PROCEEDINGS OF SCIENCE

A Fast Imaging Pipeline for Transient Detection in Interferometric Data

Vladislav Stolyarov,[a,b,*] **Xiaotong Li,**[c,d] **Ian Heywood**[e,f] **and Karel Adamek**[c,g]

[a] *Astrophysics group, Cavendish Laboratory, University of Cambridge, JJ Thomson ave, CB3 0HE, UK*

[b] *Special Astrophysical Observatory of RAS, Nizhny Arkhyz, 369167, Russia*

[c] *Department of Engineering Science, University of Oxford, Keble Rd, Oxford, UK*

[d] *Department of Physics, University of Montreal, Ave.Thérèse-Lavoie-Roux, Montréal, Canada*

[e] *Astrophysics, Department of Physics, University of Oxford, Keble Road, Oxford, OX1 3RH, UK*

[f] *Centre for Radio Astronomy Techniques and Technologies, Department of Physics and Electronics, Rhodes University, PO Box 94, Makhanda, 6140, South Africa*

[g] *Department of Physics, Silesian University in Opava, Bezručovo nám. 13, Opava, Czech Republic*

E-mail: vlad@mrao.cam.ac.uk, xiaotong.li@eng.ox.ac.uk, Ian.Heywood@skao.int, karel.adamek@slu.cz

Modern radio interferometers generate vast volumes of raw data, demanding efficient processing to enable real-time or quasi-real-time astronomical analysis. A critical challenge is the detection of transient sources within wide-field interferometric images, which requires rapid imaging and robust discrimination between true transients and false positives.

In this paper, we describe a proof of concept of a Fast Imaging Pipeline designed to address this challenge. The pipeline processes snapshot visibilities to construct images in (quasi) real time, identifies point sources in difference maps, and employs cluster analysis to distinguish genuine transients from artifacts. FITrig, a GPU-accelerated transient detector integrated into the pipeline, enables the detection of transients directly from dirty images as they are produced. Developed as part of the SKA Science Data Processor prototyping efforts, this pipeline optimizes interferometric data processing in preparation for the upcoming SKA science operations. Our approach demonstrates a scalable solution for future large-scale radio surveys, balancing speed and accuracy for transient detection.

High Energy Astrophysics in Southern Africa (HEASA2025)
16-20 September, 2025
University of Johannesburg, South Africa

*Speaker

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0) All rights for text and data mining, AI training, and similar technologies for commercial purposes, are reserved. ISSN 1824-8039 . Published by SISSA Medialab.

PoS(HEASA2025)009

1. Transients in radio astronomy

Transients in astrophysics and radio astronomy are short-lived phenomena that appear and fade away, with their signatures changing rapidly over time. Transient phenomena can be separated into two broad classes, periodic and non-periodic transients. Non-periodic transients include Fast Radio Bursts (FRBs), X-ray binary flares (XRBs), supernovae (SNe) and stellar flares. On the other hand, pulsars, long-period radio transients (LPTs) including magnetars, binary systems, and highly magnetic white dwarfs, can be attributed to periodic transients.

1.1 Snapshot surveys

The important observation techniques for transient search are snapshot surveys and follow-up observations. Among the snapshot surveys one can mention the LOFAR transients survey [1], the FAST `GPPS` survey [2], `ThunderKAT` [3], `TRON` pipeline [4], the ASKAP `VAST` survey [5], GMRT Galactic Plane Pulsar and Transient Survey [6] and `realfast` VLA survey [7]. Transient snapshot surveys are also planned in the upcoming SKA project [8].

2. Fast Imaging Pipeline

The Fast Imaging Pipeline uses a variation of the difference imaging technique, widely utilized to analyze the data from the surveys mentioned in the previous section. The key component of the pipeline is gridder/invert software, which can be CPU or GPU–based. For this proof of concept, the SDP Processing Function Library[1], or PFL was chosen because it contains the `CUDA`–based implementation of the gridder and the `cuFFT`–based image construction function.

There are different strategies to analyze snapshot data when searching for transients. The algorithm which is used in the Fast Imaging Pipeline[2], or FIP, has three main stages.

In the image construction stage, the (calibrated) visibilities are used to construct snapshot dirty images. Several gridder/invert software tools were tested, e.g. `WSclean` [9], `DUCC` [10], `PFL`. Having all images created, the difference maps are produced from the subsequent snapshot dirty images.

In the source detection stage, the pipeline runs the source detection program on each difference map to construct a merged point source catalog for all detections on all difference maps. Several algorithms were tested, for example `PyBDSF` [11], `Aegean` [12], `SoFiA-2` [13], [14].

In the last stage, the pipeline performs cluster analysis and constructs a final catalog of transient candidates. Cluster analysis software is used to find neighboring sources in the merged catalog to discriminate against spurious detections. One of the tools tested is `sklearn.cluster.MeanShift` which discovers "blobs" in a smooth density of samples.

The difference maps approach automatically implies limitations on the timescales of the detected transients. Only the fastest timescale (i.e. the raw sampling timescale) is probed, since any longer–duration (smooth) transients would be distorted or completely washed out by this technique.

[1] https://developer.skao.int/projects/ska-sdp-func/en/latest/

[2] https://gitlab.com/ska-telescope/sdp/ska-sdp-exec-func-examples/-/tree/hip-315_fast_imaging_pipeline/pipelines/fast_imaging_pipeline

PoS(HEASA2025)009

2.1 Image construction

During this stage, the pipeline creates 3D datacube with the size of $N_x \times N_y \times N_{times}$ using the CASA Measurement Set as input. Here, N_x and N_y is the size of the map in the x and y directions, and N_{times} is a number of snapshots in the observation set. Different tools can be used for dirty image construction, among the tested ones are `WSclean`, `DUCC wgridder` (CPU-based), and also GPU-based `PFL` was tested and demonstrated the highest performance.

2.2 Source detection

In this stage the pipeline performs source detection on the difference frames for two subsequent snapshots. Different point source detection tools are tested, e.g. `PyBDSF`, `Aegean` and `SoFiA-2`. `SoFiA-2` shows the best performance, but since it is designed to work with the multi-frequency datacubes, not with the snapshots, it should be fine-tuned accordingly to work with single-frequency maps.

The source detection tool generates a separate detection catalog for each snapshot, containing both transient and spurious detections. The detections of the same transient are clustered around it's true position, while spurious detections are scattered all over the map. All intermediate catalogs are finally merged into a single one that contains all detections for all snapshots.

It is also noticed that maps constructed with natural and uniform weighting [15] result in different detection positions, where the uniform weighting scheme gives a less scattered source detection distribution.

Natural and uniform weighting schemes were chosen for the FIP tests as two extreme cases to demonstrate how pixel noise and the synthesized beam width affect the result of the point source detection stage. Other schemes, for example, Briggs weighting, which is a compromise between natural and uniform, can be also used, and it's robustness parameter has to be adjusted depending on the input data.

2.3 Cluster analysis and final catalog construction

The pipeline's final stage analyzes the merged detection catalogue with the point source detections from all difference maps, and extracts the positions of transient candidates. Every transient will give a group of detections with close coordinates, while spurious detections will be scattered around the maps. The Mean shift cluster analysis technique [16] is used to locate the maxima of a density function in the detection distribution (a so-called mode-seeking algorithm, `sklearn.cluster.MeanShift`[3]). In the end, a final catalog of transient candidates is created, and the candidates can be considered as the targets for the follow-up observations.

3. Application to the real data

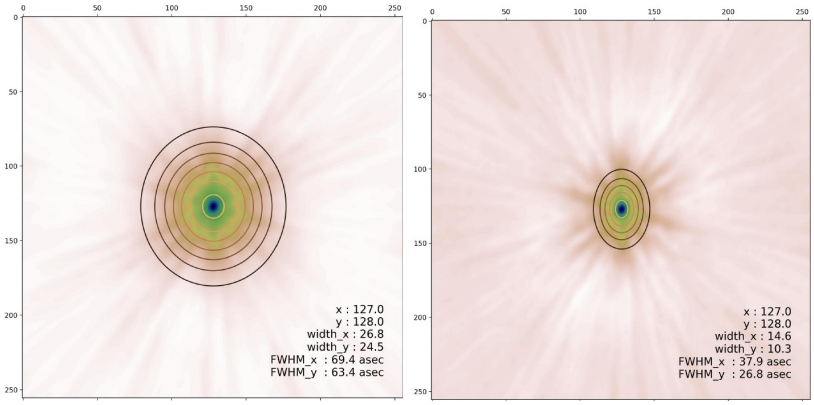
In order to test the concept of the Fast Imaging Pipeline, real MeerKAT [18] data were used. The dataset contains flagged and calibrated visibilities for the snapshot observations of a pulsar PSR J0901-4046 which has an ultra-long period of about 76 sec [17]. The dataset includes 1024

[3] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html

PoS(HEASA2025)009

frequency channels in the range 835.938 – 1712.482 MHz, with a central frequency ν_c=1283.8955 MHz, the total observing time is t_{obs}=2998 s, with N_{times}=1499 giving 2 s per snapshot.

For a "full field" fast imaging with MeerKAT we have been using 4680×4680 pixels with 1″.1 scale, which extends to approximately the 30% power point of the primary beam (1.43 deg FOV).



(a) **(b)**

Figure 1: Synthesized beams for (a) natural and (b) uniform weighting schemes on the same scale, for the MeerKAT configuration used to observe PSR J0901-4046. The FWHM for natural weighting is 69.4″ × 63.4″, and for uniform weighting is 37.9″ × 26.8″.

In Fig 1 the synthesized beams are plotted for the natural and uniform weighting schemes. It is clear that the FWHM in the case of a uniform weighting scheme is much smaller than for natural weighting, which gives better positional accuracy for the relatively bright sources in return of higher pixel noise and higher rate of spurious detections.

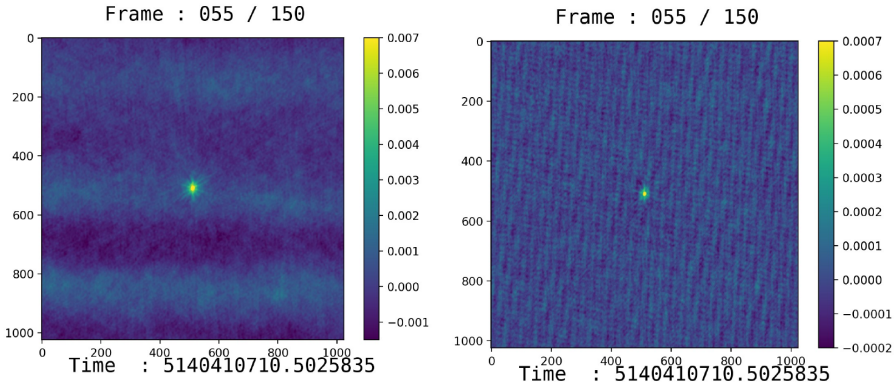
In Fig 2 snapshot images are shown for natural and uniform weighting schemes. The time per snapshot was about 0.9 s when using GPU-based PFL and NVIDIA A100 GPU, compared to 8–12 s per snapshot when using CPU–based `WSClean` on available computers. In Fig 3 maps of detections for both weighting schemes demonstrate the better accuracy of the transient position determination in the case of uniform weighting.

In the end of the pipeline, the cluster analysis algorithm confidently isolated the group of transient detections (N_{det} = 41 on 1498 difference maps) from the spurious ones, giving the correct apparent coordinates (α, δ) = (135°.3716, –40°.7678).

4. Fast Imaging Trigger

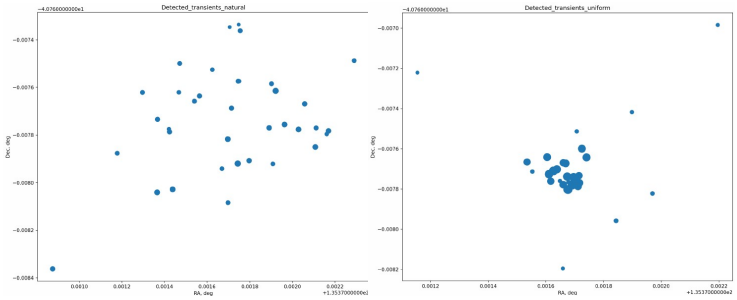
Fast Imaging Trigger (FITrig) [19] is a GPU-accelerated, statistics-based transient detector that can be embedded into the source-detection stage of the FIP, allowing transients to be detected in real time from dirty images as they are generated. Developed based on tLISI [20], a transient-oriented image assessor, FITrig provides two branches of detection (Fig 4): an image-domain branch, optimized for detecting general transients, and an image-frequency-domain branch, specifically designed to identify periodic transients. Compared with SOFIA-2, FITrig significantly improves performance: for large-scale images (e.g., 50,000 × 50,000 pixels) it achieves a 4.3× increase in

PoS(HEASA2025)009



(a) **(b)**

Figure 2: Two-second snapshot dirty images 1024×1024 pixels (a central part of full image), containing a transient constructed with (a) natural and (b) uniform weighting schemes. The scale is the same for both images, the patch size is $18'.77 \times 18'77$. Albeit the possible RFI contamination, the point source detection program can detect the transients on the difference maps with sufficient significance.



(a) **(b)**

Figure 3: The maps of detections along all 1499 frames performed with `PyBDSF`, on the subsequent frame difference maps constructed using the test MeerKAT MS, in (RA, Dec) coordinates. Circle diameters are related to the detection SNR values. The weighting schemes used are (a) natural and (b) uniform.

processing speed, while its image-domain branch reduces false positive detections (arising from non-transient sources or noise) by more than a factor of 800 at 6-σ detection threshold, with the image-frequency-domain branch providing an even greater reduction. Furthermore, FITrig retains the sensitivity to detect pulsars that are at least 20× fainter than nearby steady sources, even when operating at the Nyquist sampling limit. In short, with FITrig, the source detection stage can operate in real time, achieving higher sensitivity, higher efficiency, and lower false positive rate.

5. Conclusion

The Fast Imaging Pipeline performs quick detection of the transients, where dirty image construction can be significantly accelerated when using GPU-based gridder/invert tools. The uniform weighting scheme allows for better accuracy in source positions. Our tests confirm that `SoFiA-2` source finder seems to be the fastest among the tools tested, reducing the processing time

PoS(HEASA2025)009

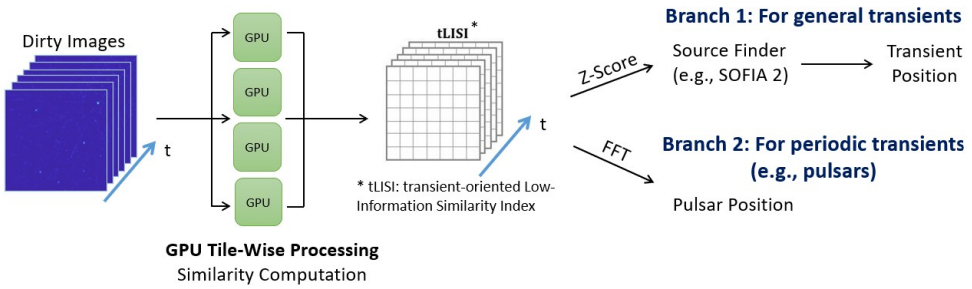


Figure 4: FITrig. Two branches are supported: image-domain and image-frequency-domain branches.

per frame of 4680×4680 pixels to 2-3 s including all three stages of the pipeline. The FITrig component is a promising addition that can significantly enhance event detection rates and enable real-time performance, pending further testing within the FIP.

Although it is only a proof of concept, the initial benchmarking of the FIP gives promising results, and the work is ongoing to convert the FIP into a useful radio astronomy tool.

References

[1] Rowlinson, A., Stewart, A. J., Broderick, J. W., et al. 2019, Astronomy and Computing, 27, 111. doi:10.1016/j.ascom.2019.03.003

[2] Han, J. L., Wang, C., Wang, P. F., et al. 2021, Research in Astronomy and Astrophysics, 21, 5, 107. doi:10.1088/1674-4527/21/5/107

[3] Fender, R., Woudt, P. A., Corbel, S., et al. 2018, MeerKAT Science: On the Pathway to the SKA, Proceedings of Science, PoS(MeerKAT2016)013. doi:10.22323/1.277.0013

[4] Smirnov, O. M., Heywood, I., Geyer, M., et al. 2025, MNRAS, 538, 1, L62. doi:10.1093/mnrasl/slaf009

[5] Murphy, T., Kaplan, D. L., Stewart, A. J., et al. 2021, PASA, 38, e054. doi:10.1017/pasa.2021.44

[6] Surnis, M. P., Joshi, B. C., McLaughlin, M. A., et al. 2018, MNRAS, 478, 4, 4433. doi:10.1093/mnras/sty1301

[7] Law, C. J., Bower, G. C., Burke-Spolaor, S., et al. 2018, ApJS, 236, 1, 8. doi:10.3847/1538-4365/aab77b

[8] Colegate, T. M. & Clarke, N. 2011, PASA, 28, 4, 299. doi:10.1071/AS11031

[9] Offringa, A. R., McKinley, B., Hurley-Walker, N., et al. 2014, MNRAS, 444, 1, 606. doi:10.1093/mnras/stu1368

[10] Reinecke, M. 2020, Astrophysics Source Code Library. ascl:2008.023

[11] Mohan, N. & Rafferty, D. 2015, Astrophysics Source Code Library. ascl:1502.007

PoS(HEASA2025)009

[12] Hancock, P. J., Trott, C. M., & Hurley-Walker, N. 2018, PASA, 35, e011. doi:10.1017/pasa.2018.3

[13] Serra, P., Westmeier, T., Giese, N., et al. 2015, MNRAS, 448, 2, 1922. doi:10.1093/mnras/stv079

[14] Westmeier, T., Kitaeff, S., Pallot, D., et al. 2021, MNRAS, 506, 3, 3962. doi:10.1093/mnras/stab1881

[15] Sault, R. J. & Oosterloo, T. A. 2007, arXiv, astro-ph/0701171. doi:10.48550/arXiv.astro-ph/0701171

[16] Comaniciu, D. and Meer, P. 2002, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 5, 603. doi:10.1109/34.1000236

[17] Caleb, M., Heywood, I., Rajwade, K., et al. 2022, Nature Astronomy, 6, 828. doi:10.1038/s41550-022-01688-x

[18] Jonas, J. & MeerKAT Team 2018, MeerKAT Science: On the Pathway to the SKA, Proceedings of Science, PoS(MeerKAT2016)001. doi:10.22323/1.277.0001

[19] Li, X., Adamek, K., Armour, W. 2025, arXiv, astro-ph/2509.21754, doi:10.48550/arXiv.2509.21754

[20] Li, X. 2024, *Image Quality Assessment-Based Fast Imaging Pipeline for Transient Detection with GPU Acceleration*, PhD thesis, University of Oxford. doi:10.5287/ora-g7nnok9o9

PoS(HEASA2025)009