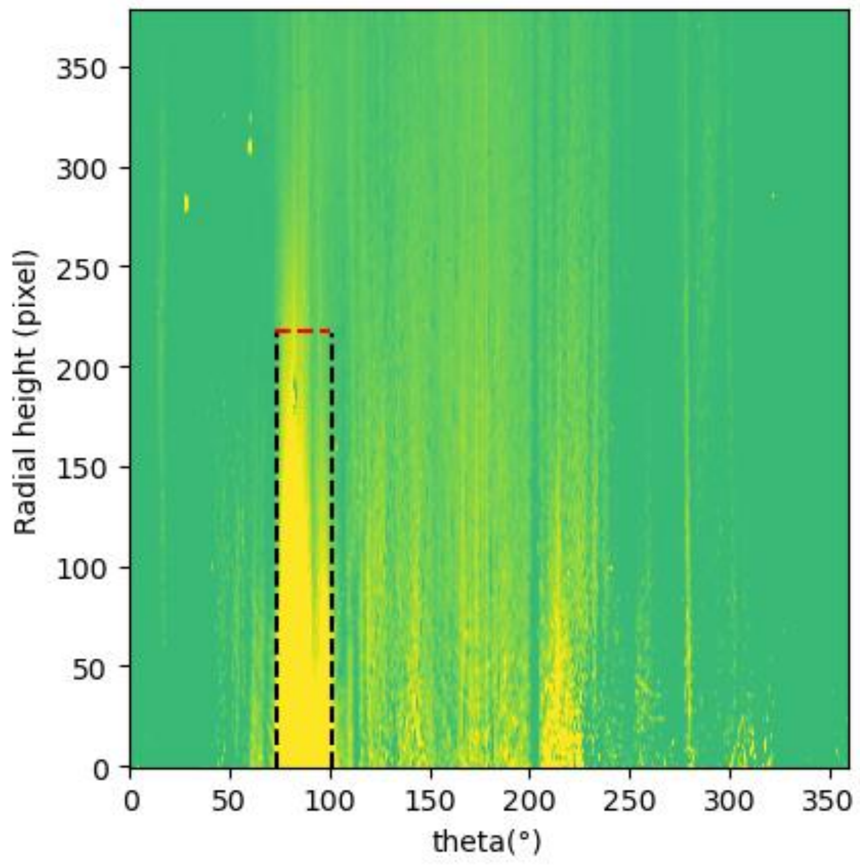
5. CME leading edge detection: within the CME core region only, fix radial heght and sum the intensity values at all angles to define the intensity distribution



Algorithm steps

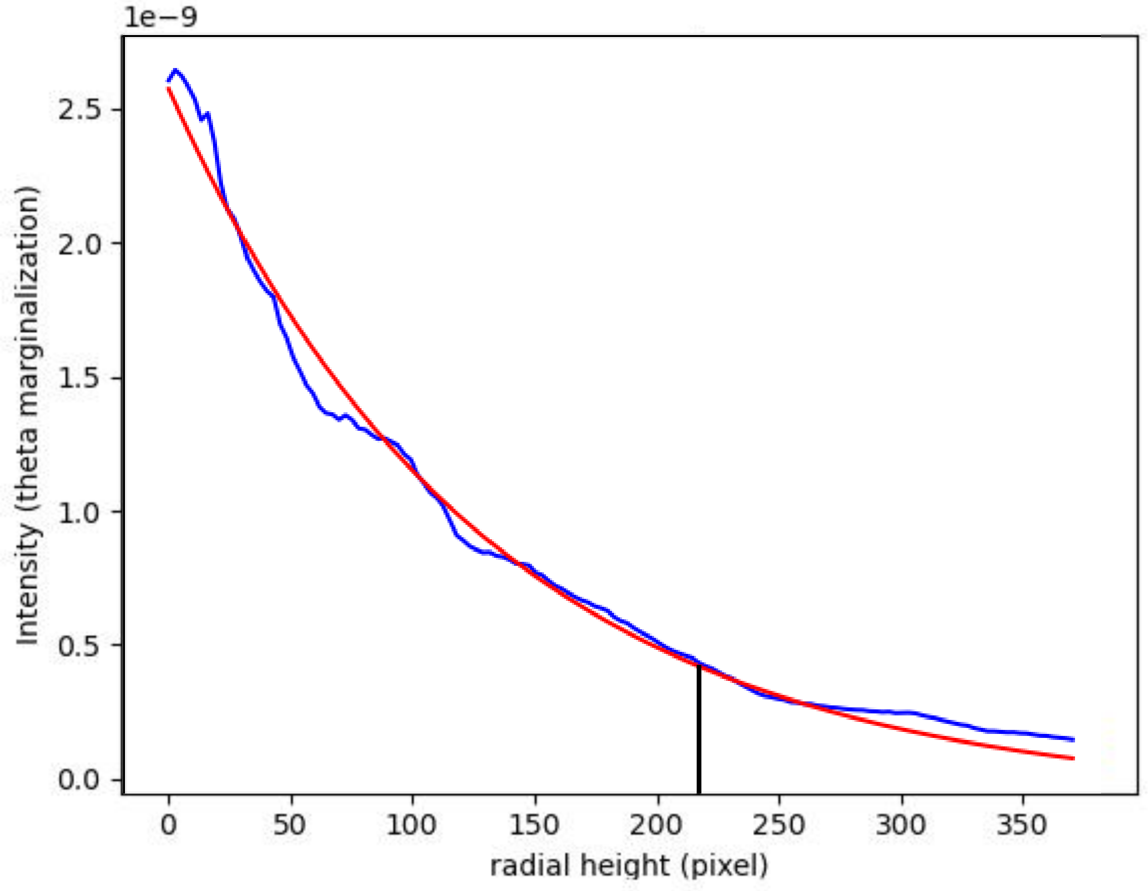
6. We then perform an exponential fit and set the CME height as 2 times the exponential scale length

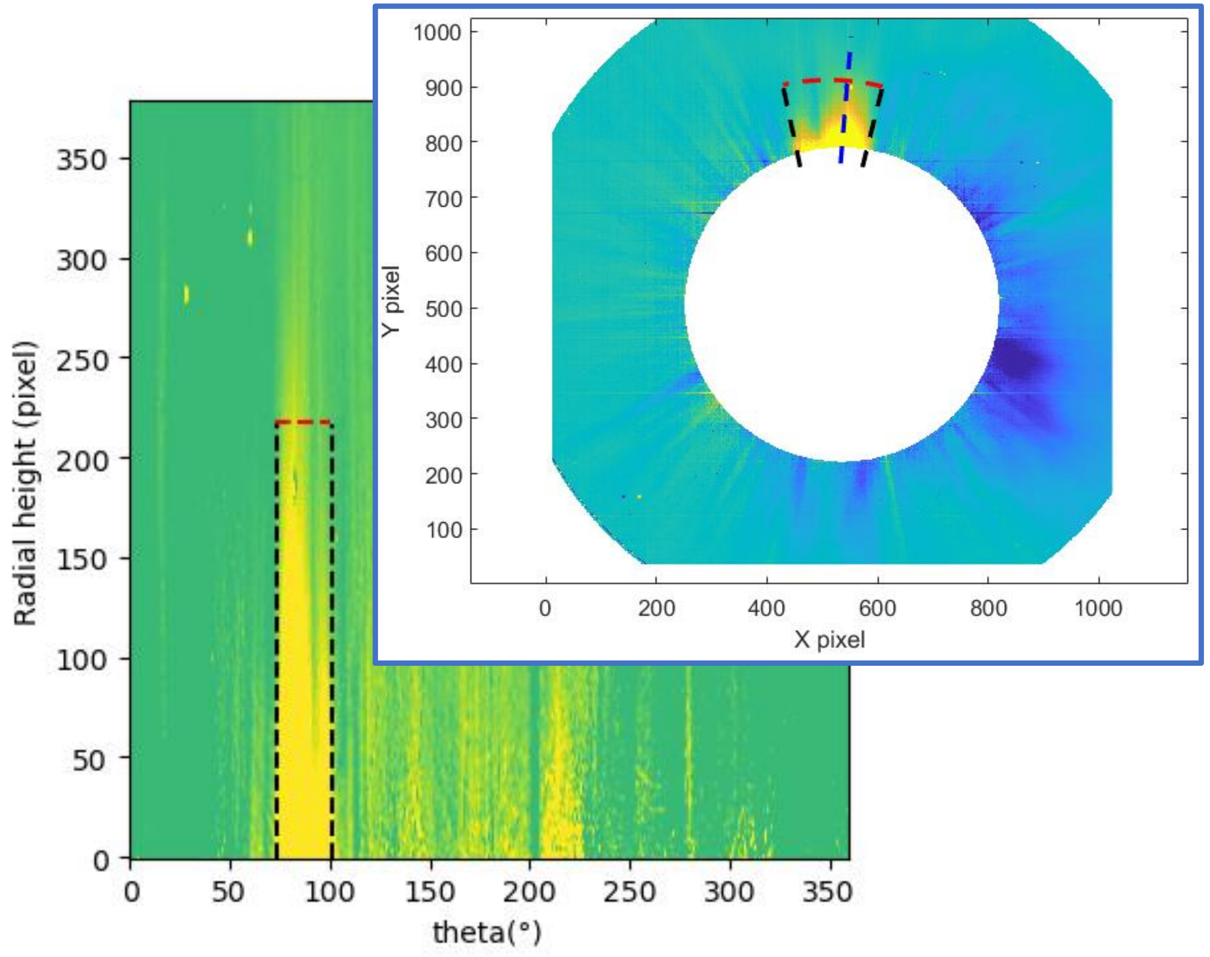


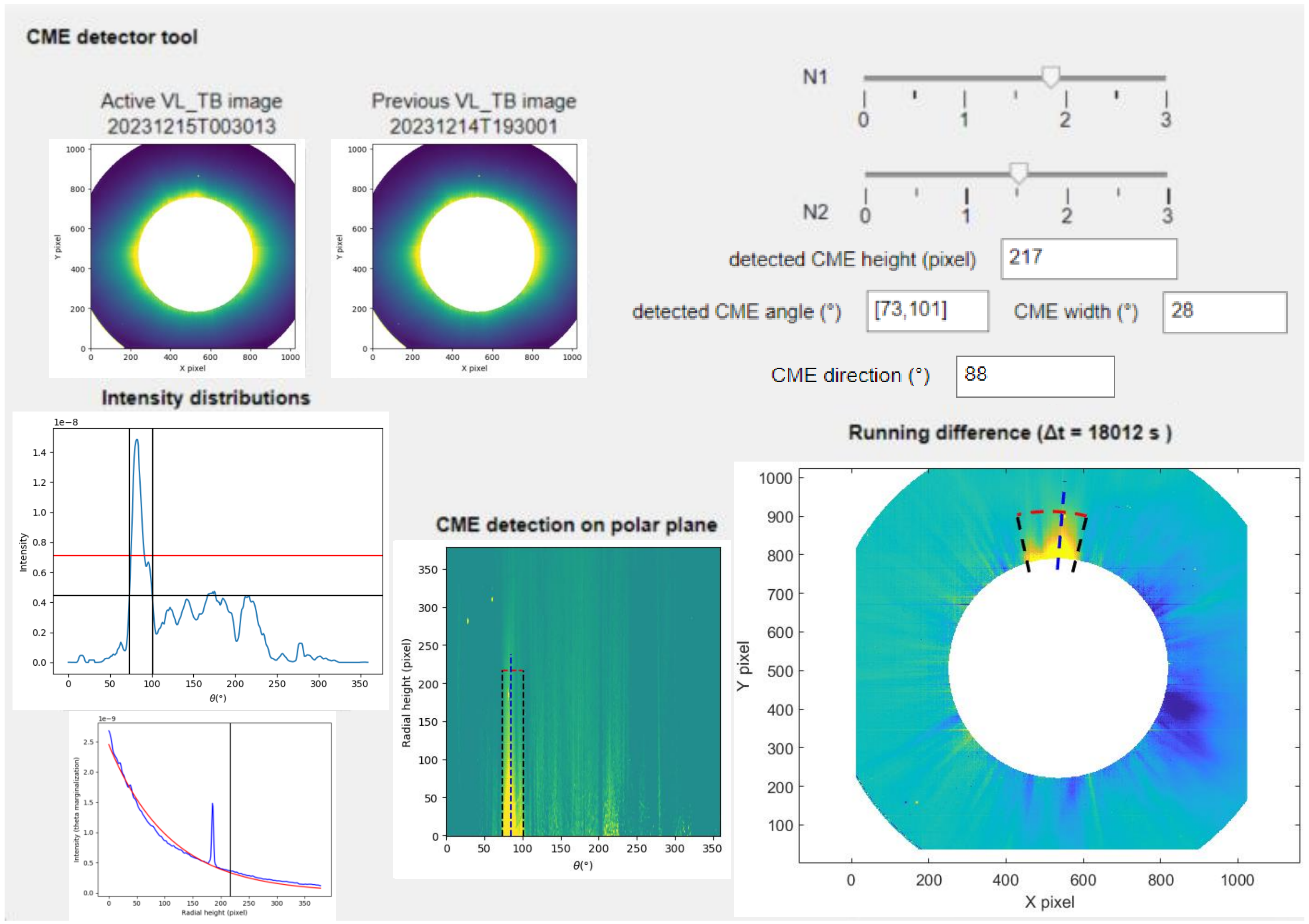


Algorithm steps

6. We then perform an exponential fit and set the CME height as 2 times the exponential scale length







CME detector tool

N1

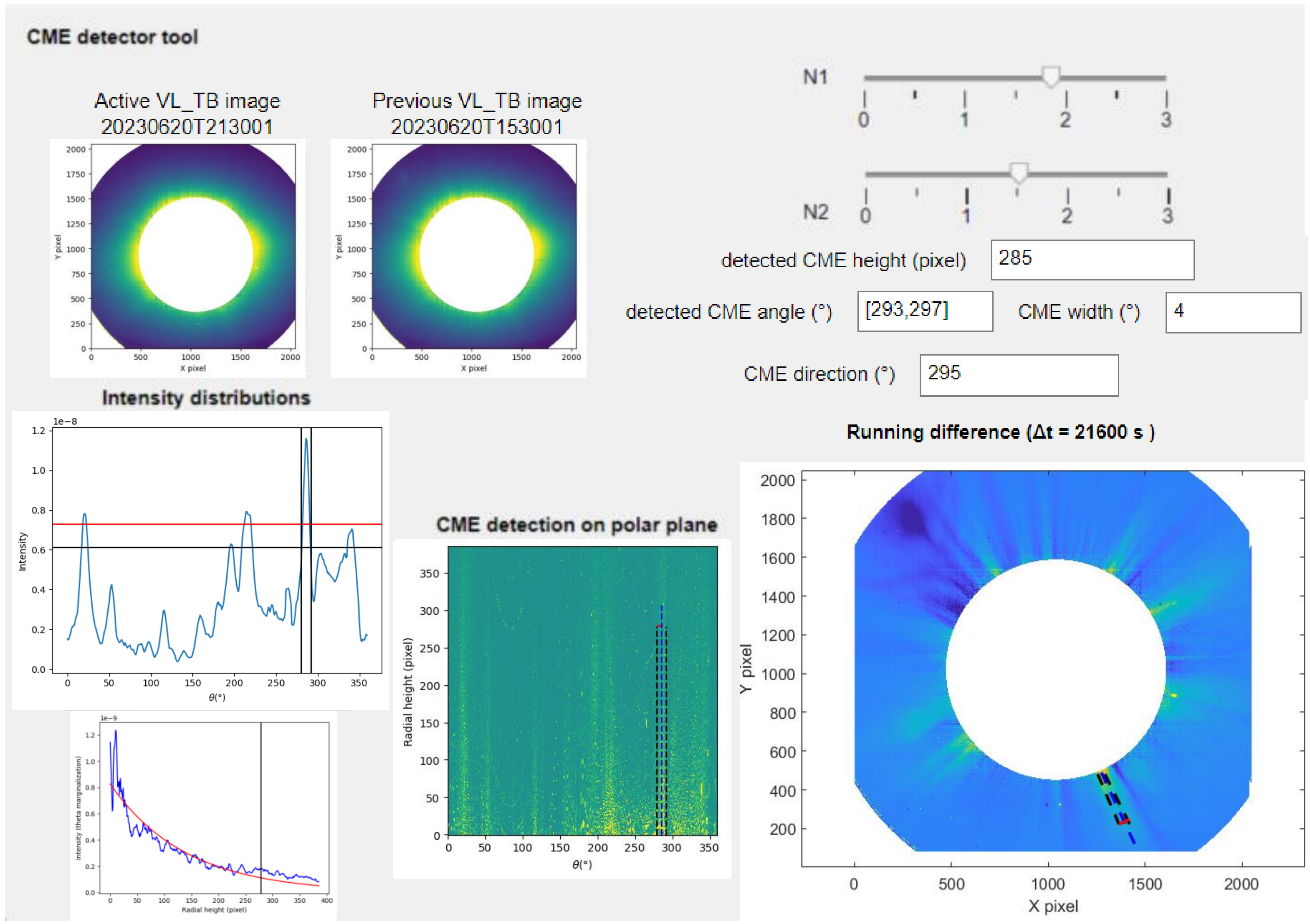
N2

detected CME height (pixel): 217

detected CME angle (°): [73,101]

CME width (°): 28

CME direction (°): 88



CME detector tool

Active VL_TB image 20230620T213001

Previous VL_TB image 20230620T153001

N1

N2

detected CME height (pixel) 285

detected CME angle (°) [293,297]

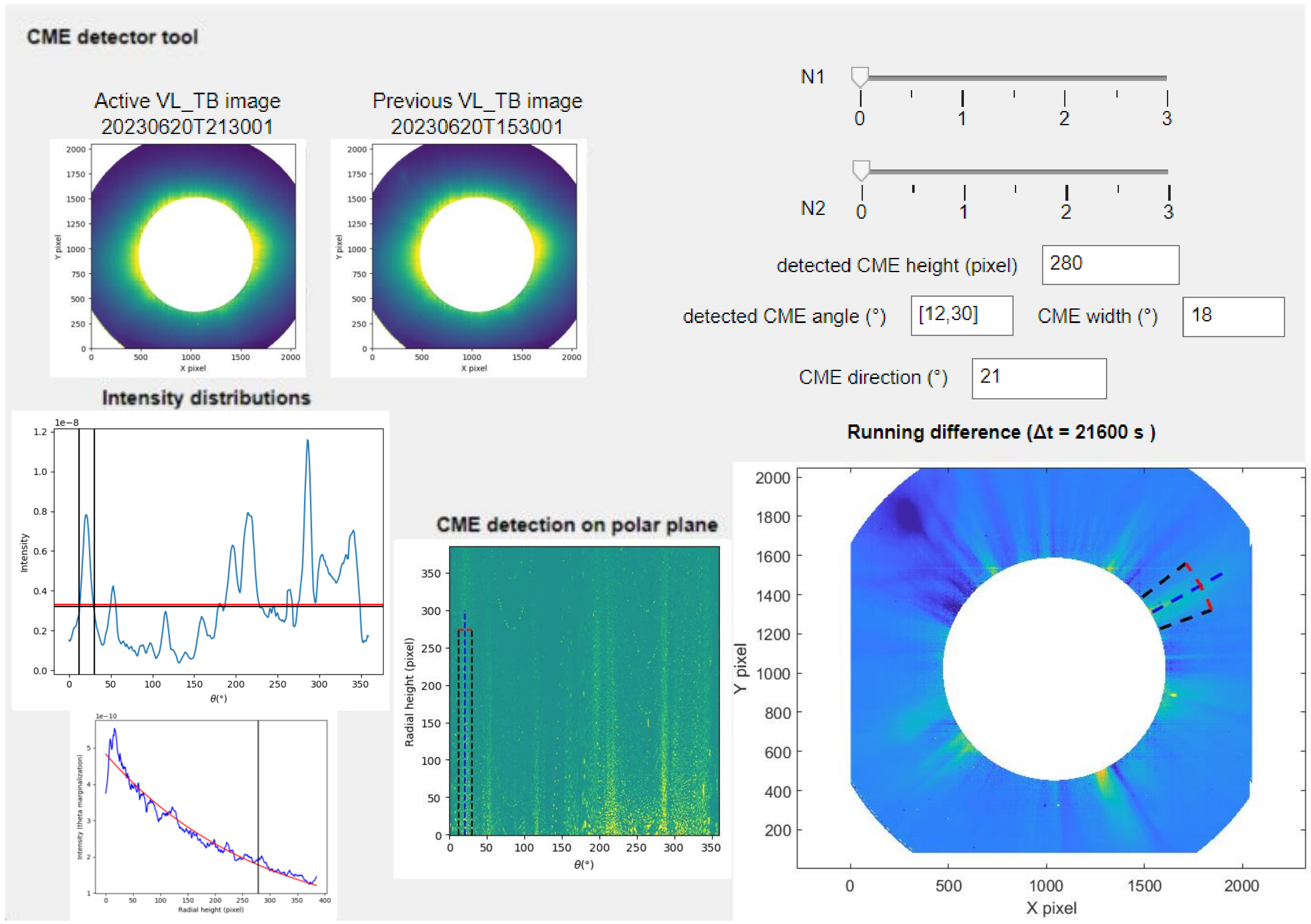
CME width (°) 4

CME direction (°) 295

Intensity distributions

CME detection on polar plane

Running difference (Δt = 21600 s)


CME detector tool

N1: 0

N2: 0

detected CME height (pixel): 280

detected CME angle (°): [12,30]

CME width (°): 18

CME direction (°): 21

To do & ongoing developement

- Regularly employ the CMEdetector to report bugs and improve it
- Adapt the algorithm behaviour to peculiar cases
- Improve the choice of the parameters (eventually make it fully automatic)

Possible applications

- CME kinematics, initial conditions for forecasting models
- CME deprojection from multiple Points of View

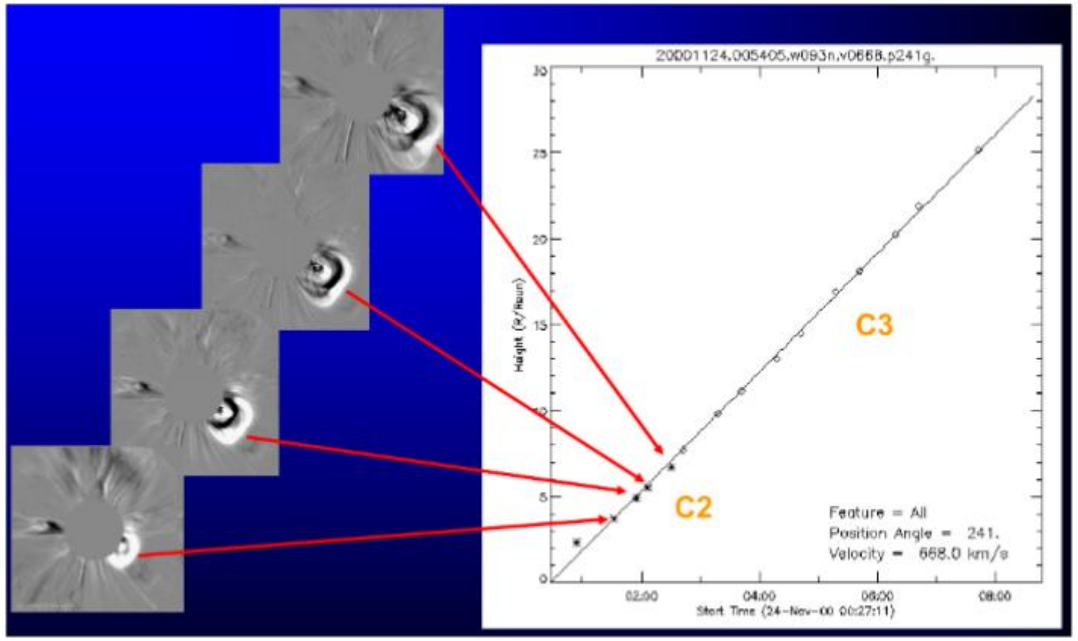
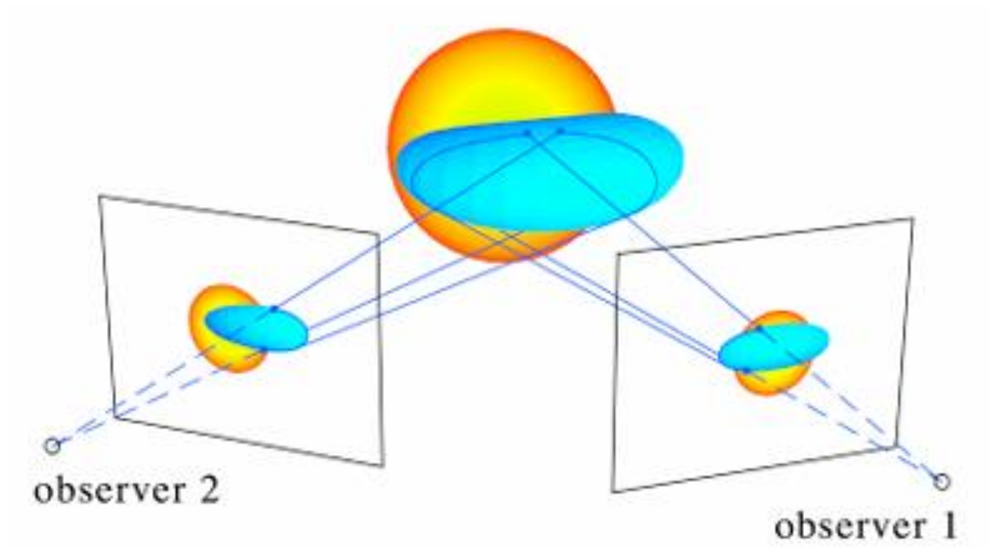
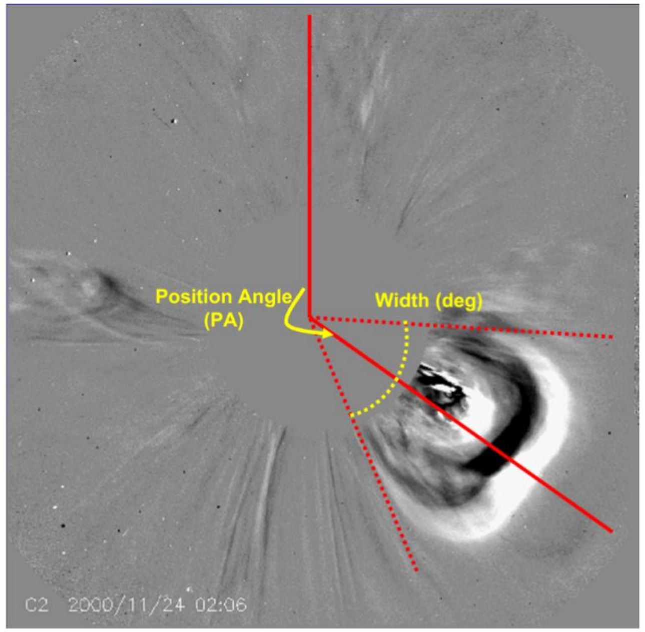


Figure 3.2: Height time measurements related to coronagraph observations and



CME detection in coronagraph FOV



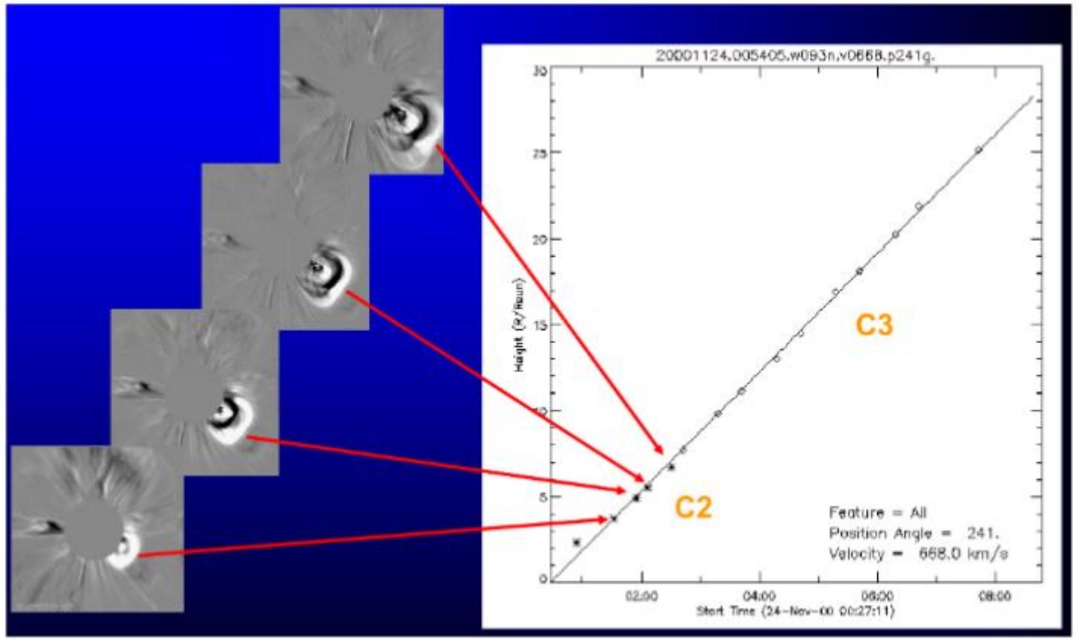


Figure 3.2: Height time measurements related to coronagraph observations and

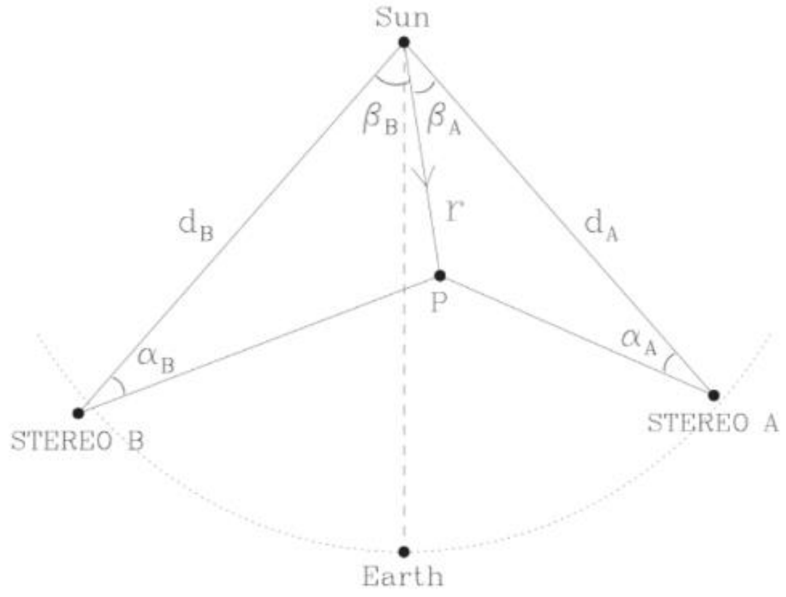
Features like CME height and width at each time are essential for:

- Studying erupting CME evolution in time
- Providing initial conditions to forecasting models
- Assess CME kinematics and morphology

The easiest deprojection model

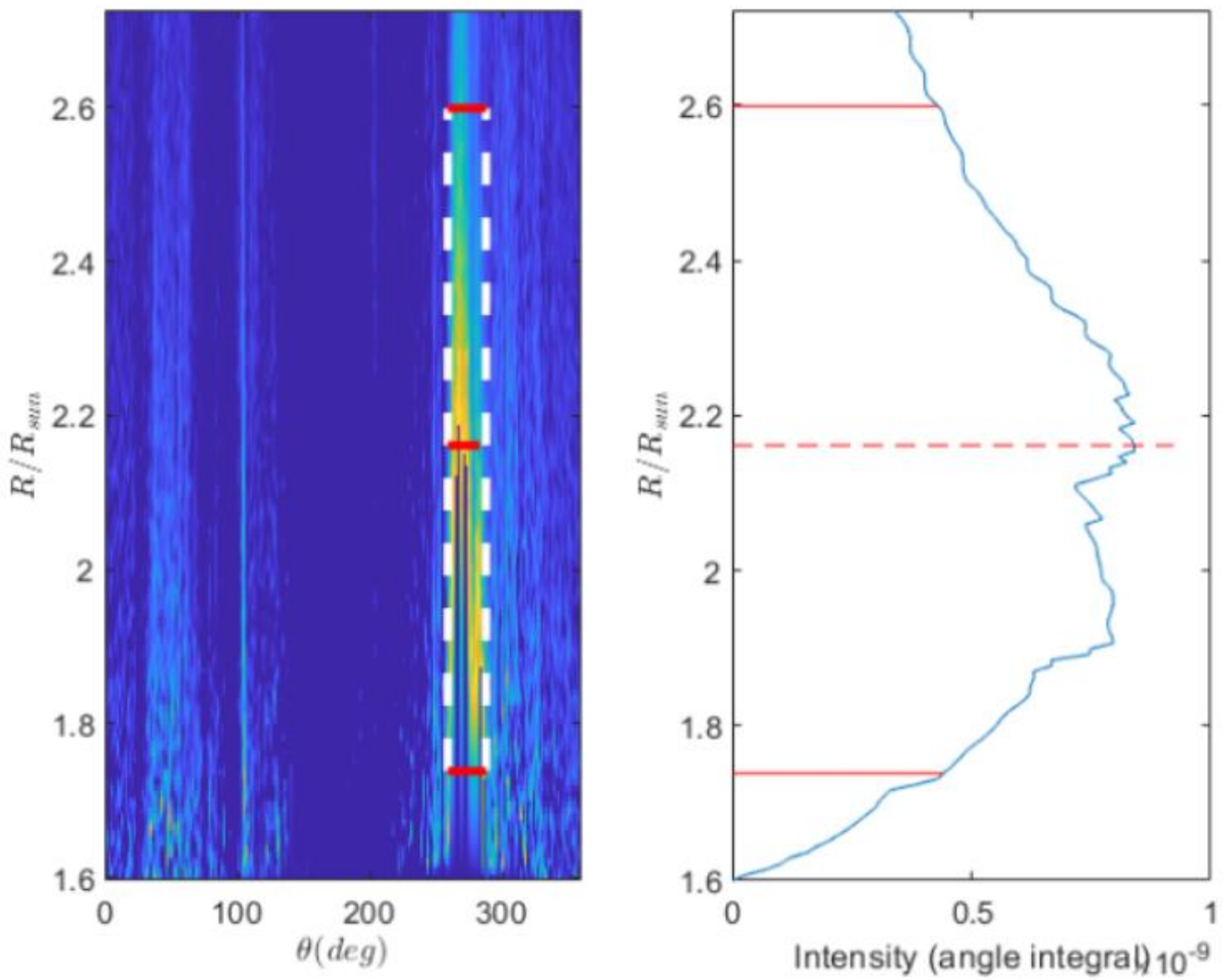
- We only take into account the farmost visible element of the detected CME
- We assume it is the same pont (discuss error with angular separation between observers)

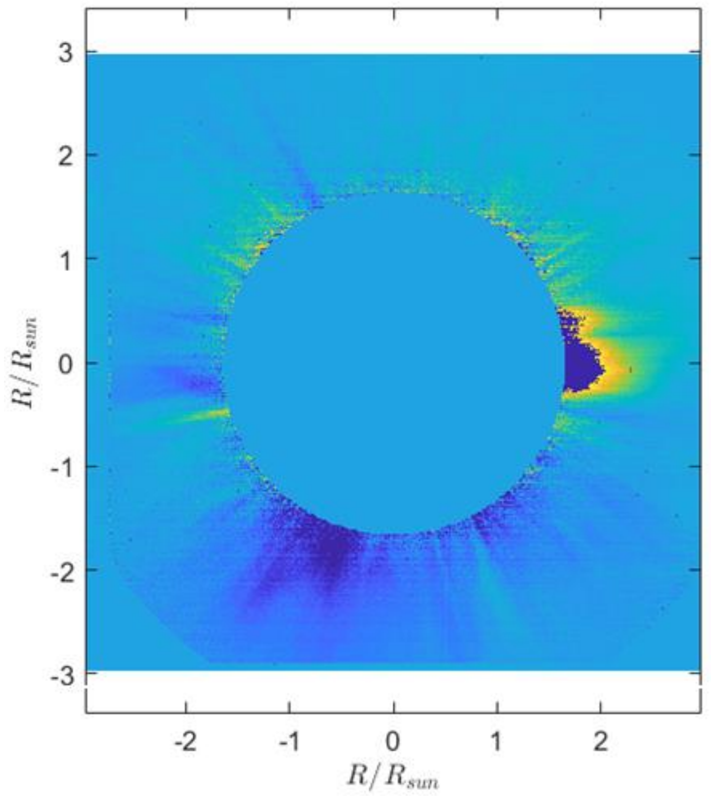
$$r\sin(\alpha_A+\beta_A) = d_A \sin\alpha_A$$
$$r\sin(\alpha_B+\beta_B) = d_A \sin\alpha_B$$
$$\beta_A+\beta_B = \gamma$$

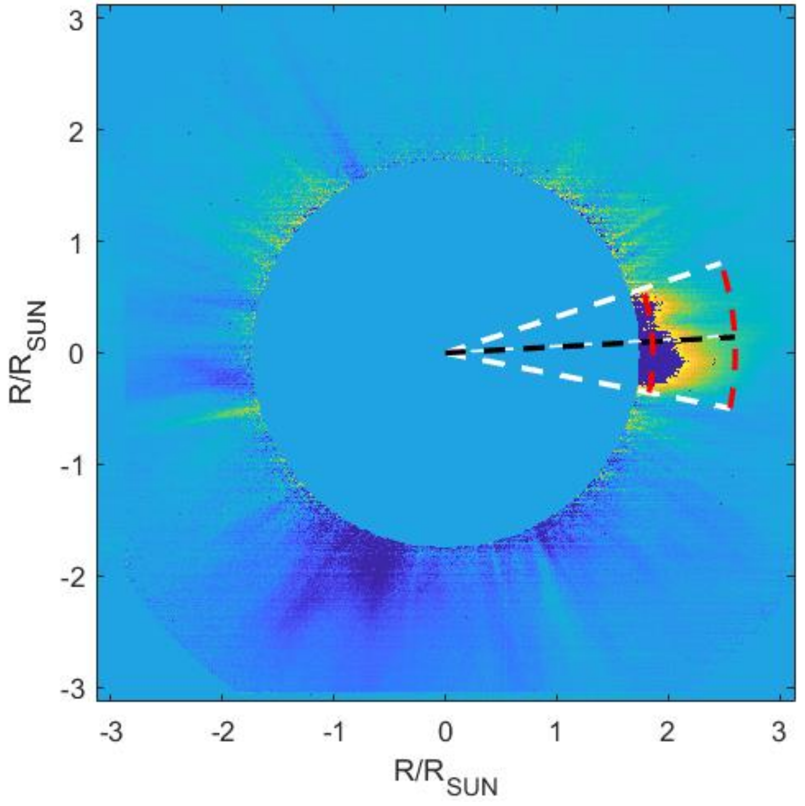


Algorithm steps

6. In the case of a maximum or bell-shape intensity distribution, take points at half maximum and define a leading and trailing edge







CME angular width
CME propagation direction (weighted average)